计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140116
52(2):343-361,2015

基于 PCM 的大数据存储与管理研究综述

吴章玲[1,2] 金培权[1,2] 岳丽华[1,2] 孟小峰[3]
[1](中国科学技术大学计算机科学与技术学院 合肥 230027)
[2](中国科学院电磁空间信息重点实验室 合肥 230027)
[3](中国人民大学信息学院 北京 100081)
(linglang@mail.ustc.edu.cn)

A Survey on PCM-Based Big Data Storage and Management

Wu Zhangling[1,2], Jin Peiquan[1,2], Yue Lihua[1,2], and Meng Xiaofeng[3]
[1](*School of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027)
[2](*Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei* 230027)
[3](*School of Information, Renmin University of China, Beijing* 100081)

Abstract Big data has become a hot topic in both academia and industry. However, due to the limitations of current computer system architectures, big data management is facing a lot of new challenges w.r.t. performance, energy, etc. Recently, a new kind of storage media called phase change memory (PCM) introduces new opportunities for advancing computer architectures and big data management, due to its non-volatility, byte-addressability, high read speed, low energy, etc. As a kind of non-volatile storage media, PCM has some unique features of DRAM, such as byte-addressability and high read/write performance, thus can be regarded as a cross-layer storage media for re-designing current storage architecture so as to realize high-performance storage. In this paper, we summarize the features of PCM, and present a survey on PCM-based data management. We discuss the related advances in terms of two aspects, namely that PCM is used as secondary storage and that PCM is used as main memory. We also introduce the current studies on the applications of PCM in various areas. Finally, we propose some future research directions on PCM-based data management so as to provide some valuable references for big data storage and management on new storage architectures.

Key words phase change memory; main memory system; hybrid main memory; big data management; big data storage

摘 要 大数据已经成为当前学术界和工业界的一个研究热点.但由于计算机系统架构的限制,大数据存储与管理在性能、能耗等方面均面临着巨大的挑战.近年来,一种新型存储介质——相变存储器(phase Change Memory, PCM)——凭着其非易失、字节可寻址、读取速度快、低能耗等诸多优点,为计算机存储体系结构和数据管理设计带来了新的技术变革前景,也为大数据存储和管理带来了新的契机.PCM 既是一种非易失存储介质,同时又具备了内存的字节可寻址和高速随机访问特性,模糊了主存和外存的界限,有望突破原有的存储体系架构,实现更高性能的存储与数据管理.概述了 PCM 存储器的发展现状;总结了目前基于 PCM 的持久存储技术和基于 PCM 的主存系统等方面的研究进展;并讨论

收稿日期:2014-10-15;修回日期:2014-12-08
基金项目:国家自然科学基金面上项目(61472376);国家自然科学基金重点项目(60833005)
通信作者:金培权(jpq@ustc.edu.cn)

了 PCM 在多个领域的应用现状. 最后,给出了基于 PCM 的大数据存储与管理研究的若干未来发展方向,从而为构建新型存储架构下的大数据存储与管理技术提供有价值的参考.

关键词 相变存储器;主存系统;混合主存;大数据管理;大数据存储

中图法分类号 TP311

随着数据的爆炸式增长,大数据已经成为当今时代学术界与工业界的热门话题. "大数据"被定义为"无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合"[1]. 目前,研究者普遍认为大数据具有 4V(volume, variety, value 和 velocity)特征,分别表现了大数据在数据规模、数据类型、数据价值和数据处理速度方面的特点. 大数据的这些特性对现有数据管理技术提出了新的挑战. 其中首当其冲的是大数据存储技术的挑战,即大数据管理如何满足持续增长的超大规模数据的高效能存储要求.

