中国人民大学
RENMIN UNIVERSITY OF CHINA

WAMDM
Innovative Data Systems Research

2010
ANNUAL REPORT

网络与移动数据管理实验室

Lab of Web&Mobile Data Management

2011 年 1 月

中国人民大学
RENMIN UNIVERSITY OF CHINA

2010
ANNUAL REPORT

网络与移动数据管理实验室
Lab of Web&Mobile Data Management

2011 年 1 月

序

与传统的谚语“三十年河东、三十年河西”相比，计算技术的变革突飞猛进，似乎呈现着“十年河东、十年河西”的规律，微软ð 谷歌ð 脸谱的发展大致可以印证这一事实。

因此从事这个领域的研究以十年一个周期布局自己的研究方向应较为合理，变换周期太快不宜成果的积累，变化周期太长容易跟不上发展的节奏，从而与主流技术脱节。这与“Outliers: The story of success”一书中的 10 年一万个小时专注做一件事的秘诀是一致的。

新世纪以来，数据库界普遍面临的一个问题是，在传统的数据库技术成熟之后，数据库研究应向何处去？凭借自己对当时技术趋势的判断，我将研究目标定位在解决数据库技术与 Web 计算和移动计算交叉结合所产生的挑战性问题，即结构多样的 Web 数据管理，半结构化 XML 数据的管理，以及移动环境下的数据管理问题，并创立了“网络与移动数据管理实验室（**Web and Mobile Data Management**）”，致力于这方面的研究，取得了一些国内外所共知的研究成果。我把这一阶段的研究概括为创新数据管理研究 **1.0**。

今年是又一个十年的伊始，我一直在思索实验室下一个十年的研究布局。我们不难发现数据库技术的变革（其实任何信息技术亦如此）主要来自三方面的驱动力，即：计算模式，硬件技术，应用模式的不断创新。基于新的三方面驱动力的需求，把对下一个十年的研究概括为创新数据管理研究 **2.0**，具体包含如下的研究方向：

1）闪存数据库系统的研究。它来自硬件技术变革的驱动力，其研究目标是针对闪存硬件特性、灵活的应用模式和传统数据库技术的不足，研究全新的闪存数据库管理技术；

2）云数据库系统的研究。它来自计算模式变革的驱动力，其研究目标是实现一种具有灵活配置、高可用性、高容错性、可扩展性和高性能的云数据库系统；

3）**Web** 与社会计算的研究。它来自应用模式变革的驱动力，其研究目标是把社会计算的方法引入 Web 数据管理，解决 Web 信息的可信和隐私保护问题；

4）**Mobile** 与隐私保护研究。它来自应用模式变革的驱动力，即 Mobile Web 需求日益迫切，其研究目标是解决移动搜索、隐私保护等关键问题；

5）纯 **XML** 数据库系统研制。过去 10 年我们系统研究了纯 XML 数据库技术，获中国计算机学会“王选奖”，我们将寻求技术转移的途径，进行产业化尝试。

数据库系统发展经历了三十年，大致呈现出了“分久必合、合久必分”规律。六七十年代广泛的应用需求的出现促成了各类数据库系统的产生。八九十年代大型网络分布计算环境的普及使得政府、企业的应用需求趋同，导致几大数据库系统的“大一统”局面出现。当下互联网特别是云计算的出现，使得应用需求再趋多样化，人们更期盼与自己的需求功能相宜的数据库系统，而不是面面俱到的“大拼盘”系统，多样化时代重新到来。最近日渐火爆的“NoSQL”运动正是迈向这一目标的尝试。我们在本年度报告里试图把这些我们观察到的、看明白或没看明白的一些问题总结成短文，与大家交流，抛砖引玉。

本年度，实验室开了一个好头。新获国家自然科学基金项目 1 项，IBM 和 MSRA 资助项目各 1 项，出版英文专著一部，国际会议论文集两部，组织期刊 JCST 专辑“Trends Changing Data Management”，发表相关方向高水平科研论文 20 余篇（包括 IEEE TKDE、Pervasive and Mobile Computing 等），获专利授权 2 个，新申请专利 6 个。在系统开发方面，我们进一步深化原有的的中文文献集成系统 C-DBLP，试图构筑一个“ScholarSpace”，它将由三部分组成：SearchScholar+EasyScholar+SocialScholar。这里将充分利用我们 Web 数据集成的技术，并融入社会网络的思想。希望这一成果能为学界同行提供更好的服务。

作为大会主席在人大成功举办第 27 届中国数据库学术会议（NDBC），特别是我们邀请到二十多位七八十岁的老专家共聚“中国数据库历史回顾和萨师煊教授追思会”，是特别感到欣慰的一件事情。感觉能为同行提供有价值的交流服务是一种享受，感谢实验室老师同学为此付出的巨大辛劳。

这里谨以此集感谢来自学校方方面面的支持，感谢国家自然基金委和 863 计划的资助，感谢所有关心和支持过我们的人们。

孟小峰

2010 年 12 月 24 日于北京

目　录

Web 数据管理（Web Data Management）

系统演示（Demo）

实验室年度亮点

成功举办第 27 届中国数据库学术会议

由中国计算机学会数据库专委会主办，中国人民大学、北京大学、清华大学共同承办的第 27 届中国数据库学术会议(NDBC2010)于 2010 年 10 月 14 日上午在中国人民大学逸夫会议中心召开。这是中国数据库界的一次盛会，有来自海外及全国各地的代表 400 余人参加本次大会。信息学院王珊教授为本次大会指导专家，孟小峰教授为本次大会主席。

本次会议内容精彩纷呈，得到与会者的充分肯定。本届会议主要关注数据库技术所面临的新的挑战问题和研究方向，着力反映我国数据库技术研究的最新进展。

会议期间，数据库专委会特别邀请见证了我国数据库研究历史发展的老专家到会，召开了“中国数据库发展历史回顾暨萨师煊教授追思会”，共有二十多位来自全国各地 70 岁以上的老专家参加，专家们满怀深情地回顾了三十多年我国数据库事业发展历程，深切缅怀了萨师煊教授对数据库学科发展所做出的开创性贡献。

实验室同学积极参加本次会议的志愿者服务工作，为会议的圆满召开付出了辛勤的劳动，志愿者给与会者留下深刻的印象，充分展示了 WAMDM 实验室成员的风采，得到大家的一致好评。

实验室开发完成学者主页自动生成系统：EasyScholar

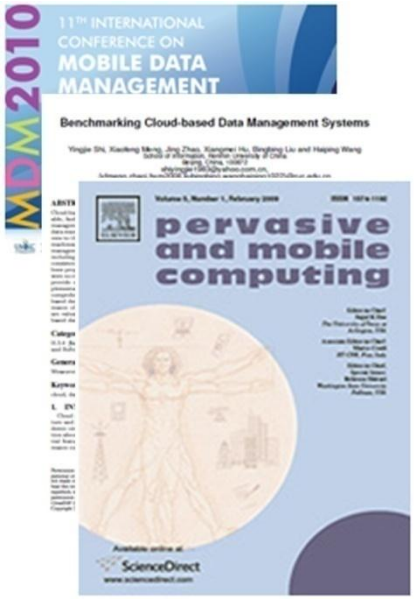
为了帮助国内学者能够快速、有效地建立和维护个人主页，展示个人信息和学术成果，WAMDM 实验室基于在 Web 数据集成研究方面的多年积累，并利用前期研发成果：面向计算机领域的中文文献集成系统 ScholarSpace，又成功开发了学者主页自动生成系统 EasyScholar。该系统以 Web 数据集成技术为核心，自动获取学者的部分学术信息，并展现在学者的个人主页中，为学者减少了大量的信息收集、整理工作。同时，该系统可以为每一个注册用户自动生成一个个人主页，用户可以进行修改、维护，并提供个人主页源代码生成、下载功能，用户可以自由发布由 EasyScholar 生成的主页。

我们期望 EasyScholar 能够成为广大中国学者展示学术信息、进行学术交流的平台，为国内学术的繁荣发展做出我们自己的贡献。

实验室发表多篇高水平学术论文

2010 年，WAMDM 实验室共发表了二十四篇高水平的学术论文，分别发表在 Journal of Pervasive and Mobile Computing (PMCJ)、计算机学报、MDM 2010、SIGIR2010、DASFAA2010、WAIM2010 等重要期刊和会议，论文涉及普适数据管理、云数据管理、闪存数据库系统、Web 数据管理等领域。

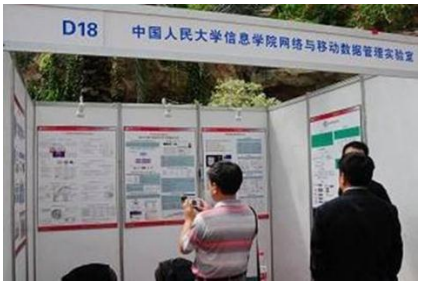
其中卢泽萍、周大等合作的论文“HV-recovery：一种闪存数据库的高效恢复方法“ 荣获 NDBC2010 “萨师煊研究生优秀论文”，胡享梅、赵婧等合作的“TaijiDB：一个双核云数据库管理系统“ 荣获 NDBC2010 最佳系统演示。



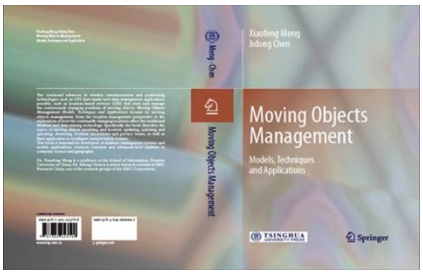
参加中国计算机大会科学技术成果展

2010 年 10 月 11-12 日，孟小峰教授应邀参加在杭州举行的 2010 中国计算机大会，并在云计算专题论坛上做了题为ñ面向云计算的数据管理ò的主题报告。

WAMDM 实验室科研成果应中国计算机学会的邀请代表中国人民大学参加了ñ高校科研成果展ò，集中展示ñ纯 XML 数据库系统 OrientX（王选奖成果）ò、ñ云数据库系统 TaijiDBò、ñWeb 数据集成系统 ScholarSpace（国家自然基金特优结题成果）ò、ñ闪存数据系统 FlashDB（国家自然基金重点项目成果）ò、ñ移动环境位置隐私保护系统（863 计划信息领域重点项目成果）ò等科研成果。中国计算机学会副理事长郑纬民教授、ACM President Alain Chesnais 先生、本届计算机大会主席、阿里巴巴集团首席架构师王坚博士等专家及来自各科研院所、高校及 IT 企业的参会人员参观了人大 WAMDM 实验室展位，并与实验室参展人员就 Web 数据集成、云数据管理、闪存数据库、XML 数据管理等研究方向开展了深入的研讨。许多企业对有关的成果产生了极大的兴趣。



出版英文学术专著



由孟小峰教授和陈继东博士撰写的英文学术专著《Moving Objects management: Models, Techniques and Applications》由计算机领域权威出版机构斯普林格（Springer）和清华大学出版社联合在国内外正式出版发行。

该书是孟小峰教授及其团队历时十年的研究成果，全书比较系统地介绍移动对象管理相关内容，包括移动对象管理模型（包括移动对象建模、移动对象更新、移动对象索引等内容），移动对象管理技术（包括移动对象查询、移动对象预测、移动数据不确定性研究等内容），和移动对象管理应用（包括动态交通导航、动态交通网络、移动对象聚类分析、位置隐私保护等内容）等。丹麦奥尔堡大学教授 Christian S. Jensen 为本书专门作序，给本书以高度的评价。

举办第二届云数据管理国际研讨会

The Second International Workshop on Cloud Data Management（CloudDB 2010）

在承办第一届云数据管理国际研讨会的基础上，WAMDM 实验室于 2010 年 10 月 30 日在加拿大多伦多又成功承办了第二届云数据管理国际研讨会，孟小峰教授担任本次研讨会的联合主席。本次研讨会依附于第 19 届信息与知识管理国际会议（CIKM2010），主要包括面向大规模数据存储和管理的云架构、云数据隐私和安全、云数据管理系统中的查询处理和索引、海量数据分析算法等主题，吸引了国内外多所著名大学和研究机构的学者参与进来共同探讨云计算技术及其发展。

实验室世博之旅

2010 年 7 月 30 日至 8 月 2 日，WAMDM 实验室师生集体前往上海参观 2010 年上海世博园。参观世博园期间，大家饱览了世界各国的特色展馆，对异域风情有了切身的感受，也深刻体会到了“城市,让生活更美好”的世博主题。经过三天的参观、游览，大家既放松了心情，增强了友谊，同时又满怀信心准备迎接下学期紧张、繁忙的学习生活。

数据管理前沿技术报告

云数据管理研究进展与展望

赵婧 胡享梅

1 引言

随着互联网的日益普及和 IT 技术的迅猛发展，互联网数据急剧膨胀，如何存储和管理海量数据已成为一个亟待解决的挑战性问题。云计算的概念应运而生，它改变了数据存储的基础架构。伴随着云计算的产生，云计算的研究备受关注，云计算的概念对于每一个 IT 界的人都不陌生。然而，关于云计算的发展前景，所有的专家学者意见并不统一。

一方面，伴随着的云计算应用的出现，改变了数据的存储方式，催生了一批新企业的诞生，这极大地促进了整个科学的进步与发展。云计算的研究话题也引起了极大地关注，在刚刚已召开的国际顶级学术会议 SIGMOD2010、VLDB2010 以及 CIKM2010 CloudDB workshop 上都有大量的关于云计算相关的研究成果。

另一方面，倘若我们仔细分析研读，会发现这些云数据管理的相关研究可以归结为以下两类：(1)将原有的研究问题使用 Map-Reduce 框架进行实现。(2)在云数据管理这个新的应用背景下，使用传统的索引、查询等技术来解决云数据管理的问题。

综上所述，云计算确实是一个划时代的变革，对科技地发展起着极大地推动作用，然而，云数据管理的研究并没有什么新的实质内容产生，无异于新瓶装旧酒。这是否意味着云计算的研究毫无意义呢？当然不是！我们知道美国国家自然基金投入大量资金进行云计算的研究，而且随着 Google、Yahoo!、Facebook 等企业的推动，出现了不少基于云计算平台的数据管理系统，而且大部分系统已经投入生产环境使用。因此，云计算是真实的，但目前的云数据管理研究现状却表明云数据管理的研究并不真实。导致这一现象的原因归结起来主要就是云计算的应用并没有明确显现，这也就导致了云计算上的需求不够明确，进而导致创新性研究的缺乏。研究的创新性依赖于需求的变革。以互联网为例，由于 TCP/IP 体系结构的发展，互联网在七十年代迅速发展起来，最初的应用仅局限于 e-mail(电子邮件)、FTP(文件下载)和 telnet(远程登录)等在大学和特殊领域推广的互联网应用，互联网技术研究也在起步阶段，随着应用的不断明晰，各种技术研究不断成熟，进而又产生各种应用需求，各大互联网公司逐渐发展起来，循环地推动了研究与技术的进步。因此，只有新的应用模式逐渐明晰，才有可能带动相关技术的发展，云计算的内涵才能够真正明晰，创新的研究模式才能够带动起来，进而达到云计算真正的成功。

本文接下来将主要从三个方面进行详细介绍云数据管理的相关研究。主要包括：(1)云数据管理的研究现状；(2)云数据管理的主要研究问题；(3)总结和展望。

2 研究现状

云计算的核心思想，是将海量的通过网络互连起来的计算资源通过统一的管理和调度，形成一个抽象的计算资源向用户按需提供服务，对于用户来讲，云计算的使用就好比用电一般，需要多少用多少，随时取用。这是一个美好的愿景，但是从系统管理的角度讲仍然存在许多亟待解决的问题：系统结构、数据模型、可扩展性、一致性、容错性等等。其中最为突出的两个问题在于海量和容错。海量包含多个角度：集群规模、数据量以及用户量。无论从那个角度来讲，都使管理系统面临着严重的挑战。另一方面，云计算平台往往采用廉价、不可靠的 PC 机来搭建 shared-nothing 集群，因此出错几率高于传统的分布式数据库中的高性能服务器。而这个问题随着集群规模的增大显得尤为突出[7]：查询设计的云数据规模越大，需要工作的节点就越多，而在查询中节点出问题的概率也会越大。文献[8]指出，Google 公司平均每个分析任务都会遇到1.2个节点错误。如果每次出错查询操作都需要重新启动的话，那么一个分析任务很有可能永远无法完成。

伴随着云计算概念的兴起，尤其是在传统的关系数据库面临各种压力与挑战的情况下，NoSQL 数据库应运而生。顾名思义，NoSQL 数据库即打破了传统的关系数据库的范式约束。由于传统的关系数据库在应付高并发读写、海量数据的高效存储和访问以及对数据库的高可扩展性和高可用性的需求显得力不从心，暴露出许多难以克服的问题，进而引发了 NoSQL 运动，其拥护者认为，关系数据库的许多主要特性面对当前的挑战非但无用武之地，反倒掣肘系统的功能及性能。比如对于数据库事务一致性需求、写实时性和读实时性的需求以及复杂的 SQL 查询，特别是多表关联查询等等。因此，各种 NoSQL 数据库放弃了关系数据库强大的 SQL 查询语言和事务一致性及范式的约束，或者采用 Key-Value 数据格式的存储以满足极高的并发读写性能，或者采用面向文档的方式以保证系统满足海量数据存储的同时具备良好的查询性能，又或者针对可扩展性展开的可伸缩数据库以增强其弹性的扩展能力。近年来，随着 NoSQL 运动的蓬勃发展，人们从初期的打破传统的关系数据库约束逐渐演变成对当今数据存储及管理可行且高效灵活的方案的探求，这与云数据管理的目的是极为相似的。在云数据管理中，我们同样要解决的是传统的关系数据库在数据及查询压力下所暴露出的实时插入性能、海量存储能力、迅速的查询检索速度以及无缝扩展等问题。NoSQL 数据库与云数据管理两者殊途同归，从满足应用需求的角度来说，最终都渴求找到一种集一致性、可用性和高容错性于一身的数据存储及管理方案以应对日益高涨的数据管理需求。

但是，根据 Eric Brewer 教授所提出著名的 CAP 理论，一致性（C: Consistency），可用性

（A: Availability），分区容错性（P: Tolerance of network Partition）三者不可兼得，必须要有所取舍。关注一致性，就需要处理由于系统不可用而导致的写操作失败的情况，而如果关注的是系统可用性，就需要做好读出的数据并不一定是最新值的准备。传统数据库保证了强一致性（ACID）和高可用性（侧重 CA），所以分布式数据库的集群实现非常困难，其扩展性受到了一定程度的限制。而近年来不断发展壮大的 NoSQL 数据库，尤其是 Key-Value 数据库就是通过牺牲强一致性，采用 BASE 模型，用最终一致性的思想来设计系统，使得系统达到高可扩展性和高可用性（侧重 AP）。但是，对于 CAP 理论也有一些不同的声音，数据库大师 Michael Stonebraker 在[9]中提出，为了 P 而牺牲 C 是不可取的。事实上，数据库系统最大的优势就对一致性的保证，如果我们放弃了一致性，也许 NoSQL 比数据库更有优势。

与云计算在工业界引起的热潮相似，近年来在学术界也涌现了大量云数据管理方向的炙手可热的话题。各种研讨会相继召开，汇集了大批对云计算充满兴趣的研究者、开发者、用户以及前沿实践人员。包括 The ACM Symposium on Cloud Computing 2010 (ACM SOCC 2010)，VLDB’2010 的座谈会 Cloud Databases: What's New?，CIKM’2010 的第二届 CloudDB Workshop 等。我们在此简要介绍一下第二届 CloudDB Workshop 的内容。第二届 ACM 云数据管理国际研讨会（CloudDB’10）于今年 10 月 30 日在加拿大的多伦多作为 CIKM’10 的相关会议召开。此次研讨会的主要目标是讨论处理基于云计算框架下的大规模数据管理的挑战问题。研讨会共收录 8 篇学术论文，研究方向包括查询效率的提高、数据挖掘计算、MapReduce 和 DBMS 的自适应算法、云安全、云框架、数据备份应用、系统测评以及二分图数据管理等。可以看出，在云数据管理领域内现有的研究工作在某种意义上来说还不成熟，还有很大的上升空间。本次研讨会中的工作重点大多集中在将现有的网格和 MapReduce 技术采用到云环境下进行开发。与会者一致认为云计算中依然存在着许多开发性的挑战，比如云数据的安全性和查询处理的高效性等。总体而言，本次 workshop 上次工作主要还是集中在如何将现有的技术如何应用在云计算环境中，研究工作仍然不够成熟，留有很大的上升空间。同时与会者一致认为，在云数据管理领域中还有许多开放性问题有待研究，如云安全、高效查询等等。

3 主要研究问题

经过我们实验室接近两年的研究与探索，我们总结出云数据管理的研究问题主要包括以下几个方面：（1）云数据存储问题的相关技术研究：xml 数据存储、key-value 存储以及 Object 存储问题。（2）查询处理问题的相关技术研究。（3）索引管理的相关技术研究。（4）事务处理的相关技术研究。（5）云数据管理的隐私保护问题的相关技术研究。下面将就这些问题进行详细的阐述。

3.1 云数据存储

随着云计算的不断发展，互联网数据的不断膨胀，逐渐呈现出数据的多样化，云数据存储面临的问题将会越来越明显，经过我们的详细调研与研究，我们总结出未来云数据存储主要面临的问题分为以下三大类： XML 存储、Key-value 存储、Object 存储。

XML 作为 Internet 数据表示和交换的工具，已经成为了众多应用领域中的标准数据格式。XML 文档的存储方式极大地影响了查询处理的效率，成为一个非常重要的研究方向。云数据管理系统的出现为 XML 文档的存储提供了一个极好的存储平台，那么究竟如何在云数据管理系统中进行 XML 文档的存储必将是一个亟待解决的研究课题。

针对绝大部分的检索都是基于主键的查询应用，使用 Key-value 存储将会是一个很好的选择。它被广泛应用于缓存、搜索引擎等领域。同时，一个好的 key-value 存储系统需要满足一些条件：可用性、可扩展性、故障恢复以及高性能。简单来说，就是数据不能丢失，服务不能中断，能对故障进行感知并能自动恢复，读写性能极高。目前已有一些开源的云数据管理系统默认就是以 key-value 格式存储数据，譬如 Cassandra，然而这些系统针对实际的应用仍然有一些问题。针对实际应用，结合开源系统进行存储的研究开发具有极大的意义。

对象存储提供了具有高性能、高可靠性、跨平台以及安全的数据共享的存储体系结构。其组成包括智能存储接口和设备，以及分布的元数据管理。在对象存储系统中，客户端可以直接访问存储设备，减少了数据存储路径中的控制路径。在对象存储中，使用对象存储设备(Object-based Storage Device，OSD)进行物理的数据存储。OSD 是连接到网络上的存储设备。它可以是磁盘、磁带或者其他的存储介质，并具有自我管理功能。伴随着云数据管理的兴起与发展，在云数据管理系统上进行对象存储也是一个极具意义的研究课题。

3.2 查询处理

从对查询接口的支持来看，目前的云数据管理系统大多只支持简单的关键字查询，不能支持数据分析工作当中经常使用的聚集、连接等查询操作。目前的云数据管理系统大多基于采用键-值模型存储数据的分布式文件系统构建。当查询被提交时，系统利用键值对数据进行搜索，这样限制了查询接口的种类，也不利于查询的处理和优化。推动云数据管理系统的动力是对于大规模海量数据的高级管理，而且其面向的用户也具有多元化的特征。未来的云数据管理系统在处理海量数据时，必须能够提供高性能查询处理。如何针对云计算平台上数据的海量性、分布性以及冗余的特性设计查询优化策略及算法显得尤为重要。

查询优化是分布式数据库系统，尤其是规模巨大的云数据库系统中的核心问题。查询优化的最终目标就是要找到最小的通信代价和最低的算法复杂度。分布式查询处理就是将一个分布式数据库上的高级查询转换成局部数据库上的一个有效的低级执行计划。转换主要包括两个方面：第一，产生输入查询的正确表示，以便执行计划能产生预期的结果；第二，对执行计划进行优化，即必须对代价函数最小化。对以上两个方面孤立成两个顺序的步骤：数据的局部化和全局化。数据局部化是将一个分布式数据库上的代数查询转换成一个等价的段查询，并通过代数转换来作进一步的简化。全局查询优化通过决策操作的顺序，结点间的数据

移动，以及数据库操作的分布和局部算法的选择来为输入的分段查询计划产生一个优化，其关键在于如何选择操作的执行顺序。选择查询操作的重点又在于连接操作的顺序，因为此类操作往往涉及多个节点上，操作尤为复杂。

3.3 索引管理

在云数据管理平台上，用户查询面临的是海量的数据，查询效率必然是一个亟待解决的问题。于是，受传统关系数据库的启发，我们可对云存储管理系统的数据建立索引，以提高用户查询的效率。那么，究竟选择何种索引就是一个关键问题。目前，针对云数据管理的索引已有一些研究工作

针对单维度的数据，我们可以建立 B+树索引，hash 索引等，文献[1][2][3]都是有关一维数据索引的研究，它们均在实际存储数据的物理节点上建立本地索引，然后在服务器端建立全局的索引，当用户查询数据时，通过服务器端的全局索引定位到本地索引，然后进行数据的查询，这样可以大大减少查询的时间。文献[1][2][3]也分别通过实验验证了论文中提出的算法的有效性。针对多维数据，我们可以建立 R 树，k-d 树等多维索引。文献[4][5]都是有关多维索引的研究。文献[4]采用 k-d 树建立本地索引，采用 R 树建立全局索引，它主要的应用于主-从体系结构中。文献[5]采用 R 树建立本地索引，采用 CAN 建立全局索引，它主要应用于点对点体系结构中。最后，二者都通过实验验证了文章中提出的索引算法的有效性。基于查询效率在海量数据处理中的关键地位，索引的研究仍将是一个值得研究的关键性问题。

3.4 multi-key 事务处理

随着 Web 应用数据规模的发展，可扩展的云数据管理系统在云计算研究中占有越来越重要的位置，而 key-value 存储方式是这种存储系统的一种主要架构，包括 BigTable, PNUTS, Dynamo 等等。在这些系统中，数据被存储成 key-value 对，但是对于大多数系统来讲原子性操作的粒度只能达到单个键值。这种特性在当前的很多 Web 应用中工作良好，但是在下一代 Web 应用中却力不从心：在线游戏，社会网络，合作编辑等等。因为这些应用需要在一组键值上进行读写操作，所以一致性的处理及事务原子性的保证要以一组键值为单位而不仅仅是单个键值。目前有一些系统可以支持 multi-key access, 例如 Google 的 AppEngine Store，可以支持一组键值的事务处理，但是这种键值组是固定的，无法动态修改，无法满足当前的 Web 应用。一种可行的解决方案是在 key-value 存储层上增加一层键值组管理层，负责键值组的建立与撤销，同时管理键值组中所有数据的读写一致性[6]，在这种方案中，如何保证键值组管理的容错性和一致性将是研究的重点所在。

3.5 云数据管理的隐私保护

云计算已成为未来海量数据管理的必然趋势，从成本和性能两方面考虑，越来越多的企业更愿意把自己的数据中心从昂贵的高性能计算机转移到公有云或私有云中。然而，中小企业以及个人用户在使用云计算带来的便利的同时，也经受着隐私泄露的风险。业界越来越多的人认为，云计算中的隐私问题已经危及到云计算未来的发展。

云计算模型中一般有三个角色：数据提供者、查询用户以及云计算平台---云数据库系统。

数据提供方将数据提供给云数据库系统存储，查询用户通过云数据库系统的平台对数据提供方的数据进行查询。对于云计算的各个参与者而言，面向隐私保护的查询处理都是迫切需要解决的问题：针对查询用户来说，如果在查询处理中隐私保护机制不完善，用户担心自己的隐私泄露，将会尽量减少使用云计算服务；针对数据提供者来说，如果用户查询将暴露其它用户的隐私，那么用户会对其服务可靠性产生质疑；对于云数据库系统而言，如果用户隐私遭到泄露，用户可能放弃该公司的云计算服务。因此，迫切的需要一种在云计算中的隐私保护查询处理技术，既能保护用户的查询隐私、数据隐私以及三方交互隐私。

4 总结和展望

综上所述，各种现象都表明，云计算是真实存在的，只是现阶段在云计算上的研究并不真实，大有新瓶装旧酒的趋势。放眼当前形势，中国应当架构符合我国实际情况的云计算平台，探索新的应用模式，从而展开云计算相关研究。我们认为，这种新的应用模式应该是大数据和云计算相结合的产物。云计算扮演的角色是应用的基础，它作为应用的载体存在，而大数据才是真正产生价值的所在。如今的应用需求实际是对服务的需求，我们如何为用户提供某种服务？能够为用户提供什么样的服务？这都需要数据来进行支撑，没有数据服务就无从谈起。因此，积累数据财富应该成为我们的习惯，在不远的将来，数据的积累也许会成为国力的标志。从这个角度讲，对“数据思维”的研究可能会带来新的意义。在传统的研究探索中，人们把精力集中于复杂算法的研究，试图设计出复杂的算法来解决问题，但是，当有了数据，也许配合了大规模的数据后，原有的问题只需要非常简单的算法即可得到很好的解决。如今风风火火的 NoSQL 运动给无论是学术界还是产业界都带来了新的机遇，人们对数据库的需求发生了改变，人们不再需要一个大而全但自己只能用到一小部分的数据存储及管理工具，这就带来了打破大一统数据管理现状的机会，我们需要做的是抓住机遇，研究出既满足实际需求，又恰如其分的高效灵活的云数据管理系统。

参考文献

[1] M．K．Aguilera, W .Golab , and M.A.Shah ： A Practical Scalable Distributed B-Tree . (VLDB’08)

[2] S.Wu and K.-L. Wu ： An Indexing Framework for Efficient Retrieval on the Cloud.

[3] S. Wu, D. Jiang, B. C. Ooi , K. L. Wu: CG-Index: A Scalable B+-tree Based Indexing Scheme for Cloud Data Management Systems(PVLDB’10)

[4] X. Zhang, Z. Wang, J. Ai, J. Lu, X. Meng: An Efficient Multi-Dimensional Index for Cloud Data Management. In Proceedings of the CIKM Workshop on Cloud Data Management (CloudDB2009), November 2, 2009, Hong Kong, China.

[5] J．Wang,S.Wu,H. Gao,J. Li, B. C. Ooi :Indexing Multi-dimensional Data in a Cloud System(sigmod’10)

[6] Sudipto Das Divyakant Agrawal Amr El Abbadi. G-Store: A Scalable Data Store for Transactional Multi key Access in the Cloud. SoCC’10

[7] J. Hamilton. Cooperative expendable micro-slice servers (cems): Low cost, low power servers for internet-scale services. In Proc. of CIDR, 2009.

[8] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI, 2004.

[9] Michael Stonebraker， Errors in Database Systems, Eventual Consistency, and the CAP Theorem，
http://cacm.acm.org/blogs/blog-cacm/83396-errors-in-database-systems-eventual-consistency-and-the-cap-theorem/fulltext

闪存数据库技术研究进展报告

——The Forth Workshop on Flash-based Database Systems

1 引言

闪存产品的性价比在不断地提高，使得闪存的应用越来越广泛，尤其是作为大数据集的替代磁盘的存储设备的应用越来越受关注。对于数据库研究者来说，建立高效的基于闪存的数据库管理系统成为数据库研究者亟待解决的问题。

以孟小峰教授为负责人的课题组（获国家自然科学基金重点项目 “闪存数据库技术研究”的资助，项目号：60833005）于 2010 年 7 月 28 号在中国北京中国人民大学召开了第四届专题研讨会。会议主要讨论了基于闪存存储器的数据库的存储管理、缓冲区管理、事务处理、查询处理、闪存存储板设计、闪存存储器在集群平台上的应用等关键问题。与会人员有来自于中国人民大学的孟小峰教授、杨楠副教授、单智勇讲师、中国科技大学的岳丽华教授、金培权副教授、以及香港浸会大学三所高校相关的硕士博士研究生，同时还有幸邀请到了百度刘斌等高级工程师和北京大学崔斌教授和他的学生们。会议包含以下几个报告：操作系统和数据库缓冲区管理算法、数据库外排序算法、闪存存储板和闪存芯片测试、数据库事务处理和 TPCC 测试结果。这些报告展示了最新的研究进展和技术成果，为基于闪存存储器的数据库的进一步研究与应用奠定基础，为基于闪存存储器的数据库理论和技术的进一步发展提供新思路。

2 闪存数据库技术研究

课题组负责人、中国人民大学孟小峰教授做了题为“闪存数据库技术研究”的报告，介绍了基于闪存存储器的数据库研究进展和新存储介质 SCM。闪存数据库系统研究主要以 NAND 型闪存为基础展开研究，目前已经在 EDBT、MDM、ACM SIGSPATIAL GIS、CIKM、WIAM、WISA、SAC、NDBC 和一些期刊上发表了数十篇文章，主要内容包括缓冲区中数据调度的问题研究、查询处理中的外排问题研究、闪存存储板的设计和实现、基于闪存的索引的改进和实现、基于闪存的数据库恢复方法的探索、闪存模拟器的设计与实现以及基于闪存、闪存模拟器和 SSD 的一些评测结果。通过对实验室结果、国际研究进展和现有系统的分析，找到适合闪存的数据库技术并实施在闪存数据库系统中，提高闪存数据库系统的性能。

SCM 存储器是一类存储器，性能介于内存和磁盘之间，全电设备无机械延迟，非易失性存储介质，具有很快的访问速度，并且成本很低。PCM 就是一种 SCM 存储器，相比于闪存和内存具有很强大的优势，所以目前越来越多的人关注这种新的存储介质，对于 PCM 的关注包括，把 PCM 放在整个体系结构中什么位置——使用 PCM 作为内存、或者扩展内存、或者页设备、或者取代磁盘，因为 PCM 的成本越来越低并且 PCM 性能比较好，所以这几种情况都有可能存在。

3 缓冲区管理

硬盘调度算法对数据库应用系统的整体性能具有关键性的影响。本次研讨会关于缓冲区管理策略主要有以下四个报告组成。

传统的硬盘调度算法是基于磁盘设计的，也就是电梯调度算法。这种算法的主要目的是尽量减少磁盘的磁头移动，因为磁头移动是机械运动，耗时很长。但是，面对 SSD 固态硬盘时，显得难以发挥 SSD 的内部优势，就好比是雇用一位电梯司机去指挥一个复杂的管弦乐团。**在这种情况下，中国人民大学的单智勇做了题为“A SSD I/O Scheduler for Database Systems”的报告，该报告提出了一种新硬盘调度算法 GP-Deadline。**设计 SSD 调度算法的七条原则：并发，无饥饿，写聚合，读聚合，读写分离，读优先和对齐块边界。在这七条原则的基础上，提出一种新的 SSD 调度算法。该算法引入 deadline 机制，以适应数据库应用系统的专门需求。该算法充分利用 SSD 的特性来发挥调度策略的作用。在后面的研究中，我们将进一步验证这七个原则的有效性。然后，在 Linux 内核实现这个新的调度算法，并且在数据库应用的负载下进行测试。

验证基于闪存的缓冲区置换算法性能的一种有效的测试手段，是在实际的 DBMS 中实现该算法，并使用测试工具（例如各种 benchmark）进行测试。PostgreSQL 是当前世界上最先进，功能最强大的自由数据库管理系统，具有良好的扩展性，适合在其中实现缓冲区置换算法，测试算法性能。为完成这项工作，需要对 PostgreSQL 缓冲区管理部分进行研究。**在这种情况下，中国科学技术大学的陈恺萌做了题为“Extending a Flash-aware Buffer Replacement Algorithm on PostgreSQL”的报告，该报告介绍了在 PostgreSQL 中替换缓冲区置换算法的工作方案。**在了解 PostgreSQL 的缓冲区管理后，以 CCF-LRU 算法为例介绍如何替换缓冲区置换算法。要在 PostgreSQL 中实现 CCFLRU 算法，主要是解决两个问题：现有的数据结构无法支持双 LRU 队列;在数据库工作过程中需要对两条 LRU 链表进行维护。在算法实现后，本文利用 TPC-C 测试工具 Sysbench 以及利用 ODBC 自行编写测试程序这两种测试手段，对改造前后的 PostgreSQL 进行了对比测试。实验结果表明，使用该改造方案实现的 CCFLRU 算法能够在 PostgreSQL 中正常工作，但想要证实两种算法实际运行性能的优劣，还需要进一步的测试工作。

现代计算机系统常常借助缓冲区来提升整体性能，缓冲区管理方法也成为计算机领域的热点研究问题之一。随着闪存得到越来越广泛的应用，闪存新特性对传统的缓冲区管理方法提出了新的挑战。传统针对磁盘的缓冲区管理办法专注于提升缓冲区的命中率，没有考虑到闪存独特的读写不均衡的特性，致使在装备了闪存的系统上不能发挥出缓冲区的最大性能。**为了更好地在闪存系统上发挥出缓冲区的性能，北京大学的吕雁飞做了题为“Operation Aware Buffer Management in Flash based Systems”的报告，该报告讨论了可以感知操作的缓冲区管理方法。**ACAR 的替换策略，即自适应代价感知缓冲区替换策略。针对不同的操作命中，ACAR 调整方式不同，这样就考虑了操作的差异的信息，更好地实现缓冲区管理。ACAR 的实验结果，表明 ACAR 比现有的方法更能有效地在闪存上的系统上管理缓冲区。

为了进一步利用操作的特性，报告还提出了两种估价操作频繁度的指标，分别是操作的近度(recency)和间隔距离。两种估价的指标各有侧重。报告结合这两种指标提出权重公式来决定替换出缓冲区的页面。但是不足之处是这种方法时间复杂度过高，报告最后提出如何进一步提高这种方法的速度是将来可能的方向之一。

在已有的基于闪存的缓冲区管理算法中，一般假设闪存的读操作的代价是远远小于写操作的代价，因此，已有的算法一般采用减少写操作的方法来提高系统的性能，但是对于不同的 SSD 来说，其读写性能的不对称性有着很大的差异，如果仅考虑减少写操作的次数，将仅对读写代价差异大的 SSD 适用。**在这种情况下，中国人民大学的汤显作了“ACR: an Adaptive Cost-Aware Buffer Replacement Algorithm for Flash Storage Devices”的报告，该报告提出了一种适用于各种闪存的基于代价的缓冲区管理策略—ACR。**提出了一种新的自适应缓冲区置换策略—ACR，此策略使用的是基于代价进行置换的策略，因此能够适用于不同的 SSD；再者，此策略将权值与整个队列相关联，使得内存操作的代价将为 O(1)；最后，此策略能够比较好的适应长序列和循环访问模式。为了验证 ACR 的性能，我们在仿真平台上进行了实验，对比了 LRU、CF-LRU 以及 CFDC 的性能。最后的实验结果显示，ACR 算法显著地提高了缓冲区置换算法的性能。

4 查询处理

外排序是 DBMS 中最基本的算法之一。数据库中很多操作都是以其为基础，因此，外排序对数据库的整体性能有很大影响。传统的外排序算法是基于磁盘而设计的，没有结合闪存的读写不均衡特性，直接移植到闪存上时，性能未能得到优化。

目前，闪存上有关外排序算法的优化研究比较少。有鉴于此，香港浸会大学的高岫做了题为“SSDSort: A New flash-aware external sorting algorithm”的报告，该报告提出了一种新的适于闪存特点的外排序算法：SSDSort。传统的外排序算法是根据归并排序而设计的。所有需要排序的数据，需要经过排序与合并的过程。在闪存环境下，写的代价相对较高。如果对于所有的页面都进行排序与合并，将会使性能大大下降。尤其对于将近排好序或者含有数据分布较集中页面的数据，写出此类页面对性能有很大影响，没能发挥闪存快速读取的特性。针对以上问题及闪存的物理特性，本文提出了一种新的外排序算法:SSDSort。为了验证 SSDSort 的性能，论文在仿真平台中进行了实验，对比了 SSDSort 与传统归并排序所需要写出的数据量。最后的实验结果显示，SSDSort 能显著的减少中间结果的写出量，提高数据库的排序性能。

5 闪存存储

闪存存储板是闪存数据库的物理基础，它的性能对闪存数据库系统的整体性能有着很大的影响。目前闪存相关的算法研究发展很快，包括 DBMS 层面的算法和 SSD 内部的控制算法。因为商用 SSD 的内部算法被厂家固化，研究者难以在商用 SSD 上进行相关研究，所以开发一个可定置的闪存存储板很有必要。先前开发的闪存存储板在测试中的表现和理论计算

误差很大，所以必须找出第一块闪存存储板的问题并作出改变。**在这种情况下，中国科学技术大学的杨濮源同学做了题为“An Improved Flash Storage Board”的报告，该报告指出了第一块闪存存储板设计中的问题并提出了改进方案。**第一块闪存存储板的一个页的读写速度均比理论值低，而且该板的写速度比读速度快，这是和闪存的 IO 特性相违背的。导致这种现象的原因是和 PC 平台 PCI 接口相关的。在这种情况下，采用 DMA 的接口工作方式成为很好的选择。在比较了各种 DMA 模式的接口芯片的优劣后，改进的闪存存储板采用了 PEX8311 作为接口芯片。同时，在改进板的设计中，地址扩展方式由原来的位扩展改成了字扩展，目的是为了在改进板上测试多通道设计的效果，以便为以后进一步的设计积累资料。目前改进板已经制作完成，正处在调试阶段，调试工作完成后，可以很快获得测试数据。

作为一种新型的存储设备，闪存具有与传统磁盘完全不同的物理特性和更加复杂的工作原理，因此为了充分利用闪存自身的特性构建更加高效的系统，对这些特性的了解和把握是必不可少的。但实际上各个闪存芯片的生产厂家出于各种商业原因，对其所生产的芯片特性要么守口如瓶要么只是给出一些保守模糊的数据，这无疑给基于闪存之上的系统设计加上了桎梏，因此针对闪存芯片自身特性的测试就变得至关重要。**在这种情况下，中国人民大学的梁智超做了题为“Hush…tell you something novel about flash memory !”的报告，该报告结合加州大学圣地亚哥分校非易失性系统实验室所作的工作介绍了一些针对闪存芯片的最新测试结果。**在这项工作中，他们对来自五个不同生产厂家的闪存芯片进行了测试，这些闪存芯片的容量不尽相同，制作工艺包括从 50nm 到 70nm 的 SLC 型和 MLC 型。测试目的主要是从性能、耗电量以及可靠性三个方面对闪存的已知特性进行验证，对未知的特性进行探索和揭示。通过对测试结果的分析，一些有趣的规律得到了呈现。针对这些测试结果得来的特性，该实验室的研究人员提出了一种新的 FTL 算法和针对闪存的数据编码方式，实验结果表明两者在性能、耗电量和闪存使用寿命上等方面能够起到积极的作用，这也从一个侧面说明了针对闪存特性的测试工作的重要性。

6 事务处理

事务处理作为 DBMS 中不可或缺的重要组成部分，它的性能对于数据库的整体性能有着重大的影响，而由于 SSD 与磁盘不同的读写特性，使得基于闪存的数据库系统的事务管理部分变得更加复杂，因此，如何快速有效的维护闪存数据库系统 ACID 特性就变得越来越重要。**在这种情况下，中国人民大学的卢泽萍、范玉雷做了题为“Session on Transaction of Flash-DB”的报告，该报告介绍了本项目组最近的关于事务方面的几项工作。**

PTLog 采用对页表记日志的方法，来有效的减少由于 WAL（先写日志）规则所带来的大量的闪存空间浪费，并利用闪存快速的随机读特性，通过改变日志结构，提供行之有效的恢复策略。HV-recovery 对闪存中天然存在的数据的历史版本使用新的日志结构加以管理和利用，提供高效的恢复。通过周期性设立检查点，减小无效日志记录的长度，节约闪存空间。引入混合式存储系统，将日志记录单独存放在磁盘上，以便对闪存数据库的恢复性能进一步提高。同时也保证了算法具有在数据库正常运行时有较小的开支、算法有比较强的可靠性、

系统失败后恢复速度快和日志文件的空间需求较小等优势。通过针对 TPCC 的分析及和开源数据库 Oracle Berkeley DB 的对比实验看出，HV-recovery 比传统数据库的恢复时的写操作数可以减少接近一半，其恢复时间与传统数据库相比，能缩短到原来的大约 1/8，与在 SSD 上的传统数据库相比，也可以缩短 40%，充分显示了本算法的优越性。

闪存昂贵、异位更新和有限擦除次数，但是闪存有很好的读写性能，所以目前采用固态硬盘和磁盘进行混合存储是比较好的解决方案。但是目前 OLTP 系统要求越来越高的并发度，在混合系统之上提高事务的并发成为一个关键问题。由于网络 DBMS 和用户双方对某些信息的一致性要求并不是很高，可以引入弱一致性来提高 DBMS 的并发。现有的 DBMS 事务子系统采用的并发控制协议基于串行化理论，主要分为两大类：单版本的和多版本的。单版本的并发控制协议主要有两阶段加锁协议、乐观协议、时间戳协议和有效性确认等等。多版本的并发控制协议主要有多版本加锁协议、多版本乐观协议、多版本时间戳协议和多版本有效性确认等等。这些并发控制协议使得事务执行都具有较高的一致性，这种一致性限制了事务并发度的提高。由于现在对于一致性的要求不是那么严格，数据存在天然的多版本特性，所以采用引入弱一致性的多版本的并发控制协议会增加事务的并发度。固态硬盘内部封装了多块闪存芯片，对外通过软件模拟成为块设备，但是这些闪存芯片之间存在着可以并发操作的现象，可以利用闪存芯片之间的并发机制来提高事物的并发度。利用混合式系统存储数据——读多的数据放在固态硬盘上，写多的数据放在磁盘上——大部分读操作发送到固态硬盘上执行，大部分写操作发送到磁盘上执行。对于固态硬盘的读操作可以不进行一致性控制，让更多的读操作尽量并发执行，使用闪存芯片之间的并发控制。由于不进行严格一致性控制，使得数据存在多版本的特性，所以提出了“基于弱一致性、芯片级并发、多版本并发控制协议和混合式存储的数据管理系统”。

PG 的 TPCC 测试平台：借助开源代码程序 BenchmarkSQL 对 PostgreSQL 数据库进行 TPCC 测试，BenchmarkSQL 源代码采用 Java 语言编写，这就使得该测试程序可以很容易移植到其它操作系统之上并能很好的运行。BenchmarkSQL 采用 DOS 命令的方式测试 PostgreSQL 的数据库，操作简洁方便。

针对磁盘单一存储系统、固态硬盘单一存储系统和混合式存储系统，采用 BenchmarkSQL 测试程序进行 PostgreSQL 的 TPCC 测试，测试结果显示固态硬盘具有良好的性能，TPCC 测试标准下的 Tpmc 值在固态硬盘上比在磁盘上提高约 10 倍，大大显示了固态硬盘的优势。

7 总结

闪存数据库中数据存储管理、索引、缓冲区管理、查询处理和事务处理等模块已经取得了很大的研究进展，对于项目以后的研究会有很大的推动作用。缓冲区管理对于提高整个闪存数据库系统的性能起着至关重要的作用，研讨会对于缓冲区管理机制进行深刻的研究，从不同角度设计基于闪存缓冲区管理策略，并把 其实现到开源数据库中来检测基于闪存的缓冲区管理策略的性能。但是根据目前的研究，闪存数据库中各个模块仍然有很大的改动空间，

尤其是闪存数据库的存储管理和事务处理应该具有较大的改动空间使其更好的适应闪存的特性，使得闪存数据库的整体性能大幅度提高，这主要体现为查询速度提高、事务处理速度快和事务并发度提高等等。

同时，研讨会还就将来硬件的发展对数据库系统带来的变化进行了专门的讨论，闪存相对于磁盘仍然还具有很大的优势，但是新硬件的产生给闪存提出了巨大的挑战。

研讨会还就闪存的发展现状、发展趋势，以及对基于闪存的数据库系统研究中存在的新的挑战进行了热烈的讨论，闪存的寿命问题受到越来越多的关注，比如百度就在致力解决闪存在服务器上的使用寿命，以提高闪存在现实应用中的性价比，这对项目以后的研究会有很大的推动作用，但是仍有许多未知的领域需要我们去探讨和研究，这就需要我们投入更多的研究工作和热情。

总之，目前的闪存数据库并没有充分发挥闪存的特性，需要根据闪存的物理特性进行重新设计，以发挥闪存的优越物理性能。

移动 Web 搜索关键技术研究

张金增

1 引言

随着移动通信和 Internet 在人们日常生活中的日益普及，移动通信带宽的大幅度提高和移动终端功能的逐渐增强，传统的服务已经不能满足用户多元化的需求，人们希望随时随地利用移动终端访问互联网上的服务，从中获取丰富的信息。移动互联网实现了 Web 和移动通信的逐步融合，使其成为产业界备受关注的领域。

随着 3G 时代的到来，越来越多的用户使用移动终端能够便捷地访问网络，根据中国互联网络信息中心发布的最新报告显示，手机网民数已达 1.137 亿，并呈直线上升趋势，并且使用手机上网的用户会越来越多。而信息搜索是用户在访问网络时最经常进行的活动之一。在日常生活中，人们经常会碰到很多“now and here”的问题，需要查询与其正在进行的活动相关的信息：

- 我的朋友现在位于什么位置？
- 电影院现在正在放映什么电影，离自己位置最近的影院有无剩余的票？
- 附近有没有比较好的吃晚餐的地方？现在有一些什么优惠活动？
- 我需要停车，现在离我最近的有停车位的停车场在哪里？

由以上问题可以看出，人们使用移动设备搜索时大多数需求都与位置密切相关，但使用传统的搜索引擎仅仅利用纯粹的文本关键字搜索，用户往往不能获得理想的查询结果。此外，与传统互联网搜索环境相比，移动终端受到了屏幕尺寸小、网络带宽有限等限制。这些不同点为新环境下移动 Web 搜索带来了许多新的挑战。

因此，本研究在移动环境下，根据移动用户的需求，将地理数据与 Web 数据进行无缝的集成；在此基础上进行高效的面向移动用户的查询处理，获得高度精确的满足用户需求的结果，从而为用户提供“Near by Now”的服务，具有非常重要研究价值。为 3G 时代下的移动 Web 搜索提供了一种新思路，具有十分广阔的应用前景。

2 移动 **Web** 搜索概述

与互联网搜索和移动数据库查询技术相比，移动 Web 搜索具有独特的特点。本部分结合相关研究工作，对移动 Web 及其特性进行了分析和总结。在此基础上，指出了移动 Web 搜索与互联网搜索存在的差异。

2.1 移动 **Web** 基本概念及其特点

移动互联网是指移动用户从自身实际需求出发，能够通过无线终端随时随地的通过无线方式接入互联网。传统的移动互联网是一个封闭的网络，其封闭性体现在网络、终端和应用三个方面。封闭的特性制约了移动互联网的发展。新型移动互联网具备如下特点：

开放性：开放性体现在网络开放、应用接口开放、内容和服务开放等多个方面，用户拥

有选择的权利。

分享和协作性：在开放的网络环境中，用户可以通过多种方式与他人共享各类资源，可以实现活动参与，协同工作。

创新性：结合 Web2.0 与移动网特征，移动互联网能够为用户提供无穷无尽的创新性业务。

开放、分享和创新构成了移动互联网的核心特征。随着移动互联网的深入发展，移动互联网现有的垄断性、封闭性终将被打破，开放性将成为移动互联网服务的基本标准，用户将具有更大的自主性和更多的选择，用户角色由被动的信息接受者转变成为主动的内容创造者，移动终端的智能性将进一步增强，用户之间的通信和内容体验将更具有交互性。

2.2 移动 Web 搜索与互联网搜索的异同点

移动 Web 搜索是指以移动网络为数据传输承载，将分布在传统互联网和移动互联网上的数据信息进行搜集整理，供手机用户查询的业务。移动搜索作为搜索技术与移动通信技术的一种结合体，融合了两种技术的各自特点。移动搜索的出现，真正打破了地域、网络的局限性，满足了用户随时随地的搜索服务请求。

庞大的手机用户群成为移动搜索的潜在用户，该类用户区别于互联网用户的特征以及移动网的特点，对搜索技术的功能实现提高了更高的要求。移动搜索与互联网搜索存在本质区别，主要表现在搜索方式、搜索要求、搜索渠道和搜索内容等多个方面。详见表 1

	移动 Web 搜索	互联网搜索
终端特点	屏幕较小、功能单一、普及率高、体积小、携带方便、承载网络覆盖面面大	大屏幕、功能丰富、普及率较低、体积较大、携带不便、承载网覆盖面小
搜索方式	关键字搜索、自然语句搜索	目录检索、关键字搜索
搜索需求	准确性、便捷性、个性化	准确性、海量性、快速性
搜索渠道	短信、搜索门户、搜索栏、IVR	搜索门户、搜索栏、浏览器地址栏
搜索内容	Wap 网站内容、传统互联网内容、运营商及服务提供商内容、传统信息提供商及黄页内容	以互联网网站内容为主，信息量十分丰富
搜索目的	搜索需要的内容、定制需要的服务	搜索需要的内容和站点
搜索限制	无	存在网络接入限制
搜索费用	流量费、服务定制费、部分搜索服务需要单独付费	免费

表 **1**　移动 **Web** 搜索与互联网搜索的差异

3　移动 Web 搜索基本框架及关键性技术

作为一种新型搜索技术,移动 Web 搜索的研究仍处于起步阶段. 这种新兴的搜索是传统搜索技术在移动平台上的延伸，真正打破了地域、网络和硬件的局限性，满足了用户随时随地的搜索需求。为了满足这些需求，需要提出一系列对移动数据进行表示、模型构建、索引和信息检索的新技术。这一节首先提出移动 Web 搜索框架，然后从地理标记 Web 资源、混合索引的构建、面向移动用户的查询处理、查询结果的排序与可视化几个方面，对移动搜索的关键性技术进行分析。

3.1 移动 Web 搜索基本框架

移动环境其位置动态变化，屏幕狭小、计算资源有限等特点对传统的文本搜索提出了高精准的查询要求，给移动Web搜索带来了许多新的挑战。在移动Web搜索领域存在着许多研究问题: 地理标记Web资源、集成更新、混合索引的构建、面向移动用户的查询处理、查询结果的排序与可视化等. 有些问题已经得到了一定程度的研究,而有些问题还处在研究的初步阶段。为了给出一个全面的认识，我们提出了移动Web搜索的整体框架，如图2所示。该框架被划分为四个模块：

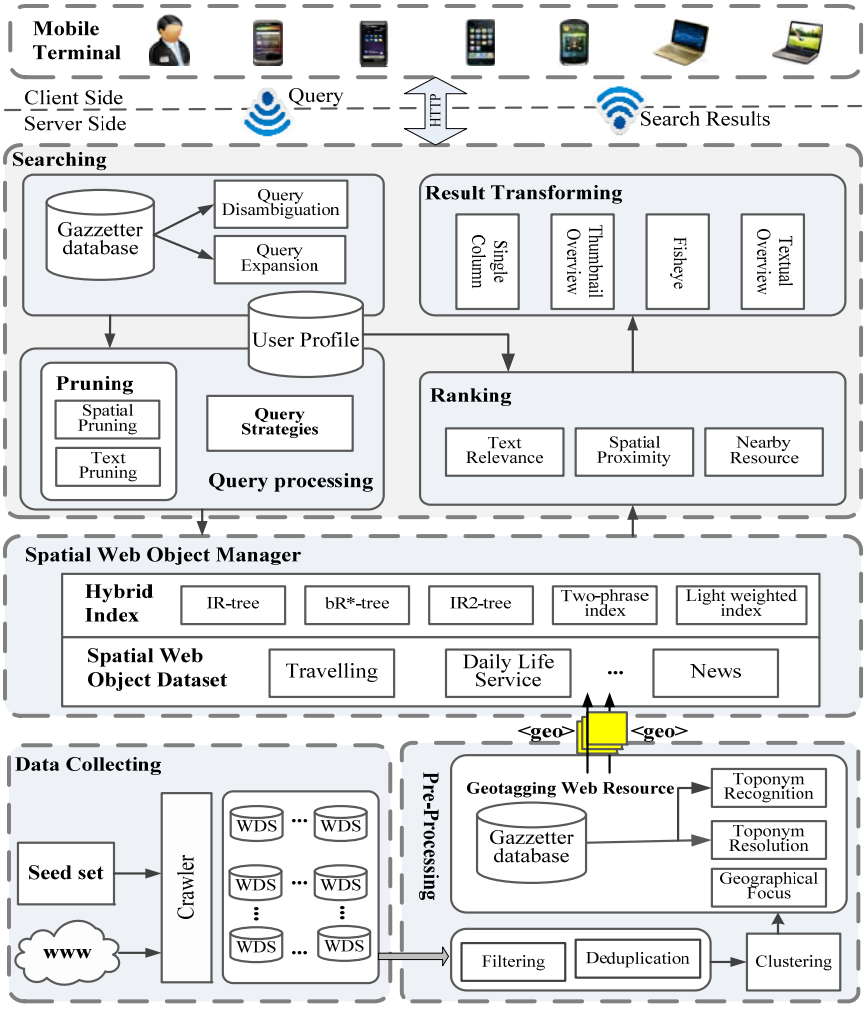


图1 移动Web搜索系统框架

- 数据搜集模块(Data Collecting)：通过种子节点集合，爬虫从 WWW 发现和下载 Web 页面，接收 URL，下载页面，分析内容，并沿着出链接重复执行一系列操作，从而获得初始的粗糙数据集。
- 预处理模块(Pre-collecting)：数据搜集阶段完成后，需要对 Web 数据源进行清洗和去重操作，并对过滤后的数据按内容所在的领域进行聚类。为了支持移动搜索，需要标记出 Web 资源所对应的地理位置或者覆盖的地理范围，完成这项任务需要地名识别、地名分辨和覆盖地理范围的确定三个步骤，在此基础上，就可以获得空间 Web 对象数据集。
- 索引模块(Indexing)：对空间 Web 对象构建索引需要综合考虑地理空间和文本两个方面，为了提高搜索的效率和访问的准确率，需要对其构建混合索引，将空间索引和文本索引进行无缝的集成。混合索引的构建用来检索出与用户需求高度相关的信息。
- 搜索模块（Searching)：用户通过移动终端设备提交查询，需要对提交的查询消除歧义并扩展；然后进行查询处理，对返回的结果按照文本相关性、距离相近性及周围环境进行综合排序；由于移动设备固有的一些固有局限，需要对用户查看的结果页面进行一定的转换处理；最终将搜索结果返回给移动终端。

3.2 移动 Web 搜索关键性技术及存在的挑战

移动环境其位置动态变化，计算资源有限等特点给移动 Web 搜索带来了许多新的挑战，本文的研究内容包括地理标记 Web 资源、为空间数据和文本信息所组成的空间 Web 对象构建混合索引、基于位置的查询处理和查询结果的处理等关键性问题。

(1) 地理标记 **Web** 资源

许多 Web 资源像商业、新闻等 Web 页面都包含大量与位置相关的信息，再加上地理位置对移动搜索结果的精确性具有决定性的作用。因此，如何准确有效的找出 Web 资源对应的地理位置是一个关键性的问题。

对于给定的 Web 资源，准确的标记出所对应的地理位置或覆盖的地理范围大致需要三个步骤：地名识别（toponym recognition）、地名分辨（toponym resolution）和覆盖地理范围的确定 （Geographical focus）。

(a) 地名的识别：需要处理 geo/non-geo 歧义的问题，目前最普遍采用的是自然语言处理方法。但这种方法仅仅适用于分析静态的文档库，对于动态和不断更新的文档库却并不适合。为了解决该问题，可以采取混合的地名识别技术，使用词性标记（POS）和命名实体识别（NER）对位置短语进行标记，搜集这些地名采用基于规则的启发式方法和基于统计的处理相结合的方法。

(b) 地名分辨：主要任务是解决 geo/geo 歧义。最常用的方法就是为识别出的地名分配一个衡量其流行度的缺省值，并结合启发式规则来完成地名的分辨。但这种方法对于那些不是很出名的地方却无能为力。对于这个问题，一个初步的想法是可以采用分层地理本体的概念去解决，在分层地理本体中建立“分辨上下文”。

(c) 覆盖地理范围的确定（Geographical focus）：就是找出文档覆盖的地理区域。使用

层次本体去决定覆盖地理范围，在该层次结构中每一个分辨出来的地名都为它的父节点贡献分值，然后选择分值最高的本体节点作为该文档的地理聚焦点。另外一种普遍使用的策略就是选择出现频度最高的地名作为地理聚焦点。

(2) 混合索引技术

移动搜索需要检索与地理上下文相关的文档，这种需求要求索引建立以文本和位置为基础。R-tree、四分树、网格等是空间索引方法，而倒排索引、位图索引、签名文件是文本信息检索中有效的索引方法。这两类索引在针对不同的应用在各自的领域都得到了很好的发展。解决移动Web搜索的最简单的索引方法是先使用倒文本索引找出匹配关键字的结果，然后采用空间索引搜索进行空间搜索。但这种两阶段的索引方法对于提前决定需要获得top-k个结果，在第一阶段所需要的查询结果个数是非常困难的，并且CPU代价和I/O代价也非常高，所以这种方法不适合移动Web搜索。因此需要设计出一种综合考虑文本和空间位置的索引结构，使其有效地整合空间索引和文本索引以保证达到最优的搜索效果。

一种就是将用于文本检索的倒排文件和用于空间搜索的 R-tree 结合起来，使用倒排文件对 R-tree 进行扩充。新建立的 R-tree 的每个节点包含的信息是以该节点为根的子树的所有对象的位置信息和文本内容的概括。叶子节点指向对象的文本文档的索引文件，而非叶子节点的文档是将该节点所有孩子节点的所有文档的并集。在使用该索引结构进行搜索时，如何针对不同的应用进行设计，使其所占用的空间较小又能拥有高效的搜索效率，还需要进行深入的研究。

(3) 面向移动用户的查询处理

查询处理算法利用构建的混合索引方法去评估空间相近性和文本相关性。对于移动用户提交的查询，返回的结果与移动用户当前的位置密切相关，提交相同的查询，其时间、位置不同，得到的结果会有很大的差异，查询的结果是需要按照空间的相近性和文本的相关性进行排序。移动环境下最常见的几种空间查询 KNN 查询、RkNN 查询、Range 查询、Closest-Pair 查询等都需要采取不用的查询策略进行处理（比如深度优先、最佳优先等）。使用这些方法对空间 Web 对象进行搜索时，针对不同的场景，需要找出合适的处理策略，并找出约束条件，使得空间剪枝和文本剪枝在查询处理的过程中同时进行，从而有效的加速搜索的进程，对整个搜索具有极其重要的作用。

(4) 查询结果的处理

移动设备由于自身的特点只能为用户提供较小的显示区域，无法浏览大量的信息，如果用户被淹没于大量查询结果中，会导致用户的满意度下降。因此需要对查询结果进行优化处理，把用户最满意的查询结果以最简洁的方式按照某种顺序进行展示。

(a) 查询结果的排序：对于从空间Web对象数据库返回的大量结果，需要将与用户查询最相关的记录排列在靠前的位置，以提高用户的满意程度。对查询结果的排序需要综合考虑多个因素对排序的影响。不仅要考虑文本的相关性，还要考虑位置的相近性和周围环境对查询结果的影响，。

(b) 查询结果记录摘要的生成：通常一个查询结果包括许多数据项，但移动设备因其自身特点只能提供较小的显示区域，如果将查询结果的全部信息都显示，会大大降低用户的浏览效率。在实际中并不需要查询一个结果的所有信息，因此需要根据用户的查询、当前的位置等信息来选择合适的数据项，进一步将所过滤出的数据项进行文本摘要处理。

4 结论

随着移动网络的日益普及,用户的规模呈指数级增长，移动 Web 搜索越来越成为学术界和工业界共同关注的热门话题。本文提出移动 Web 搜索框架，然后从地理标记 Web 资源、混合索引的构建、面向移动用户的查询处理、查询结果的排序与可视化几个方面，对移动搜索的关键性技术进行分析，指出仍然存在的挑战和可能的解决办法。目前，虽然在移动环境下的 Web 搜索的某些方面提出了一些解决方法，但总体上缺乏系统性。总的来说对移动 Web 搜索研究仍然处于刚刚起步的阶段,仍有大量关键问题需要进行深入细致的研究。因此，移动 Web 搜索具有非常重要的研究价值和广阔的应用前景。

万维网信息可信性问题

孟小峰 艾 静 马如霞
中国人民大学

关键词：万维网数据管理 万维网信息 可信性 可信度传播机制

随着万维网（Web）应用的迅速发展，目前一个门户网站的信息量已经远远超过网站管理者所能处理的能力范围。同时，也改变了人们获取信息和进行消费的方式。人们更愿意通过网络获取各种资讯，寻找数码新品、电影和歌曲，网上购物、交易，甚至在网络上找工作、找配偶。万维网已成为人们获取信息的重要途径。

然而，互联网是一把双刃剑，它在给人们带来极大方便的同时，也带来了一系列的问题。从这些纷繁复杂的信息中甄别可信信息已经变得日益重要起来。在互联网发展的过程中，由于信息可信性（Credibility）问题所引发的麻烦和损失不胜枚举。例如：2003年3月29日，国内一家网站的编辑错误地把微软公司总裁比尔·盖茨遇刺身亡的虚假消息当成CNN（Cable News Network，美国有线电视新闻网）的新闻（如图1所示），此后该消息被新浪、搜狐等一些影响力巨大的门户网站在第一时间以醒目标题转发。在半个小时之后，微软公司出面澄清，这一骇人听闻的重大新闻只是个谣言。最新发布的2009中国未成年人互联网运用状况调查报告显示，被调查的中学生大多对网上信息的真实可信度持怀疑态度，认为网上大部分或绝大部分信息真实可信的只占26.6%。互联网是否可信已经是一个困扰人们很久的问题，在一定程度上也阻碍了互联网的发展。因此，针对信息可信性问题的研究是网络进一步健康发展的需要。

图1 微软公司总裁比尔·盖茨遇刺身亡的虚假新闻

万维网信息不可信的成因

造成万维网信息不可信的原因多种多样，主要包括5类：（1）信息具有时效性。信息的效用依赖于时间并且有一定的期限，其价值与提供信息的时间密切相关。例如，某人的单位信息和联系方式随着他的工作变迁在不断地发生变化，因此当我们寻找其联系信息时就会遇到一些过时的信息，这些过时信息需要我们进行甄别；（2）信息发布者由于自身专业知识不足而发布了一些错误的信息；（3）信息发布者为了自身的利益故意制造虚假信息。例如，电子商务网站附带的用户点评论坛，其中包含大量的用户评论信息以及使用感受等，对该网站的用户决定是否要购买某种商品有非常重要的影响作用（用户总是更信任同为消费者的其他用户）。因此某些别有用心的人（如某件商品的出售者）就有可能故意假扮用户来引导评价或打分；

（4）信息发布者发布一些具有导向性的信息。例如，关于“伊拉克战争是否正确”，不同的网站可能因为立场或政治观点的不同，而选择只发布有利于自己观点的信息。这些信息描述的事件确实是在伊拉克战争中真实发生过的，所以都不是虚假信息。然而，如果只判断网站上的信息描述的是否是现实世界中真实发生的事件，而不考虑网站的情感倾向性，也会造成用户的误判；（5）万维网为信息的传播提供了良好的土壤。它的开放性和信息共享性等特征使得信息的传播更加迅速和高效。当然，这也为低可信度信息的传播提供了便利条件。同时，这些不可信信息一旦在网络中传播开来，需要很长的时间来清除。

基于上述原因，万维网上的不可信信息可分为以下几个类别：过时信息、错误信息、虚假信息、超前信息、片面信息和带有特定感情色彩的导向性信息等。根据这些信息的危害程度将信息分为：高风险信息、中度风险信息和低风险信息。其中，高风险信息主要包括：有意发布的虚假信息和病毒等；中度风险信息包括：无意造成的错误信息、过时信息、超前信息和片面信息等；低风险信息主要包括带有特定感情色彩的导向性信息等。

万维网信息的可信度

伴随着万维网技术发展，万维网信息可信性的研究面临各种各样新的挑战。万维网技术的发展经历了三代历程：Web1.0、Web2.0和Web3.0，如图2所示。

图2[1]形象地反映了三代万维网之间的异同。在Web1.0中，信息发布者与用户的角色都是固定的，因此只需要从少量的信息发布者着手控制信息的质量，就可以在很大程度上保证信息的可信度。这一时期，信息的可信度研究已经悄然兴起，并且成为一项重要的研究课题。最初的研究主要是从网站的可信性开始，研究如何设计网站从而使其更加可信。在Web2.0中，用户不再是只能被动地做信息的接收者，也可以作为信息的创造者向网上发布自己的信息。网上论坛、博客和合作知识库等网络应用应运而生。这时候

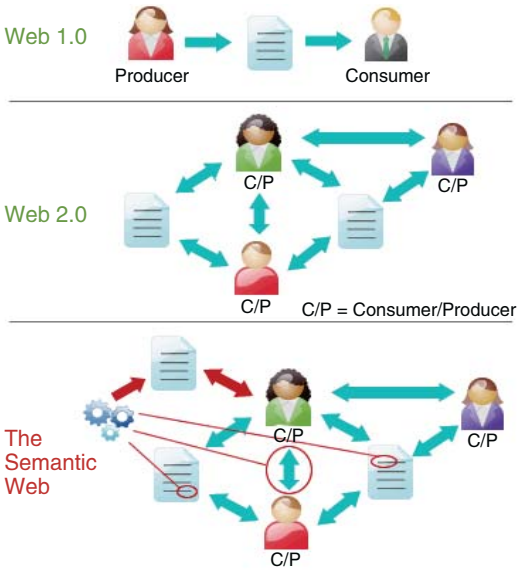


图2 Web发展的三代历程

万维网信息可信度的研究不但要考虑网站本身的特征对信息可信度的影响，还要考虑信息的生产者——普通用户的参与对万维网信息可信度的影响。用户的专业程度、历史信誉等因素都会对该用户所发布信息的可信度产生影响。Web2.0技术强调的是用户参与和交互。这一特点也使得用户之间在现实生活中的各种关系在网络中也逐步体现出来，如信任关系、合作关系等。研究这种人与人之间的相互信任关系，进而通过可信度的传播机制来计算信息的可信度也是可信度研究方面的一个重要议题。在Web3.0（语义网）中，参与者不仅仅是人，还有可能是计算机。语义网被看作是这一代网络的代表。语义网提供对数据语义的描述，使计算机能够“理解”万维网信息，实现计算机之间、计算机与人之间的智能交互，从而使万维网成为全球化信息共享的智能服务平台。计算机中的智能体（Agent）将成为网络的操作者和信息的传播者，所以需要研究智能体的可信度评估与计算等问题。与此同时，由于研究网络中的各个参与者之间的相互联系对信息可信度的评测具有重要的影响，因此不能孤立地进行研究。参与者之间的相互关系主要包括：网站之间的关系、网站与用户之间的关系、用户与智能体之间的关系等等。

下面从目前万维网信息产生过程中所涉及信息

本身、网站和用户三个方面分别论述信息可信度的研究现状。

面向信息本身的信息可信度

信息本身的可信度问题主要是从信息自身出发，根据其内容特征、相互之间的关联关系来研究信息的可信度。目前这方面的研究包括如下两个方面：

基于内容分析的可信度 本质是用信息本身的特征作为评价信息质量和可信度的标准。通常，基于内容分析的可信度计算方法主要包括两种：（1）基于拼写错误的评价方法。基本思想是使用文章中的拼写错误率作为评价这篇文章的数据质量和可信度的标准，认为网页中单词的拼写错误率与信息的质量是正相关的。这种评价方法的应用范围很广，因为它利用信息的文本特性来进行可信度及质量的评价，可以用于判断几乎任何网页的信息质量；（2）基于关键特性量化的评价方法。认为关键的信息特性对信息的质量、价值和可信度的评价有至关重要的作用，因此根据某一类具体的应用场景，通过对信息的几个基本特性的分析，将信息的可信度用量化的数值表示，进而计算出一篇文章（一个文档或一个网站）的信息可信度。从实际应用场景中抽象出数学模型是这一类方法的基本思路。

基于信息关联的可信度 信息之间并不是相互独立的，存在着各种关联，如超链接关联、引用关联、拷贝关联和语义关联等，这些关联在一定程度上影响着信息的可信度。目前这方面的研究大多基于超链接关联来分析页面的可信度，而对于其它关联关系则关注较少。基于超链接关联的信息可信度计算方法通常根据网页之间的超链接关系，如指向它的链接的重要度和可信度，来计算页面的可信度，这种思想类似于谷歌的网页排名算法PageRank。

尽管信息本身可信度的研究已经受到了一些关注，但仍存在如下问题：目前这方面的研究工作仍旧集中在文本信息的可信度上，对于图像、音频和视频信息的可信度研究相对比较少；在基于信息关联的可信度研究方面，主要集中在超链接关系的研究，对于信息间的隐含关联（如语义关联等）研究还很缺乏，解决这些隐含关联对于万维网信息可信度的影响是一个非常具有挑战性的工作。这些问题还有待进一步解决。

面向网站的信息可信度

网站可信性是人们最早关注的万维网信息可行性研究领域。其基本思想是网站的可信度与该网站中信息的可信度是正相关的，一般来说权威网站上面的信息比普通网站上的信息更为可信。网站的可信性可以从两个角度来判断：

独立网站的可信度 这类研究只考虑网站自身的特征，根据该网站中数据的质量、网站所有者的权威性、网站的领域特征等方面信息来判断网站的可信度。关键特征分析的可信度评估方法是最早用来研究网站特征对信息可信度影响的方法。例如，美国斯坦福大学的研究者曾在2001年发布了一个关于网站特点与其可信度关联关系的大规模调查报告。通过对来自美国和欧洲各国的1400多名志愿者进行问卷调查，评估51个不同的网站特点对其可信度的影响。这一工作是一个大规模的用户调查，为进一步研究万维网的信息可信性打下了基础。调查结果对如何设计更可信的网站有指导意义。

基于网站间依赖关系的可信度 主要考虑网站之间的依赖关系（如：相似依赖、相异依赖等）对网站可信度的影响，目前这方面的研究还非常少。

面向用户的信息可信度

根据Web2.0的理念，网站通常是由建设者负责建立信息发布平台与架构，主要的信息资源是由网站的用户参与发布。这种形式的万维网应用有很多种，如：BBS（Bulletin Board System，电子公告板）和合作知识库等。这些万维网应用都是合作的信息源，它包含了许多不同发布者的观点和看法，是集体智慧的集合。它的信息比仅由服务商提供信息的网站更加客观，并且更能反映大多数普通用户的感受和看法。以上这些都得益于用户的参与，然而用户的参与也给信息的可信性带来了很大的影响。用户之间的信息交互将不同的用户联系起来，

形成了一个社会网络。人们在这个网络中进行信息交流，这种信息的交互行为也给信息的可信性研究带来了巨大的挑战。因此，从用户的角度来研究信息的可信度具有现实的意义，目前这方面的研究主要有两种评估方法：基于评分和投票的可信度评估机制和基于信任传播的可信度评估机制。

基于评分和投票的可信度 基本思想是基于发布者的可信度、帖子的信息内容特征值等信息，进行归类、评价和打分，从而对以帖子为基本单位的信息的可信度进行判断。常用的归类与计算可信度值的方法是机器学习方法。目前主要的方法分为三大类：（1）根据帖子本身包含的信息与信息特征值，利用机器学习算法对帖子的信息可信度进行归类计算；（2）考虑用户可信度的影响因素，根据用户的历史记录计算其可信度，并辨别论坛中的信息的偏向性；（3）以作者为中心，建立可信度管理模型，建立作者与版本、作者与作者、作者与不同的词条（文章）之间的关联关系。

基于信任传播的可信度 在可信度传播机制中，不再是关注信息本身的可信度，而关注的是信息发布者的可信度，然后再根据信息发布者的可信度来判断信息的可信度。可信度传播机制主要建立在信任网络的基础之上。在信任网络中，节点根据其应用场景的不同代表了不同的实体，边反映了网络节点之间的信任关系。然后，以信任网络中少部分节点的可信度值作为先验知识，对目前可信的节点所信任的节点，均认为是可信节点。利用网络结构和邻近知识进行可信度值的传播。而且，信任值的传播操作，通常是通过对信任值矩阵（Trust Value Matrix）的变换操作实现的，包括矩阵加法、乘法等。这种网状结构中的信任机制，基本上都是利用名誉的传播机制，将节点的局部知识通过一个中央机制综合起来，从而得到统观全局的每个节点的可信度情况。信任网采用这种机制，使原本只知道他周围邻居的可信度的每个用户，通过整个系统采用合适的传播机制和信任值计算机制，可以预测出系统中任意两个用户之间应该对对方赋予的信任值。在P2P（Peer to Peer）网络、社交网络和语义网的环境中，信息可信度的评估通常采用传播机制来实现。绝大部分方法都是在信任网的基础上，根据自身的特点做一些改进，然后用传播机制（可能包括信任值和不信任值的传播）在节点之间传播可信度经验值。根据不同网络各自特有的特点，增加不同的机制。此外，图挖掘算法也是常用的方法之一。节点之间的网状机构可以被看作是一个图，而挖掘算法常用于对未知节点的可信值做推断。因为挖掘算法可以找出事物之间的联系，从而推断出未知的情况。

挑战与展望

虽然目前已经有许多针对万维网信息可信性的工作，但是缺乏系统性，还存在一些需要解决的问题。

图像、音频和视频信息的可信性 依据信息的载体特点可以将万维网上信息分为文字信息、声音信息和图像信息三大类。其中关于文本信息的可信性问题的相关研究工作最多。然而，近年来图像信息和音频信息在万维网上所占的比例越来越大，它们承载的信息量也非常巨大，受到研究者们越来越多的重视。目前，关于图像信息和音频信息的相关研究工作越来越多，但关于这些信息的可信性问题的研究工作却较少。因此，如何准确、有效地计算和评估图像和音频所携带的信息的可信性及可信度值，必将是未来重要的研究课题。

基于信息间隐含关联的信息可信性 目前基于信息间关联分析的可信度研究主要集中在网页之间的链接关联方面，而对信息之间隐含的其它一些关联的研究相对比较少。例如，信息之间的语义关联对可信度的影响。我们可以从语义的角度来区分信息内容的近似程度，从而利用这些相似度和信息的其它属性（如发布时间等）分析出信息之间的相互引用关系，进而根据这些引用关系来判断信息的可信度。

基于网站间依赖关联的信息可信性 网站间的依赖关系错综复杂，例如：超链接关系、拷贝关系和合作关系等。从网站的海量数据中发掘这些关系、分析这些关系之间的相互影响以及这些关系对信息可信度的影响程度和方式等都使得网站间依赖

关系的发掘具有非常大的挑战性。

构建多层次可信性关联 目前，在信息之间、网站之间、用户之间的相互关联对可信度的影响方面已经有了或多或少的研究。但是网站和信息、用户和网站、用户和信息之间也存在着很多关联，这些不同类型实体间的关联同样也影响着信息可信度。例如：网站中所有用户可信度影响了网站的可信度，进而影响着网站中信息的可信度，网站中信息的可信度反过来又可以影响网站的可信度，这些影响是相互渗透的。所以，通过构建多层次可信性关联的研究可以更加系统地研究信息的可信度。

用户发布信息的可信性验证 万维网中对已有信息的可信性评估只是一种亡羊补牢的做法，是对已经发布到万维网上的不可信信息进行甄别。然而，如图3所示，更好的办法应该是防患于未然，将不可信信息挡在万维网之外，因此我们需要在用户发布信息时就对用户和信息进行可信性验证。目前关于如何评估万维网上已有信息的可信性的研究工作有很多，然而用户发布的信息的可信性验证研究则关注较少。由于用户创造的消息种类和形式较多，如何验证这些用户的可信度及其发布信息的可信度，确保用户发布的信息是可信的，是一个非常具有挑战性的问题工作。

可信性评估算法基准测试 除了上述挑战性问题外，在万维网信息可信性研究体系中还缺少对可信性评估算法基准测试的研究。在关于万维网信息可信性的工作中，研究人员通常会提出一种或几种计算信息可信度值的算法，或者评估信息可信度值的评价机制，然后在数据集上通过实验证明这些算法和评价机制的准确性。然而，信息可信度算法和评价机制缺乏统一的数据源进行实验验证，而且没有基准测试（Benchmark）程序或规范，除了某些文章中有算法比较之外，很难知道不同的算法方法之间效果及效率的差异。因此，为万维网上的信息的可信度算法和评估机制提供统一的权威的实验数据集，以及基准测试是非常必要的。

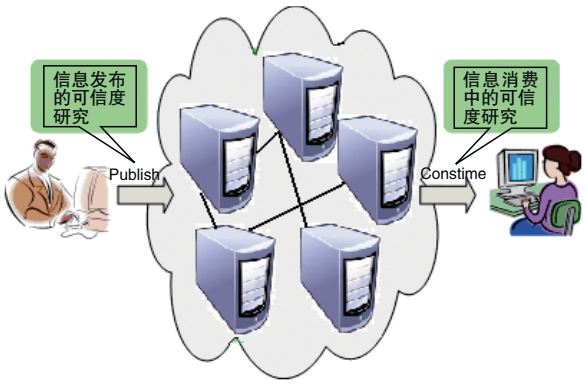


图3 万维网信息生产消费中的可信性问题

结语

万维网信息将在未来人们的生活、工作和研究中以及互联网的发展方面占有越来越重要的地位。本文分析了万维网信息可信性问题产生的原因、国内外信息可信性发展的现状，并在此基础上讨论了信息可信性问题所面临的挑战与未来的研究工作。万维网信息可信性研究对提高万维网中信息的质量和网络的进一步健康发展具有重要的意义。■

孟小峰

CCF理事。中国人民大学信息学院教授。主要研究方向为网络与移动数据管理，包括Web数据集成，XML数据库，云数据管理，闪存数据库和隐私保护等。xfmeng@public.bta.net.cn

艾 静

中国人民大学计算机应用技术专业硕士生。主要研究方向为Web数据管理。

马如霞

中国人民大学计算机软件与理论专业博士生。主要研究方向为社会网络和Web数据管理。

参考文献

[1] http://blogs.nesta.org.uk/innovation/2007/07/the-future-is-s.html

基于位置服务的隐私保护

孟小峰 潘 晓
中国人民大学

关键词：基于位置服务 位置隐私 查询隐私

隐私保护问题

应用分类

根据服务面向对象不同，基于位置的服务可以分为面向用户和面向设备两种[1]。两种服务的主要区别在于，面向用户的基于位置的服务，用户对服务拥有主控权；面向设备的基于位置的服务，用户或物品属于被动定位，对服务无主控权。根据服务的推送方式的不同，基于位置的服务应用可以分为Push服务和Pull服务。前者是被动接受，后者是主动请求。下面将用四个例子（如表1）说明上述分类。当你进入某城市时收到欢迎信息属于面向用户（你）的Push服务（欢迎信息被主动推送到你的移动设备上）；而你在该城市主动提出寻找最近邻餐馆属于面向用户（你）Pull服务；假如你是某物流公司老板，当你公司负责运输的货物，偏离预计轨道时将向你发出警报信息，这属于面向设备（货物）的Push服务（消息被推送到物流公司老板的移动设备上）；如果你主动请求察看货物运送卡车目前所在位置属于面向设备（货物）的Pull服务。

表1 LBS应用分类

	Push服务	Pull服务
面向用户服务	当你进入某城市时收到欢迎信息	请求查找最近邻餐馆
面向设备服务	在货物追踪应用中，当货物运送偏离预计轨道时给与警报信息	请求查找卡车现在所在位置

基于位置的服务与隐私

很多调查研究显示，消费者非常关注个人隐私保护。欧洲委员会通过的《隐私与电子通信法》中对于电子通信处理个人数据时的隐私保护问题给出了明确的法律规定[2]。在2002年制定的Directive文本中，对位置数据的使用进行了规范，其中条款9明确指出位置数据只有在匿名或用户同意的前提下才能被有效并必要的服务使用。这突显了位置隐私保护的重要性与必要性。此外，在运营商方面，全球最大的移动通信运营商沃达丰（Vodafone）制定了一套隐私管理业务条例，要求所有为沃达 丰客户提供服务的第三方必须遵守。这体现了运营商对隐私保护的重视。

隐私泄露

基于位置服务中的隐私内容涉及两个方面：位置隐私和查询隐私。位置隐私中的位置指用户过去或现在的位置；查询隐私指涉及敏感信息的查询内容，如查询距离我最近的艾滋病医院。任何一种隐私泄露，都有可能导致用户行为模式、兴趣爱好、健康状况和政治倾向等个人隐私信息的泄露。所以，位置隐私保护即防止用户与某一精确位置匹配；类似地，查询隐私保护要防止用户与某一敏感查询匹配。

位置服务 VS 隐私保护

回想一下，我们似乎正面临一个两难的抉择。一方面，定位技术的发展让我们可以随时随地获得基于位置的服务；而另一方面，位置服务又将泄露我们的隐私……当然，你可以放弃隐私，获得精确的位置，享受完美的服务；或者，你可以关掉定

位设备，为了保护隐私而放弃任何位置服务。是否存在折中的方法，即在保护隐私的前提下享受服务呢？可以，位置隐私保护研究所做的工作就是要在隐私保护与享受服务之间寻找一个平衡点，让鱼与熊掌兼得成为可能。

隐私保护方法

下面将介绍在基于位置服务中的三种基本的隐私保护方法。

假位置

第一种方法是通过制造假位置[3]达到以假乱真的效果。如在图1中，用户寻找最近的餐馆。白色方块是餐馆位置，红色点是用户的真实位置。当该用户提出查询时，为其生成两个假位置，即哑元（如图1中的黑色点）。真假位置一同发送给服务提供商。从攻击者的角度，同时看到三个位置，无法区分哪个是真实的哪个是虚假的。

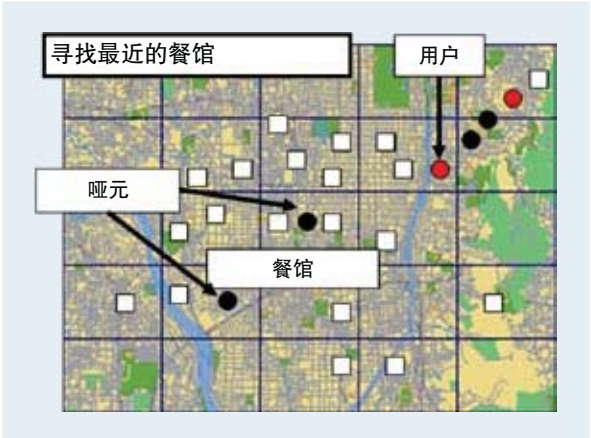


图1 假位置

时空匿名

第二种方法是时空匿名[4~6]，即将一个用户的位置通过在时间和空间轴上扩展，变成一个时空区域，达到匿名的效果。以空间匿名为例，延续图1寻找餐馆的例子，当用户提出查询时，用一个空间区域表示用户位置，如图2中的红色框。从服务提供商角度只能看到这个区域，无法确定用户是在整个区域内的哪个具体位置上。

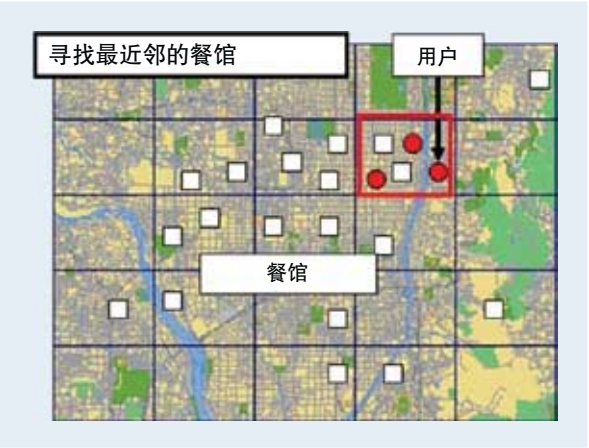


图2 时空匿名

空间加密

第三种方法是空间加密[7]，即通过对位置加密达到匿名的效果。继续前面的例子，首先将整个空间旋转一个角度（如图3），在旋转后的空间中建立希尔伯特（Hilbert）曲线。每一个被查询点P（即图3中的白色方块）对应的希尔伯特值如该点所在的方格数字所示。当某用户提出查询Q时，计算出加密空间中Q的希尔伯特值。在此例子中，该值等于2。寻找与2最近的希尔伯特值所对应的P，

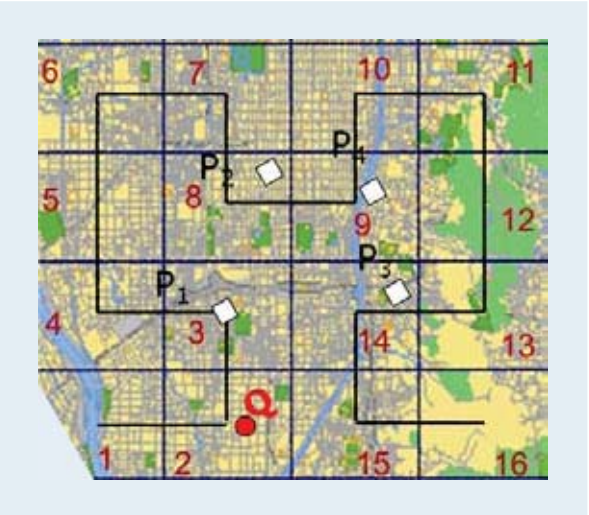


图3 加密

即P1。将P1返回给用户。由于服务提供商缺少密钥，在此例子中即旋转的角度和希尔伯特曲线的参数，故无法反算出每一个希尔伯特值的原值，从而达到了加密的效果。

感知隐私的查询处理

在基于位置的服务中，隐私保护的最终目的仍是为了查询处理，所以需要设计感知隐私保护的查询处理技术。

根据采用匿名技术的不同，查询处理方式也不同：如果采用的是假数据，则可采用移动对象数据库中的传统查询处理技术，因为发送给位置数据库服务器的是精确的位置点。如果采用时空匿名，由于查询处理数据变成了一个区域，所以需要设计新的查询处理算法。这里的查询处理结果是一个包含真实结果的超集。如果采用空间加密技术，查询处理算法与使用的加密协议有关。

隐私度与效率对比

从匿名效率和隐私度两方面对上述三种隐私保护方法进行对比[8]（如图4），可以看出加密是安全度最高的方法，但是加密解密效率较低；生成假数据的方法最简单、高效但隐私保护度较低，可根据用户长期的运动轨迹判断出哪些是假数据；从已有的工作来看，时空匿名在隐私度与效率之间取得了较好的平衡，也是普遍使用的匿名方法。下面将以时空匿名方法为主进行介绍。

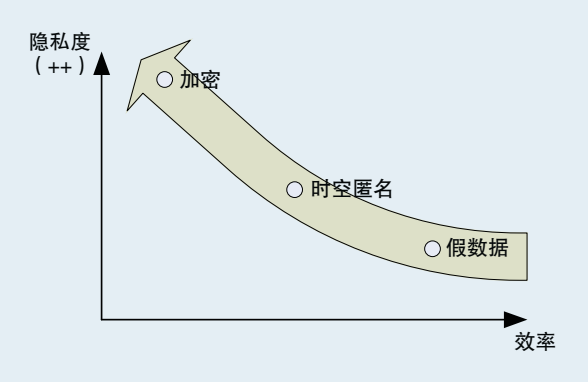


图4 隐私度与效率对比

存在的挑战

位置隐私研究中所面临的挑战包括四个方面：

1. 隐私保护与位置服务是一对矛盾；

2. 基于位置服务的请求，具有在线处理的特点，故位置匿名具有实时性要求；

3. 基于位置服务中的对象，位置频繁更新；

4. 不同用户的隐私要求大相径庭，所以隐私保护需要满足个性化的需求。

隐私保护系统结构

隐私保护系统基本实体包括移动用户和位置服务提供商，它具有如下有四种结构：独立结构[15]、中心服务器结构[4~6]、主从分布式结构[11]和移动点对点结构[9]。

独立式结构

独立结构是仅有客户端（或者移动用户）与位置服务器的客户端/服务器（Client/Server，C/S）结构。由移动用户自己完成匿名处理和查询处理的工作。该结构简单，易配置，但是增加了客户端负担，并且缺乏全局信息，隐私的隐秘性弱。

中心服务器结构

与独立结构相比，中心服务器结构在移动用户和服务提供商之间加入了第三方可信匿名服务器，由它完成匿名处理和查询处理工作。该结构具有全局信息，所以隐私保护效果较上一种好。但是由于所有信息都汇聚在匿名服务器，故可能成为系统处理瓶颈，且容易遭到攻击。

主从分布式结构

为了克服中心服务器的缺点，研究人员提出了类似主从分布式结构。移动用户通过一个固定的通信基础设施（如基站）进行通信。基站也是可信的第三方，与前者的区别在于它只负责可信用户的认证以及将所有认证用户的位置索引发给提出匿名需

求用户。位置匿名和查询处理由用户或者匿名组推举的头节点完成。该结构的缺点是网络通信代价高。

移动点对点结构

移动点对点结构与分布式结构工作流程类似，惟一不同的是它没有固定的负责用户认证的通信设施，而是利用多跳路由寻找满足隐私需求的匿名用户。所以它拥有与分布式结构相同的优缺点。