针对大数据存储问题,学术界和工业界都提出了一系列设计方案,包括 Hadoop 分布式文件系统(Hadoop distributed file system, HDFS)、以非关系型数据库(not only SQL, NoSQL)为代表的大规模分布式数据库系统设计[2-3]、基于 DRAM 的内存数据库技术[4]等. 但 HDFS 针对大文件存储而设计,无法做到高速地随机读写;NoSQL 分布式数据库技术本质上仍然是采用传统的"CPU-DRAM-磁盘/固态硬盘"的存储架构,无法避免大数据存取中 DRAM 和二级存储之间的 I/O 瓶颈;内存数据库技术试图通过 DRAM 的高性能优势来克服大数据管理与分析中的瓶颈,但大容量 DRAM 价格、能耗较高并且单节点 DRAM 容量已经很难扩充,限制了传统的基于 DRAM 的主存架构在大规模数据处理中的运用. 因此,高效能的大数据存储与管理仅从软件体系结构上考虑很难取得本质性突破,因为在大数据环境下内存与外存之间的 I/O 瓶颈很难克服. 高效能大数据存储和管理需要一种新型存储架构,结合创新的系统软件设计,改变大数据处理过程中对外存 I/O 的依赖,从而克服目前大数据存储和管理中的性能瓶颈.

近年来,闪存(flash memory)、磁性存储器(magnetic RAM, MRAM)、铁电存储器(ferroelectric RAM, FRAM)、相变存储器(phase change memory, PCM)等新型存储技术的出现,为研究适合高效能大数据存储和管理的新型存储架构带来了新的机遇. 闪存技术的快速发展对数据管理研究已经带来了巨大的冲击[5],但是,闪存由于其按页存取的方式和存取性能等因素限制,适合作为二级存储设备;基于闪存的数据管理只是优化了 I/O 延迟,在计算体系上没有本质改变. PCM 在最近几年发展迅速,已经引起了学术界和工业界的广泛关注[6]. 根据最近的一项论文统计数据[7],PCM 的研究热度接近和超出了有关闪存的论文数量. PCM 具有一些区别于已有存储介质的新特性. 首先,PCM 具有按字节存取的特点,可以与 DRAM 一样直接和 CPU 交互,这与之前的闪存、磁存储介质有着本质的不同. 其次,PCM 具有非易失性,能够提供持久数据存储的功能,没有 DRAM 的掉电即数据丢失的缺陷. 此外,PCM 基于微型相变单元存储数据的机理使其未来存储密度有望超过闪存,在存储容量上有着很大的提升空间. 因此,综合考虑介质的存储密度、成本以及性能等因素,PCM 被认为是未来最有前途的存储介质之一[8]. PCM 模糊了主存与外存的界限,在高速与海量存储方面具有巨大潜能,有望突破原有的存储体系架构,实现更高性能的存储.

本文概述了 PCM 存储器的发展现状;总结了目前基于 PCM 的持久数据存储技术、基于 PCM 的主存数据管理技术等方面的研究进展;并讨论了 PCM 在不同领域的应用现状;最后,给出了基于 PCM 的大数据管理研究的若干未来发展方向,从而为构建新型存储架构下的大数据存储与管理技术提供了有价值的参考.

1 PCM 存储器

PCM 是一种电阻式非易失性半导体存储器,以硫族化物材料作为存储介质,利用相变材料在不同状态时呈现出显著的电阻值差异性来实现数据存储. 相变材料通过电子脉冲可以在晶态(低电阻,表示为"1")和非晶态(高电阻,表示为"0")两种状态之间转换. PCM 具有速度快、非易失性以及高密度性等诸多优点,其读写和恢复数据的速度是闪存的 100 倍.

早在 20 世纪 60 年代人们就已经在研究相变存

储器,1966 年美国科学家 Ovshinsky 提出 PCM 相变存储技术[9]. 随着半导体工艺和相变材料的发展,相变存储器件性能优势逐渐显现,并逐步得到广泛的重视. 2007 年意法半导体(STMicroelectronics,ST)和 Intel 成立恒忆公司(Numonyx),专门致力于 PCM 的研发,两年后恒忆宣布量产 1 GB 的相变存储器产品. 2011 年 6 月底,IBM 研究人员在苏黎世展示了 PCM 的重大突破:多位封装,满足每单元存储多位数据的高存储容量需求,预示着 PCM 存储密度的进一步提高. IBM 实现的 PCM 测试芯片拥有 20 万个存储单元,该数据保存实验进行了 5 个月,这意味着多位 PCM 能达到适合实际使用的可靠性. 2012 年国际固态电路研讨会上,三星更是推出了 20 mm 支撑 8 GB 相变存储芯片. 而国内在 PCM 硬件设计与制造方面,上海微系统与信息技术研究所成功设计和制造了一款 1 KB 的相变存储芯片[10]和一款 8 MB 的相变存储器[11];宁波时代全芯科技有限公司 2013-11-29 向全球发布相变存储技术的芯片,成为中国第 1 家拥有相变存储技术自主知识产权的企业.