隐私保护研究内容

下面将介绍一些经典位置隐私和查询隐私保护方法以及感知隐私的查询处理技术。

隐私保护模型

先介绍一下迄今为止使用最广泛的位置k-匿名模型[10]，后面介绍的隐私保护方法均满足该模型。k-匿名是隐私保护中普遍采用的方法。位置k-匿名的基本思想是让一个用户的位置与其他（k-1）用户的位置无法区别。以位置3-匿名为例（如图5），将三个单点用户用同一个匿名区域表示，攻击者只知道在此区域中有3个用户，具体哪个用户在哪个位置，无法确定，达到了位置隐私保护目的。

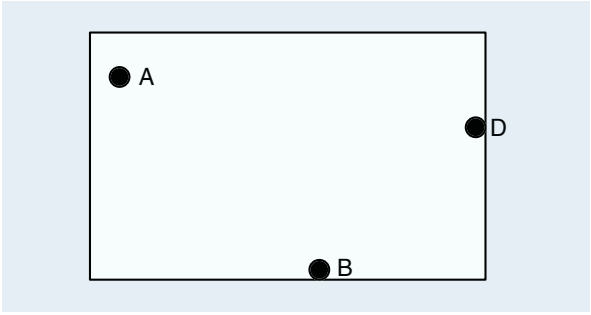


图5 位置3匿名

基于四分树的隐私保护方法

最早的匿名算法是基于四分树的隐私保护方法[10]。它解决了面对大量移动用户高效寻找满足位置k-匿名模型的匿名集的问题。其解决方法是：自顶向下地划分整个空间，如果提出查询的用户所在的区域的用户数大于 k，将整个空间等分为4份，重复这一步，直至用户所在的区域所包含的用户数小于 k，返回四分树的上一层区域，将其作为匿名区域返回。该方法的缺点是要求用户采用统一隐私度和返回匿名区域过大。

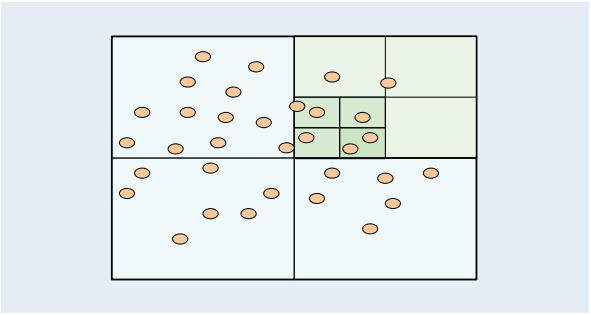


图6 基于四分树的匿名方法

个性化隐私需求匿名方法

在隐私保护中，不同的用户有不同的位置k-匿名需求，因此需要解决满足用户个性化隐私需求的位置k-匿名方法，这正是CliqueCloak[12]的贡献。其解决方法是利用图模型形式化定义此问题，并把寻找匿名集的问题转化为在图中寻找k-点团的问题。图7中，点是用户提出查询时的位置，k表示用户的最小隐私需求，圆圈代表用户可接受的最差服务质量。当新的对象m到达时，根据用户的隐私和质量要求，更新已有图，并找出m所在团。将覆盖该团所有点的最小边界矩形作为匿名区域返回。

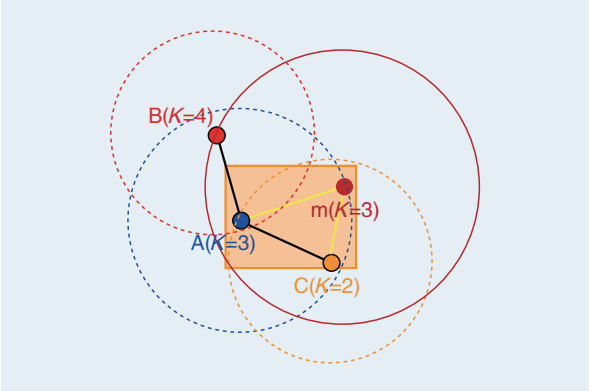


图7 个性化隐私需求匿名方法

连续查询隐私保护

前面的隐私保护工作都是针对Snapshot查询类型。如果将现有的匿名算法直接应用于连续查询隐私保护将产生查询隐私泄露。如图8所示，系统中存在6个用户{A,B,C,D,E,F}。攻击者知道存在连续查询，但并不知道连续查询是什么，以及是由谁提出的。在3个不同时刻t_i、t_{i+1}、t_{i+2}，用户A形成了3个不同的匿名集，即{A, B, D}、{A, B, F}、{A, C, E}。将三个匿名集取交，即可获知是用户A提出的查询Q1。

此问题主要是由同一用户（A）在其有效生命期内形成的匿名集不同而造成的。所以解决方法[16]是让连续查询的用户在最初时刻形成的匿名集在其查询有效期内均有效。同一个例子中，即用户A在t_i时刻形成的匿名集是{A,B,D}，则在t_{i+1},t_{i+2}时刻，匿名集依然是{A,B,D}，如图8（b）和（c）中虚线矩形所示。

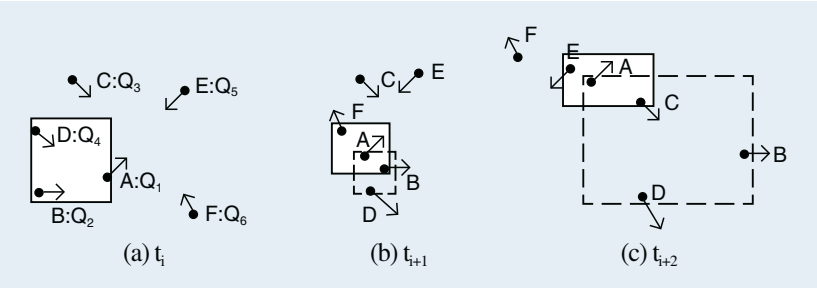


图8 连续查询隐私保护技术

感知查询差异性的隐私保护

位置k-匿名模型只能防止用户与查询建立关联，但不能切断用户与查询内容之间的关联。图9显示的是在位置匿名后发布的匿名位置和查询，符合位置3匿名。但是，攻击者可以确定，位置落于[（1,2）~（5,9）]的第一个匿名集中的用户，一定患了某种疾病。

解决此问题的基本方法[20]是在寻找匿名集的时候考虑查询语义，保证在一个匿名集中敏感查询所占比例不超过p%。如在图10中所示，即使攻击者拥有用户的真实位置，获知该用户落于哪个匿名集中，但他仍然无法获知该用户提出了何种查询。

位置	查询
[（1,2）~（5,9）]	医院
[（1,2）~（5,9）]	诊所
[（1,2）~（5,9）]	医院
[（2,5）~（4,7）]	加油站
[（2,5）~（4,7）]	加油站
[（2,5）~（4,7）]	学校

图9 查询隐私泄露

位置	查询
[（1,2）~（4,7）]	**俱乐部A
[（1,2）~（4,7）]	加油站
[（1,2）~（4,7）]	加油站
[（5,2）~（7,9）]	餐馆
[（5,2）~（7,9）]	诊所
[（5,2）~（7,9）]	学校

图10 感知查询差异性的隐私保护技术

感知隐私保护的查询处理

如何基于匿名后的位置（一个区域）为用户求得查询结果是隐私保护中必须考虑的另一重要问题。

在基于区域位置数据的查询处理技术中，位置数据可以分为两种：公开位置数据和隐私位置数据[16]。公开数据是指如加油站、旅馆和警车等公共信息，其位置是一个精确点；隐私数据属于个人数据，其位置是一个模糊的范围。根据查询点和被查询点是否是隐私数据，可以将查询分为四种（如表2）：基于公开数据的公开查询、基于隐私数据的公开查询、基于公开数据的隐私查询和基于隐私数据的隐私查询。

基于公开数据的公开查询可以用传统方法处理，基于隐私数据的公开查询和基于公开数据的

表2 感知隐私的查询类型

		被查询点	
		公开数据	隐私数据
查询点	公开数据	基于公开数据的公开查询 如：在某电影院200m内所有餐馆	基于隐私数据的公开查询 如：某加油站500米内所有出租车
	隐私数据	基于公开数据的隐私查询 如：距离我最近的加油站	基于隐私数据的隐私查询 如：离我最近的朋友

隐私查询是基于隐私数据的隐私查询的特例。所以这两种查询处理方法经过扩展可以适应第四种情况。

基于隐私数据的公开查询

首先以范围查询为例说明基于隐私数据的公开查询。如查询“某加油站500米内所有出租车”，出租车是空间匿名后得到的区域位置信息，圆是加油站附近500米的范围。最简单的方法[19]是将所有与查询范围相交的匿名框作为候选结果集，匿名框与查询范围重叠区域面积表示查询结果是真正结果的概率，在图11中查询结果即{（B, 50%），（C, 90%），（D, 100%），（E, 60%））}。

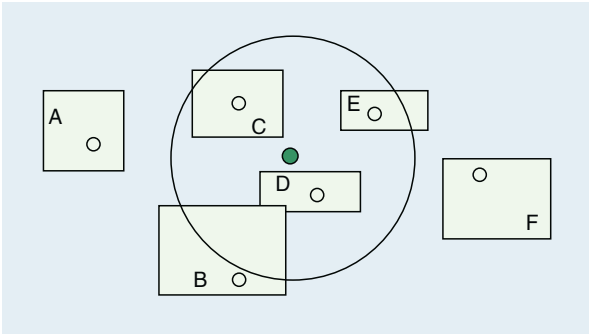


图11 基于隐私数据的公开查询

基于公开数据的隐私查询

查询距离用户现在所在位置最近的加油站。加油站如图12中的p1~p8点，用户的位置是一个匿名区域。为使候选结果集中包含真实结果，需要计算该匿名区域内每一个点的最近邻。这个结果集包含两部分[13]：所有被匿名区域覆盖的点和匿名框边上的每一个点所对应的最近邻。后者可以通过寻找被查询点间的垂直平分线与该边的交点获得。

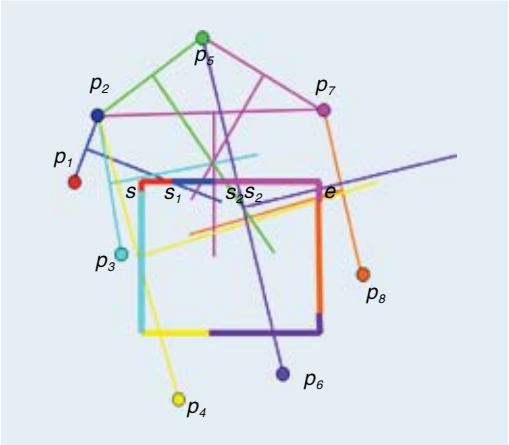


图12 基于公开数据的隐私查询

隐私保护技术面临的挑战

除了上述问题外，在基于位置的服务中隐私保护问题仍面临着很多挑战，如多技术混合的隐私保护技术、轨迹的隐私保护技术和室内位置隐私保护技术等。

多技术混合的隐私保护

如前所述，加密方法安全但不高效，时空匿名高效但相对加密方法而言不够安全。虽然目前大部分研究工作均集中在时空匿名方法上，但是我们试图在加密与时空匿名之间做些工作，研究结合加密算法的高隐秘性和空间匿名算法高效性的混合匿名模型与算法，同时保证利用此种匿名方法所获得数据的可用性，并研究基于混合匿名技术的查询处理算法。

移动轨迹的隐私保护

由于攻击者可能积累用户的历史信息分析用户的隐私，因此还要考虑对用户的连续位置保护的问题，或者说对用户的轨迹提供保护。现有大部分

的轨迹匿名技术[14]多采用发布假数据或丢掉一些取样点的方法。按照前面的分析，这样的方法不够安全，可能通过挖掘历史信息辨别真伪。因此需要研究基于时空匿名的轨迹匿名模型和算法，在保证挖掘结果正确性的前提下保证用户轨迹信息不泄漏。另外，现有的轨迹匿名多是离线（Offline）处理方式。在基于位置的服务中存在汽车导航的应用，用户需查询从A地到B地的行车路线。研究在线轨迹匿名模型和算法是另一个值得关注的问题。

室内位置隐私

研究工作大都专注于室外位置隐私保护，其实在室内也存在隐私泄露的问题。在室内安装无限传感器收集用户位置，可用于安全控制和资源管理，如当室内人数小于某个值时关掉空调设备。但是收集室内人员位置信息的同时可能会泄露个人隐私[12]。如在公司中，管理者可以监控雇员行为，并推测健康状况等。为了保护室内人员的个人隐私，需要针对室内环境特点，研究基于室内位置隐私的攻击模型、匿名模型、匿名算法和查询处理算法。

要解决以上问题，可以将现有技术如数据发布中的隐私保护技术、移动数据的查询处理技术和不确定数据的建模、查询处理技术相结合，这也许会给我们带来一些意想不到的惊喜。■

孟小峰

CCF理事、2009年CCF王选奖一等奖获得者。中国人民大学教授。主要研究方向为网络与移动数据管理等。

xfmeng@ruc.edu.cn

潘　晓

中国人民大学博士生。主要研究方向为移动数据管理，隐私保护。

smallpx@ruc.edu.cn

参考文献

[1] ABI Research, http://www.abiresearch.com/press/1483-Global+LBS+Revenues+to+Reach+ $2.6+Billion+in+2009 4

[2] J. Schiller, Jochen, A. Voisard. Location-based Services, Elsevier Science Ltd, April 200

[3] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of location privacy using dummies for location-based services, In Proc.the 25th International Conference on Distributed Computing Systems(ICPS’05), 2005

[4] M. Gruteser. and D. Grunwald. Anonymous Usage of Location-based Services Through Spatial and Temporal Cloaking, In Proc. of the International Conference on Mobile Systems, Applications, and Services (MobiSys’03), 2003，pp.163~168

[5] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model, In Proc. of the International Conference on Distributed Computing Systems (ICDCS’05), 2005

[6] M. F. Mokbel, C. Y. Chow, and W. G. Aref, The New Casper: Query Processing for Location Services without Compromising Privacy, In Proc. of the 32nd International Conference on Very Large Data Bases (VLDB’06), 2006

[7] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In Proc. of SSTD, 2007

[8] G. Ghinita. Understanding the Privacy-Efficiency Trade-off in Location-Based Queries, ACM SIGSPATIAL GIS Workshop on Security and Privacy in GIS and LBS (SPRINGL), November 2008

[9] G. Ghinita, P. Kalnis and S. Skiadopoulos. MobiHide: A Mobile Peer-to-Peer System for Anonymous Location-Based Queries, In Proceedings of International Symposium on Spatial and Temporal Databases (SSTD), July 2007

[10] C. Chow and M. Mokbel. Privacy in Location-based Services: A System Architecture Perspective. The SIGSPATIAL Special Newsletters, SIGSPATIAL Special, July 2009，Vol. 1, No. 2, pages 23~27

[11] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: Anonymous Location based Queries in Distributed Mobile Systems. In Proceedings of International Conference on World Wide Web, WWW, 2007，pages 1~10

更多参考文献请访问：www.ccf.org.cn的“中国计算机学会通讯”栏目

[12] C. Chow, M. F. Mokbel, and T. He. Tinycasper: a privacy-preserving aggregate location monitoring system in wireless sensor networks. In proceedings of SIGMOD08(demo), Vancouver, Canada ,2008，Pages 1307~1310

[13] H. Hu and D. Lee. Range nearest-neighbor query. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2006，18(1):78~91

[14] O. Abul, F. Bonchi, and M. Nanni. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases, In Proc. of International Conference on Data Engineering (ICDE’08), 2008，pp.376~385

[15] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving User Location Privacy in Mobile Data Management Infrastructures, In Proc. of Privacy Enhancing Technology Workshop (PET’06), 2006

[16] C. Chow and M. F. Mokbel, Enabling Privacy Continuous Queries for Revealed User Locations, In Proc. of the International Symposium on Advances in Spatial and Temporal Databases (SSTD’07), 2007

[17] X. Pan, X. Meng, J. Xu. Distortion-based Anonymity for Continuous Query in Location-Based Mobile Services. In the proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), Seattle, Washington，2009，November 4~6

[18] X. Pan, J. Xu, X. Meng. Protecting Location Privacy against Location-Dependent Attack in Mobile Services. In Proceedings of the ACM 17th Conference on Information and Knowledge Management(CIKM2008), page 1475~1476, Napa Valley, California, 2008，October 26~30

[19] O. Wolfson, P.A. Sistla, S. Chamberlain, and Y. Yesha. Updating and Querying Databases that Track Mobile Units, Distributed and Parallel Databases, vol. 7, no. 3,1999，pp. 257~387

[20] Z. Xiao, J. Xu, and X. Meng. p-sensitivity: a semantic privacy protection model for location-based services, Proc. of the 2nd International Workshop on Privacy-Aware Location-based Mobile Services(PALMS’08), Beijing, China, 2008

发表论文精选

ARTICLE IN PRESS

Pervasive and Mobile Computing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Out-of-order durable event processing in integrated wireless networks☆

Chunjie Zhou [a,*], Xiaofeng Meng [a], Yueguo Chen [b]

[a] School of Information, Renmin University of China, Beijing, China

[b] Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, Beijing, China

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Durable events
Complex events
Wireless networks
Out-of-order

ABSTRACT

Many events in real world applications are long-lasting events which have certain durations. The temporal relationships among those durable events are often complex. Processing such complex events has become increasingly important in applications of wireless networks. An important issue of complex event processing is to extract patterns from event streams to support decision making in real-time. However, network latencies and machine failures in wireless networks may cause events to be out-of-order. In this work, we analyze the preliminaries of event temporal semantics. A tree-plan model of out-of-order durable events is proposed. A hybrid solution is correspondingly introduced. Extensive experimental studies demonstrate the efficiency of our approach.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, the demand for all types of wireless services, as well as the increasing number of users, have led to wireless networks being widely used in various areas [1–4]. Meanwhile, a wide range of wireless network applications nowadays rely on the ability of query processing over event streams [2,5–7]. These event streams are sequences of durable (interval-based) events that are temporally ordered or out-of-order. The temporal relationships among atomic events are very important in identifying event patterns in integrated wireless networks. However, the existing literature on time management of events in integrated wireless networks focuses on either ordered durable events [8–11] or out-of-order instant events [1,12–14]. It is very important to support out-of-order durable event pattern detection in integrated wireless networks. To the best of our knowledge, no prior work on out-of-order events considers the time intervals of events. For pattern queries of durable events, the state transition depends on not only the type of events, but also the relationships and the predicates among events. However, due to NFA (Non-deterministic Finite Automata) explicitly ordering state transitions, the prior NFA-based methods are not straightforward to efficiently model negation and parallel events, which are common in durable events. In this paper, we propose to combine techniques on addressing the detection of durable events and out-of-order events.

Existing research works [5,6] almost focus on instant events, i.e., events with no duration. However, purely sequential queries on instant events are not enough to express many event patterns in the real world. For example, it has been observed that for many diabetic patients, the presence of hyperglycemia often overlaps with the absence of glycosuria [11,15]. This insight has led to the development of effective diabetic testing kits. There is a need for an efficient algorithm that can solve durable events. To model the relationships of events based on their time intervals, we propose two types of event query pattern, the sequential query pattern and the parallel query pattern.

Real-time processing of event streams generated from distributed devices is a primary challenge for today's monitoring and tracking applications. Most systems [11,16], either event-based or stream-based, assume events satisfy totally ordered

☆ This research was partially supported by the grants from the Natural Science Foundation of China (No. 60833005); the National High-Tech Research and Development Plan of China (No. 2009AA011904); and the Doctoral Fund of Ministry of Education of China (No. 200800020002).

* Corresponding author. Fax: +86 10 62512719.
E-mail address: lucyzcj@ruc.edu.cn (C.J. Zhou).

1574-1192/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2010.11.010

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

arrivals. However, in pervasive computing environments, event streams might be out-of-order at the processing engine due to machine or network failures. It has been illustrated that the existing technologies are likely to fail in such circumstances because of false missing or false positive matches of event patterns. Let us take an application for tracking books in a bookstore [7] as an example. In this application, RFID tags are attached to each book and RFID readers are placed at some strategic locations (e.g., book shelves, checkout counters and the exit) throughout the store. If two readers at a book shelf and an exit sensed the same book, but none of the checkout counters sensed the book in between the occurrence of the first two readings, then it is likely that the book is being shoplifted. If events of readings at the checkout counters arrive out-of-order, we may miss the desired results or generate false alarms for event monitoring. The correctness of event processing therefore cannot be guaranteed. Obviously, it is imperative to deal with both in-order as well as out-of-order event arrivals efficiently and in real-time.

In addition, most of the recently proposed complex event processing systems use NFA to detect event patterns [7,13,15]. However, the NFA-based approaches have three limitations [2]: (1) current NFA-based approaches impose a fixed evaluation order determined by their state transition diagram; (2) it is not straightforward to efficiently model negation in an NFA; (3) it is hard to support parallel events because NFA's explicitly order state transition. In this paper, we use tree-based query plans for both logical and physical representations of query patterns.

The main contributions of this work include:

1. We define the notations of temporal semantics and introduce two types of query patterns based on time intervals: sequential query pattern and parallel query pattern.
2. We propose a model of out-of-order durable events that includes the logical and physical expression, as well as the detection method of these events.
3. We use a tree-based query plan structure for complex event processing that is amenable to a variety of algebraic optimizations.
4. We develop a hybrid solution to solve out-of-order durable event processing that can switch from one level of output accuracy to another in real time.

The rest of this paper is organized as follows. Section 2 gives the related works. Section 3 provides the research background and some preliminaries. Section 4 introduces the model of out-of-order durable events and the detection method. Section 5 describes a hybrid solution and the optimization strategy. Section 6 gives the experiment results and we conclude in Section 7.

2. Related works

There has been some work on investigating the problems of out-of-order events and durable events respectively. However, to the best of our knowledge, there is no work considering the two aspects together, which is important in integrated wireless networks.

Studies on the problem of out-of-order events can be divided into two categories: one focuses on real time, whose output is unordered; the other pays more attention to the accuracy, whose output is ordered. Because the input event stream to the query processing engine is unordered, it is reasonable to produce unordered output events. In [13], the authors permit outputting unordered sequences and propose an aggressive strategy. The aggressive strategy produces maximal output under the assumption that out-of-order events are rare. In contrast, to tackle the unexpected occurrence of an out-of-order event, appropriate error compensation methods are designed for the aggressive strategy.

If ordered output is required, additional semantic information such as K-Slack factor [1,12] or punctuation [13] is required to "unblock" the on-hold candidate sequences from being output. The introduction of the two techniques is as follows. A naive approach [1,12] on handling out-of-order event streams is to use K-Slack as an a priori bound on the out-of-order input streams. It buffers incoming events in the input queue for K time units until the order can be guaranteed. The biggest drawback of K-Slack is the rigidity of K, which cannot adapt to the variance in the network latencies that exist in a heterogeneous RFID reader network. For example, one reasonable setting of K may be the maximum of the average latencies in the network. However, as the average latency changes, K may become either too large (thereby buffering unneeded data and introducing unnecessary inefficiencies and delays), or too small (thereby becoming inadequate for handling the out-of-order arriving events and resulting in inaccurate results). It also requires additional space and introduces more latency before evaluating events.

Another solution proposed to handle out-of-order data arrivals is to apply punctuations. This technique assumes assertions are inserted directly in the data stream in order to confirm that a certain value or time stamp will no longer appear in the future input streams [13,17]. The authors in [13] use this technique and propose a solution called the conservative method. It works under the assumption that out-of-order data may be common, and thus produces output only when the correctness can be guaranteed. A partial order guarantee (POG) model is proposed under which such correctness can be guaranteed. Such techniques are interesting, but they require some services first to be created, appropriately inserting such assertions.

On durable events, there have been a stream of research works [10,11,16,18]. Kam and Fu [18] designed an algorithm that uses the hierarchical representation to discover frequent temporal patterns. However, the hierarchical representation is ambiguous and many spurious patterns are found. Papapetrou et al. [16] proposed the H-DFS algorithm to mine frequent

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

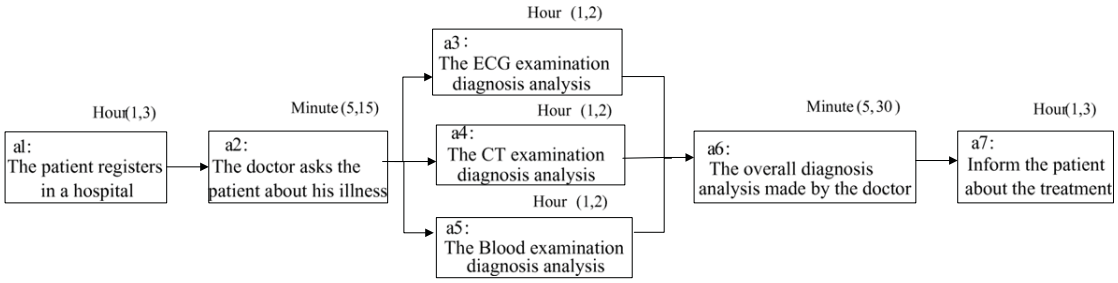


Fig. 1. An example of medical treatment flow.

arrangements of temporal intervals. Both these works transform an event sequence into a vertical representation using id-lists. The id-list of one event is merged with the id-list of other events to generate temporal patterns. This strategy does not scale well when the length of temporal patterns increases. Wu and Chen [10] devised an algorithm called TPrefix for mining non-ambiguous temporal patterns from durable events. TPrefix first discovers single frequent events from the projected database. Next, it generates all the possible candidates between the temporal prefix and discovered frequent events, and scans the projected database again for support counting. TPrefixSpan has several inherent limitations: multiple scans of the database are required and the algorithm does not employ any pruning strategy to reduce the search space. In order to overcome the above drawbacks, the authors of [11] give a lossless representation to preserve the underlying temporal structure of the events, and propose an IEMiner algorithm to discover frequent temporal patterns from durable events. However, they only use this representation for classification. The problem of out-of-order events is not considered.

3. Background and preliminaries

3.1. Research background

In medical industry, wireless network technology can effectively improve the efficiency of collaborative operations among doctors, nurses and relevant departments. Instead of cumbersome paper-based processes, most of the current hospitals track and share patient information such as medical reports, test results, and X-rays electronically. Wireless access to electronic medical records improves both the productivity of clinical staff and the quality of services. In Fig. 1, we take "the medical treatment flow" as an example to exhibit the motivation of our research.

As shown in Fig. 1, after a patient registers in a hospital, he will be dispatched to the corresponding department where he will consult a doctor about his illness. If the disease can be clearly judged, the patient will receive a diagnostic opinion directly. If further diagnosis analysis is required, the patient will be suggested to go to other departments for further examinations, such as ECG, CT and Blood tests. Doctors/nurses in these departments perform the suggested examinations and then return results back to the first doctor. The first doctor then synthesizes the diagnostic opinions from all sides, makes a decision and returns the diagnostic opinion back to the patient.

In this example, $Hour(1,\ 3)$ stands for the limits of duration of the event (the minimum is 1 h and the maximum is 3 h). Suppose many people take examinations in the hospital. Due to the network latency or some other reasons, an earlier examination may arrive at the doctor later. Although the ECG, CT and Blood samples can be sent to different departments in a certain order, the processing time might not be the same for each event, and therefore, the outgoing event stream might be out-of-order.

3.2. Temporal semantics

Each event has an ID and two timestamps. The application timestamp records the time that the event providers generate the events; the arrival timestamp is the time that events arrive at the consumer (responsible for processing the event). The application time can be further refined as a valid time and an occurrence time [3,19]. In the following we will introduce some special attributes of event timestamps in pervasive computing.

Definition 1 (*Time Granularity*)**.** The time granularity is the granularity used for describing the temporal constraints of events, such as second, minute, hour, day, week, month, year, and so on.

In the example of Fig. 1, the time granularity includes minute and hour. Before comparing them, a certain granularity should be chosen first. However, the proportion ratio between two cases of granularity may not be fixed. For example, the ratio of week and month, workday and hour are all not fixed.

Definition 2 (*Time Interval*)**.** Suppose $H = \{T_1, \ldots, T_n\}$ is a linear hierarchy of time units. In it, for all $1 \leq i < n,\ T_i \subseteq T_{i+1}$. For instance, $H = \{\text{minute, hour, day, month, year}\}$ and $minute \subseteq hour$. A time interval in H is an n-tuple $(t_1, \ldots, t_n)$ such that for all $1 \leq i \leq n$, t_i is a time-interval in the time unit of T_i.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

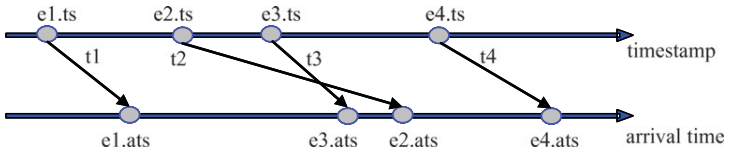


Fig. 2. Out-of-order event.

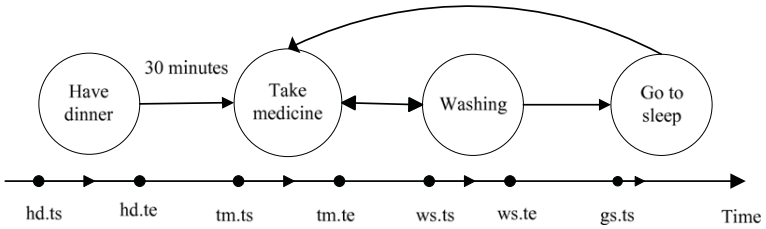


Fig. 3. Sequential query pattern.

Time intervals are ordered according to the lexicographic ordering $<_H$. Thus, time interval $T = (t_1, \ldots, t_n) <_H T' = (t'_1, \ldots, t'_n)$ iff there exists an $i (1 \leq i \leq n)$ such that $t_i <_{T_i} t'_i$ and $t_j = t'_j$ for all $j = i+1, \ldots, n$. Note that if $i = n$, then $t_j = t'_j$. When $T <_H T'$, we say that T occurs before T'. If $T = T'$, we say that T occurs simultaneously with T'.

For example, in Fig. 1, the total time interval of a1 and a2 is denoted as $T = Hour(1, 3) + Minute(5, 15)$, and the total time interval of a6 and a7 is $T' = Hour(1, 3) + Minute(5, 30)$. By using hierarchy $H = minute, H' = minute \subseteq hour, T <_{H'} T'$, because $15 <_H 30$.

Definition 3 (*Out-of-Order Event*)**.** Let $e.ats$ and $e.ts$ be the arrival timestamp and the occurrence timestamp of an event e. Consider an event stream S:$e_1, e_2, \ldots, e_n$, where $e_1.ats < e_2.ats < \cdots < e_n.ats$. For any two events e_i and e_j $(1 \leq i, j \leq n)$ from S, if $e_i.ts < e_j.ts$ and $e_i.ats < e_j.ats$, we say the stream is an ordered event stream. If however $e_j.ts < e_i.ts$ and $e_j.ats > e_i.ats$, then e_j is an out-of-order event.

In the example of Fig. 2, the timestamps of events $e1 \sim e4$ are listed in order. But we can see that event $e2$ arrives later than event $e3$, which is called out-of-order.

3.3. Query patterns

Query patterns specify how individual events are filtered and how multiple events are correlated via time-based and value-based constraints. Based on the time interval of events, query patterns can be classified into two categories: sequential query patterns and parallel query patterns.

3.3.1. Sequential query pattern

Sequential query pattern is a basic function supported by most event processing systems. The ability to synthesize events based on the ordering of previous events is useful for complex event detection. For efficiency in a stream setting, all operators that produce outputs involving more than one input event have a temporal constraint, denoted as w. For example, $seq(e_1, e_2, \ldots, e_n; w)$ outputs a complex event with $t_1 = e_1.starttime$, $t_2 = e_n.endtime$ if (i) $\forall i$ in $1, \ldots, n-1$ we have $e_i.endtime < e_{i+1}.starttime$ and (ii) $t_2 - t_1 \leq w$. Hence, *seq* constrains events of it to occur in order without overlapping.

For a concrete example of elder-care shown in Fig. 3, we should infer the elder's activities over time and remind them of something important (e.g., taking medicine), because they may be very forgetful. Suppose the man should take medicine within 30 min after having dinner. If we have already received the start and end time of having dinner, the start and end time of washing, but no start or end time of taking medicine, before the start time of going to sleep, an alarm should be issued to the man. For another example of Fig. 1, the tasks of cooperative medical treatment from "the patient registers in a hospital" to "the doctor asks the patient about his illness", to "the CT examination diagnosis analysis", to "inform the patient about the treatment", can also be regarded as a sequential query pattern.

3.3.2. Parallel query pattern

Parallel query pattern can be regarded as another feature for event processing systems, especially for durable events. In order to improve the performance, one may detect complex events concurrently. Parallel query pattern also has a temporal constraint, denoted as w. For example, $pal(e_1, e_2, \ldots, e_n; w)$ outputs a complex event with $t_1 = \min\{e_1.starttime, \ldots, e_n.starttime\}$, $t_2 = \max\{e_1.endtime, \ldots, e_n.endtime\}$, where $t_2 - t_1 \leq w$. Hence, *pal* permits overlapping among events and may cause events to be out-of-order. Parallel pattern includes conjunction and disjunction.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

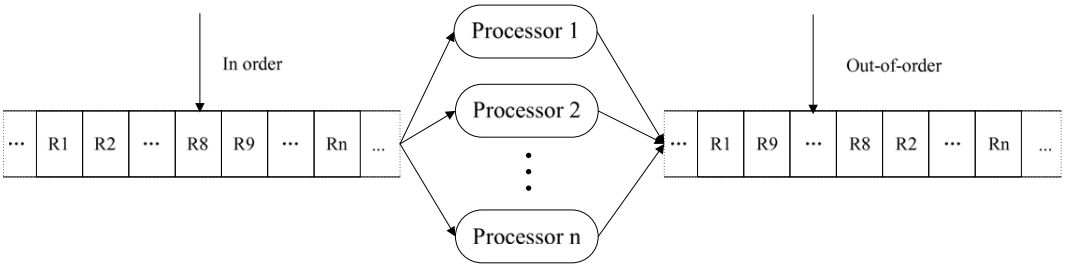


Fig. 4. Parallel query pattern.

Conjunction means both event *A* and event *B* occur within w, and their order does not matter. Disjunction means either event *A* or event *B* or both occurs within w.

For an example shown in Fig. 4, suppose there is a promotion in a bank in which the first *N* customers who satisfy certain conditions can get thank-you packs. Because there are a large number of customers in a bank, several processors are required to process the customers' requirements in parallel. Due to different service rates in different processors, the user who applied first may leave later. Also in the example of Fig. 1, the tasks of cooperative medical treatment "the ECG examination diagnosis analysis", "the CT examination diagnosis analysis" and "the Blood examination diagnosis analysis" can also be regarded as a parallel query pattern.

4. Out-of-order durable event detection

After introducing what out-of-order events are, and the problems caused by them, we start to present our method on how to detect out-of-order durable events in this section.

4.1. The model of durable events

As shown in Section 3, each event is denoted as $(ID, V_s, V_e, O_s, O_e, S_s, S_e, K)$. Here V_s and V_e denote the valid start and end time respectively; O_s and O_e denote the occurrence start and end time respectively; S_s corresponds to the system clock time upon event arrival; S_e means the system clock time when an event ends; K corresponds to an initial insert and all associated retractions, each of which reduces the S_e compared to the previous matching entry. Besides time stamps, the event may also have some other attributes, such as value, price, name, and so on.

In the following query format, Event Pattern connects events together via different event operators; the WHERE clause defines the context for event pattern by imposing predicates on events; the WITHIN clause describes the time range during which a matching event pattern must occur. Real-time Factor specifies the real-time requirement of different users.

```
<Query>::=EVENT <event pattern>
   [WHERE <value constraints>]
   [WITHIN <time constraints>]
   [Real-time Factor {0,1}]
   <event pattern>::=SEQ/PAL((Ei(relationship)Ej)
                    (!Ek)(relationship)El))(1≤i,j,k,l≤n)
   relationship::=contain,overlap,before,after,meet,finish,equal
   <time constraints>::= Time Window length W
```

The cooperative medical treatment flow in Fig. 1 can be expressed as Query Q1. *A* stands for the doctor asks the patient about his illness. *B*, *C* and *D* denote the ECG, CT and Blood examination diagnosis analysis respectively. *E* shows the overall diagnosis analysis made by the doctor. Events *B*, *C* and *D* can be executed in parallel, which are allowed to overlap with *E*.

```
  Query Q1. Cooperative Medical Treatment Pattern
EVENT PATTERN   SEQ(A(Before)PAL(B,C,D)(Overlap)E)
WHERE           A.Oe<B.Os
AND             A.Oe<C.Os
AND             A.Oe<D.Os
AND             B.price<C.price
AND             B.Oe>E.Os
AND             C.Oe>E.Os
AND             D.Oe>E.Os
WITHIN          6 workdays
Real-time Factor   1
```

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

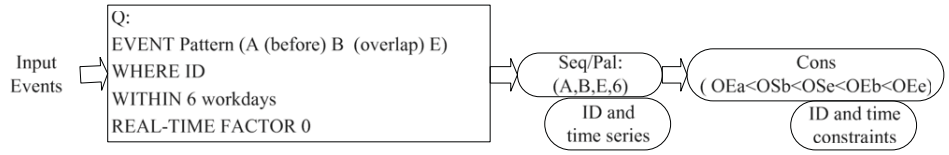
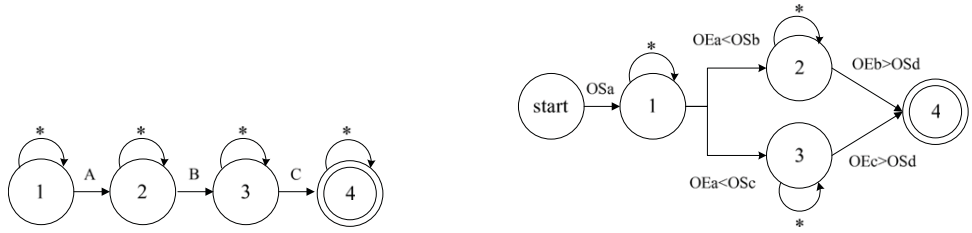


Fig. 5. Logical expression of query Q.



(a) NFA-based status transition. (b) The extended NFA expression.

Fig. 6. NFA-based expression.

4.1.1. Logical expression

A query expressed by the above language is translated into a query plan composed of the following operators: Sequential/Parallel Pattern (*Seq*/*Pal*), Negation Pattern (*Neg*), and Constraints (*Cons*) [7]. An event e_i is a positive (resp. negative) event if there is no '!' (resp. with '!') symbol used. The *Seq*/*Pal* operator denoted $Seq/Pal(E_1, E_2, \ldots, E_n, window)$ extracts all events matching to the positive event pattern specified in the query and constructs positive sequential/parallel events. *Seq*/*Pal* also checks whether all matched event sequences occur within the specified sliding window. The detailed definition and constraints of *Seq*/*Pal* can be found in Section 3.3. The *Neg* operator specified by $Neg(!E_1, (time\ constraint); \ldots; !E_m, (time\ constraint))$ checks whether there exist negative events within the indicated time constraint in a matched positive event pattern. In this paper, we don't introduce *Neg* in detail, because its operations are very similar to *Seq*/*Pal*, as in [13]. The *Cons* operator expressed as *Cons*(*P*), where *P* denotes a set of constraints on event attributes, further filters event patterns by applying the relationship specified in the query. Query Q1 can be simplified as *Q*, and Fig. 5 shows an example of the algebra plan for the pattern query *Q*.

4.1.2. Physical expression

Currently, non-deterministic finite automata (NFA) is the most commonly used method for evaluating complex event queries [2]. As the example shown in Fig. 6(a), the method starts at state 1, transits to state 2 when event *A* occurs, then to state 3 when *B* occurs, and finally to the output state when *C* occurs. A pattern is said to be matched when the NFA transitions into a final state. However, due to NFA's explicitly order state transitions, the prior NFA-based methods are not straightforward to efficiently model negation and parallel events. Here we extend the expression of NFA and process *Q* 1 as shown in Fig. 6(b). State 2 and 3 are internal states reached after a *B* or *C* event is received. State 4 is a final state indicating a match for the pattern. However, because there is a predicate involving events *B* and *C* ($B.price < C.price$), there is no simple way to evaluate the predicate when a *B* event arrives, because the corresponding event *C* may not arrive yet. Hence, it has difficulty to decide a transition for event *B*. As a result, this extended NFA also cannot be used for parallel query pattern processing with predicates, which is common in durable events.

For query patterns of durable events, the state transition depends on not only the type of events, but also the relationships and the predicates among events. In this paper, we propose to use tree-based query plans for both the logical and physical representation of query patterns. To process a query, we should first transform the query into an internal tree. Leaf nodes buffer atomic events as they arrive. Internal nodes buffer the intermediate results assembled from sub-tree buffers. Each internal node associates with one operator of the plan, along with a collection of predicates. Fig. 7(a) shows a tree plan for *Q* 1. This is a left-deep plan, because *A* and *Pal* are first combined, and their outputs are matched with *D*. A right-deep plan, where *Pal* and *D* are first combined, and then matched with *A*, is also possible.

Each node of the tree has a stack to temporarily store incoming events (for leaf nodes) or intermediate results (for internal nodes). Each stack contains a number of records, each of which has a vector of event pointers, including a start time and an end time of the event. For the start time of each instance in the stack, an extra field named *PreEve* records the nearest instance in terms of time sequence in the stack of the previous state (shown in Algorithm 1). While for the end time of each instance, *PreEve* is first set to its corresponding start time. When its start time becomes the *PreEve* of another instance, its *PreEve* changes to this instance. For example, in Fig. 7(b), the *PreEve* of $OE_a(7)$ is first set as $OS_a(3)$. The most recent instance in stack S_a of type A before $OS_b(6)$ is $OS_a(3)$. *PreEve* field of $OS_b(6)$ is set to $OS_a(3)$, as shown in the parenthesis preceding $OS_b(6)$, then the *PreEve* of $OE_a(7)$ changes to $OS_b(6)$. The start time and the end time of an internal node are the timestamps of the earliest and the latest atomic event comprising this complex event.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

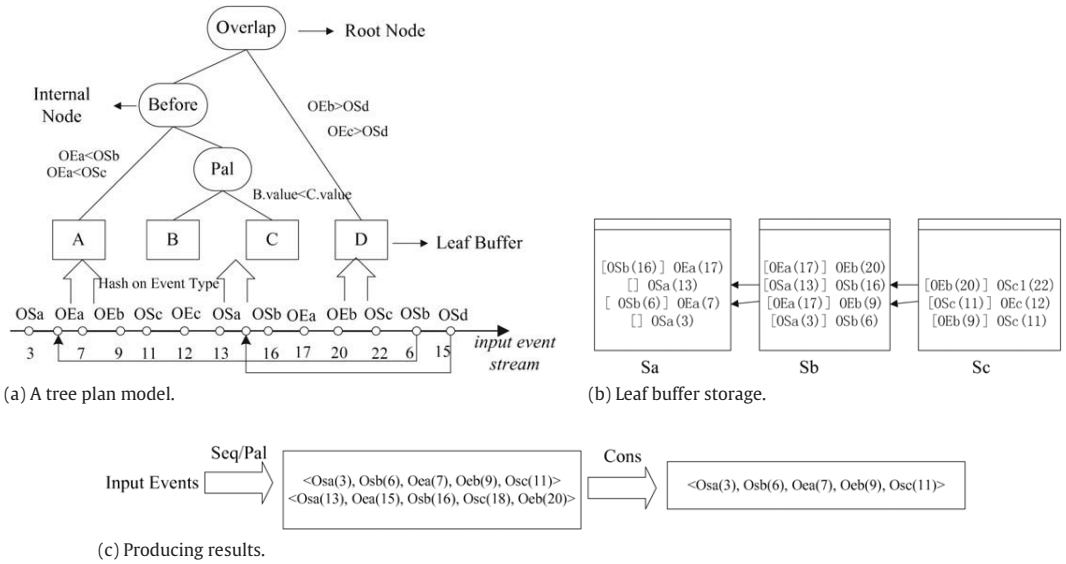


Fig. 7. Tree plan physical expression.

Algorithm 1 Storage Pattern of the Stack

Input:
The number of events in the stack, N;
The occurrence start time of the event, OS;
The occurrence end time of the event, OE;
One of the events that is already stored in the stack, k;

Output:
The correct TList of the stack;
string[ ] A;

```
for i = 0; i < N; i++ do
  for j = 1; j <= N; j++ do
    if A[i].OE has arrived and there is no k satisfying the prior pointer of A[k] is A[j].OS then
      the prior pointer of A[j].OS is set to A[i].OE;
      the prior pointer of A[j].OE is set to A[j].OS;
    else if A[i].OE has arrived and there is a k satisfying the prior pointer of A[k] is A[j].OS then
      the prior pointer of A[j].OS is set to A[i].OE;
      the prior pointer of A[j].OE is set to A[k];
    else if A[i].OE has not arrived and there is no k satisfying the prior pointer of A[k] is A[j].OS then
      the prior pointer of A[j].OS is set to A[i].OS;
      the prior pointer of A[j].OE is set to A[j].OS;
    else
      the prior pointer of A[j].OS is set to A[i].OS;
      the prior pointer of A[j].OE is set to A[k];
    end if
  end for
end for
```

Fig. 7(b) shows the stacks of each node in the tree model. In each stack, its instances are naturally sorted from top to bottom in the order of their arrival timestamps (shown in Algorithm 2). For in-order events, each received event is simply appended to the end of the corresponding stack and its *PreEve* field is set to the last event in the previous stack. However, this simple appended semantics is not applicable for the insertion of out-of-order events. An out-of-order event $e_i \in E_i$ will be allocated by the corresponding stack of type E_i in StateStack sorted by arrival timestamp. The *PreEve* field of event e_k's start time (i.e., $e_k.starttime$) in the adjacent stack with $e_k.starttime >_H e_i.starttime$ ($e_i.endtime$) will be set to $e_i.starttime$ ($e_i.endtime$), if the end time of all events in StateStack of type E_k has arrived; otherwise, it should wait until the absent end time. The *PreEve* field of event e_k's end time (i.e., $e_k.endtime$) is set to the corresponding start time $e_k.starttime$, if the start time has arrived. For example, in Fig. 7(c), suppose there are out-of-order events $OS_b(8)$ and $OE_b(10)$. The *PreEve* of $OS_b(8)$

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

Table 1
Annotated history table.

K	$Sync$	O_s	O_e	S_s	S_e
E_0	1	1	10	0	7
E_0	5	1	5	7	10

should not be set to $OS_a(3)$ until $OE_b(9)$ arrives, because before time point 8, there is only a start time of event type B, but no end time. In the same way, the *PreEve* of $OE_b(10)$ is set to $OS_b(8)$, because its corresponding start time has arrived.

Algorithm 2 Insert Operation of the Stack

Input:
The current number of events in the stack, N;
The occurrence start time of the event, OS;
The occurrence end time of the event, OE;
The new received event ID, k;
The maximal size of the stack, MS;
Output:
The correct TList of the stack;

```
1:  while k < MS do
2:    if all events in the stack are in order then
3:      the prior pointer of A[k].OS is set to A[N].OE;
4:      the prior pointer of A[k].OE is set to A[k].OS;
5:    else if there are out-of-order events in the stack, and the end time of all events have arrived then
6:      for i = N − 1; i >= 0; i − − do
7:        if A[k].OS > A[i].OS then
8:          the prior pointer of A[k].OS is set to A[i].OS;
9:        else
10:         the prior pointer of A[k].OS is set to A[i].OE;
11:         the prior pointer of A[k].OE is set to A[k].OS;
12:       end if
13:     end for
14:   else
15:     wait until there is no absent end time;
16:   end if
17: end while
```

4.2. The detection of out-of-order durable events

In order to detect out-of-order durable events, one more notation should be defined, named a synchronization point. We describe an annotated form of the history table which introduces an extra column, called *Sync*. A table with such a column is shown in Table 1. The extra column ($Sync$) is computed as follows: for insertions $Sync = Os$; for retractions $Sync = O_e$. The intuition behind the *Sync* column is that it induces a global notation of out-of-order events. For instance, if and only if the global ordering of events achieved by sorting events according to S_s is identical to the global ordering of events achieved by sorting events according to the compound key $\langle Sync, S_s \rangle$, then there are no out-of-order events in the stream.

5. Solution

5.1. Basic framework and expression

The framework of our method is shown in Fig. 8, which includes three components. The "Terminal Layer" component involves mobiles, PDAs, laptops, etc., which are the sources of events. The raw data generated from different data sources are ordered. "Event Processing Engine" stores the received events from "Terminal Layer", and handles some query processing. During the process of transmitting to "Event Processing Engine", network latencies and machine failures may cause events to be out-of-order. There are two data transition methods between "Terminal Layer" and "Event Processing Engine": push-based and pull-based, which will be discussed in Section 5.3. "Database Management" conserves historical records, event relationships and some knowledge base rules, as we have introduced in [20]. "Database Management" is really an integration of historical event records coupled with real-time event records. A knowledge base also should be stored in the "Database Management", which includes the extra information, such as the spatial location information, and the possible actions in a certain place. Relations identify the relationships among incoming atomic events in "Terminal Layer". In "Database

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

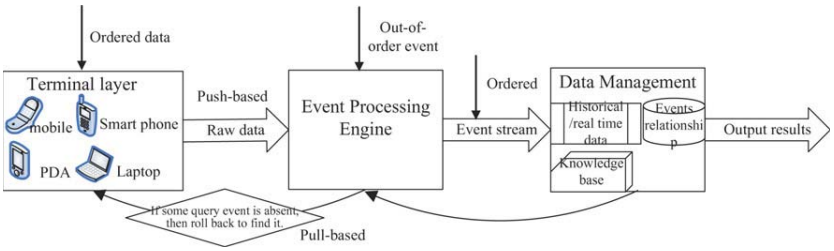


Fig. 8. Durable out-of-order solution framework.

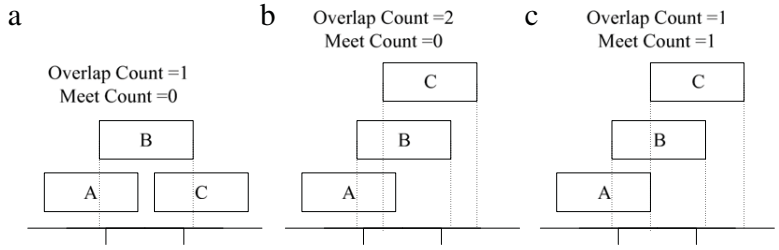


Fig. 9. Interpretation of pattern $(A\ (Overlap)\ B)\ (Overlap)\ C$.

Management" the instantaneous relation is used to denote a relation in the traditional bag-of-tuples sense, and a relation to denote a time-varying bag of tuples.

Besides the framework, specific query expression is also a challenge. Different from point-based queries, the expression of durable event queries may be ambiguous. Multiple interpretations may result in an incorrect inference of the exact relationship among events. For example, the same expression query $(A\ (overlap)\ B\ (overlap)\ C)$ may have different meanings, as shown in Fig. 9. In order to overcome this and distinguish the different interpretations of temporal patterns, the hierarchical representation with additional information is required [11]. Therefore, 5 variables are proposed, namely, contain count c, finish by count f, meet count m, overlap count o, and start count s to differentiate all the possible cases. The representation for a complex event E to include the count variable is shown as follows:

$$E = (\ldots (E_1\ R_1[c, f, m, o, s]\ E_2)\ R_2[c, f, m, o, s]\ E_3 \ldots\ R_{n-1}[c, f, m, o, s]\ E_n). \tag{1}$$

Thus, the temporal patterns in Fig. 9 are represented as:
(A Overlap [0,0,0,1,0] B) Overlap [0,0,0,1,0] C
(A Overlap [0,0,0,1,0] B) Overlap [0,0,0,2,0] C
(A Overlap [0,0,0,1,0] B) Overlap [0,0,1,1,0] C.

In the real world, different applications have different requirements for consistency. Some applications require a strict notion of correctness, while others are more concerned with real-time output. When exposed to users and handled by the system, users can specify consistency requirements on a per query basis and the system can adjust consistency at runtime. So we add an additional attribute ("Real-time Factor") to every query, as shown in Section 4.1. If the users focus on real-time output, the "Real-time Factor" is set to "1"; Otherwise, it is set to "0". Due to users' different requirements of consistency, there are two different methods, which are introduced as follows.

5.2. Real-time based method

If the "Real-time Factor" of a query is set to "1", the goal is to send out results with as small latency as possible based on the assumption that most data arrives in time and in order. Once out-of-order data arrival occurs, we provide a mechanism to correct the results that have already been erroneously output. This method is similar to, but better than the method in [13] because of the tree-plan expression, which can reduce the compensation time and frequency, as shown in Section 4.1.2.

At first, users submit queries to "Events Buffer", which handles the query processing and outputs the corresponding results directly. This method guarantees the real-time requirements and takes some urgent actions timely. However, in the case of out-of-order events, the output results may be wrong or the correct results may be lost. In order to compensate for this, two kinds of stream messages are used. Insertion $\langle +, E\rangle$ is induced by an out-of-order positive event [13], where "E" is a new event result. Deletion $\langle -, E\rangle$ is induced by an out-of-order negative event, such that "E" consists of the previously processed event. Deletion tuples cancel event results produced before which are invalidated by the appearance of an out-of-order negative event.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

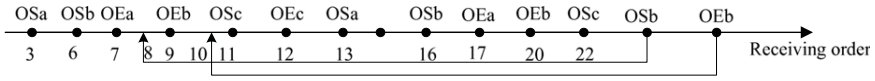


Fig. 10. Input events.

For example, the query is $(A(overlap)B(!D)(before)C)$ within 10 min. A unique time series expression of this query $\{OS_a, OS_b, OE_a, OE_b, OS_c, OE_c\}$ can be obtained based on the above interval expression method. For the event stream in Fig. 10, when an out-of-order *seq/pal* event $OS_b(6)$ is received, a new correct result $\{OS_a(3), OS_b(6), OE_a(7), OE_b(9), OS_c(11), OE_c(12)\}$ is output as $\langle +, \{OS_a(3), OS_b(6), OE_a(7), OE_b(9), OS_c(11), OE_c(12)\}\rangle$. when an out-of-order negative event $OS_d(15)$ is received, a wrong output result $\{OS_a(13), OS_b(16), OE_a(17), OE_b(20), OS_c(22)\}$ is found. So we send out a compensation $\langle -, \{OS_a(13), OS_b(16), OE_a(17), OE_b(20), OS_c(22)\}\rangle$.

When out-of-order events occur, these compensation operations should also be stored in "Database Management". Then, after a period of time, the query results generated from "Event Buffer" should first be transmitted into "Database Management", which checks the results again based on the historical records and knowledge base rules. If the results and the contents in "Database Management" are positively correlated, then output the final result; otherwise, the result should be held for a certain time, which is application-dependent. This optimization method consumes a little time, but it can eliminate many wrong results and compensation operations. From the global view, its performance is better than the existing methods.

5.3. Correct based method

If the "Real-time Factor" of a query is set to "0", the goal is to send out every correct result with less concern about the latency. Considering the time intervals, the existing methods can be improved as follows.

5.3.1. Query processing

Using our framework above, when the user submits a query to "Events Buffer", we first extract the corresponding sequential/parallel event patterns. Based on the event model introduced in Section 4.1, we can get the event sequence by a backward and forward depth first search in the DAG. The forward search is rooted at the start time of this instance e_i and contains all the virtual edges reachable from e_i. The backward search is rooted at the end time of event instances of the accepting state. It contains paths leading to and thus containing the event e_i. One final root-to-leaf path containing the new event e_i corresponds to one matched event sequence. If $e_i.endtime$ (resp. $e_i.starttime$) belongs to the accepting (resp. starting) state, the computation is done by a backward (resp. forward) search only.

Meanwhile, we can transform the query into a certain time series based on the above 5 variables, which make the representation of relationships among events unique. Compared with the time series of the query, the set of event sequences can be further filtered. For example, the precedence relationship among start time and end time of different events, the time window constraints, as well as negative events among the event sequence. After all these steps, the remaining event results are transmitted into a buffer in the "Database Management".

The buffer in the "Database Management" is proposed for event buffer and purging using the K-ISlack semantics. Different from the previous K-Slack method, we consider the time interval in this paper. It means that both the start time and the end time of the out-of-order event arrivals are within a range of K time units. That is, an event can be delayed for at most K time units. The buffer compares the distance between the checked event and the latest event received by the system. A CLOCK variable equal to the largest end time seen so far for the received events is maintained. The CLOCK is updated constantly. According to the sliding window of semantics, for any event instance e_i kept in the buffer, it can be purged from the stack if $(e_i.starttime + W) < CLOCK$. Thus, under the out-of-order assumption, the above condition will be $(e_i.starttime + W + K) < CLOCK$. This is because after waiting for K time units, no out-of-order events with start time less than $(e_i.starttime + W)$ can arrive. Thus e_i can no longer contribute to forming a new candidate sequence.

5.3.2. Optimization

In order to make some optimization, we divide the buffer into two parts: outdated event instances and up-to-date event instances, based on window constraints. A divider is set for the buffer: instances on or above it are outdated instances and instances below it are up-to-date ones. The part of outdated event instances stores the event sequence which falls out of the time window; while the up-to-date event instances keep the event sequence which is less than the allowed window range. For a stack without outdated events, the divider is set to *NULL*, while an in-order event triggers sequence construction. Only the events under the divider in each stack will be considered.

In addition, when out-of-order events occur, there may be some delay of attributes. For example, the end time of an event has been received, but no start time of it. As in Fig. 8, we take some active actions to the absent attributes, instead of waiting positively. If "Event Processing Engine" cannot find an attribute of the query, it will go back to data sources to see whether the absent one happened or not. If there is no such an attribute in data sources, then the corresponding query results are output or discarded directly. Otherwise, "Event Processing Engine" keeps waiting until the attribute arrives, then outputs through "Database Management". As introduced in Section 5.1, the "Database Management" involves both historical records and knowledge base data, which can filter some incorrect results before generating outputs.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

Table 2
Parameters and performance metrics.

Terminology	Meaning
P_{io^3}	Out-of-order event percentage
Buf	Buffer size of tree pattern
QL	Event's query length
NoR	Number of results
NoC	Number of compensation results
NoCR	Number of correct results
K	Maximum delay of out-of-order events
AET	Average execution time
AL	Average latency
RoC	Rate of compensation, NoC/NoR
ACC	Accuracy of results, NoCR/NoR

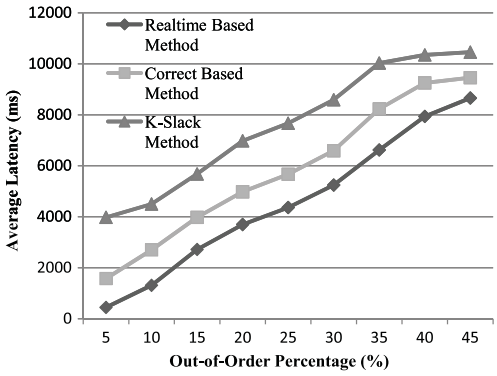


Fig. 11. Trend of average latency.

6. Experiments

In order to test and verify the above two algorithms, we designed an experimental environment to simulate the events generation and queries. A prototype using the C# language has been implemented.

6.1. Experimental environments

Our experiments involve two parts: one is the event generator; another is the event processing engine. The event generator is used for generating different types of events continuously. We adopt multi-thread to model different sensors to produce different events randomly. Then the generated events are sent to the event receiver, which is a part of event processing engine. The event processing engine includes two units: the receiver unit and the query unit. The former is just responsible for receiving the events from "sensors"; the latter takes charge of queries, and outputs the correct results. Meanwhile, it records the performance information which is shown in Table 2.

Our experiments run on two machines, with Intel Dual-Core 2.0 GHz and 2.5 GHz CPU, 2.0 G and 3.0 G RAM respectively. PC1 is used for running the Event Generator programs and PC2 for the Event Processing Engine. In PC1, we created about 1000 generators ("sensors"), each of which can produce more than 1000 different-type (A, B, C or D) events randomly. So at least 1,000,000 events will reach the receiver hosted in PC2 and wait to be queried. Based on such a large scale of event data, our experiments can test and verify the performance of the algorithms much better. Additionally, in order to make the experimental results more convincing, we run the program for 300 times, and take the average value of all results. In the following, we will focus on the key performance metrics shown in Table 2.

6.2. Experimental results

Figs. 11–15 mainly examine the impact of out-of-order percentage P_{io^3} on the performance metrics. P_{io^3} is varied from 0% to 45%. Fig. 11 shows the case when there no durable events arrive. From the figure, the average latency of three methods (Realtime Based Method, Correct Based Method and *K*-Slack Method) increases with the enlargement of out-of-order percentage, and Realtime Based Method increases faster than the other two methods, because we add the cost of compensation operations into the definition of average latency. However, if there are durable events, the naive *K*-Slack method will not work, while the trend of Realtime Based Method and Correct Based Method are almost the same, as shown in Fig. 11.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

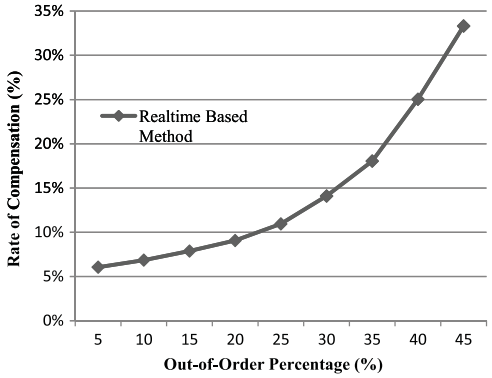


Fig. 12. Trend of rate-of-compensation.

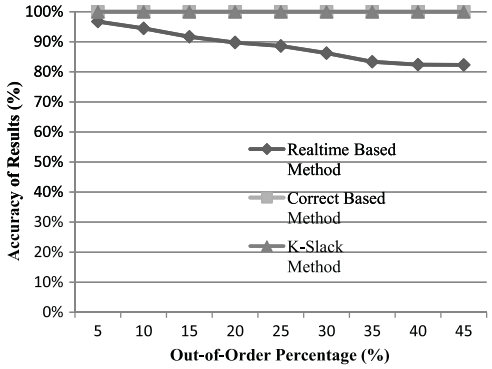


Fig. 13. Accuracy without durable events.

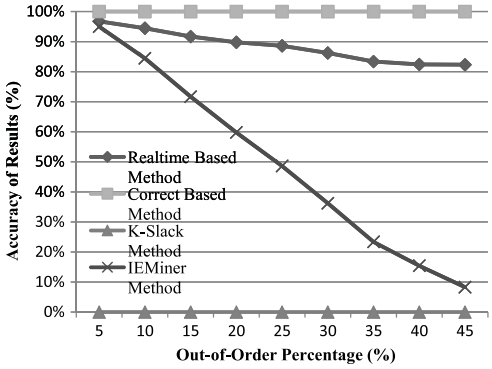


Fig. 14. Accuracy with durable events.

Fig. 12 just concerns Realtime Based Method, which has compensation operations. The rate of compensation is determined by (*NoC*/*NoR*). From the figure, we can see that with an increase of out-of-order percentage, more compensation operations are generated, and the speed of compensation rate is faster and faster.

The accuracy of results is also examined, defined as (*NoCR*/*NoR*). Fig. 13 shows the accuracy of three methods when there are no durable events. In this case, the accuracy of Correct Based Method and *K*-Slack Method are both independent of out-of-order percentage, while Realtime Based Method drops with the enlargement of out-of-order percentage. This is because with larger out-of-order percentage, more output results should be compensated.

Fig. 14 shows the accuracy of four methods (Realtime Based Method, Correct Based Method, *K*-Slack Method and IEMiner Method) when there are durable events. In this case, the accuracy of *K*-Slack Method is almost zero, because it cannot deal with out-of-order durable events. With the enlargement of out-of-order percentage, the accuracy of IEMiner Method drops fast, because it can only deal with durable events, but not out-of-order events. The accuracy of Realtime Based Method and Correct Based Method are similar to the case in Fig. 13.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

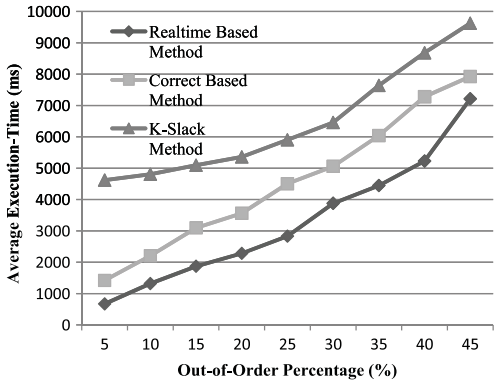


Fig. 15. Trend of average execution time.

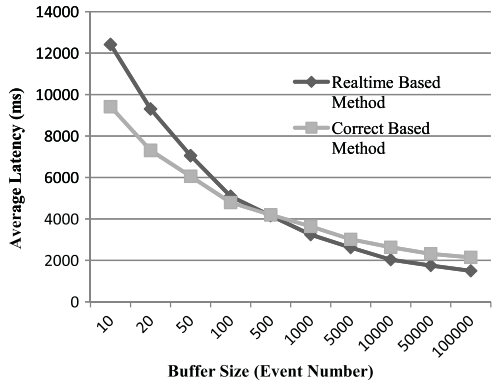


Fig. 16. Trend of average latency.

We examine the average execution time in Fig. 15, which denotes the summation of operator execution times. When there are no durable events, two observations can be found: (1) the average execution time increases as the percentage of out-of-order events increases because more recomputing operations are needed; (2) the average execution time of Correct Based Method is larger than Realtime Based Method at beginning, while with the enlargement of out-of-order percentage, they will tend to the same. But the execution time of *K*-Slack Method is always larger than the other two methods. If there are durable events, the execution time of *K*-Slack Method tends to be infinity, while the trend of Realtime Based Method and Correct Based Method are almost unchanged.

Figs. 16–18 show the impact of buffer size on performance metrics. Fig. 16 shows that the average latency of both methods decreases with the enlargement of buffer size. When the buffer size in tree-pattern is less than 500 *event number*, the average latency of Correct Based Method is less than Realtime Based Method; while the opposite situation happens when the buffer size is larger than 500 *event number*. I.e., the dropping ratio of Realtime Based Method is faster than Correct Based Method, or the buffer size has much more impact on Realtime Based Method. That is because when the buffer is too small, there must be a lot of incorrect results output, which cause too many compensation operations and extend the latency. While when the buffer size is large enough, the compensation results decrease quickly, so the average latency of Correct Based Method is larger than Realtime Based Method again.

Fig. 17 shows the accuracy trends of both methods with different buffer size. When the buffer size is near to zero, the accuracy of both methods is also about 0%, because there are almost no results generated now. However, when the buffer size is a little larger, the accuracy of both methods increases immediately. Because we can always get the prior events from the "Data Management Center", the accuracy of both methods almost stays the same with the enlargement of buffer size. That is to say, the parameter of buffer size has little effect on accuracy.

The trend of average execution time is shown in Fig. 18, which is similar to the trend of average latency. There is only a constant difference between them, from the first event's arrival time to the corresponding last event's.

Fig. 19 shows the impact of event query length on average execution time when there are no durable events. From the figure, we can see the trend can be divided into two parts. When the event query length is shorter, the average execution time of Correct Based Method and *K*-Slack Method is larger than Realtime Based Method. With the enlargement of event query length, they tend gradually to the same, and then Realtime Based Method becomes the largest. That is because when

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

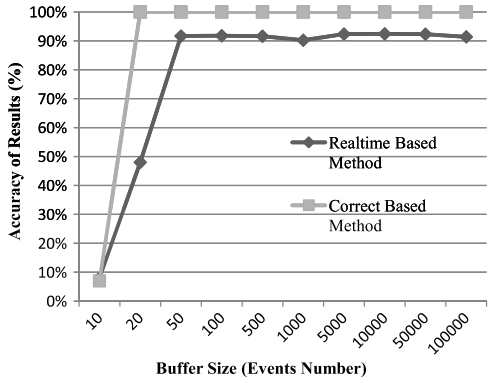


Fig. 17. Accuracy of methods.

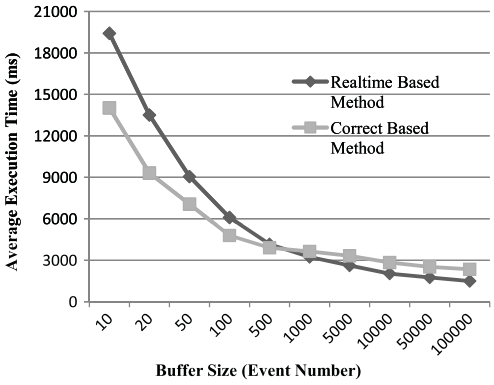


Fig. 18. Execution time on buffer size.

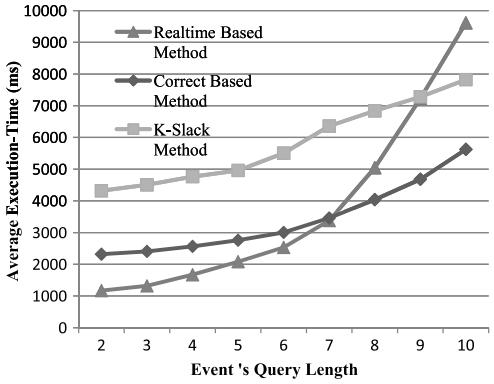


Fig. 19. Execution time on event length.

the event query length is too long, there must be many compensation operations of Realtime Based Method. The average execution time of K-Slack Method is always larger than Correct Based Method. Compared with Realtime Based Method, event query length has less impact on Correct Based Method and K-Slack Method. If there are durable events, the execution time of K-Slack Method tends to be infinite.

Fig. 20 shows the latency of the three methods increases with the increase of event query length when there are no durable events. From the figure, we can see that Realtime Based Method increases faster than the other two methods. The latency of K-Slack Method is always larger than Correct Based Method. If there are durable events, the latency of K-Slack Method tends to be infinite, because it cannot deal with durable events.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

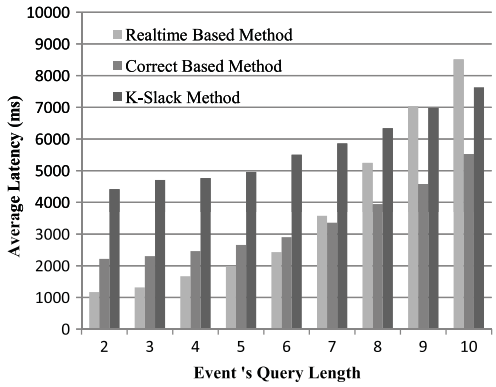


Fig. 20. Latency on event length.

7. Conclusion and future work

The goal of this work is to solve query processing of out-of-order durable events in integrated wireless networks. We proposed a tree-plan model of out-of-order durable events, which can give the logical and physical expressions. We also combined techniques on addressing the detection of durable events and out-of-order events. A hybrid solution to solve out-of-order events is studied, which can switch from one level of output accuracy to another in real time. The experimental study compares with the method with *K*-Slack and IEMiner methods, and demonstrates the effectiveness of our proposed approach.

In this paper, we primarily consider incoming events for occurrences of user-specified event patterns. In the future work, mining the automatic complex events patterns of out-of-order durable events will be studied. In addition, we will consider query optimization strategies and algorithms in the "Data Management Center", so that the query frequency can be regarded as another influencing factor on performance metrics.

Acknowledgement

We would like to thank Pengfei Dai of Beijing University of Post and Telecommunications for his helpful comments on the experiments.

References

[1] S. Babu, et al., Exploiting *K*-constraints to reduce memory overhead in continuous queries over data streams, ACM Transaction on Database Systems 29 (3) (2004) 545–580.
[2] Y. Mei, S. Madden, ZStream: a cost-based query processor for adaptively detecting composite events, in: Proceedings of the 35th SIGMOD International Conference on Management of Data, SIGMOD, 2009, pp. 193–206.
[3] M. Akdere, U. Cetintemel, N. Tatbul, Plan-based complex event detection across distributed sources, in: Proceedings of the 34th International Conference on Very Large Data Bases, VLDB, 1, 1, 2008, pp. 66–77.
[4] A.H.M. Amin, A.I. Khan, Collaborative-comparison learning for complex event detection using distributed hierarchical graph neuron (DHGN) approach in wireless sensor network, Australasian Conference on Artificial Intelligence (2009) 111–120.
[5] C. Antunes, A.L. Oliveira, Generalization of pattern growth methods for sequential pattern mining with gap constraints, Machine Learning and Data Mining in Pattern Recognition (2003).
[6] J. Pei, J. Han, B. Mortazavi, H. Pinto, Q. Chen, Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth, in: Proceedings of the 17th International Conference on Data Engineering, ICDE, 2001, pp. 215–226.
[7] E. Wu, Y. Diao, S. Rizvi, High performance complex event processing over streams, in: Proceedings of the 32th SIGMOD International Conference on Management of Data, SIGMOD, 2006, pp. 407–418.
[8] D. Alex, R. Robert, V.S. Subrahmanian, Probabilistic temporal databases, ACM Transaction on Database Systems 26 (1) (2001) 41–95.
[9] R.S. Barga, J. Goldstein, M. Ali, M.S. Hong, Consistent streaming through time: a vision for event stream processing, in: The 3rd Biennial Conference on Innovative Data Systems Research, IDAR, 2007.
[10] S. Wu, Y. Chen, Mining nonambiguous temporal patterns for interval-based events, IEEE Transactions on Knowledge and Data Engineering 19 (6) (2007) 742–758.
[11] D. Patel, W. Hsu, M.L. Lee, Mining relationships among interval-based events for classification, in: Proceedings of the 34th SIGMOD International Conference on Management of Data, SIGMOD, 2008, pp. 393–404.
[12] M.A. Hammad, et al. Scheduling for shared window joins over data streams, in: The 29th International Conference on Very Large Data Bases, Vol. 29, 2003, pp. 297–308.
[13] M. Liu, M. Li, D. Golovnya, E.A. Rundenstriner, K. Claypool, Sequence pattern query processing over out-of-order event streams, in: Proceedings of the 25th International Conference on Data Engineering, ICDE, 2009, pp. 274–295.
[14] S. Eyerman, L. Eeckhout, T. Karkhanis, J.E. Smith, A mechanistic performance model for superscalar out-of-order processors, ACM Transactions on Computer Systems (TOCS) 27 (2) (2009).
[15] F. Wang, S. Liu, P. Liu, Complex RFID event processing, The International Journal on Very Large Data Bases (VLDBJ) 18 (4) (2009) 913–931.
[16] P. Papapetrou, G. Kollios, S. Sclaroff, D. Gunopulos, Discovering frequent arrangements of temporal intervals, in: Proceedings of the 5th IEEE International Conference on Data Mining, ICDM, 2005.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

[17] L. Ding, N. Mehta, E.A. Rundensteiner, G.T. Heineman, Joining punctuated streams, Advances in Database Technology (EDBT) (2004) 587–604.
[18] P.S. Kam, A.W. Fu, Discovering temporal patterns for interval-based events, in: Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery, DaWak, 2000, pp. 317–326.
[19] S. Roger, J.G. Barga, A. Mohamed, M. Hong, Consistent streaming through time: a vision for event stream processing, in: 3rd Biennial Conference on Innovative Data Systems Research, CIDR, 2007.
[20] C.J. Zhou, X.F. Meng, A framework of complex event detection and operation in pervasive computing, in: The Ph.D. Workshop on Innovative Database Research, IDAR, 2009.

Please cite this article in press as: C.J. Zhou et al., Out-of-order durable event processing in integrated wireless networks, Pervasive and Mobile Computing (2010), doi:10.1016/j.pmcj.2010.11.010

计算机研究与发展 ISSN 1000-1239/CN 11-1777/TP
Journal of Computer Research and Development 47(1): 121-129, 2010

基于位置服务中的连续查询隐私保护研究

潘 晓 郝 兴 孟小峰
(中国人民大学信息学院 北京 100872)
(smallpx@ruc.edu.cn)

Privacy Preserving Towards Continuous Query in Location-Based Services

Pan Xiao, Hao Xing, and Meng Xiaofeng
(*School of Information, Renmin University of China, Beijing* 100872)

Abstract With advances in wireless communication and mobile positioning technologies, location-based mobile services have been gaining increasingly popularity in recent years. Privacy preservation, including location privacy and query privacy, has recently received considerable attention for location-based mobile services. A lot of location cloaking approaches have been proposed for protecting the location privacy of mobile users. However, they mostly focus on anonymizing snapshot queries based on proximity of locations at query issued time. Therefore, most of them are ill-suited for continuous queries. In view of the privacy disclosure (including location and query privacy) and poor quality of service under continuous query anonymization, a δ_p-privacy model and a δ_q-distortion model are proposed to balance the tradeoff between privacy preserving and quality of service. Meanwhile a temporal distortion model is proposed to measure the location information loss during a time interval, and it is mapped to a temporal similar distance between two queries. Finally, a greedy cloaking algorithm (GCA) is proposed, which is applicable to both anonymizing snapshot queries and continuous queries. Average cloaking success rate, cloaking time, processing time and anonymization cost for successful requests are evaluated with increasing privacy level (k). Experimental results validate the efficiency and effectiveness of the proposed algorithm.

Key words privacy; continuous query; quality of service; LBS; mobile computing

摘 要 近年来,伴随着移动计算技术和无限设备的蓬勃发展,位置服务中的隐私保护研究受到了学术界的广泛关注,提出了很多匿名算法以保护移动用户的隐私信息.但是现有方法均针对 snapshot 查询,不能适用于连续查询.如果将现有的静态匿名算法直接应用于连续查询,将会产生隐私泄露、匿名服务器工作代价大等问题.针对这些问题,提出了 δ_p-隐私模型和 δ_q-质量模型来均衡隐私保护与服务质量的矛盾,并基于此提出了一种贪心匿名算法.该算法不仅适用于 snapshot 查询,也适用于连续查询.实验结果证明了算法的有效性.

关键词 隐私;连续查询;服务质量;基于位置服务;移动计算

中图法分类号 TP392

收稿日期: 2009-06-26; **修回日期**: 2009-09-29

基金项目: 国家自然科学基金项目(60833005, 60573091);国家"八六三"高技术研究发展计划基金项目(2007AA01Z155, 2009AA011904);高等学校博士学科点专项科研基金项目(20080020002)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

随着移动计算技术和无线设备的结合, 随时随地获得个人精确位置成为可能, 促进了新一类应用程序——位置服务(location based service, LBS)的产生和发展. 但是, 人们在享受各种位置服务带来便捷的同时, 个人隐私信息泄漏问题逐渐引起广大学者的关注, 成为近年来研究的热点问题之一[1].

一般来讲, 位置服务中的隐私保护可以分为两种: 位置隐私[2]和查询隐私[3]. 如张某利用自己带有GPS的手机提出"寻找5 min内距离我最近的肿瘤医院". 这是导航系统中常见的连续最近邻查询. 一方面, 用户不想让任何人知道他现在所在位置(如"医院"); 另一方面用户也不想让任何人获知自己提出了哪方面的查询请求, 如与某特定肿瘤相关的医院查询. 前者属于位置隐私保护范畴, 后者属于查询隐私保护范畴.

为了解决位置服务中的隐私保护问题, Gruteser等人[4]提出了位置k-匿名模型: 当一个移动用户的位置无法与其他$(k-1)$个用户的位置相区别时, 称此位置满足位置k-匿名. 此模型既适用于位置隐私保护, 也适用于查询隐私保护. 以位置隐私为例, 如图1(a)所示, 用户A, B, C匿名后的位置用矩形R表示, 攻击者仅知道R中包含3个用户, 但无法确定每个用户的确切位置, 从而保护了用户的位置隐私. 类似地, 用户提出的查询隐私可以通过相同的方法得到保护. 为了解决查询隐私保护中查询差异性问题, Xiao Zhen等人[5]提出了p-敏感模型. 此模型考虑了查询敏感度和语义差异性, 要求在一个匿名集中敏感查询个数所占比例不能超过p. 如图1(b)所示, 矩形R中提出了3种查询: {肿瘤医院, 旅馆, 加油站}, 其中有关肿瘤医院的查询具有敏感性, 用户$A/B/C$提出此查询的概率均为1/3. 如果$p=1/2$, 则在这个例子中满足1/2-敏感模型.

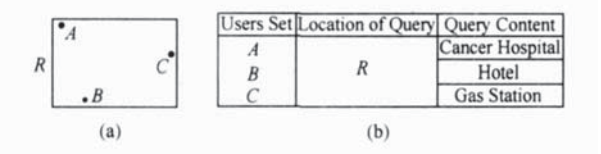


Fig. 1 Privacy in LBS. (a) Location privacy and (b) Query privacy.

图1 位置服务中的隐私保护. (a) 位置隐私; (b) 查询隐私

位置服务中现有的隐私保护工作均针对snapshot查询类型. 然而, 连续查询是位置服务中一种常见并重要的查询类型, 具有位置频繁更新和时效性①的特点. 如果将现有的匿名算法直接应用于连续查询隐私保护将产生以下3个问题: 第一, 连续查询隐私泄露, 如图2所示, 系统中存在{A, B, C, D, E, F} 6个用户, 攻击者知道存在某个连续查询, 但并不知道连续查询是什么以及是由谁提出的, 在3个不同时刻t_i, t_{i+1}, t_{i+2}, 用户A形成了3个不同的匿名集, 即{A, B, D}, {A, B, F}, {A, C, E}, 如图2中实线矩形框所示, 将3个匿名集取交, 即可获知是用户A提出的连续查询以及此连续查询类型; 第二, 加剧了匿名服务器的负担. 移动对象连续发生位置更新, 并且每发生一次更新均需要为新位置重新生成一个匿名框, 造成了匿名服务器负担过重, 从而变成系统瓶颈; 第三, 很多网络资源被浪费于传输频繁的位置更新和新生成的匿名集, 造成网络拥堵.

上述问题主要是由同一用户(A)在其有效生命期内形成的匿名集不同而造成的. 所以最简单方法是让连续查询的用户在最初时刻形成的匿名集在其查询有效期内均有效. 在前面的例子中, 用户A在t_i时刻形成的匿名集是{A, B, D}, 则在t_{i+1}, t_{i+2}时刻匿名集依然是{A, B, D}, 如图2中虚线矩形所示. 很明显, 这种方法将产生新的问题是: 第一, 位置隐私泄露, 如在图2(b)中, 在t_{i+1}时刻, A, B, D位置过于邻近, 造成匿名框过小(极端情况下集中于一点), 位置隐私泄露; 第二, 服务质量QoS(quality of services)降低. 服务质量与数据精度成反比. {A, B, D}在t_{i+2}时刻分布在距离较远的位置, 形成的匿名框过大, 造成过高的查询处理代价.

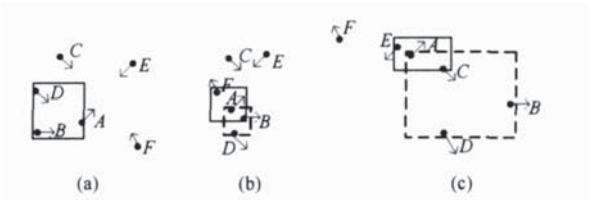


Fig. 2 Privacy preserving for continuous query. (a) Cloaking set at t_i; (b) Cloaking set at t_{i+1}; and (c) Cloaking set at t_{i+2}.

图2 连续查询隐私保护. (a) 在时刻t_i的匿名集; (b) 在时刻t_{i+1}的匿名集; (c) 在时刻t_{i+2}的匿名集

由此可见, 仅仅简单地把在最初时刻形成的匿名集作为连续查询有效期内的匿名集返回并不能解决问题. 其主要原因是现有的匿名算法仅考虑用户当前位置的邻近性, 忽略了用户未来的位置. 用户

① 时效性是指查询具有一定的生命周期, 在特殊情况下, snapshot查询其生命周期为0.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

位置邻近性会随着移动用户的运动而改变. 更具体地说, 当前邻近的对象可能在下一个时刻随着对象的散开而距离很远; 而当前距离很远的对象可能在下一个时刻重合于一个点. 所以, 此问题的难点在于: 在用户查询的有效期内选取哪一个时刻的位置进行匿名, 形成的匿名集既不会产生位置隐私泄露, 也不会产生查询隐私泄露①, 同时可以保证最好的服务质量.

针对这一问题, 提出了 δ_p-隐私模型和 δ_r-质量模型来均衡(trade-off)隐私保护与服务质量这一矛盾; 通过匿名框的周长形式化定义匿名位置的可用性, 并将其定义为两个移动对象间的时序相似性, 利用两个对象的相似性提出了一个贪心算法, 从而保护用户的位置隐私和查询隐私. 该匿名算法既适用于 snapshot 查询, 也适用于连续查询.

1　相关工作

位置隐私保护和查询隐私保护是移动计算环境中主要考虑的两个隐私问题. 最初人们并没有把位置隐私与查询隐私分开, 而是将其合二为一地看待, 即保护了位置隐私等同于保护了查询隐私. 位置 k-匿名模型是学术界广泛接受的匿名模型, 由 Gruteser 等人提出, 随后很多人对此模型进行了修正[5-6].

位置匿名的基本思想分为 3 种: 第一, 发布假位置 (dummy)[7], 即不发布真实服务请求的位置, 假位置和真实位置的距离与隐私保护程度成正比, 和服务质量成反比; 第二, 时空匿名(spatial-temporal cloaking)[2,4], 本质上是降低移动对象位置的时空粒度, 即用时空区域表示用户真实的精确位置, 区域形状不限, 可以是任意形状的凸多边形, 称此匿名区域为匿名框(cloaking region), 匿名框的大小与匿名程度成正比, 与服务质量成反比; 第三, 加密(encryption)[8], 查询点与查询结果对服务提供商来说都是隐秘的. 这里采用的方法属于时空匿名.

文献[3]首次提出了连续查询隐私保护问题, 指出将用户在初始时刻形成的匿名集作为查询有效期内的最终结果, 从而解决连续查询隐私泄露的问题. 但是该工作存在以下缺点: 第一, 如引言部分所述, 该算法生成的匿名集仅考虑移动对象初始时刻位置的邻近性, 忽略对象的运动, 造成位置隐私泄露和糟糕的服务质量; 第二, 该算法要求任何新提出的查询一定要延迟一段时间才能匿名, 从而保证任何一个查询都是从已在系统中存在一定时间的旧用户提出的, 这样的做法造成用户的服务无法即时获得响应, 即使附近已有足够多的用户供其形成匿名集; 第三, 匿名集的形成忽略各个查询有效期的时间差异性, 造成同一匿名集中的查询有效期相差过大. 最坏情况下, 连续查询与 snapshot 查询匿名在一起, 只要有一个用户查询终止, 在此匿名集中的所有查询也被迫终止. 与该工作相比, 提出的算法与其不同点在于: 第一, 考虑了查询生命周期内每一个时刻位置的邻近性; 第二, 被匿名在一起的查询具有时效性相似的特点, 即查询有效期类似; 第三, 同时适用于连续查询和 snapshot 的查询. 文献[9]也解决了连续查询隐私保护的问题. 它假设在匿名区域内, 用户位置并非均匀分布, 采用信息理论中的熵(entropy)来定义用户的隐私保护度. 但是由于熵并不考虑用户的位置是否不同, 可能造成 k 个用户重叠于一点的情况, 从而产生位置隐私泄露.

2　系统结构

与大部分现有工作[2,4]一样, 采用中心服务器结构如图 3 所示, 包括移动用户、匿名服务器和服务提供商. 处理流程为: 移动用户将查询请求 Q 发送给匿名服务器. 查询请求分为新查询(new query)和活动查询(active query)两种. 新查询是指由用户首次提出的查询请求; 活动查询是指用户在过去的时刻提出、现在依然有效的查询, 再次触发仅为位置更新.

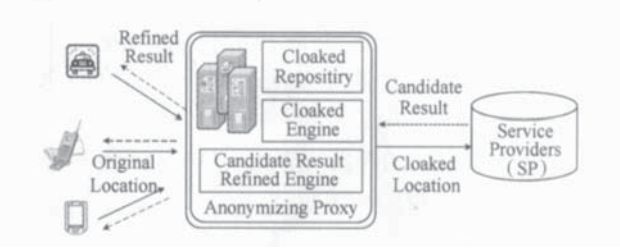


Fig. 3　System architecture for location privacy preserving.

图 3　位置隐私保护系统结构

1) 匿名服务器由知识库(cloaked repository)、匿名引擎(cloaked engine)和查询结果求精处理器(candidate result refined engine)组成. 对于新查询, 匿名引擎触发匿名算法根据系统待匿名查询的当前位置寻找匿名集, 并将匿名集发送到知识库和服务

① 匿名集中查询差异性并不属于本文解决范畴. 此问题可以用现有的 p-敏感模型加以解决.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

提供商; 如果是活动查询, 则直接从知识库中寻找该查询在提出之初形成的匿名集合. 找到该集合, 并根据其中所有对象的当前位置计算新的匿名框, 发送给服务提供商.

2) 服务提供商接收到用户匿名后的位置进行查询处理, 并将查询结果发送给匿名服务器.

3) 匿名服务器中的查询结果求精处理器将对查询结果求精后返回给移动用户.

3 预备知识

定义 1. 查询 Q. 形式化地表示每一个查询 Q 为

$$Q = (l, \boldsymbol{v}, t, T_{\exp}, con),$$

其中:

① $l = (x, y)$ 表示查询 Q 所在位置;

②速度 $\boldsymbol{v} = (v_x, v_y)$ 是一个向量, 其中 v_x/v_y 表示查询在 x/y 轴方向上的运动速度分量;

③ $(l, \boldsymbol{v}, t)$ 表示查询 Q 在时刻 t 的位置在 l 上, 并且运动速度为 $\boldsymbol{v}$;

④ $T_{\exp}$ 表示该查询过期时间;

⑤ con 表示查询内容, 如最近医院等.

定义 2. 匿名集. 形式化的定义匿名集 CR:

$$(CID, Qset, R_{L,t}, R_{v,t}),$$

其中:

① CID 表示匿名集的标识符;

② $Qset$ 是一个集合, 由匿名集中包含的查询组成;

③ $R_{L,t} = (L_{x-,t}, L_{y-,t}, L_{x+,t}, L_{y+,t})$ 表示匿名框, 是覆盖 $Qset$ 中所有用户的最小边界矩形(minimum boundary rectangle, MBR), 其中, $(L_{x-,t}, L_{y-,t})$ 和 $(L_{x+,t}, L_{y+,t})$ 是 MBR 的左下角和右上角在时刻 t 的坐标.

④ $R_{v,t}$ 是 $R_{L,t}$ 的速度边界矩形(boundary velocity rectangle, BVR). $R_{v,t} = (v_{x\min,t}, v_{y\min,t}, v_{x\max,t}, v_{y\max,t})$, 其中 $v_{x\min,t} = \min(v_{x+,t}, v_{x-,t})$, $v_{x\max,t} = \max(v_{x+,t}, v_{x-,t})$, $v_{y\min,t} = \min(v_{y+,t}, v_{y-,t})$, $v_{y\max,t} = \max(v_{y+,t}, v_{y-,t})$. $v_{x-,t}/v_{x+,t}$ 是 MBR 在 x 方向上的左/右边界速度, $v_{y-,t}/v_{y+,t}$ 是 MBR 在 y 方向上的下/上边界速度.

注意 $v_{x\max,t}/v_{y\max,t}$ 和 $v_{x\min,t}/v_{y\min,t}$ 不一定是 $QSet$ 中 x/y 方向上的最小与最大速度. 如图 4 所示匿名集包括 $Q_1 \sim Q_5$ 五个查询, 括号中的数字表示该查询的运动速度. 很明显, 此时 x 轴方向上的 $v_{x\max} = 1$, 但并不是在该方向上的最大速度 $v_{\max} = 3$. 但是随着查询的运动, 存在时刻 t_j, Q_5 超越 Q_3, 此时边界速度 $v_{x\max,tj} = 3$. 所以边界速度会随着查询的运动而改变, MBR 边界的运动是一个分段函数, 如图 5 所示.

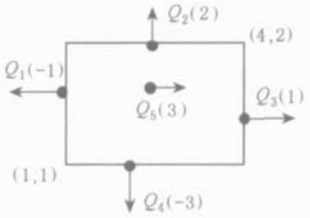


Fig. 4 Boundary velocities.

图 4 边界速度图

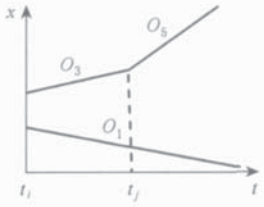


Fig. 5 WB on x-axis.

图 5 x 轴上的边界宽(WB)

定义 3. 边界长与宽. 假设匿名集 CR 在时刻 t 的匿名框为 $R_{L,t}$, 则在 x 轴上匿名框的宽为

$$WB_t = L_{x+,t} - L_{x-,t} = (L_{x+,t_i-1} - L_{x-,t_i-1}) + (v_{x+,t} - v_{x-,t}) \times (t - t_{i-1}). \quad (1)$$

同样地, 在 y 轴上匿名框的高为

$$HB_t = L_{y+,t} - L_{y-,t} = (L_{y+,t_i-1} - L_{y-,t_i-1}) + (v_{y+,t} - v_{y-,t}) \times (t - t_{i-1}), \quad (2)$$

分别记为 WB_t, HB_t.

采用降低位置信息空间粒度的方法保护位置隐私, 显然, 位置信息的粒度越低隐私保护度则越高, 但是数据可用性就越差. 所以我们使用信息扭曲度(distortion)来评价位置数据可用性. 数据扭曲度越高数据可用性则越差.

定义 4. 位置扭曲度(distortion). 假设匿名集 CR, 其匿名框是 $R_{L,t}$. 查询 $Q \in CR$, 在时刻 t, 查询 Q 的位置 l 被概化为 $R_{L,t} = (L_{x-,t}, L_{y-,t}, L_{x+,t}, L_{y+,t})$, 则位置扭曲度定义为

$$Distortion_{R_{v,t}}(Q, R_{L,t}) = \frac{(L_{x+,t} - L_{x-,t}) + (L_{y+,t} - L_{y-,t})}{A_{\text{height}} + A_{\text{width}}},$$

其中 A_{height}, A_{width} 是整个空间的高与宽, $(L_{x-,t}, L_{y-,t})$ 和 $(L_{x+,t}, L_{y+,t})$ 是匿名框 $R_{L,t}$ 在时刻 t 的左下和右上角坐标. 查询 Q 的有效期截至到时刻 $T_{\exp}$, 则 Q 在其有效期内总信息扭曲度为

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$\int_{T_s}^{T_{\exp}} Distortion_{R_{v,t}}(Q, R_{L,t})\,dt,$$

其中 T_s 是查询 Q 匿名成功的时刻.

定义 5. 匿名集的位置扭曲度. 匿名集 CR 的匿名框的边界位置和速度分别为 $R_{L,t}=(L_{x-,t}, L_{y-,t}, L_{x+,t}, L_{y+,t})$, $R_{v,t}=(v_{x\min,t}, v_{y\min,t}, v_{x\max,t}, v_{y\max,t})$. CR 在时刻 t 的位置扭曲度为 CR 中所有查询的扭曲度加和:

$$Distortion_{R_{v,t}}(CR, R_{L,t}) = \sum_{i=1}^{|CR|} Distortion_{R_{v,t}}(Q_i, R_{L,t}) = |CR| \frac{(L_{x+,t}-L_{x-,t})+(L_{y+,t}-L_{y-,t})}{A_{\text{height}}+A_{\text{width}}},$$

其中 $Q_i \in CR$. 在 CR 的有效期内总信息扭曲率为

$$\int_{T_s}^{maxT} Distortion_{R_{v,t}}(CR, R_{L,t})\,dt,$$

其中 T_s 是匿名集 CR 的生成时间, $maxT=\max_{Q\in CR} Q.T_{\exp}$.

很明显, 移动对象的状态(包括初始位置和速度)越相似, 其匿名在一起信息扭曲率则越低; 状态差异越大, 匿名后的信息扭曲率则越高. 所以扭曲率定义任意两个查询在其生命周期内的相似度.

定义 6. 时序相似度. Q_1 和 Q_2 是两个查询, $R_{L\_12,t}$ 是时刻 t 覆盖这两个查询的 MBR, 则 Q_1 与 Q_2 在其生命有效期内的相似度为

$$SimDis(Q_1, Q_2) = \int_{T_s}^{maxT} Distortion_{R_{v\_12,t}}(CS, R_{L\_12,t})\,dt,$$

其中 $maxT=\max(Q_1.T_{\exp}, Q_2.T_{\exp})$, 并且 $R_{v\_12,t}$ 是 MBR 在时刻 t 的边界速度集.

4　隐私模型

本节首先讨论一维的情况: 将查询 Q 的位置和速度分别向 x, y 轴投影. 例如图 6(a)所示, 一个匿名集包含 $\{Q_1, Q_2, Q_3\}$ 3 个查询, 直线斜率表示查询在 x 轴方向上的速度. 从图中可以看出, 在 t_w 时刻, 3 个查询具有相同的 x 坐标, 该匿名集的一维信息泄露. 同理, y 轴也存在类似的情况. 最坏情况下, 匿名框从 x, y 方向上同时收缩, 并缩为一点, 如图 6(b)所示, 此时查询位置隐私泄露. 但是, 无论哪一种情况我们都认为是不允许的. 精确位置泄露可以看作是位置一维信息泄露的特殊情况. 所以, 只要保证匿名框的长和宽在查询有效期内的任何一个时刻均不会缩小为一点则可以保证位置隐私不泄露.

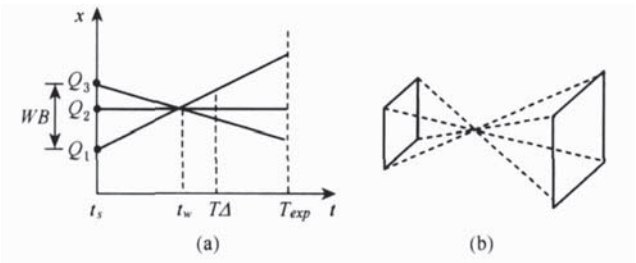


Fig. 6　Location privacy disclosed for continuous query. (a) In one dimension and (b) In two dimension.

图 6　连续查询位置隐私泄露示例. (a) 一维情况; (2) 二维情况

定义 7. δ_p-隐私模型. 设 WB_t/HB_t 是匿名框在时刻 t 的宽/高, δ_p 是用户指定的一维情况下最小的位置粒度, 则

$$\forall t \in [T_s, maxT], \min(WB_t, HB_t) \geqslant \delta_p \times P_A,$$

其中 $P_A=A_{\text{width}}+A_{\text{height}}$, 称该匿名集满足 δ_p-隐私模型.

匿名集除要满足最低隐私需求 δ_p 之外, 其位置信息扭曲度也不能过高, 否则影响服务质量. 所以, 为保证服务质量, 用户定义最高信息扭曲度 δ_q. 由于移动对象的运动, 位置扭曲度随时间不断变化. 匿名集 CR 在时刻 t_i 的信息丢失率不高于 δ_q 并不代表在查询有效期内一直大于 δ_q.

定义 8. δ_r-质量模型. 假设用户可以容忍的最差服务质量是 δ_q, 匿名集 CR 的位置匿名框为 $R_{L,t}$, 伴随边界速度 $R_{v,t}$, 则对于

$$\forall t \in [T_s, maxT], \forall Q \in CR, Distortion_{R_{v,t}}(Q, R_{L,t}) \leqslant \delta_q,$$

则称该匿名集满足 δ_r-质量模型.

为简便起见, 假设系统具有统一的隐私度需求 k, 综上所述, 成功的匿名集需要满足以下 3 个条件:

1) $|CR| \geqslant k$;

2) 设 $minT=\min_{Q\in CR} Q.T_{\exp}$, $maxT=\max_{Q\in CR} Q.T_{\exp}$, $maxT-minT<\delta_t$;

3) 匿名集 CR 满足 δ_r-隐私模型和 δ_r-质量模型.

其中, 第 1 个条件符合位置 k-匿名模型, 要求在一个匿名集中至少包含 k 个查询; 第 2 个条件保证了同在一个匿名集中的查询, 具有时效相似的特点, 即查询有效期的差距不大于 δ_t; 第 3 个条件试图在隐私保护和服务质量上寻找平衡点.

5　贪心匿名算法

算法的主要思想是当新查询 r(以后称此查询为触发查询)到来时, 依次扫描待查询集合 $RSet$ 中

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

待匿名且并未过期的查询. 判断如果这两个查询形成匿名集, 是否满足 δ_t-质量模型. 如果满足则计算这两个查询时序相似度, 将 r 与具有最小相似度的查询聚集在一起; 如果不满足则取下一个查询. 重复上述过程, 直至包含该查询的候选匿名集从 *RSet* 再也找不到合并的查询. 最后, 如果候选匿名集的大小, 即包含的用户数大于隐私度 k, 则调用算法 3, 判断该候选集是否满足 δ_p-隐私模型. 具体算法参见算法 1.

在算法 1 中包含 3 个重要步骤: 候选匿名集边界对象检测、δ_t-质量模型检测和 δ_p-隐私模型检测. 后面 3 节将逐一介绍.

算法 1. 贪心匿名算法(GCA).

/* 当有新查询 r 到来时* /
① candidate cloaking set U= null;
② put r into U;
③ for each query r_m in *RSet* /* 依次扫描 *RSet* 中待匿名的查询* /
④ if | $r.T_{exp} - r_m.T_{exp}$ | > δ_T /* 判断时效近似性* /
⑤ get the next query in *RSet*;
⑥ else
⑦ *BoundaryObjectsComputing*(r_m, *btq*, *U*); /* 计算边界对象参见第 5.1 节* /
⑧ if(δ_t-*DistortionDetection*(r_m, *btq*, *U*)) /* 判断 δ_t-质量模型参见第 5.2 节* /
⑨ *dis*= compute *SimDis*(r_m, *U*) from t_s to *maxT*; /* 计算相似距离* /
⑩ if(*mindis*> *dis*)
⑪ *mindis*= *dis*;
⑫ r_{min} = *r*;
⑬ endif
⑭ endif
⑮ endif
⑯ insert r_{min} into *U* ; /* 把查询 r 与具有最小相似距离的集合合并* /
⑰ repeat Step ③ to Step ⑫ until | *U* | doesn't change;
⑱ if(| *U* | ≥*k*)
⑲ δ_r privacy detection in Section 5.3; /* 判断 δ_p-隐私模型* /
⑳ else
㉑ insert *r* into *RSet*; /* 插入待匿名对象集合* /
㉒ endif
㉓ endfor

5.1 边界对象的检测

如上所述, 匿名框的边界对象随着对象的运动而变化. 虽然在一个候选匿名集中所有移动对象的状态(初始位置和运动速度)已知, 但是在一段时间内, 追踪所有对象的运动进而获得所有时刻的边界对象是不现实的, 代价很昂贵.

实际上, 计算边界对象没必要追踪每一个对象的运动, 只需要关注在时刻 t 的边界以及比边界对象运动速度快的对象. 图 7 给出了一维情况下在 x 轴上运动的例子. 从时刻 t_i 到时刻 t, 任意一个对象在 x 轴的位置可以通过式(3)确定:

$$x = x_{t_i} + v_x \times (t - t_i). \quad (3)$$

通过解线性方程组可以获得图 7 中不同移动对象相遇的时刻与位置(即交叉点). 并且, 不是所有的交叉点都需要计算, 如图 7 中交叉点 A 对边界没有影响, 可以忽略. 所以只需要关注那些比边界对象运动速度快的查询, 同时通过式(3)解线性方程组计算出边界更换的时间和位置, 并记录在队列 *BTQ* 中, 以帮助 δ_t-质量模型和 δ_p-隐私模型的判断. 队列 *BTQ* 中包含的每一个对象形式为<*time*, *query*, *boundary*>,其中 *time* 表示边界对象更换的时间, *query* 表示更换的边界对象标识, *boundary* 表示是候选匿名框上/下/左/右哪一个边界.

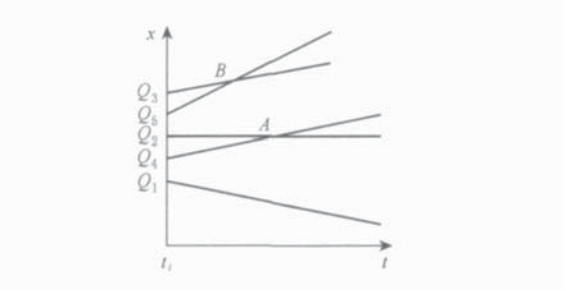


Fig. 7 Boundary objects detection.
图 7 边界对象检测

所以, 边界对象检测方法的主要思想是: 对于每一个候选匿名集, 针对 x 轴正方向, 寻找比当前边界速度大的对象, 根据式(3)解方程组计算边界更换时间; 针对 x 轴负方向寻找比当前边界速度小的对象, 计算边界更换时间. 根据对称性, y 轴方向上采用类似操作. 该算法较直观, 由于篇幅限制, 具体算法省略.

5.2 δ_t-质量模型检测

在第 5.1 节已计算出每一个候选匿名集的边界对象变更队列 *BTQ*, 结合该队列很容易获得候选匿名集边界对象及其位置. 对于 *BTQ* 中任意两个连续

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

时刻点 t_i, t_{i+1}, 设 $P_A = A_{height} + A_{width}$, $P_{L,t} = L_{x+,t} - L_{x-,t} + L_{y+,t} - L_{y-,t}$, $P_{v,t} = v_{x+,t} - v_{x-,t} + v_{y+,t} - v_{y-,t}$, 则根据 δ_t-质量模型的要求, 对任意时刻 $t \in [t_i, t_{i+1}]$,

$$\frac{1}{P_A}[P_{L,t_i} + P_{v,t_i}(t - t_i)] < \delta_t. \qquad (4)$$

计算不等式(4), 如果在$[t_i, t_{i+1}]$存在解, 则说明不满足 δ_t-质量模型. 具体见算法 2:

算法 2. δ_t-质量模型检测算法.

输入: time queue BTQ, queries set U, time t_s;

输出: false/true.

① $t_{i-1} = t_s$;
② while BTQ is not null/* 当BTQ不为空时* /
③ t_i = pop up the first time from BTQ;
/* 从BTQ中取出第1个元素赋值给t_i* /
④ $t = \frac{P_A \sigma_q - P_L}{P_V} + t_i$;
⑤ if $t \geqslant t_i$ and $t < t_{i-1}$
⑥ return false; /* 式(4)无解* /
⑦ endif
⑧ $t_{i-1} = t_i$;
⑨ endwhile
⑩ return true.

5.3 δ_p-隐私模型检测

与第 5.2 节类似, 结合边界对象变更队列 BTQ, 可以检测候选匿名集是否满足 δ_p-隐私模型. 主要思想是对于任一个候选匿名集, 取出其 BTQ 中两个连续时刻t_i, t_{i+1}, 根据式(1)和式(2)计算候选匿名框的宽和高, 分别判断是否大于 $\Delta_p = \delta_p \times \min(A_{width}, A_{height})$. 若在候选匿名集的生命有效期内的任意一个连续时间段$[t_i, t_{i+1}]$, 两个不等式均无解, 同时候选匿名集大小大于 K, 则该候选匿名集可作为匿名结果成功返回. 反之, 若有其中任何一个不满足, 则把触发查询 r 插入查询待匿名集合 $RSet$. 具体算法参见算法 3:

算法 3. δ_p-隐私模型检测算法.

输入: time queue BTQ, queries set U, time t_s; /* 输入时间队列BTQ、查询集合U、时间t* /

输出: false/true.

① $t_{i-1} = t_s$;
② while BTQ is not null/* 当BTQ不为空时* /
③ t_i = pop up the first time from BTQ
/* 设 $WB_t = L_{x+,t_i} - L_{x-,t_i}$, $HB_t = L_{y+,t_i} - L_{y-,t_i}$, $VWB = v_{x+,t} - v_{x-,t}$, $VHB_t = v_{y+,t} - v_{y-,t}$* /
④ $t_1 = \frac{\Delta_p - WB_{t_{i-1}}}{VWB_{t_{i-1}}} + t_{i-1}$; /* 判断从$t_{i-1}$到$t_i$是否每个时刻匿名框的宽均大于$\delta_p$* /
⑤ if $t_1 \geqslant t_{i-1}$ and $t_1 < t_i$
⑥ return false;
⑦ else
⑧ $t_2 = \frac{\Delta_p - HB_{t_{i-1}}}{VHB_{t_{i-1}}} + t_{i-1}$; /* 判断从$t_{i-1}$到$t_i$是否每个时刻匿名框的高均大于$\delta_p$* /
⑨ if $t_2 \geqslant t_{i-1}$ and $t_2 < t_i$
⑩ return false;
⑪ endif
⑫ endif
⑬ endwhile
⑭ return true.

每一个查询均有一个有效期, 如果该查询在有效期内没有匿名成功, 则从 $RSet$ 中去除该对象.

6 实验结果与分析

实验采用著名的 Thomas Brinkhoff[10] 路网数据生成器, 以城市 Oldenburg 的交通路网(周长大约6000 km)作为输入, 生成模拟数据. 算法采用 Java 实现, 在处理器 P4 2.0 GHz、内存 2 GB 的平台上运行. 算法中的各参数默认值如表 1 所示:

Table 1　Default System Settings

表 1　实验参数及默认取值

Parameters	Default Values
Number of Queries	10000
Privacy Level(k)	5
δ_p	0.1% of the space
δ_q	1% of the min ($width$, $height$) of the space
δ_t/s	100

实验评测了匿名成功率、匿名时间、处理时间和平均匿名代价随着隐私度(k)增加的变化情况. 隐私度 k 的增加代表用户的隐私需求更加严格, 要求更多的用户匿名在一起从而保护用户隐私. 匿名成功率是成功获得匿名查询占查询提出总数的百分比. 如图 8 所示, 随着隐私度 k 的增长成功率逐渐下降. 但是, 即使隐私度增加到 $k = 8$, 成功率依然保持在 90% 以上. 查询匿名时间指的是触发查询提出请求

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

到匿名成功的时间. 查询处理时间是指任何查询从提出到匿名成功的时间. 查询处理时间比匿名时间多了等待时间. 如图 9 所示, 无论是匿名时间还是处理时间, 均随着隐私度的增加而增长. 这是因为随着隐私度的增长, 每一个查询均需要更多的时间处理、等待才能匿名成功. 用每一个查询的匿名框的平均周长代表查询的平均匿名代价. 周长越长则查询处理代价越高. 如图 10 所示, 用户的匿名代价随着隐私度的增加而呈线性增长. 即随着隐私度的增加, 匿名框需要覆盖更多较远的对象从而满足隐私需求.

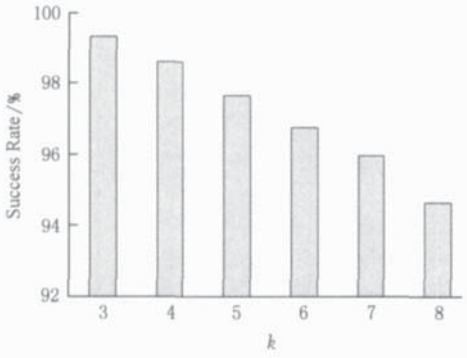


Fig. 8 Success rate.

图 8 匿名成功率

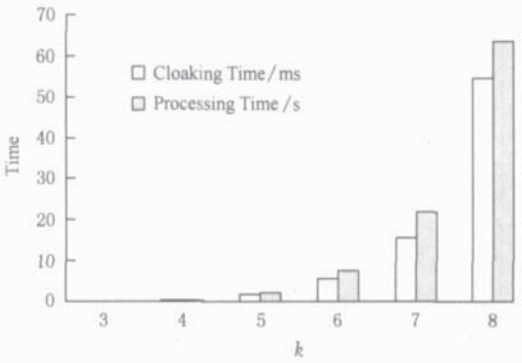


Fig. 9 Cloaking time and processing time.

图 9 匿名时间和处理时间

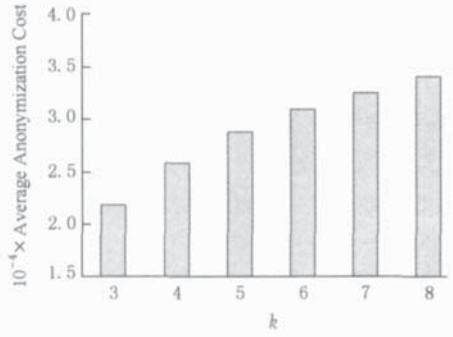


Fig. 10 Average anonymization cost.

图 10 平均匿名代价

7 结 论

本文研究了基于位置服务中连续查询的隐私保护问题. 阐述了现有的静态匿名算法不适用于连续查询隐私保护, 产生查询隐私泄露、匿名服务器工作代价大等问题. 并提出了 δ_t-隐私模型和 δ_q-质量模型可以有效地均衡隐私保护与服务质量, 在默认设置下, 基于该模型的贪心匿名算法可以在 2 ms 内匿名成功, 匿名成功率可达到 98%.

参 考 文 献

[1] Pan Xiao, Xiao Zhen, Meng Xiaofeng. Survey of location privacy-preserving [J]. Journal of Frontiers of Computer Science and Technology, 2007, 1(3): 268-281 (in Chinese) (潘晓, 肖珍, 孟小峰. 位置隐私研究综述[J]. 计算机科学与探索, 2007, 1(3): 268-281)

[2] Mokbel M F, Chow C Y, Aref W G. The new Casper: Query processing for location services without compromising privacy [C] // Proc of the 32nd Int Conf on Very Large Data Bases (VLDB). New York: ACM, 2006: 763-774

[3] Chow C, Mokbel M F. Enabling privacy continuous queries for revealed user locations [C] // LNCS 4605: Proc of the Int Symp on Advances in Spatial and Temporal Databases (SSTD). Berlin: Springer, 2007

[4] Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking [C] // Proc of the Int Conf on Mobile Systems, Applications, and Services (MobiSys). New York: ACM, 2003: 163-168

[5] Xiao Zhen, Xu Jianliang, Meng Xiaofeng. P-sensitivity: A semantic privacy-protection model for location-based services [C] // Proc of the 2nd Int Workshop on Privacy-Aware Location-Based Mobile Services (PALMS). Piscataway, NJ: IEEE, 2008: 47-54

[6] Bamba B, Liu L. Supporting anonymous location queries in mobile environments with privacy grid [C] // Proc of Int Conf on World Wide Web (WWW). New York: ACM, 2008: 237-246

[7] Kido H, Yanagisawa Y, Satoh T. Protection of location privacy using dummies for location-based services [C] // Proc of the 26th Int Conf on the Physics of Semiconductors (ICPS). Piscataway, NJ: IEEE, 2005: 1248-1248

[8] Ghinita G, Kalnis P, Khoshgozaran A, et al. Private queries in location based services: Anonymizers are not necessary [C] // Proc of ACM SIGMOD 2008. New York: ACM, 2008

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[9] Xu T, Cai Y. Location anonymity in continuous location based services [C] //Proc of Int Symp on Advances in Geographic Information Systems(GIS). New York: ACM, 2007

[10] Brinkhoff T. A framework for generating network-based moving objects [J]. An Int Journal on Advances of Computer Science for Geographic Information Systems (GeoInformatica), 2002, 6(2): 153-180

Pan Xiao, born in 1981. PhD candidate at Renmin University of China. Her main research interests focus on mobile data management.

潘　晓, 1981 年生, 博士研究生, 主要研究方向为移动数据管理.

Hao Xing, born in 1985. Master candidate at Renmin University of China. Her main research interest focuses on mobile data management.

郝　兴, 1985 年生, 硕士研究生, 主要研究方向为移动数据管理(haoxing@ruc.edu.cn).

Meng Xiaofeng, born in 1964. Professor and PhD supervisor, Secretary General of Dadabase Society of China Computer Federation. His main research interests include Web data integration, XML database, and mobile data management.

孟小峰, 1964 年生, 教授, 博士生导师, 中国计算机学会数据库专委会秘书长, 主要研究方向为 Web 数据管理、XML 数据库、移动数据管理(xfmeng@ruc.edu.cn).

Research Background

With the blooming of sensor and wireless mobile devices, it is easy to access mobile users' location anytime and anywhere. On one hand, location based services are more and more valuable and important. On the other hand, privacy issues, including location privacy and query privacy, in location based services raised by such applications, have attracted more and more attention. In this paper, we consider query privacy preserving for snapshot and continuous queries. To address this issue, we firstly propose δ_p-privacy model and δ_q-distortion model to balance the tradeoff between privacy preserving and quality of services, and use the perimeter of cloaking region to measure the distortion of the location information. Then, the location distortion is mapped to the temporal similar distance between two queries. Finally, a greedy algorithm is proposed to find cloaking set for snapshot and continuous queries. Average cloaking success rate, cloaking time, processing time and anonymization cost for successful requests are evaluated with increasing privacy level. Experimental results validate the efficiency and effectiveness of our proposed algorithm. This research is partially supported by the National Natural Science Foundation of China(Nos. 60833005, 60573091), the National 863 High-Tech Research and Development Plan of China (Nos. 2007AA01Z155, 2009AA011904).

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2010/04(12)-1057-16
DOI: 10.3778/j.issn.1673-9418.2010.12.001

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-51616056

普适计算中复合事件检测的研究与挑战*

周春姐+, 孟小峰

中国人民大学 信息学院, 北京 100872

The Researches and Challenges of Complex Event Detection in Pervasive Computing*

ZHOU Chunjie+, MENG Xiaofeng

School of Information, Renmin University of China, Beijing 100872, China

+ Corresponding author: E-mail: lucyzcj@ruc.edu.cn

ZHOU Chunjie, MENG Xiaofeng. The researches and challenges of complex event detection in pervasive computing. Journal of Frontiers of Computer Science and Technology, 2010, 4(12): 1057-1072.

Abstract: In pervasive computing environments, wide deployment of sensor devices has generated an unprecedented volume of atomic events. However, most applications such as healthcare, surveillance and facility management, as well as environmental monitoring require such events to be filtered and correlated for complex pattern detection. Therefore how to extract interesting, useful and complex events from low-level atomic events is becoming more and more important in daily life. At present, there are a lot of researches of complex event detection, and each has its own particular research points. Some pay attention to the time information, especially the importance of time interval; some research in the complex event detection in distributed data sources; recently some propose the probabilistic data management on complex event detection. Due to the increasingly importance of complex event detection, this paper analyzes the challenges in the research of complex event detection, and gives a survey of existing researches from three aspects including event types, time information, and precision of data. Finally, some open issues and future researches are given.

Key words: pervasive computing; sensor; complex event detection; time interval; probabilistic data

---

*The National Natural Science Foundation of China under Grant No.60833005, 60573091(国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z155 (国家高技术研究发展计划(863)); the Doctoral Fund of Ministry of Education of China under Grant No.200800020002 (国家教育部博士点基金).

Received 2010-04, Accepted 2010-06.

摘 要: 普适计算环境中, 传感器设备的大规模使用产生了数量巨大、错综复杂的原子事件, 而现实中的许多应用却更注重复合事件的检测, 例如：健康护理、监督设施管理、环境/安全监控等, 因此如何从这些底层的原子事件中抽取人们感兴趣的、有用的复合事件就变得越来越重要。目前, 针对复合事件检测有大量的研究, 其内容各有侧重。有的重视时间因素, 特别强调时间段的重要性; 有的研究分布式数据源中的复合事件检测; 近期有人提出了不确定性数据上的复合事件检测。由于复合事件检测日益重要, 对复合事件检测研究中存在的挑战性问题进行了分析, 从事件类型、时间因素和数据的精确程度 3 个方面归纳总结了复合事件检测现有的研究成果, 并指出了未来的发展方向。

关键词: 普适计算; 传感器; 复合事件检测; 时间段; 不确定性数据

文献标识码：A 中图分类号：TP391

1 引言

随着计算机、通信、网络、微电子、集成电路等技术的发展, 信息技术的硬件环境和软件环境发生了巨大变化。这种变化使得通信和计算机构成的信息空间, 与人们生活和工作的物理空间正在逐渐融为一体。普适计算(pervasive computing)的思想就是在这种背景下产生的。普适计算环境的特点是以人为本, 为用户提供更高效精确的、无处不在的人性化服务, 即系统可以根据用户的爱好、需求对服务进行自由的裁剪和定制。为了达到上述目标, 在追踪和监控等实际应用中, 正在大规模地使用无线感知网络和无线射频技术(radio frequency identification, RFID)等传感器设备。这些设备的广泛部署产生了大量的、直接反映物理世界的原子事件(定义见第 2 章)。一个标准的传感器设备通常有成千上万条数据记录, 这使得操作人员通过观测每一条记录来发现异常事件变得非常困难; 此外, 在事后分析异常事件时, 需要操作人员及时找出相关记录。然而传统的检测方法缺乏智能分析, 数据无法被有效地检索, 只能根据大致的时间段来人工查找, 导致数据分析工作消耗了大量的工作时间和精力。

解决上述问题的一个有效方法是对事件进行自动智能分析, 对数据集中出现的用户感兴趣的事件进行实时提取和记录, 从而达到及时报警并利用存储的事件信息来有效地检索数据。例如, 在零售管理系统中, 通过将一系列原子事件整合成复合事件(定义见第 2 章), 可以及时地发现偷窃行为。当发生这样一种场景时：商品被从货架上取下→没有结账→带出商店, 系统就会自动发出预警信号。

复合事件检测在现实世界中有许多应用, 例如：健康护理[1−2]、监督设施管理[3]、环境监控[4]、供应链管理[5]以及各种普适计算应用[6−8], 都需要将大量原子事件过滤成相互关联的复合模式检测, 或者转化成可以直接服务于终端应用的、富含语义的新事件。其中, 室内环境下的应用如：

(1) 健康护理中需要系统实时地从大量的传感器数据中推断出被看护者的行为, 看护者可能想获知一系列的行为流[1−2,9], 如：被看护者是否按时吃药, 是否按时吃饭, 在睡觉之前是否刷牙, 以及状态是否正常等, 从而判断出被看护者是否已经被很好地照顾;

(2) 通过对大量原子事件的分析处理, 抽取出用户感兴趣的、有用的复合事件, 长期的复合事件的存储管理, 可以总结分析出用户的行为模式, 从而可以在默认的情况下, 自动为用户提供各种服务;

(3) 安全系统可以通过分析不同日期的、相同时刻的、与用户相关的复合事件, 来决定是否

需要实施某些预警信息。室外环境下的应用如：在机场、车站、港口、建筑物周围，以及街道、小区等场所，可用于检测、分类、跟踪和记录过往行人、车辆及其他可疑物体，或用于判断是否有行人及车辆在禁区内发生长时间徘徊、停留、逆行、奔跑、打斗等异常行为。

由上可见，这些应用需求都需要根据用户的指定，或者根据某些规则，自动地从这些底层的、原子性的事件中抽取一些复合事件，从而将反映物理世界的原始数据实时地转化成可直接服务于终端应用的信息。因此，复合事件检测的研究意义重大，它将会成为未来普适计算环境中非常重要的一部分，并将渗入到未来生活的各个方面。

本文结构如下：第 2 章将用例子说明什么是复合事件检测，以及其研究的必要性；第 3 章介绍在事件检测研究中所面临的挑战；第 4 章归纳总结了事件检测的研究现状；第 5 章说明有待进一步解决的问题；最后是总结。

2 复合事件检测的基本概念

符合普适计算“以人为本”的特点，复合事件检测的目标就是把从物理空间得到的原子事件转化成用户感兴趣的，可以直接服务于用户的复合事件。本章将给出复合事件检测中的一些基本概念。

事件被定义为用户感兴趣的行为。例如：在房间中发现某个人、启动 CPU 定时器、在网络中拒绝某个攻击服务等都是不同应用领域中的事件的例子。所有的事件都表示特定的行为，然而他们的复杂性却存在很多差异。例如：启动定时器是一个瞬间的、简单的发现，而拒绝攻击服务则需要计算多个简单事件。因此，事件可被划分为原子事件和复合事件[10]。下面分别给出原子事件和复合事件的详细定义。

定义 **1**(原子事件) 原子事件是在某个时间点上瞬间发生的、原子性的事件。可表示为 $E_{atomic_i}=Action<o_i,p_i,t_i>$，其中 o_i 表示某个对象(object)，它可以是某个人，也可以是某个物体；p_i 表示地点或位置(place)，即对象 o_i 的当前位置；t_i 表示某时间点；Action 表示 t_i 时刻，对象 o_i 在 p_i 位置的行为，即是一个原子事件 E_{atomic_i}。例如：Coffee(‘Mary’, ‘Room 301’, 10:00 am)表示上午 10 点这一刻，Mary 在 301 房间喝咖啡。

定义 **2**(复合事件) 复合事件通常是由用户指定的、或者自动地从原子事件中抽取出来的，某个时间段上发生的事件。可表示为 $E_{complex_i}=<Q_i,E_i,T_i>$，其中 Q_i 表示某个查询(query)，它只在用户指定的情况下是有效的，用来表示用户的查询条件，而当自动抽取时其值为空；E_i 表示一系列原子事件的集合，即 $E_i=\{1\leqslant i\leqslant n\,|\,E_{atomic_i}\}$，集合中的各个原子事件之间是相互关联的，它们之间存在某种集合运算关系(如正相关、负相关、并行、串行等)；T_i 表示某个时间段，在时间段 T_i 中，查询/抽取出一系列相互关联的原子事件就组成一个复合事件。例如：当检测到原子事件“Mary is in her office”, “Mary is in coffee room”, “Mary is in her office again”时，就可以从中抽取出复合事件“Mary is getting coffee”。复合事件是由原子事件或其他复合事件通过一系列事件组合运算得到的。

复合事件检测的处理过程如图 1 所示。

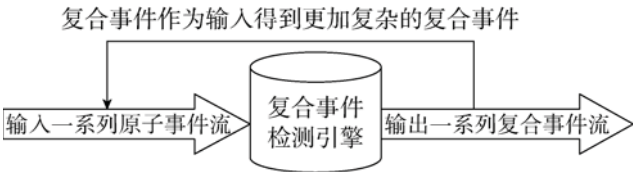


Fig.1 The principle of complex event detection

图 1 复合事件检测原理图

将一系列的原子事件流作为输入，经过复合事件检测引擎的分析处理，输出结果就是一系列复合事件。考虑到特定情况下用户可能需要更为复杂的复合事件，因此复合事件检测引擎应该支

持将输出的复合事件再次作为输入，从而得到更加复杂的复合事件。复合事件检测的方法有两种：(1) 根据用户指定。即用户提出某个查询规则，复合事件检测引擎根据此查询规则，从输入的一系列原子事件中检测出满足条件的一系列复合事件; (2) 自动检测。即复合事件检测引擎根据用户的生活习性和历史记录，自动地从一系列输入的原子事件中检测出用户感兴趣的，能够为用户提供直接服务的复合事件。

在普适计算环境中，原子事件的数量是巨大的，构成复合事件的不同原子事件之间的关系是复杂的，大量的事件分布在不同节点上，产生的事件具有分布、并发、异步、随机、不确定等特点。并且，事件源可能是传感器、无线设备、移动设备等，事件的到达因网络连接的不稳定以及移动设备的移动而发生时延、失序的情况。因此理想的复合事件检测方法应该能够高效地进行分布式检测、并发异步事件的检测、失序事件的检测以及断接检测、移动检测。在普适计算环境中，事件分布在地理位置分散的节点中，其中有些节点是移动节点。若用一个中心节点来探测基本事件，形成最后的复合事件表达式，那么这个节点将成为事件探测的瓶颈。因此，要高效地检测事件，就要根据普适计算环境的特点和系统要求选择合适的检测方法。目前存在的复合事件检测方法有：(1) 基于事件树的复合事件检测[11]; (2) 基于图的检测方法[12]; (3) 基于自动机的复合事件检测[13]; (4) 基于 Petri 网的复合事件检测[14]; (5) 流水线操作的检测方法[15]。上述每种复合事件检测方法都各有利弊：GEM[11]考虑了事件发生与检测之间的延迟，并通过指定最大能容忍的延迟来处理事件检测的失序。但它假定了存在一个良好的全局同步时钟，这不适合没有集中管理以及存在时钟漂移的大规模松耦合的分布式系统。由于没有考虑不可预知的延迟，不能有效进行移动数据库中的断接检测和移动检测；Snoop[12]只提供简单的时间模型，把事件看作一个确定的时间点，原子事件根据定义来确定时间点，而复合事件的时间则根据其语义来定义时间点，这比较适合应用于集中式系统或局域网；ODE[13]数据库中使用的有限状态自动机表示事件，能直观地表达现实中的事件，建立自动机并据此检测复合事件。但是纯粹的自动机不检测带参数的事件，不能表示事件的时序关系，不能检测并发事件，这不符合分布式系统的需求。

由于上述的各种复合事件检测方法都没有考虑不确定性的问题，而不确定性又是普适计算环境中的本质特征。因此对现有研究工作的归纳总结不是按照不同的检测方法来分类，而是从复合事件检测的 3 个特征(时间因素、复合事件表示方法、数据的精确程度)入手，分别说明和归纳(见第 4 章)。

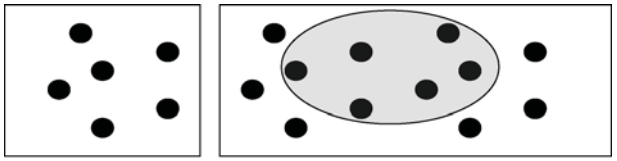
3 复合事件检测研究面临的挑战

复合事件检测的应用领域广泛，研究意义重大，但是在普适计算环境下，数目巨大的、并发的、无序的原子事件，以及大量不确定性数据的产生给复合事件检测的研究工作带来了很多挑战。其中包括：

(1) 大量的事件流。由于传感器等设备的大量部署，每秒钟都会有成千上万的事件产生。例如，在零售管理系统中，商品的入库和出库、上架和下架、购买、结账、带出商店等都会产生大量的事件流。复合事件检测必须能妥善处理这些规模巨大的事件流，准确分析各事件之间的相互关联和影响。

(2) 时间窗口大小的选定。复合事件检测中通常使用一个滑动时间窗口来保存一系列感兴趣的事件。在许多应用场景中，这些窗口是很大的，并且与某个查询相关的事件在窗口中是分散分布的。不像原子事件检测那么简单，复合事件检测必须抽取所有相关的事件，返回满足某个查

询的所有可能的结果，这就使数据处理的复杂性很大。例如，图 2(a)表示时间窗口为过去 6 小时的事件分布情况；图 2(b)表示时间窗口为过去 12 小时的事件分布情况。与 2(b)相比，2(a)中包含的事件数目少，检测的复杂度也小，但是由于它选取的时间段短，因此会出现检测不全的现象，无法返回所有可能的结果。例如图中椭圆阴影的检测结果在时间窗口图 2(a)中就无法得到。图 2(b)中虽然能检测全面些，但是包含的事件多，从而检测的复杂度很大。可见，事件数目和检测全面性是两个矛盾体，如何权衡两者，选定合理的时间窗口大小是一个挑战性的问题。



(a) T=6 h
(a) T=6小时
(b) T=12 h
(b) T=12小时

Fig.2 The choice of time window
图 **2** 时间窗口的选择

(3) 时间同步问题。事件发生的时刻及事件之间发生的先后关系，表明了大量的基本事件是如何构成相关的复合事件的，因此基于时序的事件检测要求各节点的时间同步。而普适计算环境中，没有集中统一的管理。节点之间松散耦合，各节点的时钟偏频和漂移会造成节点间的时间不同步，节点移动与不稳定连接使事件发生到事件检测之间产生不可预测的延迟，这些情况会造成事件检测失序。

(4) 事件的不确定性。复合事件检测中存在两类不确定性，一种是事件的局部不确定性，另一种是事件的全局不确定性。当进行事件检测时只考虑元组/对象自身的不确定性，认为它们独立于其他的对象/元组，称之为事件的局部不确定性；例如：第 2 章所提到的喝咖啡的例子中，Mary 想什么时间喝咖啡，到哪个咖啡屋喝咖啡，以及喝咖啡持续的时间等都是不确定的。如表 1(a)所示，Mary 喝咖啡的时间可能是 10:15 am 也可能是 9:55 am，可能在 1 号咖啡屋也可能在 2 号咖啡屋，持续时间可能是 15 min 也可能是 17 min。但是这些因素只由 Mary 自身的意愿决定，与其他人无关，因此称为事件的局部不确定性。当需要考虑事件之间的关联性和相互影响时，称之为事件的全局不确定性。还以 Mary 喝咖啡为例，假设 Mary 喜欢在多数人喝咖啡的时候，和 Joe 一起去喝咖啡，这时 Mary 喝咖啡的时间、地点、持续时间等不仅仅由 Mary 自身的意愿决定，而且受周围其他人的情况以及 Joe 的意愿等不确定性因素共同决定。如表 1(b)所示，7 月份由于受到 Joe 的影响，

Table 1(a) The event local probability
表 **1(a)** 事件的局部不确定性

日期	姓名	开始喝咖啡的时间	咖啡屋的编号	喝咖啡持续时间/min
3 月 1 日	Mary	10:15 am	1	15
3 月 2 日	Mary	9:55 am	2	17
3 月 1 日	Joe	8:55 am	1	10
3 月 2 日	Joe	9:05 am	1	8

Table 1(b) The event global probability
表 **1(b)** 事件的全局不确定性

日期	姓名	开始喝咖啡的时间	咖啡屋的编号	喝咖啡持续时间/min
7 月 1 日	Mary	9:35 am	1	12
7 月 2 日	Mary	9:25 am	1	10
7 月 1 日	Joe	9:35 am	1	12
7 月 2 日	Joe	9:25 am	1	10

Mary 喝咖啡的时间提前了，持续时间也比 3 月份缩短了，但是他们不同日期的喝咖啡时间和持续时间等仍然是不确定的。在许多重要应用中，不确定性已经成为本质特征，因此在进行复合事件检测时必须全面考虑事件的不确定性。

4 复合事件检测的研究现状

如第 2 章所提到的，已有的基于事件树的复合事件检测、基于图的检测、基于自动机的复合事件检测等都不能很好地满足普适计算环境的需求。因此，针对普适计算环境的特点，将主要从复合事件检测的 3 个特征入手对目前的研究情况进行归纳总结。目前对复合事件检测的研究一般包含以下几个方面：从事件类型来考虑，描述了复合事件的表示方法；从时间角度来考虑，描述了各种时序表示方法；从数据的精确程度来考虑，对不确定性数据进行分析处理。现有研究工作对这 3 个方面的考虑各有侧重，总结如下。

4.1 现有研究的分类

根据复合事件检测的上述 3 个特性，对现有的研究研究工作进行分类总结如图 3 所示。其中，3 个坐标轴分别对应 3 个特性：时间(时间点和时间段)、数据(精确的和不确定的)、事件(原子的和复合的)。3 个坐标轴将空间分为 8 个象限，如图

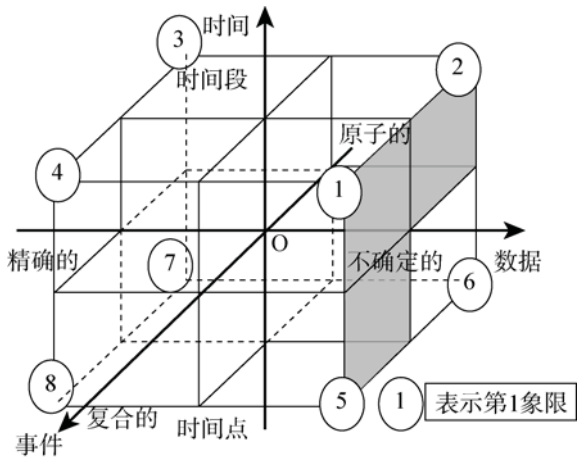


Fig.3 The existing researches of complex event detection

图 3 复合事件检测的现有研究总结

中的标注，每个象限都对应 3 个特性的不同属性值。

如图 3，第 7 象限的立方体区域表示在时间点上针对确定性数据的原子事件的研究，目前这方面的研究工作已经有很多[16]；第 3、4、6、8 象限的立方体区域包括时间段上的精确数据的原子事件，时间段上的精确数据的复合事件，时间点上的不确定性数据的原子事件，时间点上的精确数据的复合事件，目前已经有一些这方面的研究工作[10,17–22]；第 1、2、5 象限的立方体区域主要是针对不确定数据方面的，包括时间段上的不确定性数据的原子事件，时间段上的不确定性数据的复合事件和时间点上的不确定性数据的复合等，目前还没有相关的研究工作。其中，如表 2 所示，第 3、4、6、8 象限的立方体区域中有关于时间点、原子事件和确定性数据的[16]；有关于时间点、复合事件和确定性数据的[10,17]；有关于时间段、原子事件和确定性数据的[18–20]；有关于时间段、复合事件和确定性数据的[21]；有关于不确定性数据的[22]。

Table 2 The comparison of existing researches

表 2 现有研究工作的分析比较

相关研究工作	时间段	复合事件	不确定性数据
文献[16]	No	No	No
文献[10, 17]	No	Yes	No
文献[18–20]	Yes	No	No
文献[21]	Yes	Yes	No
文献[22]	No	No	Yes

4.2 复合事件表示方法

如第 2 章所提到的，事件可被划分为原子事件和复合事件，复合事件是从一系列原子事件中抽取得到的。因此，提出了一种最简单的复合事件表示方法[22]。例如，“Mary 喝咖啡”这个复合事件可被表示成 3 个连续的原子事件：(1) “Mary 在她的办公室里”；(2) “Mary 在咖啡屋”；(3) “Mary 又回到她的办公室里”。如果假设 Mary 的

办公室在220房间，咖啡屋位置不确定，则这个复合事件可被表示为：

$q_{MaryCoffee}=At(\text{‘}Mary\text{’}, \text{‘220’})$

$(\sigma_{CRoom(l)}\, At(\text{‘}Mary\text{’}, l)); At(\text{‘}Mary\text{’}, \text{‘220’})$

这种方法存在一个强假设，就是事件是相互独立的，事件与事件之间不会相互影响。但这与实际情况是很不符合的，比如上面的例子中，如果还存在一个事件“咖啡屋里没有咖啡了”，这时按照上述方法(不考虑事件之间的相互关联和影响)就会得出错误的结论，因为实际上 Mary 是没有喝到咖啡的。

文献[17]提出了一种事件描述语言，它可以用来实现事件的过滤、关联和相互转换。下面以在第 1 章中所提到的零售管理为例，来详细介绍这种事件描述语言。在零售管理中，处理商品的存放错误通常会耗费大量的时间和人力。此事件描述语言与RFID技术相结合，就能提供一种方法来自动处理这一过程，从而减少大量的人力，并能加快货架的补充。处理商品误放的过程可以表示如下：

Event SEQ(shelf-reading x, shelf-reading y, ! (any(counter-reading, shelf-reading) z))

Where $[id] \wedge x.shelf\_id \neq y.shelf\_id \wedge x.shelf\_id = z.shelf\_id$

Within 1 hour

其中，SEQ 操作保证用户感兴趣的事件以特定的顺序发生。上述过程认为“在货架 1 上读到一个条款，紧接着在货架 2 上又读到相同的条款，并且此条款没有在结账口读到，也没有被重新放回到货架 1 上”是商品误放的情况，但是“在货架 1 上读到一个条款，在结账口又读到相同的条款”不属于商品误放的情况。where 中用已经定义好的变量来比较不同事件之间的属性，上述过程比较的是满足 SEQ 的 3 个事件的 *id* 属性。“$x.shelf\_id \neq y.shelf\_id$”保证了 shelf-reading 的前两项指向不同的货架，而“$x.shelf\_id = z.shelf\_id$”保证了如果 any 操作返回一个 shelf-reading，那么所读取的条款不是来自货架 1 的。within 指定一个时间段，例如上述过程中的1小时，用户感兴趣的事件必须在此时间段内发生。但是这种方法也存在几个限制条件，他假设所有的事件都是完全有序的，不考虑事件的并发性；并且，他只考虑了直接由原子事件组合得到的复合事件，而没有考虑由复合事件整合成的更为复杂的事件。

由以上分析可见，已有的处理复合事件的方法都存在一定的缺陷，假设条件太强，考虑不太全面，因此一种新的、全面的、高效的复合事件检测方法亟待提出。

4.3 时序关系表示方法

每个事件都对应一个时间段，用来表示其发生周期。对原子事件而言，其时间段就是一个点，开始和结束时间重合；对复合事件而言，其对应的时间段包含所有子事件的时间段[23]。现有的研究工作大多只考虑了时间点，都假设事件是没有持续时间的[24–27]。这种假设通常将事件简化成一个有序序列，例如：“头痛->胃痛->呕吐”。然而现实世界中的很多事件都是有持续时间的，并且这些事件之间的时序关系也是很复杂的[18–20]，所以上述这种时序模式不足以表达复杂的时序关系。在医疗、多媒体、气象学和财政学等领域，事件的持续时间都起了很重要的作用。例如：许多糖尿病患者的症状都是血糖的增高和尿糖的缺失同时并存，只有准确地表示这两者的重叠性才能很好地诊断；又如，骨热病的一般症状是在发热后的第三天血小板就开始减少，只有能够很好地表示这种间隔性和时序性，才能准确地把握治疗时间。

为了有效地抽取基于时间段的复合事件，就需要一种独特的、无损耗的表示方法来获取事件之间的时态关系。传统的方法是用 Allen 的区间代数[28]来表示基于时间段的两个事件之间的时态关系，如表 3 所示。然而，当要获取 3 个或更多事件之间的时态关系时，这种方法就失效了。

Table 3 The temporal relationship between E_i and E_j

表 3 事件 E_i 和 E_j 之间的时态关系

Relation	Interval algebra	Dual relation
E_i Before E_j	(E_i.end < E_j.start)	After
E_i Meet E_j	(E_i.end = E_j.start)	Met-by
E_i Overlap E_j	(E_i.end > E_j.start)∧(E_i.end < E_j.end) ∧ (E_i.start < E_j.start)	Overlapped-by
E_i Start E_j	(E_i.start = E_j.start) ∧ (E_i.end < E_j.end)	Started-by
E_i Finished-by E_j	(E_i.end = E_j.end) ∧ (E_i.start < E_j.start)	Finish
E_i Contain E_j	(E_i.start < E_j.start) ∧ (E_i.end > E_j.end)	During
E_i Equal E_j	(E_i.start = E_j.start) ∧ (E_i.end = E_j.end)	Equal

许多研究者试图运用分层的方法来表示事件之间的时态关系[29−30]，但是这种表示法是有损耗的，因为它不能保持事件潜在的瞬时结构。任何有损耗的表示法都会导致许多伪造的模式，例如非频繁模式可能会成为频繁模式。

现有的基于时间段的很多算法或者是有损耗的表示法[18]，或者没有很好的扩展性[19−20]。如表 3 中对事件的时态表示就存在歧义性，给定表示法(A Overlap B) Overlap C，无法推断 C 仅仅与 B 重叠，或者 C 同时与 A、B 重叠。图 4 给出了这种时态模式的不同解释，这些不同的解释就会导致对具有确切关系的事件的错误推断。为了解决这个问题，提出了[21]一种无损耗的表示法，它是在分层表示法的基础上引入了一些附加信息。即用 5 个变量(包含数目 c、结束数目 f、相交数目 m、重叠数目 o、开始数目 s)来区分所有可能的情况。因此，复合事件 E 可表示为：

$$E=(E_1\ R_1[c,f,m,o,s]\ E_2)\ R_2[c,f,m,o,s]\ E_3)\ldots R_{n-1}[c,f,m,o,s]\ E_n)$$

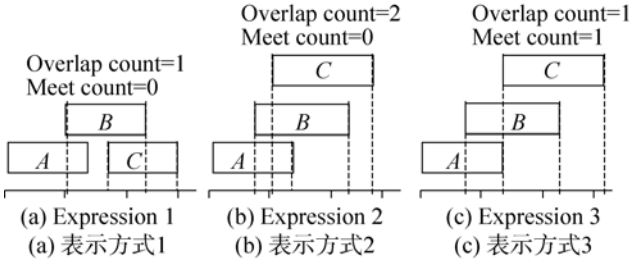


Fig.4 Different expressions of pattern (A Overlap B) Overlap C

图 4 模式(***A* Overlap *B*) Overlap *C*** 的不同解释

图 4 中的时态模式可分别表示如下：

(A Overlap[0,0,0,1,0] B) Overlap[0,0,0,1,0] C

(A Overlap[0,0,0,1,0] B) Overlap[0,0,0,2,0] C

(A Overlap[0,0,0,1,0] B) Overlap[0,0,1,1,0] C

目前对复合事件检测的研究中大多引入时间窗口的概念。时间窗口表示事件存储在事件序列中的时间，超过这个时间，此事件的记录则会被抛弃。它表示一个时间范围，有 3 个参数，即窗口的起始时间 TWB、终止时间 TWE 和窗口的大小 TWL，其中，TWE=TWB+TWL。时间窗口的大小可由系统默认或程序员来指定。前者是静态不变的，不能随着网络传输情况的变化而改变；后者可以通过一定的算法动态设置窗口。窗口设置过大，会在事件队列中存储过多失效事件；设置过小，会使许多复合事件检测不到。因此，复合事件检测中时间方面的研究还需要更加的深入细致。

4.4 数据的精确程度

复合事件检测中的一个重要挑战就是数据的不确定性(例如：RFID 数据)。产生数据不确定性的原因很多，例如：(1) 数据错误或丢失，这主要是由电源波动和噪声等因素造成的。研究表明，在现实应用中，RFID 的读取比率仅为 60%～70%，也就是说至少有 30%的数据被丢失[31−32]；(2) 数据矛盾，例如两个 Sensor 读取的 Mary 的位置不同，该如何确定 Mary 的真实位置[31]；(3) 粒度不匹配，例如选取的时间粒度是 ms，而某个事件查

询是“Mary 在 2008 年 5 月的前 3 天在做什么？”这就会导致(3×24×60×60×1 000)不确定的可能性。然而，现有的复合事件检测系统都假设数据是确定的，例如：Cayuga[3]、SASE[15]，和 SnoopIB[33]，这些系统都无法检测不确定性事件流。

处理不确定性数据的常用方法是建立一个数据模型，并且将原始数据作为模型的输入数据。其中一个标准的方法就是用时态图模型，而最简单的时态图模型就是隐式马尔可夫模型(hidden Markov model, HMM)。通常，HMM 通过一系列观察值来推断隐含的状态信息。例如，基于传感器数据来推断一个人的位置。在实时应用中，通常运用一种更加复杂的技术——滤除(smoothing)[34]。滤除技术不仅可以推断一个人的更加精确的位置，而且可以提供不同时刻此人的位置之间的关系。例如，如果知道 Mary 在 t=7 时刻进了办公室，那么很有可能在 t=8 时刻她仍然在办公室。也就是说，Mary 在 t 时刻的位置和 $t+1$ 时刻的位置是相关的。

关于不确定性的、相关联的隐式马尔可夫模型已经有很多的研究[35]，但是这些研究都没有考虑不同时间之间的关联性。为了更好地满足现实应用，提出了在相关联的、不确定性数据流上的复合事件查询方法[22]。

假设 $A_1, A_2, ..., A_k$ 是 k 个属性值，其中 A_i 在区域 D_i 范围内取值。令 $\overline{D} = D_1 \times D_2 \times \cdots \times D_k$，并且 $\overline{D}_\perp = \overline{D} \cup \{\perp\}$。$A_1, A_2, ..., A_k$ 上的部分随机变量是一个函数 $P: D_\perp \to [0,1]$，并且 $\sum_{d \in D_\perp} p(d) = 1$。例如，假设 e 为事件“Mary 上午 10 点在 326 房间喝咖啡的可能性是 0.2，在房间 327 喝咖啡的可能性是 0.7。”那么该属性值的随机变量可表示为：

P [e = ‘Room326’] = 0.2

P [e = ‘Room327’] = 0.7

同时，可以注意到，事件“Mary 上午 9 点 50 在 326 房间喝咖啡”与事件“Mary 上午 9 点 55 在 326 房间喝咖啡”是正相关的，而与事件“Mary 上午 9 点 50 在 327 房间喝咖啡”是负相关的。

假设 $\overline{e} = (e^{(1)}, e^{(2)}, ..., e^{(t)}, ...)$ 是一系列不确定性事件流，那么如果满足 $P[e^{(t+1)} \mid e^{(1)}, e^{(2)}, ..., e^{(t)}] = P[e^{(t+1)} \mid e^{(t)}]$，则 $\overline{e}$ 就是马尔可夫链。从而序列 $\overline{d} = (d^{(1)}, d^{(2)}, ..., d^{(t)})$ 的不确定性值就可以用贝叶斯规则定义为：

$$\mu(\overline{d}) \overset{def}{=} P[e^{(1)} = d^{(1)}] \cdot \prod_{i=2,\cdots t} P[e^{(i)} = d^{(i)} \mid e^{(i-1)} = d^{(i-1)}]$$

目前在复合事件检测中考虑不确定性的研究还很少，然而在许多重要的应用中都涉及到数据的不确定性，因此对不确定的复合事件检测进行研究分析已势在必行。

4.5 复合事件检测方法

基于以上的介绍，对普适计算中复合事件检测的 3 个特征已经有了比较深入的理解。下面针对普适计算中复合事件检测的两类主流方法(见第 2 章，分别介绍一个典型的研究工作。

4.5.1 根据用户的指定实现复合事件检测

已有的事件检测实现模型都是基于某种特定的数据结构，如基于事件树的、基于有向图的、基于自动机的、基于 Petri 网的等。在这些模型中，查询执行必须严格符合该特定数据结构的内在模式，而不能使用其他类似的方法来实现查询。并且，这些模型的扩展性也很差，它们无法支持更为丰富的查询语言，从而无法满足很多重要应用的需求。为了克服以上不足，提出了一种复合事件检测方法[17]，其关键的数据结构就是一个带有查询的事件序列，而没有严格的限制，其扩展性也有所改善。以下面的查询为例来详细介绍。

Event SEQ (A $x1$, B $x2$, ! (C $x3$), D $x4$)

Where [$attr1$,$attr2$]∧$x1$.$attr3$=‘1’∧$x1$.$attr4$<$x4$.$attr4$

Within T

在这个查询中，A、B、C、D 表示 4 种不同的事件类型，SEQ 操作保证这些事件以特定的顺序

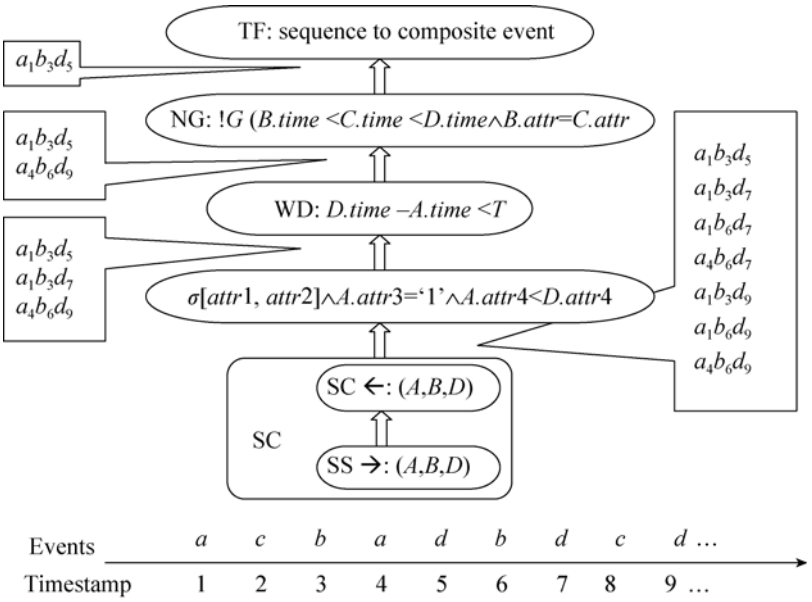


Fig.5 The process of complex event detection with query

图 5 带查询的复合事件检测过程图

发生, 即 *A* 先发生, 其次 *B* 发生, 最后 *D* 发生, 并且 *B* 与 *D* 之间不允许有 *C* 发生; *attr*1、*attr*2 是 *A*、*B*、*C*、*D* 的公共属性, *attr*3 是事件 *A* 的属性, 事件 *A* 在属性 *attr*4 上的值小于事件 *D* 在属性 *attr*4 上的值; *T* 表示一个特定的时间窗口大小。这种带查询的复合事件检测的实现过程如图 5 所示。

图5 中最下面的事件流, 其小写字母(如 *a*)表示的事件类型就是对应的大写字母值(如*A*)。每个事件下面的数字表示该事件发生的时间。

SSC(sequence scan and construction, 序列扫描和构造)是由两个操作组成的：(1) SS(序列扫描), 即扫描事件流, 从而发现与之匹配的子序列类型。以上述查询为例, SSC 就将“!*C*”从 SEQ(*A*, *B*,!*C*,*D*) 移除, 从而得到子序列类型 (*A*,*B*,*D*); (2) SC(序列构造), 即反向查询, 从而构造所有的事件序列, 将事件流转换成事件序列流。其中每个事件序列都唯一对应一种子序列类型。如图 5, SSC 的输出就是从底层事件流中产生的 7 个事件序列, 每个事件序列都对应子序列类型(*A*,*B*,*D*)的 3 个元组, 其中小写字母表示事件类型, 下标表示发生的时间。

σ (selection, 选择)根据查询条件对每个事件序列进行过滤, 除掉不满足条件的所有事件序列, 如图 5, 7 个事件序列中只有 3 个通过了选择。

WD(window, 时间窗口)检查每个事件序列中第一个事件和最后一个事件之间的时间差是否小于所设定的时间窗口大小 *T*。图 5 中, *T* 被设定为 6 小时, 因此, 第二个事件序列又被过滤掉。

NG(negation, 否认)处理被 SSC 忽略的 SEQ 中的否定元组。如图 5, 对每个输入的事件序列检查在事件 *b* 与 *d* 之间是否有 *c* 发生。如果存在这样的事件 *c*, 那么该事件序列就被删除, 图 5 中 NG 的第二个输入序列又被过滤掉了。

TF(transformation, 转换)将所得到的事件序列转换成复合事件。

尽管这种带查询的复合事件检测方法存在诸多优点, 但它本身还存在一些强假设条件, 有待进一步完善。如：假设所输入的事件流是流水线型的, 有严格的时间序列, 而不考虑事件的并发等情况; 另外, 没有考虑更加复杂的复合事件类型, 即没有考虑将得到的复合事件再次作为输入, 从而得到更加复杂的复合事件的情况。

Table 4 The generation of 4-pattern
表 4 4-模式的产生

频繁 3-模式	频繁 2-模式
(*A* Overlap[0,0,0,1,0] *B*) Before[0,0,0,1,0] *D*	*C* Contain[1,0,0,0,0] *D*
(A Before[0,0,0,0,0] *F*) Before[0,0,0,0,0] *G*	*A* Before[0,0,0,0,0] *D*
(*A* Overlap[0,0,0,1,0] *B*) Overlap[0,0,0,2,0] *C*	*B* Before[0,0,0,0,0] *D*
(*A* Overlap[0,0,0,1,0] *C*) Contain[1,0,0,0,0] *D*	*A* Before[0,0,0,0,0] *F*
(*A* Before[0,0,0,0,0] *D*) Before[0,0,0,0,0] *F*	*A* Overlap[0,0,0,1,0] *C*
(*A* Overlap[0,0,0,1,0] *B*) Before[0,0,0,0,0] *F*	*B* Overlap[0,0,0,1,0] *C*
(*B* Overlap[0,0,0,1,0] C) Contain[1,0,0,0,0] *D*	*D* Before[0,0,0,0,0] *F*
(*B* Before[0,0,0,0,0] *D*) Before[0,0,0,0,0] *F*	*A* Overlap[0,0,0,1,0] *B*
	B Before[0,0,0,0,0] *F*

4.5.2 自动检测

提出了一种基于时态模式的复合事件检测方法[21]。以往的研究都是从两个(k–1)时态模式中产生 k 级模式，但是这种方法会导致产生大量无用的模式，因此文献[21]引入模式中“控制事件”(dominant event)的概念。如果一个事件在模式 P 中发生，并且此事件的结束时间是 P 的所有事件中最晚的，那么就称该事件是模式 P 的控制事件。因此，如果(k–1)-模式中的控制事件正好是一个 2-模式的第一个事件，则这两个模式就可以做连接，组成一个 k-模式的复合事件。如表 4 所示，3-模式的控制事件有下划线，2-模式的第一个事件被加粗。可以通过连接有共享控制事件的 3-模式和 2-模式来得到 4-模式。例如，连接表 4 中第 1 列的第 3 个模式和第 2 列的第 1 个模式，可以得到 4-模式((*A* Overlap[0,0,0,1,0] *B*) Overlap[0,0,0,2,0] *C*) Contain[1,0,0,0,0] *D*。

这种复合事件检测方法的关键是维护频繁 2-模式的实时更新，将不满足条件的时态模式及时地从频繁 2-模式的列表中删除。该条件 1 定义为：如果一个(k+1)模式是由一个频繁 k 模式和一个 2-模式产生的，并且该 2-模式至少在(k–1)个频繁 k 模式中发生过，那么该(k+1)模式就可以作为一个候选模式(详细证明见文献[21])。基于上面的定义，从一系列频繁 k 模式中产生一个 2-模式列表的同时，会为列表中的每个实体记数，这个数字表示包含该实体的频繁模式的数目。当一个实体的记数小于(k–1)时，就将该实体从 2-模式列表中删除，因为它已经不能用于产生(k+1)模式。例如表 4 中的 2-模式“*F* Before[0,0,0,0,0] *G*” 只在频繁 3-模式的第 2 行中出现。如果用该 2-模式来扩充频繁 3-模式“(*A* Before[0,0,0,0,0] *D*) Before[0,0,0,0,0] *F*)”，就会产生候选模式“((*A* Before[0,0,0,0,0] *D*) Before[0,0,0,0,0] *F*) Before[0,0,0,0,0] *G*”。由条件1，它当中的每个子模式必须是频繁的，但它的子模式“(*D* Before[0,0,0,0,0]*F*) Before[0,0,0,0, 0] *G*” 不是频繁的。因此，即使保留该记数小于(k–1)的 2-模式，也无法由它产生有效的候选时态模式。

这种基于时态模式的复合事件检测方法的算法表示如图 6。首先扫描数据库得到所有频繁原子事件(第 1 行)，这些事件被放入频繁集合 FrequentSet 中(第 2 行)。然后调用函数 GetNext-CandidateSet[21]得到一个初始化的 2 级候选集合 CandidateSet(第 3～4 行)，本算法的目标就是从 CandidateSet 中得到频繁时态模式。对事件列表集合 EventListSet 中的每个 EL 都调用 CountSupport[21] 来获取 CandidateSet 中每个时态模式的支持数目(第 6～8 行)。当 EventListSet 中的每个 EL 都被检测过之后，就可以得到频繁模式了(第 9 行)。函数 GetNextCandidateSet 返回下一级的候选时态模式(第 10～11 行)。当 CandidateSet 为空时，算法终止(第 12 行)。

```
1. Scan database and obtain all single frequent events
2. FrequentSet ← all single frequent events
3. CandidateSet ← GetNextCandidateSet (FrequentSet)
4. Level ← 2
5. Repeat
6.     For all (EL∈EventListSet) do
7.         CountSupport (Level, EL, CandidateSet)
8.     End for
9.     FrequentSet ← obtain frequent patterns
10. CandidateSet ← GetNextCandidateSet (FrequentSet)
11. Level ← Level +1
12.Until (CandidateSet = Φ)
```

Fig.6 Algorithm of complex event detection based on temporal pattern

图 6 基于时态模式的复合事件检测算法

5 研究展望

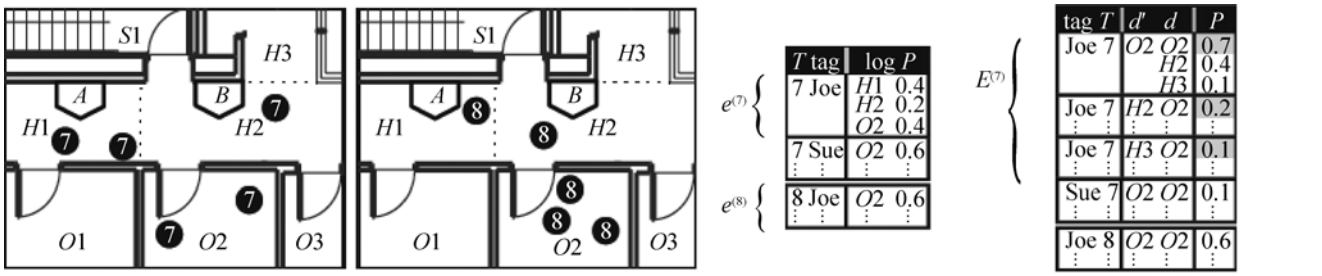
随着复合事件检测应用的日益广泛，在复合事件检测的研究与分析中涌现出了许多有趣的问题。如：在“事件”方面，除了要考虑原子事件和复合事件之外，还要考虑更加复杂的事件，即考虑事件类型的等级性，另外，在复合事件检测的过程中，还要充分考虑各事件之间的关联性；在“时间”方面，需要考虑事件无序性的情况，即在实际应用中事件并不都是完全有序的；在“数据”方面，需要考虑普适计算环境下不可避免的数据不确定性问题，分析不确定性事件的检测(包括局部不确定性和全局不确定性)。具体说明如下。

5.1 事件的不确定性

现有的复合事件检测研究通常都假设事件是精确的，然而在许多现实应用中事件都是不确定的。不确定性问题是普适计算环境下的一个本质问题，也是复合事件检测中的一个主要问题。例如在“智能家居”应用中，各传感器数据是不确定的，家居中人的行为模式和习性也是不确定的，如何根据这些不确定的数据推导出对用户有用的、确定的信息是一个挑战性问题；在路网应用中，各车辆和信息载体都是运动的，它们的位置是不确定的，获取的信息也可能是不确定的，如何有效地处理这些不确定性数据也是很重要的。虽然近几年不确定性数据研究成为一个热点问题，但在不确定性复合事件检测中还有很多问题有待深入探讨。例如：由于各种原因产生的传感器数据的不确定性、事件的局部不确定性、事件之间相互关联的全局不确定性等(见第3章的具体分析)。目前的研究工作主要集中在不确定性数据的表示模型以及不确定性数据的查询处理等方面，但是这些模型和算法在计算代价、查询效率等方面还存在诸多缺陷。今后的研究除了对表示模型和查询处理继续优化之外，还要在不确定性数据的存储与索引技术，位置相关的不确定性数据服务，不确定性数据的分析与挖掘技术等方面进行深入探讨。总之，随着大量不确定数据的产生，对不确定的复合事件检测的研究分析将变得越来越重要。不确定性复合事件检测是一个亟待解决的问题。

5.2 事件之间的关联性

现有的复合事件检测研究通常都存在一个强假设，就是事件流之间是相互独立的，不同事件之间不会相互关联，也不会互相影响。但是在现实应用中，事件之间存在着千丝万缕的联系。例如：第 3 章中 Mary 和 Joe 喝咖啡的时间、地点以及持续时间都是相关的；4.2 节中商品是否误放与商品是否结账是相关的；4.3 节中患者是否有糖尿病与血糖的增高和尿糖的缺失是否同时并存是相关的，等等。如果不考虑这些事件之间的关联性，就会得出如 4.2 节所描述的错误结论。因此在进行复合事件检测时，必须全面地考虑同一个体不同时刻之间的关系，以及不同个体之间的相互作用和相互影响，另外可能还需要考虑此个体的身份职位等因素的影响和关联。在此，以文献[22]中的一个例子来说明。如图 7(a)和 7(b)分别表示 Joe 在 T=7 和 T=8 时刻的位置，由图可知其位置是不确定的；文献[22]中以某位置所含粒子的个数占全部个数的比例来简单地表示对



(a) The location of Joe at T=7 (a) T=7时Joe的位置　(b) The location of Joe at T=8 (b) T=8时Joe的位置　(c) The location relationship of Joe and Sue (c) Joe和Sue位置的关系　(d) The location relationship of Joe at T=7 and T=8 (d) Joe在T=7和T=8时的位置相关性

Fig.7　The relationship of events

图 7　事件的关联性

象在该位置的概率；由 7(c)看到，Joe 和 Sue 的位置也是相互关联的，即根据 Joe 的位置的不确定度，可以大致推断出 Sue 的位置。7(c)中 T=7 时刻 Joe 在 $H1$ 和 $O2$ 的概率都是 0.4，但是如果知道 Sue 是 Joe 的秘书，即 Sue 和 Joe 一般是在一起的，那么根据 T=7 时刻 Sue 在 $O2$ 的概率为 0.6，可以推断出 T=7 时刻 Joe 很可能也在 $O2$ 位置。然而包括文献[22]在内的目前的研究工作都没有考虑这些因素的影响和关联；又由 7(d)看到，Joe 在 T=7 和 T=8 时刻的位置是相互关联的，即根据 Joe 在 T=7 时刻的位置的不确定度，可以大致推断出 Joe 在 T=8 时刻的位置。如果 Joe T=7 时刻在 $O2$ 位置，那么 T=8 时刻 Joe 很有可能还在 $O2$ 位置，所以$<O2,O2>$的概率为 0.7，记为 $P<O2,O2>$=0.7，比 $P<H2,O2>$=0.2 和 $P<H3,O2>$=0.1 大的多。

5.3　复合事件类型的等级性

目前的很多研究工作都是将事件从原子类型转化成复合类型，很少有研究工作是将复合类型的事件进一步转化成更为复杂的复合事件类型。后者的研究是要将前者的输出结果作为输入，因此前者的研究也是进行更为复杂的事件复合过程的重要一步。4.5.1 小节中[17]提出的复合事件检测方法就无法处理更为复杂的复合事件检测过程，因为其输入事件流仅限于含有时间戳的原子事件，也就是说这种检测方法的输出结果无法作为输入。然而随着现实世界应用的日益广泛，更为复杂的复合事件检测必将越来越重要。例如第 1 章中提到的健康护理，要想知道被照顾人是否已经被很好地照顾，需要知道被照顾人的一系列行为流，如："他是否按时吃药了？是否按时吃饭了？在睡觉之前是否刷牙了？体温血压是否正常？等"。在这个例子中，可以把健康护理的整个流程看作一个更为复杂的复合事件，其中牵涉到的一系列行为既可能是原子事件，也可能是复合事件。如检查"他是否按时吃药"就是由以下原子事件构成的："倒了一杯水"、"拿起了药瓶"、"喝水"等，因此检查"他是否按时吃药"就是一个复合事件，而检查"体温、血压的示数是否正常"就是原子事件。用图 8 来直观地表示复合事件类型的等级性。今后的研究工作需要考虑这种等级性，根据不同的需求检测出合理的复合事件类型。

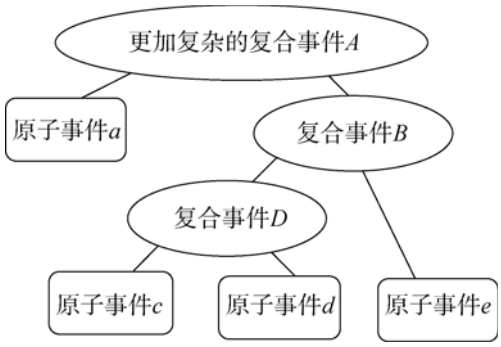


Fig.8　The gradability of complex events

图 8　复合事件类型的等级性

5.4　事件的无序性

目前的很多研究都是在进入事件处理系统

之前，为每个事件设定一个时间戳。这些时间戳是离散的、有序的，能够反映这些事件的实际发生顺序，他们通常都假设这些事件是完全有序的。如 4.5.1 小节中[17]提出的复合事件检测方法就假设输入的原子事件是完全有序的，也就是说不考虑事件的并发性和重叠性。而实际上这种假设并不是在所有场景下都成立，在很多情况下，事件可能是同时发生的。例如：一个复合事件通常从它的原子事件中获取时间戳，当这些由很多原子事件组成的复合事件用来检测更为复杂的复合事件时，事件之间全序性的假设就不成立了。仍以第 1 章提到的健康护理为例，在进行“健康护理”这个更为复杂的复合事件检测的过程中，刷牙、量体温等事件很有可能是同时进行的，并且由于不同人的生活习性不同，此过程中各原子事件或复合事件之间很有可能是完全无序的，因此传统的检测方法就无效了，未来的复合事件检测研究中必须考虑事件的无序性特征。

5.5 事件的分布性

普适计算中的大量设备如相机、汽车、手机、家电等均将具有一定的存储能力，用于收集和存储相应的各种数字信息。其中每个移动设备均是一个数据源，每个数据源能力有限，且不存在集中固定的服务器为各移动数据源提供支持，也就是说许多现实应用(例如监控环境)中包含大量的分布事件源(如硬件传感器和软件接收器等)。但是目前的研究工作通常都假设所有相关的事件是集中的，很少考虑分布式环境中的复合事件检测机制。如果将传统的检测方法应用到分布式环境中，效率会很低，因为它需要检测所有原子事件和处理单元的交互，而实际上组成复合事件的往往只是所有原子事件中的一小部分。例如路网交通安全监控中所获得的原始数据/信息，并不都是用户感兴趣的，有些甚至是完全无用的。因此，如何从众多的分布式事件源中选取有价值的信息，进行有效的查询、监控和分发是普适计算环境下的基本问题。分布式环境中复合事件检测的目标就是检测尽可能少的原子事件，同时又不遗漏任何用户感兴趣的复合事件，其中要全面考虑原子事件的并发性、异步性、不确定性等因素，因此还需要做大量的研究工作。

6 结论

随着传感器和无线设备的大规模使用，产生了数量巨大的原子事件，因此如何从这些底层的原子事件中抽取人们感兴趣的、有用的复合事件就变得越来越重要。目前研究人员围绕复合事件检测，从时间因素、复合事件表示方法、数据的精确度等方面做了很多研究。从上述 3 个方面，对近几年国际上在该领域的主要研究成果进行了回顾和总结，并提出了仍然存在的问题和今后的研究方向。总的来说，复合事件检测的日益重要和普适计算环境下大量不确定性数据的产生，使得这一研究领域中的关键问题需要进一步深入探索。

References:

[1] Liao L, Patterson D J, Fox D, et al. Learning and inferring transportation routines[J]. Artif Intell, 2007, 171(5/6): 311–331.

[2] Philipose M, Fishkin K P, Perkowitz M, et al. Inferring activities from interactions with objects[J]. IEEE Pervasive Computing, 2004, 3(4): 50–57.

[3] Demers A J, Gehrke J, Hong M, et al. Towards expressive publish/subscribe system[C]//Proceedings of Advances in Database Technology Conference, 2006: 627–644.

[4] Jobst D, Preissler G. Mapping clouds of SOA-and business-related events for an enterprise cockpit in a Java-based environment[C]//Proceedings of the 4th ACM International Conference on Principles and Practices of Programming In Java(PPPJ), 2006: 230–236.

[5] Wang F, Liu P. Temporal management of RFID data[C]//Proceedings of the 31st VLDB Conference, September

2005.

[6] Das S K. The role of prediction algorithms in the MavHome smart home architecture[J]. IEEE Wireless Communications, 2002, 9(6):77–84.

[7] Lamming M, Bohm D. SPECs: Another approach to human context and activity sensing research, using tiny peer-to-peer wireless computers[C]//Proceedings of the 5th International Conference on Ubiquitous Computing, 2003, 2864: 192–199.

[8] McCarthy J F, Anagnost T D. Event manager: Support for the peripheral awareness of events[D]. Hebrew Union College, 2000: 227–235.

[9] Virone G, Wood A, Selavo L, et al. An assisted living oriented information system based on a residential wireless sensor network[C]//Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference, April 2006.

[10] Mert A, Ugur G, Nesime T. Plan-based complex event detection across distributed sources[J]. The VLDB Endowment, 2008, 1(1): 66–77.

[11] Samani M, Sloman M, Gen M. A generalized event monitoring language for distributed systems[J]. IEEE/IOP/BCS Distributed Systems Engineering, 1997, 4(2): 96–108.

[12] Chakravarthy S, Rasad V K, Anwar E. Anatomy of a composite event detector, Technical Report UF–CIS–TR–93–039[R]. Gainesville: University of Florida, 1993.

[13] Gehani N H, Jagadish H V, Shmueli O. Event specification in an active object–oriented database[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1992: 81–90.

[14] Gatziu, S, Dittrich K R. Events in an active object-oriented database system[C]//Proceeding of the 1st Int'l Conference on Rules in Database Systems, 1993: 23–39.

[15] Wu E, Diao Y, Rizvi S. High-performance complex event processing over streams[C]//Proceedings of the 2006 ACM SIGMOD, New York, USA, 2006: 407–418.

[16] Jagrati A, Diao Y L, Daniel G, et al. Efficient pattern matching over event streams[C]//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, 2008: 147–160.

[17] Eugene W, Diao Y L, Shariq R. High-performance complex event processing over streams[C]//Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, Chicago, IL, USA, 2006: 407–418.

[18] Kam P S, Ada W F. Discovering temporal patterns for interval-based events[C]//Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery, 2000.

[19] Panagiotis P, George K, Stan S, et al. Discovering frequent arrangements of temporal intervals[C]//Proceedings of the 5th IEEE International Conference on Data Mining , 2005: 354–361

[20] Wu S Y, Chen Y L. Mining nonambiguous temporal patterns for interval-based events[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(6): 742–758.

[21] Dhaval P, Wynne H, Mong L L. Mining relationships among interval-based events for classification[C]//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, 2008: 393–404.

[22] Christopher R, Julie L, Magdalena B, et al. Event queries on correlated probabilistic streams[C]//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, 2008: 715–728.

[23] Zimmer D, Unland R. On the semantics of complex events in active database management systems[C]//Proceedings of the 15th International Conference on Data Engineering (ICDE), 23-26 March, 1999.

[24] Antunes C, Oliveira A L. Generalization of pattern-growth methods for sequential pattern mining with gap constraint[C]//Proceedings of the Int'l Conference Machine Learning and Data Mining (MLDM'03), 2003.

[25] Pei J, Han J, Mortazavi A B, et al. Prefixspan: Mining sequential patterns *e* efficiently by prefix-projected pattern growth[C]//Proceedings of the 17th International Conference on Data Engineering (ICDE), April 2001.

[26] Mannila H, Toivonen H, Verkamo I. Discovery of frequent episodes in event sequences[C]//Proceedings of the

Annual ACM SIGKDD Conference, 1995.

[27] Agrawal R, Srikant R. Mining sequential patterns[C]// Proceedings of the IEEE International Conference on Data Engineering (ICDE), 1995.

[28] Allen J F. Maintaining knowledge about temporal intervals[J]. Communications of the ACM, 1983, 26(11).

[29] Hakeem A, Sheikh Y, Shah M. A hierarchical event representation for the analysis of videos[C]//Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004), San Jose, California, July 25-26, 2004.

[30] Zhao T, Nevatia R, Hongeng S. Hierarchical language-based representation of events in video streams[C]//Proceedings of the IEEE Workshop on Event Mining, 2003.

[31] Jeffery S. Adaptive cleaning for RFID data streams[C]// Proceedings of the 32nd VLDB Conference, September 2006.

[32] Floerkemeier C, Lampe M. Issues with RFID usage in ubiquitous computing applications[C]//Proceedings of the 2nd Pervasive Conference, April 2004.

[33] Adaikkalavan R, Chakravarthy S. Snoopib: Interval-based event specification and detection for active databases[J]. Data Knowledge Engineering, 2006, 59(1): 139–165.

[34] Levinson S E, Rabiner L R, Sondhi M M. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition[J]. Journal of Bell System Technology, 1983, 62(4).

[35] Kanagal B, Deshpande A. Online filtering, smoothing and probabilistic modeling of streaming data, Technical Report CS-TR-4867[R]. Maryland: University of Maryland, 2007-05.

ZHOU Chunjie was born in 1981.She is a Ph.D. candidate at Renmin University of China. Her research interests include pervasive computing, mobile data management and workflow management, etc.

周春姐(1981–), 女, 山东烟台人, 中国人民大学博士研究生, 主要研究领域为普适计算, 移动数据管理和工作流管理等。

MENG Xiaofeng was born in 1964. He is a professor and doctoral supervisor at Renmin University of China, and the member of CCF. His research interests include Web data management, XML database and mobile data management, etc.

孟小峰(1964–), 男, 河北邯郸人, 中国人民大学教授、博士生导师, CCF 会员, 主要研究领域为 Web 数据管理, XML 数据库, 移动数据管理等。

Benchmarking Cloud-based Data Management Systems

Yingjie Shi, Xiaofeng Meng, Jing Zhao, Xiangmei Hu, Bingbing Liu and Haiping Wang
School of Information, Renmin University of China
Beijing, China, 100872
shiyingjie1983@yahoo.com.cn,
{xfmeng,zhaoj,hxm2008,liubingbing,wanghaiping1022}@ruc.edu.cn

ABSTRACT

Cloud-based data management system is emerging as a scalable, fault tolerant and efficient solution to large scale data management. More and more companies are moving their data management applications from expensive, high-end servers to the cloud which is composed of cheaper, commodity machines. The implementations of existing cloud-based data management systems represent a wide range of approaches, including storage architectures, data models, tradeoffs in consistency and availability, etc. Several benchmarks have been proposed to evaluate the performance. However, there were no reported studies about these benchmark results which provide users with insights on the impacts of different implementation approaches on the performance. We conducted comprehensive experiments on several representative cloud-based data management systems to explore relative performance of different implementation approaches, the results are valuable for further research and development of cloud-based data management systems.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*

General Terms

Measurement,Performance

Keywords

cloud, data management, benchmark

1. INTRODUCTION

Cloud computing has emerged as a prevalent infrastructure and attracted a lot of attention of companies and academic circles. Though there has not been a standard definition about cloud computing, we can summarize the substantial features of it: scalability, fault tolerance, high performance cost, pay-as-you-go, etc. Data management system is one of the applications that are deployed in the cloud, many cloud-based data management systems have been proposed and are serving online right now: BigTable[1] in Google, Cassandra[2] and Hive[3] in Facebook, HBase[4]in Streamy, PNUTS[6] in Yahoo! and many other systems. In order to merge with the cloud computing platform, the data management system should have high availability and fault tolerance, flexible scalability, and the ability to run in the heterogeneous environment. Developers of the existing systems choose different solutions to make their data management systems work well in the cloud depending on their different application scenarios.

The developers of many companies are wondering whether their data management applications can be moved to the cloud to get better performance with less cost. However, the application environments and implementation approaches of existing cloud-based data management systems are so various that it is difficult for developers to determine which system is more suitable for their applications. The significance of conducting comprehensive experiments on the cloud-based data management systems can be summarized as follows: showing the performance advantage of different systems and providing the users with impacts of different technical issues on the performance. The test results and analysis are useful for both further research and development of cloud-based data management systems.

There have been several benchmarks proposed to evaluate the performance of cloud-based data management systems, including performance evaluation of the Google's BigTable [1] which is for systems that do not support structured query language, the performance comparison of Hadoop[15] and some parallel DBMSs [8] which put emphasis on structured query of the systems, and the Yahoo! Cloud Serving Benchmark(YCSB)[9] framework which supplies several workloads with different combinations of insert, read, update and scan. Several experiment reports depending on these benchmarks can also be found, however, all of them focus on one or two systems' performance evaluation without comparison analysis depending on their implementation approaches. Three cloud-based management systems are included in[9]: Cassandra, HBase and PNUTS, they present results of three workloads: update heavy, read heavy and short ranges. And all the workloads in YCSB focus on serving systems just like PNUTS, which provide online read or write access to data. The analytical systems are not included in their workload. And all the systems in their experiment do not use replication, which is widely used in the cloud-based systems for data availability, so they do not evaluate the fault tolerance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CloudDB'10, October 30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0380-4/10/10 ...$10.00.

or availability of these systems either. We conduct comprehensive experiments on the representative cloud-based data management systems, which cover the different approaches on storage architectures and data models. Our workloads and tasks originate from benchmarks in [1] and [8], we make some extensions to investigate the factors that affect performance of different implementations.

The rest of this article is organized as follows. Section 2 summarizes the existing cloud-based data management systems with emphasis on storage architecture and data model. Section 3 describes the workloads and tasks in the benchmark. The test results and analysis are given in Section 4. We describe the future work and conclude the article in Section 5.

2. AN OVERVIEW OF EXISTING CLOUD-BASED DATA MANAGEMENT SYSTEMS

Cloud-based data management system will not replace the traditional RDBMS in the near future, however, it supplies another choice for the applications which are suitable to be deployed in the cloud: large scale data analysis and data management in the web applications. Different from transactional applications, data involved in the analysis are rarely updated, so ACID guarantees in the transactional applications are not needed. Data analysis applications are often deployed on shared-nothing parallel databases, but with the increase of data scale, systems will have to scale vertically which costs a lot of money and time to get better performance. Cloud-based data management systems provide a flexible and economical solution to scale horizontally with commodities, and the scaling server resources are transparent to the applications. In the web data management applications, response time is one of the most important requests except for scalability and fault tolerance. Big data is produced during the interaction between customers and the sites, and we can not see the increase end in sight. Many companies which supply social network services have moved some of their applications to cloud-based data management systems because of data explosion[13]. During the existing cloud-based data mangement systems, BigTable, HBase, HyperTable, Hive and HadoopDB[10] are mostly used for analytical data management applications, while PNUTS and Cassandra are used for web data management. Different applications originate from different implementation approaches, next we will compare the technical issues from storage architecture and data model.

2.1 Storage Architecture

Depending on the persistency design, we can classify the cloud-based data management systems into two kinds: File System-based systems and DBMS-based systems. BigTable, HBase, HyperTable, Hive and Cassandra are File System-based softwares. HBase and HyperTable are open-source implementations of BigTable's architecture, they are called the BigTable-like systems. The BigTable-like systems and Hive all store data in distributed file systems, which is master-slave organized. While Cassandra uses file system directly as the storage layer, which is peer-to-peer organized. Table 1 shows the different file systems they use.

SQL Azure, PNUTS, HadoopDB and Voldemort use traditional DBMS as the storage layer. DBMS's inherent features such as query optimization, index techniques can be directly utilized in this kind of systems, and it's easy for this kind of system to support SQL to the users. However, under this kind of architecture, DBMS layer can be a bottleneck for data storage, because all the data storage optimization can only be executed on top of DBMS.

Table 1: File Systems in the Storage Layer

Project Name	File System
BigTable	GFS[14]
HBase	HDFS[16]
HyperTable	KFS[17], HDFS
Hive	HDFS
Cassandra	Local File System

Generally speaking, FileSystem-based systems inherit several merits from MapReduce[7] if they use MapReduce as the framework, such as scalability, fault-tollerance, adapting to heterogeneous environment, etc. However, most of these systems can not support SQL, except for Hive which can support part of SQL called HQL. DBMS-based systems can support SQL, however, there is a lot of work to do on scalability, fault tolerance, and support for semi-structured and unstructured data.

2.2 Data Model

The data models of existing cloud-based data management systems are extremely different, we classify them into two kinds: the key-value data model and the simplified relational data model.

The key-value data model is a sparse, distributed, persistent multidimensional sorted map[1], it is simple and flexible. There are no different data types, all data is stored as bytes. The elements in the multidimensional map are not only rows and columns, but also column families, timestamps, etc. The rows and columns represent what they mean in the relational data model, but the rows are sparse: each row can have different number of columns in one table, and columns can be added during the process of data loading. The unique row key identify one record, it is mapped to a list of column families. The column family is mapped to a list of columns, while the column is mapped to a list of timestamps, then the timestamp is mapped to the value. The value is fixed by a key consisting of row key, column family, column name and timestamp, we can simply summarize the map relationship as <row key,<column family,<columname ,<timestamp, value >>>>. A set of columns are put together into one column family, which is also the data access unit. Data of the same column family is stored in one file on disk, so clients are suggested to put the columns which are often queried together into one column family to get better performance. Timestamps are used in two ways: indexing multiple versions of data, and conflict resolution.

Data in the DBMS-based systems is eventually stored in the RDBMS, they adopt the relational data model with some varieties in order to support distributed applications. For example, traditional relational data model ensures entity integrity and referential integrity, however, now most cloud-based data management systems do not ensure referential integrity. Hive is one of the systems that use the relational data model, data in Hive is organized into tables which are analogous to tables in relation databases, each table has a

Table 2: Tasks in data read and write benchmark

Task Name	Details
Sequential Write(i)	Write rows into an empty table under sequential row keys.
Sequential Write	Write rows into a table which has already stored data under sequential row keys.
Random Write(i)	Write rows into an empty table under random row keys.
Random Write	Write rows into a table which has already stored data under random row keys.
Sequential Read	Read rows under sequential row keys.
Random Read	Read rows under random row keys.
Scan	Scan the whole table.

corresponding HDFS directory[3]. Hive does not support primary key or foreign key yet.

The relational data model has solid theoretical basis and refined implement technologies, however, it is difficult to use it directly in the cloud environment. The key-value data model is simple and easy to implement, however, all systems of this data model support APIs instead of a uniform language like SQL, which supplies sophisticated DDL and DML operations. So there is a lot of work to do to widen the appliation scope of the key-value model systems.

3. WORKLOADS OF THE BENCHMARK

We conducted two benchmarks on several existing cloud-based management systems: data read and write benchmark and structured query benchmark. During the data read and write benchmark, seven tasks are defined to evaluate the read and write performance during different situations. The structured query benchmark focus on some basic operations in the structured query language, including key words matching, range query, aggregation and so on. In fact many cloud-based data management systems do not support SQL, we evaluate their performance in this benchmark by coding the client through the APIs they provide. Next we will describe the details of workloads in these two benchmarks.

3.1 Data Read and Write Benchmark

The principles of this benchmark originate from the performance evaluation section of BigTable paper[1], we also add some tasks included in a test report[11] which shows the test results of HBase-0.20.0 on 5 servers. There are seven tasks as Table 2 shows.

All of the tasks operate on a column family with one column, one row is written or read during one operation. The row size is 1010 bytes: 1000 bytes for value, and 10 bytes for row key. We write a string of 1000 bytes into one row as the value, each string is composed by characters random generated. The sequential read and write are operations with row keys in order, while random operations using row keys out-of-order, and the motivation is to determine whether performance can be affected by different row key choosing methods. Initial write is an operation against an empty table, while the other kind of write is operation against a table which has already stored data partitioned in the whole system. The motivation is to examine how the existing data can affect the write performance of systems with different storage architectures. Scan is also reading rows under sequential row keys, the difference between scan and sequential read is that we call the special interfaces the systems supply. During the task of sequential read, one row is returned once we call the API, but during the task of scan, all the rows in the table are returned once we call the scan API. Scan is one of the most important applications in the cloud-based data management systems during data analysis. The replication factors of all systems involved are set to 3, which is widely used in the distributed systems.

In addition to the data read and write performance, scalability is also an important characteristic of the cloud-based systems. A system has scalability means that more servers will create more capacity and the scaling server resources are transparent to the applications. Speedup is widely used to measure the scalability of distributed systems. In order to evaluate the scalability of systems in a more detailed way, we also compute the speedup during this benchmark by executing these tasks on systems deployed on different numbers of servers.

3.2 Data Load and Structured Query Benchmark

Structured query language is widely supported by traditional data management systems, and it makes applications development on the cloud system much easier. Until now, the DBMS-based systems can support part of SQL, while most of the FileSystem-based systems don't provide SQL APIs. We conduct this benchmark on these two kinds of systems, and for systems that do not support SQL, we implemented the structured query through coding on other APIs they provide to analyze their performances. The workload of this benchmark originates from Pavlo's work[8], it is used to evaluate performance of cloud-based data management systems and parallel databases, focusing on structured data load and query. Our motivation of conducting this benchmark is to compare the performances of cloud-based data management systems with different architectures and implementation approaches on structured data query. Three tables are involved in the dataset, Table 3 described the structures of them. The table of rankings and uservisits simulate the the page ranks and visit logs of web pages. There are five tasks in this benchmark: data load, grep query, range query, aggregation and fault tolerance.

- Data Load: Data is loaded into cloud-based data management systems from files on local file system of the client, the replication factor of data is also set to 3.
- Grep Query: The table of grep contains a column of 10 bytes as the key, and a column of 90 bytes as the field to be patterned with some key words, during our test we use the key word 'XYZ'.
- Range Query: Table involved in this task is "rankings", which stores information of page URL and page rank. The range query will return the records with page rank in a special region.
- Aggregation: Compute the total adRevenue generated from each sourceIP in the table "uservisits", grouped by the column of "soureIP".

Table 3: Tables in the dataset

Table Name	Table Structure
grep	key VARCHAR(10), field ARCHAR(90)
rankings	pageRank INT, pageURL VARCHAR(100),avgDuration INT
uservisits	sourceIP VARCHAR(16), destURL VARCHAR(100), visitDate DATE, adRevenue FLOAT, userAgent VARCHAR(64), countryCode VARCHAR(3), languageCode VARCHAR(6), searchWord VARCHAR(32), duration INT

Table 4: Testbed Setup

CPU	Quad Core 2.33GHz(5 servers) Quad Core 2.66GHz(15 servers)
RAM	7GB(5 servers) 8GB(15 servers)
Disk	1.8TB(5 servers) 2TB(15 servers)
Operating System	Linux: Ubuntu 9.04 Server
Network	1000Mbps

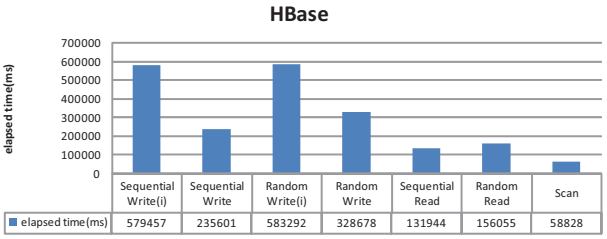


Figure 1: HBase performance on 20 nodes

The tasks above are basic operations in structured query, most cloud-based data management systems can't support sophisticated queries currently, however, we can overview their performances in structured query from these simple operations. In a cloud-based data management system with high fault tolerance, a query does not have to be restarted when one of the servers involved in the query failed. We also evaluate the fault tolerance of the systems in this benchmark through comparing the elapsed time during the normal situation and fault situation.

4. PERFORMANCE EVALUATION

In this section we describe the implementation details of the benchmark and the analysis of test results. Four systems are focused in the benchmark: HBase, Cassandra, Hive and HadoopDB. We tried to evaluate systems that can cover all the architecture types from open source software: HBase is one of the BigTable-like systems based on master-slave architecture; Cassandra is also Filesystem-based and adopts the P2P architecture; during all the FileSystem-based systems we have surveyed, only HIVE can support SQL; HadoopDB is one of the systems that are based on DBMS. We tune each system to get the best performance in our platform, and every task is executed three times to compute the average result. We choose the latest versions of these systems when we conduct the benchmark: HBase 0.20.3, Cassandra 0.6.0-beta3, Hive 0.6.0. All the cloud-based data management systems are under active development, so our results can reflect the current situations, and the results maybe different in the later versions of systems. All of the systems are deployed on 20 servers in our testbed, and the setup of these servers is shown in Table 4.

4.1 Data Read and Write Benchmark

HBase and Cassandra are involved in this benchmark, both of them are FileSystem-based, and neither of them can support SQL. The difference of these two systems is the architecture: HBase is Master-Slave organized, and Cassandra is P2P organized. 5,242,880 rows are involved in every task of this benchmark, and as mentioned in Section 3.1, the row size is 1010 bytes, so the data size of this benchmark is about 5G. The implementation details of this benchmark on these two systems are as follows:

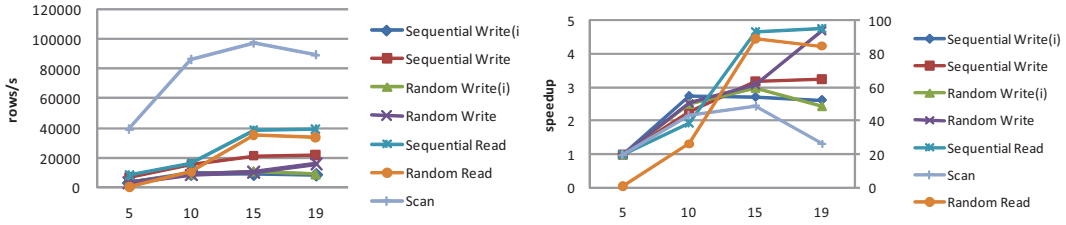
HBase: We adopt the performance evaluation package in HBase 0.20.3 and modified some code. All the tasks are implemented through MapReduce framework.

Cassandra: We code the evaluation client through the basic data read and write APIs Cassandra provides. There is no MapReduce interface in Cassandra, we use multithreads in the client to increase the parallelism degree. The client servers are also servers in Cassandra.

4.1.1 Test Results of HBase

Figure 1 illustrates the test results of HBase on 20 nodes(1 master and 19 slaves). The horizontal-axis represents the task type, and the vertical-axis shows the elapsed time. We can find that read speed is faster than that of write, this is very different from the test results of BigTable[1], in which random write is 7 times faster than random read, while sequential write is 1∽2 times faster than sequential read. Initial writes are slower than writes against an existing table. At first of the initial write, there are just two or three servers working in the task, and as time goes, more and more nodes are involved. But for the writing against an existing table which has already been distributed on 20 servers, there are 20 servers working at the beginning. For the writing against an existing table, tasks are easily to be splitted into the whole system, but for initial write, tasks are splitted among the servers as data is inserted into the servers. So the parallelism degree of writes against an existing table is bigger than that of the initial write. Scan performs better than both sequential read and random read. Scan is reading rows sequentially, and different from sequential read task which gets one row at once, it can read many rows together each time, so it costs less RPCs than sequential read.

We conduct these seven tasks on 5 slaves, 10 slaves, 15 slaves and 19 slaves separately, Figure 2(a) illustrates the scalability results of HBase, the horizontal-axis represents the number of slaves in the system, and the vertical-axis represents how many rows are operated per second of each task. We can see the performance gets better as the number of nodes increases, although the acceleration is not linear.



(a) The scalability of HBase (b) The speedup of HBase

Figure 2: The test results of HBase

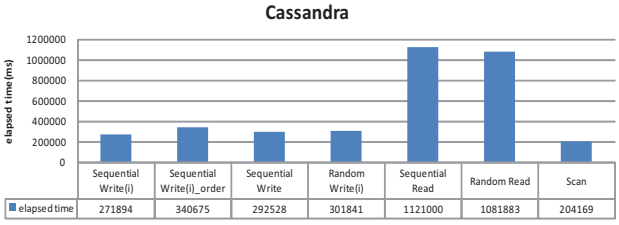


Figure 3: Cassandra performance on 20 nodes

In order to compute the speedup, we adopt the elapsed time of system with 5 servers as the base time T_{base}. And the speedup S_k is computed as: $S_k = T_k / T_{base}$. T_k is the elapsed time of task on systems of k servers. Figure 2(b) illustrates the speedup, the speedup of random read is so big that we use the right axis to present it, other tasks are presented in the left axis. The data I/O unit of HBase is block, in which there are several rows, and a whole block has to be read into memory in order to get one row. During the random read task, data is more likely to be partitioned into different region servers, this means that more compute resources can be used as the number of servers increases, so the speedup of random read is the biggest.

4.1.2 Test Results of Cassandra

In the benchmark of Cassandra, we add a task called sequential write in_order because Cassandra supports three kinds of data partitioning strategy, and two of them are used often: random partitioning and order preserving partitioning. In the random partitioning, Cassandra uses MD5 hash internally to hash the keys to determine where to place the keys on the node ring[2]. While in the order preserving partitioning, rows are stored by key order, aligning the physical structure of the data with the sort order[18].

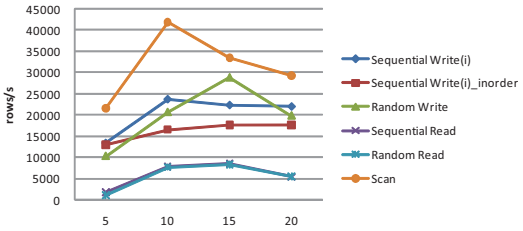
Figure 3 illustrates the test results of Cassandra on 20 nodes, and we can summarize several findings. Firstly, the sequential write in_order costs more time than sequential write, because they choose different data partitioners: the former task adopts order preserving partitioner, while the latter task adopts random partitioner. When the rows are inserted with sequential row keys, the hash function of order preserving partitioner will partition the rows to a smaller scale of servers than random partitioner does. Secondly, different from HBase, writes are faster than reads in Cassandra. Writes to each ColumnFamily of Cassandra are grouped together in an in-memory structure called memtable, then they are flushed to disk when the memtable size exceeds the threshold which is set through the parameter called MemtableThroughputInMB. This means that writes cost no random I/O, compared to a b-tree system which not only has to seek to the data location to overwrite, but also may have to seek to read different levels of the index if it outgrows disk cache[19]. Thirdly, we can see that writes against an existing table performs almost the same as writes against a new table, it is also different from HBase. This is determined by the data partitioning mechanism of Cassandra. Which server one row is partitioned into is decided by a hash function, which has nothing to do with whether there is data stored on the server before. So in the scalability test, we do not distinguish initial writes or not, all writes are operations against an empty table.

The scalability results of Cassandra are in Figure 4(a), and the speedup results are in Figure 4(b). We compute the speedup with the same way of HBase mentioned in Section 4.1.1. Most tasks perform best at the point of 15 nodes, the performances descends when there are 20 servers in the cluster. The speedup of random read is the biggest, which is the same to HBase.

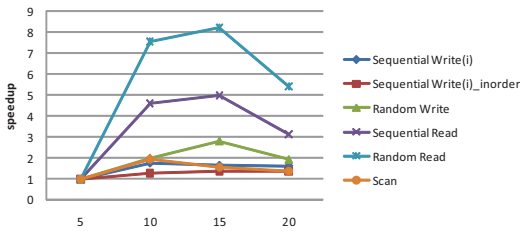
The comparison of performance between HBase and Cassandra is shown in Figure 5. We run the same workload on the two systems with 20 servers. Cassandra performs better than HBase in writes: sequential write is 2.1 times faster, and random write is 1.9 times faster. HBase performs better in reads: random read is 6.9 times faster, sequential read is 8.5 times faster, and the scan is 3.5 times faster. We also compare their performances with 15 servers, 10 servers and 5 servers, the situation that HBase performs better in read and Cassandra performs better in write exists. HBase is more suitable in the analysis applications during which data is written once and read many times, while Cassandra is more suitable to manage data in the web applications where the read traffic is heavy and full scan of the whole table is rare.

4.2 Data Load and Structured Query Benchmark

As described in Section 3.2, three tables are involved in this benchmark. Table 5 shows the data sizes of these three tables. We adopt the data generating code which is available on HadoopDB's website[12]. Systems involved in this benchmark are Hive, HadoopDB, HBase and Cassandra. HadoopDB is based on DBMS, and we deployed PostgreSQL as the storage layer in our test. HBase and Cassandra are FileSystem-based, and they don't support SQL, we code the test client by calling the APIs they provide. Next we will describe the implementation methods of every task and show the results.

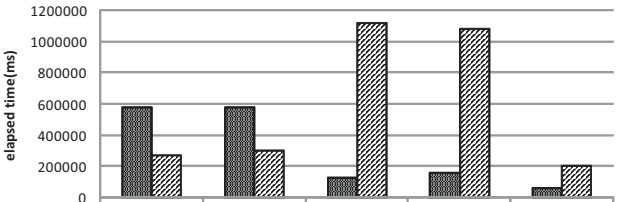


(a) The scalability of Cassandra



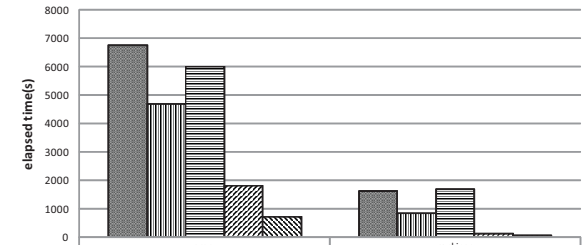
(b) The speedup of Cassandra

Figure 4: The test results of Cassandra



	Sequential Write(i)	Random Write(i)	Sequential Read	Random Read	Scan
HBase	579457	583292	131944	156055	58828
Cassandra	271894	301841	1121000	1081883	204169

Figure 5: Performance comparison between HBase and Cassandra



	grep	rankings
HBase(no MR)	6724.3	1609.6
HBase	4678	803
Cassandra	5991.7	1650.7
HadoopDB	1778.5	102.65
Hive	682.7	25.1

Figure 6: Results of data loading

Table 5: Data size

Table Name	No. of Rows	File Size
grep	500 million	50 GB
rankings	2.3 million	1.4G
uservisits	500 million	61GB

4.2.1 Data Load

The data model of HBase and Cassandra is key-value pair, we design the schemas in order to execute structured queries on them. The implementation approaches of data load are as follows:

- **HBase:** We create one table in HBase for each dataset, and each column belongs to one column family. The row key of grep is "key" and row key of rankings is "pageURL", data is loaded through "Put List" in HBase. We also run this task through MapReduce framework to get better performance.
- **Cassandra:** There are no tables in Cassandra, we use one column family to stand for one dataset. 10 client processes load data parallelly to get better throughput.
- **Hive:** Hive provides commands to load data from local file system or from HDFS, we use the following command to load data from local file system: LOAD DATA LOCAL INPATH '/home/test/grep.dat' INTO TABLE grep;
- **HadoopDB:**There are four steps to load data into HadoopDB: doing global partition to divide data file in HDFS into small files(the number of small files is equal to the number of servers in HadoopDB), loading the files into local file system of each server in the cluster, doing local partition to divide small files into chunks, then loading chunks into PostgreSQL through copy command. We sum the time these four tasks cost together as the execution time of data load on HadoopDB.

Figure 6 illustrates the results of loading data into grep and rankings. The data interface of Hive adopts MapReduce as the workflow framework. And Hive just checks whether records in the data file accord with the constrains defined in table creating. After the data checking, the whole data file is moved from local file system to the directory of Hive in HDFS directly. While HBase and Cassandra have to check every record and add some meta data such as row key, column name, and timestamp to organize SSTables, and they have to partition the record to servers of the cluster. So Hive cost least time to load data. HBase doesn't encapsulate MapReduce procedure in its data load interface, so we conduct two methods to load data into HBase: using multi client processes and using MapReduce framework. HBase performs better with the method of using MapReduce framework.

4.2.2 Grep Selection

The table of grep contains two columns: a column of 10 bytes as the key, and a column of 90 bytes as the field to be patterned with the keyword "XYZ". The keyword appears once every 108299 rows in the table of grep. We can execute the following SQL directly on Hive and HadoopDB:
SELECT key, field FROM grep WHERE key like '%XYZ%';
Because neither of HBase and Cassandra can support SQL, we complete this task through the simple APIs they provide,

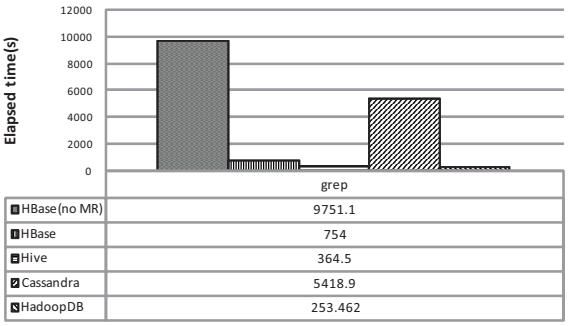


Figure 7: Result of grep select

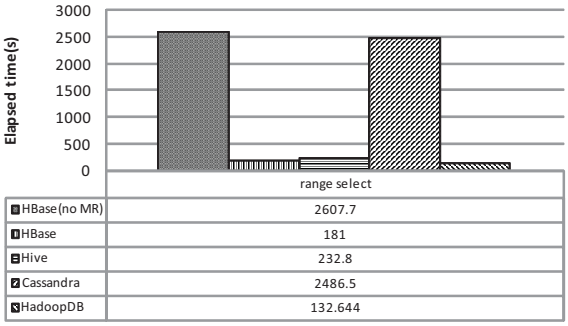


Figure 8: Result of range query

and the implementations of this query on these two systems are as follows:

- **HBase:** We write codes through the class of "Filter", which supplies several kinds of data filtering patterns in HBase. The "SubstringComparator" of HBase is case insensitive, so we implement a new substring comparator. The program is implemented in two ways: with multi client processes and with MapReduce framework.
- **Cassandra:** There is no interfaces of filtering rows in Cassandra. So we fetch the rows through "get range slice" API, then match each row with "XYZ". 10 client processes run parallelly to complete this query, and we choose the longest elapsed time of these processes as the final time of this task.

The results are illustrated in Figure 7, systems using MapReduce as the framework of the query get better performance, such as Hive, HadoopDB and HBase.

4.2.3 *Range Query*

Table involved in this task is "rankings", which stores information of page URL and page rank. The range query is based on page rank, this query will return the records with page rank in a special region. The following command can be executed on Hive and HadoopDB:

SELECT pageRank, pageURL FROM rankings
WHERE pageRank > 10;

The implementation approaches of this query on HBase and Cassandra are almost the same as described in Section 4.2.2. And we just use different filter patterns in this task. The results are shown in Figure 8. Both of the range query and grep selection need full scan of the table, systems using MapReduce as the framework of the query get better performance.

4.2.4 *Aggregation*

During the four systems, only Hive and HadoopDB support aggregation. This task computes the total adRevenue generated from each sourceIP in the table "uservisits". The following command can be executed on Hive and HadoopDB:

SELECT sourceIP, SUM(adRevenue) FROM uservisits
GROUP BY sourceIP;

Figure 9 shows the results of our test on Hive and HadoopDB. Both of them adopt MapReduce as the framework. SQL command the user put forward to Hive is translated into MapReduce operations that can be executed on HDFS. While

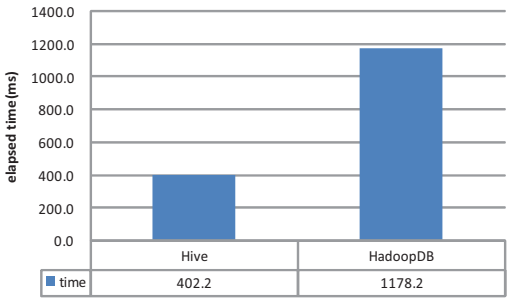


Figure 9: Result of aggregation

the SQL command put into HadoopDB is translated into MapReduce operations, these operations are then translated into SQL commands that can be executed on PostegreSQL. Maybe there is room of HadoopDB to optimize the workflow in the future.

4.3 Fault Tolerance

We choose the grep selection task to test fault tolerance, because the size of table "grep" is big and this query costs more time, so we can have enough time to create the error. During the test, we create a connection error to make one server which is involved in the query fail. After error is made during the query execution procedure, we record three items: Firstly, we check whether the whole query should be restarted by the client. Secondly, if the query continue running without terminating, we check whether the final result is correct. Thirdly, we record the elapsed time of the query and then compare it with the result without error during the query.

Hive and HadoopDB achieve fault tolerance through MapReduce, in which the failed task will be moved to another server by the job tracker. HBase does not encapsulate MapReduce in the filter APIs, so we test its fault tolerance in two ways: call the APIs directly and in MapReduce framework. For MapReduce-based systems, we can observe the job progress in Hadoop JobTracker Web GUI to determine which server is doing the query job and to be killed. In order to create error on query of Cassandra, we kill the server which a thread is fetching data from. In the test of HBase, if we call the filter API directly without MapReduce, when error happens on one server which is involved in the query, the whole job terminates without fault tolerance. The queries on HBase with MapReduce, Hive, HadoopDB and Cassan-

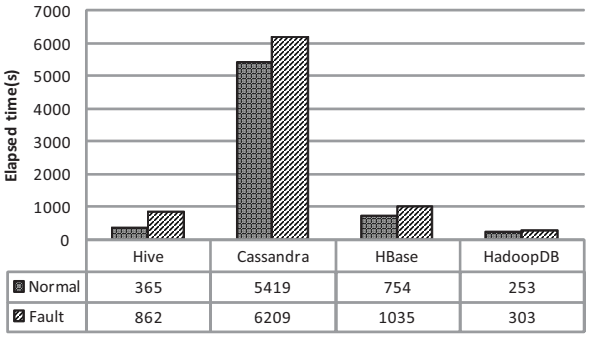


Figure 10: Result of fault tolerance test

dra continue running after error happens, and the results are all correct. Figure 10 illustrates the comparison of results under normal situation and fault situation of the four systems. The elapsed time in the fault situation is 1∼ 2 times longer than that of the normal situation.

5. CONCLUSIONS

Significant process has been made in developing data management systems on the cloud, and various cloud-based data management systems for production use have emerged. We summarize the implementation techniques of existing cloud-based data management systems from storage architecture and data model. Then we evaluate a set of systems in performance, scalability and fault tolerance, the systems cover aforementioned implementation approaches. Our test results show that systems for analysis applications perform better in data read and scan, while systems for web applications perform better in data write. Though most of the FileSystem-based systems do not support SQL, we implement some basic structured queries through the APIs they provide, and we find that some of them have almost the same or even better performance over DBMS-based systems during the structured query benchmark. The results and analysis will be valuable both for developers of cloud-based data management systems and users who are trying to move their applications to the cloud.

It is important to remember that the cloud-based data management is a very fluid field. We choose the latest version of systems when we conduct the benchmark, however, more advances will undoubtedly appear and new systems will emerge. The cloud-based data management systems are more attractive when they scale to very large workload. It's too expensive to construct an environment with hundreds or thousands of servers. In the future work, we will extend the workload scale through simulation tools and new findings about the cloud-based data management systems will be discovered.

6. ACKNOWLEDGMENTS

The authors thank anonymous reviewers for their constructive comments. This research was partially supported by the grants from the Natural Science Foundation of China (No.60833005); the National High-Tech Research and Development Plan of China (No.2007AA01Z155,2009AA011904); and the Doctoral Fund of Ministry of Education of China (No. 200800020002).

7. REFERENCES

[1] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, "Bigtable: A distributed storage system for structured data," in *Proceedings of the 7th Conference on USENIX Symposium on Operating Systems Design and Implementation*, Seattle, Washington, November 2006, pp. 205–218.

[2] Cassandra. Available at http://incubator.apache.org/cassandra/.

[3] A. Thusoo, J. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff and R. Murthy,"Hive-A Warehousing Solution Over a MapReduce Framework," in *VLDB*, Lyon, France, August 2009, pp. 1626–1629 .

[4] HBase. Available at http://hadoop.apache.org/hbase/.

[5] D. J. Abadi,"Data Management in the Cloud: Limitations and Opportunities," in *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*,2008, pp. 5–14.

[6] B. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H. Jacobsen, N. Puz, D. Weaver and R. Yerneni,"PNUTS: Yahoo!'s Hosted Data Serving Platform," in *VLDB*, Auckland, New Zealand, August 2008, pp. 1277–1288 .

[7] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, pp. 107–113, January 2008.

[8] A. Pavlo, A. Rasin, S. Madden, M. Stonebraker, D. DeWitt, E. Paulson, L. Shrinivas, and D. J. Abadi. A comparison of approaches to large-scale data analysis. In *Proceedings of the 35th SIGMOD international conference on Management of data(SIGMOD 2009)*, pages 165–178, 2009.

[9] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears,"Benchmarking Cloud Serving Systems with YCSB," in *ACM Symposium on Cloud Computing (SOCC 2010)*, Indianapolis, Indiana, June 2010.

[10] A. Abouzeid, K. BajdaPawlikowski, D. Abadi, A. Silberschatz, A. Rasin,"HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads," in *VLDB 2009*, Lyon,France, August 2009, pp.922–933.

[11] HBaseReport. Available at http://cloudepr.blogspot.com/.

[12] DataGenerate. Available at http://database.cs.brown.edu/projects/mapreduce-vs-dbms/.

[13] Cassandra UseCase. Available at http://wiki.apache.org/cassandra/UseCases.

[14] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *Proceedings of SOSP'03*, New York, USA, December 2003, pp. 29–43.

[15] Hadoop. [Online]. Available: http://hadoop.apache.org

[16] HDFS. Available at http://hadoop.apache.org/hdfs/.

[17] KFS. Available at http://kosmosfs.sourceforge.net/.

[18] Apache Cassandra Glossary. Available at http://io.typepad.com/glossary.html.

[19] Cassandra FAQ. Available at http://wiki.apache.org/cassandra/FAQ.

ESQP: An Efficient SQL Query Processing for Cloud Data Management

Jing Zhao, Xiangmei Hu and Xiaofeng Meng
School of Information, Renmin University of China
Beijing, China, 100872
{zhaoj, hxm2008, xfmeng}@ruc.edu.cn

ABSTRACT

Recently, the cloud computing platform is getting more and more attentions as a new trend of data management. Currently there are several cloud computing products that can provide various services. However, most cloud platforms are not designed for structured data management. So they rarely support SQL queries directly. Even though some platforms support SQL queries, their bottoms are traditional relational database, therefore, the cost for executing a subquery in RDBS may influence the overall query performance. How to improve query efficiency in cloud data management system, especially query on structured data has become a more and more important problem. To address the issue, an efficient algorithm about query processing on structured data is proposed. Our approach is inspired by the idea of MapReduce, in which a job is divided into several tasks. Based on the distributed storage of one table, this algorithm divides a user query into different subqueries, at the same time, with replicas in cloud, a subquery is mapped to k+1 subqueries. Every subquery has to wait in the queue of the slave where the query data store. To balance the load, our algorithm also takes two scheduling strategies to dispatch the subquery. Besides, in order to reduce the client's long waiting time, we adopt the pipeline strategy to process result returning. Finally, we demonstrate the efficiency and scalability of our algorithm with kinds of experiments. Our approach is quite general and independent from the underlying infrastructure and can be easily carried over for implementation on various cloud computing platforms.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Query processing*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms

Algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CloudDB'10, October 30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0380-4/10/10 ...$10.00.

Keywords

distributed query, query processing, query transformation

1. INTRODUCTION

With the rapid growth of the amount of data, how to manage massive information becomes a challenging problem. It changes the infrastructure of data storage and generates a new technology called cloud computing. Existing cloud computing systems include Amazon's Elastic Computing Cloud(EC2)[1], IBM's Blue Cloud[2] and Google's GFS[5]. They adopt flexible resources management mechanism and provide good scalability. There are also some open source cloud computing projects, such as Apache Hadoop project's HDFS[9] and HBase[8], which are the open source implementation of Google's GFS and BigTable[4], and Cassandra[7], which brings together Dynamo's[6] fully distributed design and Bigtable's ColumnFamily-based data model.

Although Google's BigTable[4] stores data with table structure, column based storage model and timestamps are designed to improve the flexibility of one record. In other words, column based storage model is just more applicable for unstructured and semi-structured data storage, but not for structured data.

On the other hand, Because the popularity of traditional RDBMS and data warehouse, currently the analytical data of most enterprisers used in business planning, problem solving, and decision support are structured data. Analytical data has its special characteristics: ACID guarantees are typically not needed; Particularly sensitive data can often be left out of the analysis; shared-nothing architecture is a good match for analytical data management. These characteristics of the data and workloads of typical analytical data management applications are well-suited for cloud deployment[14]. At the same time, these analysis data is close to even more than PB level[10]. How to query and analyze on these data is a challenge problem.

Cloud computing platforms contain hundreds and thousands of heterogeneous commodity hardware, and they process workloads and tasks in parallel. This is a typical characteristic of cloud computing infrastructures. When a user submits a query, master nodes in the cluster must decompose the query into subqueries, dispatch them to slave nodes for concurrently processing and merge the results returned from slave nodes. The results of the query can not returned to users until all subqueries on slave nodes execute completely. However, in cloud computing platforms, slave nodes always can not finish subqueries at the same time, some of them may execute more quickly while some may be slower. There-

fore, we have to consider load balance to help us query more efficiently rather than dispatch the same amount of subqueries to each slave node. Meanwhile, as we know, cloud computing platforms always have k replicas of data for fault tolerance, which can also be used for efficient query answering. k replicas of data means that $k+1$ equivalent subqueries on the same data partition can be generated, and we can dynamically assign one of them to the appropriate slave node according to the load of system. In other words, replicas in cloud computing platform contribute to efficient subquery scheduling, which is very essential for query response.

In Summary, this paper makes the following contributions:

- Inspired by the idea of MapReduce, we propose a new efficient SQL query processing algorithm (ESQP) using data replicas in cloud storage.
- We describe how to decompose queries into subqueries according to query operator/oprande pair, which can run in parallel.
- We propose two scheduling algorithms in query procedure to achieve load balancing, and then improve query process efficiency.
- In order to reduce response time of the query, a pipeline strategy is employed when results return.
- We perform a series of experiments on large scale of machine nodes with large volume of data. The experiment confirms that our algorithm is efficient and scalable.

The rest of this paper is organized as follows: Section 2 and Section 3 describe related works and the current query on cloud computing separately. Section 4 presents our efficient SQL query processing algorithms for cloud data management, including query transformation, subquery scheduling and execution, and result return. In Section 5, we present the experimental results to demonstrate the efficiency and scalability of our methods. Finally, we make a conclusion and discuss some possible future work in Section 6.

2. RELATED WORK

2.1 Distributed Query Processing

Query processing problem is a difficult and extensive problem in distributed environments. There are many important aspects of this problem, including query decomposition, data localization, global and local optimization, etc. A detail discussion of each aspect is out of the scope of this paper, and what we want to discuss here is the cost of query processing. As we all know, *total cost* [16] is a good measure of resource consumption. And the total cost includes CPU, I/O, and communication costs in distributed database system. As network becomes faster and faster, the communication cost does not dominate local processing cost. Therefore, many researches consider a weighted combination of these three cost components rather than communication cost merely. Cloud database systems share many properties of distributed and parallel database systems, but scale well into hundreds or thousands of nodes. Although a typical cluster connect large scale of nodes via a high-bandwidth network, the communication cost is quite important due to the huge size of dataset. Data in cloud data management system is always uniformly distributed, so that the larger dataset is, the more communication cost may be produced, especially in join query. Therefore, one of our aim is to minimize communication cost at run time by exploiting the replicated data.

2.2 MapReduce

Google's MapReduce programming model mainly focuses on supporting distributed solution for web-scale data processing[3]. It decomposes data processing into two functions: *map* functions, reading an input key-value pairs and outputting intermediate key-value pairs; *reduce* functions, which merges the intermediate pairs with the same key into the final output. All *map* and *reduce* operations can be performed in parallel by partitioning the input dataset and handling different partitions concurrently by cluster.

This model provides good load balancing, fault tolerance and low communication cost. In order to achieve dynamic load balancing, TaskTrackers are assigned tasks as soon as they finish them. As communication cost component is probably the most important factor considered in distributed query, so that Master schedules *map* tasks on the machine that contains a replica of the corresponding input data. Furthermore, MapReduce programming model spawn backup tasks for the tasks run on slow workers to shorten job completion time and reexecute completed or in-progress *map* tasks and in-progress *reduce* tasks to ensure fault tolerance of workers.

However, this model has its own limitations. Users have to translate their applications into *map* and *reduce* tasks to achieve parallelism. Due to the commonality of this model, it takes sorting as the necessary step before *reduce* function. But this translation and sorting is really unnecessary for some simple SQL operations such as selection and projection. Furthermore, as indicated in [17], complex applications such as join, which requires extra stages of *map* and *reduce*, does not quite fit into this model. The implementation of *map* and *reduce* functions, especially the strategies of functions optimization would get users into trouble.

So we try to employ the basic idea of MapReduce programming model, including partition, single task re-execution, scalability and fault tolerance. We adopts the strategy of this model by decomposing a SQL query into multiple subqueries according to the corresponding data replicas. Meanwhile, we take advantages of techniques of traditional DBMS and parallel database system.

2.3 MapReduce and SQL

There are some work on combining ideas of MapReduce with database system. Typical examples include Apache's Hive[12], Yale's HadoopDB[11], Microsoft's SCOPE[13], etc. However, some of these work focus on system hybrid, while others focus on the SQL-like interface. HadoopDB[11] provides a hybrid solution at system level, using MapReduce framework for query distribution, inheriting the scheduling from Hadoop for fault tolerance and coordination ability, and take PostgreSQL servers as database engine for query processing. SCOPE and Hive separately provides a kind of SQL-like language. They integrate SQL-like language into MapReduce-like software to increase user productivity and system efficiency.

We do not use any database system for query processing but we employ some key techniques, including index and

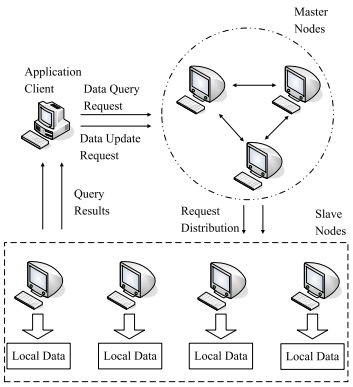


Figure 1: Framework of Query Processing in Cloud

pipeline, to improve the efficiency of subquery processing. Moreover, although we employ the basic idea of MapReduce, we design a structure for query distribution and processing, which does not base on or combine with Hadoop, so that we can take control of the whole progress of query processing and ESQP can be easily carried over for implementations on various of cloud computing platforms

3. QUERY IN THE CLOUD

As we know, a cloud computing platform(a cluster) consisting of hundreds or thousands of PC is responsible for data computing and storage. As Figure 1 shows, there are two types of nodes in the cluster: master nodes and slave nodes. Master nodes store some meta data about the whole cluster while slave nodes store the regular data. In other words, slave nodes store data records and their replicas for security. So the query on the cloud platform is different from central or parallel database. In the cloud platform, client query is often presented against the master nodes. After that the mater nodes decide which slave nodes are relevant to the query and then the query is passed to the slave nodes to do the query processing directly. The general query processing in cloud computing platform is in Figure 1. So a typical query in the cloud computing platform can be divided into two phases: locate the slave nodes which stores the relevant data and process query on the slave nodes directly. The procedure cloud be expressed as algorithm 1.

Algorithm 1 Process query on cloud

```
1: procedure SET PROCESSQUERY(Query q)
2:     Set nodes = empty;
3:     nodes.add(getRelativeNodes(q));
4:     Set results = empty;
5:     for (each node n in the nodes) do
6:         results.add(n.retrieveRecords(q));
7:     end for
8:     returnresults;
9: end procedure
```

From the above discussion,we can see that:

- The query processing problem is much more difficult in cloud computing environments than in centralized ones, because the query processing is not complete by one machine.
- The huge scale of cluster leads query processing in cloud environment problem be different from in parallel ones.
- Most of cloud computing systems decide which replica of data to be used for query before query processing. These predefined replica may result in more cost in some cases.

In order to query efficiently, we have to improve query processing by some means. In the following part of the paper, we will discuss how to query more efficiently on the cloud platform. The details will be listed below.

4. QUERY PROCESSING

As we mentioned earlier, the key problems of structured query processing in cloud database system lie in structured query translation, load balance of the whole system and data transfer among nodes. In order to query efficiently, our approach employs four key ideas:

- We exploit replicas in cloud database system for query translation in order to provide better alternatives for scheduler.
- A scheduler and scheduling metric are developed to ensure load balancing and reduce the total runtime of each query.
- We adopt DigestJoin [15] to reduce the size of data that has to be transferred in join operation.
- In order to avoid client's longtime waiting, pipeline and ASAP are employed in subquery processing model.

In the remainder of this section, we discuss the details of our design. First, we describe the data and query model the algorithm is optimized for, and then present the execution of query, including query transformation, subquery scheduling and execution, and result returning. Finally, we focus on transformation of query for a number of relational operators.

4.1 Data and query model

Due to our method in general supporting common SQL query in cloud computing system, the data need to be distributively stored in cloud system. Generally, a table is a collection of records, each of which is identified by a unique key, and each table is divided into n parts, each part replicated k times and are stored in different nodes in cluster. k is usually much smaller than the number of nodes in cluster while $k = 2$ for most cloud computing system. The meta information, such as the storage information about each replica of each partition. These information is reported to master nodes in cluster, which are in charge of subqueries scheduling. Typical cloud computing systems usually provide good support for key/value based queries, therefore, we assume that the data in cloud computing system has an index in key field, based on which we provide an efficient join processing method.

We focus on providing low latency for read-only SQL query, including generalized selection, projection, aggregation and join. Generalized selection means retrieving not only a single record by primary key but also a number of records satisfied any condition in any fields of a table. All these operations need to scan all data without index. Therefore, low latency means that the first record of query results should return as soon as possible to avoid client's longtime waiting.

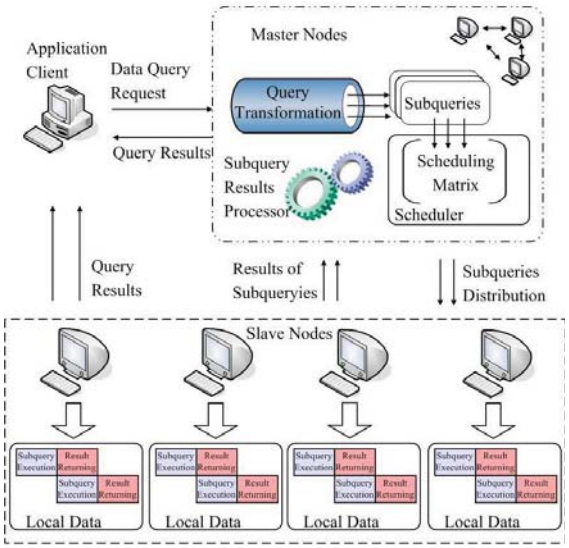


Figure 2: Execution Overview

4.2 Query execution

A key goal of our processing algorithm is to minimize the response time of a query in cloud database system. As figure 1 shows, in the cloud platform, master nodes store some meta data about the whole system and are in charge of distributing query to coordinate slave nodes. When a slave node receives the request from master nodes, it retrieves data locally or communicate with other slave nodes for relevant data according to the operation type and stores results locally. Results are returned to client directly after result generating.

From this progress we can see that the key components of query processing which influence latency are:

- Query transformation: A user query should be transformed into a set of independent subqueries that can be execute parallelly on nodes of cluster. Parallelism can significantly reduce query latency in all. Local execution is another important aspect for low latency, so that we try to make sure that the percentage of subqueries that could be executed locally as large as possible.

- Query dispatch: Assignment of subqueries plays an important role in query processing. System can achieve load banlancing via good and reasonable scheduling of subqueries, and then minimize the total runtime of the query.

- Subquery execution: Slave nodes employ the idea of pipeline to accelerate a number of subqueries processing rather than repeat the following three steps, receive subquery, process subquery and return result serially, parallel execution of previous result returning and current query processing can save much time in that we don't need to wait for results transfer.

We present the details of execution in the following. Figure 2 shows this in diagram form.

4.2.1 Query transformation

Each user query is transformed into a set of subqueries according to the partition of involved tables, each of which can be executed independently. There are kinds of SQL operators, which lead to various transformations. But all kinds of transformations are based on the partitions of tables and their replicas. We consider a user query as an *operator/operand* pair. *Operator* includes generalized selection, projection, aggregation and join, and *operand* here means the data blocks of table where operators retrieve from. Because the operator is constant to the specific SQL operation, we generate subqueries by modifying the operands of the original query. The operand can be classified into two categories: single table for the first three operators and multiple tables for join operator. We maintain a list of subqueries, each of which has two kinds of transformed operands set for multiple tables, while one set for single table. The procedure could be expressed as algorithm 2, and now we present the transformation of operands in detail:

Single Table: Operands of the first three kinds of operators are single table. We simply replace the original operand, a single table in FROM clause, with a number of location sets of replicas to create the subqueries. Each set contains all copies of a partition in one table. For instance, table R is divided into m parts R_1, R_2, ..., and R_m with a backup factor $k = 2$, hence we create m sets, each of which is composed of R_i $(i = 1, 2, ..., m)$, R_{i_1}, R_{i_1}, ..., R_{i_k}, where R_{i_j} is the copy of R_i. These subqueries can be run in parallel, locally and independently.

Multiple Tables: We consider two tables here because operation on multiple tables can be split into a set of operations on two tables. The transformed operands are classified into two kinds: one is partitions of tables without intersection and the other is partitions of tables with intersection, which implies that there is a slave machine that store two replicas of blocks belonging to different tables. Therefore, we create three sets for a subquery to store the location information of two blocks, one of which shows the location of replicas from two blocks that stored in the same slave node, called *intersectionset*, and the other two separately express the location information of replicas from different tables, called *replicaset*. It is necessary to state that, employing the basic idea of DigestJoin[15], the operator on *intersectionset* operand is original joining, while the operator of subquery without intersectant replicas includes not only JOIN but also EXTRACT and RELOAD, where EXTRACT means extracting digest data from nearest node before JOIN operation and RELOAD refers to reloading relevant data to compose query result after JOIN operation.

For a cluster of n slaves, take table R joins table S for example. As our data model stated above, supposing R_i is a part of R and S_j is a part of S, then we have replicas R_{i_1}, R_{i_2},..., R_{i_k} for R_i and S_{j_1}, S_{j_2}, ..., S_{j_k} for S_j. We suppose that R_i, R_{i_1}, R_{i_2},..., R_{i_k} are stored in $slave_i$, $slave_{i_1}$, $slave_{i_2}$,..., $slave_{i_k}$ and S_i, S_{i_1}, S_{i_2},..., S_{i_k} are stored in $slave_j$, $slave_{j_1}$, $slave_{j_2}$,..., $slave_{j_k}$. We declare that R_i intersect S_j, if and only if there is any $i_s == j_t$, where $s, t = 1, 2, ..., k$. We store i_s in intersection set when intersection occurred and set the other sets into empty, while assigning i, i_1, i_2,..., i_k to one replica set, j, j_1, j_2,..., j_k to anther and intersection set is set to empty when there is no intersection. Figure 3 shows how to maintain these informa-

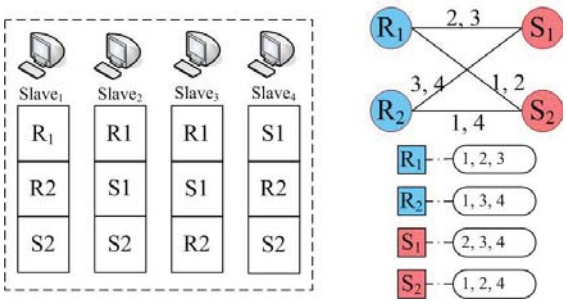


Figure 3: Totally 3 copies of partitions R_1, R_2, S_1 and S_2 are separately stored in four slave nodes. The edges of bipartite graph shows the intersection information of R_i and S_j, and lists in the the corner express location of replicas of each partition, each of which is assigned to coordinate subquery when intersection set is empty.

tion for partitions R_1, R_2, S_1 and S_2 with replica factor of $k = 2$.

Algorithm 2 Query Transformation

```
1: procedure SET TRANSFORMQUERY(Query, q)
2:     Set subqueries = empty;
3:     if (q.type is JOIN) then
4:         Set partsA = getPartitionsOfTable(q.tableA);
5:         Set partsB = getPartitionsOfTable(q.tableB);
6:         for (each part p in partsA) do
7:             Set replicasA = getReplicasOfPart(p);
8:             for (each part p in partsB) do
9:                 Set intersection = empty;
10:                Set locationsA = empty;
11:                Set locationsB = empty;
12:                Set replicasB = getReplicasOfPart(p);
13:                if (replicasA and replicasB intersect) then
14:                    intersection.add(location of intersected
    replicas);
15:                else
16:                    locationsA = replicasA;
17:                    locationsB = replicasB;
18:                end if
19:                subqueries.add(intersection,locationsA,
    locationsB);
20:            end for
21:        end for
22:    else
23:        Set parts = getPartitionsOfTable(q.table);
24:        for (each part p in parts) do
25:            Set blocks = getReplicasOfPart(p);
26:            subqueries.add(blocks);
27:        end for
28:    end if
29:    return subqueries;
30: end procedure
```

4.2.2 Subquery Scheduling

While the subqueries can be executed in parallel, according to the expression above, the number of subqueries is equivalent to the number of table's partitions or the product of numbers of two tables' parts, which far exceeds the number of nodes in cloud platform, and different performances of machines in the cluster lead to heterogeneous load, so we develop a scheduler and scheduling matrix to coordinate the

$$\begin{array}{c} \\ slave_1 \\ slave_2 \\ slave_3 \\ slave_4 \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \begin{array}{c} L_i \\ 2 \\ 1 \Rightarrow \\ 3 \\ 2 \end{array} \begin{array}{c} (L_i-a)(L_i-c) \\ 1 \\ 1 \\ 2 \\ 1 \end{array} \begin{array}{c} \\ 1 \\ 1 \\ 3 \\ 1 \end{array}$$

Figure 4: A constructed matrix with 4 slave nodes and 4 subqueries to be scheduled. Parameter L_i of each slave node represents the number of subqueries waiting for slave node $slave_i$.

execution of subqueries, which dynamically changes loads on slave nodes to minimize the response time of the query.

In general, each slave only executee subqueries on replicas of parts that it is stored locally, particularly for JOIN operator, a subquery is dispatched to the slave store one of its operands. And every subquery is composed of a operator and a operand set which contains location information of subquery, in other words, we regard a subquery as a set of at least $k + 1$ equivalent subqueries. Therefore, we exploit a scheduling matrix to decide which subqueries are given to a slave node. We take subqueries as horizontal axis and slave nodes as vertical axis. The element of this matrix are numbers in a union $\{0, 1, 2\}$, where $M_{ij} = 0$ means that the partition where subquery $subQ_j$ retrieve does not have any copies stored in slave node $slave_i$, $M_{ij} = 1$ expresses that one of the copies of subquery $subQ_j$'s partition is stored in $slave_i$ or both retrieved tables' partition have copies in $slave_i$, and $M_{ij} = 2$ implies subquery's original operator is JOIN and only one table involved has copies stored in $slave_i$. In the other word, we consider a row of the matrix as an unordered of subqueries which are waiting for dispatching. Figure 4 shows the scheduling matrix for a cluster consisted of 4 slave nodes, with a backup factor $k = 1$. In this figure, each slave node has a parameter representing the number of subqueries that are waiting for execution on a certain node.

The greedy scheduler grants a subquery as soon as possible to a slave node when it becomes free. The system achieves load balancing effectively through such an approach because fast slave nodes can take on more workload to lighten slower nodes. Moreover, scheduler creates a subquery list which consists of the operator, operand and the status of this query, including *waiting*, *processing*, *processed*, *gettingResult*, and *finished*, and a status list of slave nodes. It communicates with slave nodes according to these two lists and scheduling matrix through two message types. The scheduler sends a slave node a *dispatch* message, which notifies it to start processing subquery. As a subquery is assigned to slave node, the scheduler changes the status of it from *waiting* to *processing* and removes all equivalent subqueries from scheduling matrix. And when slave node has finished execution of current subquery, it returns a $free$ message to the scheduler, which will change the status of the subquery to *processed* and reset the slave node's state to *free*. The procedures are described as algorithm 3, 4 and 5.

There are kinds of scheduling algorithms. We have implemented two: **Random Scheduling.** Whenever a slave node becomes free, our scheduler randomly chooses a subquery from its waiting queue. **Global Scheduling.** We adopt the idea of global optimization. A subquery which balances all waiting queues is chosen. Before choosing a subquery, we compute the length of waiting queue for each slave

Algorithm 3 Initialize Scheduling Matrix

```
1: procedure MATRIX INITIALIZEMATRIX(ListsubQueries)
2:     Matrix M = empty;
3:     for (each subquery subq_i in subqueries list) do
4:         if (subq.operator is a unary operator) then
5:             for (each location loc in subq.operand.locs) do
6:                 M_loci = 1;
7:             end for
8:         else
9:             if (subq.operand.intersectionset is empty) then
10:                for (each loc in subq.operand.firstSet) do
11:                    M_loci = 2;
12:                end for
13:                for (each loc in subq.operand.secondSet) do
14:                    M_loci = 2;
15:                end for
16:            else
17:                for (each loc in subq.operand.intersection) do
18:                    M_loci = 1;
19:                end for
20:            end if
21:        end if
22:        subq.status = waiting;
23:    end for
24:    return M;
25: end procedure
```

Algorithm 4 Subquery scheduling

```
1: procedure BOOLEAN SCHEDULE(List subQueries)
2:     InitializeMatrix(subQueries);
3:     while (scheduling matrix != 0) do
4:         for (each free slave slave_i) do
5:             subq = chooseSubquery(slave_i);
6:             dispatch(subq,slave_i);
7:             for (each element in column set of subq) do
8:                 element=0;
9:             end for
10:            subq.status = processing;
11:            slave_i.status = busy;
12:        end for
13:    end while
14:    return true;
15: end procedure
```

Algorithm 5 Select a subquery

```
1: procedure SUBQUERY SELECT(int slaveLoc)
2:     SubQuery subq = empty;
3:     double variance = POSITIVE INFINITY;
4:     List length = QueueLength(Matrix M);
5:     for (each subquery q in waiting list) do
6:         generate length list lengths;
7:         if (variance > varianceOf(lengths)) then
8:             variance = varianceOf(lengths);
9:             subq = q;
10:        end if
11:    end for
12:    return subq;
13: end procedure
```

nodes by removing every possible subquery, which comes from the waiting queue of free slave node, thus we have l length lists, where l is the number of possible subqueries. The variance of each list is calculated and the subquery corresponding to the smallest variance is assigned to the slave node.

As is shown by Figure 4, L_i represents the number of subqueries waiting for distribution of each slave node. Supposing $slave_1$ is free, a and c are two probable subqueries in its waiting queue. Thus we separately pre-compute the length of each slave node's waiting queue removing a and c. **Random** would randomly assign a or c to $slave_1$, while **Global** would chose a for load balance of the system due to the variance of $L_i - a$ is 0.33, which is smaller than $L_i - c$'s variance 1.

4.2.3 Subquery Execution and results returning

When a slave node receives *dispatch* message sent by scheduler, it starts the execution thread, storing results locally. Instead of returning results as quickly as it generates, the slave node sends the $free$ message back to scheduler to report that the subquery is completed, and the master node creates a result handling thread to get back the results asynchronously. The slave node processes subqueries in full sail rather than being distracted by transportation of results, thus the subquery execution on slave nodes seems a pipeline, returning of the results of previous subquery and current subquery's processing go simultaneously, which reduces the overall runtime of a query in a sense. The subquery execution and result returning procedures are as algorithm 6 and 7, and the implementation of algorithm 6 is invoked by slave nodes repeatedly while the algorithm 7's implementation is called by scheduler.

Algorithm 6 Subquery Processing

```
1: procedure VOID PROCESSSUBQUERY(SubQuery subq)
2:     Result results = getResult(subq);
3:     String fileName = storeResults(results);
4:     send free message back to scheduler;
5: end procedure
```

Algorithm 7 Result Returning

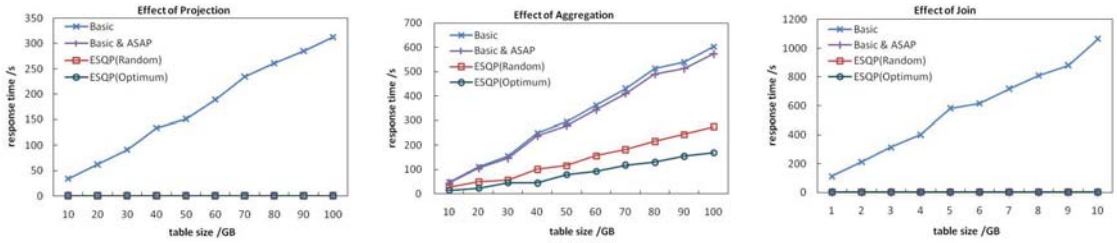
```
1: procedure RESULTS GETRESULTS(SubQuery subq, String loc)
2:     subq.status = gettingResult;
3:     Results middleR = fetchResult(subq,loc);
4:     return middleR;
5: end procedure
```

4.2.4 Result Representation

In order to minimize the *response time* of the query, the main idea of our approach is that returning the results to clients as soon as possible, even if only one record of all results is ready. Some of results could be returned to users once the result processor get them, for example, the results of generalized selection, projection and joins, but in some cases, the results of subqueries are not the just result of original query, so that some retreatments are required, as aggregation. The combination involves *order* and *aggregate*:

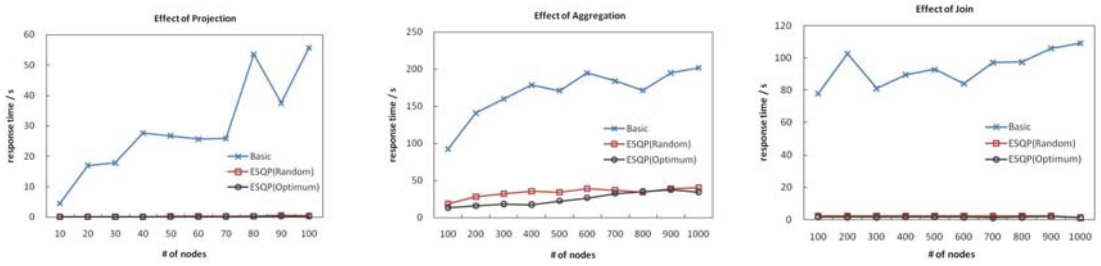
Order: Although the results of the subqueries are well sorted locally on slave nodes, a global sorting must be carried out to form an ordered result of original query.

Aggregate: J. Gray et al.[18] classified aggregate function F() into three categories: *Distributive,Algebraic* and *Holistic*. We only consider first two types in SQL aggregate operators. These aggregate functions are equivalent to aggregation of original functions, such as *COUNT()*, *MIN()*, *MAX()*, *SUM()*, or combination of additional functions, for instance *AVERAGE()*. The result processor compute the aggregate result of query according to different aggregate functions. Take *MIN()* and *AVERAGE()* for example, result processor take the minimum of values fetched from slave



(a) Performance of projection (b) Performance of aggregation (c) Performance of join

Figure 5: Query response time for different queries by scaling up the size of table



(a) Performance of projection (b) Performance of aggregation (c) Performance of join

Figure 6: Query response time for different queries by scaling up the number of slave nodes

nodes to find the final minimum value, while slave nodes report *SUM()* and *COUNT()* of subset for *AVERAGE()* function and result processor adds these two components and then divides to produce the global average.

4.3 Fault Tolerance

Inspired by the approach taken by MapReduce[3], the fault tolerance strategy is to restart all subqueries which are marked as *processing*, *processed*, or *gettingResult* when the corresponding slave node is out of contact. Although our algorithm can deal with the failure of slave nodes, we will take this as a future work for lack of space.

5. PERFORMANCE EVALUATIONS

We now evaluate the performance and scalability of our query processing in cloud databases. Testing for query processing breaks down into two suites: efficiency and scalability tests, to demonstrate the effect and scalability of of our query processing technique, and load balancing tests, to test our subquery execution scheduling.

5.1 Experiment Setup

Our testing infrastructure includes 11 machines which are connected together to simulate cloud computing platforms - 1 master and 10 slaves. Each contains a Inter Core 2 2.33GHz CPU, 8GB of main memory and 2TB hard disk. Machines ran Ubuntu 9.10 Server OS. Communication bandwidth was 1Gbps.

We use this infrastructure to simulate different size of cloud computing systems. We conducted 10 simulation experiments, ranging from 100 nodes to 1000 nodes. Each time 100 more nodes are considered to be added into the cloud computing system. Our algorithm is implemented in Java. We use a small telecom CDR data sample to generate 100 GB of data with about 200 bytes per tuple. These data are used as 10 different sets, ranging from 10 GB to 100 GB with 10 GB increment. Three typical queries are hired to testify our algorithm's efficiency and scalability, including projection, aggregation and join requirements. And for join query the dataset is a little different from above dataset. We use data from 1 GB to 10 GB as a table and join two tables with the same size.

5.2 Performance of typical queries

We design two sets of experiments to evaluate the performance of the query processing of three typical queries. For each query, we separately scale up the size of data, which indicates the total number of subqueries of one original query due to the fixed size of one data block in the system, and the number of slave nodes in the cluster. First of all, for a fixed 50 nodes cluster, we increase the size of data, and then for a fixed 10 GB data, we scale up the number of slave nodes. Response time, which is the interval between the query started and the first result of the query returned to the user, is used as the metric in the experiments. Respectively, we use four methods to execute each typical query, including basic method, ASAP based basic method, random optimum scheduling method and global optimum scheduling method. Basic method decomposes query into multiple certain subqueries rather than $k + 1$ equivalent subqueries for each table partition, and it is marked *busy* until all results are returned to master nodes. And finally the master nodes return the result of original query after all subqueries execution finished, while ASAP based basic method return results as soon sa possible. Optimal methods not only take the advantage of table's multiple backups, but also employ ASAP approach, pipeline and scheduler to reduce the response time as best as we can. All results are obtained based on 5 runs.

Figure 5 and 6 show our results straightforwardly. Results show very good performance. Random and Global ESQP have similar performance in our dataset in that our data is distributed uniformly. As can be seen in figure 5, the cost of optimal method answering the projection and join query in 50 nodes and 100 GB only is less than 1 second, and aggregation query is only 100-300 seconds, which shows that our method is very efficient. ASAP based basic method also has good performance in figure 5(a) and 5(c) because the response time is determined by query decomposition and the fastest subquery execution, where our method has no obvious superiority. But according to our experimental records, the total cost of a query execution in this method is much more than ESQP method.

Figure 5 and 6 also illustrate the scalability of our methods. These graphs show that our distributed efficient SQL query processing method scales almost linearly with the table size or the number of nodes. Benefiting from pipeline strategy, when the queries don't have aggregation, we return the result of query as soon as we get it from subquery execution node, and we stop the mission timer at the point that first result is received by client. Therefore, the response time of this kind query is only influenced by query decomposition, which is always dominated by subqueries dispatching time. On the other hand, the response time of aggregation query is consist of query decomposition time, query dispatch time and time of the lowest subquery's execution and result return. Although the scale up of the cluster makes nonunhiform distribution of subqueries in basic method that leads to the load unbalance problem, which causes the relevant curves in figure 6 are not smooth or monotonic, the figure shows our method is high available and scale to hundreds of nodes, and figure 5 shows the performance of our method is very good when data size is scaling up.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a newly more efficient query algorithm to deal with SQL query. According to different kinds of queries, we adopted different subqueries dispatch. Besides, the algorithm took advantage of the idea of divide and conquer. In order to get higher efficiency, we not only used scheduling algorithms to get load balance, but also we utilized pipeline technique to process result return. Finally, we proved the efficiency and scalability of our approach with vast experiments.

For future work, as the number of slave nodes increases, although our query processing algorithm has very good scalability, the query cost does not reduce lineally because of the computation of the large matrix. Therefore, we will study more advanced query schedulers to make our algorithm more scalable, and do more research on large-scale concurrent queries answering which brings further scalability problems. Besides, currently our scheduling algorithms choose subquery according to a heuristic method-variance to achieve maximum load balancing. So we also plan to research a more accurate measure of load balance.

7. ACKNOWLEDGMENTS

The authors thank anonymous reviewers for their constructive comments. This research was partially supported by the grants from the Natural Science Foundation of China (No.60833005); the National High-Tech Research and Development Plan of China (No.2007AA01Z155čň2009AA011904); and the Doctoral Fund of Ministry of Education of China (No. 200800020002).

8. REFERENCES

[1] M. Lynch. Amazon elastic compute cloud (amazon ec2). [Online]. Available: http://aws.amazon.com/ec2/

[2] IBM. Ibm introduces ready-to-use cloud computing. [Online]. Available: http://www-03.ibm.com/press/us/en/pressrelease/22613.wss

[3] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, pp. 107–113, January 2008.

[4] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, "Bigtable: A distributed storage system for structured data," in *Proceedings of the 7th Conference on USENIX Symposium on Operating Systems Design and Implementation*, Seattle, Washington, November 2006, pp. 205–218.

[5] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *Proceedings of SOSP'03*, New York, USA, December 2003, pp. 29–43.

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazons highly available key-value store," in *Proceedings of the 21st ACM Symposium on Operating Systems Principles(SOSP '07)*, Washington, USA, October 2007, pp. 205–220.

[7] Cassandra. [Online]. Available: http://incubator.apache.org/cassandra/

[8] HBase. [Online]. Available: http://hadoop.apache.org/hbase

[9] Hadoop. [Online]. Available: http://hadoop.apache.org

[10] C. Monash. The 1-petabyte barrier is crumbling. [Online]. Available: http://www.networkworld.com/community/node/31439

[11] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz and A. Rasin, "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads," in *Proceedings of VLDB'09*, Lyon, France, August 2009, pp. 922–933.

[12] A. Thusoo, J. Sen Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff and R. Murthy, "Hive: a warehousing solution over a map-reduce framework," in *Proceedings of VLDB'09*, Lyon, France, August 2009, pp. 922–933.

[13] R. Chaiken, B. Jenkins, P. Larson, B. Ramsey, D. Shakib, S. Weaver and J. Zhou, "SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets," in *Proceedings of PVLDB'08*, Auckland, New Zealand, August 2008.

[14] D. J. Abadi, "Data Management in the Cloud: Limitations and Opportunities," in *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2009, pp. 3–12.

[15] Y. Li, S. T. On, J. Xu, B. Choi, and H. Hu, "DigestJoin: Exploiting Fast Random Reads for Flash-based Joins," in *Proceedings of the 10th International Conference on Mobile Data Management (MDM '09)*, Taipei, Taiwan, May 2009, pp. 152–161.

[16] M. S. Sacco and S. B. Yao, "Query Optimization in Distributed Data Base Systems," in *Adcances in Computers*, New York, United States, 1982, pp. 225–273.

[17] R. Pike, S. Dorward, R. Griesemer and S. Quinlan, "Interpreting the Data: Parallel Analysis with Sawzall," in *Scientific Programming Journal*, 2005, pp. 227–298.

[18] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow and H. pirahesh, "Data cube: A relational aggregation operator generalizing group-by, cross-tabl, and sub-totals," in *J. Data Mining and Knowledge Discovery*, 1997, vol. 1, pp. 29–53.

Report on the First International Workshop on Cloud Data Management (CloudDB 2009)

Xiaofeng Meng[1] Jiaheng Lu[1] Jie Qiu[2] Ying Chen[2] Haixun Wang[3]
{xfmeng,jiahenglu}@ruc.edu.cn, {qiujie, yingch}@cn.ibm.com, haixunw@microsoft.com
[1]School of Information and DEKE, MOE, Renmin University of China, Beijing China
[2]IBM Research - China
[3]Microsoft Research Asia, Beijing

1. INTRODUCTION

The first ACM international workshop on cloud data management was held in Hong Kong, China on November 6, 2009 and co-located with the ACM 18th Conference on Information and Knowledge Management (CIKM). The main objective of the workshop was to address the challenge of large data management based on cloud computing infrastructure. The workshop brings together researchers and practitioners in cloud computing and data-intensive system design, programming, parallel algorithms, data management, scientific applications and information-based applications interested in maximizing performance, reducing cost and enlarging the scale of their endeavors.

The workshop attracted 11 submissions from Asia, Canada, Europe and the United States, out of which the program committee finally accepted 5 full papers and 3 short papers. The accepted papers focused on cloud-based indexing and query processing, cloud platform availability, cloud replication and system development.

2. KEYNOTE PRESENTATION

The keynote speech, titled "EMC Decho: A Real World Use Case of Cloud Computing" was delivered by Bill Sun, Engineer Manager in EMC Center of Excellence China. He presented EMC's perspective about cloud computing, and shared some of their experience in building a reliable infrastructure to provide personal cloud services for millions of users. Bill Sun highlighted some of challenges they are facing in building robust and reliable infrastructures and described their potential remedies.

The first challenge is the increasing importance that personal information has to the eyes of the users. As a simple example, the first picture of one's newborn son is a digital artifact that a user is likely to want to preserve through one or several generations. The value of digital information to users has become such important that preserving one's digital data cannot be tied to a particular device or application. The information saved in these devices are more valuable than devices and should live much longer.

The second challenge is to search effectively in various devices. Current-day technologies for searching personal archives of digital data still have strong limitations. For example, how can you easily find your pictures in Las Vegas taken in July last year? How can you find a presentation you got from a colleague six months back about a particular project? Most of time, we have to organize the information by five "C": i.e. Context, Content, Calendar, Coordinates and Contacts. Effective management of such personal information with five "C" is a big challenge.

The major conclusion is that personal cloud is what they believe the most effective approach to address those challenges in personal information management, where all the information are saved in a secure well managed cloud storage system. Decho will work as the center or the hub to synchronize all users' information across multiple devices through the personal cloud, including PC, cell phone, or even NetBook.

3. RESEARCH PAPERS

The technical paper session consisted of eight presentations, whose main points are summarized next. Together, they give a glimpse to the exciting new developments spurred by data management in the cloud. These papers cover a variety of topics. We believe that these papers will provide researchers and developers with a brief glimpse into this exciting new technology, specifically from the perspective of cloud data management.

The paper entitled *Personalization as a Service: Architecture and Case Study* focuses on how to provide personalized services for individual users in the cloud environment. H. Guo, J. Chen, W. Wu and W. Wang first analyzed the main issues and challenges of using the traditional server-side user profiles for personalized services in the cloud. Then they presented the architecture of Personalization as a Service (PaaS) in which the client-side user modeling method is employed to support personalized cloud services. The

main idea is to decouple user modeling components from cloud services by observing the user's interactions on all of their cloud client devices collectively. As a result, user model can be shared across cloud services and used in a pay-as-you-go way. The client-side user modeling avoids the server overhead and provides unique user experiences with minimal user intervention. They finally give a case study of a personalized cloud search service solution according to the PaaS architecture.

X. Zhang, J. Ai, Z. Wang, J. Lu and X. Meng proposed an efficient approach to build a multi-dimensional index for cloud computing systems in the paper *"An Efficient Multi-Dimensional Index for Cloud Data Management"*. Their approaches can process typical multi-dimensional queries including point queries and range queries efficiently. Besides, frequent change of data on big amount of machines makes the index maintenance a challenging problem. To cope with this problem they proposed a cost estimation-based index update strategy that can effectively update the index structure. They describe experiments showing that their indexing techniques improve query efficiency by an order of magnitude compared with alternative approaches, and scale well with the size of the data. Their approach is quite general and independent from the underlying infrastructure and can be easily carried over for implementation on various cloud computing platforms.

The topic considered in *Packing the Most Onto Your Cloud* by A. Aboulnaga, Z. Wang and Z. Zhang is one particular optimization problem, namely scheduling sets of Map-Reduce jobs on a cluster of machines (a computing cloud). They present a scheduler that takes job characteristics into account and finds a schedule that minimizes the total completion time of the set of jobs. Their scheduler decides on the number of cluster nodes to assign to each job, and it tries to pack as many jobs on the machines as the machine resources can support. To enable flexible scheduling and packing of jobs onto machines, they run the Map-Reduce jobs in virtual machines, although their scheduling approach can be applied in any Map-Reduce scheduler. Their scheduling problem is formulated as a constrained optimization problem, and they experimentally demonstrate using the Hadoop open source Map-Reduce implementation that the solution to this problem results in benefits up to 30%.

Query Processing of Massive Trajectory Data based on MapReduce is addressed by Q. Ma, B. Yang, W. Qian and A. Zhou. Traditional trajectory data partitioning, indexing, and query processing technologies are extended so that they may fully utilize the highly parallel processing power of large-scale clusters. They also showed that the append-only scheme of MapReduce storage model can be a nice base for handling updates of moving objects. Preliminary experiments show that this framework scales well in terms of the size of trajectory data set. The limitation of traditional trajectory data processing techniques and their future research direction are also discussed.

The approach considered in *Leveraging a Scalable Row Store to Build a Distributed Text Index* by N. Li, J. Rao, E. Shekita and S. Tata is a distributed text index called HIndex, by judiciously exploiting the control layer of HBase, which is an open source implementation of Google's Bigtable. Such leverage enables them to inherit the good properties of availability, elasticity and load balancing in HBase. They also present the design, implementation, and a performance evaluation of HIndex.

F. Wang, J. Qiu, J. Yang, B. Dong, X. Li, and Y. Li proposed a metadata replication based solution to enable Hadoop high availability by removing single points of failures in Hadoop in the paper titled *Hadoop High Availability through Metadata Replication*. Single points of failures mean that the whole system becomes out of work due to the failure of critical nodes where only a single copy of data exists. The solution involves three major phases. In the initialization phase, each standby/slave node is registered to active/primary node and its initial metadata (such as version file and file system image) are caught up with those of active/primary node. In the replication phase, the runtime metadata (such as outstanding operations and lease states) for fail-over in future are replicated. Finally, in the fail-over phase, standby/new elected primary node takes over all communications. The solution presents several unique features for Hadoop, such as runtime configurable synchronization mode. The experiments demonstrate the feasibility and efficiency of their solution.

In the Paper entitled *How Replicated Data Management in the Cloud can benefit from a Data Grid Protocol - the Re:GRIDiT Approach*, L. Voicu and H. Schuldt developed, implemented and evaluated the Re:GRIDiT protocol for managing data in the grid. Re:GRIDiT provides support for concurrent access to replicated at different sites without any global component and supports the dynamic deployment of replicas. Since it has been designed independent from any underlying grid middle-ware, it can be seamlessly transferred to other environments like the cloud. They present the Re:GRIDiT protocol, show its applicability for cloud data management, and provide performance results of the evaluation of the protocol in realistic cloud settings.

The topic in *The Design of Distributed Real-time Video Analytic System* by T. Yu, B. Zhou, Q. Li, R. Liu, W. Wang and C. Chang is to propose a distributed scalable infrastructure VAP (Video Analytic Platform) for supporting real-time video stream analysis. In VAP, the application requirements are represented as a Directed Acyclic Graph (DAG), where nodes stand for video analysis computation modules and links show data flow and dependencies between nodes. VAP leverages UIMA (Unstructured Information Management Architecture) framework as the data flow control engine and multiple commodity databases as the storage and computation resources. The actual executions of video analysis computation modules have been pushed down into database engine to minimize the data movement cost.

4. CONCLUSION

CloudDB 2009 was the first CIKM-associated workshop addressing the challenges of large database services based on the cloud computing infrastructure. Whilst these emerging services have reduced the cost of data storage and delivery by several orders of magnitude, there is significant complexity involved in ensuring large data service can scale when one needs to ensure consistent and reliable operation under peak loads. Cloud-based environment has the technical requirement to manage data center virtualization, lower cost and boost reliability by consolidating systems on the cloud.

A first conclusion that can be drawn from this workshop is that the cloud systems should be geographically dispersed to reduce their vulnerability due to earthquakes and other catastrophes, which increase technical challenge on a great level of distributed data interpretability and mobility. Data interoperability is even more essential in the future as one component of a multi-faceted approach to many applications.

A final conclusion is that existing research works in the area of cloud-based data management are still somehow immature and significant room for progress exists. The works presented in the workshop mainly focused on adapting existing Grid and Map/Reduce techniques to the cloud environment. The participants agreed that many open challenges still remain such as cloud data security and the efficiency of query processing in the cloud. The participants also expressed interest in the organization of a conference dedicated to the issues raised by data management in the cloud.

5. ACKNOWLEDGEMENT

We would like to thank the program committee members, keynote speakers, authors and attendees, for making CloudDB 2009 a successful workshop. Jiaheng Lu was supported by 863 National High-Tech Research Plan of China (No: 2009AA01Z133). Finally, we also express our great appreciation for the support from Renmin University of China and IBM research-China.

Eleventh International Conference on Mobile Data Management

ACR: an Adaptive Cost-Aware Buffer Replacement Algorithm for Flash Storage Devices

Xian Tang, Xiaofeng Meng

School of Information, Renmin University of China
{txianz,xfmeng2006}@gmail.com

***Abstract*—Flash disks are being widely used as an important alternative to conventional magnetic disks, although accessed through the same interface by applications, their distinguished feature, i.e., different read and write cost in the aspects of time, makes it necessary to reconsider the design of existing replacement algorithms to leverage their performance potential.**

Different from existing flash-aware buffer replacement policies that focus on the *asymmetry* of read and write operations, we address the "*discrepancy*" of the *asymmetry* for different flash disks, which is the fact that exists for a long time, while has drawn little attention by researchers since most existing flash-aware buffer replacement polices are somewhat based on the assumption that the cost of read operation is neglectable compared with that of write operation. In fact, this is not true for current flash disks on the market.

We propose an adaptive cost-aware replacement policy (ACR) that uses three *cost*-based heuristics to select the victim page, thus can fairly make trade off between clean pages (their content remain unchanged) and dirty pages (their content is modified), and hence, can work well for different type of flash disks of large *discrepancy*. Further, in ACR, buffer pages are divided into clean list and dirty list, the newly entered pages will not be inserted at the MRU position of either list, but at some position in the middle, thus the once-requested pages can be flushed out from the buffer quickly and the frequently-requested pages can stay in buffer for a longer time. Such mechanism makes ACR adaptive to workloads of different access patterns. The experimental results on different traces and flash disks show that ACR not only adaptively tunes itself to workloads of different access patterns, but also works well for different kind of flash disks compared with existing methods.

I. Introduction

Though primarily designed for mobile devices due to its superiority such as low access latency, low energy consumption, light weight and shock resistance, flash-based storage devices have been steadily expanded into personal computer and enterprise server markets with ever increasing capacity of their storage and dropping of their price. In the past several years, the density of NAND flash memory increased twofold and this trend will continue until year 2012 [1]. Existing operating systems are already providing facilities to take advantage of flash disks (e.g., Solid State Drive) [2].

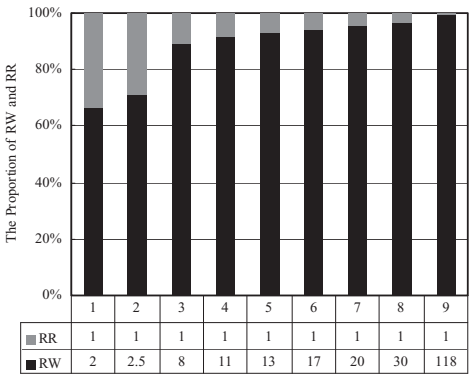
Typically, a flash disk managed by an operating system is a block device which provides the same interface type as a magnetic disk, however, their I/O characteristics are widely disparate. A flash disk usually demonstrates extremely fast random read speeds, but slow random write speeds, and the best attainable performance can hardly be obtained from database servers without elaborate flash-aware data structures and algorithms [3], which makes it necessary to reconsider the design of IO-intensive and performance-critical software to achieve maximized performance.

Buffer is one of the most fundamental component in modern computing. It is widely used in storage systems, databases, web servers, file systems, operating systems, etc. Any substantial progress in buffer replacement algorithms will affect the entire modern computational stack. Assuming that the secondary storage consists of magnetic disks and there is no difference for the time delay between read and write operations, the goal of existing buffer replacement policies [4]–[10] is to minimize the buffer miss ratio for a given buffer size. When the buffer is full and the current requested page is not in the buffer, the replacement policy has to select an in-buffer page as the victim, if the victim is a dirty page, it will be written back to disk before paging in the requested page so as to guarantee data consistency, which may be a performance bottleneck since the process or thread requesting for the requested page must wait until write completion. Early in two decades ago, [11] has realized the fact that whether a page is read only or modified is an important factor which will affect the performance of a replacement policy and should be considered in the replacement decision. As flash disks are becoming an important alternative to magnetic disks, this phenomena should be paid more attention than ever.

Considering the asymmetric read and write operation of flash disks, researchers have proposed flash-aware replacement algorithms [12]–[16] in the past yeas. Based on the assumption that the cost of random read operation is *neglectable* compared with that of random write operation, the fundamental idea behind these policies is reducing random write operations by firstly paging out clean pages arbitrarily no matter how frequently they are requested, which means that the cost of random write operations dominates the overall cost of a replacement policy. However, from Fig. 1, we can get an important observation that is not consistent with the above assumption: the cost of random read operation should *not* be neglected for all cases, since the time consumed by random write and read operation for different type of flash devices various largely. Though all flash devices demonstrate fast random read speeds and slow random write speeds, it is not difficult to see that paging out clean pages before dirty pages without considering their reference frequency is not reasonable for all cases.

978-0-7695-4048-1/10 $26.00 © 2010 IEEE
DOI 10.1109/MDM.2010.34

IEEE computer society



	1	2	3	4	5	6	7	8	9
RR	1	1	1	1	1	1	1	1	1
RW	2	2.5	8	11	13	17	20	30	118

Fig. 1: The normalized proportion of time consumed by random write (RW) and random read (RR) operations for NAND flash disks. The numbers on the X axis represent 9 flash disks, "1" is Samsung MCAQE32G8APP-0XA, "2" is Samsung K9WAG08U1A, "3" is Samsung K9XXG08UXM, "4" is Samsung K9F1208R0B, "5" is Samsung K9GAG08B0M, "6" is Hynix HY27SA1G1M, "7" is Samsung K9K1208U0A, "8" is Samsung K9F2808Q0B, "9" is Samsung MCAQE32G5APP [17].

Moreover, since the cost of write operations is more expensive than that of read operations, reducing write operations in many cases will improve the overall performance. However, for a sequence of write requests, the write operations cannot be further reduced by keeping *once-requested* dirty pages in the buffer for a long time, but by keeping *frequently-requested* dirty pages from being paged out too early. Existing flash-aware replacement algorithms do not differentiate between frequently-requested dirty pages and once-requested dirty pages, which makes them, though delay the time of evicting a dirty page by paging out clean pages firstly, fail to make further improvement on the hit ratio of *frequently-requested* dirty pages when *once-requested* dirty pages occupying too much space, such that previously *frequently-requested* dirty pages may be paged out before some *once-requested* dirty pages because of their different recency.

Further, [9] pointed out that "real-life workloads do not admit a one-size-fits-all characterization. They may contain long sequential request or moving hot spots. The frequency and scale of temporal locality may also change with time. They may fluctuate between stable, repeating access patterns and access patterns with transient clustered references. No static, a priori fixed replacement policy will work well over such access patterns".

Different from the previous buffer replacement policies that focus on either the various access patterns with uniform access cost, or the asymmetry of access cost of flash, in this paper, we further address the impact imposed by the *discrepancy* of the ratio of write cost to read cost on different flash disks. This motivates us to design an adaptive cost-based buffer replacement policy that possesses three features: (1) low overall I/O cost of serving all requests based on flash disks of different ratios of write cost to read cost, (2) constant-time complexity per request, (3) adaptive to dynamically evolving workloads.

We propose a new buffer replacement policy, namely, Adaptive Cost-aware buffer Replacement (ACR). First, ACR uses three cost-based heuristics to select the victim page, thus can fairly make trade off between clean pages and dirty pages, and hence, can work well for different kind of flash disks with large discrepancy of the ratio between read and write operations. Second, ACR organizes buffer pages into clean list and dirty list, the newly entered pages are not inserted at the MRU position of either list, but at some position in the middle. As a result, the once-requested pages can be flushed out from the buffer quickly and the frequently-requested pages can stay in buffer for a longer time. This mechanism makes ACR adaptive to workloads of different access patterns and can really improve the hit ratio of frequently-requested pages so as to improve the overall performance.

Moreover, ACR maintain a buffer directory, namely, ghost buffer, to remember recently evicted "*once-requested*" buffer pages. The hits on the ghost LRU lists are used to adaptively determine the length of the buffer list and identify more frequently-requested pages, such that ACR can adaptively decide how many pages each list should maintain in response to an evolving workload.

The remainder of this paper is organized as follows. Section II introduces background knowledge about flash disks and existing buffer replacement polices. Section III introduces our ACR algorithm and the experimental results are presented in Section IV. We conclude our work in Section V.

II. Background and Related Work

In this section, we firstly review the most important hardware characteristics of flash disks, then give a detailed discussion of existing replacement policies, which motivates us to devise the new replacement policy.

A. Flash Memory

Generally speaking, there are two different types of flash memories: NOR and NAND flash memories[1]. Compared with NAND flash memory, NOR flash memory has separate address and data buses like EPROM and static random access memory (SRAM) while NAND flash memory has an I/O interface which control inputs and outputs. NOR flash memory was developed to replace programmable read-only memory (PROM) and erasable PROM (EPROM) for efficient random access while NAND flash memory was developed for data storage because of its higher density. Flash disks usually consist of NAND flash chips.

There are three basic operations on NAND flash memories: read, write, and erase. Read and write operations are performed in units of a page. Erase operations are performed

[1]http://www.dataio.com/pdf/NAND/MSystems/MSystems_NOR_vs_NAND.pdf

in units of a block, which is much larger than a page, usually contains 64 pages. NAND flash memory does not support in-place update, the write to the same page cannot be done before the page is erased. Moreover, Each block of flash memory may worn out after the specified number of write/erase operations. To avoid the premature worn out of blocks caused by highly localized writes, it is necessary to distribute erase operations evenly over all blocks.

To overcome the physical limitation of flash memory, flash disks employ an intermediate software layer called Flash Translation Layer (FTL), which is typically stored in a ROM chip, to emulate the functionality of block device and hide the latency of erase operation as much as possible. One of the key roles of FTL is to redirect a write request on a page to an empty area erased previously. Therefore, FTL needs to maintain an internal mapping table to record the mapping information from the logical address number to the physical location. This internal mapping table is maintained in volatile memory. The reconstruction of the mapping table is at startup or in case of a failure. The details of the implementation of FTL are device-related and supplied by the manufacturer, which are transparent to users.

Compared with magnetic disks, although NAND flash memories have various advantages such as small and lightweight form factor, solid-state reliability, no mechanical latency, low power consumption, and shock resistance [18], they also possess inherent limitations, say asymmetric operation latencies, and the degree of the asymmetry various largely from one to another. Specifically, a flash memory has asymmetric read and write operation characteristics in terms of performance and energy consumption. It usually demonstrates extremely fast random read speeds, but slow random write speeds. Moreover, as shown in Fig. 1, the ratio of the cost of write and read operation for different flash disks various largely. Therefore when designing flash-aware buffer replacement policy, not only the asymmetry of read and write should be considered, but also the discrepancy of the asymmetries of different flash disks should be paid more attention.

B. Buffer Replacement Policies

Consider the typical scenario where a system consists of two memory levels: main (or buffer) and auxiliary. Buffer is significantly faster than the auxiliary memory and both memories are managed in units of equal sized pages.

Assuming that the secondary storage consists of magnetic disks and the costs of all eviction operations are equal to each other, the goal of existing buffer replacement policies is to minimize the buffer miss ratio for a given buffer size. The miss ratio reflects the fraction of pages that must be paged into the buffer from the auxiliary memory. For example, recent studies on replacement algorithms such as 2Q [7], ARC [9], LIRS [8], CLOCK [4], LRU-K [6], FBR [5] and LRFU [10] mainly aim to improve the traditional LRU heuristic, which consider page recency or balance both recency and frequency to reduce miss rate. However, the above assumption is not hold anymore when applied to flash disks because of the asymmetric access

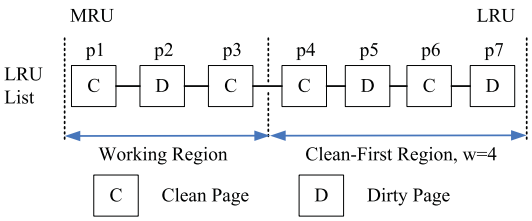


Fig. 2: The CFLRU Replacement Policy

times. This adds another dimension to the management of flash disk based buffer.

The replacement problem for buffers with non-uniform access time can be modeled by the weighted buffering problem. The goal is to minimize the total cost to serve the request sequence. [19] proposed an optimal *off-line* algorithm for this problem in $O(sn^2)$ time by reducing it to the minimal cost maximum flow problem [20], where s is the buffer size and n is the number of total requests. Unfortunately, this optimal algorithm is resource intensive in terms of both space and time, even though it knows all prior knowledge of the complete request sequence.

For an *online* algorithm, any knowledge about the future requests is unknown in advance. Recently, researchers have proposed many online flash-aware buffer replacement policies.

The flash aware buffer policy (FAB) [13] maintains a block-level LRU list, of which pages of the same erasable block are grouped together. When a hit occurs on a page, the group containing the page is moved to the beginning of the LRU list. When a miss occurs, the group that has the largest number of pages will be selected as victim and all dirty pages in this group will be paged out. FAB is mainly used in portable media player applications where most write requests are sequential.

BPLRU [14] also maintains an block-level LRU list. Different from FAB, BPLRU [14] uses an internal RAM of SSD as a buffer to change random write to sequential write to improve the write efficiency and reduce the number of erase operation. However, this method cannot really reduce the number of write requests from main memory buffer.

Clean first LRU (CFLRU) [12] is a flash aware buffer replacement algorithm for operating systems. It was designed to exploit the asymmetric performance of flash IO by first paging out clean pages arbitrarily based on the assumption that writing cost is much more expensive. Fig. 2 illustrates the idea of CFLRU. The LRU list is divided into two regions: the working region and the clean-first region. Each time a miss occurs, if there are clean pages in the clean-first region, CFLRU will select the least recent referenced clean page in the clean-first region as a victim. Only when there is no clean page in the clean-first region, the dirty page at the LRU position of the clean-first region is selected as a victim. The size of the clean-first region is controlled by a parameter w called the window size. Compared with LRU, CFLRU reduces the write operations significantly.

TABLE I: Summary of notations

Notation	Description
L_C	the LRU list containing clean pages
L_{CT}	the top portion of L_C
L_{CB}	the bottom portion of L_C
δ_C	the number of clean pages contained in L_{CB}
L_D	the LRU list containing dirty pages
L_{DT}	the top portion of L_D
L_{DB}	the bottom portion of L_D
δ_D	the number of dirty pages contained in L_{DB}
L_{CH}	the LRU list containing page id of once-requested clean pages
L_{DH}	the LRU list containing page id of once-requested dirty pages
C_r	the cost of reading a page from a flash disk
C_w	the cost of writing a dirty page to a flash disk
s	the size of the buffer in pages
M_{L_D}	the number of physical operations on pages in L_D
M_{L_C}	the number of physical operations on pages in L_C
R_{L_D}	the number of logical operations on pages in L_D
R_{L_C}	the number of logical operations on pages in L_C

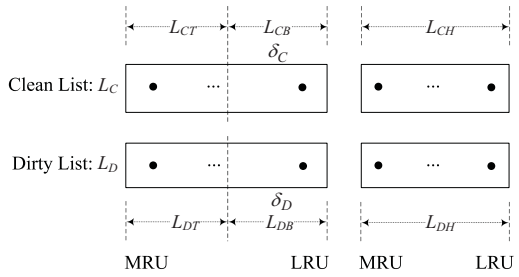


Fig. 3: The ACR Replacement Policy

Based on the same idea, [15] makes improvements over CFLRU by organizing clean pages and dirty pages into different LRU lists to achieve constant complexity per request. Further, CFDC [16] improves the system performance by clustering together dirty pages whose page numbers are close to each other, thus can improve the efficiency of write operations. In CFDC, a cluster has variable size determined by the set of pages currently kept, which is different from block-level LRU list.

III. The ACR Policy

A. Data Structures

As shown in Fig. 3, ACR splits the LRU list into two LRU lists, say L_C and L_D. L_C is used to keep *clean* pages and L_D is used to keep *dirty* pages. Assume that the buffer contains s pages when it is full, then $|L_C \cup L_D| = s \wedge L_C \cap L_D = \emptyset$. Further, L_C is divided into L_{CT} and L_{CB}, and $L_{CT} \wedge L_{CB} = \emptyset$, L_{CT} contains *frequently-requested clean* pages while L_{CB} contains *once-requested clean* pages and frequently-requested clean pages that are *not* referenced for a long time. Similarly, L_D is also divided into L_{DT} and L_{DB}, and $L_{DT} \wedge L_{DB} = \emptyset$. L_{DT} contains *frequently-requested dirty* pages while L_{CB} contains *once-requested dirty* pages and frequently-requested dirty pages that are *not* referenced for a long time. The sizes of L_{CB} and L_{DB} will dynamically change with the change of access patterns, which are controlled by δ_C and δ_D, respectively. Besides the pages that are in the buffer, we use a ghost buffer to trace the past references by recording the page id of those pages that are paged out from L_C or L_D. The ghost buffer is also divided into two LRU lists, say, L_{CH} and L_{DH}, which are used to keep the past references of clean and dirty pages, respectively. All pages in L_{CH} and L_{DH} are those that are *never* being requested again since they were paged into the buffer last time, that is, they are the *once-requested* pages. Fixing this parameter is potentially a tuning question, in our experiment, $|L_{CH} \cup L_{DH}| = s/2$. The notions used in this paper are shown in Table I.

B. Cost-based Eviction

If the buffer is full and the currently requested page p is in the buffer, then it is served without access the auxiliary storage, otherwise, ACR will select from L_C or L_D a page x for replacement according to the metrics of "*cost*", not clean or dirty. The *cost* associated to L_C (L_D), say C_{L_C} (C_{L_D}), is a weighted value denoting the overall replacement cost caused by the pages in L_C (L_D). The basic idea behind our replacement policy is that the length of L_C (L_D) should be proportional to the ration of the replacement cost of the pages in L_C (L_D) to that of all buffer pages according to recent m requests, in our experiment, m equals to half the buffer size, i.e., $m = s/2$. This ratio can be formally represented as Formula 1:

$$\beta = C_{L_C}/(C_{L_C} + C_{L_D}) \tag{1}$$

The policy of selecting a victim page can be stated as: If $|L_C| < \beta \cdot s$, which means that L_D is too long, then the LRU page in L_D should be paged out, otherwise L_C is too long and the LRU page of L_C should be paged out, the "s" in this inequation is the buffer size in pages.

In the following discussion, we call the read and write operations that are served in buffer are logical hereafter, while ones that reach the disk are referred to as physical. The cost of reading a page from the flash disk is C_r, while the cost of writing a page to a random position in a flash disk is C_w. We present a family of methods to compute the values of C_{L_C} and C_{L_D} to decide the optimal scheme.

1) Conservative Scheme: Let M_{L_C} be the number of physical operations on pages in L_C and M_{L_D} the number of physical operations on pages in L_D. The first scheme used for computing C_{L_C} and C_{L_D}, which we refer to as conservative, is given by Formula 2 and Formula 3. Upon eviction, C_{L_C} and C_{L_D} are examined to compute the value of β.

$$C_{L_C} = \begin{cases} C_r, & M_{L_C} = 0 \\ M_{L_C} \cdot C_r, & M_{L_C} \neq 0 \end{cases} \tag{2}$$

$$C_{L_D} = \begin{cases} C_w, & M_{L_D} = 0 \\ M_{L_D} \cdot (C_w + C_r), & M_{L_D} \neq 0 \end{cases} \tag{3}$$

Note that before L_C or L_D seeing the first physical operation, C_{L_C} and C_{L_D} are assigned with C_r and C_w, respectively.

The conservativity of Formula 2 and Formula 3 lies in that they take into account only physical operations on pages, not logical ones. Therefore the conservative scheme does not try to induce the access pattern from the logical operation. Rather, it waits until the logical operation has been translated into physical accesses.

2) Optimistic Scheme: Though physical operations capture the actual cost paid by L_C and L_D, their sequences are dictated by logical operations. Moreover, while the pages remain in the buffer, many logical operations may occur between two consecutive physical operations. Formula 2 and Formula 3 will only record physical operations on these pages, and thus, if the workload changes, L_C and L_D will take many physical operations before conservative adapts. This motives us to design an "*optimistic*" version of the eviction scheme that works only on logical operations and thus, can adapt to new workloads as quickly as possible; the optimistic scheme is given by Formula 4 and Formula 5, where R_{L_C} refers to the number of logical operations on the pages of L_C while R_{L_D} the number of logical operations on the pages of L_D. R_{L_C} (R_{L_D}) is incremented by 1 when a logical read (write) occurs on a page in L_C (L_D). The two counters hold the total logical read and write operations on L_C and L_D, respectively. Upon eviction, our method will compute the total cost L_C and L_D would pay if these operations were physical.

$$C_{L_C} = R_{L_C} \cdot C_r \tag{4}$$

$$C_{L_D} = R_{L_D} \cdot (C_w + C_r) \tag{5}$$

The optimistic scheme is not conservative in the number of evicted pages. It assumes that when the workload changes from read-intensive to write-intensive (or vice-versa), the selection of a victim page should be changed from L_D to L_C (or vice-versa). Thus, optimistic adapts quickly to changing workloads. However, if the changes of the access pattern do not last long enough for the eviction cost to be expensed, the overall cost paid by the system grows.

Notice that the optimistic scheme tries to minimize the cost of future physical operations on L_C and L_D based on its history of logical operations. Consider the case that before a new eviction, L_C having been logically read a large number of times, then L_C upon eviction is found to be strongly read-intensive and the selection of victim page is very probably from L_D, that is, the LRU page p of L_D will be paged out. After that, if there is a write request on p, an expensive write cost is already paid by L_D. In such a case, not only the benefit from the cost-based eviction never realized, but also the system performance degenerates.

3) Hybrid Scheme: Logical and physical operations are two different operations, however, they all have impact on the overall performance. Both the conservative and optimistic scheme introduced above choose to consider only one of them, therefore may not really work in some cases. To minimize the total cost of physical operations, we introduce a hybrid scheme that takes both physical and logical operations into account by combining the strong points of the conservative and optimistic scheme, while avoid their weak points.

Assume that n is the number of pages in a file and s the number of pages allocated to the file in the buffer. Therefore the probability that a logical operation will be served in the buffer is s/n, and the probability that a logical operation will be translated to a physical one is $(1 - s/n)$. In our hybrid scheme, the probability is used to compute the overall impact of logical operations on L_C and L_D, as shown by Formula 6 and 7. Upon eviction, our method will compute the total cost of L_C and L_D by considering the impacts of both logical and physical operations.

$$C_{L_C} = (R_{L_C} \cdot (1 - s/n) + M_{L_C}) \cdot C_r \tag{6}$$

$$C_{L_D} = (R_{L_D} \cdot (1 - s/n) + M_{L_D}) \cdot (C_w + C_r) \tag{7}$$

When selecting a victim page, the logical operations allow our hybrid scheme to recognize changes in the access pattern very quickly like the optimistic scheme. Moreover, it is also not so eager as the optimistic scheme to page out the expensive dirty page by considering the actually happened physical operations. By taking into account the cost of actually happened physical operations, the hybrid scheme has a realistic view of the impact the logical operations imposed on the buffer.

C. The ACR Replacement Policy

We now introduce the whole ACR replacement policy that adapts and tunes the length of L_C and L_D in response to an observed workload. Before running, $\delta_C = \delta_D = 0$. For easier discussion, we call a request on a page that is not in the buffer a *miss-request*, otherwise a *hit-request*.

As show in Algorithm 1, in the beginning stage before the buffer is full, i.e., $|L_C \cup L_D| < s \wedge |L_{CH} \cup L_{DH}| = 0$, if the request on p is a *miss-request* and $p's$ page id is not in $L_{CH} \cup L_{DH}$, Algorithm 1 will execute the code in Case III. Since $|L_C \cup L_D| < s$, ACR increases the logical and physical counters according to the operation type, then fetch p into the buffer and insert it to the MRU position of L_{CB} or L_{DB} according to the value of T. At last, δ_C or δ_D will increase by 1 by calling the procedure AdjustBottomProtionList(). If the current request on p is a *hit-request*, that is, $p \in L_C \cup L_D$, ACR will execute the code in Case I. Specifically, if $p \in L_{CB}$ (L_{DB}), it means that p should not stay anymore in L_{CB} (L_{DB}), since L_{CB} (L_{DB}) is used to maintain once-requested clean (dirty) pages and frequently-requested clean (dirty) pages that are not requested yet for a long time. Then ACR will move p to the MRU position of L_{CT} or L_{DT} and adjust the size of L_{CB} and L_{DB}, respectively.

If the buffer is full. For a *hit-request* corresponding to Case I (line 1-8 of Algorithm 1), the process is already discussed in the above paragraph. If the current request is a *miss-request*, then ACR will check whether $p's$ id is contained in $L_{CH} \cup L_{DH}$ or not. If $p's$ id is contained in $L_{CH} \cup L_{DH}$, which corresponds to Case II, it means that p has not been request after it entered into $L_{CB} \cup L_{DB}$. In this case, ACR will firstly

Algorithm 1: ACR(page p, type T) /* ACR is triggered on each request on a page p, T denotes the type of operation on p, T can be either "$read$" or "$write$"*/

Case I: $p \in L_C \cup L_D$, a buffer hit has occurred.

1 **if** $(p \in L_C)$ **then** $\{R_{L_C} \leftarrow R_{L_C} + 1$; **if** $(p \in L_{CB})$ **then** $\{\delta_C \leftarrow max\{0, \delta_C - 1\};\}\}$;
2 **else** $\{R_{L_D} \leftarrow R_{L_D} + 1$; **if** $(p \in L_{DB})$ **then** $\{\delta_D \leftarrow max\{0, \delta_D - 1\};\}\}$
3 **if** $(T = read \wedge p \in L_C)$ **then** {move p to the MRU position of L_{CT};}
4 **else if** $(p \in L_D)$ **then** {move p to the MRU position of L_{DT};}
5 **else** {move p to the MRU position of L_{DB};}
6 **if** (p is moved from L_C to L_D) **then** $\{p.hit \leftarrow 0;\}$
7 **else** $\{p.hit \leftarrow p.hit + 1;\}$ /*$p.hit$ is the number of hit occurred on p since it entered into L_C or L_D*/
8 AdjustBottomPortionList();

Case II: $p \in L_{CH} \cup L_{DH}$, a buffer miss has occurred.

9 evictPage(); $p.hit \leftarrow 0$;
10 **if** $(p \in L_{CH})$ **then** $\{\delta_C \leftarrow min\{|L_C|, \delta_C + 1\};\}$
11 **else** $\{\delta_D \leftarrow min\{|L_D|, \delta_D + 1\};\}$
12 **if** $(T = read)$ **then** {fetch p from the disk; insert it to the MRU of L_{CT}; $M_{L_C} \leftarrow M_{L_C} + 1$; $R_{L_C} \leftarrow R_{L_C} + 1$;}
13 **else** {fetch p from the disk; insert it to the MRU of L_{DT}; $R_{L_D} \leftarrow R_{L_D} + 1$;}
14 AdjustBottomPortionList();

Case III: $p \notin L_C \cup L_D \cup L_{CH} \cup L_{DH}$, a buffer miss has occurred.

15 $p.hit \leftarrow 0$;
16 **if** $(|L_C \cup L_D| = s)$ **then** {evictPage();}
17 **if** $(T = read)$ **then** {fetch p from the disk; insert it to the MRU of L_{CB};$R_{L_C} \leftarrow R_{L_C} + 1$; $M_{L_C} \leftarrow M_{L_C} + 1$;}
18 **else** {fetch p from the disk; insert it to the MRU of L_{DB};$R_{L_D} \leftarrow R_{L_D} + 1$;}
19 AdjustBottomPortionList();

Procedure evictPage()

1 $\beta \leftarrow C_{L_C}/(C_{L_C} + C_{L_D})$; /*$\beta$ is computed based on the recent $s/2$ requests*/

Case I: $|L_C| < \beta \cdot s$ /* L_D is longer than expected*/.

2 $M_{L_D} \leftarrow M_{L_D} + 1$;
3 let q be the page in the LRU position of L_{DB} and $q.hit$ the number of hit occurred on q since it entered into L_D;
4 **if** $(q.hit > 0)$ **then** {write $q's$ content to disk; delete q; return;}
5 **if** $(|L_{CH} \cup L_{DH}| = s/2)$ **then** {delete the item in the LRU position of L_{DH};}
6 delete q and insert the page id of q as a new item in the MRU position of L_{DH};

Case II: $|L_C| \geq \beta \cdot s$ /* L_C is longer than expected*/.

7 let q be the page in the LRU position of L_{CB} and $q.hit$ the number of hit occurred on q since it entered into L_C;
8 **if** $(q.hit > 0)$ **then** {delete q; return;}
9 **if** $(|L_{CH} \cup L_{DH}| = s/2)$ **then** {delete the item in the LRU position of L_{CH};}
10 delete q and insert the page id of q as a new item in the MRU position of L_{CH};

Procedure AdjustBottomPortionList()

1 **if** $(|L_C \cup L_D| = s)$ **then**
2 Move the MRU (or LRU) page of L_{CB} and L_{DB} (or L_{CT} and L_{DT}) to LRU (MRU) position of L_{CT} and L_{DT} (or L_{CB} and L_{DB}) to make $|L_{CB}| = \delta_C \wedge |L_{DB}| = \delta_D$;
3 **else** $\{\delta_C \leftarrow |L_{CB}|; \delta_D \leftarrow |L_{DB}|;\}$

call evictPage() to select a victim page and page it out to make room for p, then δ_C or δ_D will increase by 1 since the size of L_{CB} or L_{DB} is too small. After that, ACR will fetch p from disk and insert it to the MRU position of L_{CT} or L_{DT}. At last, ACR will adjust the length of L_{CB} and L_{DB}. If the current request is a *miss-request* and $p's$ id is not in $L_{CH} \cup L_{DH}$, this case corresponds to Case III. Compared with the case that buffer is not full, ACR will firstly evict a page in this case.

Note that in ACR, pages that are served only once in the whole processing will be inserted at the MRU position of L_{CB} or L_{DB}, thus will be paged out earlier than those served more than once. Moreover, the pages in L_{CT} and L_{DT} can further utilize the space of L_{CB} and L_{DB} to make them staying longer in the buffer, such that the hit ratio of frequently-requested pages can be actually improved, especially for dirty pages. By using a hash table to maintain the pointers to each page in the buffer, the complexity of ACR for each page request is $O(1)$ and is only greater than the complexity of LRU by some constant c.

1) Adaptivity: The adaptivity of ACR lies in two aspects: (1) ACR continually revises the parameter δ_C and δ_D that are used to control the size of L_{CB} and L_{DB}. The fundamental intuition behind is: if there is a hit on page p of $L_{CB}(L_{DB})$ that mainly contains *once-requested* pages, then p becomes a frequently-requested page from now on and should be placed in $L_{CT}(L_{DT})$, and we should increase the size of $L_{CT}(L_{DT})$. Similarly, if $p's$ page id is in $L_{CH}(L_{DH})$ that records the historical reference information of once-requested pages, then we should increase the size of $L_{CB}(L_{DB})$. Hence, on a hit in $L_{CB}(L_{DB})$, we decrease $\delta_C(\delta_D)$, and on a hit in $L_{CH}(L_{DH})$, we increase $\delta_C(\delta_D)$. If the workload is to change from one access pattern to another one or vice versa, ACR will track such change and adapt itself to exploit the new opportunity. (2) ACR will choose a page for replacement according to the accumulative replacement costs of L_C and L_D, which gives a fair chance to clean and dirty pages for competition. Together, the two aspects of adaptivity makes ACR very wise in exploiting the asymmetry of flash IO and the new opportunity of various access pattern.

2) Scan-Resistant: When serving a long sequence of one-time-only requests, ACR will only evict pages in $L_{CB} \cup L_{DB}$ and it never evicts pages in $L_{CT} \cup L_{DT}$. This is because, when requesting on a totally new page p, i.e., $p \notin L_C \cup L_D \cup L_{CH} \cup L_{DH}$, p is always put at the MRU position of L_{CB} or L_{DB}. It will not impose any affect on pages in $L_{CT} \cup L_{DT}$ unless it is requested again before it is paged out from L_{CB} or L_{DB}. For this reason, we say ACR is scan-resistant. Furthermore, a buffer is usually used by several processes or threads concurrently, when a scan of a process or thread begins, less hits will be encountered in $L_{CB} \cup L_{DB}$ compared to $L_{CT} \cup L_{DT}$, and, hence, according to Algorithm 1, the size of L_{CT} and L_{DT} will grow gradually, and the resistance of ACR to scans is strengthened again.

3) Loop-Resistant: A loop requests is a sequence of pages that are served in a special order repeatedly. We say that ACR is loop-resistant means that when the size of the loop is larger than the buffer size, ACR will keep partial pages of the loop sequence in the buffer, and hence, achieve higher performance. We explain this point from three aspects in the case that the size of a loop is larger than the buffer size.

(1) the loop requests only pages in L_C. In the first cycle of the loop request, all pages are fetched into the buffer and inserted at the MRU position of L_{CB} sequentially. Before each insertion, ACR will select a victim page q. If q is the LRU page of L_{DB}, then after the insertion of p in the MRU position of L_{CB}, ACR will adjust the size of L_{CB} and p will be adjusted to the LRU position of L_{CT}; otherwise p is still at the MRU position of L_{CB}. With the processing of the loop requests, more pages of the loop sequence will be moved to L_{CT} and these pages are thus kept in buffer, therefore the hit ratio will not be zero anymore. (2) the loop requests only pages in L_D. This is same to (1). (3) the loop contains pages in both L_C and L_D. In this case, obviously, dirty pages will stay in buffer longer than clean pages and the order of the pages eviction is not same as they entered in the buffer, and hence, ACR can process them elegantly to achieve higher hit ratio.

IV. Experiments

A. Experimental Setup

The goal of our experiment is to verify the effectiveness of ACR for flash disks of different characteristics on read and write operations. For a flash disk, the performance of a buffer replacement algorithm is affected by the number of physical read and write. However, the implementation of FTL is device-related and supplied by the disk manufacturer, and there is no interface supplied for users to trace the number of write and read. Therefore, we choose to use a simulator [21] to count the numbers of read and write operations. We implemented three existing state-of-the-art replacement policies for comparison, i.e., LRU, CFLRU [12] and CFDC [16]. We implemented three versions of ACR based on the three heuristics (Conservative, Optimistic and Hybrid), which are denoted as ACR-C, ACR-O and ACR-H, respectively. All were implemented on the simulator using Visual C++ 6.0. For CFLRU, we set the "window size" of "clean-first region" to 75% of the buffer size, for CFDC, the "window size" of "clean-first region" is 50% of the buffer size, and the "cluster size" of CFDC is 64.

We simulated a database file of 64MB, which corresponds to 32K physical pages and each page is 2KB, the buffer size ranges from 2K pages to 8K pages.

We have generated 4 types of synthetical traces which will access all pages randomly. The statistics of the four traces are shown in Table II, where $x\%/y\%$ in column "Read/Write Ratio" means that for a certain trace, $x\%$ of total requests are about read operations and $y\%$ about write operations; while $x\%/y\%$ in column "Locality" means that for a certain trace, $x\%$ of total operations are performed in a certain $y\%$ of the total pages.

We select two flash disks for our experiment, the first is Samsung MCAQE32G5APP, the second is Samsung MCAQE32G8APP-0XA [17]. The ratio of the cost of random

TABLE II: The statistics of the traces used in our experiment

Trace	Total Requests	Read/Write Ratio	Locality
T1	3,000,000	90% / 10%	60% / 40%
T2	3,000,000	80% / 20%	50% / 50%
T3	3,000,000	60% / 40%	60% / 40%
T4	3,000,000	80% / 20%	80% / 20%

read to that of random write is 1:118 and 1:2, respectively. The reason for the huge discrepancy of the two flash disks lies in that the first flash disk is based on MLC NAND chip, while the second flash disk is based on SLC NAND chip. Both type of flash disks are already adopted as auxiliary storage in many applications. In our experiment, the simulator assume that the page size is 2KB, and each block contains 64 pages.

We choose the following metrics to evaluate the six buffer replacement policies: (1) number of physical read operations, (2) number of physical write operations, and (3) running time. The running time is computed by adding up the cost of read and write operations, though there may exist some differences compared with the results tested on a real platform, they reflect the overall performance of different replacement policies by and large with neglectable tolerance. We do not use hit ratio as a metric since it cannot really reflect the overall performance. The results of the second metrics in our experiment include the write operations caused by the erase operations of flash disks.

Note that if running on a real flash disk, CFDC may achieve better performance, this is because CFDC can make many random write to sequential write. On the contrary, CFLRU suffers from high CPU cost, which is also not reflected in our results.

B. Experimental Results and Analysis

1) Impact of large discrepancy on read and write operation: Fig. 4 shows the results of random read, random write and normalized running time on trace T1 to T4 for Samsung MCAQE32G5APP flash disk. Fig. 4 (a), (d), (g) and (j) are the results of the number of random read operations on trace T1 to T4, from which we know that LRU has least read operations, the reason lies in that LRU does not differentiate read and write operations, thus it will not delay the paging out of dirty pages in the buffer. On the contrary, CFLRU firstly pages out clean pages, thus it needs to read in more pages than LRU, CFDC, ACR-C, ACR-O and ACR-H. We can see from Fig. 4 (b), (e), (h) and (k) that LRU consumes more write cost than all other methods, and among the five flash-based methods, i.e., CFLRU, CFDC, ACR-C, ACR-O and ACR-H, CFDC and ACR-O suffer from more write operations, this is because, CFDC will page out all pages in a cluster before paging out pages in other clusters, and ACR-O often makes wrong predictions for the four traces when the cost ratio is 1:118. Although CFLRU suffers from less write operations than LRU, CFDC and ACR-O, we can see that ACR-C and ACR-H consume less write operations than CFLRU, this is because for the cost ratio of 1:118, (1) ACR-C and ACR-H often make correct predictions, (2) ACR-C and ACR-H maintain more dirty pages in the buffer than CFLRU. Fig. 4 (c), (f), (i) and (l) present the results of normalized running time, from which we know that ACR-C and ACR-H achieve higher performance than LRU, CFLRU, CFDC and ACR-O. The reason lies in that the cost of write operation is much more expansive than that of read operation for Samsung MCAQE32G5APP flash disk.

Thus for flash disks with large discrepancy on read and write operations, by firstly paging out clean pages, CFLRU, CFDC, ACR-C, ACR-O and ACR-H are better than LRU since they improve the overall performance by reducing the costly write operations significantly. Moreover, ACR-C and ACR-H are better than CFLRU and CFDC, because they make correct prediction and frequently-requested dirty pages stay in buffer longer than the once-requested dirty pages, thus can further reduce the cost of write operations.

2) Impact of small discrepancy on read and write operation: Fig. 5 just shows the results for trace T1 and T2 running on Samsung MCAQE32G8APP-0XA flash disk for limited space. Fig. 5 (a) and (d) are the results of the number of random read operations on trace T1 and T2, from which we know that LRU, ACR-C, ACR-O and ACR-H have least read operations, the reason lies in that LRU does not differentiate read and write operations, thus it will not delay the paging out of dirty pages in the buffer. Although our ACR policy keeps more dirty pages in buffer than clean pages since the cost of read operation is still cheaper than that of write operation, ACR achieves competing performance to LRU for read operation by improving the hit ratio of frequently requested clean pages. CFLRU and CFDC firstly page out clean pages, thus they need to read in many more pages than LRU and ACR. As a result, they suffer from large read cost. We can see from Fig. 4 (b) and (e) that the number of write operations of ACR-C, ACR-O and ACR-H becomes larger than that in Fig. 4, this is because the ratio of read and write becomes smaller than before, and our policy will pay more attention to clean pages. Though CFLRU and CFDC have less write operations than LRU, they waste many more read operations, which makes them achieving worse performance than LRU, ACR-C, ACR-O and ACR-H for flash disks of small discrepancy on read and write operation, as shown in Fig. 5 (c) and (f).

Therefore, for flash disks with small discrepancy on read and write operations, ACR-C, ACR-O and ACR-H are better than LRU, CFLRU and CFDC, because ACR-C, ACR-O and ACR-H only consume the same or less read operations than LRU, which is much less than that consumed by CFLRU and CFDC; though still need to consume more write operations than CFLRU and CFDC, the saved cost of read operation is much more than that wasted by write operations.

3) Impact of different heuristics: By comparing Fig. 4 and Fig. 5, we can see that for trace T1 to T4, ACR-O is not efficient as ACR-C and ACR-H for flash disks of large discrepancy of read and write operation, but is better than ACR-C and ACR-H for flash disks of small discrepancy. For flash disks of large discrepancy, LRU is very inefficient, both CFLUR and CFDC can work better than LRU, but for flash disks of small discrepancy, LRU is more efficient than CFLRU

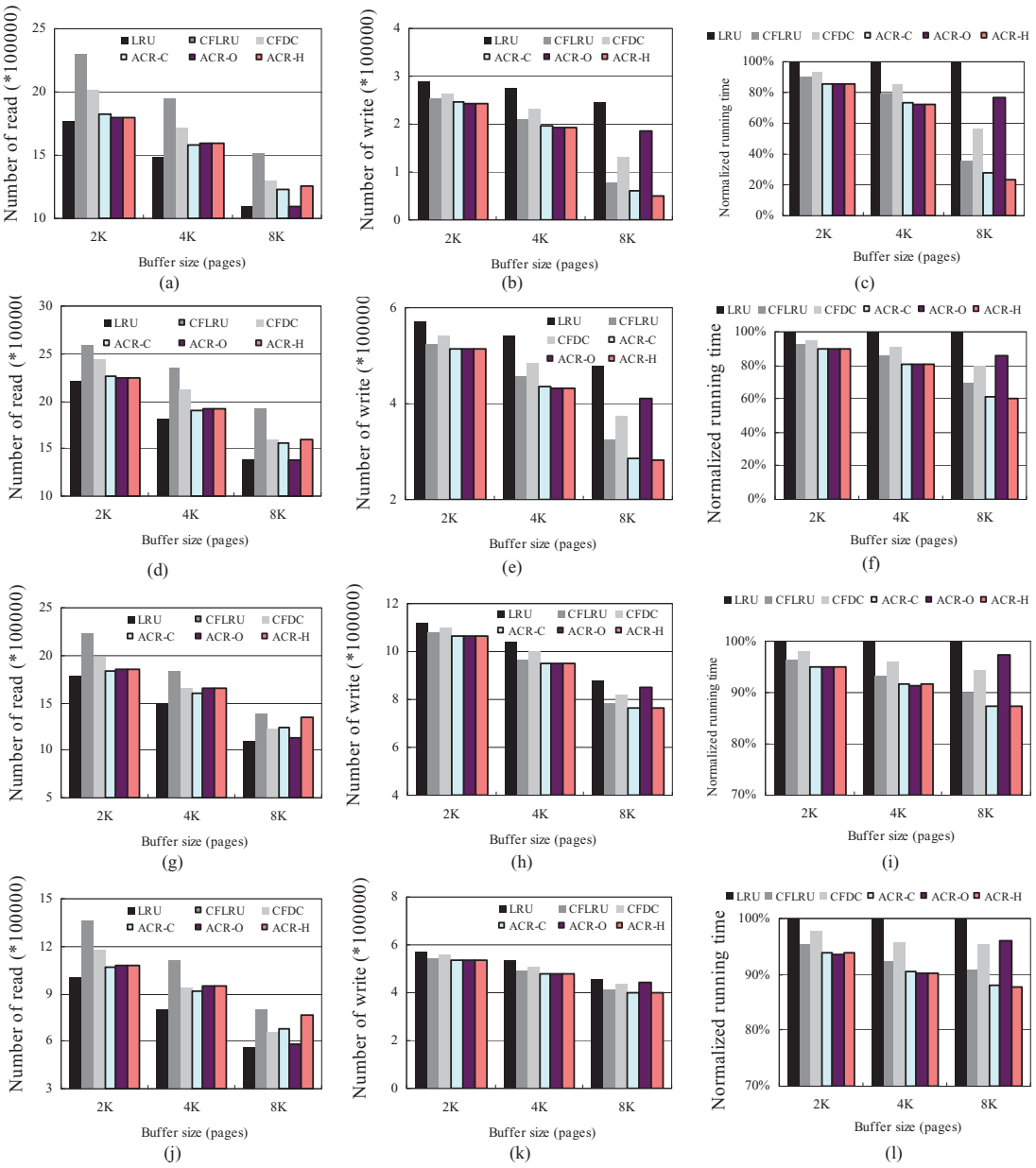


Fig. 4: The comparison of random read, random write and normalized running time on trace T1 to T4 for Samsung MCAQE32G5APP flash disk, (a) to (c) is the result for trace T1, (d) to (f) for trace T2, (g) to (i) for trace T3, and (j) to (l) for trace T4.

and CFDC, the reason lies in that CFLRU and CFDC firstly page out clean pages arbitrarily. If the dirty pages in the buffer are not re-referenced in the near future, then many clean pages will be paged out after they are paged in the buffer for a little time, which will cause many read operations.

In a summary, ACR-C, ACR-O and ACR-H work very efficient when being applied to flash disks with different ratio of read and write costs. Moreover, ACR-C, ACR-O and ACR-H can also work very efficient on workloads of different access patterns.

V. Conclusions

Considering the fact that the discrepancy of the ratio of write cost to read cost for different flash disks various largely and has great affect on designing flash-based buffer replacement policy, in this paper, we address this problem and propose an adaptive cost-based replacement policy, namely ACR. Different from the previous buffer replacement policies that focus on either the various access patterns with uniform access cost, or the asymmetry of access cost for flash, ACR considers

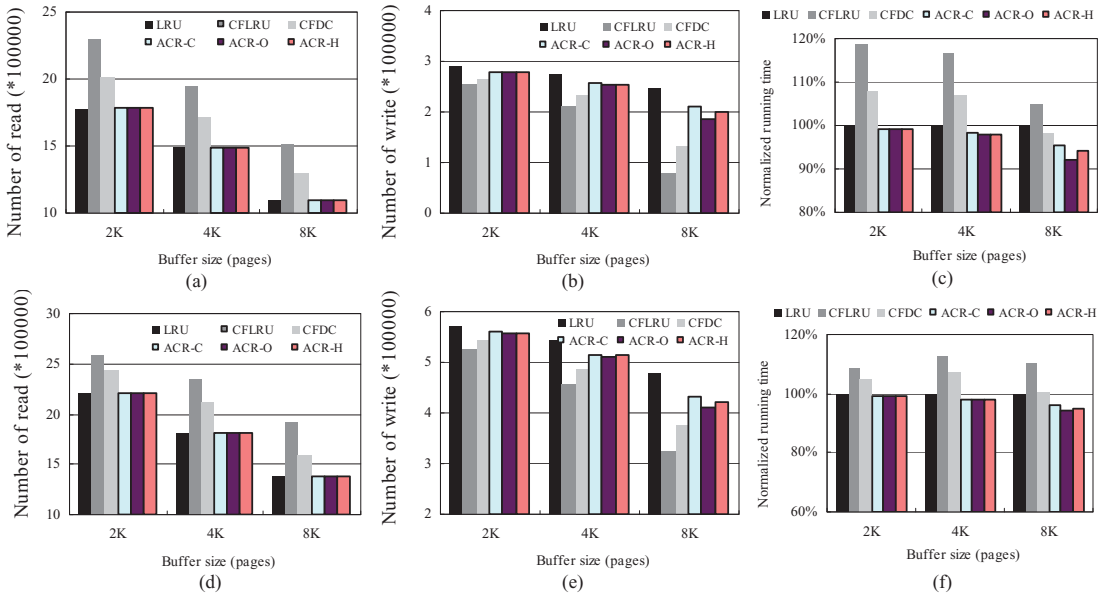


Fig. 5: The comparison of random read, random write and normalized running time on trace T1 and T2 for Samsung MCAQE32G8APP-0XA flash disk, (a) to (c) is the result for trace T1, (d) to (f) for trace T2.

all the above aspects and organizes buffer pages into clean list and dirty list, and the newly entered pages will not be inserted at the MRU position of either list, but at some position in middle, thus the once-requested pages can be flushed out from the buffer quickly and the frequently-requested pages can stay in the buffer for a longer time. Moreover, ACR uses three cost-based heuristics to select the victim page, thus can fairly make trade off between clean pages and dirty pages. The experimental results on different traces and flash disks show that ACR not only adaptively tunes itself to workloads of different access patterns, but also works well for different kinds of flash disks compared with existing methods.

We plan to make further improvement on ACR by considering changing the write operations from random write to sequential write and implement ACR in a real platform to evaluate it with various real world workloads for flash-based applications.

ACKNOWLEDGMENT

This research was partially supported by the grants from the Natural Science Foundation of China (No.60833005); the National High-Tech Research and Development Plan of China (No.2009AA011904); and the Doctoral Fund of Ministry of Education of China (No. 200800020002).

REFERENCES

[1] C. gyu Hwang, "Nanotechnology enables a new memory growth model," in *Proceedings of the IEEE*, 2003, pp. 1765– 1771.
[2] "Windows vista operating system: Readyboost," in *Microsoft Corp.*
[3] S.-W. Lee and B. Moon, "Design of flash-based dbms: an in-page logging approach," in *SIGMOD Conference*, 2007, pp. 55–66.
[4] Ö. Babaoglu and W. N. Joy, "Converting a swap-based system to do paging in an architecture lacking page-reference bits," in *SOSP*, 1981, pp. 78–86.
[5] J. T. Robinson and M. V. Devarakonda, "Data cache management using frequency-based replacement," in *SIGMETRICS*, 1990, pp. 134–142.
[6] E. J. O'Neil, P. E. O'Neil, and G. Weikum, "The lru-k page replacement algorithm for database disk buffering," in *SIGMOD Conference*, 1993, pp. 297–306.
[7] T. Johnson and D. Shasha, "2q: A low overhead high performance buffer management replacement algorithm," in *VLDB*, 1994, pp. 439–450.
[8] S. Jiang and X. Zhang, "Making lru friendly to weak locality workloads: A novel replacement algorithm to improve buffer cache performance," *IEEE Trans. Computers*, vol. 54, no. 8, pp. 939–952, 2005.
[9] N. Megiddo and D. S. Modha, "Arc: A self-tuning, low overhead replacement cache," in *FAST*, 2003.
[10] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C.-S. Kim, "Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE Trans. Computers*, vol. 50, no. 12, pp. 1352–1361, 2001.
[11] W. Effelsberg and T. Härder, "Principles of database buffer management," *ACM Trans. Database Syst.*, vol. 9, no. 4, pp. 560–595, 1984.
[12] S.-Y. Park, D. Jung, J.-U. Kang, J. Kim, and J. Lee, "Cflru: a replacement algorithm for flash memory," in *CASES*, 2006, pp. 234–241.
[13] H. Jo, J.-U. Kang, S.-Y. Park, J.-S. Kim, and J. Lee, "Fab: flash-aware buffer management policy for portable media players," in *IEEE Transactions on Consumer Electronics*, 2006, pp. 485–493.
[14] H. Kim and S. Ahn, "Bplru: A buffer management scheme for improving random writes in flash storage," in *FAST*, 2008, pp. 239–252.
[15] I. Koltsidas and S. Viglas, "Flashing up the storage layer," *PVLDB*, vol. 1, no. 1, pp. 514–525, 2008.
[16] Y. Ou, T. Härder, and P. Jin, "Cfdc: a flash-aware replacement policy for database buffer management," in *DaMoN*, 2009, pp. 15–20.
[17] "http://www.datasheetcatalog.net."
[18] F. Douglis, R. Cáceres, M. F. Kaashoek, K. Li, B. Marsh, and J. A. Tauber, "Storage alternatives for mobile computers," in *OSDI*, 1994, pp. 25–37.
[19] M. Chrobak, H. J. Karloff, T. H. Payne, and S. Vishwanathan, "New results on server problems," *SIAM J. Discrete Math.*, vol. 4, no. 2, pp. 172–181, 1991.
[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.
[21] P. Jin, X. Su, and Z. Li, "A flexible simulation environment for flash-aware algorithms," in *CIKM*, 2009.

FClock:一种面向SSD的自适应缓冲区管理算法

汤 显 孟小峰
(中国人民大学信息学院 北京 100872)

摘 要 现有的各种基于闪存的缓冲区管理算法针对闪存读写代价的不对称性进行改进,实际中既存在同一闪存读写代价的不对称性问题,也存在不同闪存不对称性之间的巨大差异性问题,而后者一直没有得到足够的重视.文章提出一种基于闪存硬盘(SSD)的自适应缓冲区管理算法FClock,FClock将数据页组织为两个环形数据结构(*CC*和*DC*),分别用于存储缓冲区中的只读数据页和已修改数据页.当需要选择置换页时,FClock使用基于代价的启发式来选择置换页,可在未修改的数据页和已修改的数据页之间进行公平的选择,适用于不同种类的SSD.针对数据库、虚存和文件系统中数据页访问存在高相关性的特点,提出基于"平均命中距离"的访问计数方法来调整数据页的访问频率.基于不同SSD和不同存取模式的实验结果说明,FClock的综合性能优于已有方法.

关键词 闪存;数据库;缓冲区;置换策略;CLOCK
中图法分类号 TP391 **DOI号**: 10.3724/SP.J.1016.2010.01460

FClock: An Adaptive Buffer Replacement Algorithm for SSD

TANG Xian MENG Xiao-Feng
(*School of Information, Renmin University of China, Beijing* 100872)

Abstract Different from existing flash-aware buffer replacement policies that focus on the asymmetry of read and write operations, the authors address the "discrepancy" of the asymmetry for different flash disks, which is the fact that exists for a long time, while has drawn little attention by researchers since most existing flash-aware buffer replacement polices are somewhat based on the assumption that the cost of read operation is neglectable compared with that of write operation. This paper proposes an adaptive replacement policy (FClock) which has two ring-shaped data structures, i.e. *CC* (their content remain unchanged) and *DC* (their content is modified), to manage clean pages and dirty pages in the buffer, respectively. When selecting a victim page, FClock uses cost-based heuristics to fairly make trade off between clean pages and dirty pages, and hence, can work well for different type of flash disks of large discrepancy. Further, for the problem of "correlated references" to database, virtual memory and file systems, this paper proposes a reference counter based on "average hit distance" to control the reference frequency. The experimental results on different traces and flash disks show that FClock not only adaptively tunes itself to workloads of different access patterns, but also works well for different kind of flash disks compared with existing methods.

Keywords flash; database; buffer; replacement policy; CLOCK

收稿日期:2010-06-11.本课题得到国家自然科学基金(60833005,60573091)、国家"八六三"高技术研究发展计划项目基金(2007AA01Z155,2009AA011904)、教育部博士点基金项目(200800020002)资助.**汤 显**,女,1978年生,博士研究生,主要研究方向为闪存数据库.E-mail: txianz@gmail.com.**孟小峰**,男,1964年生,教授,博士生导师,研究领域包括网络数据管理、云数据管理、移动数据管理、XML数据管理、闪存数据库系统以及隐私保护.

1 引　言

基于闪存的存储设备以其低延迟、低能耗、小巧轻便及高抗震性等特点广泛应用于移动设备上,随着闪存容量的不断增大和价格的降低,其应用领域已逐步扩展到个人计算机和企业服务器市场.过去几年 NAND 型闪存的容量不断增长,并且这种趋势将至少持续到 2012 年[1].目前各种应用中都将闪存硬盘 SSD 看成一个块设备并使用与磁盘一样的存取接口,但这两种硬盘的 I/O 特性却存在很大的差异.闪存硬盘的随机读速度远快于其随机写速度,在一些对性能要求苛刻或者涉及频繁数据处理的应用场合,如数据库服务器,如果不能根据闪存的特性来设计合适的数据结构和算法,就难以获得最佳性能.

缓冲区是现代计算机最基本的组成部分之一.它在存储系统、数据库、网络服务器、文件系统以及操作系统中都有广泛的应用.缓冲区置换算法的任何进展都会影响现代计算机的整体性能.假设用磁盘做辅存且读写操作的时间延迟相同,那么对于给定大小的缓冲区,现存的缓冲区置换算法[2-8]的目标就是最小化缓冲区的缺页率.当缓冲区已满并且请求的数据页不在缓冲区中时,缓冲区置换算法首先从当前缓冲区选择一个用于置换的数据页,如果所选的数据页是脏页(其内容被修改过),就必须将该页的内容写回硬盘,然后才能将请求的数据页读入缓冲区,以保证数据的一致性.这种操作可能会成为系统性能的瓶颈,这是因为请求页的线程或进程必须等待写操作的完成.早在 20 年前,Effelsberg 等[9]已经认识到缓冲区中数据页的读写状态是影响置换策略的重要因素,由于闪存硬盘有望替代磁盘成为新一代的数据存储设备,并且闪存硬盘的读写代价存在不对称性问题,在设计置换策略时更应考虑读写状态的差异性.

针对闪存读写操作的不对称性问题,近几年研究者已经提出了几种适用于闪存的缓冲区置换策略[10-14].然而,这些置换策略在实际中存在以下问题:

(1)适用的闪存硬盘类型有限

已有的基于闪存的缓冲区置换算法的基本假设是闪存的随机读代价相对于随机写代价来说可忽略不计,因此这些缓冲区置换策略都是通过无条件先置换只读页来减少随机写操作的次数,从而提高系统的性能.然而,如图 1 所示,该假设和实际情况并不相符,即随机读代价和随机写代价相比,并非在所有情况下都可以被无条件忽略.尽管所有的闪存设备都表现出较快的随机读速度和较慢的随机写速度,但对于不同的闪存设备而言,读写代价的比例差别很大,显然在不考虑只读页的操作代价和访问频率的情况下就无条件首先置换只读页是不合理的.

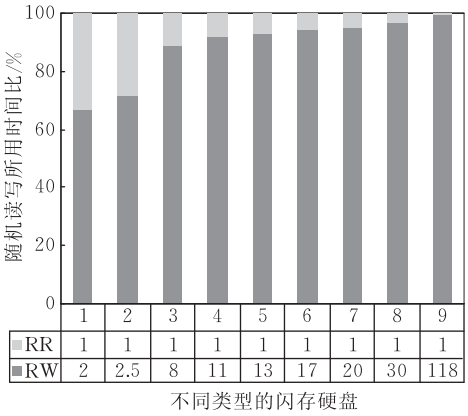


图 1　NAND 闪存硬盘随机读(RR)和随机写(RW)消耗的时间比例(X 轴为 9 种闪存硬盘,1 是三星 MCAQE32G8APP-0XA,2 是三星 K9WAG08U1A,3 是三星 K9XXG08UXM,4 是三星 K9F1208R0B,5 是三星 K9GAG08B0M,6 是现代 HY27SA1G1M,7 是三星 K9K1208U0A,8 是三星 K9F2808Q0B,9 是三星 MCAQE32G5APP①)

(2)多线程处理能力不足

已有的基于闪存的缓冲区置换策略都是基于 LRU 置换策略进行改进,LRU 的优势在于它实现简单,操作代价低,但 LRU 也有自身的局限性:①每次命中的数据页必须移动到最近使用(MRU)的位置.实际中多个线程可能都试图移动各自的数据页到 MRU 位置,而 MRU 位置通过锁保护来保证一致性和正确性.由于所有命中操作都在等待该操作的完成,就会引起大量锁争用的问题.在高性能和高吞吐量环境中,例如虚存、数据库、文件系统和存储控制器中,这种情况是不可接受的.②LRU 没考虑数据页的"访问频率".和 LRU 对应,CLOCK 算法克服了 LRU 算法的上述缺点.通过将数据页组织成时钟形式的环形缓冲区,CLOCK 算法无需在每次命中数据页后移动数据页的位置,因此不会出现锁争用的问题.因此,CLOCK 算法在实际系统中,如 DB2、SQL Server、Postgresql 等,得到了广泛应用.

针对现有基于闪存的缓冲区置换算法存在的问题,本文提出一种基于闪存的 clock 算法 FClock 来解决以上问题.FClock 为每个数据页维护一个"访问位"并将缓冲区中的数据页组织成两个时钟形式

① http://www.datasheetcatalog.net

的环形结构 *CC*(Clean Clock)和 *DC*(Dirty Clock),分别用于管理未修改的数据页和修改过的数据页.当数据页被初次读入缓冲区时,它的访问位被置为0.当命中某个数据页时,将其访问位加1.当需要选择一个置换页时,FClock 并非直接置换出未修改页,而是首先根据启发式规则判断被置换的数据页是已修改页还是未修改页,然后再从 *DC* 或者 *CC* 中找访问位是0的数据页进行置换,以此来进行自适应的调整.最后,受文献[3]的"局部性过滤"原则的启发,FClock 使用基于平均命中距离的技术来消除短期频繁访问而长期不访问的数据页长时间驻留内存的问题.

2 背景及相关工作

2.1 闪存存储器

一般来说,有两种不同类型的闪存芯片,分别是NOR 型闪存和 NAND 型闪存.NOR 型闪存芯片和EPROM 以及 SRAM 一样,有专用的地址和数据总线;而 NAND 闪存芯片无专用的地址和数据总线,用一个 IO 接口来控制输入输出.NOR 型闪存芯片可用来替换可编程的只读存储器(PROM)和可擦除的 PROM(EPROM)来进行有效的随机存取;NAND 型闪存芯片由于其存储容量较高,主要用来存储数据.闪存硬盘(SSD)中使用的通常是 NAND 型芯片.

NAND 型闪存芯片上有3种基本操作:读、写和擦除.读和写都是以数据页为单位进行操作;擦除是以块为单位进行操作,一个块通常包含64个页.NAND 型芯片不支持原地更新,如果某个页上有数据,就无法对该页直接进行覆盖写操作.为了避免对某些块进行频繁的写和擦除操作之后所造成的数据块失效的问题,通常使用磨损平衡技术将写和擦除操作均匀地分布在所有的数据块上.

为了克服闪存芯片的物理限制,闪存硬盘利用一个软件层来模拟块设备的功能,并尽量使得擦除操作的延迟不为用户所见,这个软件层通常称为闪存转换层(FTL),它一般存储在 SSD 的 ROM 芯片中.FTL 的主要作用是将对一个数据页的写请求重新映射到一个已擦除的空白数据页上.因此,FTL 需要维护一个内部映射表来记录逻辑地址和物理地址之间的映射信息.该映射表在系统启动时构造,并在 SSD 的易失性存储器中进行维护.FTL 的实现细节与具体的设备相关,由制造商提供,对用户是透明的.

2.2 缓冲区置换策略

典型的计算机系统包含两层存储器,分别是主存(缓冲区)和辅存(外部存储介质,如磁盘或者 SSD).缓冲区的存取速度远快于辅存,二者一般使用相同大小的数据页.

已有基于磁盘的缓冲区置换策略[2-8]假定每次置换操作的代价相同,其目标是最小化缓冲区的缺页率.缺页率反映了必须从辅存读入缓冲区的数据页的比例.CLOCK[2]、FBR[3]、LRU-K[4]、2Q[5]、LIRS[6]、ARC[7]和 LRFU[8]等算法主要通过使用启发式方法来提高系统的性能,通过考虑数据页在缓冲区中的滞留时间和使用频率来减少缺页率.由于闪存的读写时间不对称,以上假设对于闪存来说是不成立的.因此,当设计基于闪存的缓冲区管理算法时,需要考虑读写的不对称性问题.

具有不对称存取时间的缓冲区置换问题可模拟成加权缓冲区问题,其目的是最小化请求序列的总代价.针对该问题,文献[15]提出了复杂度为$O(sn^2)$的最优离线算法,其中 s 表示缓冲区中数据页的个数,n 表示请求序列的长度.该问题可进一步归结为最小代价最大流问题[16]进行求解.由于该算法的时间和空间复杂度很高,即使提前知道完整的请求序列,其运行也需要耗费大量的时间和空间资源.对于在线算法,不可能提前知道任何未来的请求序列.研究者已经提出了许多在线的基于闪存的缓冲区管理算法.

基于闪存的缓冲区置换策略(FAB)[11]维护了一个块层 LRU 链表,同一物理块的数据页被聚集到一起.FAB 主要用在多数写请求都是顺序写的便携式媒体播放器上.

BPLRU[12]也维护了一个块层 LRU 链表.与 FAB 不同,BPLRU 使用 SSD 内部的 RAM 作为缓冲区,将随机写变成顺序写来提高写操作的效率和减少擦除操作的次数.

CFLRU[10]是利用闪存读写性能的不对称性提出一种优先置换只读页的缓冲区置换策略,这种策略假设闪存的写代价远远大于读代价.CFLRU 的基本思想如图2所示.其中 LRU 链表分成两个部分:工作区(Working Region)和置换区(Clean-First Region).每当发生缺页中断时,如果在置换区中存在只读的数据页,CFLRU 就会从中选择最近最少使用的只读页进行置换,如图2的 $p6$.只有当置换区中没有只读页时,才选择链表尾部的修改页 $p7$ 进行置换.置换区的大小是由参数 w 控制的.

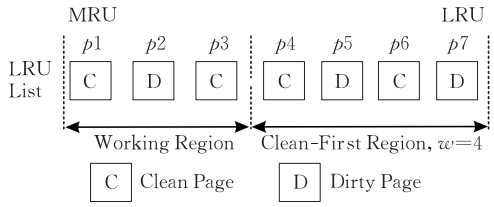


图 2　CFLRU 置换策略示意图

基于相同的思想,文献[13]将 CFLRU 置换区中的数据页根据其修改状态组织为不同的队列,从而可以将选择置换页操作的时间复杂度降为 $O(1)$. CFDC[14]通过对 CFLRU 置换区中的数据页进行重新组织来提升 CFLRU 算法的执行效率. 如图 3 所示,CFDC 的缓冲区也分为两部分,分别是工作区(Working Region)和置换区(Priority Region). 在 CFDC 的置换区中,根据数据页是否为修改页将其组织到两个队列中,其中只读页放在 Clean queue 中,所有的修改页放在不同的 Cluster 中,这些 Cluster 用 Dirty queue 进行组织,同一个集合中修改页的物理位置比较接近. 和 FAB 算法的块层 LRU 算法相比,CFDC 中的块大小是可变的.

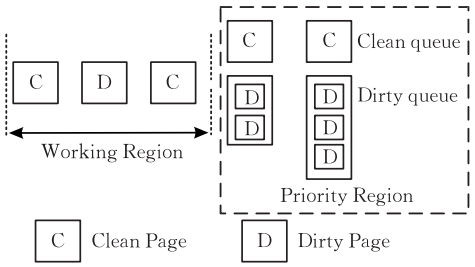


图 3　CFDC 置换策略示意图

3　FClock 算法

3.1　数据结构

如图 4 所示,FClock 将缓冲区中的数据页根据其读写状态组织为两个环形数据结构 CC 和 DC,分别维护只读页和修改页. 假定缓冲区可以存放 s 个数据页,则 $|CC \cup DC| = s \wedge CC \cap DC = \varnothing$. FClock 维护了一个全局计数器 $Counter$,每当发生一次数据页请求,$Counter$ 值加 1. 对于缓冲区中的每个数据页 p,FClock 为其关联 3 个变量:T、C 和 I,其中 T 表示 p 进入缓冲区的时间(当时的 $Counter$ 值),否则为最近一次被命中的时间);C 是 p 的访问位计数器,表示 p 被访问的频繁程度;I 表示 p 最近两次被命中之间对其它数据页访问的次数,称为"命中距离". 例如,假设数据页的请求序列为"r_1, r_2, r_3, r_4, r_1",则在访问完第一个 r_1 之后,$Counter=1$,因此 $r_1.T=1$,第二次访问完 r_1 后,$r_1.I = Counter - r_1.T - 1 = 3$,表示最近两次命中 r_1 之间对其它数据页进行了 3 次访问,$r_1.T = Counter = 5$. 另外 FClock 还为 CC 和 DC 各维护一个变量 F,用于表示 CC 和 DC 从最近一次命中数据页后到目前为止发生页缺失的次数. 图 4 中的虚线环用于处理循环和序列模式的数据页访问,称为子环,CC 和 DC 中的子环分别用 SC_{CC} 和 SC_{DC} 表示. 本文所用符号的意义如表 1 所示.

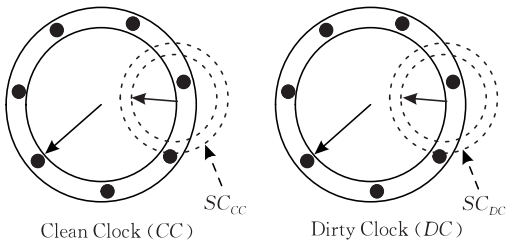


图 4　FClock 置换策略示意图

表 1　本文所用符号及其意义说明

符号名称	意义说明
CC	维护只读页的 clock(Clean Clock)
DC	维护修改页的 clock(Dirty Clock)
s	缓冲区可容纳的数据页个数
SC_{CC}	CC 中的子环
SC_{DC}	DC 中的子环
ζ	过去 s 次访问的命中页的平均访问间隔
C_r	从 SSD 读入一个数据页的代价
C_w	向 SSD 写入一个数据页的代价
M_{CC}	过去一段时间内(s 个数据页访问)CC 中数据页的物理操作次数
M_{DC}	过去一段时间内(s 个数据页访问)DC 中数据页的物理操作次数
R_{CC}	过去一段时间内(s 个数据页访问)CC 中数据页的逻辑操作次数
R_{DC}	过去一段时间内(s 个数据页访问)DC 中数据页的逻辑操作次数
$Counter$	全局计数器

3.2　基于代价的置换页选择策略

如果缓冲区满且当前请求的数据页 p 在缓冲区中,则可直接从缓冲区中访问此页;反之,FClock 置换策略将按照"代价"从 CC 或 DC 中选择一个页 x 进行置换,并从 SSD 读入数据页 p. FClock 的基本思想是:CC 和 DC 的大小应该和其在过去一段时间内由于数据页缺失所付出的代价成比例(式(1)). 假设缓冲区最多可放 s 个数据页,FClock 的置换策略可表述为:若 $|CC| < \beta s$,则 DC 过大,那么选择 DC 中指针所指的计数为 0 的数据页进行置换;反之从 CC 中选择计数为 0 的数据页进行置换. 本文中过去一段时间指过去 s 次访问,CC 的代价记为 C_{CC},DC 的代价记为 C_{DC}. 式(1)表示 CC 的大小占总缓冲区的比例.

$$\beta = C_{CC} / (C_{CC} + C_{DC}) \tag{1}$$

当数据页驻留缓冲区时,在连续两个物理操作之间可能发生多次逻辑操作. 尽管物理操作体现了

CC 和 DC 实际付出的代价，访问序列却是以逻辑操作的方式呈现的. 虽然逻辑操作和物理操作不同，但对于系统的性能来说都有影响. 一方面，物理操作对存取模式变化本身的反应比较迟钝；另一方面，虽然考虑逻辑操作可以快速侦测存取模式的变化，但对于存取模式剧烈变化的情况不太适用. 可见，单纯使用任何一种操作计算代价都不够全面.

为了最小化物理操作的代价，受文献[17]的启发，本文提出一种基于时钟数据结构并结合物理操作和逻辑操作优点的代价计算方案. 假定对数据页的存取是相互独立的，缓冲区可以存放 s 个数据页，n 是被处理文件中数据页的个数，则对某个数据页的逻辑操作在缓冲区中命中的概率是 s/n，而一个逻辑操作被转换为物理操作的概率是 $(1-s/n)$. 如式(2)和式(3)所示，本文提出的代价计算方案同时考虑了逻辑操作和物理操作.

$$C_{CC}=(R_{CC}\cdot(1-s/n)+M_{CC})\cdot C_r \tag{2}$$

$$C_{DC}=(R_{DC}\cdot(1-s/n)+M_{DC})\cdot(C_w+C_r) \tag{3}$$

当选择一个置换页时，通过考虑逻辑操作的影响，FClock 可以较快识别存取模式的变化并进行相应的调整. 而且，由于物理操作的影响也考虑在内，FClock 可以适应存取模式剧烈变化的情况.

如前所述，代价的计算基于过去一段时间内(s 个数据页访问)的统计结果. 本文通过使用一个最大长度为 s 的队列来记录过去一段时间数据页的访问情况，队列中每个元素 e 对应了一次数据页的访问，e 由两个数据成员组成：nC 和 $bHit$，其中 nC 的取值可以是 CC 或者 DC，表示相应的数据页请求发生在 CC 或者 DC 中，$bHit$ 表示对数据页的访问请求是否命中，以此来区别逻辑操作和物理操作. 基于该队列，可以在每次发生数据页请求时，在 $O(1)$ 代价的基础上更新 CC 和 DC 的代价值. 由于队列长度有限，因此维护该队列所需的内存非常有限. 代价的计算分为 3 步：(1) 去除头元素 e_h，(2) 加入新元素 e_t，(3) 根据以上介绍的方案计算代价 C_{CC} 和 C_{DC}. 其中第(1)步去除队头元素后，和第(2)步加入新元素后需要根据去除的元素和加入的新元素修改当前 CC 和 DC 对应的逻辑操作和物理操作的次数.

3.3　数据页访问位修改策略

虚存、数据库以及文件系统中经常会出现对同一个数据页多次连续访问后不再访问或者间隔较长时间后再访问的问题. 假设频繁访问的数据页之间的间隔大于两倍缓冲区大小. 原始的 clock 算法(即二次机会算法的改进)为每个数据页关联一个访问位，数据页初次进入缓冲区时，访问位置 0，当某个数据页在缓冲区中命中时，访问位置 1，当时钟指针扫过该页且其访问位为 1 时，访问位置 0. 显然这种方法会每隔一段时间(大于两倍缓冲区大小)将频繁访问的数据页置换出缓冲区. 而改进的 clock 算法要么在每次访问后都让访问位加 1，要么在加 1 的基础上为访问位设定一个最大值，都会导致短时间内频繁访问但以后不再访问的数据页长时间驻留内存. 因此，对于短时间内频繁访问的问题，本文提出用平均命中距离来解决这一问题.

定义 1(平均命中距离 ζ). 对于过去一段时间的访问序列“$r_1,r_2,\cdots,r_n$”而言，平均命中距离指该序列中所有被命中的数据页的命中距离的平均值，计算方法见式(4)，其中 m 指过去 n 次访问数据页时命中的次数.

$$\zeta=\frac{\sum_{i=1}^{m}r_i.I}{m} \tag{4}$$

例如，对于访问序列“$r_1,r_2,r_2,r_3,r_2,r_5,r_1,r_2$”而言，假设缓冲区大小为 8，初始状态为空. 当访问最后一个 r_2 时，可知在过去 8 次访问中，r_1 在第 7 次访问时被命中，其命中距离 $r_1.I=5$，而 r_2 在第 3、第 5 及第 8 次访问时被命中，其命中距离分别为 0、1 和 2. 由式(4)可知该序列过去 8 次访问的平均命中距离是 $(5+0+1+2)/4=2$.

FClock 使用平均命中距离来衡量某次命中是否为短时间内的频繁访问，其访问位修改策略可表述为：如果某个数据页的命中距离不小于平均命中距离 ζ，则该数据页的访问计数加 1，否则保持不变. 需要注意的是，虽然从表面上看该策略可能造成一直频繁访问的数据页的访问位永远不能增加的问题，实际上，如图 5 和算法 1 所示的过程，Hash 表中维护了数据页的有效访问位置，通过该变量，可知频繁访问数据页的访问位的值会得到慢慢增加. 这一点在算法 1 后面的例子中进行具体的说明.

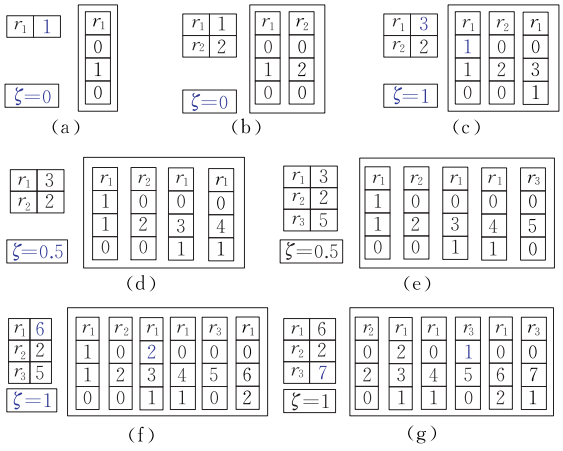


图 5　计算平均命中距离示意图

由于平均命中距离是基于过去一段时间的访问序列来进行计算的,每当执行一次新的数据页访问时,如果按照式(4)进行计算的话,显然时间复杂度太高.为此,本文提出一种在常量时间内计算平均命中距离的方法.

如图 6 所示,计算平均命中距离时使用两种数据结构,分别是队列和 Hash 表.队列用于维护过去一段时间内数据页的访问情况,本文实验中,过去一段时间指过去 s 次数据页的访问,s 是缓冲区中最多容纳的数据页个数,队列中元素的成员 I、T、C 的意义在 3.1 节进行了阐述.Hash 表用于记录在过去 s 次访问中队列中的数据页在队列中的有效位置,该有效位置用于计算每个数据页的命中距离以及整个缓冲区的平均命中距离 ζ.限于篇幅,这里用例子来说明如何在常量时间内计算平均命中距离 ζ.

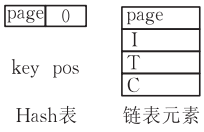


图 6　计算平均命中距离的数据结构

例如,对于访问序列"$r_1, r_2, r_1, r_1, r_3, r_1, r_3$",假设 $s=6$,初始情况下队列和 Hash 表均为空.为了说明的方便,Hash 表中数据页在队列中的位置用数据页的 T 值表示.该序列的处理过程如下:

(1)访问 r_1(图 5(a)),r_1 的 I、T 的值分别是 0、1,$\zeta=0$,然后将 r_1 加入 Hash 表中;

(2)访问 r_2(图 5(b)),r_2 的 I、T 值分别是 0、2、0,$\zeta=0$,然后将 r_2 加入 Hash 表中;

(3)访问 r_1(图 5(c)),r_1 的 I、T 值分别是 0、3,由于 r_1 命中,通过 r_1 在 Hash 表中找其在队列中的有效位置 1,得到有效位置处的 T 值 1,进而可以得到 r_1 的命中距离 1.由于 $1>\zeta=0$,将 Hash 表中 r_1 的有效位置对应的 I 值更新为 1,同时更新 Hash 表中 r_1 的有效位置更新为 3,最后计算平均命中距离 $\zeta=1$;

(4)访问 r_1(图 5(d)),r_1 的 I、T 值分别是 0、4,由于 r_1 命中,通过 r_1 在 Hash 表中找其在队列中的有效位置 3,得到有效位置处的 T 值 3,进而可以得到 r_1 的命中距离 0.由于 $0<\zeta=1$,则 Hash 表中 r_1 的有效位置及其对应的 I 值不变,最后计算平均命中距离为 $\zeta=0.5$;

(5)访问 r_3(图 5(e)),r_3 的 I、T 值分别是 0、5,由于 r_3 没有命中,将 r_3 及其有效位置 5 加入 Hash 表中,ζ 保持不变;

(6)访问 r_1(图 5(f)),r_1 的 I、T 值分别是 0、6.由于 r_1 命中,通过 r_1 在 Hash 表中找其在队列中的有效位置 3,得到有效位置处的 T 值 3,进而可以得到 r_1 的命中距离 2.由于 $2>\zeta=0.5$,则 Hash 表中 r_1 的有效位置 3 对应的 I 值变为 2,同时将 Hash 表中 r_1 的有效位置更新为 6,最后计算平均命中距离为 $\zeta=1$;

(7)访问 r_3(图 5(g)),r_3 的 I、T 值分别是 0、7.由于 r_3 命中,通过 r_3 在 Hash 表中找其在队列中的有效位置 5,得到有效位置处的 T 值 5,进而可以得到 r_3 的命中距离 1.由于 $1\geq\zeta=1$,则 Hash 表中 r_3 的有效位置 5 对应的 I 值变为 1,同时将 Hash 表中 r_3 的有效位置更新为 7,最后计算平均命中距离为 $\zeta=1$.

计算平均命中距离的方法在 FClock 算法中用 update(ζ)来表示.

3.4　FClock 算法

FClock 算法的具体流程如算法 1 所示.在缓冲区未满的初始阶段,即 $|CC\cup DC|<s$,如果请求页 p 没有命中(算法 1 中的 Case II),则根据对 p 的读(8～10 行)或者写(12～14 行)操作类型从 SSD 上读取到 CC 或者 DC 中,最后在第 15 行更新平均命中距离的值.如果 p 命中,即 $p\in CC\cup DC$,对应算法 1 的 Case I,如果 $p\in CC$(第 1 行),则 CC 的逻辑操作次数加 1,然后将其命中距离置 0;否则如果 $p\in DC$(第 2 行),则 DC 的逻辑操作次数加 1,然后将其命中距离置 0.如果操作类型是 write 并且 $p\in CC$(第 3 行),则将 p 从 CC 中移到 DC 中.在第 4 行,如果 p 的命中距离不小于平均命中距离,则将 p 的访问计数加 1.最后在第 5 行更新平均命中距离的值.

算法 1.　FClock(page p, type T).

/*FClock 在每次系统请求数据页 p 时被触发,T 表示对 p 的操作类型,可以是 read 或者 write */

Case I: $p\in CC\cup DC$　　/*p 被命中*/

1. if ($p\in CC$) then {$R_{CC}\leftarrow R_{CC}+1$; $CC.F\leftarrow 0$;}
2. else {$R_{DC}\leftarrow R_{DC}+1$; $DC.F\leftarrow 0$;}
3. if (T=write and $p\in CC$) then Move p to DC;
4. if ($p.I\geq\zeta$) then {$p.C\leftarrow p.C+1$; }
5. update(ζ);

Case II: $p\notin CC\cup DC$　　/*p 没有命中*/

6. if ($|CC\cup DC|=s$) then evictPage();
7. if (T=read) then
8. 　　$R_{CC}\leftarrow R_{CC}+1$; $M_{CC}\leftarrow M_{CC}+1$;
9. 　　fetch p from the disk;
10. 　　Insert (p,CC);
11. else
12. 　　$R_{DC}\leftarrow R_{DC}+1$;
13. 　　fetch p from the disk;
14. 　　Insert (p,DC);

15. update(ζ);

过程 1. evictPage()

1. $\beta \leftarrow C_{CC}/(C_{CC}+C_{DC})$

Case I. $|CC| < \beta s$ /*DC 过大,从 DC 中移除数据页*/

2. $M_{DC} \leftarrow M_{DC}+1$;

3. Let q be the page pointed by clock hand of DC (or SC_{DC});

4. while ($q.C>0$) do {$q.C \leftarrow q.C-1$; $q \leftarrow q->next$;}

5. write q's content to SSD; delete q;

Case II. $|CC| \geq \beta s$ /*CC 过大,从 CC 中移除数据页*/

6. Let q be the page pointed by clock hand of CC (or SC_{CC});

7. while ($q.C>0$) do {$q.C \leftarrow q.C-1$; $q \leftarrow q->next$;}

8. delete q;

过程 2. Insert(p,$clockx$).

/*p 为数据页,$clockx$ 可以是 CC 或者 DC */

1. if ($clockx.F \geq \lambda|clockx|$) then

/*λ 是调节因子,取值在[0,1]区间*/

2. add p to SC_{clockx};

3. else add p to $clockx$;

4. $clockx.F \leftarrow clockx.F+1$.

如果缓冲区已满,当数据页命中时,其操作和前面一段所介绍的内容相同. 如果数据页没有命中,则 FClock 会在算法 1 的第 6 行首先调用 evictPage() 从 CC 或者 DC 中选择一个数据页进行置换. evictPage 的具体操作见过程 1,其基本思想和计算方法已在 3.2 节中进行了说明. 然后在第 7～14 行根据对 p 的操作类型从 SSD 读入 p 并将其放入 CC 或者 DC 中,最后在第 15 行更新平均命中距离的值.

在 FClock 中,将数据页插入 CC(DC)时,调用了 Insert()过程,在第 1 行会判断 CC(DC)的 F 值(自从最近一次命中后发生的连续页缺失次数),如果 $F>\lambda|clockx|$,则 FClock 认为目前的数据页存取模式为序列存取,同时构造 CC(DC)的子环 SC_{CC}(SC_{DC}),并将 p 放入 SC_{CC}(SC_{DC})中;否则直接在第 3 行将 p 放入 CC(DC)中. 注意子环 SC_{CC}(SC_{DC})是 CC(DC)的一部分. 在第 4 行,将 CC(DC)的缺页数 F 加 1. 过程 Insert 中 λ 是调节因子,取值在[0,1]区间.

例如,对于访问序列"r_1,r_2,r_1,r_1,r_3,r_1,r_3"而言,其过程如下:

(1)访问 r_1,没命中. 从 SSD 上读入 r_1 并按照其操作类型放入 CC 或者 DC 中,同时在算法 1 的第 15 行使用 3.3 节介绍的方法更新 $\zeta=0$.

(2)访问 r_2,没命中. 处理过程同(1).

(3)访问 r_1,命中. 则在算法 1 的第 4 行根据 $r_1.I=1>\zeta=0$,则 $r_1.C$ 加 1,如图 5(c)所示,然后更新 ζ 的值为 1.

(4)访问 r_1,命中. 由于 $r_1.I=0<\zeta=1$,则 $r_1.C$ 保持不变,如图 5(d)所示,然后更新 ζ 的值为 0.5.

(5)访问 r_3,没命中. 和 r_2 的处理相同,如图 5(e)所示,ζ 保持不变,$r_3.C=0$.

(6)访问 r_1,命中. 注意这时 $r_1.I$ 的计算依赖于图 5(e)中的 Hash 表,可知 r_1 的有效位置为 3,进而可知 $r_1.I=2>\zeta=0.5$,因此 $r_1.C=r_1.C+1=2$,如图 5(f)所示,然后在算法 1 第 5 行更新 $\zeta=1$.

(7)访问 r_3,命中. 由于 $r_3.I=1\geq\zeta=1$,因此 $r_3.C$ 加 1,如图 5(g)所示,随后在算法 1 第 5 行更新 $\zeta=1$.

3.5 分 析

FClock 的自适应性体现在两方面:(1)基于代价的置换策略,当需要选择一个置换页时,FClock 从 CC 或者 DC 中根据各自的累加代价和公平地选择合适的置换页. 当读操作较多时,CC 会慢慢变大,相反,DC 会慢慢变大. 因此 FClock 能很好地处理同一闪存读写的不对称性以及不同闪存读写不对称性的巨大差异性,可以应用到不同类型的 SSD 上;另外,由于 FClock 在计算代价时同时考虑了物理操作和逻辑操作,FClock 可以适应不同的存取模式. (2)FClock 使用平均命中距离来控制数据页的引用计数,可以使得频繁访问的数据页的引用计数的值慢慢而不是快速增加,可以避免二次机会算法快速换出间隔较长时间后频繁访问的数据页被过早换出的问题,同时可以避免每次命中就加 1 的改进 CLOCK 算法所造成的无用数据页长时间驻留内存的问题. 和 CFLRU 及 CFDC 相比,FClock 考虑了引用计数,并且可以避免 LRU、CFLRU 及 CFDC 存在的锁争用问题.

FClock 可以很好地处理序列存取模式. 当需要处理序列引用时,FClock 使用 F 来检测 CC(DC)的页缺失次数,当达到一定程度时,即可认为出现了序列存取模式,这时,FClock 通过构造 CC(DC)的子环 SC_{CC}(SC_{DC})来处理新来数据页的插入和移除操作,从而不会对子环以外的数据页产生影响. 相比之下,CFLRU 及 CFDC 没有考虑存取模式的影响,这一点在第 4 节的实验结果部分也得到了证明.

当循环请求序列涉及的操作类型既包含读操作,也包含写操作时,FClock 可以很好地处理循环存取模式. 原因在于闪存读写代价不对称,而 FClock 根据代价而不是存取的先后顺序选择置换页,因此对于长循环而言,FClock 将打乱循环存取模式的置换顺序. 相比之下,CFLRU 和 CFDC 由于首先置换只读页,因此可以一定程度上打乱循环存取模式的置换顺序,但 FClock 的自适应性使得这种

打乱存取模式的行为具有自适应性,可以根据不同闪存的读写特点进行调整,而 CFLRU 和 CFDC 不具备这一特点,从而导致其性能下降,这一点在第 4 节的实验结果中也得到了进一步证明.

4　实　验

4.1　实验环境

本文的实验目的是验证 FClock 算法针对不同读写代价的 SSD 的有效性.我们选择两种 SSD 进行实验:(1)三星 MCAQE32G5APP,简便起见,用 FD1 表示;(2)三星 MCAQE32G8APP-0XA,用 FD2 表示.FD1 和 FD2 的随机读写的比率分别是1∶118 和 1∶2.这两个闪存硬盘的读写性能存在巨大的差异,这是因为 FD1 是由 MLC 类型的 NAND 芯片构成,而 FD2 是由 SLC 类型的 NAND 芯片构成.

对 SSD 来说,缓冲区置换算法的性能受物理读写次数的影响,然而 FTL 层的实现是设备相关的,由硬盘制造商提供,并没有为用户提供跟踪读写次数的接口.因此,我们选择使用模拟器[18]来进行测试.我们实现了 5 种置换策略来进行比较,即 LRU、CLOCK[2]、CFLRU[10]、CFDC[14] 及本文提出的 FClock.所有的置换策略都用 Visual C++实现的.我们将 CFLRU 算法中"置换区"的"窗口大小"设为缓冲区大小的 75%,将 CFDC 的"置换区"的"窗口大小"设为缓冲区的 50%,将 CFDC 的"聚类大小"设为 64.参数取自对应文献实验中所采用的数值.

我们将数据库的文件大小模拟为 64MB,相当于 32000 个的物理页,每页为 2KB.缓冲区的大小范围从 2000 个页到 8000 个页.本文实验中,模拟器假定数据页的大小是 2KB,每个数据块包含 64 个数据页.

我们生成了 4 种类型的测试数据,其统计数据如表 2 所示,其中"读/写比率"列中的"$x\%/y\%$"表示对某种测试数据来说,所有请求的 $x\%$为读操作、$y\%$为写操作;"局部性"列中的"$x\%/y\%$"表示对某种测试数据来说,在 $y\%$的页上有 $x\%$的操作.

表 2　实验所用测试数据的统计信息

编号	总的请求	读/写比例	局部性
T1	3000000	90%/10%	60%/40%
T2	3000000	80%/20%	50%/50%
T3	3000000	60%/40%	60%/40%
T4	3000000	80%/20%	80%/20%

表 1 中读写代价 C_r和 C_w可以通过 SSD 的技术手册得到,或者通过执行一定量的读写操作后取平均值来获得.本文实验所用数据来自于技术手册.

我们选择以下标准来评价缓冲区置换策略:(1)物理读操作的次数,(2)物理写操作的次数,(3)运行时间.其中运行时间是通过将读操作和写操作次数之和相加得到的.

4.2　性能比较和分析

4.2.1　读写操作代价差异巨大的 SSD 上性能比较

读操作性能比较. 图 7 展示了 4 种已有方法和基于本文提出的基于代价的 FClock 方法在 FD1 上运行 T1、T2、T3 及 T4 时随机读次数比较.可以看出,和 LRU 及 CLOCK 相比,基于闪存的算法

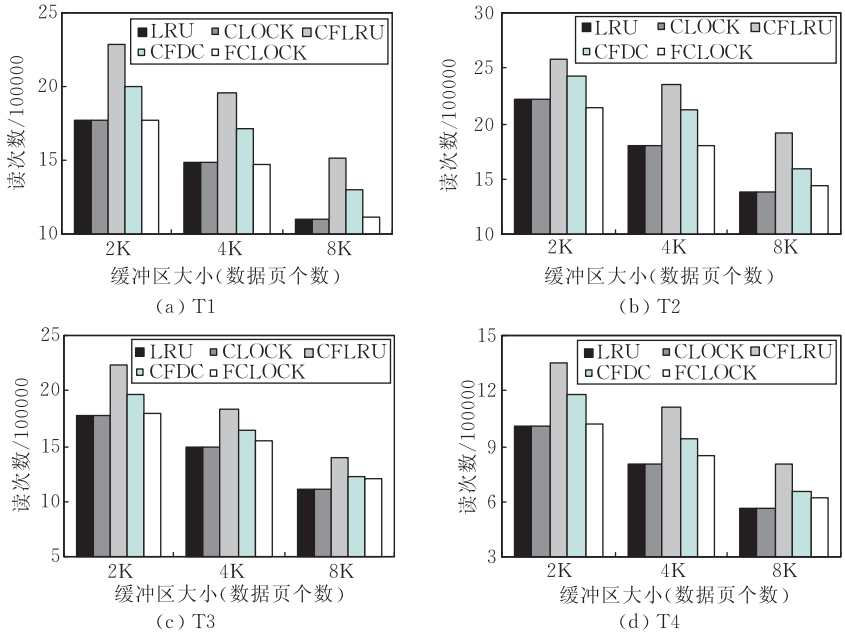


图 7　不同方法在 FD1 上运行 T1～T4 时读次数比较

(CFLRU、CFDC、FClock)需要更多的读次数,但本文提出的方法 FClock 所需的读次数在 T1 到 T4 上远少于 CFLRU 和 CFDC.可见,不考虑只读页的操作频率就直接进行置换导致 CFLRU 和 CFDC 需要付出很多不必要的物理读操作.

写操作性能比较. 图 8 展示了 4 种已有方法和基于本文提出的基于代价的 FClock 方法在 FD1 上运行 T1、T2、T3 及 T4 时随机写次数比较.可以看出,基于闪存的算法(CFLRU、CFDC、FClock)涉及的写操作的次数远少于基于磁盘的 LRU 和 CLOCK 算法.同时,尽管 CFLRU 首先置换只读页,本文提出的方法 FClock 依然好于 CFLRU,原因是进行置换时,由于 FD1 的读写比例差异巨大,本文方法将在缓冲区中保留更多的修改页.

运行时间比较. 图 9 展示了不同方法在 FD1 上运行 T1、T2、T3 及 T4 时运行时间的比较.可以看出,基于闪存的算法(CFLRU、CFDC、FClock)所需的运行时间远少于 LRU 和 CLOCK 算法.这是因为

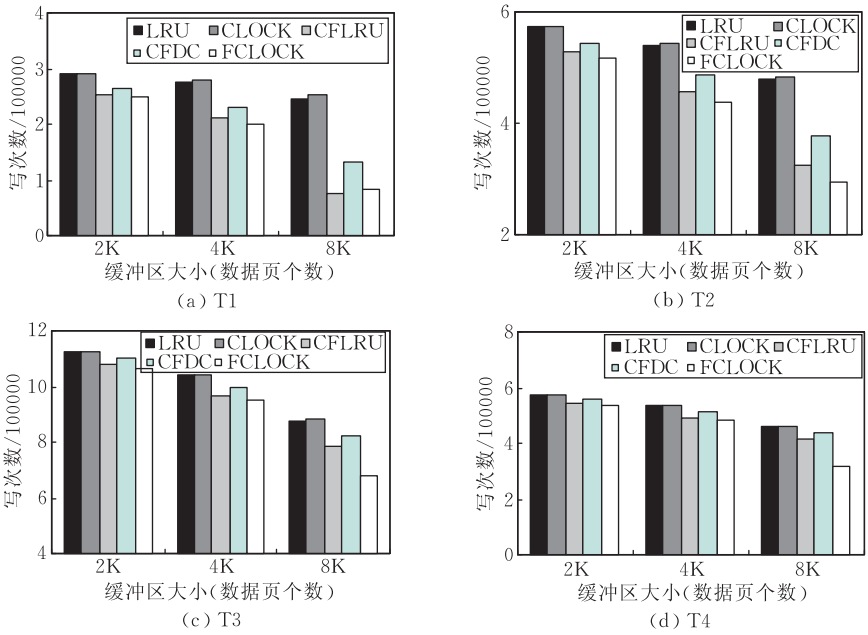


图 8　不同方法在 FD1 上运行 T1～T4 时写次数比较

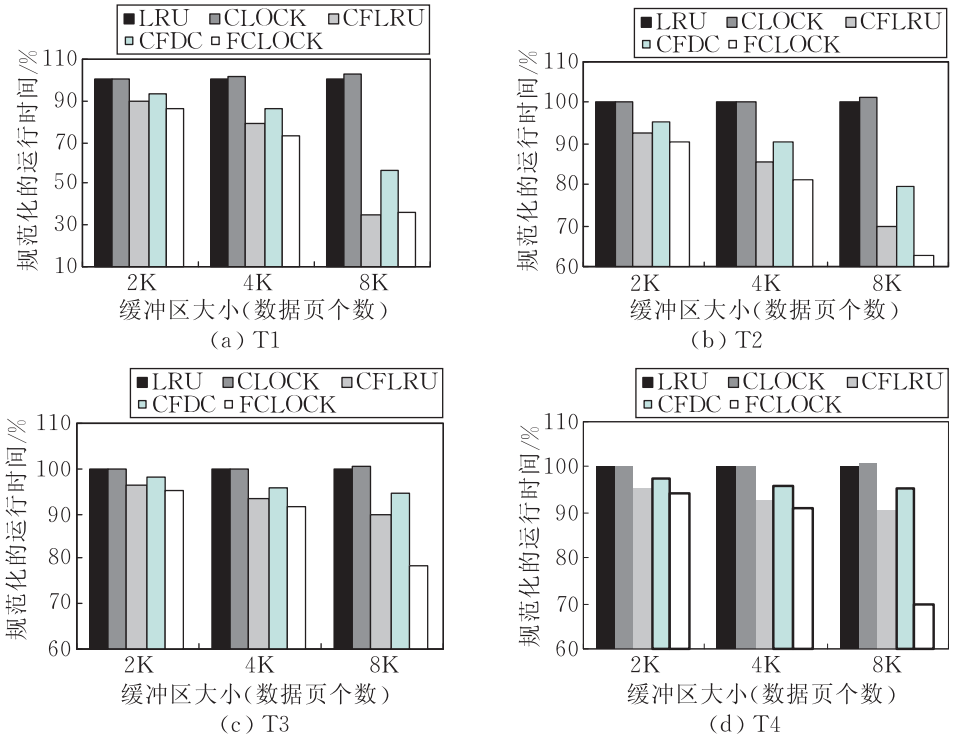


图 9　不同方法在 FD1 上运行 T1～T4 时运行时间比较

对于 FD1 来说,读写代价相差 118 倍,而且对基于闪存的置换算法来说,CFLRU 和 CFDC 优先置换只读页,FClock 算法给予写操作更高的权重,因此会大量减少写操作的次数,最终导致整体性能提升.

4.2.2　读写操作代价差异小的 SSD 上的性能比较

读操作性能比较. 图 10 展示了 4 种已有方法和基于本文提出的基于代价的 FClock 方法在 FD2 上运行 T1、T2、T3 及 T4 时随机读次数比较. 可以看出,和 LRU 及 CLOCK 相比,CFLRU 和 CFDC 由于优先置换只读页,因此需要更多的物理读操作. FClock 既考虑了不同读写状态的数据页操作代价,同时也能更好地处理数据页的存取模式,整体而言,虽然也给予写操作较大的权重,但依然能达到和 LRU 及 CLOCK 算法类似的读操作次数.

写操作性能比较. 图 11 展示了不同方法在 FD2 上运行 T1、T2、T3 及 T4 时随机写次数比较. 可以看出,由于 CFLRU 和 CFDC 无条件优先置换只读页,因此所需写操作的次数最少. 而本文提出的方法综合考虑存取模式及物理操作代价,因此在读写操作代价相差不大的情况下,与 CFLRU 和 CFDC 相比,将更多考虑读操作的权重,因此写操作的次数明显多于 CFLRU 和 CFDC,但仍然少于 LRU 和 CLOCK 算法.

运行时间比较. 图12展示了不同方法在FD2

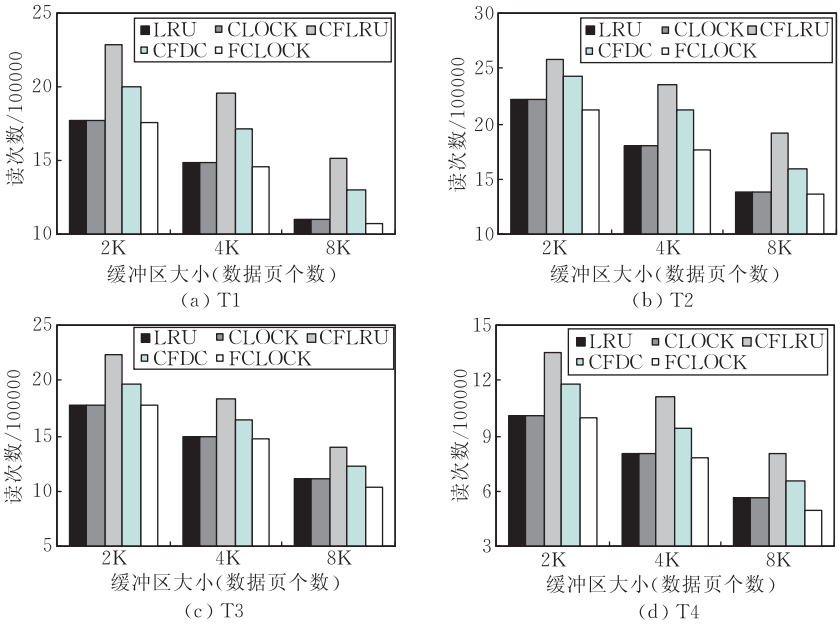


图 10　不同方法在 FD2 上运行 T1～T4 时读次数比较

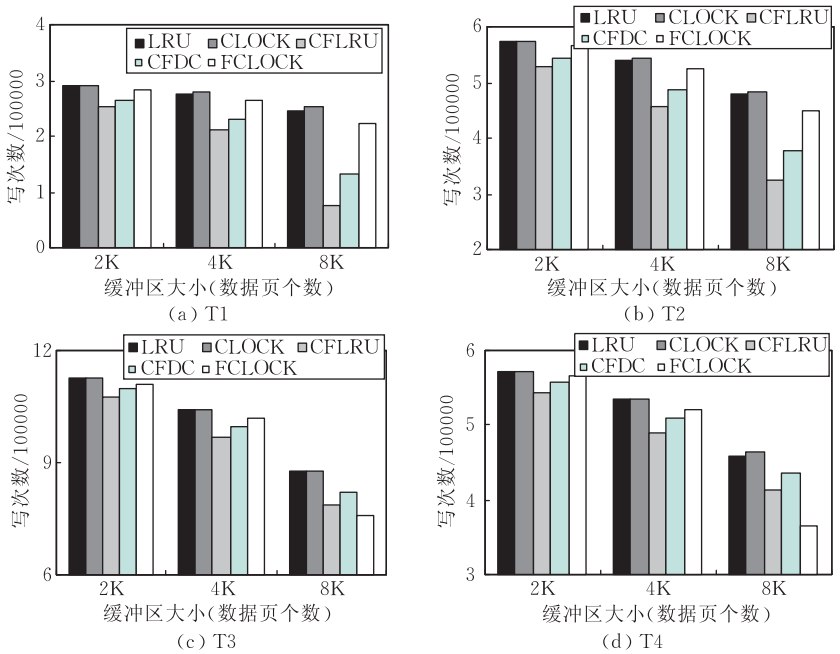


图 11　不同方法在 FD2 上运行 T1～T4 时写次数比较

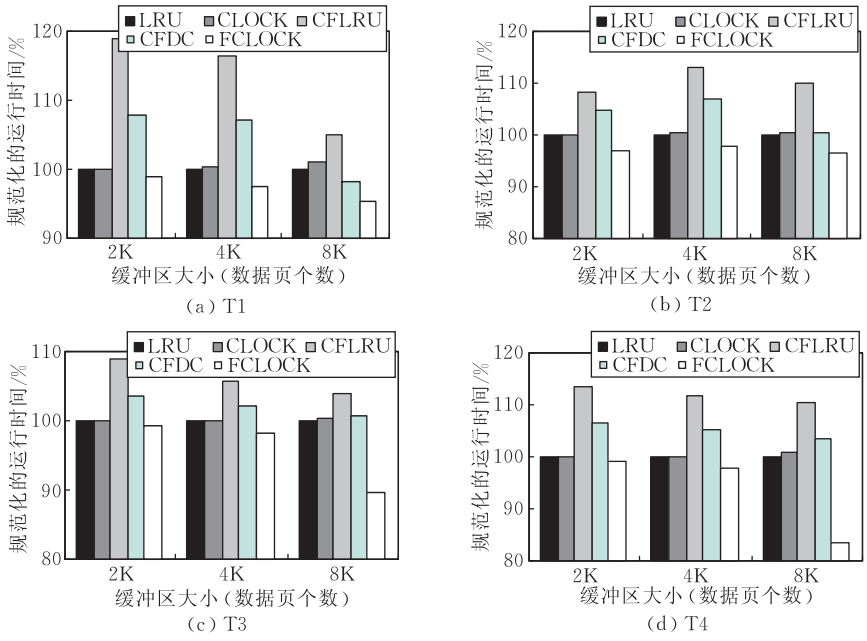


图 12 不同方法在 FD2 上运行 T1～T4 时运行时间比较

上运行 T1 及 T2 后的运行时间比较.可以看出,由于 FD2 读写操作的代价相差不大,而 CFLRU 和 CFDC 的读操作次数远远多于其他方法,因此二者所需的总运行时间远多于其他方法.而本文提出方法的读次数和 LRU 及 CLOCK 差不多,且写次数比 LRU 及 CLOCK 少,因此整体性能最好.

4.2.3 不同 SSD 硬盘的性能比较

通过比较图 7～图 12 可以看出,对于读写操作代价差异巨大的 SSD,如 FD1、LRU 和 CLOCK 的整体性能不如基于闪存的置换算法,但对于读写操作代价差异不大的 SSD,如 FD2、LRU 和 CLOCK 的整体性能要好于 CFLRU 及 CFDC,而本文提出的 FClock 在选择置换页时,根据操作的代价进行操作,可以在只读页和修改页之间进行公平的选择,因此可以适用于不同读写比例的 SSD.

5 结论和展望

针对现有基于闪存的缓冲区管理算法没有考虑不同闪存读写代价不对称性之间的巨大差异性问题以及 LRU 算法存在锁争用问题,本文提出一种基于闪存硬盘(SSD)的自适应缓冲区管理算法 FClock,FClock 将缓冲区中的数据页组织为只读环和修改环,使用基于代价的启发式来选择置换页,可在未修改的数据页和已修改的数据页之间进行公平的选择,适用于不同种类的 SSD 及存取模式.针对数据库、虚存和文件系统中数据页访问存在高相关性的特点,提出基于"平均命中距离"的访问计数方法来调整数据页的访问频率.基于不同 SSD 和不同存取模式的实验结果说明,FClock 的综合性能优于已有方法.

参 考 文 献

[1] Hwang C-G. Nanotechnology enables a new memory growth model. Proceedings of the IEEE, 2003, 91(11): 1765-1771

[2] Babaoglu O, Joy W. Converting a swap-based system to do paging in an architecture lacking page-reference bits. ACM SIGOPS Operating Systems Review, 1981, 15(5): 78-86

[3] Robinson J T, Devarakonda M V. Data cache management using frequency-based replacement//Proceedings of the ACM SIGMETRICS. Boulder, Colorado, USA, 1990: 134-142

[4] O'Neil E J, O'Neil P E, Weikum G. The LRU-K page replacement algorithm for database disk buffering//Proceedings of the ACM SIGMOD Conference. Washington, 1993: 297-306

[5] Johnson T, Shasha D. 2Q: A low overhead high performance buffer management replacement algorithm//Proceedings of the 20th International Conference on Very Large Data Base. Santiago de Chile, Chile, 1994: 439-450

[6] Jiang S, Zhang X. Making LRU friendly to weak locality workloads: A novel replacement algorithm to improve buffer cache performance. IEEE Transactions on Computers, 2005, 54(8): 939-952

[7] Megiddo N, Modha D S. ARC: A self-tuning, low overhead replacement cache//Proceedings of the FAST'03 Conference on File and Storage Technologies. San Francisco, California, USA, 2003: 115-130

[8] Lee D, Choi J, Kim J-H, Noh S H, Min S L, Cho Y, Kim C-S. LRFU: A spectrum of policies that subsumes the least

recently used and least frequently used policies. IEEE Transactions on Computers, 2001, 50(12): 1352-1361

[9] Effelsberg W, Haerder T. Principles of database buffer management. ACM Transactions on Database Systems, 1984, 9(4): 560-595

[10] Park S-Y, Jung D, Kang J-U, Kim J, Lee J. CFLRU: A replacement algorithm for flash memory//Proceedings of the 2006 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES). Seoul, Korea, 2006: 234-241

[11] Jo H, Kang J-U, Park S-Y, Kim J-S, Lee J. FAB: Flash-aware buffer management policy for portable media players. IEEE Transactions on Consumer Electronics, 2006, 52(2): 485-493

[12] Kim H, Ahn S. BPLRU: A buffer management scheme for improving random writes in flash storage//Proceedings of the 6th USENIX Conference on File and Storage Technologies. San Jose, CA, USA, 2008: 239-252

[13] Koltsidas I, Viglas S. Flashing up the storage layer//Proceedings of the VLDB Endowment. Auckland, New Zealand, 2008, 1(1): 514-525

[14] Ou Y, Haerder T, Jin P. CFDC: A flash-aware replacement policy for database buffer management//Proceedings of the 5th International Workshop on Data Management on New Hardware (DaMoN'09). Providence, Rhode Island, USA, 2009: 15-20

[15] Chrobak M, Karloff H J, Payne T H, Vishwanathan S. New results on server problems. SIAM Journal on Discrete Mathematics, 1991, 4(2): 172-181

[16] Cormen T H, Leiserson C E, Rivest R L, Stein C. Introduction to Algorithms. USA: The MIT Press, 2001

[17] Tang X, Meng X F. ACR: An adaptive cost-aware buffer replacement algorithm for flash storage divices//Proceedings of the 2010 Eleventh International Conference on Mobile Data Management (MDM). Kansas City, Missouri, USA, 2010: 33-42

[18] Jin P, Su X, Li Z. A flexible simulation environment for flashaware algorithms//Proceedings of the ACM Conference on Information and Knowledge Management. Hong Kong, China, 2009: 2093-2094

TANG Xian, born in 1978, Ph. D. candidate. Her research interests focus on flash database system.

MENG Xiao-Feng, born in 1964, professor, Ph. D. supervisor. His research interests include Web data management, Cloud data management, mobile data management, XML data management, Flash-aware DBMS, privacy protection.

Background

Flash disks are being widely used as an important alternative to conventional magnetic disks. Although accessed through the same interface by applications, their distinguished feature, i. e. different read and write cost in the aspects of time, makes it necessary to reconsider the design of existing replacement algorithms to leverage their performance potential.

Different from existing flash-aware buffer replacement policies that focus on the asymmetry of read and write operations, we address the "discrepancy" of the asymmetry for different flash disks, which is the fact that exists for a long time, while has drawn little attention by researchers since most existing flash-aware buffer replacement polices are somewhat based on the assumption that the cost of read operation is neglectable compared with that of write operation. In fact, this is not true for current flash disks on the market.

This paper proposes an adaptive replacement policy (FClock) which has two ring-shaped data structures, i. e. *CC* (their content remain unchanged) and *DC* (their content is modified), to manage clean pages and dirty pages in the buffer, respectively. When selecting a victim page, FClock uses cost-based heuristics to fairly make trade off between clean pages and dirty pages, and hence, can work well for different type of flash disks of large discrepancy. Further, for the problem of "correlated references" to database, virtual memory and file systems, this paper proposes a reference counter based on "average hit distance" to control the reference frequency. The experimental results on different traces and flash disks show that FClock not only adaptively tunes itself to workloads of different access patterns, but also works well for different kind of flash disks compared with existing methods.

This research was partially supported by the grants from the National Natural Science Foundation of China (No. 60833005); the National High Technology Research and Development Program (863 Program) of China (No. 2009AA011904); and the Doctoral Fund of Ministry of Education of China (No. 200800020002). This project aims at constructing the fundamental theory and design principles of flash-based database including a series of key problems such as system architecture, storage management and indexing, query processing, transaction processing, buffer management, etc. The work introduced in this paper belongs to buffer management and is very important for this project to construct flash-based databases.

HV-recovery：一种闪存数据库的高效恢复方法

卢泽萍 孟小峰 周 大
（中国人民大学信息学院 北京 100872）

摘 要 和磁盘相比，闪存作为一种新型的存储设备，具有读写速度快、抗震、省电、体积小等优点.因此，当前的研究普遍认为闪存将取代磁盘成为新一代的数据库二级存储设备.但是，由于闪存具有和磁盘不同的一些固有的读取特性，将当前基于磁盘设计的数据库直接移植到闪存上时，并不能充分发挥闪存设备的优越性.在数据库的恢复过程中，由于闪存的异地更新和重写之前先擦除的特性将带来大量高代价的小的随机写，直接使用传统的恢复方法在闪存数据库中就更难以充分利用闪存的优越性.因此，文中提出了一种对闪存中天然存在的数据的历史版本来进行管理和利用的恢复方法HV-recovery，来改进undo恢复的性能.通过和开源数据库Oracle Berkeley DB的比较，实验结果表明HV-recovery是原有的恢复算法性能的2～8倍，充分说明了其优越性.

关键词 闪存；闪存数据库；固态硬盘；恢复；日志
中图法分类号 **DOI号**：10.3724/SP.J.1016.2010.00000

HV-recovery：A High Efficient Recovery Technique for Flash-based Database

LU Ze-Ping MENG Xiao-Feng ZHOU Da
（*School of Information*，*Renmin University of China*，*Beijing* 100872）

Abstract Flash memory，as a new kind of data storage media，has a lot of attractive characteristics when compared with Hard Drive Disk（HDD）such as fast access speed，shock resistance，power saving，lighter form and low noise. Therefore flash memory is considered as the main storage device instead of disk in the next generation. However，traditional disk-based database can't take full advantage of high I/O performance of flash memory if we transfer it to flash memory without modification. The main reason is flash memory embraces different access characteristics with HDD. As for recovery，the situation becomes more serious because the out-of-place update model and erase-before-rewrite of flash memory lead to high cost of large quantity of minor random writes during the course of recovery. In this paper we proposed a recovery method，HV-recovery，to improve the performance of undo. HV-recovery makes use of the history versions of data which is naturally emerged in flash memory duo to the out-of-place update. Experimental results on Oracle Berkeley DB show that our HV-recovery outperforms traditional recovery in 2X～8X. The results demonstrate the high efficiency of our method.

Keywords flash memory；flash-based DBMS；SSD；recovery；logging

1 引 言

随着信息技术的飞速发展，数据呈爆炸性增长，海量的数据对数据库系统性能的要求也越来越高.而作为当前比较主流的二级存储介质，磁盘因为其内部的机械移动已经成为IO性能的瓶颈，越来越不能满足实际应用系统对数据存取带宽的需求.在

收稿日期：2010- - ；最终修改稿收到日期：2010- - .本课题得到国家自然科学基金（60833005，60573091）、国家"八六三"高技术研究发展计划项目基金（2007AA01Z155，2009AA011904）、教育部博士点基金项目（200800020002）资助.**卢泽萍**，女，1985年生，硕士，主要研究兴趣为闪存数据库系统.E-mail：luzeping_july@yahoo.com.cn.**孟小峰**，男，1964年生，教授，博士生导师，主要研究兴趣为Web数据管理、XML数据库、移动数据管理.**周 大**，男，1980年生，博士，主要研究方向为闪存数据库存储、索引和事务处理.

过去的 20 年里,CPU 处理速度增加了 570 倍,而磁盘的访问速度却只增加了 20 倍[1].可见,CPU 和主要二级存储器磁盘之间的带宽鸿沟已经成为了制约计算机系统处理能力提高的主要瓶颈.

值得庆幸的是,闪存作为一种新型的固态存储设备,由于其读写速度快、消耗电量低、抗震、小巧轻便等优点,已经受到越来越多的关注.随着容量的不断增大和单位价格的不断下降,许多研究者纷纷预测闪存将逐渐取代磁盘成为新的主流二级存储设备.图灵奖得主 Gray Jim 在 2005 年就曾预测说"就像磁盘取代磁带一样,闪存将会取代磁盘"[1].即使对现有的数据库不做任何改进,直接移植到闪存上,其性能也能提高大约 10 倍左右[2].

但是,由于闪存其固有的特性,若将现有的面向磁盘的传统数据库直接运行在闪存存储器上,还不能充分发挥闪存的优越性.因此,当前迫切需要将传统的数据库进行改进,让其更好的适应闪存本身的特点,以进一步提高闪存数据库的性能[3-4].

数据库开发和应用的实践表明,数据库恢复技术作为数据库系统中不可缺少的组成部分,对整个系统的性能影响是非常大的[5-7].在恢复过程中,为了对事务已经更改的数据项进行还原,通常需要对数据库中的一些已经赋予新值的数据进行重写.而这种大量的小随机重写操作,对闪存的代价是非常巨大的,不但浪费空间,而且非常耗时.因此,迫切需要一种高效且稳定性强的闪存数据库的恢复技术.

本文针对闪存存储器中天然存在的历史版本数据,提出了一种充分利用这些数据的历史版本,从而进行恢复的一种新型的恢复方法 HV-recovery.总的来说,本文所做的主要贡献如下:

(1)本文研究了闪存数据库中的恢复问题,并提出了新的适用于闪存的恢复方法.

(2)提供简单有效地恢复操作.有效地减少在恢复过程中容易出现的冗余写操作,从而大幅度减少恢复时间.

(3)优化日志结构.减少过多的日志冗余,从而提供高效的日志文件.

(4)提高空间利用率.减少大量垃圾数据的存在,从而提高存储设备中的空间利用率.

本文第 2 节介绍了闪存特殊的物理特性给恢复带来的挑战以及相关工作;第 3 节详细介绍了本文设计的 HV-recovery 的基本原理;第 4 节提出了怎样针对 HV-recovery 中的设计进行进一步的性能优化;第 5 节用分析及实验结果证明设计的优越性;最后第 6 节进行了总结.

2 问题定义及相关工作

闪存和磁盘读写特性的不同使得将现有的传统的数据库移植到闪存上时会出现的问题.下面将具体介绍在恢复中出现问题的原因和现有的一些改进方案及它们所存在的问题.

2.1 闪存存储器的物理特性

没有机械延迟.我们知道,在磁盘中,访问数据的时间主要用于移动磁头以及等待磁盘旋转.而闪存没有像磁头一样的机械部件,其随机访问模式和顺序访问模式的开销是相当的.这样,就可以把数据离散的分布,这并不会使访问的开销增加.

重写之前先擦除.众所周知,在磁盘中,如果需要更新数据,这些数据的新版本可以直接原地覆盖在旧版本所占有的地址上,这就是所谓的原地更新.可是在闪存中,在数据的旧版本没有被擦除前,是不能在原地写入新的版本的.也就是说,如果修改一个数据,就需要对整个块(通常为 64K 或 128K)上的数据进行擦除,这是代价非常巨大的.因此,在闪存中往往会采取异地更新的方式,即把数据的新版本写入另外的空闲空间中,而不直接在原地覆盖.

读写速度不一致.在闪存中,不同的访问操作的速度差别很大.一般来说,读的速度很快,写的速度略慢,擦除的速度最慢.因此,在设计新的基于闪存的数据结构中,应当尽量减少写操作和擦除操作,可以适当增加读操作,以整体上提高系统性能.

有限的擦除次数.虽然闪存中的块是可以进行反复擦除的,但每个块的擦除次数是有限的,一般为 10000～100000 次.因此,就必须尽量减少写入的次数,以间接减少擦除的次数,来延长闪存的使用寿命.

2.2 问题定义

在面向磁盘的数据库系统中,基于日志的恢复技术被广泛采用[5-7].不同的协议之下,日志记录的设计、日志/数据缓冲区的管理、检查点机制、记录日志和恢复的过程都很不一样.以最为常见的 undo 日志为例,当事务 T 需要将数据库元素 X 的取值 v 改变时,undo 日志就会将形如$\langle T,X,v\rangle$的日志记录记到磁盘上,当需要对事务 T 进行恢复,则需要在外存中重新写入 X 的值 v.

若将这个过程移到闪存上,举例来说,如果数据库中存在一个如表 1(a)所示的数据表,当需要将数据表中的 A 值由 v_1 修改为 v_2 时,就要写入一条新的记录,如表 1(b)所示得最后 1 行.而如果需要将 A

值进行恢复的时候，就要再写入一条其实早已存在于内存中的记录，如表 1(c)所示. 这就可以看出，最后 1 条记录和第 1 条记录是相同的. 也就是说，这其实是存在冗余的. 因此，闪存中通常存在着大量的数据的历史版本，而显式的恢复过程又不断的写入已经存在的数据项. 这是既浪费空间，又浪费时间的.

表 1　undo 日志在闪存数据库中存在的问题

(a) 数据表初始状态

	value	flag
A	v_1	1
B	v_b	1
C	v_c	1
…	…	…

(b) 修改 A 的取值后的表

	value	flag
A	v_1	0
B	v_b	1
C	v_c	1
A	v_2	1
…	…	…

(c) 对 A 值进行恢复后

	value	flag
A	v_1	0
B	v_b	1
C	v_c	1
A	v_2	0
A	v_1	1
…	…	…

同时，我们已经知道，闪存通常采取异地更新，但是因为闪存每次写的单位为页，即使是有所改进的闪存，其一页也通常只能写 4 次. 也就是说，不管一次要写入的数据量多大，至少需要占用闪存中四分之一个页的大小，而通常来说，一个需要恢复的数据项可能并没有这么大. 这样，就更带来了额外的空间浪费.

同时，这些额外的写操作会有较高的时间代价，并且因为一些额外的空间浪费，就会带来一些本不必要的擦除操作，其时间代价更为巨大. 因此，在恢复中所进行这种大量的小的随机重写对闪存的代价也是非常可怕的，这就需要设计新的恢复方法.

2.3　相关工作

随着技术的不断发展，闪存的优势越来越明显，有越来越多的研究关注于如何在基于闪存的数据库中提供更高的性能，其中较有影响力的工作包括 IPL[8]、FlashLogging[9]、Transactional Flash[10] 和 PORCE[11] 等.

IPL 彻底改变了闪存数据库中的存储结构，它将闪存上每个块中的页分为两个部分：数据页和日志页. 当对一个块中的数据页进行修改时，为了避免闪存的原地更新带来的巨大代价，IPL 只是将修改以日志的形式保存在其数据所在块的日志页中. 并且在日志区域满时，进行日志记录与数据的合并，来减少存储空间. 这种存储方式因为将对数据库的改变通过日志方式保存，可以间接的对数据库提供恢复. 但是，这需要对现有的数据库进行较大的修改，并不能方便的移植于不同的数据库中.

FlashLogging 提出了一种使用多个性价比高且更适合于日志的访问和存储模式的 USB 设备，来取代 SSD 记录日志. 因为 USB 的存储容量一般来说相对较小，FlashLogging 设计了一种轮转式的阵列组织方式来有效的管理这些分散的存放在 USB 设备中的日志记录，并提供恢复. 这种方法需要大量的 USB 设备阵列，并且会需要对多个 USB 设备进行读写，这是非常耗时的，而且随着 SSD 价格的不断下降，USB 设备的价格优势也在渐渐消失，因此，这并不是一个方便的系统搭建模式.

另外，在闪存中，若使用 FTL 层来屏蔽 Flash 的物理特性，则需要维护一个物理地址和逻辑地址的映射表，而将闪存作为嵌入式系统的存储设备时，则因为常常会出现断电的情况，就容易丢失这个映射表. 因此，PORCE 提供了一种在断电之后如何提供物理地址和逻辑地址映射的恢复方法. 而针对基于闪存的文件系统，SAC2006[12] 和 TOS2006[13] 提出了一种如何利用闪存的特性，来提供对基于闪存的文件系统的快速的载入和崩溃之后的恢复的方法. 然而，这些方法是针对于文件系统，而不是我们讨论的数据库系统，虽然设计思路上可以有较好的参考，但其性能并不能直接的与我们的设计相互比较.

3　HV-recovery

通过之前的分析，可以发现，在闪存存储设备中，在恢复时，完全没有必要用显式的回滚操作来重新写入数据元素在事务更新前的内容. 考虑到在闪存中数据项新旧版本的同时存在，可以利用旧版本来加快回滚和恢复的过程，而不需要发起更为昂贵的写操作来多次写入一个已经存在的数据项内容. 本文的设计就是考虑最大限度的利用数据项之前的历史版本来进行恢复. 图 1 是 HV-recovery 的一个整体的体系结构图.

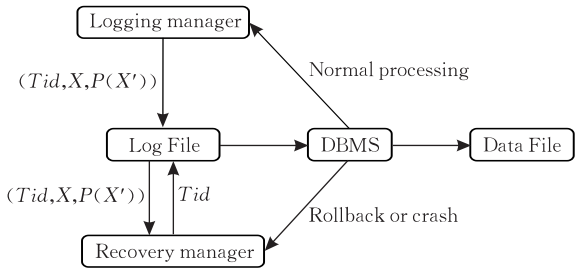


图 1 HV-recovery 体系结构图

在数据库的正常运行阶段，随着事务对数据文件的不断更改，日志管理器同时将形如(Tid, X, $P(X')$)的日志记录存放在日志文件中，当数据库发生崩溃或是要对事务进行回滚时，恢复管理器就根据得到的需要进行恢复的事务的 ID 读取日志文件，获得相关的日志记录，并根据这些日志记录进行恢复. 以下将具体介绍其中各部分的具体实现.

3.1 更新日志

在对日志记录进行管理时，HV-recovery 使用了一个版本列表 version_list 来保存日志记录，其结构如表 2 所示. 作为日志记录，其中存储的是被修改的各数据库元素的历史版本的地址信息，引起该数据库元素更改的事务的标识以及该数据项的旧值.

表 2 version_list 结构图

T_Id	Element	PreAddress	PreValue
T_1	X	$P(X')$	X'
T_2	A	$P(A')$	A'
T_4	D	NULL	NULL
T_4	B	$P(B')$	B'
T_4	B	Delete	NULL
T_6	C	$P(C')$	C'
T_2	Y	$P(Y')$	Y'
T_1	X	$P(X'')$	X''
T_1	*Commit*	NULL	NULL
…	…	…	…

当需要数据库系统对数据项进行更新时，在将新的数据版本写入新地址的同时，日志管理器会将这个数据项的旧值、旧地址以及该事务标识存入到 *version_list* 中. 并将其作为日志记录保存在永久性存储器中. 同时，*HV-recovery* 的日志记录类似于 *undo* 日志，必须遵守两条规则，也就是：

规则 A. 如果事务改变了数据库元素，则日志记录必须在数据库元素的新值写到二级存储器前写出.

规则 B. 如果事务提交，则其事务提交日志记录操作必须在事务改变的所有数据库元素已写到二级存储器之后再执行，但应尽快.

这样，在恢复中，只要对于在日志记录中显示为未提交的事务，对在日志记录中所保存的该事务所做的所有修改进行还原，就可以保证数据库对于事务所要求的 ACID 特性.

另外，在 HV-recovery 中，若同一个事务对某一个它已更新过的数据项又有新的更新，就增加一条新的日志记录，保存新的更新操作. 如表 2 的第 1 条记录和第 8 条记录所示.

而如果是不同事务对同一数据项进行修改，在满足数据库对于并发设计的要求的前提下，对日志记录部分来说，也是产生一条新的日志记录在 version_list 中. 如表 2 中，第 2 条记录和第 7 条记录所示，T_2 修改了 A 之后又修改了数据项 Y，那日志管理器就将这两次修改作为两条不同的日志记录来存储.

若数据库插入一个新的数据库元素，则与更改日志记录类似，产生一条新的日志记录，只是该日志记录的旧值和旧地址项被设置为空，以识别为插入操作. 如表 2 中第 3 条记录所示.

若数据库删除一个数据项，在写入原有的日志记录的同时，再增加一条日志记录，使用相同的事务标识和数据项，但是将其旧版本地址设置为一个定义了的删除标识. 如表 2 中第 4 条记录和第 5 条记录所示，T_4 删除了一个数据库元素 B，则为 T_4 和 B 增加两条新的日志记录，并把后一条日志记录的历史版本的值设置为 delete 标识，旧值设为空.

3.2 事务提交日志

在 HV-recovery 中，每当有事务提交，就在日志记录文件，也就是 version_list 中对该事务添加一个新的提交记录. 具体来说，就是 HV-recovery 为每个事务设置了一个 *commit* 元素，当某个事务提交时，就为该事务增加一条日志记录，在这个日志记录中，将该事务的 *commit* 元素的地址置为空. 如表 2 中第 9 条记录所示，当 T_1 事务提交，日志管理器就对 T_1 事务设置 *commit* 元素，并将该日志记录的旧版本地址字段设为空.

注意，因为本文的日志记录必须满足规则 B，也就是说，当事务提交日志记录到达二级存储器的时候，该事务所改变的所有数据库元素已经写到二级存储器上了. 因此，此时数据库已经提交了该事务的所有更新操作. 相反，如果事务提交日志记录未到达二级存储器，则在恢复时，不管这个事务的修改在数据库中完成了多少，这个事务所作的所有操作都将被还原，从而保证事务的原子性.

另外，日志文件中存在的已经提交的日志记录会增加日志文件的长度，同时，会使得对日志文件的访问变成随机模式. 但是，正如之前所介绍的，因为

闪存设备的随机读和连续读的访问时间的差异不明显，所以这些日志记录的存在对于其它事务的恢复效率的影响是非常微小，几乎可以不必考虑.

不过，这种日志记录长久保存是以大量的闪存存储空间为代价的，考虑到当前闪存存储设备的价格还不是很低廉，这会使得应用系统的成本价格提高，所以在第 4 节中，本文会提供一个进一步改进的方案.

3.3　恢复过程

当系统发生崩溃或者事务需要进行回滚时，由于 HV-recovery 对日志记录和数据更新的提交顺序满足规则 A 和规则 B，所以，只需要对在日志记录中体现为尚未提交的事务进行恢复.

首先，像其它恢复方式相同，先读入在数据库的二级永久性存储器中的需要恢复事务的日志记录. 对于同一个事务修改的同一个数据项的所有记录，选择所有记录中的第一条记录. 因为日志记录是顺序添加的. 而一个事务对某一数据项不断地更改，就不断的在后面添加新的日志记录，这就保证了其第一次保存的历史版本的地址恰好就是在该事务修改之前的数据项的值.

然后根据这些日志记录，恢复管理器读出需要恢复的各数据项的历史版本的地址，从纸质中取出其所存的数据，判断是否与日志记录中存的旧值相同，若相同，将已写入新更新数据内容的地址标识为无效，将原地址标识为有效，并将原地址赋给上层索引结构，从而完成恢复；若不同，则只有重新写入.

4　HV-recovery 方法的改进

为了进一步提高 HV-recovery 的性能表现，在本节中，针对 HV-recovery 中还存在的一些问题提供了一些改进的方法.

4.1　设立检查点

为了减少数据文件和日志文件对存储空间的浪费，本文采取了一种周期性设立检查点的措施，来有效的提高空间利用率.

4.1.1　对于日志文件的操作

首先，可以看出，HV-recovery 中的日志记录的更新是非常频繁的，随着事务的不断进行，需要不断地向日志文件中添加新的日志记录，而随着事务的不断提交，又使得大量的日志记录变为无效. 而在一般情况下，事务的回滚率通常不会太大，也就是说，其实在日志文件中存在着大量无效的日志记录，而这些日志记录的存在，增加了日志文件的长度，也占用了过多的闪存存储空间.

因此，在检查点中采取一种最简单的转移操作，将日志文件中尚有效的日志记录进行转移并进行整合. 也就是说，在设立的检查点中，先找到一个干净的块，然后逐条检查每条日志记录是否有效，在日志记录中选出尚有效的记录，写入到新的空闲块，当对某一旧日志块上的所有日志记录都检查过一遍后，就对该旧块进行擦除. 通过之前的介绍，可以看出，实际需要转移的日志记录相对于大量的已经提交的事务的日志记录而言，其数量是相当小的，这样其转移的代价也是可以接受的.

4.1.2　对于数据文件的操作

由于我们在闪存中采取异地更新，因此，为了显式的使用历史版本的数据，在实现时，在存储设备看来，我们将更新操作改为了插入，而删除操作只是记录了日志，并没有将历史数据删除，或标识为无效，这样虽然防止了我们将需要的历史版本的数据进行回收，但同时造成了系统中有过多的历史数据，存储空间大量浪费.

因此，在检查点时，我们会在对日志记录进行扫描的同时，将无效的日志记录中显示为被应该删除或替代掉的数据标识为无效，以便让垃圾回收机制对空间进行回收管理，提高空间利用率.

4.1.3　检查点时间间隔设置

而检查点的间隔时间的确定，是与闪存中日志文件的大小以及闪存的总存储空间有关的. 间隔时间太短会因为过多的擦除操作而浪费时间，并缩短闪存的寿命，而间隔时间过长又会浪费存储空间. 因此，可以根据闪存的总存储空间设计固定的可接受的日志文件的大小. 当到达某个阈值的时候就设立检查点开始进行转移操作.

同时，类似于 undo 日志的检查点，不但可以设立静态的检查点，也可以设置动态的检查点. 在检查点的开始阶段保存正在活跃的事务的 ID，这样，就可以不必等到所有事务都提交完毕后，再设立检查点，对日志文件进行转移. 并且也可以在数据库负载量较小的时候进行检查点的日志记录转移操作，从而进一步减少系统负担.

4.2　混合式存储系统

另外，也可以看出，HV-recovery 中对日志记录的主要操作就是一些小的追加写操作和一些擦除操作. 而之前的对闪存的硬件特性的介绍可以得知，这些操作对于闪存而言是非常昂贵的. 因此，可以看出，日志记录其本身的特点是不适用于闪存的.

而一般来说，基于闪存的数据库是指将大量的

对其操作较多的数据文件保存在闪存中，以利用其优越的读写速度来提供更好的数据库性能. 而当前的数据库一般都支持将日志文件和数据文件分开存储，因此，可以考虑在存储时使用混合式系统，如图 2 所示. 将数据记录存放在闪存上，同时将并不是非常适用于闪存的日志记录存放在磁盘中.

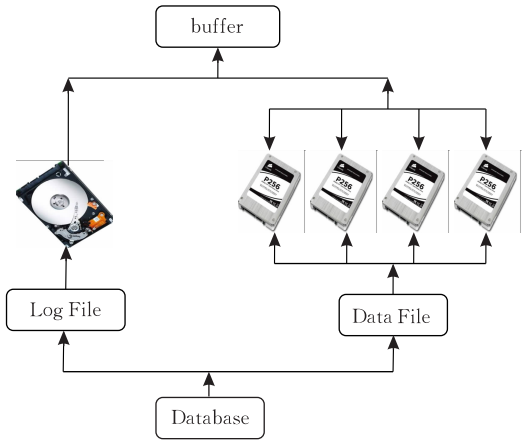


图 2 混合式存储系统结构图

通过这样的设计，就可以在不增加系统恢复算法复杂度的同时，节约大量的日志记录所占用的闪存空间，为数据库系统服务. 降低了数据库系统搭建的成本的同时，从整体上并没有影响数据库系统的性能表现.

5 实验结果及分析

本章通过对 HV-recovery 在闪存上和磁盘上的实际对比实验，从恢复时的写操作数、恢复时间等方面来证明 HV-recovery 的优越性.

5.1 实验环境

本文设计了两个实验平台，其中一个的存储设备配置了SSD，是 80G 的 Intel SSDSA2MH080G1GC，另一个配置的是磁盘，是 250G 7200rpm ST3250310AS，其中有 8M 的缓存. 除此之外，两个实验平台具有相同的配置，SSD 和磁盘都是通过 SATA 接口接入. CPU 是 Intel(R) Core(TM) 2 Duo CPU E8300 @ 2.83GHz，物理内存为 2GB，操作系统是 Windows XP Professional 2002 Service Park 2.

5.2 与相关工作的对比与分析

之前的介绍中提到，IPL 主要是给出一种新型的存储模式，以提供性能较高的对数据库的操作. 它所提出的恢复技术主要是针对这种新的存储方式的一种扩展，因此，IPL 不能像 HV-recovery 一样，方便的扩展到现有的大量数据库中. 同时，IPL 的存储方式是针对原始的闪存存储器的，而不是现在被简单广泛应用的 SSD 上的，因此，IPL 具有相当大的局限性.

同时，因为 IPL 的论文中并没有给出针对其恢复性能的实验结果，并且，其设计思想的实现必须在原始的闪存存储器上，而论文中也没有对其实现细节进行详细的阐述，这就对我们重现其工作带来了较大的困难，难以提供定量的对比结果.

另外，FlashLogging 给出在 TPCC 执行过程中突然崩溃需要扫描日志记录的时间，大约是基于磁盘的 2/3 左右，而 HV-recovery 的恢复时间是基于磁盘的 1/8 左右(在 5.4 中会详细阐述).

当然，两种设计的实验环境和对比细节不完全相同，把实验结果进行直接进行比较不是很具有说服力. 但是，由于 FlashLogging 要求搭建 USB 阵列，我们在短时间内较难重现，因此，我们在本文中难以给出直接的量化比较. 但是，从单纯的恢复时间的比较，我们至少可以相信，HV-recovery 的性能表现并不会比 FlashLogging 的表现差.

5.3 恢复时的写操作数

TPC 供了一系列系统性能的压力测试标准，其中的 TPCC 通过规定数据库原始数据生成以及查询负载的相关指导标准来模拟了 OLTP 的处理场景，是数据库系统事务处理性能的标准测试集.

TPCC 规定了在 OLTP 中典型的 5 种事务，包括 New-Order、Payment、Order-Status、Delivery 以及 Stock-Level，在下面的分析计算中，对这些事务类型分别简称为 T_1、T_2、T_3、T_4、T_5. 并且，TPCC 模拟实际情况，设定了这 5 种事务各自所占的比例，分别为 45%、43%、4%、4%、4%，不妨将这些值用 P_1、P_2、P_3、P_4、P_5 表示. TPCC 还根据实际情形为每一种事务定义了一系列的插入、删除和更新操作，不妨把每种事务的需要进行的操作数记为 N_1、N_2、N_3、N_4、N_5，如表 3 所示.

表 3 TPCC 关于 5 类事务的规定

Transaction	Id	Percent/%	Operations
New-Order	T_1	$P_1=45$	N_1
Payment	T_2	$P_2=43$	N_2
Order-Status	T_3	$P_3=4$	N_3
Delivery	T_4	$P_4=4$	N_4
Stock-Level	T_5	$P_5=4$	N_5

TPCC 模拟了大量用户同时对系统进行并发访问的模式，此处设存在的用户并发数为 N_{user}. 同时，容易理解，在任一时刻，正在并发执行的事务都完成了其中的一部分，也就是说，每个事务都有一个自己

的完成率，它是一个从 0～100％之间的一个随机数，记为 C.

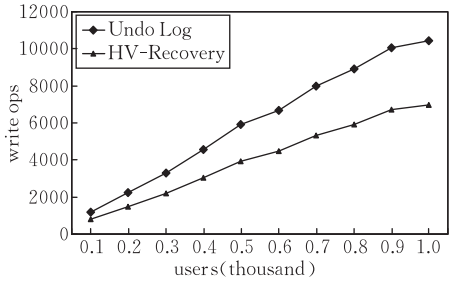
因此，可以知道，对一次有 N_{user} 个并发的 *TPCC* 测试而言，任一时刻系统发生崩溃，其需要恢复的数据量 N_{recovery} 为

$$N_{\text{recovery}}=\sum_{i=1}^{5}N_i\times\left(\sum_{j=0}^{N_{\text{user}}}T_j\times C_j\right),$$

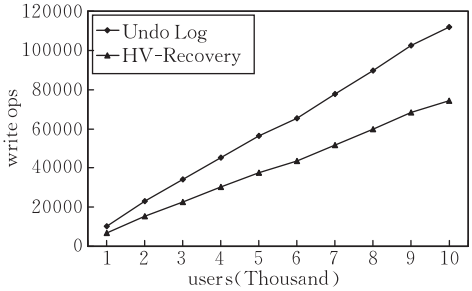
其中保证对每一种事务的相互比例满足 TPCC 的要求，也就是说，$T_1:T_2:T_3:T_4:T_5=P_1:P_2:P_3:P_4:P_5$.

通过之前的介绍可以看出，在恢复过程中，对于每一个需要进行恢复的数据项，HV-recovery 都可以比传统的恢复方式减少一次写操作，也就是说，HV-recovery 只需要 $2N_{\text{recovery}}$ 个写操作和 N_{recovery} 个读操作就可以完成恢复，而传统的恢复方式至少需要 $3N_{\text{recovery}}$ 个写操作.

实验结果如图 3(a)、(b)所示，可以发现，随着并发用户数的不断增加，与传统的 undo 日志相比，在数据库恢复阶段，HV-recovery 可以大量的减少写操作. 一般而言，在并发用户数从 100 增加到 10000 时，HV-recovery 可以减少大约有 400～37000 次写操作. 在减少写操作的同时，也节约了大量的闪存空间，提高了闪存的空间利用率.



(a) TPCC标准中的恢复更新数(1)



(b) TPCC标准中的恢复更新数(2)

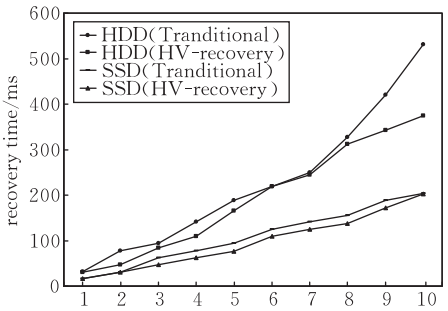
图 3　在 TPCC 标准下的实验结果

同时，因为写操作的时间代价比较大，从理论上分析，HV-recovery 可以在恢复时节省大量的时间，因此，我们将 HV-recovery 实现到现有的数据库中，用实验结果来验证其可以节省大量的恢复时间.

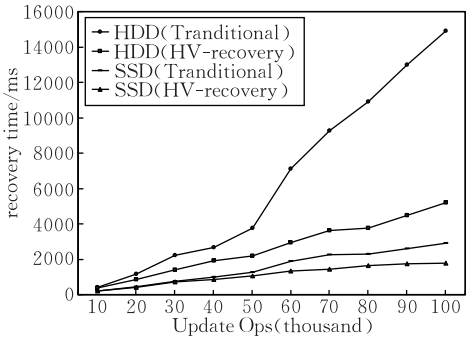
5.4　在 Berkeley DB 中实现 HV-recovery

为了方便的将 HV-recovery 在现有的数据库中进行实现，本文选择的是开源的 Oracle Berkeley DB 数据库，编程语言为 C 语言，编译环境是 Microsoft Visual Studio 2005，硬件环境如 5.1 节所述.

在实验中，在开始后不断地对数据库中的内容进行各种操作，然后，再显式的对事务回滚，即进行恢复操作，并记录恢复所耗费的时间，其结果如图 4 所示. 由实验结果可以看出，HV-recovery 相比于传统的基于磁盘的数据库有明显的优越性. 能够大幅度的减少恢复时间. 并且随着数据量的不断增加，这种优越性也越来越明显. 在较小的数据量时，如图 4(a)所示，使用传统数据库恢复技术所用的恢复时间平均是使用 HV-recovery 进行恢复的 2～3 倍，而数据量不断增大时，HV-recovery 的优势也成倍增加. 如图 4(b)所示，最好情况下，基于磁盘上的传统数据库的恢复时间是 HV-recovery 恢复时间的 8.3 倍，充分体现了 HV-recovery 的优越性.



(a) Berkeley DB中的恢复时间(1)



(b) Berkeley DB中的恢复时间(2)

图 4　在 Berkeley DB 中的实验结果

而且，与将传统的数据库直接移到 SSD 上的情况相比，HV-recovery 也体现了较大的优势. 一般而言，在 SSD 上，HV-recovery 要比 Berkeley DB 中传统的 undo 日志的方法要提高 20％，而随着数据量的增大，这种优势也体现的更为明显，如图 4(b)所示，某些情况下，HV-recovery 要比 Berkeley DB 中

传统的 undo 日志的方法要提高 38%，能很好的体现 HV-recovery 的优越性.

6 结 论

闪存即将取代磁盘成为下一代主流的二级存储器，但由于其特有的物理特性，目前基于磁盘设计的恢复技术不能充分的利用闪存的优越性. 为此，本文提出了一种新颖的具有较高性能的恢复方式 HV-recovery.

HV-recovery 对闪存中天然存在的数据的历史版本使用 version_list 结构加以管理和利用，提供高效的恢复. 通过周期性设立检查点，减小无效日志记录的长度，节约闪存空间. 引入混合式存储系统，将日志记录单独存放在磁盘上，以便对闪存数据库的恢复性能进一步的提高. 同时也保证了算法具有在数据库正常运行时有较小的开支、算法有比较强的可靠性、系统失败后恢复速度快和日志文件的空间需求较小等优势.

通过针对 TPCC 的分析及和开源数据库 Oracle Berkeley DB 对比实验看出，HV-recovery 比传统数据库的恢复时的写操作数可以减少接近一半，其恢复时间与传统数据库相比，能缩短到原来的大约 1/8，与在 SSD 上的传统数据库相比，也可以缩短 40%，充分显示了本算法的优越性.

参 考 文 献

[1] Gray Jim. Tape is dead disk is tape flash is disk RAM locality is king//Pacific Grove: Microsoft, Gong Show Presentation at Third Biennial Conference on Innovative Data Systems Research: 1, 2007

[2] Lee S W, Moon B, Park C, Kim J M, Kim S W. A case for flash memory SSD in enterprise database applications//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver, Canada, 2008: 1075-1086

[3] Lee Sang-Won, Moon Bongki, Park Chanik. Advances in flash memory SSD technology for enterprise database applications//Proceedings of the 35th SIGMOD International Conference on Management of Data. Providence, USA, 2009: 863-870

[4] Kim Yi-Reun, Whang Kyu-Young, Song Il-Yeol. Page-differential logging: An efficient and DBMS-independent approach for storing data into flash memory//Proceedings of the 2010 International Conference on Management of Data. Indianapolis, USA, 2010: 363-374

[5] Haerder T, Reuter A. Principles of transaction-oriented database recovery. ACM Computing Surveys, 1983, 15: 287-317

[6] Reuter A. Performance analysis of recovery techniques. ACM Transactions on Database Systems, 1984, 15: 526-559

[7] Garcia-Molina Hector, Ullman Jeffrey D, Widom Jennifer. Database System Implementation. USA: Prentice Hall, 1999

[8] Lee Sang-Won, Moon Bongki. Design of flash-based DBMS: An in-page logging approach//Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. Beijing, China, 2007: 55-66

[9] Chen Shimin. FlashLogging: Exploiting flash devices for synchronous logging performance//Proceedings of the 35th SIGMOD International Conference on Management of Data. Providence, USA, 2009: 73-86

[10] Prabhakaran Vijayan, Rodeheffer Thomas L, Zhou Lidong. Transactional flash//Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation. San Diego, USA, 2008: 147-160

[11] Chung Tae-Sun, Lee Myungho, Ryu Yeonseung, Lee Kangsun. PORCE: An efficient power off recovery scheme for flash memory. Journal of Systems Architecture: the EUROMICRO Journal, 2008, 54: 935-943

[12] Wu Chin-Hsien, Kuo Tei-Wei, Chang Li-Pin. Efficient initialization and crash recovery for log-based file systems over flash memory//Proceedings of the 2006 ACM Symposium on Applied Computing. Dijon, France, 2006: 896-900

[13] Wu Chin-Hsien, Kuo Tei-Wei, Chang Li-Pin. The design of efficient initialization and crash recovery for log-based file systems over flash memory. ACM Transactions on Storage, 2006, 2: 449-467

LU Ze-Ping, born in 1985, M. S. candidate. Her current interests include recovery of Flash-based database systems.

MENG Xiao-feng, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include web data management, native XML databases, mobile data management, etc.

ZHOU Da, born in 1980, Ph. D. candidate. His current interest include indexing, query processing and transaction processing of Flash-based database systems.

Background

Due to its superiority such as low access latency, low energy consumption, light weight, and shock resistance, the success of flash memory as a storage alternative for mobile computing devices has been steadily expanded into personal computer and enterprise server markets with ever increasing capacity of its storage. However, since flash memory exhibits its poor performance for small-to-moderate sized writes requested in a random order, existing database systems may not be able to take full advantage of flash memory without elaborate flash-aware data structures and algorithms.

We consider the recovery technique for flash based database systems. Now there is few researches on this issue, and the works has been published is either use a special storage structure or based on devices which are difficult to rebuild.

Our research group focuses on the design and implementation for flash-based database systems. We mainly solve the degradation of performance when we transfer the traditional database to flash memory without any modification. We have published several papers about index and buffer management for flash-based databases on internal and external conferences. And this work is the first one about recovery in our group. This paper is used for improve the recovery performance for flash-based databases.

In this paper, we proposed a new recovery method called HV-recovery to provide recovery in flash based database systems without too many changes on current databases. HV-recovery makes use of the history versions of data which is naturally emerged in flash memory duo to the out-of-place update. So it just needs to change the recovery component and logging component of databases, and could leave the others as it is. So HV-recovery is convenient to transfer to almost all the commerce databases. Meanwhile, HV-recovery is also effective. Experimental results on Oracle Berkeley DB show that our HV-recovery outperforms traditional recovery in 2X～8X. The results demonstrate the high efficiency of our method.

This research was partially supported by the grants from the National Natural Science Foundation of China under grant Nos. 60833005, 60573091; National High Technology Research and Development Program (863 Program) of China (Nos. 2007AA01Z155, 2009AA011904).

ViDE: A Vision-Based Approach for Deep Web Data Extraction

Wei Liu, Xiaofeng Meng, *Member*, *IEEE*, and Weiyi Meng, *Member*, *IEEE*

Abstract—Deep Web contents are accessed by queries submitted to Web databases and the returned data records are enwrapped in dynamically generated Web pages (they will be called *deep Web pages* in this paper). Extracting structured data from deep Web pages is a challenging problem due to the underlying intricate structures of such pages. Until now, a large number of techniques have been proposed to address this problem, but all of them have inherent limitations because they are Web-page-programming-language-dependent. As the popular two-dimensional media, the contents on Web pages are always displayed regularly for users to browse. This motivates us to seek a different way for deep Web data extraction to overcome the limitations of previous works by utilizing some interesting common visual features on the deep Web pages. In this paper, a novel vision-based approach that is Web-page-programming-language-independent is proposed. This approach primarily utilizes the visual features on the deep Web pages to implement deep Web data extraction, including data record extraction and data item extraction. We also propose a new evaluation measure *revision* to capture the amount of human effort needed to produce perfect extraction. Our experiments on a large set of Web databases show that the proposed vision-based approach is highly effective for deep Web data extraction.

Index Terms—Web mining, Web data extraction, visual features of deep Web pages, wrapper generation.

1 Introduction

The World Wide Web has more and more online Web databases which can be searched through their Web query interfaces. The number of Web databases has reached 25 millions according to a recent survey [21]. All the Web databases make up the deep Web (hidden Web or invisible Web). Often the retrieved information (query results) is enwrapped in Web pages in the form of data records. These special Web pages are generated dynamically and are hard to index by traditional crawler-based search engines, such as Google and Yahoo. In this paper, we call this kind of special Web pages *deep Web pages*. Each data record on the deep Web pages corresponds to an object. For instance, Fig. 1 shows a typical deep Web page from Amazon.com. On this page, the books are presented in the form of data records, and each data record contains some data items such as title, author, etc. In order to ease the consumption by human users, most Web databases display data records and data items regularly on Web browsers.

However, to make the data records and data items in them machine processable, which is needed in many applications such as deep Web crawling and metasearching, the structured data need to be extracted from the deep Web pages. In this paper, we study the problem of automatically extracting the structured data, including data records and data items, from the deep Web pages.

The problem of Web data extraction has received a lot of attention in recent years and most of the proposed solutions are based on analyzing the HTML source code or the tag trees of the Web pages (see Section 2 for a review of these works). These solutions have the following main limitations: First, they are Web-page-programming-language-dependent, or more precisely, HTML-dependent. As most Web pages are written in HTML, it is not surprising that all previous solutions are based on analyzing the HTML source code of Web pages. However, HTML itself is still evolving (from version 2.0 to the current version 4.01, and version 5.0 is being drafted [14]) and when new versions or new tags are introduced, the previous works will have to be amended repeatedly to adapt to new versions or new tags. Furthermore, HTML is no longer the exclusive Web page programming language, and other languages have been introduced, such as XHTML and XML (combined with XSLT and CSS). The previous solutions now face the following dilemma: should they be significantly revised or even abandoned? Or should other approaches be proposed to accommodate the new languages? Second, they are incapable of handling the ever-increasing complexity of HTML source code of Web pages. Most previous works have not considered the scripts, such as JavaScript and CSS, in the HTML files. In order to make Web pages vivid and colorful, Web page designers are using more and more complex JavaScript and CSS. Based on our observation from a large number of real Web pages, especially deep Web pages, the underlying structure of current Web pages is more complicated than ever and is far different from their layouts on Web browsers. This makes it more difficult for existing solutions to infer the regularity of the structure of Web pages by only analyzing the tag structures.

Meanwhile, to ease human users' consumption of the information retrieved from search engines, good template

- *W. Liu and X. Meng are with the School of Information, Renmin University of China, Beijing 100872, China. E-mail: gue1976@gmail.com, xfmeng@ruc.edu.cn.*
- *W. Meng is with the Department of Computer Science, Watson School of Engineering, Binghamton University, Binghamton, NY 13902. E-mail: meng@cs.binghamton.edu.*

Manuscript received 30 Dec. 2007; revised 12 Aug. 2008; accepted 16 Feb. 2009; published online 17 Apr. 2009.
Recommended for acceptance by V. Ganti.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2007-12-0629.
Digital Object Identifier no. 10.1109/TKDE.2009.109.

1041-4347/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

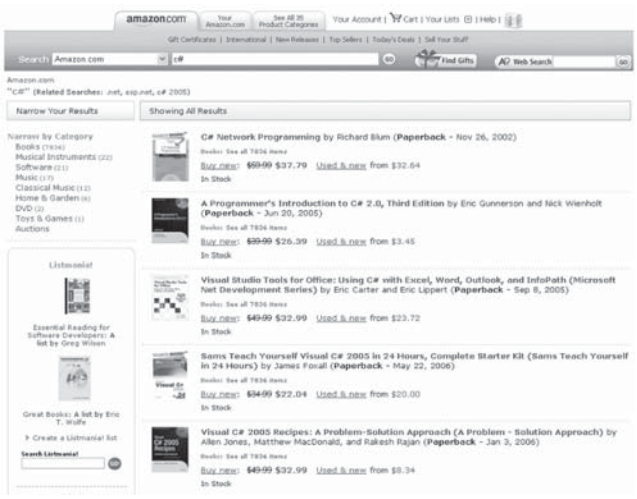


Fig. 1. An example deep Web page from Amazon.com.

designers of deep Web pages always arrange the data records and the data items with visual regularity to meet the reading habits of human beings. For example, all the data records in Fig. 1 are clearly separated, and the data items of the same semantic in different data records are similar on layout and font.

In this paper, we explore the visual regularity of the data records and data items on deep Web pages and propose a novel vision-based approach, Vision-based Data Extractor (ViDE), to extract structured results from deep Web pages automatically. ViDE is primarily based on the visual features human users can capture on the deep Web pages while also utilizing some simple nonvisual information such as data types and frequent symbols to make the solution more robust. ViDE consists of two main components, Vision-based Data Record extractor (ViDRE) and Vision-based Data Item extractor (ViDIE). By using visual features for data extraction, ViDE avoids the limitations of those solutions that need to analyze complex Web page source files.

Our approach employs a four-step strategy. First, given a sample deep Web page from a Web database, obtain its visual representation and transform it into a Visual Block tree which will be introduced later; second, extract data records from the Visual Block tree; third, partition extracted data records into data items and align the data items of the same semantic together; and fourth, generate visual wrappers (a set of visual extraction rules) for the Web database based on sample deep Web pages such that both data record extraction and data item extraction for new deep Web pages that are from the same Web database can be carried out more efficiently using the visual wrappers.

To our best knowledge, although there are already some works [3], [4], [23], [26], [28] that pay attention to the visual information on Web pages, our work is the first to perform deep Web data extraction using primarily visual features. Our approach is independent of any specific Web page programming language. Although our current implementation uses the VIPS algorithm [4] to obtain a deep Web page's Visual Block tree and VIPS needs to analyze the HTML source code of the page, our solution is independent of any specific method used to obtain the Visual Block tree in the sense that any tool that can segment the Web pages into a tree structure based on the visual information, not HTML source code, can be used to replace VIPS in the implementation of ViDE.

In this paper, we also propose a new measure, *revision*, to evaluate the performance of Web data extraction tools. It is the percentage of the Web databases whose data records or data items cannot be perfectly extracted (i.e., at least one of the precision and recall is not 100 percent). For these Web databases, manual revision of the extraction rules is needed to achieve perfect extraction.

In summary, this paper has the following contributions: 1) A novel technique is proposed to perform data extraction from deep Web pages using primarily visual features. We open a promising research direction where the visual features are utilized to extract deep Web data automatically. 2) A new performance measure, *revision*, is proposed to evaluate Web data extraction tools. This measure reflects how likely a tool will fail to generate a perfect wrapper for a site. 3) A large data set consisting of 1,000 Web databases across 42 domains is used in our experimental study. In contrast, the data sets used in previous works seldom had more than 100 Web databases. Our experimental results indicate that our approach is very effective.

The rest of the paper is organized as follows: Related works are reviewed in Section 2. Visual representation of deep Web pages and visual features on deep Web pages are introduced in Section 3. Our solutions to data record extraction and data item extraction are described in Sections 4 and 5, respectively. Wrapper generation is discussed in Section 6. Experimental results are reported in Section 7. Finally, concluding remarks are given in Section 8.

2 Related Work

A number of approaches have been reported in the literature for extracting information from Web pages. Good surveys about previous works on Web data extraction can be found in [16] and [5]. In this section, we briefly review previous works based on the degree of automation in Web data extraction, and compare our approach with fully automated solutions since our approach belongs to this category.

2.1 Manual Approaches

The earliest approaches are the manual approaches in which languages were designed to assist programmer in constructing wrappers to identify and extract all the desired data items/fields. Some of the best known tools that adopt manual approaches are Minerva [7], TSIMMIS [11], and Web-OQL [1]. Obviously, they have low efficiency and are not scalable.

2.2 Semiautomatic Approaches

Semiautomatic techniques can be classified into sequence-based and tree-based. The former, such as WIEN [15], Soft-Mealy [12], and Stalker [22], represents documents as sequences of tokens or characters, and generates delimiter-based extraction rules through a set of training examples. The latter, such as W4F [24] and XWrap [19], parses the document into a hierarchical tree (DOM tree), based on which they perform the extraction process. These approaches require manual efforts, for example, labeling some sample pages, which is labor-intensive and time-consuming.

2.3 Automatic Approaches

In order to improve the efficiency and reduce manual efforts, most recent researches focus on automatic approaches instead of manual or semiautomatic ones. Some representative automatic approaches are Omini [2], RoadRunner [8], IEPAD [6], MDR [17], DEPTA [29], and the method in [9]. Some of these approaches perform only data record extraction but not data item extraction, such as Omini and the method in [9]. RoadRunner, IEPAD, MDR, DEPTA, Omini, and the method in [9] do not generate wrappers, i.e., they identify patterns and perform extraction for each Web page directly without using previously derived extraction rules. The techniques of these works have been discussed and compared in [5], and we do not discuss them any further here. Note that all of them mainly depend on analyzing the source code of Web pages. As a result, they cannot avoid the inherent limitations described in Section 1. In addition, there are several works (DeLa [27], DEPTA, and the method in [20]) on data item extraction, which is a preparation step for holistic data annotation, i.e., assigning meaningful labels to data items. DeLa utilizes HTML tag information to construct regular expression wrapper and extract data items into a table. Similar to DeLa, DEPTA also operates on HTML tag tree structures to first align data items in a pair of data records that can be matched with certainty. The remaining data items are then incrementally added. However, both data alignment techniques are mainly based on HTML tag tree structures, not visual information. The automatic data alignment method in [20] proposes a clustering approach to perform alignment based on five features of data items, including font of text. However, this approach is primarily text-based and tag-structure-based, while our method is primarily visual-information-based.

The only works that we are aware of that utilize some visual information to extract Web data are ViNTS [30], ViPERS [25], HCRF [32], and VENTex [10]. ViNTs use the visual content features on the query result pages to capture content regularities denoted as Content Lines, and then, utilize the HTML tag structures to combine them. ViPER also incorporates visual information on a Web page for data records extraction with the help of a global multiple sequence alignment technique. However, in the two approaches, tag structures are still the primary information utilized, while visual information plays a small role. In addition, both of them only focus on data record extraction, without considering data item extraction. HCRF is a probabilistic model for both data record extraction and attribute labeling. Compared to our solution, it also uses VIPS algorithm [4] to represent Web pages, but the tag information is still an important feature in HCRF. And furthermore, it is implemented under an ideal assumption that every record corresponds to one block in the Visual Block tree, but this assumption is not always correct according to our observation to the real Web pages (about 20 percent of deep Web pages do not meet this assumption). VENTex implements the information extraction from Web tables based on a variation of the CSS2 visual box model. So, it can be regarded as the only related work using pure visual features. The main difference between our approach and VENTex is their objectives. VENTex aims to extract various forms of tables that are embedded in common pages, whereas our approach focuses on extracting regularly arranged data records and data items from deep Web pages.

TABLE 1
Font Attributes and Examples

Font factor	Example	Font factor	Example
Size	A (10pt)	underline	A
face	A(Sans Serif)	italic	*A*
color	A (red)	weight	**A**
strikethrough	~~A~~	frame	A

3 Visual Block Tree and Visual Features

Before the main techniques of our approach are presented, we describe the basic concepts and visual features that our approach needs.

3.1 Visual Information of Web Pages

The information on Web pages consists of both texts and images (static pictures, flash, video, etc.). The visual information of Web pages used in this paper includes mostly information related to *Web page layout* (location and size) and *font*.

3.1.1 Web Page Layout

A coordinate system can be built for every Web page. The origin locates at the top left corner of the Web page. The X-axis is horizontal left-right, and the Y-axis is vertical top-down. Suppose each text/image is contained in a minimum bounding rectangle with sides parallel to the axes. Then, a text/image can have an exact coordinate (x, y) on the Web page. Here, x refers to the horizontal distance between the origin and the left side of its corresponding rectangle, while y refers to the vertical distance between the origin and the upper side of its corresponding box. The size of a text/image is its height and width.

The coordinates and sizes of texts/images on the Web page make up the Web page layout.

3.1.2 Font

The fonts of the texts on a Web page are also very useful visual information, which are determined by many attributes as shown in Table 1. Two fonts are considered to be the same only if they have the same value under each attribute.

3.2 Deep Web Page Representation

The visual information of Web pages, which has been introduced above, can be obtained through the programming interface provided by Web browsers (i.e., IE). In this paper, we employ the VIPS algorithm [4] to transform a deep Web page into a Visual Block tree and extract the visual information. A Visual Block tree is actually a segmentation of a Web page. The root block represents the whole page, and each block in the tree corresponds to a rectangular region on

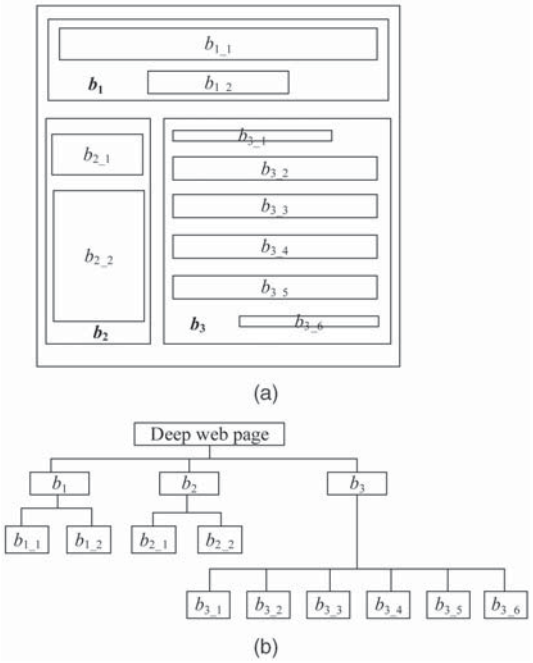


Fig. 2. (a) The presentation structure and (b) its Visual Block tree.

the Web page. The leaf blocks are the blocks that cannot be segmented further, and they represent the minimum semantic units, such as continuous texts or images. Fig. 2a shows a popular presentation structure of deep Web pages and Fig. 2b gives its corresponding Visual Block tree. The technical details of building Visual Block trees can be found in [4]. An actual Visual Block tree of a deep Web page may contain hundreds even thousands of blocks.

Visual Block tree has three interesting properties. First, block a contains block b if a is an ancestor of b. Second, a and b do not overlap if they do not satisfy property one. Third, the blocks with the same parent are arranged in the tree according to the order of the corresponding nodes appearing on the page. These three properties are illustrated by the example in Fig. 2. The formal representations for internal blocks and leaf blocks in our approach are given below. Each internal block a is represented as $a = (CS, P, S, FS, IS)$, where CS is the set containing its child blocks (note that the order of blocks is also kept), P is the position of a (its coordinates on the Web page), S is its size (height and width), FS is the set of the fonts appearing in a, and IS is the number of images in a. Each leaf block b is represented as $\mathrm{b} = (P, S, F, L, I, C)$, where the meanings of P and S are the same as those of an inner block, F is the font it uses, L denotes whether it is a hyperlink text, I denotes whether it is an image, and C is its content if it is a text.

3.3 Visual Features of Deep Web Pages

Web pages are used to publish information to users, similar to other kinds of media, such as newspaper and TV. The designers often associate different types of information with distinct visual characteristics (such as font, position, etc.) to make the information on Web pages easy to understand. As a result, visual features are important for identifying special

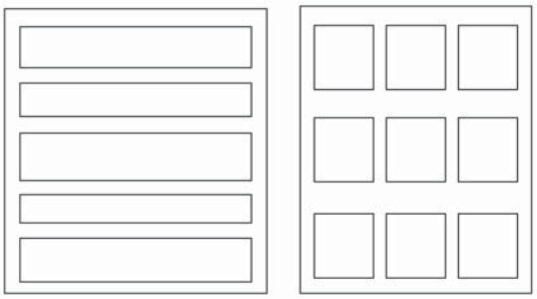


Fig. 3. Layout models of data records on deep Web pages.

information on Web pages. Deep Web pages are special Web pages that contain data records retrieved from Web databases, and we hypothesize that there are some distinct visual features for data records and data items. Our observation based on a large number of deep Web pages is consistent with this hypothesis. We describe the main visual features in this section and show the statistics about the accuracy of these features at the end of this Section 3.3.

Position features (PFs). These features indicate the location of the data region on a deep Web page.

- $PF1$: Data regions are always centered horizontally.
- $PF2$: The size of the data region is usually large relative to the area size of the whole page.

Since the data records are the contents in focus on deep Web pages, Web page designers always have the region containing the data records centrally and conspicuously placed on pages to capture the user's attention. By investigating a large number of deep Web pages, we found two interesting facts. First, data regions are always located in the middle section horizontally on deep Web pages. Second, the size of a data region is usually large when there are enough data records in the data region. The actual size of a data region may change greatly because it is not only influenced by the number of data records retrieved, but also by what information is included in each data record. Therefore, our approach uses the ratio of the size of the data region to the size of whole deep Web page instead of the actual size. In our experiments in Section 7, the threshold of the ratio is set at 0.4, that is, if the ratio of the horizontally centered region is greater than or equal to 0.4, then the region is recognized as the data region.

Layout features (LFs). These features indicate how the data records in the data region are typically arranged.

- $LF1$: The data records are usually aligned flush left in the data region.
- $LF2$: All data records are adjoining.
- $LF3$: Adjoining data records do not overlap, and the space between any two adjoining records is the same.

Data records are usually presented in one of the two layout models shown in Fig. 3. In Model 1, the data records are arranged in a single column evenly, though they may be different in width and height. LF1 implies that the data records have the same distance to the left boundary of the data region. In Model 2, data records are arranged in

TABLE 2
Relevant Visual Information about the Top Five Data Records in Fig. 1

	Images (pixel)	plain texts		link texts	
		Total font number	Shared font number	Total font number	Shared font number
record1	115*115	5	5	2	2
record2	115*115	5	5	2	2
record3	115*110	5	5	2	2
record4	115*115	5	5	2	2
record5	115*115	5	5	2	2

multiple columns, and the data records in the same column have the same distance to the left boundary of the data region. Because most deep Web pages follow the first model, we only focus on the first model in this paper, and the second model can be addressed with minor implementation expansion to our current approach. In addition, data records do not overlap, which means that the regions of different data records can be separated.

Appearance features (AFs). These features capture the visual features within data records.

- $AF1$: Data records are very similar in their appearances, and the similarity includes the sizes of the images they contain and the fonts they use.
- $AF2$: The data items of the same semantic in different data records have similar presentations with respect to position, size (image data item), and font (text data item).
- $AF3$: The neighboring text data items of different semantics often (not always) use distinguishable fonts.

$AF1$ describes the visual similarity at the data record level. Generally, there are three types of data contents in data records, i.e., images, plain texts (the texts without hyperlinks), and link texts (the texts with hyperlinks). Table 2 shows the information on the three aspects for the data records in Fig. 1. We can see that these five data records are very close on the three aspects. $AF2$ and $AF3$ describe the visual similarity at the data item level. The text data items of the same semantic always use the same font, and the image data items of the same semantic are often similar in size. The positions of data items in their respective data records can be classified into two kinds: *absolute position* and *relative position*. The former means that the positions of the data items of certain semantic are fixed in the line they belong to, while the latter refers to the position of a data item relative to the data item ahead of it. Furthermore, the items of the same semantic from different data records share the same kind of position. $AF3$ indicates that the neighboring text data items of different semantics often use distinguishable fonts. However, $AF3$ is not a robust feature because some neighboring data items may use the same font. Neighboring data items with the same font are treated as a *composite data item*. Composite data items have very simple string patterns and the real data items in them can often be separated by a limited number of symbols, such as ",", "/," etc. In addition,

Fig. 4. Illustrating visual features of deep Web pages.

the composite data items of the same semantics share the same string pattern. Hence, it's easy to break composite data items into real data items using some predefined separating symbols. For example, in Fig. 4, four data items, such as publisher, publishing date, edition, and ISBN, form a composite data item, and they are separated by commas. According to our observation to deep Web pages, the granularity of the data items extracted is not larger than what HTML tags can separate, because a composite data item is always included in one leaf node in the tag tree.

Content feature (CF). These features hint the regularity of the contents in data records.

- $CF1$: The first data item in each data record is always of a mandatory type.
- $CF2$: The presentation of data items in data records follows a fixed order.
- $CF3$: There are often some fixed static texts in data records, which are not from the underlying Web database.

The data records correspond to the entities in real world, and they consist of data items with different semantics that describe the attribute values of the entities. The data items can be classified into two kinds: *mandatory* and *optional*. Mandatory data items appear in all data records. For example, if every data record must have a title, then titles are mandatory data items. In contrast, optional items may be missing in some data records. For example, "discounted price" for products is likely an optional unit. The order of different types of data items from the same Web database is always fixed in data records. For example, the order of attributes of data records from Bookpool.com in Fig. 4 is "title," "author," "publisher," "publish time," "edition," "ISBN," "discount price," "save money," "availability," etc. Fixed static texts refer to the texts that appear in every data record. Most of them are meaningful labels that can help users understand the semantics of data items, such as "Buy new" in Fig. 4. We call these static texts static items, which are part of the record template.

Our deep Web data extraction solution is developed mainly based on the above four types of visual features. PF is used to locate the region containing all the data records on a deep Web page; LF and AF are combined together to extract the data records and data items.

Statistics on the visual features. To verify the robustness of these visual features we observed, we examined these features on 1,000 deep Web pages of different Web databases from the General Data Set (GDS) used in our

TABLE 3
The Statistics on the Visual Features

Feature type		Statistics	Feature type		Statistics
Position Features	*PF1*	99.9%	Appearance Features	*AF1*	99.5%
	PF2	99.9%		*AF2*	100%
				AF3	92.8%
Layout Features	*LF1*	99.3%	Content Features	*CF1*	100%
	LF2	100%		*CF2*	100%
	LF3	100%		*CF3*	6.5%

experiments (see Section 7 for more information about GDS). The results are shown in Table 3. For most features (except $AF3$ and $CF3$), their corresponding statistics are the percentages of the deep Web pages that satisfy them. For example, the statistics of 99.9 percent for $PF1$ means that for 99.9 percent of the deep Web pages, $PF1$ feature "data regions are always centered horizontally" is true. From the statistics, we can conclude that these visual features are very robust and can be reliably applied to general deep Web pages. For $AF3$, 92.8 percent is the percentage of the data items that have different font from their following data items. For $CF3$, 6.5 percent is the percentage of the static data items over all data items.

We should point out that when a feature is not satisfied by a page, it does not mean that ViDE will fail to process this page. For example, our experiments using the data sets to be described in Section 7 show that among the pages that violate $LF3$, 71.4 percent can still be processed successfully by ViDE, and among the pages that violate $AF1$, 80 percent can still be correctly processed.

3.4 Special Supplementary Information

Several types of simple nonvisual information are also used in our approach in this paper. They are *same text*, *frequent symbol*, and *data type*, as explained in Table 4.

Obviously, the above information is very useful to determine whether the data items in different data records from the same Web database belong to the same semantic. The above information can be captured easily from the Web pages using some simple heuristic rules without the need to analyze the HTML source code or the tag trees of the Web pages. Furthermore, they are specific language (i.e., English, French, etc.) independent.

TABLE 4
Nonvisual Information Used

Special complementary information	Remarks
Same text	Given two texts, we can determine whether or not they are the same.
Frequent symbol	Given the deep web pages of a web database, if some symbols/words (e.g., ISBN, $) appear in all the data items of an attribute, they are called frequent symbols.
Data type	They are predefined, including image, text, number, date, price, email, etc

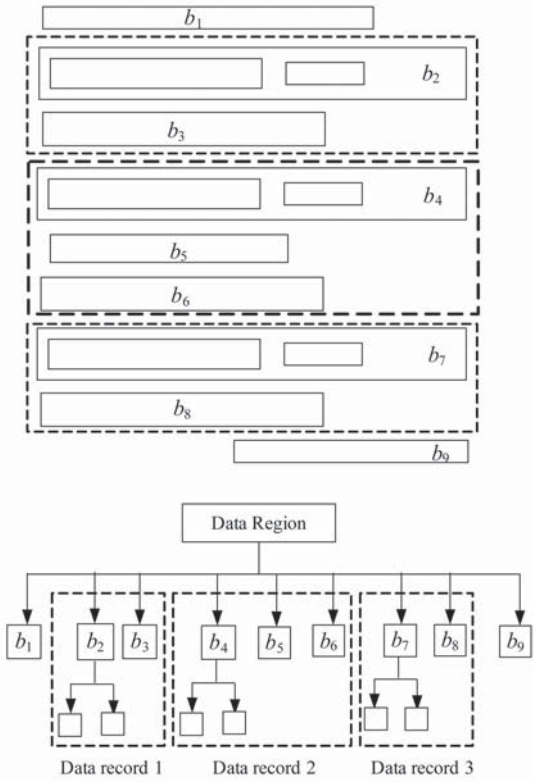


Fig. 5. A general case of data region.

4 Data Records Extraction

Data record extraction aims to discover the boundary of data records and extract them from the deep Web pages. An ideal record extractor should achieve the following: 1) all data records in the data region are extracted and 2) for each extracted data record, no data item is missed and no incorrect data item is included.

Instead of extracting data records from the deep Web page directly, we first locate the data region, and then, extract data records from the data region. $PF1$ and $PF2$ indicate that the data records are the primary content on the deep Web pages and the data region is centrally located on these pages. The data region corresponds to a block in the Visual Block tree. We locate the data region by finding the block that satisfies the two position features. Each feature can be considered as a rule or a requirement. The first rule can be applied directly, while the second rule can be represented by $(area_b/area_{page}) > T_{region}$, where $area_b$ is the area of block b, $area_{page}$ is the area of the whole deep Web page, and T_{region} is a threshold. The threshold is trained from sample deep Web pages. If more than one block satisfies both rules, we select the block with the smallest area. Though very simple, this method can find the data region in the Visual Block tree accurately and efficiently.

Each data record corresponds to one or more subtrees in the Visual Block tree, which are just the child blocks of the data region, as Fig. 5 shows. So, we only need to focus on the child blocks of the data region. In order to extract data

records from the data region accurately, two facts must be considered. First, there may be blocks that do not belong to any data record, such as the statistical information (e.g., about 2,038 matching results for java) and annotation about data records (e.g., 1, 2, 3, 4, 5 (Next)). These blocks are called *noise blocks* in this paper. Noise blocks may appear in the data region because they are often close to the data records. According to $LF2$, noise blocks cannot appear between data records. They always appear at the top or the bottom of the data region. Second, one data record may correspond to one or more blocks in the Visual Block tree, and the total number of blocks in which one data record contains is not fixed. In Fig. 5, block b_1 (statistical information) and b_9 (annotation) are noise blocks; there are three data records (b_2 and b_3 form data record 1; b_4, b_5, and b_6 form data record 2; b_7 and b_8 form data record 3), and the dashed boxes are the boundaries of data records.

Data record extraction is to discover the boundary of data records based on the LF and AF features. That is, we attempt to determine which blocks belong to the same data record. We achieve this in the following three phases:

1. Phase 1: Filter out some noise blocks.
2. Phase 2: Cluster the remaining blocks by computing their appearance similarity.
3. Phase 3: Discover data record boundary by regrouping blocks.

4.1 Phase 1: Noise Blocks Filtering

Because noise blocks are always at the top or bottom, we check the blocks located at the two positions according to $LF1$. If a block at these positions is not aligned flush left, it will be removed as a noise block. This step does not guarantee the removal of all noise blocks. For example, in Fig. 5, block b_9 can be removed in this step, while block b_1 cannot be removed.

4.2 Phase 2: Blocks Clustering

The remaining blocks in the data region are clustered based on their appearance similarity. Since there may be three kinds of information in data records, i.e., images, plain text, and link text, the appearance similarity between blocks is computed from the three aspects. For images, we care about the size; for plain text and link text, we care about the shared fonts. Intuitively, if two blocks are more similar on image size and font, they should be more similar in appearance. The formula for computing the appearance similarity between two blocks b_1 and b_2 is given below:

$$sim(b_1, b_2) = w_i * simIMG(b_1, b_2) + w_{pt} * simPT(b_1, b_2) + w_{lt} * simLT(b_1, b_2), \quad (1)$$

where $simIMG(b_1, b_2)$, $simIMG(b_1, b_2)$, and $simLT(b_1, b_2)$ are the similarities based on image size, plain text font, and link text font, respectively. And w_i, w_{pt}, and w_{lt} are the weights of these similarities, respectively. Table 5 gives the formulas to compute the component similarities and the weights in different cases. The weight of one type of contents is proportional to their total size relative to the total size of the two blocks.

A simple one-pass clustering algorithm is applied. First, starting from an arbitrary order of all the input blocks, take the first block from the list and use it to form a cluster. Next, for each of the remaining blocks, say b, compute its similarity with each existing cluster. Let C be the cluster that has the largest similarity with A. If $sim(b, C) > T_{as}$ for some threshold T_{as}, which is to be trained by sample pages (generally, T_{as} is set to 0.8), then add b to C; otherwise, form a new cluster based on b. Function $sim(b, C)$ is defined to be the average of the similarities between b and all blocks in C computed using (1). As an example, by applying this method to the blocks in Fig. 1, the blocks containing the titles of the data records are clustered together, so are the blocks containing the prices and so on.

TABLE 5
Formulas and Remarks

formulas	remarks
$simIMG(b_1,b_2)=\frac{Min\{sa_i(b_1),sa_i(b_2)\}}{Max\{sa_i(b_1),sa_i(b_2)\}}$	$sa_i(b)$ is the total area of images in block b. $sa_b(b)$ is the total area of block b. $fn_{pt}(b)$ is the total number of fonts of the plain texts in block b. $sa_{pt}(b)$ is the total area of the plain texts in block b. $fn_{lt}(b)$ is the total number of fonts of the link texts in block b. $sa_{lt}(b)$ is the total area of the link texts in block b.
$w_i=\frac{sa_i(b_1)+sa_i(b_2)}{sa_b(b_1)+sa_b(b_2)}$	
$simPT(b_1,b_2)=\frac{Min\{fn_{pt}(b_1),fn_{pt}(b_2)\}}{Max\{fn_{pt}(b_1),fn_{pt}(b_2)\}}$	
$w_{pt}=\frac{sa_{pt}(b_1)+sa_{pt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	
$simLT(b_1,b_2)=\frac{Min\{fn_{lt}(b_1),fn_{lt}(b_2)\}}{Max\{fn_{lt}(b_1),fn_{lt}(b_2)\}}$	
$w_{lt}=\frac{sa_{lt}(b_1)+sa_{lt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	

4.3 Phase 3: Blocks Regrouping

The clusters obtained in the previous step do not correspond to data records. On the contrary, the blocks in the same cluster all come from different data records. According to $AF2$, the blocks in the same cluster have the same type of contents of the data records.

The blocks need to be regrouped such that the blocks belonging to the same data record form a group. Our basic idea of blocks regrouping is as follows: According to $CF1$, the first data item in each data record is always mandatory. Clearly, the cluster that contains the blocks for the first items has the maximum number of blocks possible; let n be this maximum number. It is easy to see that if a cluster contains n blocks, these blocks contain mandatory data items. Our regrouping method first selects a cluster with n blocks and uses these blocks as seeds to form data records. Next, given a block b, we determine which record b belongs to according to $CF2$. For example, suppose we know that title is ahead of author in each record and they belong to different blocks (say an author block and a title block). Each author block should relate to the nearest title block that is ahead of it. In order to determine the order of different semantic blocks, a minimum bounding rectangle is

Algorithm block regrouping
Input: $C_1, C_2, \ldots, C_m$: a group of clusters generated by blocks clustering from a given sample deep web page P
Output: $G_1, G_2, \ldots, G_n$: each of them corresponds to a data record on P
Begin
//Step 1. sort the blocks in C_i according to their positions in the page (from top to bottom and then from left to right)
1 for each cluster C_i do
2 for any two blocks $b_{i,j}$ and $b_{i,k}$ in C_i //$1 \le j < k \le |C_i|$
3 if $b_{i,j}$ and $b_{i,k}$ are in different lines on P, and $b_{i,k}$ is above $b_{i,j}$
4 $b_{i,j} \leftrightarrow b_{i,k}$; //exchange their orders in C_i;
5 else if $b_{i,j}$ and $b_{i,k}$ are in the same line on P, and $b_{i,k}$ is in front of $b_{i,j}$
6 $b_{i,j} \leftrightarrow b_{i,k}$;
7 end until no exchange occurs;
8 form the minimum-bounding rectangle Rec_i for C_i;
//Step 2. initialize n groups, and n is the number of data records on P
9 $C_{max} = \{C_i \mid |C_i| = \max\{|C_1|, |C_2|, \ldots\ldots, |C_m|\}\}$; // $n = |C_{max}|$
10 for each block $b_{max,i}$ in C_{max}
11 Initialize group G_i;
12 put $b_{max,i}$ into G_i;
//Step 3. put the blocks into the right groups, and each group corresponds to a data record
13 for each cluster C_i
14 if Rec_i overlaps with Rec_{max} on P
15 if Rec_i is ahead of (behind) Rec_{max}
16 for each block $b_{i,j}$ in C_i
17 find the nearest block $b_{max,k}$ in C_{max} that is behind (ahead of) $b_{i,j}$ on the web page;
18 place $b_{i,j}$ into group G_k;
End

Fig. 6. The algorithm of blocks regrouping.

formed for each cluster on the page. By comparing the positions of these rectangles on the page, we can infer the order of the semantics. For example, if the rectangle enclosing all title blocks is higher than the rectangle enclosing the author blocks, then title must be ahead of its corresponding author. Based on this idea, the algorithm of block regrouping is developed as shown in Fig. 6.

This algorithm consists of three steps. Step 1 rearranges the blocks in each cluster based on their appearance order on the Web page, i.e., from left to right and from top to bottom (lines 1-7). In addition, a minimum bounding rectangle is formed for each cluster on the page (line 8). In Step 2, n groups are initialized with a seed block in each group as discussed earlier, where n is the number of blocks in a maximum cluster, denoted as C_{max}. According to $CF1$, we always choose the cluster that contains the first mandatory data item of each record as C_{max}. Let $b_{max,k}$ denote the seed block in each initial group Gk. Step 3 determines to which group each block belongs. If block $b_{i,j}$ (in C_i, C_i is not C_{max}) and block $b_{max,k}$ (in C_{max}) are in the same data record, then $b_{i,j}$ should be put into the same group $b_{max,k}$ belongs to. According to $LF3$, no two adjoining data records overlap. So, for $b_{max,k}$ in C_{max}, the blocks that belong to the same data record with $b_{max,k}$ must be below $b_{max,k-1}$ and above $b_{max,k+1}$. For each C_i, if data record R_i is ahead of R_{max}, then the block on top of R_i is ahead of (behind) the block on top of R_{max}. Here, "ahead of" means "on the left of" or "above," and "behind" means "on the right of" or "below." According to $CF2$, $b_{i,j}$ is ahead of

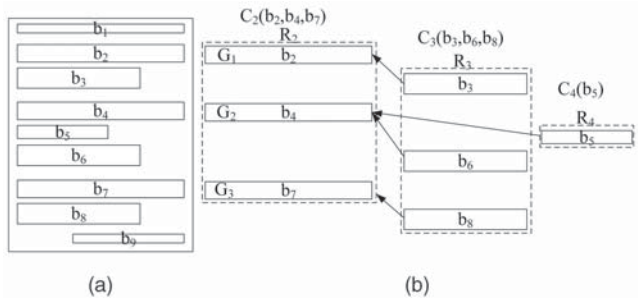


Fig. 7. An illustration of data record extraction.

$b_{max,k}$ if they belong to the same data record. So, we can conclude that if $b_{max,k}$ is the nearest block behind $b_{i,j}$, then $b_{i,j}$ should be put into the group $b_{max,k}$ belongs to. Obviously, the complexity of this algorithm is $O(n^2)$, where n is the number of data records in the sample page.

Example for data record extraction. Fig. 7 illustrates the case in Fig. 5. First, b_9 is removed according to $LF1$. Then, the blocks on the left in Fig. 7b are clustered using (1). Altogether, four clusters are formed and the blocks in them are also rearranged: $C_1\{b_1\}$, $C_2\{b_2, b_4, b_7\}$, $C_3\{b_3, b_6, b_8\}$, and $C_4\{b_5\}$. Next, C_2 is C_{max}, and b_2, b_4, and b_7 form three initial groups G_1, G_2, and G_3, respectively. Since R_3 and R_4 overlap with R_2 and R_3 is below R_2, we group b_3, b_6, and b_8 with b_2, b_4, and b_7 (the nearest block above it in C_2), respectively. At last, G_1 is $\{b_2, b_3\}$, G_2 is $\{b_4, b_5, b_6\}$, and G_3 is $\{b_7, b_8\}$. Each group forms a complete data record.

5 Data Item Extraction

A data record can be regarded as the description of its corresponding object, which consists of a group of data items and some static template texts. In real applications, these extracted structured data records are stored (often in relational tables) at data item level and the data items of the same semantic must be placed under the same column. When introducing CF, we mentioned that there are three types of data items in data records: mandatory data items, optional data items, and static data items. We extract all three types of data items. Note that static data items are often annotations to data and are useful for future applications, such as Web data annotation. Below, we focus on the problems of segmenting the data records into a sequence of data items and aligning the data items of the same semantics together.

Note that data item extraction is different from data record extraction; the former focuses on the leaf nodes of the Visual Block tree, while the latter focuses on the child blocks of the data region in the Visual Block tree.

5.1 Data Record Segmentation

$AF3$ indicates that composite data items cannot be segmented any more in the Visual Block tree. So, given a data record, we can collect its leaf nodes in the Visual Block tree in left to right order to carry out data record segmentation. Each composite data item also corresponds to a leaf node. We can treat it as a regular data item initially, and then, segment it into the real data items with the heuristic rules mentioned in $AF3$ after the initial data item alignment.

```
Algorithm data item matching
Input: item1, item2: two data items
Output: matched or unmatched: the match result (Boolean)
Begin
1 if (font(item1) ≠ font(item2))
2     Return unmatched;
3 if (position(item1) = position(item2))
4     return matched;
5 if (itemp1 and itemp2 are matched)    // itemp1 and itemp2 are the data
items immediately in front of item1 and item2 respectively
6       return matched;
7 else
      return unmatched;
End
```

Fig. 8. The algorithm of data item matching.

5.2 Data Item Alignment

$CF1$ indicates that we cannot align data items directly due to the existence of optional data items. It is natural for data records to miss some data items in some domains. For example, some books have discount price, while some do not.

Every data record has been turned into a sequence of data items through data record segmentation. Data item alignment focuses on the problem of how to align the data items of the same semantic together and also keep the order of the data items in each data record. In the following, we first define visual matching of data items, and then, propose an algorithm for data item alignment.

5.2.1 Visual Matching of Data Items

$AF2$ indicates that if two data items from different data records belong to the same semantic, they must have consistent font and position, including both absolute position and relative position. In Fig. 8, a simple algorithm to match two visually similar data items from different data records is described.

The first four lines of the algorithm say that two data items are matched only if they have the same absolute position in addition to having the same font. Here, absolute position is the distance between the left side of the data region and the left side of a data item. When two data items do not have the same absolute position, they can still be matched if they have the same relative position. For match on relative position, the data items immediately before the two input data items should be matched (from line 5 to line 6). As an example, for the two records in Fig. 4, the titles can be matched based on the absolute positions and the authors on the relative positions.

Because two data items of different semantics can also be visually similar, $AF2$ cannot really determine whether two data items belong to the same semantic. But the fixed order of the data items in the same data record ($CF2$) can help us remedy this limitation. So, we further propose an effective algorithm for data item alignment that utilizes both $CF2$ and $AF2$.

5.2.2 Algorithm for Data Item Alignment

$CF2$ says that the order of data items in data records is fixed. Thus, each data record can be treated as a sequence of data items. We can utilize this feature to align data items. Our goal is to place the data items of the same semantic in the same column. If an optional data item does not appear in a data record, we will fill the vacant position with a predefined blank item. Based on this insight, we propose a multialignment algorithm that can process all extracted data records holistically step by step. The basic idea of this algorithm is described as follows: Initially, all the data items are unaligned. We align data items by the order in their corresponding data records. When we encounter optional data items that do not appear in some data records, these vacant positions will be filled with the predefined blank item. This ensures that all data records are aligned and have the same number of data items at the end. Our data item alignment algorithm is shown in Fig. 9.

```
Algorithm data item alignment
Input: a set of extracted data records {ri | 1≤i≤n}
Output: a set of data records {ri | 1≤i≤n} with all the data items aligned
Begin
1  currentItemSet=ϕ;
2  currentCluster=ϕ;
//put the first unaligned data item of each ri into currentItemSet:
// Itemi^U(i) refers to the first unaligned item of the ith data record
3  currentItemSet ↔ Itemi^U(i) (1≤i≤n);
4  while currentItemSet≠ϕ
5      use the data item matching algorithm to group the data items
in currentItemSet into k clusters {Ci | 1≤i≤k} (k≤n);
6      for each cluster Ci
7        for each rj that does not have a data item in Ci
8          if Itemj^U(j)+k is matched with data items in Ci
9                Log position k;
10         else
11               Log position 0;
12   Pi = max value of these logged positions for Ci;
     /*Till now, each cluster Ci has a position Pi */
13   if any PL==0
14     currentCluster=CL;
15   else
16     currentCluster=CL whose PL is max {P1, P2,...,PK};
17   for each rj whose Itemj^U(j) is in currentCluster CL
18     remove Itemj^U(j) from currentItemSet;
19     if Itemj^U(j)+1 exists in rj
20           put Itemj^U(j)+1 into currentItemSet;
21   for each rj that has no item in currentCluster CL
22     insert a blank item ahead of Itemj^U(j) in rj;
23   U(j)++;
End
```

Fig. 9. The algorithm of data item alignment.

The input is n data records $\{r_1, r_2, \ldots, r_n\}$, and each data record r_i is denoted as a sequence of data items $\{item_i^1, item_i^2, \ldots, item_i^m\}$. Any data item has a unique position in its corresponding sequence according to the semantic order. In each iteration, we only process the next unaligned data item of every data record and decide which ones should be ahead of all others. The complexity of this algorithm is $O(n^2 * m)$, where n is the number of data records in the sample page and m is the average number of data items per data record.

Example for data item alignment. The example shown in Fig. 10 explains the process of data item alignment.

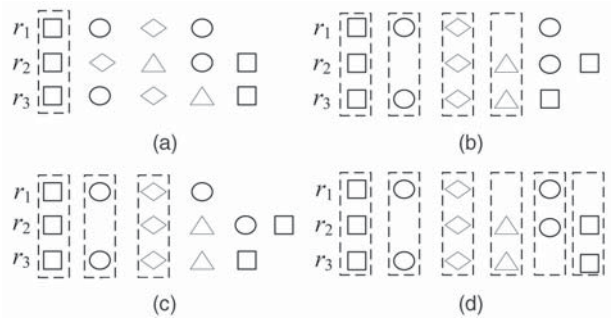


Fig. 10. An example of data item alignment.

Suppose there are three data records $\{r_1, r_2, r_3\}$ and each row is a data record. We use simple geometric shapes (rectangle, circle, triangle, etc.) to denote the data items. The data items represented by the same shape are visually matched data items. We also use $item_i^j$ to denote the jth data item of the ith data record. Initially (Fig. 10a), all current unaligned data items $\{item_1^1, item_2^1, item_3^1\}$ of the input data records are placed into one cluster, i.e., they are aligned as the first column. Next (Fig. 10b), the current unaligned data items $item_1^2, item_2^2, item_3^2$ are matched into two clusters $C_1 = \{item_1^2, item_3^2\}$ and $C_2 = \{item_2^2\}$ (line 5 in Fig. 9). Thus, we need to further decide which cluster should form the next column. The data items in C_1 can match $item_2^4$, and the position value 2 is logged (lines 6-12), which means that $item_2^4$ is the third of the unaligned data items of r_2. The data items in C_2 can match $item_1^3$ and $item_3^3$, and the position value 1 is logged (lines 6-12). Because 1 is smaller than 2 (line 16), the data items in C_1 should be ahead of the data items in C_2 and form the next column by inserting the blank item into other records at the current positions (lines 21-22). The remaining data items can be aligned in the same way (Figs. 10c and 10d).

6 Visual Wrapper Generation

ViDE has two components: ViDRE and ViDIE. There are two problems with them. First, the complex extraction processes are too slow in supporting real-time applications. Second, the extraction processes would fail if there is only one data record on the page. Since all deep Web pages from the same Web database share the same visual template, once the data records and data items on a deep Web page have been extracted, we can use these extracted data records and data items to generate the extraction wrapper for the Web database so that new deep Web pages from the same Web database can be processed using the wrappers quickly without reapplying the entire extraction process.

Our wrappers include data record wrapper and data item wrapper. They are the programs that do data record extraction and data item extraction with a set of parameter obtained from sample pages. For each Web database, we use a normal deep Web page containing the maximum number of data records to generate the wrappers. The wrappers of previous works mainly depend on the structures or the locations of the data records and data items in the tag tree, such as tag path. In contrast, we mainly use the visual information to generate our wrappers. Note that some other kinds of information are also utilized to enhance the performances of the wrappers, such as the data types of the data items and the frequent symbols appearing in the data items. But they are easy to obtain from the Web pages. We describe the basic ideas of our wrappers below.

TABLE 6
Explanation for (f, l, d)

Parameter	Value	Remarks
f	font	the font used by the data items of this attribute
l	Boolean	*True* denotes that the data items of this attribute are link texts
d	image, text, number, date, email, etc	the data type of this attribute

6.1 Vision-Based Data Record Wrapper

Given a deep Web page, vision-based data record wrapper first locates the data region in the Visual Block tree, and then, extracts the data records from the child blocks of the data region.

Data region location. After the data region R on a sample deep Web page P from site S is located by ViDRE, we save five parameters values (x, y, w, h, l), where (x, y) form the coordinate of R on P, w and h are the width and height of R, and l is the level of R in the Visual Block tree.

Given a new deep Web page P^* from S, we first check the blocks at level l in the Visual Block tree for P^*. The data region on P^* should be the block with the largest area overlap with R on P^*. The overlap area can be computed using the coordinates and width/height information.

Data record extraction. For each record, our visual data record wrapper aims to find the first block of each record and the last block of the last data record (denoted as b_{last}).

To achieve this goal, we save the visual information (the same as the information used in (1)) of the first block of each data record extracted from the sample page and the distance (denoted as d) between two data records. For the child blocks of the data region in a new page, we find the first block of each data record by the visual similarity with the saved visual information. Next, b_{last} on the new page needs to be located. Based on our observation, in order to help the users differentiate data records easily, the vertical distance between any two neighboring blocks in one data record is always smaller than d and the vertical distance between b_{last} and its next block is not smaller than d. Therefore, we recognize the first block whose distance with its next block is larger than d as b_{last}.

6.2 Vision-Based Data Item Wrapper

The data alignment algorithm groups data items from different data records into columns or attributes such that data items under the same column have the same semantic. Table 6 lists useful information about each attribute obtained from the sample page that can help determine which attribute a data item belongs to.

The basic idea of our vision-based data item wrapper is described as follows: Given a sequence of attributes

$\{a_1, a_2, \ldots, a_n\}$ obtained from the sample page and a sequence of data items $\{item_1, item_2, \ldots, item_m\}$ obtained from a new data record, the wrapper processes the data items in order to decide which attribute the current data item can be matched to. For $item_i$ and a_j, if they are the same on f, l, and d, their match is recognized. The wrapper then judges whether $item_{i+1}$ and a_{j+1} are matched next, and if not, it judges $item_i$ and a_{j+1}. Repeat this process until all data items are matched to their right attributes.

Note that if an attribute on a new page did not appear on the sample page, the data item of the attribute cannot be matched to any attribute. To avoid such a problem, several sample pages may be used to generate the wrapper. This can increase the chance that every attribute appears on at least one of these sample pages.

This process is much faster than the process of wrap-per generation. The complexity of data records extraction with the wrapper is $O(n)$, where n is the number of data records in the page. The complexity of data items extraction with the wrapper is $O(n * m)$, where n is the number of data records in the test page and m is the average number of data items per data record.

7 Experiments

We have implemented an operational deep Web data extraction system for ViDE based on the techniques we proposed. Our experiments are done on a Pentium 4 1.9 GH, 512 MB PC. In this section, we first describe the data sets used in our experiments, and then, introduce the performance measures used. At last, we evaluate both ViDRE and ViDIE. We also choose MDR [17] and DEPTA [29] to compare with ViDRE and ViDIE, respectively. MDR and DEPTA are the recent works on Web data record extraction and data item extraction, and they are both HTML-based approaches.

7.1 Data Sets

Most performance studies of previous works used small data sets, which are inadequate in assuring the impartiality of the experimental results. In our work, we use a large data set to carry out the experiments.

GDS. This data set is collected from CompletePlanet (www.completeplanet.com), which is currently the largest deep Web repository with more than 70,000 entries of Web databases. These Web databases are classified into 42 categories covering most domains in the real world. GDS contains 1,000 available Web databases. For each Web database, we submit five queries and gather five deep Web pages with each containing at least three data records.

Special data set (SDS). During the process of obtaining GDS, we noticed that the data records from two-thirds of the Web databases have less than five data items on average. To test the robustness of our approaches, we select 100 Web databases whose data records contain more than 10 data items from GDS as SDS.

Note that the deep Web pages collected in the testbed are the ones that can be correctly displayed by the Web browser we used. An example where a page is not correctly displayed is when some images are displayed as small red crosses. This will cause the positions of the texts on the result page to shift.

TABLE 7
Performance Measures Used in the Evaluation of ViDE

	precision	*recall*	*revision*
ViDRE	$\frac{DR_c}{DR_e}$	$\frac{DR_c}{DR_r}$	$\frac{WDB_t - WDB_c}{WDB_t}$
ViDIE	$\frac{DI_c}{DI_e}$	$\frac{DI_c}{DI_r}$	

7.2 Performance Measures

All previous works use *precision* and *recall* to evaluate their experimental results (some also include F-measure, which is the weighted harmonic mean of *precision* and *recall*). These measures are also used in our evaluation.

In this paper, we propose a new metric, *revision*, to measure the performance of an automated extraction algorithm. It is defined to be the percentage of the Web databases whose data records or data items are not perfectly extracted, i.e., either *precision* or *recall* is not 100 percent. This measure indicates the percentage of Web databases the automated solution fails to achieve perfect extraction, and manual revision of the solution is needed to fix this. An example is used to illustrate the significance of this measure. Suppose there are three approaches (A1, A2, and A3) which can extract structured data records from deep Web pages, and they use the same data set (five Web databases and 10 data records in each Web database). A1 extracts nine records for each site and they are all correct. So, the average *precision* and *recall* of A1 are 100 and 90 percent, respectively. A2 extracts 11 records for each site and 10 are correct. So, the average *precision* and *recall* of A2 are 90.9 and 100 percent, respectively. A3 extracts 10 records for four of the five databases and they are all correct. For the fifth site, A3 extracts no records. So, the average *precision* and *recall* of A3 are both 80 percent. Based on average *precision* and *recall*, A1 and A2 are better than A3. But in real applications, A3 may be the best choice. To make *precision* and *recall* 100 percent, all wrappers generated by A1 and A2 have to be manually tuned/adjusted, while only one wrapper generated by A3 needs to be manually tuned. In other words, A3 needs the minimum manual intervention.

Because our experiments include data record extraction and data item extraction, we define *precision*, *recall*, and *revision* for them separately.

In Table 7, DR_c is the total number of correctly extracted data records, DR_r is the total number of data records, DR_e is the total number of data records extracted, DI_c is the total number of correctly extracted data items, DI_r is the total number of data items, and DI_e is the total number of data items extracted; WDB_c is the total number of Web databases whose *precision* and *recall* are both 100 percent and WDB_t is the total number of Web databases processed.

7.3 Experimental Results on ViDRE

In this part, we evaluate ViDRE and also compare it with MDR. MDR has a similarity threshold, which is set at the default value (60 percent) in our test, based on the suggestion of the authors of MDR. Our ViDRE also has a

TABLE 8
Comparison Results between ViDRE and MDR

	dataset	*precision*	*recall*	*revision*
ViDRE	GDS	98.7%	97.2%	12.4%
	SDS	98.5%	97.8%	10.9%
MDR	GDS	85.3%	53.2%	55.2%
	SDS	78.7%	47.3%	63.8%

TABLE 9
Comparison Results between ViDIE and DEPTA

	dataset	*precision*	*recall*	*revision*
ViDIE	GDS	96.3%	97.2%	14.1%
	SDS	95.6%	98.4%	11.6%
DEPTA	GDS	75.3%	71.6%	32.8%
	SDS	66.1%	54.1%	37.6%

similarity threshold, which is set at 0.8. In this experiment, the input to ViDRE and MDR contains the deep Web pages and the output contains data records extracted. For ViDRE, one sample result page containing the most data records is used to generate the data record wrapper for each Web database. Table 8 shows the experimental results on both GDS and SDS. Based on our experiment, it takes approximately 1 second to generate the data record wrapper for each page and less than half second to use the wrapper for data record extraction.

From Table 8, we can make the following three observations. First, ViDRE performs significantly better than MDR on both GDS and SDS. Second, ViDRE is far better than MDR on revision. ViDRE needs only to revise slightly over 10 percent of the wrappers, while MDR has to revise almost five times more wrappers than ViDRE. Third, the precision and recall of ViDRE are steady on both SDS and GDS, but for MDR, they drop noticeably for SDS. Our analysis indicates that: for *precision* of ViDRE, most errors are caused by failing to exclude noise blocks that are very similar to the correct ones in appearance; for *recall* of ViDRE, most errors are caused by mistaking some top or bottom data records as the noise blocks; for MDR, its performance is inversely proportional to the complexity of the data records, especially data records with many optional data items.

7.4 Experimental Results on ViDIE

In this part, we evaluate ViDIE and compare it with DEPTA. DEPTA can be considered as the follow-up work for MDR, and its authors also called it MDRII. Only correct data records from ViDRE are used to evaluate ViDIE and DEPTA. For ViDIE, two sample result pages are used to generate the data item wrapper for each Web database. Table 9 shows the experimental results of ViDIE and DEPTA on both GDS and SDS. Our experiments indicate that it takes between 0.5 and 1.5 seconds to generate the data item wrapper for each page and less than half second to use the wrapper for data item extraction.

From Table 9, we can see that the observations we made in comparing the results of ViDRE and MDR remain basically valid for comparing ViDIE and DEPTA. In addition, we also found that DEPTA often misaligns two data items of different semantics if they are close in the tag tree and have the same tag path, and this leads to the misalignment of all the data items in the same data record that follow the misaligned data items. In contrast, ViDIE can easily distinguish them due to their different fonts or positions.

We also tried to use one sample page and three sample pages to generate the data item wrapper for each Web database. When one page is used, the performance is much lower; for example, for SDS, the precision, recall, and revision are 91.7, 95, and 32.3 percent, respectively. This is caused by the absence of some optional data items from all the data records in the sample page used. When more sample pages are used, the likelihood that this will happen is significantly reduced. When three pages are used, the results are essentially the same as shown in Table 9, where two sample pages are used. This suggests that using two sample pages to generate the data item wrapper for each Web database is sufficient.

We also conducted experiments based on the data sets used in [30] and provided by [13], and the results are similar to those shown in Tables 8 and 9. These results are not shown in this paper due to space consideration.

8 Conclusions and Future Works

With the flourish of the deep Web, users have a great opportunity to benefit from such abundant information in it. In general, the desired information is embedded in the deep Web pages in the form of data records returned by Web databases when they respond to users' queries. Therefore, it is an important task to extract the structured data from the deep Web pages for later processing. In this paper, we focused on the structured Web data extraction problem, including data record extraction and data item extraction. First, we surveyed previous works on Web data extraction and investigated their inherent limitations. Meanwhile, we found that the visual information of Web pages can help us implement Web data extraction. Based on our observations of a large number of deep Web pages, we identified a set of interesting common visual features that are useful for deep Web data extraction. Based on these visual features, we proposed a novel vision-based approach to extract structured data from deep Web pages, which can avoid the limitations of previous works. The main trait of this vision-based approach is that it primarily utilizes the visual features of deep Web pages.

Our approach consists of four primary steps: Visual Block tree building, data record extraction, data item extraction, and visual wrapper generation. Visual Block tree building is to build the Visual Block tree for a given sample deep page using the VIPS algorithm. With the Visual Block tree, data record extraction and data item extraction are carried out based on our proposed visual features. Visual wrapper generation is to generate the

wrappers that can improve the efficiency of both data record extraction and data item extraction. Highly accurate experimental results provide strong evidence that rich visual features on deep Web pages can be used as the basis to design highly effective data extraction algorithms.

However, there are still some remaining issues and we plan to address them in the future. First, ViDE can only process deep Web pages containing one data region, while there is significant number of multidata-region deep Web pages. Though Zhao et al. [31] have attempted to address this problem, their solution is HTML-dependent and its performance has a large room for improvement. We intend to propose a vision-based approach to tackle this problem. Second, the efficiency of ViDE can be improved. In the current ViDE, the visual information of Web pages is obtained by calling the programming APIs of IE, which is a time-consuming process. To address this problem, we intend to develop a set of new APIs to obtain the visual information directly from the Web pages.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation of China under grant 60833005, the National High Technology Research and Development Program of China (863 Program) under grant 2007AA01Z155 and 2009AA011904, the Doctoral Fund of Ministry of Education of China under grant 200800020002, the China Postdoctoral Science Foundation funded project under grant 20080440256 and 200902014, and US National Science Foundation (NSF) grants IIS-0414981 and CNS-0454298. The authors would also like to express their gratitude to the anonymous reviewers for providing some very helpful suggestions.

REFERENCES

[1] G.O. Arocena and A.O. Mendelzon, "WebOQL: Restructuring Documents, Databases, and Webs," *Proc. Int'l Conf. Data Eng. (ICDE)*, pp. 24-33, 1998.
[2] D. Buttler, L. Liu, and C. Pu, "A Fully Automated Object Extraction System for the World Wide Web," *Proc. Int'l Conf. Distributed Computing Systems (ICDCS)*, pp. 361-370, 2001.
[3] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma, "Block-Level Link Analysis," *Proc. SIGIR*, pp. 440-447, 2004.
[4] D. Cai, S. Yu, J. Wen, and W. Ma, "Extracting Content Structure for Web Pages Based on Visual Representation," *Proc. Asia Pacific Web Conf. (APWeb)*, pp. 406-417, 2003.
[5] C.-H. Chang, M. Kayed, M.R. Girgis, and K.F. Shaalan, "A Survey of Web Information Extraction Systems," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 10, pp. 1411-1428, Oct. 2006.
[6] C.-H. Chang, C.-N. Hsu, and S.-C. Lui, "Automatic Information Extraction from Semi-Structured Web Pages by Pattern Discovery," *Decision Support Systems*, vol. 35, no. 1, pp. 129-147, 2003.
[7] V. Crescenzi and G. Mecca, "Grammars Have Exceptions," *Information Systems*, vol. 23, no. 8, pp. 539-565, 1998.
[8] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, pp. 109-118, 2001.
[9] D.W. Embley, Y.S. Jiang, and Y.-K. Ng, "Record-Boundary Discovery in Web Documents," *Proc. ACM SIGMOD*, pp. 467-478, 1999.
[10] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krpl, and B. Pollak, "Towards Domain Independent Information Extraction from Web Tables," *Proc. Int'l World Wide Web Conf. (WWW)*, pp. 71-80, 2007.
[11] J. Hammer, J. McHugh, and H. Garcia-Molina, "Semistructured Data: The TSIMMIS Experience," *Proc. East-European Workshop Advances in Databases and Information Systems (ADBIS)*, pp. 1-8, 1997.
[12] C.-N. Hsu and M.-T. Dung, "Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web," *Information Systems*, vol. 23, no. 8, pp. 521-538, 1998.
[13] http://daisen.cc.kyushu-u.ac.jp/TBDW/, 2009.
[14] http://www.w3.org/html/wg/html5/, 2009.
[15] N. Kushmerick, "Wrapper Induction: Efficiency and Expressiveness," *Artificial Intelligence*, vol. 118, nos. 1/2, pp. 15-68, 2000.
[16] A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira, "A Brief Survey of Web Data Extraction Tools," *SIGMOD Record*, vol. 31, no. 2, pp. 84-93, 2002.
[17] B. Liu, R.L. Grossman, and Y. Zhai, "Mining Data Records in Web Pages," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 601-606, 2003.
[18] W. Liu, X. Meng, and W. Meng, "Vision-Based Web Data Records Extraction," *Proc. Int'l Workshop Web and Databases (WebDB '06)*, pp. 20-25, June 2006.
[19] L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," *Proc. Int'l Conf. Data Eng. (ICDE)*, pp. 611-621, 2000.
[20] Y. Lu, H. He, H. Zhao, W. Meng, and C.T. Yu, "Annotating Structured Data of the Deep Web," *Proc. Int'l Conf. Data Eng. (ICDE)*, pp. 376-385, 2007.
[21] J. Madhavan, S.R. Jeffery, S. Cohen, X.L. Dong, D. Ko, C. Yu, and A. Halevy, "Web-Scale Data Integration: You Can Only Afford to Pay As You Go," *Proc. Conf. Innovative Data Systems Research (CIDR)*, pp. 342-350, 2007.
[22] I. Muslea, S. Minton, and C.A. Knoblock, "Hierarchical Wrapper Induction for Semi-Structured Information Sources," *Autonomous Agents and Multi-Agent Systems*, vol. 4, nos. 1/2, pp. 93-114, 2001.
[23] Z. Nie, J.-R. Wen, and W.-Y. Ma, "Object-Level Vertical Search," *Proc. Conf. Innovative Data Systems Research (CIDR)*, pp. 235-246, 2007.
[24] A. Sahuguet and F. Azavant, "Building Intelligent Web Applications Using Lightweight Wrappers," *Data and Knowledge Eng.*, vol. 36, no. 3, pp. 283-316, 2001.
[25] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," *Proc. Conf. Information and Knowledge Management (CIKM)*, pp. 381-388, 2005.
[26] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma, "Learning Block Importance Models for Web Pages," *Proc. Int'l World Wide Web Conf. (WWW)*, pp. 203-211, 2004.
[27] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," *Proc. Int'l World Wide Web Conf. (WWW)*, pp. 187-196, 2003.
[28] X. Xie, G. Miao, R. Song, J.-R. Wen, and W.-Y. Ma, "Efficient Browsing of Web Search Results on Mobile Devices Based on Block Importance Model," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom)*, pp. 17-26, 2005.
[29] Y. Zhai and B. Liu, "Web Data Extraction Based on Partial Tree Alignment," *Proc. Int'l World Wide Web Conf. (WWW)*, pp. 76-85, 2005.
[30] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C.T. Yu, "Fully Automatic Wrapper Generation for Search Engines," *Proc. Int'l World Wide Web Conf. (WWW)*, pp. 66-75, 2005.
[31] H. Zhao, W. Meng, and C.T. Yu, "Automatic Extraction of Dynamic Record Sections from Search Engine Result Pages," *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, pp. 989-1000, 2006.
[32] J. Zhu, Z. Nie, J. Wen, B. Zhang, and W. Ma, "Simultaneous Record Detection and Attribute Labeling in Web Data Extraction," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 494-503, 2006.

Wei Liu received the BS and MS degrees in computer science from Shandong University in 1998 and 2004, respectively, and the PhD degree in computer science from Renmin University of China in 2008. Since 2008, he has been a postdoctoral fellow in computer science in the Institute of Computer Science and Technology of Peking University. His research interests include Web data extraction and deep Web data integration.

Xiaofeng Meng received the BS degree from Hebei University, the MS degree from Remin University of China, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, all in computer science. He is currently a professor in the School of Information, Renmin University of China. His research interests include Web data integration, native XML databases, mobile data management, and flash-based databases. He is the secretary general of Database Society of the China Computer Federation (CCF DBS). He has published more than 100 technical papers. He is a member of the IEEE.

Weiyi Meng received the BS degree in mathematics from Sichuan University, China, in 1982, and the MS and PhD degrees in computer science from the University of Illinois at Chicago, in 1988 and 1992, respectively. He is currently a professor in the Department of Computer Science at the State University of New York at Binghamton. His research interests include Web-based information retrieval, metasearch engines, and Web database integration. He is a coauthor of a database book *Principles of Database Query Processing for Advanced Applications*. He has published more than 100 technical papers. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

2010 Sixth International Conference on Semantics, Knowledge and Grids

A Holistic Solution for Duplicate Entity Identification in Deep Web Data Integration

Wei Liu [1,2], Xiaofeng Meng [3]

[1] *Institute of Computer Science and Technology, Peking University, Beijing 100871, China*

[2] *Institute of Scientific and Technical Information of China*

[1] gue1976@gmail.com

[3] *School of Information, Renmin University of China, Beijing 100872, China*

[3] xfmeng@ruc.edu.cn

Abstract—**The proliferation of deep Web offers users a great opportunity to search high-quality information from Web. As a necessary step in deep Web data integration, the goal of duplicate entity identification is to discover the duplicate records from the integrated Web databases for further applications(e.g. price- comparison services). However, most of existing works address this issue only between two data sources, which are not practical to deep Web data integration systems. That is, one duplicate entity matcher trained over two specific Web databases cannot be applied to other Web databases. In addition, the cost of preparing the training set for *n* Web databases is C_n^2 times higher than that for two Web databases. In this paper, we propose a holistic solution to address the new challenges posed by deep Web, whose goal is to build one duplicate entity matcher over multiple Web databases. The extensive experiments on two domains show that the proposed solution is highly effective for deep Web data integration.**

I. INTRODUCTION

With the proliferation of deep Web, a flood of high-quality information(usually in form of structured records) can be accessed through online Web databases. The recent statistics[1] reveal that there are more than 450,000 Web databases in the current Web. These Web databases offer web services, RSS feeds, even provide APIs to allow data to be exchanged between them. Deep Web data integration aims to combine Web information to provide users a unified view.

At present, many issues in the field of deep Web data integration, such as interface integration[2][3] and Web data extraction[4,5], have been widely studied. However, as a necessary step, identifying the duplicate entities(records) from multiple Web databases has not received due attention yet. In one domain(book, music, computer, etc.), there are often a large proportion of duplicate entities across Web databases, so it is necessary to identify them for further applications, such as de-duplication or price-comparison services. Due to the heterogeneity of Web databases, the duplicate entities usually exist in various inconsistent presentations. In this paper, we study the duplicate entity identification problem in the context of deep Web data integration.

Until now, there are already lots of research works to address this issue, but most of them only focus on the two-data-source situation. However, when facing the lots of Web databases, they have to build C_n^2 matchers, where *n* is the number of Web databases. This makes the unaffordable costs for both preparing training set and building the duplicate identification matcher. In most existing deep Web data integration systems, the duplicate entity matchers are manually built under small scale and static integration scenarios. In contrast, in large scale deep Web data integration scenarios, this process needs to be as automatic as possible and scalable to large quantities of Web databases. We will review previous works in Section VI.

In this paper, we propose a domain-level solution to address the challenges posed by deep Web, which means, the trained matcher can identify the duplicate entities over multiple one-domain Web databases. The intuition behind our solution is that, given a domain, the number of attributes is convergent, and further, each attribute plays a definite role on the duplicate entity identification problem. In another word, the importance (or weight) of an attribute is actually domain-dependent. For example, in Book domain, "title" is always more important than "publisher" to determine whether two book records refer to the same book. our solution consists of the following three main steps.

Semi-automatic training set generation: in previous works, the training set(matched record pairs) was prepared through manually, which is unpractical when facing lots of *WDB*s(Web databases for short). Thus, a semi-automatic method is proposed to generate the training set automatically, which can significantly reduce the labelling cost.

Attribute weight training: in order to weigh the importance of the similarity of each attribute reasonably, we propose an iterative training approach to learn the attribute weights of. The basic idea is that, under the ideal weights, the similarity of any two matched records must be larger than that of any two unmatched records.

In summary, the contributions of this paper are:

- As our problem, it is first time to probe the duplicate entity identification problem in the context of deep Web data integration, where lots of Web databases pose new challenges to this issue.
- As our insight on the observation, we discover the attributes in a same domain play definite roles on the problem of duplicate entity identification, which makes it possible to build one matcher over multiple Web databases in one domain.
- As our solution, we propose a holistic solution on

978-0-7695-4189-1/10 $26.00 © 2010 IEEE
DOI 10.1109/SKG.2010.38

IEEE computer society

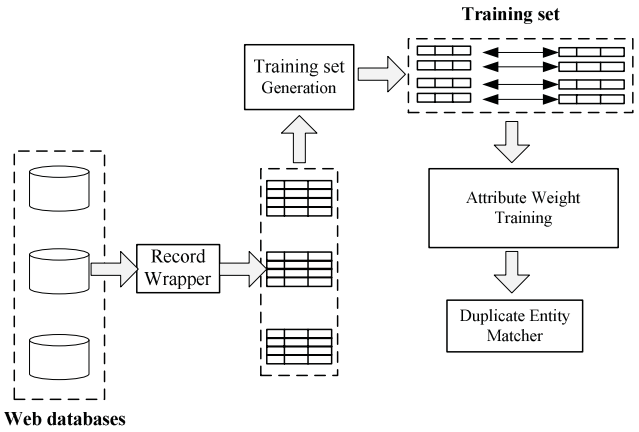


Fig. 1. Solution Overview

duplicate entity identification under the context of deep Web data integration. Our experiments show the promise of this solution.

The rest of paper is organized as follows. In section 2, we present the overview of our solution. Section 3 proposes a semi-automatic method to produce the training set. Section 4 proposes a novel approach of attribute weight learning. An experimental evaluation for our approach is shown in Section 5. In Section 6, we talk about the related works. Section 7 discusses several further opportunities and then concludes the paper.

II. SOLUTION OVERVIEW

In this paper we proposed a practical domain-level solution to address the problem of duplicate entity identification under the context of deep Web data integration. Figure 1 shows the overview of our solution. The input is the records in different *WDB*s. The output is a set of record pairs, where each pair denotes two duplicate records. [14] for Web data item extraction have been proposed and confirmed to achieve satisfactory accuracy.

There are three primary components in our solution. Their functions are introduced briefly as follows.

- **Record Wrapper:** In general, the records in Web databases are embedded in web pages when users submit queries. The function of this component is automatically extracting the structured records from web pages at attribute level.
- **Semi-automatic training set generation:** This component aims at semi-automatically obtaining enough duplicate records as the training set from the records wrappered from *WDB*s. Each training sample refers to two duplicate records.
- **Attribute weight training:** Each attribute is assign an appropriate weight by the component by employing our proposed iterative training-based approach, and two thresholds T_1 and T_2 ($T_1 > T_2$) are also learned.
- **Duplicate entity matcher**: Given two inputted records, their similarity can be computed with the weights of the shared attributes, and further, the two records can be determined whether being duplicates using the thresholds.

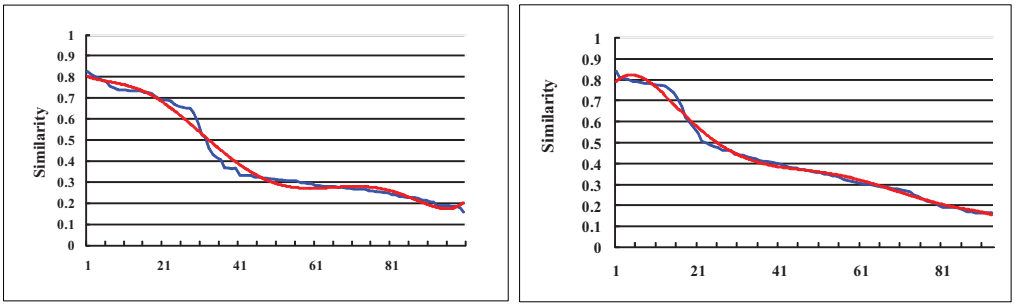
Wrapper belongs to the research field of web data extraction has been widely studied, and many automatic approaches have been proposed to address this issue. The idea of duplicate entity matcher is rather simple and direct. So no more discussions are for them in this paper due to the limitation of paper length. The rest of this paper will focus on the underlying techniques of the two components training set generation and attribute weight training.

III. SEMI-AUTOMATIC TRAINING SET GENERATION

In previous approaches, the training set was always prepared manually in advance. That is, domain experts label some record pairs to be duplicates or not as the training set. Unfortunately, lots of Web databases makes manually labelling training set time-consuming and error-prone. Obviously, the labelling cost for n Web databases is C_n^2 times of that for 2 Web databases. In this section, a semi-automatic method is introduced to generate the training set for n Web databases.

Instinctively, if two records from different *WDB*s are determined to be matched (i.e. they are duplicates), they often share more texts than the unmatched ones. So a naive approach is to regard each record as a short text document, and determine whether two records are duplicates by comparing text similarity, such as *tf-idf* function. But obviously the accuracy is not satisfying and not stable. We have evaluated this naive approach on two domains(book, computer), and the accuracies are only about 83% and 47% respectively. We check the results and divide all matched record pairs into correct ones and wrong record ones. The correct record pairs refer to the duplicates in fact, while the wrong record pairs are not. If ranking all matched record pairs by their *tf-idf* similarity in descending order, we find most correct record pairs congregates in the head, while most wrong ones are in the tail. This phenomenon motivates us to obtain training set(the right ones) from the head.

Figure 2 shows the curves of the record pair sequences for two domains. Through farther analysing, we find that: if the total matched record pairs are enough(say, more than 100), two distinct inflexions divide the whole curve into three segments(head segment, body segment, and tail segment).



(a) Book domain (b) Computer domain

Fig. 2 The relationship of the ith record pair and its similarity

Most correct ones locate at the head segment, and most wrong ones locate at the tail segment. The body segment is mainly the mixture of correct ones and wrong ones. So it is feasible to regard the record pairs in the head segment as the training set. And the problem is how to find the head segment in the curve.

In order to detect the first arc accurately, we resort to a mathematic mean which consists of two steps: curve fitting and curvature computation. In the first step, given a sequence of similarity values, point them in a two-dimensional reference frame, where *y* axes denotes the similarity value and *x* axes denotes the similarity ranking result. The least squares fitting method is applied for curve fitting (the red curve in Figure 2). The least squares fitting method is a very popular mathematic method of fitting data, and so its technique detail is not discussed here anymore. In the second step, the curvature for each similarity value in the curve is computed, and the similarity with the maximum downward curvature is located. Then this similarity value in the curve is we want to locate. One training set is obtained for every two *WDB*s. Suppose *n WDB*s, totally C_n^2 training sets have to be generated. The final training set is the sum of these training sets.

However, the training set is often not perfect, which means several wrong matched data record pairs may mix in. If their experiments are based on the noisy training set, the accuracy will be far away from what they reported in their experiments. So a quick one-pass checking for the training set is needed to get rid of the wrong matched data record pairs. Though this is manual, the cost is obviously far less than the traditional way. Intuitively, to guarantee the quality of the training set, the size of the unlabeled training set should be large enough. The related experiments will be given in Section 6 to guide us to leverage this problem.

IV. ATTRIBUTE WEIGHT TRAINING

In this part, we study the problem of training the appropriate domain-level attribute weights using the training set obtained from Web databases. As a result, the trained attribute weights can be applied for any two Web database in this domain.

A. Preliminaries

An iterative training mechanism is proposed to this task, and we call it IBITA (Inequalities Based Iterative Training Approach) in this paper. Figure 3 shows its architecture. IBITA starts with two *R* sets from WDB_A and WDB_B. Without loss of generality, we suppose that there are *m* attribute among *n WDB*s. For each record pair $<R^i, R^j>$, we define the record similarity as follows.

Definition 5.1. (**Record Similarity**) The similarity of R^i and R^j is the weighted sum of the similarities of the shared attributes. Correspondingly, weight w_k ($1<= k<=m$) is assigned to the corresponding attribute am^k to show its contribution to the similarity of R^i and R^j. Formally, record similarity is denoted below:

$$S(R^i, R^j) = \sum_{k=1}^{m} w_k \times S(am^k) \quad (1)$$

Using weight vector $<w_1, w_2 \ldots\ldots w_m>$(*WV* for short), we can measure the similarity of any record pair $< R^i, R^j >$ as a real number larger than 0. The ideal weights vector is hoped to make all the matched record pairs and non-matched record pairs take on a distinct bipolar distribution when projecting their similarities on the axis as shown in Figure 4. The bipolar distribution requires all those matched record pairs (denoted as circles) to locate at the starboard of the axis, while all those non-matched record pairs (denoted as rectangles) to locate at the larboard of the axis. We would like the optimal weight vector ($WV_{optimal}$) which makes the bipolar distribution on the axis most distinct, that is, bring the largest distance of matched and non-matched record pairs marked on Figure 4. Meanwhile, two thresholds are also needed to determine each record pair to be "matched", "non-matched", or "possibly matched".

Suppose the training set contains *n* matched record pairs, where each pair describes one same entity. We use $< R^i, R^i >$ to denote the matched record pair, and use $< R^i, R^j >(i \neq j)$ to denote the non-matched pair.

B. Inequalities-based Metrics

By observing the bipolar distribution shown in Figure 4, we find that *WV* needs to be adjusted to meet the following condition: The similarity of a matched pair is greater than the similarity of a non-matched pair. Formally, the similarity of *n* uniquely matched pair $< R^i, R^i >$ should be greater than any of the $n*(n-1)$ non-matched pairs $< R^j, R^k >$ ($j \neq k$). Therefore, a group of $n * n * (n - 1)$ inequalities can be correspondingly obtained as follows:

$$\{S(R^i, R^i) \geq S(R^j, R^k)\} \quad 1 \leq i, j, k \leq n, j \neq k \qquad \textbf{(2)}$$

Totally $n*(n-1)$ inequalities are generated. We try to find $WV_{optimal}$ from the solution space of Inequalities 2. Intuitively,

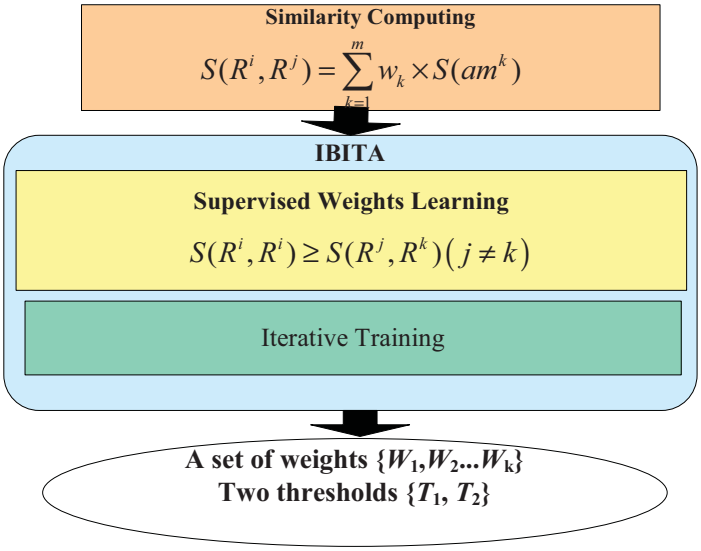


Fig. 3. General IBITA architecture

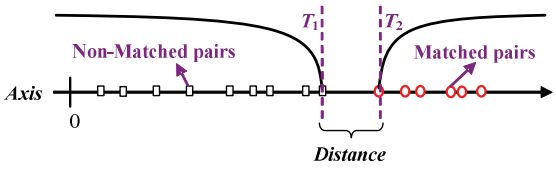


Fig. 4. The ideal bipolar distribution

we have to solve these $n * (n - 1)$ inequalities, a right (not optimal) *WV* can be got. However, the exponential growth of the number of inequalities is too costly for real applications. So we use the following subset of these inequalities instead of Inequa. (2):

$$\{S(R^i, R^i) \geq S(R^i, R^j)\} \quad 1 \leq i, j \leq n, i \neq j \qquad \textbf{(3)}$$

For any *WV* in the solution space of Inequa. (3), there will be two possibilities: the *WV* satisfies Inequa. (2), or not satisfies Inequa. (2). In another word, not all the *WV*s of Inequa. (3) can make the *n* matched record pairs and $n * (n - 1)$ non-matched record pairs a bipolar distribution as we wanted (see Figure 5(a)). Some *WV*s may lead to the cross-region situation shown in Figure 5(b), where it is still guaranteed on each axis, the matched record pair is closer to the starboard than all the non-matched record pairs. The cross-region situation means not all the similarities of *n* matched record pairs are larger than the similarities of all $n*(n–1)$ non-matched record pairs. This cross-region situation is thus caused where the *n* matched record pairs in training data set cannot be divided into matched or non-matched. As we can see from Figure 5(b), the similarity of the non-matched record pair $< R^2 , R^y > (y >= 2)$ is larger than the similarity of the matched record pair $< R^1 , R^1 >$. The confusion in this situation can be described as that: if the similarity of the new record pair falls into the cross-region formed by T_1 and T_2, the system will not be able to judge whether this two records represent the same entity due to the ambiguity they have. So what we need to do next is to try to obtain a *WV* in the solution space of Inequa. (2) using Inequa. (3).

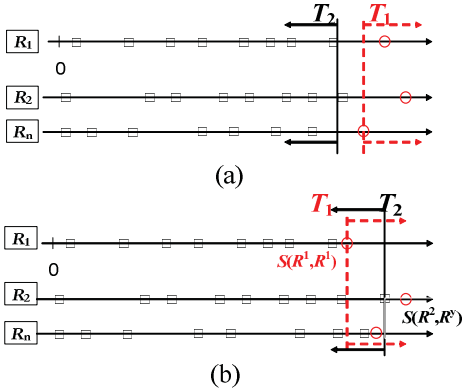


Fig. 5. Ideal situation and Cross-region situation

C. Iterative Training

Given a *WV* in the solution space of Inequa. (3), the similarity of $< R^i , R^j >$ in the training set can be derived as the weighted sum of the similarities of all attributes. In the training set containing *n* matched record pairs there are $n * n$ record similarities being computed, each of which corresponds to one random combination of R^i $(1 \leq i \leq n)$ and R^j $(1 \leq j \leq n)$. We project these n^2 record similarities to *n* axes and try to iteratively analyse different similarity distributions on the axes caused by different *WV*s in order to find $WV_{optimal}$.

For each $R^i (1 \leq i \leq n)$ we build an axis, and *n* similarities are projected on the axis as shown in Figure 5. The similarities of R^i are located in the *i*th axis. The circle denotes matched record pair $<R^i, R^i>$ which are closest to the starboard of the

axes, while the small rectangles denote non-matched record pairs $<R^i, R^j>$ $(i \neq j)$.

Given a *WV*, the minimum similarity of all *n* matched pairs is regarded as a threshold T_1 (dashed line in Figure 5) and the maximum similarity of all $n * (n - 1)$ non-matched pairs is regarded as a threshold T_2 (real line in Figure 5). Formally, we denote them as the following form:

$$\begin{cases} T_1 = min\{S(R^i, R^i)_{WV}\} \\ T_2 = max\{S(R^i, R^j)_{WV}\} \end{cases} \quad (i \neq j)$$

where $S(R^i, R^j)_{WV}$ is the similarity of R^i and R^j being computed with current *WV*.

If $T_2 < T_1$, we can guarantee the similarity of $< R^i, R^i >$ is larger than the similarity of $< R^i, R^j >$ $(i \neq j)$. So the ideal situation is $T_2 < T_1$, and cross-region situation is $T_1 < T_2$.

There are two main steps in the implement of this component which tries to obtain $WV_{optimal}$ staring at an arbitrary *WV* in the solution space of Inequa. (3). The first step is to obtain a *WV* satisfying Inequa. (2) from the *WV* of Inequa. (3), and the second step is to obtain $WV_{optimal}$ from a *WV* of Inequa. (2).

Step 1 *WV* of Inequalities 3→ *WV* of Inequalities 2

At the beginning, a *WV* is got by solving Inequa. (3), and further T_1 and T_2 are got. If $T_2 < T_1$, this means this *WV* satisfies Inequa. (2), and the next step is activated. Otherwise, the *WV* caused the cross-region situation, just like Fig. 5 (b). The goal of this step is to obtain a *WV* of Inequa. (2) using the *WV* of Inequa. (3). Next, for $T_1 < T_2$, it is represented in the following form:

$$min\{S(R^i, R^i)_{WV}\} < max\{S(R^i, R^j)_{WV}\} (i \neq j) \quad (4)$$

Then Inequa. (5) is formed by appending Inequa. (4) to Inequa. (3), and *WV'* is obtained by solving Inequa. (5). The left of this step is repeating the above process until the *WV* satisfies Inequa. (2).

The main idea of this step is to iteratively append the inequalities which do not satisfy Inequa. (2) to Inequa. (3) until a *WV* satisfying Inequa. (2) is got. In another word, the solution space continues shrinking during the process and a *WV* in the solution space of Inequa. (3) has more probability to be in the solution space of Inequa. (2). Actually, there is more than one inequality which does not satisfy Inequa. (2), but only one inequality (Inequa. (4)) is appended every iteration, due to the consideration of efficiency improvement. In practical, the iteration process is less than 4 times averagely.

Step 2 *WV* of Inequalities 2→ $WV_{optimal}$

This process starts at a *WV* of Inequa. (2). The current *WV* can guarantee the similarity of any matched record pair is larger than that of any non-matched record pair in the training set. In order to reach high accuracy, we need get $WV_{optimal}$ which can make the matched record pairs and non-matched record pairs the most distinct bipolar distribution. In another word, $WV_{optimal}$ can make the distance of T_1 and T_2 (i.e. $T_1 - T_2$) reach the maximum.

In order to make the description concisely and without confusion, we use Inequa. (4) to denote all the inequalities appended to Inequa. (3). Suppose Inequa. (5) is Inequa. (3) and the inequalities appended to Inequa. (3) in the first step. So Inequa. (5) is denoted as the following:

$$\begin{cases} S(R^i, R^i) - S(R^i, R^j) \geq 0\} \quad (1 \leq i, j \leq n, j \neq i) \\ max\{S(R^i, R^j)_{WV}\} - min\{S(R^i, R^i)_{WV}\} > 0 (j \neq i) \end{cases} \quad (5)$$

Initially, the zeros in the right side of inequalities is replaced by T_1-T_2, and the new inequalities (e.g. Inequa. (6)) are denoted as the following:

$$\begin{cases} S(R^i, R^i) - S(R^i, R^j) \geq 0 \quad (1 \leq i, j \leq n, i \neq j) \\ \max\{S(R^i, R^j)_{WV}\} - \min\{S(R^i, R^i)_{WV}\} > T_1 - T_2 \ (i \neq j) \end{cases} \quad (6)$$

WV' is got by solving Inequa. (6), and further T'_1 and T'_2 are got. Then $T'_1 - T'_2$ replaces $T_1 - T_2$ in Inequa. (6), and the above process is repeated until $(T'_1 - T'_2) - (T_1 - T_2) < \sigma$, σ is set in advance, and the smaller σ is, the current *WV* is closer to $WV_{optimal}$. In practice, σ is set to be 0.12.

Till now, for any number of *WDB*s in one domain, IBITA can ultimately bring to us an optimum group of quantified weights $WV_{optimal}$ and two stabilized thresholds T_1 and T_2. Then it is easy to compute the similarity for any two records (R^i, R^j) from different web databases using $WV_{optimal}$. Via comparing the similarity value with T_1 and T_2, we can easily determine they are matched or not. If the similarity of the record pair falls into the possibly matched region(i.e. $T_2 \leq S(R^i, R^j) \leq T_1$), it needs to be manually checked.

V. EXPERIMENTS

A prototype system, DWDEI(Deep-web Duplicate Entity Identifier), has been implemented based on our solution. We evaluate this system over the real Web databases on two popular domains. The test bed and the evaluation measures are introduced first. Then, a series of experiments are conducted for evaluation.

A. Data Set

TABLE 1: WEB DATABASES IN THE EXPERIMENTS

ID	Web database	Description
1	*Amazon*	www.amazon.com/□textbooks/□
2	*Bookpool*	www.bookpool.com/□
3	*Blackwell*	www3.interscience.wiley.com/browse/BOOK□
4	*ClassBook*	www.classbook.com/□
5	*Bookbyte*	www.bookbyte.com/□

(a) Book Domain

ID	Web database	Description
1	*Superwarehouse*	http://www.superwarehouse.com/□
2	*Amazon*	http://www.amazon.com/Computers□
3	*CNET*	http://reviews.cnet.com/desktop-computers/□
4	*Computers4sure*	www.computers4sure.com/□
5	*Bookbyte*	www.pcconnection.com/□

(b) Computer Domain

The test bed for evaluation is the Web databases on book and computer domains. For each domain, we select 5 popular web sites as the Web databases. Table 1 lists these Web databases. The reason that we select these *Web databases as the test bed is there are enough*

duplicate records among them. Therefore, there are totally C_5^2 Web database pairs in each domain. And we use 5 of them to produce the training set to learn the attribute weights and two thresholds in this domain. For each Web database pair, we submit 6 queries. For each query, we would select top 100 records from the query results. Both the training set and testing set are coming from the returned query results.

The characteristics of our data set can be concluded as follows. (1) For each submit query, the returned query results from 2 paired *WDB*s shared a large proportion of overlapping entities. (2) The scale of our data set is quite large that thetotal number of ∇⌉⌋≀∇⌈ ⊣⟩∇∫ has achieved more than 800 for each pair of *WDB*s. (3) The submitted queries are completely different, which guarantees that there is almost no overlap between the query results. All those features of our data set ensure the objectivity of our experimental results.

B. Evaluation Measures

Four criteria are defined to evaluate the effectiveness of our solution, which are listed below.

$$\text{Precision } M = \frac{|PredictedMP \cap ActualMP|}{|PredictedMP|}$$

$$\text{Precision } N = \frac{|PredictedNP \cap ActualNP|}{|PredictedNP|}$$

$$\text{Uncertainty} = \frac{|UncertainP|}{|PredictedMP + PredictedNP + UncertainP|}$$

$$\text{Precision}T = \frac{|PredictedMP \cap ActualMP| + |PredictedNP \cap ActualNP|}{|PredictedMP + PredictedNP + UncertainP|}$$

where ActualMP is the set of real matched record pairs in the testing set and PredicatedMP is the set of matched record pairs identified by DWDEI. Similarly, ActualNP is the set of real non-matched record pairs in the testing set and PredicatedNP is the set of non-matched record pairs identified by DWDEI. In addition, UncertainP denotes the set of record pairs that cannot be determined by DWDEI.. Those uncertain pairs need to be further manually checked.

C. Evaluation of semi-automatic training set generation

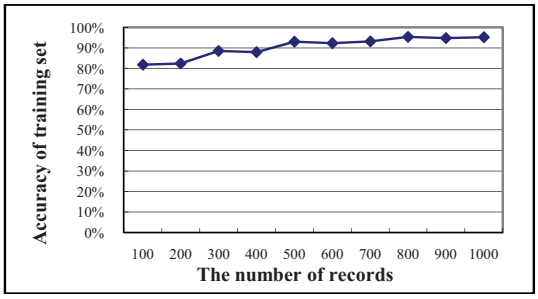


Fig. 6. The Experimental results of semi-automatic training set generation

In this part, we conduct the experiment to evaluate the effectiveness of the component semi-automatic training set generation. It is obvious that the performance of our solution depends greatly on the quality of the training set. In our solution, we wrapper the records from the result pages through submitting some popular queries. X axis in Figure 6 refers to the number of records wrappered from each result page.

As it can be seen from Figure 6, the accuracy tend to be stable as the number as the number of records increases. The accuracy is convergent at about 95% when the number of record pairs is larger than 800. Hence, we recommend more than 800 records are wrappered from each Web database to generate the training set in practice. Though manual checking is needed to pick out the 5% errors, the cost is far less than the traditional way because identifying two records are duplicates or not is much easier than finding the duplicate for one record from a large number of ones.

D. Overall Performance

TABLE II. WEIGHTS FOR BOOK DOMAIN

	Title	Author	Publisher	Date	Price	Edition
Weight	0.34	022	0.13	0.12	0.09	0.06

TABLE III. WEIGHTS FOR COMPUTER DOMAIN

	Model	CPU	Monitor	RAM	HD	CD
Weight	0.28	0.09	0.07	0.06	0.04	0.02

TABLE IV. EXPERIMENTAL RESULTS FOR BOOK DOMAIN

	PrecisionM	PrecisionN	Uncertainty	PrecisionT
P12	0.989	1	0.020	0.994
P13	0.985	0.956	0.036	0.937
P14	0.965	0.983	0.009	0.970
P15	0.916	1	0.024	0.955
P23	0.96	0.986	0.024	0.970
P24	0.928	0.905	0.044	0.941
P25	0.977	0.991	0.009	0.985
P34	0.896	0.946	0.014	0.931
P35	0.979	0.948	0.022	0.966
P45	0.919	0.974	0.022	0.944
AVG	0.951	0.968	0.022	0.959

TABLE V. EXPERIMENTAL RESULTS FOR COMPUTER DOMAIN

	PrecisionM	PrecisionN	Uncertainty	PrecisionT
P12	0.858	0.997	0.026	0.914
P13	0.811	0.952	0.033	0.882
P14	0.954	0.895	0.012	0.923
P15	1	0.863	0.029	0.940
P23	0.876	0.813	0	0.848
P24	0.834	0.776	0.019	0.828
P25	0.819	0.848	0.017	0.836
P34	0.849	0.958	0.013	0.893
P35	0.943	0.919	0.007	0.931
P45	0.978	0.950	0.014	0.969
AVG	0.892	0.897	0.017	0.896

For each domain, totally 10(i.e. C_5^2) Web database pairs are produced. We use Pij to denote a specific Web database pair.

For example, P24 refers to the 2nd web database and the 4th web database. Web database pairs P12, P13, P14, P15 and P23 are used to learn the optimal attribute weights $WV_{optimal}$ and the thresholds T_1 and T_2. The test bed consists of two parts: (a) the record pairs from P12, P13, P14, P15 and P23; (b) the record pairs from P24, P25, P34, P35 and P45. Table 2 and Table 3 show the normalized attribute weights for Book domain and Computer domain respectively. Due to the space limitation, only top 6 frequent attributes are listed.

Table 4 and Table 5 show the accuracy of DWDEI for book domain and computer domain respectively. As it can be seen from Table 4 and Table 5, our experimental results reveal 3 features of DWDEI: (1) Our solution performs well on all the four measures. This shows that our solution is highly effective. (2) The measure UncertaintyP are extremely low (AVG 2.2% on book domain and AVG 1.7% on computer domain), which greatly reduces the manual intervention. (3) The small decrease in performance on several Web database pairs strongly indicates that our approach is very robust, considering the facts that record pairs are from completely new Web database pairs.

In addition, we also observe that the performance on book domain is a little better than that on computer domain. The main reason for this phenomenon is that, the value ranges of computer attributes are often small, so it is more difficult to differentiate the matched record pairs and non-matched record pairs.

E. *Performance comparison with previous related works using Cora dataset*

To compare with the works in this field, we also conduct the experiment on the popular Cora dataset. Cora dataset is the standard in the duplicate entity identification community and is frequently used as the test bed for evaluation. Cora contains 2191 5-field citations to 305 computer science papers. The goal of this experiment is two-fold. First, *Cora* is often as the test bed in the related works. The experiments can be used for the performance comparison between our approach and previous works which also carried out their experiments on it. Second, we believe DWDEI can also be applied to unacquainted Web databases.

TABLE VI. EXPERIMENTAL RESULTS ON CORA DATA SET

Precision	*Recall*	*F-measure*
97.6%	92.7%	95.1%

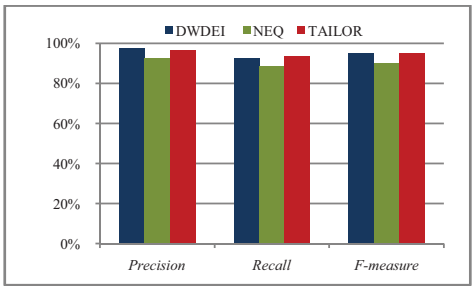


Fig. 7. Performance comparison among

Since only one threshold is learned in the previous works, we use the mean of as the threshold. We compare DWDEI with two recent related works [35] and [36]. From the experimental results shown in Table 6 and the performance comparison on F-measure shown in Figure 7, two conclusions can be made. First, the performance of DWDEI on *Cora* is very good on both the three traditional measures. Second, the performance on *Cora* is a little better than those reported by the related works. The experiments indicate that DWDEI based on our solution takes on the domain-level character. That is, DWDEI can still achieve a satisfactory performance among new Web databases(the training set is not generated from them) in this domain when enough important attributes are covered.

VI. RELATED WORK

The goal of duplicate entity identification is to identify the duplicated records in the same or different databases that refer to the same real world entity, even if the records are not identical. It is well known that the duplicate entity identification problem have been studied for decades. This first work can be seen in [10], which is proposed by Fellegi-Sunter in the late 1950s and 1960s. A recent survey[34] has been given to summary the research works in this field.

In this Section, we give a more detailed category for the works on duplicate entity identification according their techniques, and present primary representative works for each class.

A large number of works and solutions have been proposed to address this challenging problem. These works mainly focused on the entity identification problem from two aspects: attribute similarity comparison and duplicate records detection. Attribute similarity comparison produces a vector of similarity scores corresponding to attribute pair comparison result; with this similarity vector, each record pair is classified as a match or non-match using different classification techniques. Attribute similarity comparison often use some string distance metrics. Edit distance[22], as a generic metric, can be considered as the first metric for string comparison. The following proposed metrics, such as affine gap distance[23] and Jaro distance[24], etc, define different penalties for the position of gap or the string order, which can be applied to some special situations, such as person name and address. For example, affine gap distance can work well when matching strings that have been truncated or shortened, while Jaro distance allows for better local alignment of the strings. However, all of them cannot address the situation due to various representations which is very common across multiple Web databases.

For identifying record pairs as matching or non-matching, there are several class of solutions [20]: rule-based methods that use matching rules given by human experts; supervised learning techniques which use labelled examples to learn rules or train a probabilistic model such as Bayesian network, SVM, a decision tree and so on; unsupervised learning techniques that can label the matched records from the training data automatically; distance-based methods which avoid the

need for training data by defining a distance metric and an appropriate matching threshold. Actually, the matchers they generated can only work well over two Web databases.

Recently, the value of additional information for duplicated entity identification has been recognized by researchers, such as semantic relationships or mappings. The rich information present in the associations between references is exploited for reference reconciliation [16]. [19] described a source conscious compiler for entity resolution which exploits information about data sources and employs multiple matching solution to improve matching accuracy. Moma matching system uses a library of matching algorithm and the combination of their results to improve match quality [21]. But it is an overhead problem to build and maintains the library of matching algorithm and selects the suitable algorithm.

Overall, there are two significant differences between our work and the previous works. First, most of previous works only deal with entity identification problem in the specific data sources, so it is hard to produce one robust matcher to cover multiple Web databases. Second, most previous works prepare their training sets in the manual way, which is impractical when facing lots of Web databases. Instead, we propose an automatic approach to generate the flawed training set, and only a quick one-pass check is needed to pick up the errors. This can reduce the labelling cost significantly.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we study the problem of duplicate entity identification for deep Web data integration. We first give an observation to the attributes in one domain and hypothesize that their roles are definite or domain-dependent. Then, we propose a holistic approach to address this problem, which includes training set achieving, attribute mapping, and attribute weight assigning. In the experiments, we choose two representative domains (book and computer) to evaluate our approach, and the experimental results prove its accuracy is satisfying in practical. In the future, we will expand the scale of our from two aspects: (a) increase the number of Web databases in each domain; (b) extend our experiments to more important domains, such as automobile, movie and research papers, etc.

References

[1] Chang K. C., He B., Li C., Patel M., Zhang Z.. Structured Databases on the Web: Observations and Implications. SIGMOD Record 33(3): 61-70 (2004).

[2] Dragut E. C., Wu W., Sistla A. P.: Merging Source Query Interfaces on Web Databases: ICDE 2006: 46

[3] Wu W., Doan A., Yu C. T.: WebIQ: Learning from the Web to Match Deep-Web Query Interfaces. ICDE 2006: 44

[4] Zhao H., Meng W., Wu Z., V. Raghavan: Fully automatic wrapper generation for search engines. WWW 2005: 66-75s

[5] Liu B., Grossman R. L., Zhai Y.: Mining data records in Web pages. KDD 2003: 601-606 [6]

[6] Tejada S., Knoblock C. A., Minton S.: Learning domain-independent string transformation weights for high accuracy object identification. KDD 2002: 350-359

[7] Sarawagi S., Bhamidipaty A.: Interactive de-duplication using active learning. KDD 2002: 269-278

[8] Zhang J., Ling T. W., Bruckner R. M., Liu H.: PC-Filter: A Robust Filtering Technique for Du-plicate Record Detection in Large Databases. DEXA 2004: 486-496.

[9] Newcombe H. B., Kennedy J. M., Axford S.J.: Automatic linkage of vital records. Science, 130(3381):954-959, 1959.

[10] Fellegi I. P. and Sunter A. B.: A theory for record linkage. Journal of the American Statistical Association, 64(328):1183-1210, 1969.

[11] Cochinwala M., Kurien V., Lalk G.: Improving generalization with active learning. Information Sciences, 137(1-4):1-15, 2001.

[12] Bilenko M., Mooney R. J., Cohen W. W.: Adaptive name matching in information integration. IEEE Intelligent Systems, 18(5):16-23, 2003.

[13] He B., Chang K. C.-C.: Making holistic schema matching robust: an ensemble approach. KDD 2005: 429-438

[14] Newcombe H. B., Kennedy J. M., Axford S.J., and James A.P.: Automatic linkage of vital records. Science, 130(3381):954-959, October 1959.

[15] Jaro M. A.. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association, 84(406):414-420, June 1989.

[16] Dong X., Halevy A., and Madhavan J.. Reference reconciliation in complex information spaces. SIGMOD 2005, 85-96.

[17] Winkler W. E.. Improved decision rules in the felligi-sunter model of record linkage. Technical Report Statistical Research Report Series RR93/12, U.S. Bureau of the Census, Washington, D.C., 1993.

[18] Cochinwala M., Kurien V., Lalk G.: Improving generalization with active learning. Information Sciences, 137(1-4):1-15, September 2001.

[19] Shen W., DeRose P., Vu L., Doan A., Ramakrishnan R.. Source-aware Entity Matching: A Compositional Approach. ICDE 2007, 196-205.

[20] N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: Similarity measures and algorithms (tutorial). SIGMOD 2006, 802-803.

[21] A. Thor, E. Rahm. MOMA - A Mapping-based Object Matching System. CIDR 2007, 247-258.

[22] Levenshtein V.I.. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Doklady Akademii Nauk SSSR, vol. 163, no. 4, pp. 845-848, 1965, original in Russiantranslation in Soviet Physics Doklady, vol. 10, no. 8, pp. 707-710, 1966.

[23] Waterman M.S., Smith T.F., and Beyer W.A.. Some Biological Sequence Metrics. Advances in Math., vol. 20, no. 4, pp. 367-387, 1976.

[24] M.A. Jaro. Unimatch: A Record Linkage System: Users Manual. technical report, US Bureau of the Census, Washington, D.C., 1976.

[25] Sarawagi S., Bhamidipaty A.. Interactive deduplication using active learning. KDD 2002: 269-278

[26] Tejada S., Knoblock C. A., Minton S.. Learning domain-independent string transformation weights for high accuracy object identification. KDD 2002.

[27] Ananthakrishna R., Chaudhuri S., Ganti V.. Eliminating fuzzy duplicates in data warehouses. VLDB 2002

[28] Guha S., N. Koudas, A. Marathe. Merging the results of approximate match operations. VLDB 2004: 636-647.

[29] Chaudhuri S., Ganti V., Motwani R.. Robust identification of fuzzy duplicates. ICDE 2005: 865-876.

[30] Wang Y. R., Madnick S. E.. The inter-database instance identification problem in integrating autonomous systems. ICDE 1989: 46-55.

[31] Hernandez M. A., Stolfo S. J.. Real-world data is dirty: Data cleaning and the merge/purge problem. Data Mining and Knowledge Discovery, 2(1):9-37, January 1998

[32] Galhardas H., Florescu D., Shasha D.. Declarative data cleaning: Language, model, and algorithms. VLDB 2001: 371-380.

[33] Zhang Z., He B., Chang K. C.-C.: Light-weight Domain-based Form Assistant: Querying Web Databases On the Fly. VLDB 2005: 97-108

[34] Elmagarmid A. K., Ipeirotis P. G., Verykios V. S.: Duplicate Record Detection: A Survey. IEEE Trans. Knowl. Data Eng. 19(1): 1-16 (2007)

[35] Elfeky M., Verykios V., and Elmagarmid A.. TAILOR: A record linkage toolbox. ICDE 2002, 17–28.

[36] Arasu A., Ré C., Suciu D.: Large-Scale Deduplication with Constraints Using Dedupalog. ICDE 2009: 952-963

Towards Task-Organised Desktop Collections

Yukun Li
School of Information
Renmin University of China
liyukun@ruc.edu.cn

David Elsweiler
Department of Computer Science,University of Erlangen
david@elsweiler.co.uk

Xiaofeng Meng
School of Information
Renmin University of China
xfmeng@ruc.edu.cn

ABSTRACT

In this paper we promote the idea of automatic task-based document organisation. To make this possible we present a simplified task model and evaluate a number of algorithms for detecting which documents are associated with particular tasks. Our findings demonstrate the feasibility of such an approach, but work must be done to improve the performance for practical implementation.

1. INTRODUCTION

As people acquire ever more information as a result of personal and work activities, the management of this information becomes a serious problem and an important research issue [6]. Previous literature suggests that the tasks people perform and the activities associated with personal information plays an important role in how the information will be managed and re-found [4, 11]. There are also a number of anecdotal scenarios that highlight the importance of user task and activities in Personal Information Management (PIM) behaviour:

(1) We know that people often multi-task and experience difficulties when switching between tasks [3]. To support multi-tasking it would be useful for the user to have access to resources associated with each task; (2) When restarting a personal computer, especially after a change in workplace (i.e. from office to home) a user may need to access the files related to specific tasks in order to continue his work; (3) When starting a new task similar to a task already completed, it may be helpful to access documents associated with the previous task for reference purposes; (4) When an experienced user wants to help someone with less experience complete a task, it is often useful, for demonstration purposes, to re-find personal documents associated with the completion of this or a similar activity in the past. (5) When writing a progress report or summary of work completed, it is often useful to review completed activities retrospectively and see which documents have been created, used or modified.

To support these kinds of scenarios PIM systems need to be able to associate documents with user activities. Currently the only way to achieve this is to rely on user annotation and filing. Nevertheless, there is a large body of evidence suggesting that people are not willing or able to achieve a consistent organization, which meets all of their needs over long-periods of time [8, 9, 13].

It would be very advantageous if tasks were able to be modelled in such a way that they could be automatically detected and appropriate documents and data items could be associated with activities. It is possible, for example, that such a model could be used to implement a task-based replacement for or enhancement to the the traditional desktop metaphor. Nevertheless, there several challenges that need to be faced before such as model can be realised.

Firstly, it is difficult to define the concept of a task. Tasks can be considered at various granularities e.g. a project could be considered a task at a high-level, but would naturally consist of many sub-tasks and sub-sub tasks. Tasks can also be of various complexities [2]. Evaluating any model created or algorithm used to detect tasks is also challenging because in addition to the many problems associated with PIM evaluations, such as personalisation and privacy problems [4], there are no publicly available data sets, nor any available benchmarks or frameworks for evaluation.

In this paper we present our work in addressing these challenges. We formally define the concept of a user task and use the definition as the foundation for a task-based model for PIM. We also outline our thoughts on evaluation and describe some early results from experiments performed to test the performance of various algorithms which associate resources with tasks.

1.1 Related Work

PIM is the area of research concerned with how people store, manage and re-find information [6]. It is a multi-disciplinary field and researchers have been actively trying understand user behaviour, such as how people interact with information and tools [9, 13], what psychological factors are important [1] and how improved tools can support user behaviour and needs [10, 5].

A large amount of PIM behaviour is performed on desktop computers, where the standard PIM model is based on the office metaphor of files and folders. Several scholars have identified the limitations of this model and suggested moving to other means of interacting with information [7, 5]. One suggested method has been to organise information based on the activities or tasks the user performs. Studies have shown the importance of activities and tasks to PIM, including behavioural strategies to allow tasks to be managed [13] and indicating that while working, people regularly need to switch between concurrent tasks [3] and have difficulty managing resources as a result. The general consensus is that the desktop metaphor provides inadequate resources for task management and switching [11] and as a result,

Copyright is held by the author/owner(s). SIGIR'10, Workshop on Desktop Search, July 23, 2010, Geneva, Switzerland .

several prototype systems have been designed to assist with these situations, either by visualising resources in different ways e.g.[11] or making tasks the basis for organisation [12].

The common theme and, in our opinion, the major limitation of the task-based solutions proposed to date is that they all require the user to indicate which resources are associated with each task, which places the cognitive burden on the user in the same way the desktop metaphor does [8]. In this paper we explore methods to automatically associate resources with tasks. We first present a definition of a task and a model from which resources can be associated with tasks and continue propose and evaluate a number of algorithms for automatic association.

2. TASK MODEL

The basic building block for our tasks is a personal data item (PDI), which we define as *an item which has been created, accessed or modified by a person and is the result of user interaction*.

According to this definition, a file, an email or a folder can all be regarded as PDIs. What the definition also makes explicit is that PDIs only refer to items that the user has interacted with and that have not automatically been generated by software. This is important as in this work we focus on illustrating a task model and methods for identifying tasks based on desktop collections. The aim of any model would be to exploit user recollections for activities associated with PDIs and users will not remember any item with which they have had no explicit interaction.

A task has three basic elements: a goal (i.e. the thing to be accomplished), process (i.e. the activities performed to achieve the goal), and time (the period of time taken to complete the task). Each of these elements must be included and formalised in the task definition.

User activities with desktop computers can be divided into two types based on their interactions. ***Read-only activities*** involve only accessing a PDI, and include activities, such as reading documents, listening to music, etc.. The second type of activity, ***creative activities***, involve generating new or updating existing PDIs. Thus, the goal of a creative activity can be materialized as the set of PDIs modified by the user. In this paper, our focus is on this second category of tasks. According the above criteria, a personal task can be described as follows:

Definition 2.1 (Personal Task). . *A personal task PT is described as a 4-tuple (TD, DI, OL, TL), where TD represents a written description of the task and is described as a vector of tokens. DI represents the related data items, and is denoted as a 2-tuple (GI, RI), where GI (goal items) is a set of items generated by users during the process of completing the task, and RI (referenced items) is a set of items accessed in order to complete the task. OL (operations list) is a sequential list user operations performed to complete the task. TL describes the life-cycle of the task and is denoted as a 2-tuple (TS, TE), where T_s is the start time of the task and T_e is the end time.*

Naturally, as mentioned above, tasks can be of varying levels of complexity. For example, notifying colleagues about an upcoming meeting would be a task requiring the creation of only one item, i.e. an email. Preparing a grant application, on the other hand, could also be considered a task, but may involve the creation or modification of multiple items. Consequently we consider two types of tasks: *Simple Tasks*, which have a single goal item and *Complex Tasks*, which have multiple goal items.

In practical terms, a complex task can be regarded as the combination of many simple tasks. Therefore, if simple tasks can be identified then this could form the basis task identification in general. For this reason we focus here on identifying simple tasks.

3. IDENTIFYING PERSONAL TASKS

According to the definition above, there are several elements of a simple task that need to be identified: the task description, the life-cycle of the task, referenced PDIs and a task goal PDI. From this list, detecting referenced PDIs is the most challenging problem. For simple tasks, any created or modified PDI could be considered a task goal item. A task description can be generated by applying techniques, such as TF-IDF, to the PDI content or utilising a file or folder name (for non-text-based PDIs). The life-cycle information can also be easily attained by taking the created and modified times of the goal item (GI).

Below, we will outline a number of basic methods for identifying reference files based on PDI properties and patterns of user interactions. We evaluate the performance of the various methods in Section 4.

3.1 Life-cycle based method

A simple method of detecting PDIs associated with tasks is to take all files accessed within the life-cycle of a task as its references. This method will achieve perfect recall i.e. all of the files associated with a task will be detected, but the fact that people multi-task and continue tasks over long time periods will inevitably result in very low precision i.e. many inappropriate files being taken as task references. Nevertheless, the high recall property makes the approach a useful baseline algorithm for evaluations.

3.2 Directory-based method

We know from studies of folder organisations that people often organise their information items based on activity [9, 13]. Thus, a simple approach to associating PDIs with tasks is to utilise the information implicitly provided by the user through his folder structure. Given a task goal file, we can take all files located in the same folder (as well as the folder itself) as its references. An obvious limitation of the method is that files can be organised in other ways (e.g. time, people etc.) so it is likely that in some cases appropriate RIs will be located across folders – such references will not be detected using this method.

3.3 Sequential distance-based method

Another assertion we can make regarding user behaviour is that items accessed and modified within similar time periods relate to the same task. If this is true then the sequential distance – the number of items accessed or modified between two items in a sequential access list – will reflect, to some degree, the relationship between the items.

This method is simple to implement, but also has limitations. We expect it to work well when the user does not multi-task, but poorly for multi-tasking situations. Finding an appropriate threshold distance is also a problem. We assume that a small threshold will lead to higher precision

but lower recall and a larger threshold will lead to a low precision but higher recall. We provide some data regarding threshold selection in the experimental section below.

3.4 Operational pattern-based method

A further assertion we can make about user-behaviour is that after referring to a file, the user will modify the goal item of the task. If this is the case we can expect modify operations on the goal item to be surrounded by read operations of reference items for that particular goal item. Inspection of user interaction logs (see experiment below), seems evidence this assertion. We can exploit this with the following algorithm

Algorithm 1 Operational Pattern Based Algorithm

Input: An existing access list $L' = X_1, X_2, ..., X_n$; An existing task set TS; Latest accessed file X_{n+1};
Output: An updated task set TS.

procedure $Identify\ Simple\ Task(L', TS, X_{n+1})$
  if then $X_{n+1}.operation = "Modify"$
    if then $\nexists t \in TS \wedge t.goalitem = X_{n+1}.item$
      Create a new task t
      Add $X_n.item$ into t.reference
    else
      Find a task t where $t.goalitem = X_{n+1}.item$
      Add $X_n.item$ into t.reference
      Find $MSL\ L'' = (X_k, X_{k+1}, ..., X_{n+1})$
      Add $X_i.item (k+1 \leq i \leq n)$ into t.references
    end if
  end if
end procedure

When a modification operation is detected,i.e. a GI is processed, the references for that task will be updated. First, the algorithm will find the latest read operation for the GI within the access list L'. Following this it checks to see if two records in L' exist, which point to the same DI. If not, the access list is denoted as a Minimum Sequential Loop(MSL) and all DIs accessed within this MSL are regarded as references of the GI. Unlike the sequential distance-based method, here no threshold value needs to be specified in advance. However, like the SD method this approach has the disadvantage that multi-tasking behaviour may negatively affect performance.

4. EVALUATING PERFORMANCE

A dataset collected via a naturalistic investigation formed the basis of our evaluation. By developing and deploying a custom-designed piece of software built based on APIs for Microsoft Windows operating system, we captured user file accesses during the course of normal PC usage. We recorded two types of operation: "read-operations" where items were accessed or read and "modify-operations", where items were newly created or modified. For each operation we stored the associated file name, the directory-path and a timestamp.

8 participants (4 male, 4 female, aged between 25 and 40), volunteered to take part over a period of approximately one year. The participants were all researchers or research students at a major Chinese university and although they represent a relatively homogeneous population, they are all busy people who struggle with multi-tasking and PIM in general. The data represented their activities with their office computers.

In total 54,545 operations were recorded (avg per participant =6818 st dev =3947). 79% of the operations were reads and and 21% were modify operations.

To create a "gold-standard" from which to compare the performance of the algorithms, we conducted a second experimental phase where the same participants were asked to manually indicate the referenced items for a given set of goal items. We selected 10 goal items per participant so that those chosen included both files modified often and only a few times. To account for the difficulties in retrospectively annotating referenced files, the participants used software, which showed a goal item and a list of potential referenced items (selected by the life-cycle based method). The participants had access to meta-data about each of the items and could also open the files to examine the content before deciding if it was a referenced file. Additionally, we asked the participants to explain the relationship between the item and the goal item. We used these data as a means to evaluate the algorithms described in Section 3.

5. RESULTS AND DISCUSSION

Figure 1 depicts the performance achieved by the various algorithms. As these graphs show, it was possible to attain high recall scores – all of the algorithms achieved average scores of 0.71 and above. However, precision was harder to attain with the best performance being achieved by the OP method (0.43).

The Directory-based approach achieved the lowest recall (0.71), with some referenced files evidently being stored in different directories. Figure 1(b) shows that some participants tended to place task-related items in the same folder (e.g. user 5) and this allowed high recall to be achieved. What is also clear from the low precision scores, is that all of the participants had files in the examined directories, which they did not consider to be related by task.

Figure 1(f) shows how the sequential distance threshold (sd) influenced the performance for the SD method. Confirming our hypotheses, when the threshold=1, we achieved the optimum F-score (0.34) and precision (0.27), but as the threshold is increased, the F-score and precision deteriorated as recall improved. This result indicates that in a practical implementation of the algorithm, it would not be necessary to use a high threshold to achieve best performance.

Figure 1(a-d) shows how the algorithms performed across users, demonstrating that it was easier to achieve higher precision with some users e.g. user 6. We hypothesize that users, for which better performance could be attained, tend to multi-task less often. If this is true it highlights a paradoxical situation where the people most likely to need support are also the most difficult to provide support for. This is an obvious weakness in the methods proposed.

What our results do show, however, is that there is potential for algorithms to automatically related items by task. Our relatively simple algorithms were able to achieve good recall although improvement is needed in terms of precision. To achieve this improvement we need to investigate ways of combining evidence from the various sources exploited in the basic algorithms presented here. Even with current performance levels, we believe that, if embedded within an appropriate user interface, the algorithms could be useful in assisting users to manually organise their documents. These

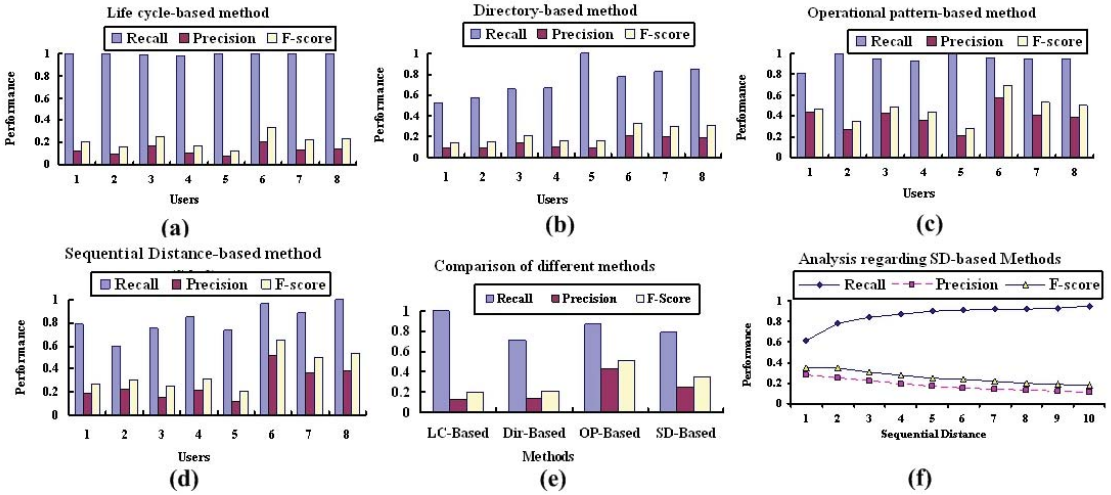


Figure 1: The Performance Achieved by the Various Algorithms

two threads represent our future plans for this work.

6. CONCLUSIONS

In this paper we have used examples from the literature to demonstrate the potential benefits of a task-oriented approach to PIM. We suggested that to realise these benefits tasks should be detected automatically and to achieve this we proposed a task model and several methods of detecting resources associated with tasks. Our model abstracts and simplifies the concept of a task, allowing the performance of the proposed algorithms to be evaluated. The evaluation results suggest that it may be feasible to achieve an automatic means of task-detection, but work must be done to improve the performance for practical implementation.

7. ACKNOWLEDGMENTS

This research was partially supported by the grants from the Natural Science Foundation of China (No.60833005); the National High-Tech Research and Development Plan of China (No.2009AA011904); and the Doctoral Fund of Ministry of Education of China (No. 200800020002).

8. REFERENCES

[1] D. Barreau, *Special issue on the social and psychological aspects of personal information management*, Journal of Digital Information **10** (2009), no. 5.

[2] K. Byström and K. Järvelin, *Task complexity affects information seeking and use*, Information Processing and Management **31** (1995), no. 2, 191–213.

[3] M. Czerwinski, E. Horvitz, and S. Wilhite, *A diary study of task switching and interruptions*, CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems (New York, NY, USA), ACM Press, 2004, pp. 175–182.

[4] D. Elsweiler and I. Ruthven, *Towards task-based personal information management evaluations*, SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (New York, NY, USA), ACM Press, 2007, pp. 23–30.

[5] E. Freeman and D. Gelernter, *Lifestreams: a storage model for personal data*, SIGMOD Record (ACM Special Interest Group on Management of Data) **25** (1996), no. 1, 80–86.

[6] W. Jones and J. Teevan (eds.), *Personal information management*, Seattle: University of Washington Press, 2007.

[7] V. Kaptelinin and M. Czerwinski, *Beyond the desktop metaphor designing integrated digital work environments*, MIT Press, 2007.

[8] M.W. Lansdale, *The psychology of personal information management.*, Appl Ergon **19** (1988), no. 1, 55–66.

[9] T. W. Malone, *How do people organize their desks?: Implications for the design of office information systems*, ACM Trans. Inf. Syst. **1** (1983), no. 1, 99–112.

[10] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich, *Data mountain: using spatial memory for document management*, UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology (New York, NY, USA), ACM Press, 1998, pp. 153–162.

[11] G. Robertson, G. Smith, B. Meyers, P. Baudisch, M. Czerwinski, E. Horvitz, D. C. Robbins, and D. Tan, *Beyond the desktop metaphor*, ch. Explorations in Task Management on the Desktop, pp. 101–138, MIT-Press, 2006.

[12] S. Stumpf and J. Herlocker, *Tasktracer: Enhancing personal information management through machine learning*, Proc. Workshop on Personal Information Management, SIGIR, 2006.

[13] S. Whittaker and C. Sidner, *Email overload: exploring personal information management of email*, CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems (New York, NY, USA) (M. J. Tauber, ed.), ACM Press, 1996, pp. 276–283.

Exploring Desktop Resources Based on User Activity Analysis

Yukun Li, Xiangyu Zhang and Xiaofeng Meng
School of information, Renmin University of China
Beijing, China
liyukun@ruc.edu.cn, zhangxy@live.com,
xfmeng@ruc.edu.cn

ABSTRACT

Relocation in personal desktop resources is an interesting and promising research topic. This demonstration illustrates a new perspective in exploring desktop resources to help users re-find expected data resources more effectively. Different from existing works, our prototype OrientSpace has two features: automatically extract and maintain user tasks to support task-based exploration, and support vague search by exploiting associations between desktop resources.

Categories and Subject Descriptors: H.5.2 [User Interfaces]:Prototyping.

General Terms: Design, Human Factors, Management.

Keywords: Desktop resources, Task exploration, Association exploration.

1. INTRODUCTION

Nowadays, the most widely used approaches to explore desktop resources is by Windows Resource Explorer(WRE) and Desktop Search Tools(DST). WRE demands users to recall precise path information, and DST demands users to remember exact key words. However, there're many occasions when users can not remember the promising keywords or pathes. In fact, users often expect to relocate desktop resources based on user tasks([1],etc). Some existing works make good efforts to tackle this problem, like prototype Haystack [2] and Phlat [3]. But these works paid little attention to the role of user activities for personal data relocation. This demonstration is try to overcome the disadvantages of the extisting works, and help users to explorer personal desktop resources based user activities and associations between desktop resources.

2. SYSTEM OVERVIEW

Figure 1 shows the interface of OrientSpace system. It has two major features: task-based resources exploration and association-based resources exploration.

Task-based Resource Exploration. In this work we define each task as a set of desktop files related to generating a special personal document, and identify each task based on analyzing user access sequential list on desktop resources. The left area of figure 1 shows a list of user tasks ranked by time, which are extracted automatically through detecting and analyzing user operations. By clicking one of the tasks, user will get the documents related to this task. This would be especially useful for those people who don't spend enough time in organizing their documents.

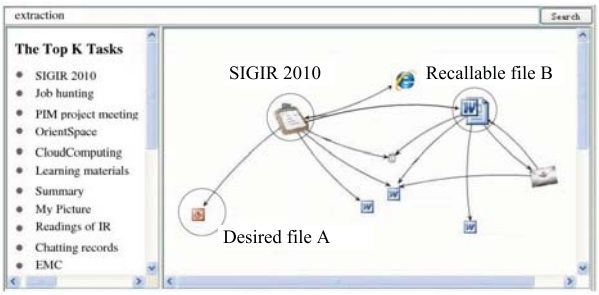


Figure 1: System Interface

Association-based Resource Exploration. The right area of figure 1 represents an association graph of documents and tasks. This is very useful when a user can not remember the right keyword and directory for the desired file, but can remember some information about other files with relation to it. As shown in figure 1, the user expects to find file A, and can not remember its keywords and directory, but can remember a keyword "extraction" of another document B associated to a same task "SIGIR 2010" with the desired file A. She can first find B by keyword "extraction", then relocate file A by this association-based explorer. Currently supported associations by OrientSpace include: have common keywords, belong to the same task, attached to email and so on.

3. ACKNOWLEDGMENTS

This research was partially supported by the grants from the National High-Tech Research and Development Plan of China (No:2007AA01Z155).

4. REFERENCES

[1] P. Vakkari. Task based information searching. In: Cronin, B. (Ed.) [ARIST 37]: 413-464, 2003.

[2] D.R. Karger et al. Haystack: A General-Purpose Information Management Tool for End Users Based on Semistructured Data, *CIDR 2005:13-26.*

[3] E. Cutrell et al. Fast, flexible filtering with phlat –Personal Search and Organization Made Easy. CHI 2006:261-270.

Copyright is held by the author/owner(s).
SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
ACM 978-1-60558-896-4/10/07.

计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
47(Suppl.): 433-437, 2010

TaijiDB:一个双核云数据库管理系统

胡享梅 赵 婧 孟小峰 王仲远 史英杰 刘兵兵 王海平
(中国人民大学信息学院 北京 100872)
(hxm2008@ruc.edu.cn)

TaijiDB: A Dual-Core Cloud-Based Database System

Hu Xiangmei, Zhao Jing, Meng Xiaofeng, Wang Zhongyuan, Shi Yingjie, Liu Bingbing, and Wang Haiping
(*School of Information, Renmin University of China, Beijing* 100872)

Abstract Taiji is a Chinese cosmological term, which means two modes can be uniform relatively. In order to leverage the advantages of cloud storage based on master-slave and p2p structures, we propose a project called Taiji, which is a dual-core cloud-based database system. This system can support SQL to manage the Big Data in the cloud.

Key words cloud computing; cloud-based database system

摘 要 太极是一个中国古代哲学术语——即两种模式可以相对统一.利用基于云存储的主从结构和点对点结构各自的优点,融合两种结构,构建了一个双核的云数据库管理系统——太极.系统支持使用SQL语言对云数据库系统中的海量数据进行管理.

关键词 云计算;云数据库系统

中图法分类号 TP311.13

随着数据量的迅猛增长,如何存储和管理海量复杂数据已成为一个亟待解决的挑战性问题.云计算应运而生,它改变了数据存储的基础架构.现有的云计算系统包括:亚马逊的弹性云计算(EC2)[1]、IBM的蓝云[2]和谷歌的GFS[3].它们都采用了弹性资源管理机制并提供很好的可扩展性.另外,也有一些开源项目,譬如Apache Hadoop项目的HDFS[4]和HBase[5]以及Cassandra[6].HDFS和HBase是谷歌GFS和BigTable[7]的开源实现,Cassandra则是亚马逊Dynamo[8]分布式实现和Bigtable列簇数据模型的融合.

云计算系统通常有2种底层结构:主从结构和点对点结构.表1展示了基于上述两种结构的云计算系统在各方面的对比.

表1 基于主从与点对点结构的云计算系统比较

	主从结构	点对点结构
CAP[9]	通常关注于一致性和高可用性	通常关注于可用性和划分容错性
数据写操作	如果Region服务器意外停机,则在数据重新分布前,写操作会被阻止	每个节点都是平等的,因而"写操作永远不会失败"
MapReduce	支持MapReduce框架	不支持MapReduce
系统性能	Master节点可能会成为瓶颈	在通信负载较大时,系统性能会迅速下降
应用场景	适合分析型数据管理应用	适合事务型数据管理应用

收稿日期:2010-06-25
基金项目:国家自然科学基金项目(60833005,60573091);国家"八六三"高技术研究发展计划基金项目(2007AA01Z155,2009AA011904);教育部博士学科点专项科研基金项目(200800020002)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

从表 1 中可看出,这 2 种结构各具优势,但也存在自身结构性问题.因此,我们希望构建一个混合系统,充分利用上述 2 种底层结构的优势.

在另一方面,除了 HIVE[10] 系统,现有的云计算系统都不支持 SQL 语言,这就意味着原有的基于传统 DBMS 的数据管理系统无法平滑地迁移到云存储平台上,进而导致云存储无法大规模普及.同时,HIVE 系统仅支持较为简单的类 SQL 查询语句.因此,我们需要在云数据管理系统中解决这一挑战性问题.基于上述原因,我们提出构建一种"双核"的基于云的数据库管理系统.

太极是中国古代哲学术语,它包含阴阳两种状态.太极具有两层意思:一方面,阴阳两种状态相对统一;另一方面,阴阳轮转,派生万物,从混沌变为清晰.我们认为这两个特点与系统的目标极为相似:首先,我们希望结合不同架构,利用各自优势;其次,我们需要管理海量的纷繁复杂的数据,使之有条有理,便于查询.因此,我们以太极来命名我们的工程.

我们的贡献主要包括:1)太极建立了一个适配层来桥接底层的主从结构和点对点结构.上层通过统一的 API 调用进行操作,底层的差异对上层透明.2)太极支持部分 SQL 查询语句,包括建表、插入数据、选择数据、删除数据和数据表等.对 SQL 语句的支持有助于降低应用开发的难度,从而使云计算和云服务等到更好的推广和应用.

1 应用场景

基于云的数据库系统旨在支持下一代数据存储和管理.图 1 展示了移动应用领域中 WAP 应用框架.移动终端通过 WAP 服务器访问网络站点.WAP 服务器产生话单数据并通过 FTP 传输到话单

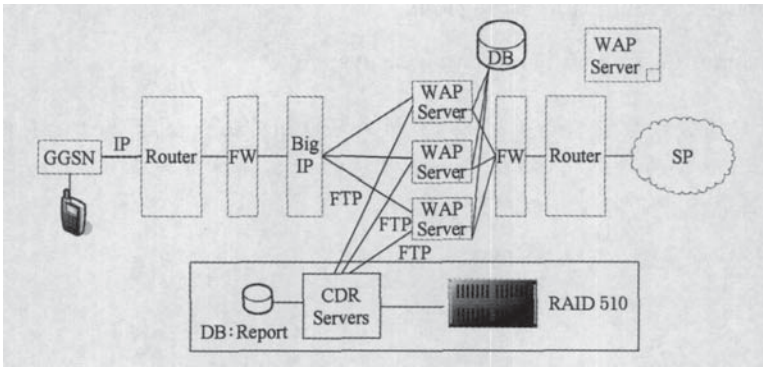


图 1 WAP 应用体系结构

数据服务器.话单数据在通信过程中产生,用于描述通信细节.该数据主要面向两类应用:1)个人通信详情查询;2)数据分析及决策支持.话单数据的规模巨大:每 5 min 产生的数据量近 4.5 GB.随着手机用户数量的增长,在不久的将来将会产生越来越多的话单数据.

如何存储和管理这些数据将是一个极具挑战性的难题.目前各数据中心普遍采用昂贵的高端数据服务器进行集中式存储,因此企业只能在磁盘中保留最近 3 个月的话单数据.

话单数据适于存储在基于云的 DBMS 中,基于云的 DBMS 可以部署在廉价、低端并且自适应的硬件上.为了满足数据导入和用户查询的需求,话单数据管理系统在数据读和写操作上都必须有高性能.

我们使用太极获取来自 WAP 服务器的话单数据并进行分析,分析结果交由报表数据库存储.话单数据表结构如表 2 所示:

表 2 话单数据结构表

属性	类型
service_ID	string
service_type	int
**MSISDN*	string
IP_address	string
**ts_start*	datetime
ts_end	datetime
ts_start_content_fetching	datetime
ts_edn_content_fetching	datetime
session_or_ID	int
fetched_URL	string
online_time	long

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

若用户想查询一天之中访问过的所有 WAP 站点,我们可通过执行如下 SQL 查询语句获得结果:

SELECT *MSISDN*,*ts_start*,*ts_end*,*fetched_URL*

FROM *COR*

WHERE *MSISDN*='1395451XXXX'

and *ts_start*≥'2009-09-15 00:00:00.000'

and *ts_end*<'2009-09-16 00:00:00.000'.

另一方面,若运营商需要统计一个 URL 每天被访问的时间总和,可执行以下 SQL 语句获得结果:

SELECT *fetched_URL*, *sum*(*ts_end* − *ts_start*)

FROM *CDR*

WHERE *ts_start* ≥ '2009-09-15 00:00:00.000'

and *ts_end*<'2009-09-16 00:00:00.000'

GROUPBY *fetched_URL*.

为了满足该应用场景的所有需要,我们设计了太极数据库管理系统.太极的框架主要包括 3 层:数据存储层、查询处理层和应用层,如图 2 所示:

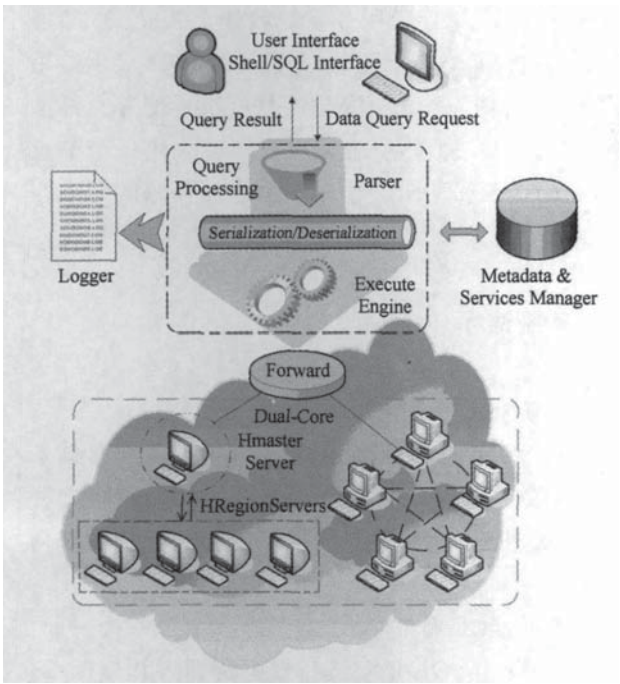


图 2 太极的框架图

顶层是应用层.太极支持 SQL 以便于复杂的数据管理,同时便于应用开发商实现其服务向云计算的无缝迁移.该层通过查询语言以及丰富的 API 支持多种 Web 应用.客户端可通过 shell 接口或 SQL 接口提交用户查询.

中间层是查询处理层.太极通过该层将 SQL 语句解析为原子操作序列,序列化或反序列化数据并调用执行引擎来完成操作.同时,该层包含元数据和服务管理器,日志等用于监控云数据库系统的状态.

底层是双核存储层.通过可配置的存储架构为用户提供灵活的存储管理数据方式.底层可使用主从和点对点的存储结构,综合二者的优点,为上层提供统一的 API 实现.该层支持云存储的特性:备份、并行性、容错性、主键划分和同步.

太极为云数据管理提供强大的双核模型,并为云上的应用开发提供便利的方式——标准的 SQL 支持.同时,充分利用双核设计的优势和根据应用需求(如事务、一致性和负载均衡)来自动选择合适的架构都极具研究意义,这将在本文的第 4 部分讨论.

2 系统架构

图 3 显示了太极的组成部分以及 Hadoop 和 Cassandra 的通信.太极主要包括 5 个组成部分:

1.前端接口模块."双核"云数据库管理系统提供 SQL 接口、Shell 和应用程序编程接口(API).用户不仅可通过 SQL 接口得到记录形式的结果,还可以执行文件级别的操作,如从文件中进行数据载入,以及使用 API 接口将数据导出到文件中.

2.查询处理模块.为前端接口提供两种查询接口:基于 SQL 的查询接口和基于编程的查询接口.当 SQL 语句被提出时,SQL 处理器进行 SQL 解析和查询优化,并且将 SQL 语句翻译成命令,调用统一的 API 和存储层进行通信.如果编程应用接口被用户调用,它们也会依照客户端库翻译成执行计划进行执行.

3.统一执行引擎模块.为上层提供统一的应用,而无需关注两种存储模式的不同处理方法.它从统一 API 提取标识参数,并通过 API 触发存储管理器.

4.存储管理器模块.负责控制数据存储位置.用户可以指定使用 HBase 或者 Cassandra 作为存储引擎.

5.运营维护模块.存储数据库元信息.元信息被 SQL 处理器和驱动器使用,还可用于监控系统的操作和运行状态.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

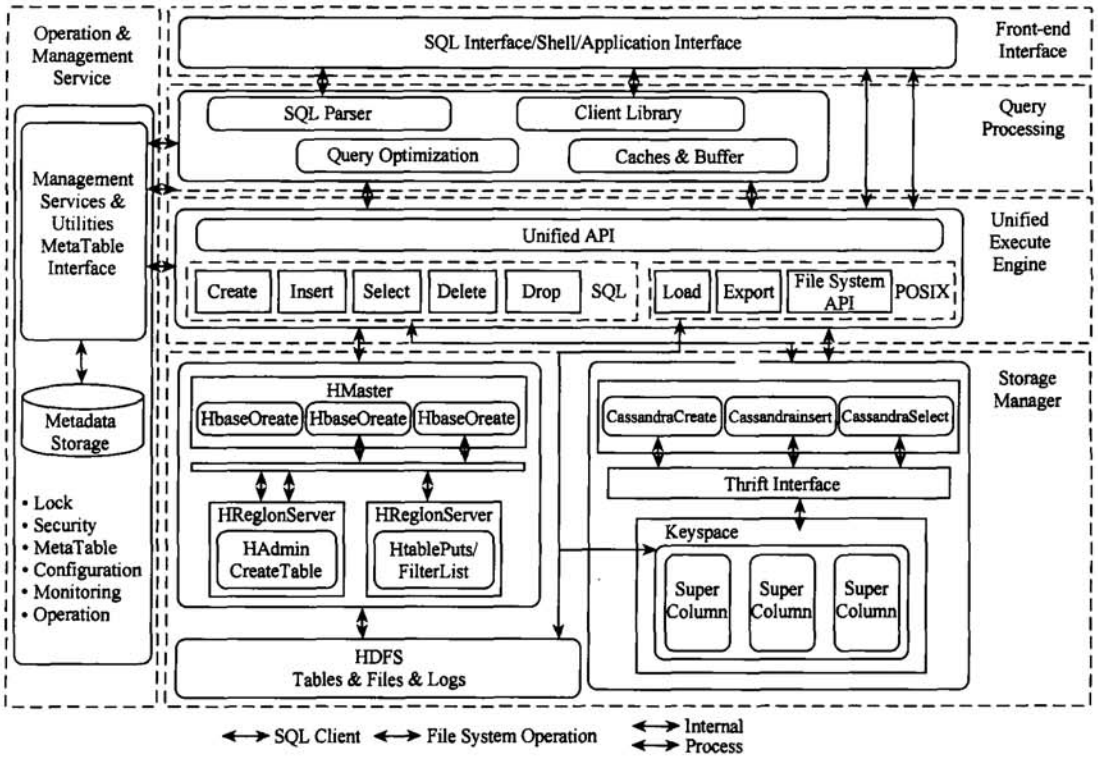


图 3 “双核”云数据库管理系统的体系结构

2.1 存储管理模块

太极拥有双核存储引擎. 它支持这些特性：备份、并行、容错、键值划分和同步. 在此部分中，有两种存储模式：主从模式和点对点模式. HBase 以主从模式组织，主服务器管理文件系统命名空间，控制用户的文件访问. 集群的数据节点负责执行具体读写操作，同时根据主节点的命令进行数据块的创建、删除和复制. Cassandra 基于点对点结构，使用 Gossip 协议管理集群成员. 在主从模式中，我们为一张表使用一个列族，表中的每一列与列族中一个限定词匹配. 在点对点模式中，整个数据库是一个关键词空间，表对应超级列，超级列下的列与用户表的属性列相对应. 在“双核”云数据库管理系统中，由于 Cassandra 无法动态添加列族，我们并未使用列族表示一张表.

2.2 运营维护模块

在“双核”云数据库管理系统中，运营维护模块负责元数据管理、操作管理和系统监控. 系统包含两种元数据：1）表结构信息，如表名、字段名、字段类型、表存储等；2）用户信息，如用户名、密码、权限等. 由于元数据规模有限且多服务于事务操作，我们采用 RDMBS——MySql 存储. 我们通过元数据接口对操作元数据. 管理服务系统收集监控信息，如资源、系统健康、数据配置等，并且通过 GUI 展示给用户. 同时它具备系统报警功能并可重置配置文件参数. 通过 GUI 操作，管理员可开启或关闭每个节点上的 DBMS 和 OS，创建基于角色的资源队列或在集群中动态地添加（或删除）节点以适应负载变化.

3 系统演示

系统演示主要包含以下 3 部分：

1. “双核”存储. 我们将展示太极的双核存储系统的有效性. 应用可在主从结构和点对点结构 2 种模式之间进行平滑地切换，无需对 API 调用做任何修改.

2. 功能. 我们将演示的功能主要包括 2 个方面：1）建表、使用 SQL 语句进行数据的插入和选择；2）通过 API 在系统和文件之间进行数据的导入和导出.

3. 性能. 我们将演示太极在电子通信场景下的应用. 我们使用话单数据作为我们的测试实例. 演示使用的话单数据超过 3.8TB，分布存储在 20 个节点上.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4 未来工作

太极目前正被应用在在电信领域管理不同类型的话单数据.用户可以在SQL语句中指定底层数据的存储结构——使用主从机构或者点对点结构.我们认为两种类型的数据模式和数据管理应该更加紧凑地融合在一起，而不仅仅是在存储层，同时，查询计划和存储模式的选择可以更加智能化.我们的未来工作主要包括：

1.建立一种根据列类型、列大小和已存储在表中的数据等来自动选择存储模式的服务.

2.目前，太极中一个表只能只采用一种存储结构.我们打算将表和备份数据本别采用不同模式进行存储，进而丰富我们的查询优化算法.

3.我们打算将论文文献[11]中提出的多维索应用到太极中以优化多列查询，同时采用该论文中的基于代价估计的索引更新策略来有效地更新索引结构.

参 考 文 献

[1] Lynch M. Amazon elastic compute cloud (Amazon ec2). [2010-06-25]. http://aws.amazon.con/ec2/

[2] IBM. IBM introduces ready-to-use cloud computing. [2010-06-25]. http://www-03.ibm.com/press/us /en/pressrelease/22613.wss

[3] Ghemawat S, Gobioff H, Leung S T. The google file system //Proc of SOSP'03. New York: ACM, 2003: 29-43

[4] HDFS. [2010-06-25]. http://hadoop.apache.org/hdfs/

[5] Hbase. [2010-06-25]. http://hadoop.apache.org/hbase/

[6] Cassandra. [2010-06-25]. http://incubator.apache.org/cassandra

[7] Chang F, Dean J, Ghemawat S, et al. Bigtable: A distributed storage system for structured data //Proc of the 7th Conf on USENIX Symp on Operating Systems Design and Implementation. Berkeley, CA: USENIX Association, 2006: 205-208

[8] DeCandia G, Hastorun D, Jampani M, et al. Dynamo: Amazons highly availbale key-value store //Proc of the 21st ACM Symp on Operating Systems Principles (SOSP'07). New York: ACM, 2007: 205-220

[9] Fox A, Brewer E A. Harvest, yield, and scalable tolerant systems //Proc of the the 7th Workshop on Hot Topics in Operating Systems. 1999: 174-178

[10] HIVE. [2010-06-25]. http://hadoop.apache.org/hive/

[11] Zhang X, Ai J, Wang Z, et al. An efficient multi-dimensional index for cloud data management //Proc of the CIKM Workshop on Cloud Data Management (CloudDB2009). New York: ACM, 2009: 17-24

胡享梅 女，1985年生，硕士研究生，主要研究方向为云数据管理.

赵 婧 女，1985年生，硕士研究生，主要研究方向为云数据管理.

孟小峰 男，1964年生，研究员，博士生导师，主要研究方向为Web数据管理、个人数据空间管理、XML数据管理、移动数据管理、闪存数据库技术以及云数据管理.

王仲远 男，1985年生，硕士，主要研究方向为数据集成、云数据管理.

史英杰 女，1983年生，博士研究生，主要研究方向为云数据管理.

刘兵兵 男，1987年生，硕士研究生，主要研究方向为云数据管理.

王海平 男，1987年生，硕士研究生，主要研究方向为云数据管理.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
47(Suppl.): 438-441, 2010

OrientPrivacy:移动环境下的隐私保护服务器

黄 毅 潘 晓 孟小峰
(中国人民大学信息学院 北京 100872)
(westyi@ruc.edu.cn)

OrientPrivacy: An Anonymizer for Privacy Preserving in Mobile Services

Huang Yi, Pan Xiao, and Meng Xiaofeng
(*School of Information, Renmin University of China, Beijing* 100872)

Abstract This demo presents OrientPrivacy, an anonymizer for privacy preserving in mobile services. It consists of three main components: 1) Simulated moving objects generator. It generates locations of mobile users and also accepts input of points of interested (POI) and the roadmap in which the users are. 2) Location cloaking module, in which a user's exact location is extended into a region according to the users' privacy requirements. 3) Anonymized results output module. It shows the cloaked regions on the roadmap, as well as the performance parameters for cloaking algorithms.

Key words privacy protection; location based service; mobile service

摘 要 系统展示了移动计算服务中的隐私保护服务器——OrientPrivacy. 它主要由3部分组成:1)移动数据生成模块. 可以模拟生成移动用户的查询和位置信息,导入用户兴趣点(POI)和用户所在城市的地图;2)隐私处理模块. 根据用户的隐私保护需求,采用隐私处理算法,将用户的精确位置转换成匿名区域,同时将用户的敏感查询进行隐匿;3)匿名结果展示模块. 展示隐私处理的结果,并展示移动用户的匿名区域和隐私服务质量参数.

关键词 隐私保护;移动计算;位置服务

中图法分类号 TP311.13

近年来,随着无线通信技术和移动定位技术的发展,基于位置的服务(location-based services, LBS)日益普遍. 学术界很多研究者关注如何在保证服务质量的前提下保护移动用户隐私. 总的来说,在位置服务中有2种隐私类型:用户位置隐私[1](保护特定用户的位置[2])和查询内容隐私[3](保护特定用户的查询内容[4]). 例如:假设张三通过手机向服务提供商(例如百度地图)发起了一个查询"离我最近的皮肤病医院在哪儿?". 从保护位置隐私角度看,张三想隐藏他的精确位置(例如他在一个餐厅或者酒吧);从查询内容隐私角度看,张三想隐藏他具体想查询的内容——最近的皮肤病医院.

同很多已有的工作[5-7]一样,本系统采用中心服务器结构. 它由移动用户、可信的隐私处理服务器和不可信的位置服务商(service provider, SP)组成. 移动用户向隐私处理服务器发起格式为(id,l,q,p)的查询,其中id为用户的标识,l为用户的位置,q代表查询内容,p是用户的最低隐私需求. 在收到用户查询后,隐私处理服务器将用户的标识id换成伪造的标识id',同时根据用户的隐私需求,调用隐私

收稿日期:2010-06-25
基金项目:国家自然科学基金项目(60833005,60573091);国家"八六三"高技术研究发展计划基金项目(2007AA01Z155,2009AA011904);教育部博士学科点专项科研基金项目(200800020002)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

保护算法为用户生成一个匿名区域 R,然后将匿名后的查询(id', R, q)发给位置服务提供商.位置服务提供商经过查询后,将查询结果返回给隐私处理服务器.最后,隐私处理服务器将该结果求精后返回给用户.在本演示系统中,我们模拟构建了隐私处理服务器 OrientPrivacy,模拟实现了根据用户隐私需求和场景保护移动用户隐私的目的.

1 OrientPrivacy 系统结构

OrientPrivacy 系统主要由 3 个部分组成(见图 1):1)移动数据生成模块.输入移动用户的位置,POI 和用户所在城市的地图数据;2)隐私处理模块.运用隐私算法将用户的精确位置 l 处理为匿名区域 R,并且将用户的敏感查询进行隐匿;3)匿名结果展示模块.它将用户的匿名区域在地图上展示出来,同时显示匿名处理的性能参数.本节主要关注隐私处理模块.

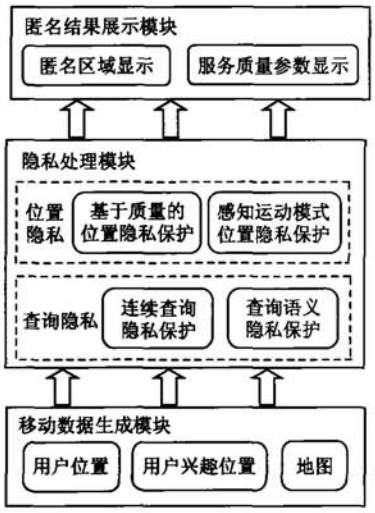


图 1 OrientPrivacy 系统结构

OrientPrivacy 将最近的工作中[1-4]集成在一起,包括位置隐私保护和查询隐私保护.其中基于质量的位置隐私保护算法和感知运动模式隐私保护算法针对位置隐私,连续查询隐私保护和查询语义隐私保护算法针对于查询隐私保护.本节剩余部分将分别介绍隐私处理模块中的关键技术.

1.1 基于质量的位置隐私保护算法

为了平衡位置隐私和服务质量二者之间的关系,文献[2]提出了基于质量的匿名模型.在这个算法中,用户可以指定个性化的隐私需求(即最小匿名级别)和服务质量需求(即最大匿名区域大小).算法维护一个有向图,根据这个有向图计算出用户的匿名集.假设新到请求为 r,有向图 r 邻居的匿名集合的出节点集和入节点集的大小分别是 k_o 和 k_i,r 的最小匿名级别为 $r.k$.如果 $k_o \geqslant r.k-1$ 和 $k_i \geqslant r.k-1$ 同时成立,那么 r 的出节点集的最小边界矩形(minimum boundary rectangle, MBR)就是 r 的匿名区域.详细的算法请参考文献[2].

1.2 感知运动模式的位置隐私保护算法

基于质量的位置隐私保护算法可以保护移动用户的位置隐私,但是,它没有考虑用户的连续移动对位置隐私泄露的影响.如果一个攻击者(例如 SP)可以收集用户一段时间的历史匿名区域并获知运动模式(例如速度),那么他就有可能攻破用户的位置隐私.这种攻击被称为位置依赖攻击.现有的大多数位置 K 匿名算法只考虑了快照式的隐私处理,并不能有效地防止位置依赖攻击.系统中使用 ICliqueCloak 算法,利用增量维护极大团集寻找匿名集的方法,从极大团中直接寻找匿名集.该算法考虑了移动对象的运动模式和连续位置更新.详细的算法请参考文献[1].

1.3 连续查询隐私保护算法

基于质量的位置隐私保护算法和感知运动模式的位置隐私保护算法都只适用于快照式查询,在不同的时间,匿名集里面的用户是不一样的.在连续查询的情形下直接应用上述算法是不足以保护查询隐私或者导致很差的服务质量.文献[3]采用 δp 隐私模型和 δp 差异模型来衡量用户隐私需求和服务质量.位置信息扭曲度(distortion)用匿名区域周长表示,进而映射为两个连续查询相似度(包括初始位置、运动速度和有效期):

$$SimDis(Q_1, Q_2) = \int_{T_s}^{T_{exp}} Distortion_{R_{12},t}(CS_{12}, RL_{12}, t)\mathrm{d}t, \quad (1)$$

其中,CS_{12}是由连续查询 Q_1 和 Q_2 组成的匿名集,R 是 Q_1 和 Q_2 在 t 时刻的匿名区域,T_s 是连续查询成功处理的开始时刻,T_{exp}是连续查询失效的时间.系统将查询根据上述公式计算出来的相似距离进行聚类,使得一个聚集中查询位置信息扭曲度最小.当有新查询到来或离开时,系统增量维护这些查询组成的聚集,并根据从这些聚集中直接寻找匿名集.详细的算法请参考文献[3].

1.4 查询语义隐私保护算法

连续查询隐私保护算法只关注连续查询隐私,

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

并没有考虑查询语义,因此可能会遭受查询同质性攻击(query homogeneity attack).极端情形下,如果一个匿名集中的所有查询的内容都是相同的,那么尽管连续查询保护了位置隐私,查询内容依然会泄露.为了防止查询同质性攻击,文献[4]提出了一种新的保护模型,称为 p-敏感度模型.它在保护用户位置隐私的同时考虑了查询敏感度和查询语义信息.通过在匿名集中加入一些非敏感的查询,达到迷惑攻击者的目的.p-敏感度模型可以用下面的公式来表示:

$$P(u^* \rightarrow Q_s)=\frac{|\{u^*.S_r \mid u^*.S_r \in Q_s\}|}{|u^*.S_r|}<p, \quad (2)$$

其中,$|u^*.S_r|$($|\{u^*.S_r \mid u^*.S_r \in Q_s\}|$)表示匿名集中(敏感)查询的个数.攻击者获知用户提出敏感查询内容的概率不会超过用户指定的概率 p.在算法的实现中,使用划分枚举树来找到符合 p-敏感度模型要求的匿名集.详细的匿名算法请参考文献[4].

2 演示环境与场景

2.1 演示环境

演示系统是一种在线访问的 Web 应用,利用 Java 语言开发,使用在线的 Google Maps for Flex 接口来展示地图.它的运行环境为:2.0 GHz CPU,2 GB 内存,Windows 7 操作系统.

2.2 演示场景

图 2 是系统的界面.它主要由 4 部分组成:A 部分显示地图、移动用户的位置以及经过隐私保护处理返回的匿名区域;B 部分是移动用户的控制界面,可以选择不同的隐私保护算法和指定不同的隐私保护参数;C 部分以表格的形式显示模拟用户的位置和服务质量参数;D 部分展示了模拟移动用户的位置更新频率.整幅图显示了使用基于质量的位置隐私保护算法处理 200 个移动用户中 30 个查询请求的匿名结果.当 OrientPrivacy 启动的时候,将会自动加载北京的地图和 POI,并且生成模拟的用户位置,用户可以设定系统中的隐私级别 K,最长匿名时间和最大匿名边界大小,然后,OrientPrivacy 运用用户指定的隐私保护算法计算出用户的匿名区域并在地图上显示.

图 2 系统界面

2.3 演示步骤

首先展示移动数据的生成.通过对服务器参数的修改,可以生成不同数量、不同运动模式和不同运动速度的移动物体,同时可以指定生成具有不同隐私需求的查询请求.

其次展示不同的隐私保护算法.点击地图上的移动物体,设定好隐私需求参数,系统会模拟向隐私处理服务器发起查询请求,然后服务器会根据选择的算法对查询进行隐私处理,并在界面展示隐私保护结果,即模拟移动用户的匿名区域和服务质量参数.基于用户的请求,可以几种不同的隐私保护之间切换,以查看不同隐私保护算法的差异.

最后展示隐私保护的结果.设定不同的移动用户位置和查询请求更新速度,界面的更新速度也会随之改变.通过点击用户的信息,可以查看用户的匿名结果和服务器的隐私处理参数.

3 未来工作

目前,OrientPrivacy 系统可以对用户发起的基于位置的查询进行位置隐私和查询隐私进行保护.其中用户的位置数据和查询都是模拟生成的,并非实际应用中的真实数据.以后考虑使用真实的移动数据和查询信息.此时为了能够处理大量真实用户的查询请求,服务器的性能会是一个瓶颈,可以考虑将它部署在云计算平台上.同时,应该智能地为用户配置隐私需求.目前系统中需要用户自行指定隐私需求,而现实中大多数用户并不了解隐私级别、最大匿名区域这些专业术语,隐私处理器应该根据用户所处的情景信息自动地配置用户的隐私需求,并选择适用的隐私保护算法进行处理.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

参 考 文 献

[1] Pao Xiao, Xu Jianliang, Meng Xiaofeng. Protecting location privacy against location-dependent attack in mobile services // Proc of the 17th ACM Conf on Information and Knowledge Management. New York: ACM, 2008: 1475-1476

[2] Xiao Zhen, Meng Xiaofeng, Xu Jianliang. Quality aware privacy protection for location-based services //Proc of the 12th Int Conf on Database Systems for Advanced Applications. Berlin: Springer, 2007: 434-446

[3] Pan Xiao, Meng Xiaofeng, Xu Jianliang. Distortion-based anonymity for continuous query in location-based mobile services //Proc of the 17th ACM SIGSPATIAL Int Conf on Advances in Geographic Information Systems. New York: ACM, 2009: 256-265

[4] Xiao Zhen, Xu Jianliang, Meng Xiaofeng. p-Sensitivity: A semantic privacy-protection model for location-based services //Proc of the 9th Int Conf on Mobile Data Management Workshops. Piscataway, NJ: IEEE, 2008: 47-54

[5] Chow C, Mokbel M F, Tian H. TinyCasper: A privacy-preserving aggregate location monitoring system in wireless sensor networks //Proc of the 28th ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2008: 1307-1310

[6] Du Jing, Xu Jianliang, Tang Xueyan, et al. iPDA: Supporting privacy-preserving location-based mobile services //Proc of the 8th Int Conf on Mobile Data Management. Piscataway, NJ: IEEE, 2007: 212-214

[7] Mokbel M F, Chow C, et al. The new casper: A privacy-aware location-based database server //Proc of the 23rd Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2007: 1499-1500

黄 毅 男,1987 年生,硕士研究生,主要研究方向为数据隐私保护、移动数据管理等.

潘 晓 女,1981 年生,博士,主要研究方向为移动数据管理等.

孟小峰 男,1964 年生,教授、博士生导师,主要研究方向为 Web 数据管理、移动数据管理、XML 数据管理等.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

科研成果

一、学术专著

学术专著名称：《Moving Objects management: Models, Techniques and Applications》

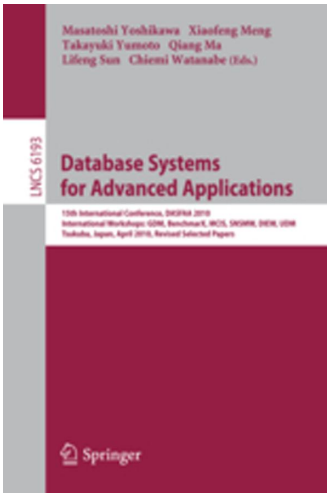
学术专著作者：孟小峰 陈继东

该书是孟小峰教授及其团队历时十年的研究成果，全书比较系统地介绍移动对象管理相关内容，包括移动对象管理模型（包括移动对象建模、移动对象更新、移动对象索引等内容），移动对象管理技术（包括移动对象查询、移动对象预测、移动数据不确定性研究等内容），和移动对象管理应用（包括动态交通导航、动态交通网络、移动对象聚类分析、位置隐私保护等内容）等。

丹麦奥尔堡大学教授 Christian S. Jensen 为本书专门作序，给本书以高度的评价，认为“The book meets the need for a coherent account of the state-of-the-art on important topics in the area of moving-object data management, which is at the core of the evolving mobile Internet. It comes highly recommended to research students and researchers new to the topics covered, as well as to experienced researchers”（该书全面系统地阐述了移动互联网的核心技术--移动对象数据管理的重要研究议题。值得高度推荐给研究生及相关研究人员）。Christian S. Jensen 教授目前担任国际数据库顶级期刊 VLDBJ 主编、国际数据库组织 SIGMOD 副主席，是移动数据管理领域著名专家。

二、论文集、专刊

论文集名称：**Database Systems for Advanced Applications**



论文集编辑者：Masatoshi Yoshikawa, Xiaofeng Meng 等

The 15th International Conference on Database Systems for Advanced Applications（DASFAA2010）于 2010 年 4 月 1 日到 4 日在日本举行，会议论文集由 Springer 出版，孟小峰教授为本次会议的程序委员会成员，也是 DASFAA 2010 International Workshops: GDM, BenchmarX, MCIS, SNSMW, DIEW, UDM 论文集的编委之一。

论文集名称：**Proceedings of the Second International Workshop on Cloud Data Management**

论文集编辑者：Xiaofeng Meng, Ying Chen, Jiaheng Lu, Jianliang Xu

The Second International Workshop on Cloud Data Management (CloudDB 2010) 于 2010 年 10 月 30 日在加拿大多伦多举行，孟小峰教授担任本次研讨会的联合主席，并负责本次研讨会论文集的编辑工作。

专刊名称：**Special Section on Trends Changing Data Management**

专刊客座编辑：孟小峰教授，王海勋博士

计算机科学技术学报（JCST）Special Section on Trends Changing Data Management 专刊（2010 Vol.25 No.3）于 2010 年 5 月出版发行，孟小峰教授、王海勋博士受邀担任该专刊客座编辑。

三、论文列表

普适数据管理（Pervasive Data Management）

- *C. Zhou, X. Meng: Out-of-Order Durable Event Processing in Integrated Wireless Networks. Accepted for publication in Journal of Pervasive and Mobile Computing (PMCJ). (2010).
- C. Zhou, X. Meng: IO3: Interval-based Out-of-Order Event Processing in Pervasive Computing. In Proceedings of the 15th International Conference on Database Systems for Advanced Applications (DASFAA 2010): 261-268, April 1-4, 2010, Japan.
- 周春姐，孟小峰，文洁：Flickr 中的复合事件检测. 计算机研究与发展，卷 47（增刊）：1-7，2010.10. (第二十七届中国数据库学术会议，北京)
- 潘晓，郝兴，孟小峰：基于位置服务中的连续查询隐私保护研究. 计算机研究与发展，卷 47(1)：121-129，2010.1.
- 周春姐，孟小峰：普适计算中复合事件检测的研究与挑战. 计算机科学与探索，卷 4(12)：1057-1072，2010.12.

云数据管理（Cloud Data Management）

- *Y. Shi, X. Meng, J. Zhao, X. Hu, B. Liu, H. Wang: Benchmarking Cloud-based Data Management Systems. In proceedings of the CIKM Workshop on Cloud Data Management(CloudDB2010): 47-54, October 30, 2010, Toronto, Canada.
- J. Zhao, X. Hu, X. Meng: ESQP: An Efficient SQL Query Processing for Cloud Data Management. In proceedings of the CIKM Workshop on Cloud Data Management (CloudDB2010): 1-8, October 30, 2010, Toronto, Canada.
- X. Meng, Y. Chen, J. Lu, J. Xu: Report on the Second International Workshop on Cloud Data Management (CloudDB 2010). In Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM2010): 1969-1970, October 30, 2010, Toronto, Canada.
- X. Meng, J. Lu, J. Qiu, Y. Chen, H. Wang: Report on the First International Workshop on Cloud Data Management (CloudDB2009).SIGMOD Record,Vol.39(1):58-60,March 2010.

闪存数据库系统（**Flash-based Database Systems**）

- *X. Tang, X. Meng: ACR: an Adaptive Cost-Aware Buffer Replacement Algorithm for Flash Storage Devices. In Proceedings of the 11th International Conference on Mobile Data Management (MDM 2010): 33-42，May 23-26, 2010, Kansas City, Missouri, USA.
- D. Zhou, X. Meng: A Flash-Aware Random Write Optimized Database. In Proceedings of the 11th International Conference on Mobile Data Management (MDM 2010): 276-278,May 23-26, 2010, Kansas City, Missouri, USA.
- 汤显，孟小峰：FClock：一种面向 SSD 的自适应缓冲区管理算法. 计算机学报，卷 33(8)：1460-1471，2010.8. (第二十七届中国数据库学术会议，北京)
- 卢泽萍，孟小峰，周大：HV-recovery：一种闪存数据库的高效恢复方法. 计算机学报.(第二十七届中国数据库学术会议，北京) (NDBC2010“**萨师煊优秀论文**”)
- 梁智超，周大，孟小峰：Sub-Join：面向闪存数据库的查询优化算法. 计算机科学与探索，卷 4(5)：401-409，2010.5.
- 周大，梁智超，孟小峰：HF-Tree：一种闪存数据库的高更新性能索引结构. 计算机研究与发展，卷 47(5)：832-840， 2010.5.

Web 数据管理（**Web Data Management**）

- W. Liu, X. Meng, W. Meng: ViDE: A Vision-Based Approach for Deep Web Data Extraction. IEEE Transactions on Knowledge and Data Engineering(TKDE). Vol.22(3): 447-460 (2010).
- W. Liu, X. Meng, J. Yang, J. Xiao: Duplicate Identification in Deep Web Data Integration. In proceedings of the 11th International Conference on Web-Age Information Management (WAIM2010): 5-17, July 15-17, 2010, Jiuzhaigou, China.
- Y. Kou, Y. Li, X. Meng: DSI: A Method for Indexing Large Graphs Using Distance Set. In proceedings of the 11th International Conference on Web-Age Information Management (WAIM2010):297-308, July 15-17, 2010, Jiuzhaigou, China.
- W. Liu, X. Meng: A Holistic Solution for Duplicate Entity Identification in Deep Web Data Integration. In proceedings of the 6th International Conference on Semantics, Knowledge & Grids(SKG2010): 267-274, Nov. 1-3, 2010, Ningbo, China.

- Y. Li, D. Elsweiler, X. Meng: Towards Task-Organised Desktop Collections. In Proceedings of the ACM SIGIR Workshop on Desktop Search: Understanding, Supporting, and Evaluating Personal Data Search (DS2010): 21-24 , July 23, 2010, Geneva, Switzerland.
- 黄静，陆嘉恒，孟小峰：高效的 XML 关键字查询改写和结果生成技术. 计算机研究与发展，卷 47(5)：841-848，2010.5.

系统演示（**Demo**）

- Y. Li, X. Zhang, X. Meng: Exploring desktop resources based on user activity analysis. In Proceedings of the 33rd Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval (SIGIR2010): 700 , July 19-23, 2010, Geneva, Switzerland.
- 胡享梅，赵婧，孟小峰，王仲远，史英杰，刘兵兵，王海平：TaijiDB：一个双核云数据库管理系统. 计算机研究与发展，卷 47（增刊）：433-437，2010.10.(第二十七届中国数据库学术会议，北京) (NDBC2010 **最佳系统演示**)
- 黄毅，潘晓，孟小峰：OrientPrivacy：移动环境下的隐私保护服务器. 计算机研究与发展，卷 47（增刊）：438-441，2010.10.(第二十七届中国数据库学术会议，北京)

注：标’*’为年度代表论文。

毕业生学位论文

1. 李玉坤，数据空间模型与查询技术研究（Research on Data Space Model and Query Processing）,中国人民大学，博士生毕业论文，2010.5.6
2. 潘晓，位置隐私保护技术研究（Location Privacy Preserving）,中国人民大学，博士生毕业论文，2010.5.6
3. 周大，闪存数据库系统存储和索引技术研究（Storage and Indexing on Flash-based Database Systems）,中国人民大学，博士生毕业论文,2010.5.6
4. 王仲远，面向领域的 Web 数据集成技术研究（Research on Domain-Oriented Web Data Integration），中国人民大学，硕士生毕业论文，2010.5.21
5. 寇玉波，数据空间中图搜索技术的研究（Research on Graph Searching Techniques in DataSpace)，中国人民大学，硕士生毕业论文，2010.5.21
6. 张相於，数据空间索引关键技术研究（Research on Dataspace Indexing Techniques), 中国人民大学，硕士生毕业论文，2010.5.21
7. 郝兴，连续密度查询处理关键技术研究（Research on Key Techniques of Continuous Density Queries),中国人民大学，硕士生毕业论文，2010.5.21
8. 徐俊劲，基于同义词规则的字符串近似搜索技术研究（Research on Efficient String Similarity Search Using Synonyms),中国人民大学，硕士生毕业论文，2010.5.21
9. 艾静，Web 信息可信性及用户隐私保护问题研究（Research on Web Information Credibility and Privacy Preserving on the Searchable Internet）,中国人民大学，硕士生毕业论文，2010.5.21

四、专利

已授权专利：

1. 基于视觉的 **Web** 数据抽取系统和方法

发明名称：基于视觉的 Web 数据抽取系统和方法

申请人：孟小峰

专利号：ZL200810056103.4

获批时间：2010-2-17

2. 一种智能 **Web** 查询接口系统及其方法

发明名称：一种智能 Web 查询接口系统及其方法

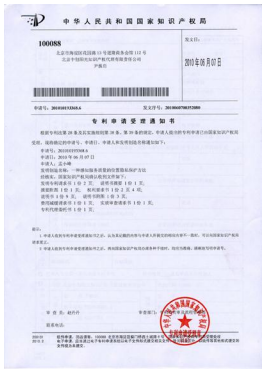
申请人：孟小峰

专利号：ZL200810056104.9

获批时间：2010-08-13

已申请专利

1. 一种感知服务质量的位置隐私保护方法



发明名称：一种感知服务质量的位置隐私保护方法

申请人：孟小峰

申请号：201010193368.6

申请时间：2010-06-07

2. 一种防止位置依赖攻击的位置隐私保护方法

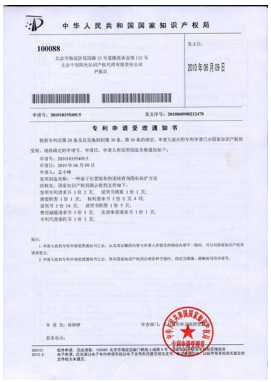
发明名称：一种防止位置依赖攻击的位置隐私保护方法

申请人：孟小峰

申请号：201010193366.7

申请时间：2010-06-07

3. 一种基于位置服务的连续查询隐私保护方法



发明名称：一种基于位置服务的连续查询隐私保护方法

申请人：孟小峰

申请号：201010195409.5

申请时间：2010-06-09

4. 一种基于闪存的自适应缓冲区置换方法

发明名称：一种基于闪存的自适应缓冲区置换方法

申请人：孟小峰

申请号：201010566968.2

申请时间：2010-11-25

5. 一种基于闪存的数据库恢复方法

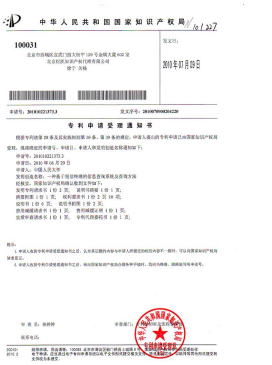
发明名称：一种基于闪存的数据库恢复方法

申请人：孟小峰

申请号：201010552789.3

申请时间：2010-11-19

6. 一种基于短信终端的信息查询系统及查询方法



发明名称：一种基于短信终端的信息查询系统及查询方法

申请人：孟小峰

申请号：201010221373.3

申请时间：2010-6-29

五、科研项目

课题来源：国家自然科学基金面上项目

课题名称：Web 信息可信性研究

课题负责人：孟小峰

课题起止年限：2011 年 1 月 至 2013 年 12 月

课题简介：

随着网络与通信技术的迅速发展，Web 上的信息越来越多，并且仍以惊人的速度快速增长，Web 已经成为人们获取信息的重要途径之一。然而，这些信息纷繁复杂、鱼目混珠，如何鉴别这些信息的可信性已经成为一个日益重要的研究问题。本课题从基于内容分析的可信度评估机制、基于评分和投票的可信度评估机制、基于传播的可信度评估机制、基于数据源与记录关联分析的可信性评价机制四个方面深入分析了国内外信息可信度研究的现状。在此基础上，本课题拟从浅层网络和深层网络的角度出发，针对浅层网络和深层网络的特点和现有技术的缺陷，系统地研究 Web 上信息可信度的基本理论和实现方法，重点研究浅层网络中基于网页间关联的可信性研究、深层网络中基于数据源、数据记录以及双层关联机制的可信度研究四个方面的问题。通过本课题的研究为 Web 上信息可信度的进一步研究与应用提供理论方法、技术支撑和新的思路。

课题来源：国家“核高基”项目“非结构化数据管理”

课题名称：纯 XML 数据库系统

课题负责人：孟小峰

课题起止年限：2010 年 1 月 至 2011 年 12 月

课题简介：

主流非结构化数据类型包括音频、图像、视频、图形、文本等。这些这些数据都具有一定的语义特征或结构特征，而这些特征的用 XML 来描述十分合适。非结构化数据管理系统方面，需要支持包括各种文档类型在内的媒体内容的全文检索和特征表达与查询。基于 XML 数据库实现内容的特征表达与查询，是一种可行的选择。此外，进过国内外研究人员的不断努力，到目前为止纯 XML 数据管理技术已相对成熟，利用 XML 数据库实现这些特征数据的管理成为可能。因此研究纯 XML 数据库系统分课题非常必要，它是顺利完成课题的基础，是课题不可缺少的一部分。分课题的课题目标包括以下 3 个方面：1）. 研发一套具有自主知识产权的软件，支持 XML 数据的多粒度存储，支持 XML 标准查询语言 XQuery 和 Xpath，提供 Java API。2）部署三项典型示范应用，将纯 XML 数据管理系统集成到非结构数据管理系统中，并成功应用于新闻媒体，数字图书馆，流程工业三个行业示范应用 3）实现一套完整的基准测试平台，研究和建立一套完整的基准测试指标体系和测试平台。

课题来源： **IBM** 开放协作研究项目（**Open Collaborative Research OCR**）

课题名称：Cloud based Large Scale Data Management

课题负责人：孟小峰

课题起止年限： 2010 年 至 2012 年

课题简介：

WAMDM 实验室团队在 2009 年 9 月，通过 IBM SUR 资助，开始承接了关于云数据管理系统的开发工作。这次由于团队的优秀表现，IBM 公司又提供了 IBM 开放协作研究（Open Collaborative Research OCR）项目资助，资助课题为ñCloud based Large Scale Data Managementò。两项资助联合起来，共同搭建一个开源的云数据管理平台。

IBM 开放协作研究(Open Collaborative Research 简称 OCR)项目，是由 IBM 设立的一项科学研究资助计划，该计划旨在加强与国际一流高校科研人员的技术合作与交流，促进科研成果的快速转化。这次申请中，IBM 中国研究院向美国总部提交了十份申请。通过多轮讨论，答辩，表决，最后仅人民大学这个项目获得资助。这次我院的 OCR 项目在全球范围内申请成功，确实难度较大，成功来之不易。

相信通过这个项目的支持和团队成员的共同努力，必将使中国人民大学信息学院在国际云数据管理研究上占据一席之地，并带动国内相关研究工作的开展。

课题来源：**MSRA Faculty Award**

课题名称：Cloud-based Database System

课题负责人：孟小峰

课题起止年限：2010 年 至 2011 年

课题简介：

如何有效的管理大规模的数据在很多领域，例如：医疗保健，移动通信等，已经成为了一个很具有挑战性的问题。逐渐增大的数据量（甚至达到 PB 级别以上）对数据存储的体系结构，大规模的并行化查询处理，和数据分析处理提出了严峻的挑战。同时，大规模数据中心和计算机集群的兴起也产生了一个新的商业模式：云计算，基于云计算，公司和个人可以租借存储和计算能力，而不用花费大量的资本投资来构建和准备大规模的计算机安装。因此，基于云的数据存储和管理是一个快速发展的商业模式。在这个课题，我们设计基于云的数据库系统，用来支持下一代的信息管理和大规模分析处理的数据管理解决方法。课题的目标在于研究新的能够处理下一代的管理大规模数据的应用程序并且能够应用于多个领域（医疗保健，移动通信等）的数据库系统。

课题来源： 国家自然科学基金 青年项目

课题名称： 混沌人工神经网络的特性研究与混沌控制及其在联想记忆中的应用

课题负责人：杨刚

课题起止年限：2011 年 1 月 至 2013 年 12 月

课题简介：

混沌现象普遍存在于人脑活动中，混沌动态与人工神经网络的结合为实现真实世界智能计算提供了新契机，混沌神经网络在组合优化、联想记忆和人工智能等领域具有广阔的应用前景，各领域对混沌神经网络的适用性和混沌控制的灵活性提出了更高的要求。为了探索混沌神经网络动态特性，提高混沌神经网络性能，建立满足大规模联想记忆应用需求的自适应混沌神经网络模型，本课题拟基于数学推理和大量实验，利用数据统计分析的方法，形成混沌神经网络的特性分析结果，构造混沌神经网络的特性分布与参数调节模型，用于指导混沌神经网络的性能提高和混沌动态控制；构建灵活的自适应混沌动态控制策略，实现混沌神经网络中混沌动态的多运行轨迹变换，并将得到的特性模型和控制策略应用于联想记忆，提出具有自适应多对多联想记忆功能的混沌神经网络算法；同时研究增加网络存储容量和应用效率的优化算法，使得本课题的算法和模型适用于实际的应用系统。

学术交流

一、学术活动任职

Prof. Xiaofeng Meng:

Program Committee member, The 25th ACM Symposium on Applied Computing Mobile Database Systems Track(**SAC2010 MDS**)**,** March 22-26, 2010, Sierre, Switzerland

Workshop Co-Chair, The 15th International Conference on Database Systems for Advanced Applications(**DASFAA2010**), April 1-5, 2010, Tsukuba, Japan

Program Committee member, The 15th International Conference on Database Systems for Advanced Applications(**DASFAA2010**), April 1-5, 2010, Tsukuba, Japan

Program Committee member, The 11th International Conference on Mobile Data Management (**MDM 2010**), May 23-26, 2010, Kansas City, Missouri, USA

CCF DB Society Liaison， The 11th International Conference on Web-Age Information Management (**WAIM 2010**), July 15-17, 2010, Jiuzhaigou, China

Program Committee member, 21st International Conference on Database and Expert Systems Applications(**DEXA2010**), August 30-September3, 2010, Bilbao, Spain

Demo Co-Chair, The 36th International Conference on Very Large Databases(**VLDB2010**), September 13-17, 2010, Singapore

General Chair, The 27th National Database Conference of China (**NDBC2010**), October, 13-16, 2010, China.

Workshop Chair, The 2rd International Workshop on Cloud Data Management (**CloudDB2010**), October 26-30, 2010,Toronto, Canada

Guest Editor, Special Section on Trends Changing Data Management, was published by Journal of Computer Science and Technology (**JCST**)

二、学术交流

2010.4.1-2010.4.4

孟小峰教授与周春姐博士参加 DASFAA2010

2010 年 4 月 1 日至 2010 年 4 月 4 日，孟小峰教授与周春姐博士参加在日本筑波举办的第 15 届数据库系统与高级应用国际会议(DASFAA2010)。孟小峰教授担任此次会议专题讨论委员会的联合主席。周春姐博士作了题为“IO^3: Interval-based Out-of-Order Event Processing in Pervasive Computing”的报告。

2010.5.22

孟小峰教授应邀参加第二届中国云计算大会云计算核心技术架构分论坛

2010 年 5 月 22 日，孟小峰教授应邀在北京举行的第二届中国云计算大会云计算核心技术架构分论坛上作“云数据管理技术”主题报告。在次报告中，孟小峰教授介绍了中国人民大学网络与移动数据管理实验室(WAMDM)在云数据管理研究上所开展的一些工作，其中实验室对多个开源系统(如 Hadoop、Cassandra、HBase、Hive 等)的分析比较和所做的基准测试报告引起与会者的极大兴趣，许多专家学者对于孟小峰教授团队在云数据管理研究上所开展的扎扎实实的工作表示赞许，也有众多企业纷纷表示了合作意向。

2010.6.4

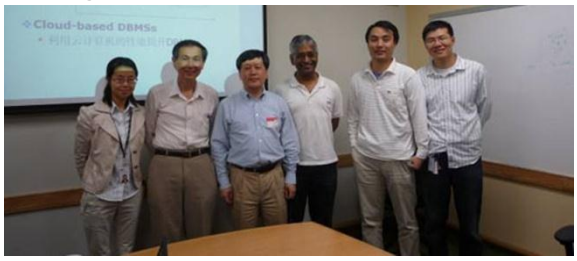
孟小峰教授应邀参加 2010 教育部-IBM 高校合作项目年会暨十五周年庆

2010 年 6 月 4 日，2010 教育部-IBM 高校合作项目年会暨十五周年庆在上海同济大学召开，来自教育部、国家留学基金委、全国 62 所 IBM 合作伙伴高校的领导老师以及 IBM 公司高层领导、资深专家和媒体届的朋友们约 240 人参加，孟小峰教授作为特邀演讲嘉宾参加了此次会议，并就“云计算及其应用”发表演讲。

2010.7.5-2010.7.13

孟小峰教授应邀参加微软教育峰会**(FacultySummit2010)**

2010 年 7 月 5 日至 2010 年 7 月 13 日，微软教育峰会 FacultySummit2010 在美国西雅图微软总部召开，来自全球高校的约 350 多位教师应邀参加此次大会，国内代表团由 16 人组成，信息学院孟小峰教授应邀请参加了此次盛会。峰会期间孟小峰教授访问了华为在美新设立的研究中心、IBM Almaden 研究中心、Facebook、Intel 及 Google 等企业，并联络了人大的校友。



2010.7.19-2010.7.23

孟小峰教授和李玉坤博士参加 **SIGIR2010**

2010 年 7 月 19 日至 2010 年 7 月 23 日，孟小峰教授与李玉坤博士参加在瑞士日内瓦城举办的第 33 届 ACM SIGIR 国际会议(SIGIR 2010)。李玉坤博士的一篇 Demo ñExploring desktop resources based on user activity analysisò 在此次会议上进行了演示，并在 DS2010 研讨会上做题为ñTowards Task-Organized Desktop Collectionsò的报告。

2010.9.13-2010.9.17

孟小峰教授参加 **VLDB2010**

2010 年 9 月 13 日至 2010 年 9 月 17 日，孟小峰教授参加在新加坡国举办的第 36 届 VLDB 2010 国际会议。孟小峰教授担任此次会议系统演示专题的联合主席。

2010.10.30

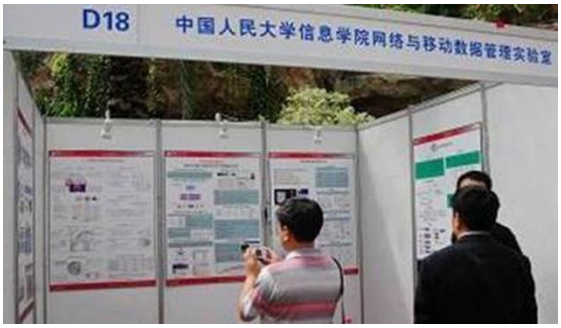
孟小峰教授和史英杰博士参加 **CloudDB2010**

2010 年 10 月 30 日，孟小峰教授与史英杰博士参加在加拿大多伦多举办的第二届云数据管理国际研讨会，孟小峰教授担任本次研讨会的联合主席。这次会议是由 WAMDM 实验室承办。史英杰博士作了题为ñESQP: An Efficient SQL Query Processing for Cloud Data Managementò 、ñBenchmarking Cloud-based Data Management Systemsò的报告。

2010.10.11-2010.10.12

孟小峰教授应邀参加“网联世界计算无限”为主题的 **2010** 中国计算机大会

2010 年 10 月 11 日至 2010 年 10 月 12 日，网联世界计算无限”为主题的 2010 中国计算机大会在杭州第一世界大酒店举行，孟小峰教授应邀在云计算专题论坛上做了题为“面向云计算的数据管理”的主题报告，从数据管理研究的角度讨论了云计算模式及云数据管理在研究界的发展和演变，提出了云数据管理新型应用带来的研究课题，并简单介绍了 WAMDM 实验室在云数据管理方面的研究工作。孟小峰教授的主题报告得到了参会人员的广泛关注，得到与会者一致好评。



在高校科研成果展中，集中展示“纯 XML 数据库系统 OrientX（王选奖成果）”、“云数据库系统 TaijiDB”、“Web 数据集成系统 ScholarSpace（国家自然基金特优结题成果）”、“闪存数据系统 FlashDB（国家自然基金重点项目成果）”、“移动环境位置隐私保护系统（863 计划信息领域重点项目成果）”等科研成果。

2010.11.12

孟小峰教授应邀参加“中创软件基金人才奖”颁奖仪式并做学术报告

2010 年 11 月 12 日，主题为“举中华英才，创软件伟业”的第十五届“中创软件基金颁奖仪式暨历届获奖者代表学术报告会”在南京举行。孟小峰教授应邀做数据库方面的研究进展报告。孟小峰教授在报告中指出中国应该构建符合其国力的云基础架构，探讨新的应用模式，积累数据财富，建立“数据思维”的方法。孟小峰教授同时指出，在云计算的推动下，“NoSQL 运动”的出现对我国数据库的研究带来新的机遇。

2010.11.18

孟小峰教授应邀参加 **2010** 年“**Google** 教育高峰会”

2010 年 11 月 18 日，2010 年“Google 教育高峰会”在上海举行，来自全国高校的 60 多位代表应邀参加此次会议。孟小峰教授应邀代表人民大学参加此次峰会。本次高峰会旨在探讨在“电子商务”和“移动计算”两大领域的研究课题。此次孟小峰教授在参会期间与 Google 负责高校合作和研究的部门进行了进一步的沟通，拟在移动计算研究方面开在合作，并适时引入 Google 的教育资源开设相关课程，提供本科生到 Google 实习的机会等。

2010.11.24

孟小峰教授应邀参加第 **50** 期**“**双清论坛**”**

2010 年 11 月 24 日，国家自然科学基金委员会第 50 期“双清论坛”在天津大学举行。孟小峰教授应邀在会上做了“Big Data and cloud computing”的报告。 孟小峰教授在报告中指出，现代科技发展的事实告诉我们，信息技术的发展一直以来影响着人类社会的方方面面。特别是近年来，互联网的爆炸式发展产生了大量数据，这些数据对政治、经济等具有非常重要的意义。面对如此巨大的数据（Big Data），需要考虑基于数据思维的方式去解决和思考问题。所谓数据思维，就是要求我们用数据说话，基于数据去发现问题，并基于数据去解决问题。

三、学术报告

2010.5.19

纽约州立大学宾汉姆顿分校孟卫一教授来信息学院做学术报告

2010 年 5 月 19 日，孟卫一教授在信息学院办公楼四层报告厅作了题为“OSA: Opinion Shift Analysis”的学术报告。孟小峰教授主持了报告会。互联网时代，公众会对一些论坛和社区的某些话题发表评论意见，而这些意见在一个时间段内会引发显著变化。孟卫一教授第一次提出了 Opinion Shift Analysis（OSA）这个研究问题。在此次报告中，孟卫一教授对 OSA 做了深入的分析，采用对特定话题建立概率模型和基于语法的意见分析方法，以实现对一个时间段内特定话题的意见显著变化以及引起其变化的原因的自动检测。

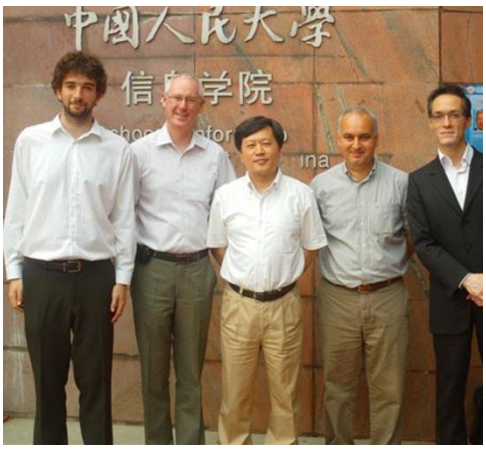
2010.6.29

美国伊利诺大学芝加哥分校 **Clement YU** 教授来信息学院做学术报告

2010 年 6 月 29 日，美国伊利诺大学芝加哥分校 Clement YU 教授来我院作题为“On the Construction of a Sentimental Word Dictionary”的学术报告，孟小峰教授主持了报告会。Clement YU 教授首先介绍了具有情感色彩词语的词典的构造过程。然后讲述了在构造过程中引入推理准则，该准则以已有的极性词语作为输入，产生极性同义词集合。推理的一个重要的结果就是词典内部和词典之间产生了不一致性。并指出产生所有词的极性的过程以及检测一致性的问题都是 NP-完全问题。

2010.6.29

法国驻华大使馆科技专员 **Patrick NEDELLEC** 等人访问 **WAMDM** 实验室



2010 年 6 月 29 日，法国驻华大使馆科技专员 Patrick NEDELLEC 等人访问 WAMDM 实验室，孟小峰教授对法国学者的来访表示欢迎，介绍了 WAMDM 实验室近十年的研究课题以及正在研究的课题。法国电信 ParisTech 大学的 Talel Abdessalem 博士简要介绍了其在 Web 数据抽取、XML 数据管理、和移动数据管理方面的研究成果，来自 ITAAPY 公司的 Luis Belmar-Letelier 博士介绍了基于版本的文档管理系统 LPOD 项目情况。来访学者对 WAMDM 实验室的云数据管理研究成果给予很好的评价。

2010.9.8

澳大利亚新南威尔士大学王伟博士来 **WAMDM** 实验室做学术报告

2010 年 9 月 8 日，应孟小峰教授邀请，澳大利亚新南威尔士大学的王伟博士来实验室做学术报告。王伟博士做了题为“Similarity Join Algorithms: An Introduction”的报告。在这次报告中，王伟博士首先分析和讨论了相似度连接查询在数据管理中的应用场景及其存在的挑战，然后介绍了现有的一些解决方法。其中重点介绍了基于集合的相似度连接查询和基于字符串的相似度连接查询。王伟博士和 WAMDM 实验室的同学就最新的研究问题进行了深入的交流和探讨，为大家的研究提供一些非常好的建议。

2010.10.16

IBM Almaden 研究中心 **Hui-I Hsiao** 博士和何斌博士来 **WAMDM** 实验室访问

2010 年 10 月 16 日，应孟小峰教授邀请，IBM Almaden 研究中心的 Hui-I Hsiao 博士和何斌博士到实验室做学术报告。何斌博士做了题为“Mykonos: Software Exploitation of SCM”的报告。在这次报告中，主要是分析了 SCM 和 PCM 的特性，以及 PCM 能在现有系统架构中扮演什么样的角色。针对目前 Key-Value 存储对传统事务的支持较弱，结合新硬件 PCM 的特性，提出一种基于 PCM 的 Key-Value 存储系统，其中重点讲述了该系统的设计原则、系统架构和事务处理等问题，最后举例说明了系统执行的过程。

2010.11.26

百度基础架构部侯振宇等五人来 **WAMDM** 实验室访问

2010 年 11 月 26 日，应孟小峰教授邀请，百度基础架构部侯振宇等五人来实验室访问，并进行了学术交流。本次交流活动主要针对目前 WAMDM 实验室与百度合作的闪存数据库课题。百度技术人员首先就搜索引擎中的海量数据存储问题做了详细讲解，介绍了百灵分布式处理平台，包括百灵分布式平台下的存储模型，随机查询/批量查询处理、时效性模型等。最后介绍了支持线上应用的 Key-Value 存储系统以及相关的问题。双方就相关的技术问题进行了深入交流与探讨。

2010.11.29

Google 中国研究院副院长张智威博士来 WAMDM 实验室做学术报告

2010 年 11 月 29 日，应孟小峰教授邀请，Google 中国研究院副院长张智威博士来实验室做学术报告。张智威博士做了题为“Confucius & ‘Its’ Intelligent Disciples”的报告。在这次报告中，张智威博士首先介绍了 Google Q&A 系统，结合 Facebook 中的搜索引擎缺陷，就搜索与社会网络的关系给出了清晰而生动的讲解，并从技术层面介绍了二者的区别与联系。随后，张智威博士介绍了谷歌 Mobile 2014 计划中的一些研究热点，如三网合一中的数据管理、Indoor and 3D 导航等问题。

2010.12.2

香港浸会大学胡海波博士和许建良博士来 WAMDM 实验室访问

2010 年 12 月 2 日，应孟小峰教授邀请，香港浸会大学的胡海波博士和许建良博士到实验室做学术报告。胡海波博士做了题为“Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism”的学术报告。胡海波博士介绍了 SMC-based framework 和 PH-based framework 两框架下针对用户、数据拥有者及云端三方的数据隐私保护技术。另外，许建良博士和实验室同学就数据隐私保护、移动数据管理、闪存数据管理等问题进行了深入的交流和探讨，为大家的研究提供了非常好的建议。

2010.12.10

丹麦奥尔胡斯大学 Christian S. Jensen 教授来 WAMDM 实验室访问

2010 年 12 月 10 日下午，Christian S. Jensen 教授访问了孟小峰教授所领导的 WAMDM 实验室。孟小峰教授首先介绍了国内数据库的发展及取得的研究成果，然后从 Database、Web 及 Mobile 三个角度介绍了 WAMDM 实验室近十年取得的研究成果，及未来十年的规划；陆嘉恒博士介绍了面向移动用户的 Web 数据集成问题，寻求双方开展进一步合作的机会；周春姐博士、霍峥博士分别介绍了基于 Flick 的序列挖掘、位置隐私保护等研究工作。Christian S. Jensen 教授对孟小峰教授所领导团队的研究工作给予高度评价，针对博士生目前研究存在的问题给予指导，希望通过国际合作促进双方的研究。

四、举办会议

第 **27** 届中国数据库学术会议

2010 年 10 月 14 日上午，第 27 届中国数据库学术会议，在中国人民大学逸夫会议中心召开。此次会议中国计算机学会数据库专委会主办，中国人民大学、北京大学、清华大学共同承办。这是中国数据库界的一次盛会，有来自海外及全国各地的代表 400 余人参加本次大会，来自 80 多所大专院校研究机构。信息学院王珊教授为本次大会指导专家，孟小峰教授为本次大会主席。

纪宝成校长代表学校到会致辞，向与会者介绍了“人民满意，世界一流”发展思路。纪校长特别强调由我校萨师煊教授等老一辈学者一手开创的中国数据库研究事业至今已有三十多年的历史，期间经历了几代人的不懈努力，取得了有目共睹的成绩，在此十分感谢数据库届同仁长期对我校相关学科的支持。本次会议提供了向国内数据库同行学者展示人民大学及信息学院的极好的机会，也是信息学院为庆祝中国人民大学命名组建六十周年而举办的重要活动。纪校长会见了老专家代表、大会特邀讲者等。

大会开幕式由大会主席、孟小峰教授主持。孟小峰教授代表会议组织方介绍了会议的基本情况。本届会议主要关注数据库技术所面临新的挑战问题和研究方向。会议共收论文 320 篇，经过双匿名网络评审，最终录用论文 124 篇，录用率为 38.8%。本次会议汇集 4 个大会报告、5 个数据库新技术报告、4 个云数据库管理报告、10 场分组报告、25 个系统演示、3 场辅导报告、以及研究生论文指导研讨会等。提供赞助的企业包括 EMC China Lab，HP China Lab，Google China，IBM，人大金仓，以及清华出版社，华章出版社，高教出版社等。

在 NDBC2010 举办之际，中国计算机学会数据库专委会特别邀请见证了我国数据库研究历史发展的老专家到会，召开“中国数据库发展历史回顾暨萨师煊教授追思会”。来自全国各地二十多位 70 岁以上的老专家满怀深情地回顾了三十多年我国数据库事业发展历程，深切缅怀了萨师煊教授对数据库学科所作出的开创性贡献，特别是他包容的胸怀和提携后辈人才成长的为人品格为大家所推崇。

会议于 10 月 15 日下午闭幕，闭幕式上中国计算机学会数据库专委会为老专家们了颁发了“中国数据库发展贡献奖”和中国计算机学会数据库专委会荣誉委员证书。大会对人民大学信息学院出色的组织工作表示感谢。此次人大志愿者给与会者留下很好的印象，充分展示了人大学子的风采，得到大家的一致好评。全体参会人员为他们献上了数分钟的掌声以示感谢。

大会开幕式前，纪宝成校长会见了老专家代表、大会特邀讲者等。中国人民大学常务副校长袁卫教授专门出席了本次大会晚宴，代表学校致欢迎词并向大家敬酒。晚宴上袁校长与参会的人大老师、同学、及校友合影留念。

2010.5.7

信息学院召开“云计算发展与展望”专题研讨会

2010 年 5 月 7 日上午，由信息学院副院长孟小峰教授主持的“云计算发展与展望”研讨会在理工楼配楼 1 层会议室召开。在次研讨会上孟小峰教授做了题为“云计算发展与展望”的报告，并强调信息学院要建立以云计算为核心，包括云存储、云数据管理、云安全、云电子商务和云知识管理等内容在内的云计算研究团队。

2010.5.31

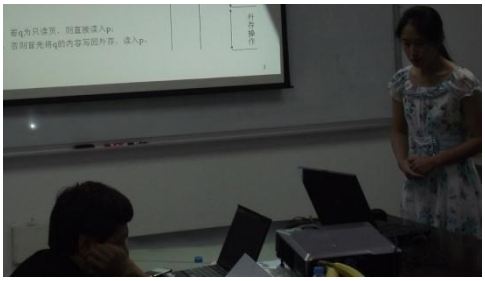
信息学院召开“智能交通系统管理”专题学术研讨会

2010 年 5 月 31 日上午，“智能交通系统管理”专题学术研讨会在理工楼配楼 2 层会议室顺利召开。此次研讨会由信息学院副院长孟小峰教授主持。此次研讨会请到了中科院软件所的丁治明研究员和北京市交通发展研究中心智能交通部的邓小勇部长。孟小峰教授为大家做题为“智能交通系统管理前瞻”的报告。孟老师首先为大家简单介绍了智能交通系统管理研究的重要意义和发展方向。随后，孟老师以前一阶段我院开展的“云计算研讨会”为例，说明了整合零散科研力量，统一科研方向的必要性。

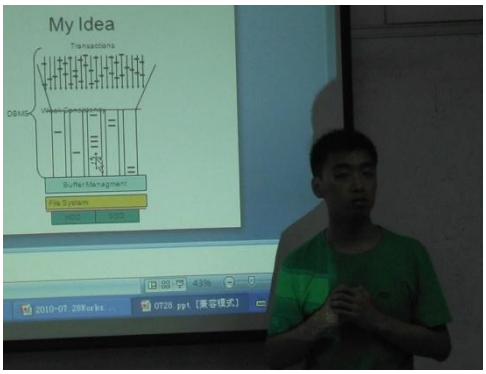
2010.7.28

第四届闪存数据库系统研讨会

2010 年 7 月 28 日，第四届闪存数据库系统研讨会于在北京中国人民大学举行，这是在以中国人民大学孟小峰教授为负责人的国家自然科学基金重点项目“闪存数据库技术研究”的支持下创立的学术交流平台，也是课题组探索的一种新的课题组织方式。与会人员有来自于中国人民大学的孟小峰教授、中国科技大学的岳丽华教授、金培权副教授、以及香港浸会大学三所高校相关的硕士博士研究生，同时还邀请到了百度刘斌等高级工程师和北京大学崔斌教授。



会议包含以下几个报告：操作系统和数据库缓冲区管理算法、数据库外排序算法、闪存存储板和闪存芯片测试、数据库事务处理和 TPCC 测试结果。这些报告展示了最新的研究进展和技术成果，为基于闪存存储器的数据库的进一步研究与应用奠定基础，为基于闪存存储器的数据库理论和技术的进一步发展提供新思路。百度致力解决闪存在服务器上的使用寿命，以提高闪存在现实应用中的性价比，这对项目以后的研究会有很大的推动作用。



WAMDM 实验室的汤显博士作了“ACR: an Adaptive Cost-Aware Buffer Replacement Algorithm for Flash Storage Devices”的报告；范玉雷博士做了题为“Session on Transaction of Flash-DB”的报告，该报告介绍了本项目组最近关于事务方面的工作。

2010.10.19

社会计算与人文社会科学研究研讨会

2010 年 10 月 19 日上午，孟小峰教授在人大逸夫楼主持召开了“社会计算与人文社会科学研究”研讨会。副校长冯惠玲教授和科研处处长杜鹏教授出席会议，国内社会计算领域的权威学者中科院自动化所王飞跃研究员、国际社会计算领域的著名学者香港中文大学的金国庆教授，以及经济学院刘元春教授、信息资源管理学院安小米教授、社会与人口学院冯仕政副教授、新闻学院胡百精副教授、信息学院社会计算研究组余力博士分别就社会计算与人文社会科学研究从不同的专业视角做了专题研讨报告。

2010.10.30

第二届云数据管理国际研讨会

2010 年 10 月 30 日，WAMDM 实验室在加拿大多伦多成功承办了第二届云数据管理国际研讨会，孟小峰教授担任本次研讨会的联合主席。本次会议主题主要包括：大规模数据管理系统设计、云数据隐私保护与安全等方面。本次研讨会汇集了中国人民大学、加拿大滑铁卢大学、美国宾夕法尼亚大学等多所知名大学的科研成果，最终共录用 8 篇论文。会议期间，WAMDM 实验室的史英杰博士做了“ESQP: An Efficient SQL Query Processing for Cloud Data Management”、“Benchmarking Cloud-based Data Management Systems”报告，受到与会人员的广泛好评。

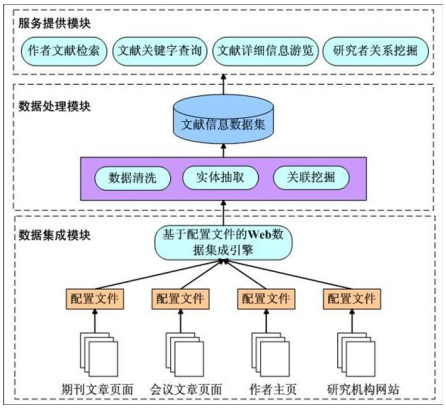
2010 中国计算机大会
参展系统

ScholarSpace：面向计算机领域的中文文献集成系统

Web数据管理是实验室持续近十年的研究方向，在Web数据集成研究我们提出了一种面向领域的Web数据集成技术框架，并利用该技术框架开发了ScholarSpace等多个应用系统。ScholarSpace是一个以作者为中心的自动集成、增量更新的计算机领域中文文献集成系统，在业内颇受好评。

系统特点

- ❖ **精选数据源：**只收录国内计算机领域权威期刊和学术会议的数据
- ❖ **以作者为中心：**系统内容组织以作者为中心，自动构建各种关联
- ❖ **集成丰富信息：**集成作者图片、论文列表、发表数量曲线、合作作者列表、单位信息、承担项目信息等，构建作者个人学术页面
- ❖ **可视化展示：**系统基于Flash技术动态展示一名作者与其合作作者的关系
- ❖ **同名作者区分：**利用GHOST算法及基于作者单位信息的聚类算法，实现同名作者区分功能



系统架构

- ❖ 数据集成模块

  利用面向领域的数据集成技术构建基于配置文件的Web数据集成引擎，集成多数据源

- ❖ 数据处理模块

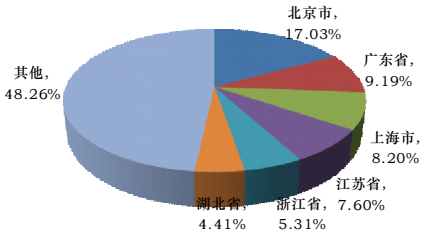
  通过数据清洗、实现发现和关联挖掘建立以“实体-关联”组织的文献信息数据库

- ❖ 服务提供模块

  提供学术主页生成、作者/文献检索、实体关联发掘等应用服务

社会关注

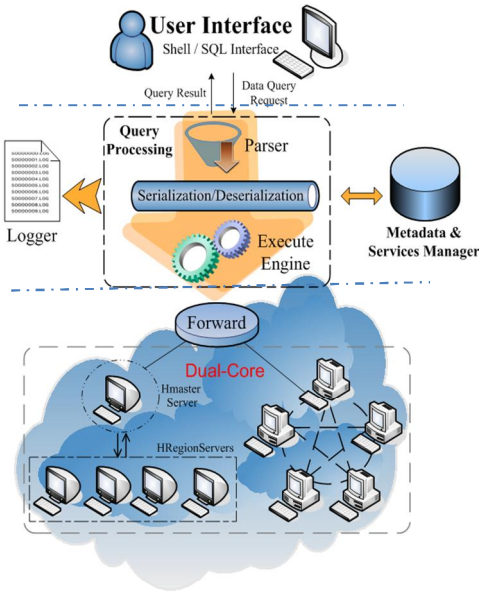
中国计算机学会会刊/网站新闻



系统访问来源分布图

云数据管理系统TaijiDB

TaijiDB系统综合主从（Master-slave）和点对点（P2P）两种云数据存储架构，继承了两种架构在CAP、数据写操作、MapReduce、系统性能、应用场景等方面的优势与特点。TaijiDB支持SQL和Shell两种接口，可在低端机器上进行部署，可靠性高，扩展性强，并能自动进行冗余备份和错误恢复，管理海量的纷繁复杂的混乱数据，使之有条有理，便于查询。

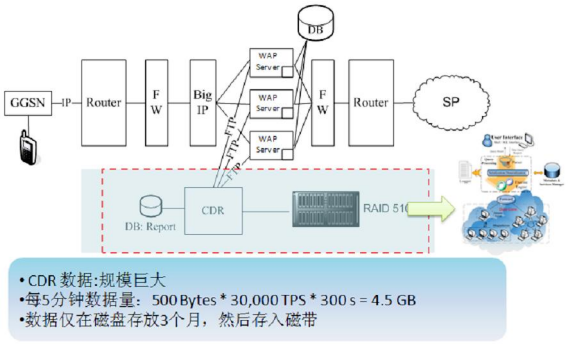


系统架构

- **用户接口层：**为用户和应用程序提供SQL、Shell和编程API接口，将用户的查询请求转发给查询处理层，并接收处理查询处理层返回的查询结果。
- **查询处理层：**对接收到的查询请求进行SQL解析和查询优化，生成查询计划，通过统一的API接口将查询计划转发给存储层进行实际查询处理，并将查询结果返回给用户接口层。另外进行操作日志和元数据的管理。
- **存储管理层：**负责实际数据存储和节点机器的维护，如数据拆分，数据备份，容错处理，节点恢复，节点定位等，根据查询计划对相关节点进行实际数据的存取。

系统功能

- 建立适配层桥接底层的主从结构和点对点结构。上层通过统一API调用进行操作，底层的差异对上层透明。
- 支持部分SQL查询语句，包括建表、插入数据、选择数据、删除数据和数据表等。
- 支持部分文件操作



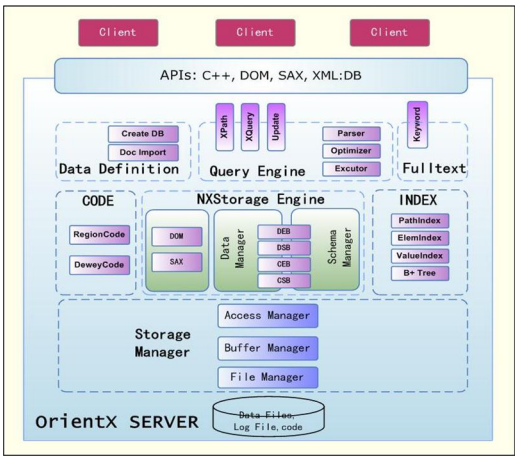
应用场景

OrientX：纯XML数据库系统

OrientX 是由中国人民大学网络与移动数据管理实验室开发的纯XML数据库系统。从2002年以来，我们不断完善OrientX系统，到目前为止已经陆续发布了6个版本。系统依据XML的数据模型和特性高效地存储数据，同时支持XQuery和Xpath查询标准以及XQuery/Update更新标准。OrientX 采用C/S模式，客户端提供了图形化用户界面方便用户操作和查询数据，服务器端提供一套访问数据库的API接口。

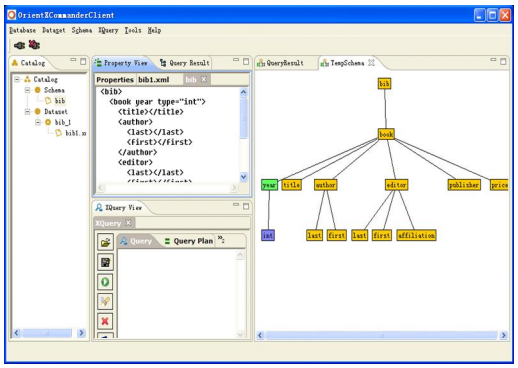
开发版本

- OrientX 1.0 (2002-2003):包括存储模块、模式管理和数据据管理模块三个模块;
- OrientX 1.5 (2003-2004) :增加了XPath查询处理模、编码模块和索引模块;
- OrientX 2.0 (2004-2005):增加了基于导航的查询引擎;
- OrientX 2.5 (2005-2006): 添加了基于代数的查询引擎;
- OrientX3.0 (2006-2007)：支持XQuery/Update标准的插入、删除、替换以及重命名四个操作;
- OrientX3.5 (2008-2009)：支持XQuery/Update的transform操作并加入整枝匹配算法。



OrientX3.5 系统结构图

系统功能



用户界面

- 数据库的建立和维护：管理模式信息，创建和删除数据库;
- 数据操作：包括对数据库数据的检索以及增删改操作;
- 数据组织、存储和管理功能：提高存储空间的利用率以及查找、增删改等操作的时间效率;
- 应用程序接口：提供了一套C/ C++ API接口;
- 图形化用户界面：方便用户进行数据库管理、查询和更新操作以及查看查询执行计划。

OrientPrivacy：隐私保护系统

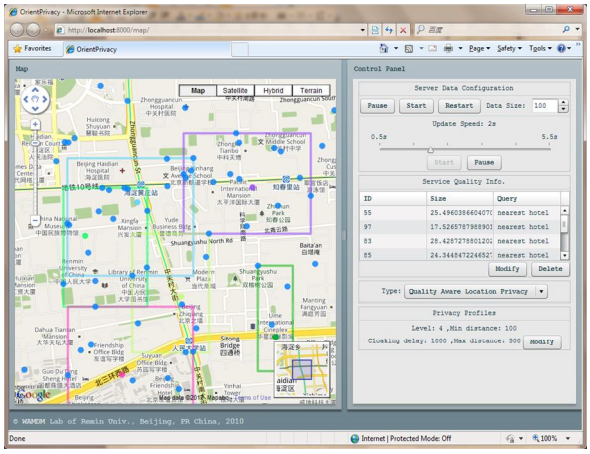
随着无线通讯技术和移动定位技术的发展，移动用户的隐私也越来越受到人们的关注。基于位置位置服务中有两种隐私类型：用户位置隐私和查询内容隐私。OrientPrivacy系统集成多种隐私保护方法，可以根据用户隐私需求保护其位置隐私和查询内容隐私。

系统特点

- 采用多种算法，可以保护用户的位置隐私和查询内容隐私
- 用户可以配置隐私需求参数，服务器根据其参数和不同查询类型自动选取合适的隐私保护算法
- 服务器界面可以实时显示隐私保护处理状态，对比匿名前后结果
- 服务器端可以在线访问
- 使用真实的地图数据

手机主界面

手机端隐私保护参数设置

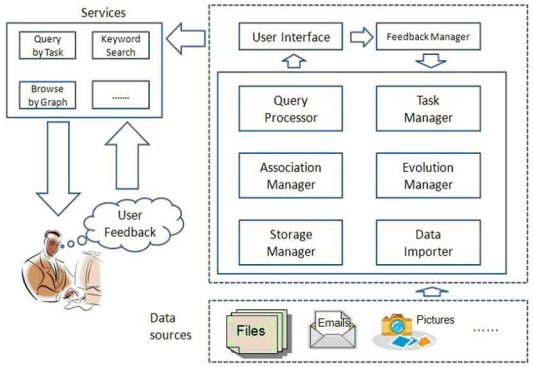


服务器端主界面

OrientSpace：个人数据空间系统

在信息社会，计算机和通信技术的发展使数据量急速膨胀，每个人都面临着数据海量以及高度异质带来的挑战，却缺乏有效的管理工具。例如，用户很难在电脑中找到一个特定的文件。因此，如何管理海量、分布、异构数据成为一个值得关注的问题。

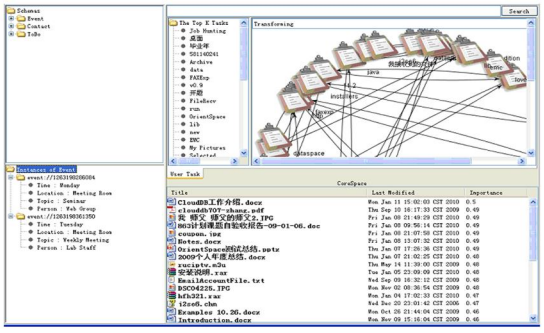
个人数据空间系统OrientSpace即基于数据空间理论，以个人数据空间管理为应用背景，关注数据管理中的任务管理、演化管理、数据关联管理、用户反馈管理等，致力于为用户提供高质量的数据服务。



OrientSpace系统框架图

关键技术

- 通过用户行为监控引擎识别用户访问行为
- 自动形成用户任务
- 构建并自动更新核心数据空间
- 构建并自动更新任务空间
- 基于三元组的全局式存储策略
- 多级索引、选择性索引与混合索引相结合的索引策略



OrientSpace主界面

系统特色

- 灵活的数据模式，允许用户灵活自由地创建和修改数据模式
- 支持复杂多样的数据关联，并支持基于关联的查询
- 支持基于用户行为的数据空间演化，充分考虑了用户的个性化信息
- 提供自动建立和更新数据空间的功能
- 支持基于任务的数据操作，便于用户浏览任务及其关联的文件

下载地址：http://idke.ruc.edu.cn/projects/pds.htm

六、 **Flash DB:** 闪存数据库系统

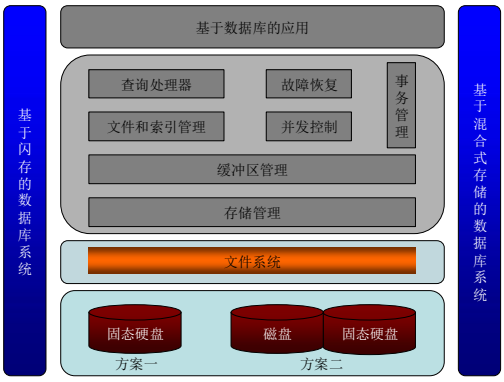
Flash DB：闪存数据库系统

随着闪存容量的快速增长，闪存上的数据管理问题已成为新的挑战性问题。从数据库技术的角度研究闪存数据管理；从闪存的器件特性入手，构建新型的闪存数据库系统(简称Flash DB)。

闪存特性

- 非易失性、高速、高抗震性、低功耗、小巧轻便；
- 无机械延迟，即没有寻道时间和旋转时间；
- 原位更新代价过大，一般采用换位更新方式；
- 读/写/擦除速度存在较大差异，读操作速度明显高于写操作，而擦除操作则更慢；
- 块具有擦除次数限制，超过限制后，该块将成为坏块，不再可靠。

固态硬盘



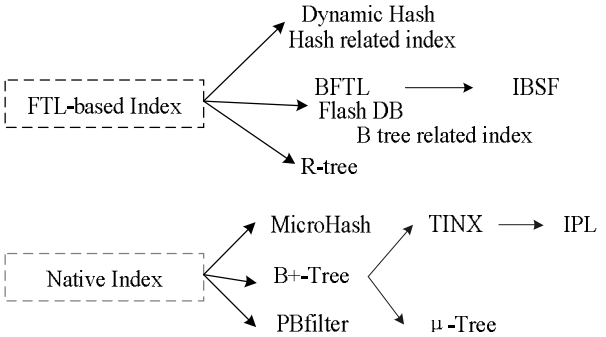
闪存数据库系统体系结构

闪存数据库系统

- 可剪裁闪存数据库体系结构;
- 闪存数据库存储管理;
- 闪存数据库索引;
- 闪存数据库缓冲区管理;
- 闪存数据库查询处理;
- 闪存数据库事务处理;
- 闪存数据库隐私保护;

基于闪存的索引

- 基于**FTL**的索引：FTL为闪存转换层，把闪存转换为块设备来使用，基于FTL的索引主要分为Hash、B-tree和R-tree三种;
- 基本的索引方法：主要分为Hash、B+-Tree和Pbfilter三种;



实验室世博之旅

WAMDM 实验室世博之旅

为了放松心情，缓解压力，2010 年 7 月 30 日至 8 月 2 日，WAMDM 实验室师生集体前往上海参观 2010 年上海世博园。参观世博园期间，大家饱览了世界各国的特色展馆，对异域风情有了切身的感受，也深刻体会到了“城市,让生活更美好”的世博主题。经过三天的参观、游览，大家既放松了心情，增强了友谊，同时又满怀信心准备迎接下学期紧张、繁忙的学习生活。

城市，让生活更美好！

实验室集体合影

韩国企业联合馆

夜幕中的东方明珠！

看看我是谁？

附 录

实验室研讨会

2010.12.10 Venue: FL1, Meeting Room, Information Building	
Zhichao Liang (Flash Group)	A novel method to extend flash memory lifetime in flash-based DBMS Abstract: As the capacity increases and the price drops gradually, flash memory is becoming the promising replacement of disk, even in the enterprise applications. However, flash memory suffers from erase-before-write and limited write-erase cycles at the same time, which means the abuse of write,especially small and random write, will wear a flash block out quickly. We analyze the free space management in traditional DBMS and point out its disadvantage when used on flash device. In addition, we also propose a new solution involving free space management and buffer management to extend the lifetime of flash memory by reducing the number of write I/O.
Xiaoying Qi (Flash Group)	An Operation Aware Flash Translation Layer for Enterprise-class SSDs Abstract: Flash translation layer is an important firmware in flash-based devices. It is critical to affect the performance of flash-based devices. So when SSDs are used in enterprise-class environment, FTL should be redesigned to improve the whole performance. In this report, we introduce an operation aware flash translation layer for enterprise-class SSDs.
2010.12.03 Venue: FL1, Meeting Room, Information Building	
Wei Tong (Web Group)	A Structured Approach to Query Recommendation With Social Annotation Data [ppt] Abstract: Query recommendation has been recognized as an important mean to help users search and also improve the usability of search engines.
Sen Yang (Web Group)	Introduction to OpenScholar Abstract: OpenScholar is a web system to build scholars' homepage automatic. Its features of searching scholars' infomation and dynamic maintenance can help users build their homepages easily and fast.
2010.11.26 Venue: FL1, Meeting Room, Information Building	

Haiping Wang (Cloud Group)	**Research of query optimization in the cloud** Abstract: In cloud data management systems,data is partitioned into blocks and replicated.It is nesscary to translate some data blocks when we do some types of query processing.So we did some research on how to finish the query with little costs.
Xiaojian Zhang (Web Group)	**Record Linkage with Uniqueness Constraints and Erroneous Values** [ppt] Abstract: This paper presents some challenges of record linkage and data fusion in heterogeneous data sources with uniqueness constraints and erroneous values, models those records by utilizing K-partite graph, and proposes clustering algorithm and matching algorithm to cope with duplicates and conflicting data.
2010.11.19 Venue: FL1, Meeting Room, Information Building	
Yun Deng (Web Group)	**Evaluating Entity Resolution Results** [ppt] Abstract: Entity Resolution is an important technique in data integration. Similar to clutering and partition, ER tries to identity the same entity among messes of records. This report focus on an ER results measure,GMD.
Jing Zhao (Cloud Group)	**Research on Query Processing** Abstract: Query Processing is an difficult problem in both parallel database and cloud-based database. We briefly introduce basic query processing steps in centralized database and parallel database, and talk something about web-scale query processing, including MapReduce debates, MapReduce-based join algorithms, etc. Finally, we introduce main idea of our work and some future work.
2010.11.14 Venue: FL1, Meeting Room, Information Building	
Lizhen Fu (XML Group)	**Diversification for Keyword Search on Graph Data** Abstract: Keyword search is the de facto information retrieval mechanism for data on theWorld WideWeb. It also proves to be an effective mechanism for querying semi-structured and structured data, because of its user friendly query interface.Recently, query processing over graph-structured data has attracted increasing attention.In this report,we focus on the semantic Diversification of results from keyword search on graph.
Qingling Cao (Flash Group)	**Enterprise Application of SSD** [ppt] Abstract: SSD is becoming more and more popular in enterprise.But there is a

	question,if the platform ready for SSD？This report solved the question.And it also introduced about SSD RAID.
2010.11.06 Venue: FL1, Meeting Room, Information Building	
Yingjie Shi (Cloud Group)	**CIKM2010 Story** Abstract: In this talk, I presented some papers and one panel related to Cloud Data Management in CIKM2010. Then I gave some summary of CIKM2010.
Bingbing Liu (Cloud Group)	**RHP:a new partitioner to improve the efficiency of range query in cassandra** Abstract: The conflicting problems of ensuring data-access load balancing and efficiently processing range queries leads to that cassandra can't support range query very well.So how to trade off them is the key point.
2010.10.30 Venue: FL1, Meeting Room, Information Building	
Dongqi Liu (Mobile Group)	**Spatial-temporal sequence views query demo** [ppt] Abstract: We have taken some informations of views on flicker to analyse how to traverse these views from the realistic perspective.If a user wants to traverse the views in a limited time,he may have several solutions,but which one is the most valuable one?Based on our ideas,we give three solutions to slove this problem,and will show you the solutions in our demo.
Long Liu (Cloud Group)	**Survey of Object-based Storage** [ppt] Abstract: Object-based Storage, a new approach to storage technology, is a subject of academic research and development in the storage industry. This survey describes the main points of object-based storage technology from five aspects. That is why we introduce the concept of object-based storage, what it is, how to take advantage of it, what the status of object-based storage in both industry and academic research is, and what we can do about it.
Yi Huang (Mobile Group)	**Android Development tutorial** [ppt] Abstract: Android, released by Google on Nov. 5th, 2007, is a Linux kernel-based operating system designed for smartphones. In the past three years, Android system has archived a great market share and this share is still increasing. Meanwhile, Android has been attracting more and more developers who have made contributions to more than 100,000 applications in the second largest online app store called Android Market. This tutorial introduces application development on Android platform and the mechanism of Android as well.
2010.10.23 Venue: FL1, Meeting Room, Information Building	

Fan Yulei (Mobile Group)	**Flash-based Multi-Version Data Storage** Abstract: Because of characteristics of Flash Memory and Data storage of PostgreSQL, More update operations and small random write operations run on flash memory. These operations will degrade the performance of DBMS and age of flash memory. Flash-based Multi-Version Data Storage(FMVDS) is proposed to reduce update and write operations and finally reduce erase times. In FMVDS, transaction table item with timestamp and data record with a point to older version data implement high concurrency control and quickly recovery.
Daxing Jiang (MSRA)	**Context-Aware Search** Abstract: Introduce the research on context-aware search in MSRA.
2010.09.25 Venue: FL1, Meeting Room, Information Building	
Youzhong MA (Web Group)	**Entity Resolution with Evolving Rules** [ppt] Abstract: Entity resolution (ER) identifies database records that refer to the same real world entity. In practice,ER is not a one-time process,but is constantly improved as the data, schema and application are better understood. We address the problem of keeping the ER result up-to-date when the ER logic ñevolvesò frequently. A naive approach that re-runs ER from scratch may not be tolerable for resolving large datasets. This paper investigates when and how we can instead exploit previous ñmaterializedò ER results to save redundant work with evolved logic. We introduce algorithm properties that facilitate evolution, and we propose efficient rule evolution techniques for two clustering ER models: match-based clustering and distance-based clustering. Using real data sets, we illustrate the cost of materializations and the potential gains over the naive approach.
Jinzeng Zhang (Mobile Group)	**VLDB paper report** Abstract: This report includes two parts.The fisrt is retrieving top-k prestige-based relevant spatial web objects,this method proposes the concept of prestige-based relevance, the top-k spatial web objects is ranked according to both prestige-based relevance and location proximity.The second part introduces how to mine significant sematic location from GPS data,this method models the relationships between locations and the relationships between locations and users with a two-layered graph.Based on this,this paper proposes a new ranking model which assign significance to locations.
Yingjie Shi (Web Group)	**Paper Summary of VLDB2010** Abstract: Papers of VLDB2010 about cloud are classified into four aspects:

	Cloud Data Management Systems, Benchmark, Query Processing and open questions. This report introduces the motivation, key technology and inspiration to our research work.
2010.09.18 Venue: FL1, Meeting Room, Information Building	
Zhongyun Wang (Graduate)	**New Experience in MSRA** Abstract: Introduce personal life , feelings in MSRA.
Da Zhou (Graduate)	**Introduction to Cloud and Flash Memory Management** Abstract: Share new findings and thoughts about cloud computing and flash memory management.
2010.06.19 Venue: FL1, Meeting Room, Information Building	
Zheng Huo (Mobile Group)	**Privacy-preserving of Trajectory Data: A Survey** [ppt] Abstract: This survey discussed trajectory data privacy preservation techniques in 4 motivating applications. For online trajectory data privacy preservation, service is centric, trade-off is between QoS and privacy preservation; For offline trajectory data privacy preservation, data is centric, trade-off is between data quality and privacy preservation.
Qingsong Guo (XML Group)	**XML Keyword Query Refinement** [ppt] Abstract: In this report, we discussed about the problem of query refinement in traditional IR and novel XML keyword search. The main part we mentioned is about the task and ways of XML keywords query refinement. In addition, we classified the existing work of XML keywords query refinement, and give out my own work on it.
2010.06.12 Venue: FL1, Meeting Room, Information Building	
Ruxia Ma (Web Group)	**Credibility on the Web: A Survey** Abstract: This survey discussed credibility on the web from three kinds of entities
Wei Chen (Web Group)	**Information Quality and Trustworthiness in Wikipedia** Abstract: In this talk we discussed the problem of information quality and trustworthiness of Wikipedia and introduced some research topics. In addition, we gave an brief overview of current research papers about this topic in WWW, WICOW etc.
2010.06.05 Venue: FL1, Meeting Room, Information Building	

Xiangmei Hu (Cloud Group)	**Index for cloud data management** Abstract: This report mainly introduces why we build index on cloud data management、some related work about index for cloud data management and our work progress on index research.
Haiping Wang (Cloud Computing Group)	**NoSQL Overview** [ppt] Abstract: This report simply introduced NoSQL,four reasons why nosql concept was introduced, the history, definition,Three fundamental theories of NoSQL and categories of NoSQL databases.
2010.05.29 Venue: FL1, Meeting Room, Information Building	
Lizhen Fu (XML Group)	**Keyword search on Graph** Abstract: In this report, I introduce methods that perform keyword search on graph data. Keyword search provides a simple but user-friendly interface to retrieve information from complicated data structures. In this discussion, I focus on three major challenges of keyword search on graphs. First, an answer to a keyword search on graphs,or, what qualifies as an answer to a keyword search. second, what constitutes a good answer, or how to rank the answers;Third, how to perform keyword search efficiently.
Lizhen Fu (XML Group)	**The Integration of TelecommuniCations Networks, Cable TV Networks and The Internet** [ppt] Abstract: This report introduces the conception The Integration of TelecommuniCations Networks, Cable TV Networks and The Internet firstly.then present its development Process and its advantages. At last,I describe the current situation of Integration of the three kides of networks at abroad.
2010.05.22 Venue: FL1, Meeting Room, Information Building	
Yubo Kou (Web Group)	**Elementary Structure-based Graph Matching** Abstract: Past graph matching techniques is vertex-based. Which means they first find candidate set for each node in the query, then perform searching algorithm to find a match. This approach cost too much since there might be too many candidates for each node, and these candidates will form a large search space. To reduce the search space, it is profitable to elevate the granularity of matching algorithm
Wei Wang (XML Group)	**Data deduplication** Abstract: This report introduces some methods of data deduplication, such as Hash-based algorithms, Delta algorithms.

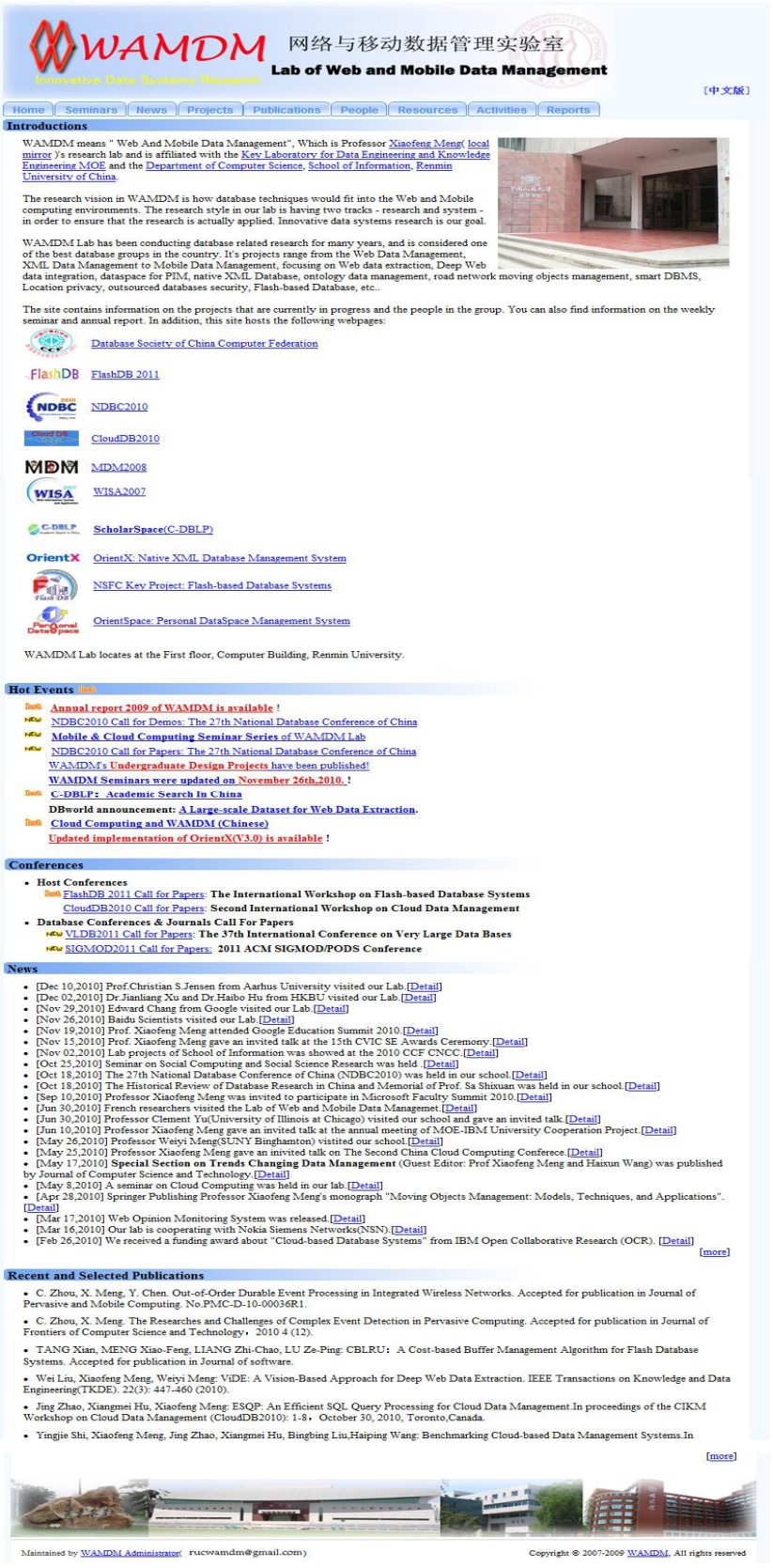
2010.05.08 Venue: FL1, Meeting Room, Information Building	
Yingjie Shi (Web Group)	**Benchmark results and analysis** Abstract: This report introduces the test results of benmarks on cloud-based DBMSs, and does analysis on the restuls.
Haiping Wang (Cloud Computing group)	**Architecture and Design of Distributed Database Systems** [ppt] Abstract: This report introduces serval kinds of architectures about Distributed Database Systems based on relational data model, it also introduces two horizonal and a verical fragmentatin method and the allocation model for DDBMS.
2010.04.24 Venue: FL1, Meeting Room, Information Building	
Xuan Zhou (CSIRO, Australia)	**Integrating User Interfaces of DB and IR Systems** Abstract: In contrast to classical databases and IR systems, real-world information systems have to deal increasingly with very vague and diverse data structures. While current object-relational database systems require clear and unified data schemas, IR systems usually ignore the structured information completely. Malleable schemas, as recently introduced, provide a novel way to deal with vagueness,ambiguity and diversity by incorporating imprecise and overlapping definitions of data structures. In this talk, I will introduce a novel query relaxation scheme that enables users to find best matching information by exploiting malleable schemas. Our scheme utilizes duplicates to discover the correlations within a malleable schema, and then uses these correlations to appropriately relax users' queries.Then, it ranks results of the relaxed queries according to their respective probability of satisfying the original queryôs intent. Our experiments with real-world data confirmed its performance and practicality.
2010.04.17 Venue: FL1, Meeting Room, Information Building	
Zhichao Liang (Flash Group)	**Hush-Tell You Something Novel About Flash Memory !** Abstract: This report introduces some work of Non-volatile Systems Laboratory in UCSD in which a lot of tests on flash memory were done. According to the test results, some applications were deviced, including a variation-aware FTL which is called Mango, a flash-aware data encoding and a system architecture for data-centric applications whose name is Gordon.
Yulei Fan (Mobile Group)	**Existed DBMS on SSD** Abstract: By analysis of IOps of HDD and SSD,we can compare IOps of SSD with IOps of HDD. By analysis of tpcc of MySQL and PG on SSD and HDD, we can compare performance of existing DBMS on SSD

	with that on HDD. Then we propose some ideas
2010.04.03 Venue: FL1, Meeting Room, Information Building	
Zhongyuan Wang (Web Group)	**Web Pages Extraction Technologies in the Opinion Monitoring System** Abstract: This report introduces two web pages extraction technologies in our opinion monitoring system, and some popular tools for system development.
Yi Huang (Mobile Group)	**An Introduction to Flex** [ppt] Abstract: Nowadays Flex is very popular in developing Rich Internet Applications. This report introduces what is Flex and its history and also discusses its mechanism, advantages, applications and the differences between other RIA techniques.
Jing Zhao (Web Group)	**System Environment and MapReduce Framework** Abstract: This report includes the introduction of the construction of our cloud data management platform and a brief talk about MapReduce framework.
Zhichao Liang (Flash Group)	**An Introduction to the Source Insight** [ppt] Abstract: This report introduces a project-oriented program editor and code browser,Source Insight,which parsers your source code and maintains its own database of symbolic information dynamically while you work,and presents useful contextual information to you automatically.
2010.03.27 Venue: FL1, Meeting Room, Information Building	
Chunjie Zhou (Web Group)	**IO3:Interval-based Out-of-order Event Processing in Pervasive Computing** Abstract: In pervasive computing environments, complex event processing has become increasingly important in modern applications. A key aspect of complex event processing is to extract patterns from event streams to make informed decisions in real-time. However, network latencies and machine failures may cause events to arrive out-of-order. In addition, existing literatures assume that events do not have any duration, but events in many real world application have durations, and the relationships among these events are often complex. In this work, we first analyze the preliminaries of time semantics and propose a model of it. A hybrid solution including time-interval to solve out-of-order events is also introduced, which can switch from one level of output correctness to another based on real time. The experimental study demonstrates the effectiveness of our approach.
Bingbing Liu (Cloud Group)	**ICDE2010 Keynote - what's new in the cloud** [ppt] Abstract:

	This report talks about why we should do cloud computing,how to do and what to do.
Yukun Li (Web Group)	**Survey of ICDE2010 and SIGMOD2010** Abstract: Based on the accepted papers, this presentation made a survey on recent international database conferences ICDE2010 and SIGMOD2010, and analyzed the research focuses of database area.
2010.03.20 Venue: FL1, Meeting Room, Information Building	
Da Zhou (Flash Group)	**RWConvertor: Random Write Optimization for SSD** Abstract: With the development of electronic technologies, Solid State Drive (SSD) emerge as new data storage media with low power consumption, high shock resistance and lightweight form. Besides these, the most attractive characteristic is the high random read speed because of no mechanical latency. Therefore SSD have been widely used in laptops, desktops, and data servers in place of hard disk during the past few years. However, poor random write performance becomes the bottle neck in wider applications. Random write is almost two orders of magnitude slower than both random read and sequential access, so write-intensive applications have very low performance on SSD. In this paper, the first time we propose to insert unmodified data into random write sequence in order to convert random writes into sequential writes, and then data sequence can be flushed at the speed of sequential write. Further, we improve the write performance by Optimum Converted Write Sequence (OCWS). Strict mathematical proof decides the location and number of inserted data items during the course of getting OCWS. We also optimized our method with throughput, which is decided by gain and granularity, of OCWS when applied in data stream.
2010.03.13 Venue: FL1, Meeting Room, Information Building	
Jinzeng zhang (XML Group)	**Approaches to internet of things** Abstract: As the next generation of information technology,the internet of things has drawn public attenention.It enables the internet to reach out into the real world of physical objects.This report first gives the concept of the internet of things,then introduces the system architecture and key techniques and gives three applications.Fianlly,I put forward to the furture direction.
Xing Hao (Mobile Group)	**Related Work about Internet of Things** [ppt] Abstract: This report gives an overview of the related and future work about Internet of Things and focus on the The RFID Ecosystem Experience handled by University of Washington.
2010.03.06 Venue: FL1, Meeting Room, Information Building	

Yingjie Shi (Web Group)	**Open Source Cloud-based DBMS Experiments** Abstract: This report introduces existing expriment benchmarks of cloud-based DBMS experiments. We describe the testbed of our experiment, and show the tasks and results.
Zhongyuan Wang (Web Group)	**System Architecture Design and Implementation of Cloud-based Database System** Abstract: The Cloud-based Database project at WAMDM aims at researching new storage and database system which can support the next generation of data storage and management and applied to mobile communications. This report introduced the architecture design and implementation of our cloud-based database system.
2010.01.09 Venue: FL1, Meeting Room, Information Building	
Dr. Yueguo Chen (Invited Talk)	**Time series and Interactive media** Abstract: Time series and interractive media have large applications in computer games or so. One of the most important problem for pattern detection in streaming time series could be how to define a effective distance metric.We propose a novel warping distance and efficient approach for continuous pattern detection. For the interavtive media database, it focus on the index,storage structure for smart media objects, similarity metrics and query procesing on multimedia data.
Xiaoying Qi (Flash Group)	**FTL Algorithms and Native Flash Experiments** Abstract: This report introduces five flash translation layer algorithms, such as BAST, FAST, LAST, and DFTL etc. We mainly describe the main ideas of those algorithms and their realization. Then we introduce the native flash experiments.

实验室网站



http://idke.ruc.edu.cn/wamdm

实验室成员

Faculty Members

Xiaofeng Meng
孟小峰
博士，教授，博导
WAMDM 实验室负责人

Nan Yang
杨楠
博士后，副教授

Qing Liu
刘青
博士，副教授

Yunpeng Cai
柴云鹏
博士，讲师

Gang Yang
杨刚
博士，讲师

Zhiyong Shan
单智勇
博士，讲师

Ph.D. Candidates

Chunjie Zhou
周春姐

Xian Tang
汤显

Yulei Fan
范玉雷

Lizhen Fu
富丽贞

Zheng Huo
霍峥

Jinzeng Zhang
张金增

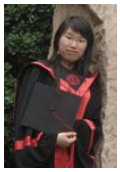
Yingjie Shi
史英杰

Ruxia Ma
马如霞

Xiaojian Zhang
张啸剑

Youzhong Ma
马友忠

M.Sc. Students



Jing Zhao
赵婧

Xiangmei Hu
胡享梅

Wei Wang
王伟

Qingsong Guo
郭青松

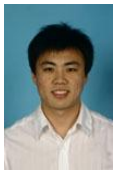
Zeping Lu
卢泽萍

Zhichao Liang
梁智超

Xiaoying Qi
綦晓颖

Yi Huang
黄毅

Haiping Wang
王海平



Bingbing Liu
刘兵兵

Wei Chen
陈威

Jie Wen
文洁

Dongqi Liu
刘东琦

Long Liu
刘龙

Qingling Cao
曹庆铃

Wei Tong
童薇

Yun Deng
邓云

Sen Yang
杨森

实验室毕业生

2010 年毕业生去向

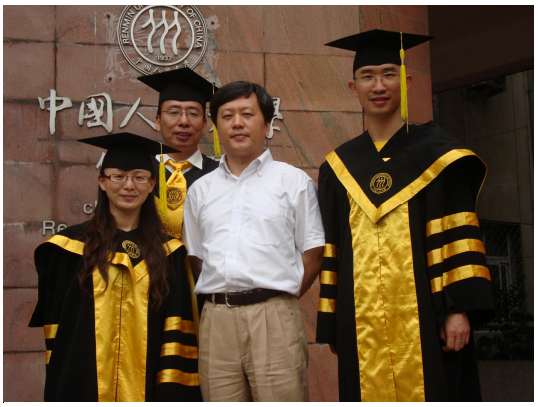
姓名	学历	时间	毕业去向
李玉坤	博士	2010 年 7 月	天津理工大学
潘晓	博士	2010 年 7 月	石家庄铁道大学
周大	博士	2010 年 7 月	中国移动研究院
徐俊劲	硕士	2010 年 7 月	百度
王仲远	硕士	2010 年 7 月	微软亚洲研究院
艾静	硕士	2010 年 7 月	国家发改委国家投资项目评审中心
郝兴	硕士	2010 年 7 月	百度
张相於	硕士	2010 年 7 月	搜狗

2009 年毕业生去向

姓名	学历	时间	毕业去向
周军锋	博士	2009 年 7 月	燕山大学
姜芳芃	博士	2009 年 7 月	徐州师范大学
贾琳琳	硕士	2009 年 7 月	中国农业银行
黄静	硕士	2009 年 7 月	中国工商银行软件开发中心
朱金清	硕士	2009 年 7 月	百度
王伟	硕士	2009 年 7 月	百度
向锂	硕士	2009 年 7 月	中化集团石油中心

实验室毕业生集体照片

2010 年毕业生



2009 年毕业生

2005 年毕业生

内部资料，妥善保存

网络与移动数据管理实验室
地址：中国人民大学原信息楼一层
网址：http://idke.ruc.edu.cn/wamdm
电话：010-62512719
传真：010-62514798
编辑：马友忠 张啸剑