相变存储器的读写工作过程分为 3 部分[12]:Set(写"1")、Reset(写"0")和 Read. Reset(晶态到非晶态)和 Set(非晶态到晶态)每一种操作的电压以及时间延迟是不一样的. Set 操作需要施加一个宽脉冲(时间长)、弱电流(电压低),使得元件内部加热到 350℃,改变相变材料态为晶态;Reset 操作却需施加窄脉冲(时间短)、强电流(电压高),而元件内部材料温度达到 610℃. Read 操作则只需一个低电压,系统根据检测到的电流判断存储信息是"0"还是"1",Read 操作经过元件的电流很小,基本没有焦耳热产生,不会引起相变,不会造成数据破坏.

PCM 的主要特性可归纳为下面 7 个方面[13]:

1) 高存储密度. PCM 存储单元中的硫化物材料会根据电压大小的不同发生相变,从而呈现不同的电阻值. 根据这一特点,可使用不同的阻值区来存储字节组合,实现一个存储单元存储多个字节的多位封装技术. 另外,PCM 的制造工艺可达到 22 nm 级,意味着相同大小的 PCM 存储器包含更多的存储单元,可存储更多的数据.

2) 非易失性. PCM 通过存储单元内部硫化物的电阻值保存数据,硫化物的电阻值不会随着电压或电流的消失而变化,所以 PCM 是非易失(non-volatile)的存储介质,使得 PCM 可以不需要像 RAM 那样需要持续地供电来保留信息.

3) 字节可寻址. PCM 具有类似于 DRAM 的字节可寻址特性.

4) 可原位更新. PCM 具有位可变性(bit alterability),通过改变硫化物电阻值来表示"0"或"1",不需要擦除操作.

5) 低能耗. PCM 基于微型存储单元中硫化物相变后阻值变化来存储数据,没有机械转动装置,且具有低电压的特性(0.2 V～0.4 V),读写操作的能耗较闪存低,保存数据也不需要像 DRAM 那样定期刷新电流,是新一代的绿色存储器.

6) 读写不对称. PCM 的读取延迟约为 50 ns,比闪存缩短 1～2 个数量级,具有可媲美 DRAM 的读取带宽. 但是写延迟较大,虽然 PCM 不需要像闪存一样写前擦除,只需要"0"与"1"之间的转换,但是由于内部 Set 操作需要较长时间才能完成,与 DRAM 的写延迟仍有一定差距. 读写不对称性不仅表现在延迟上,同时也表现在读写能耗上. 由于每个 PCM 单元内部 Set,Reset 操作所需电压与 Read 操作所需电压存在较大差距,PCM 的读能耗与 DRAM 相近,但是写能耗却大于 DRAM.

7) 寿命有限. PCM 元件经过多次 Set 和 Reset 操作过程,其中相变材料界面会变得非常粗糙,导致器件单元失效,所以 PCM 写次数有限,一般可达 10^8 次,远远超过闪存的寿命. 对于基于 PCM 的芯片设计来说,有限的寿命是设计中必须加以考虑的问题之一.

表 1 给出了 PCM 与 DRAM 和闪存的对比. 从表 1 可看出,PCM 兼有 DRAM 和闪存的优点,具有很好的应用前景. 如果我们能够设计新的存储体系架构对 PCM 加以合理的使用,并根据相应的存储

Table 1 The Comparison of PCM, Flash and DRAM[14]

表 1 PCM 与闪存、DRAM 的对比[14]

Parameters	Cell Size/F^2	Read Latency/ns	Write Speed/Mbps	Endurance	Retention/years
DRAM	6—8	60	≈8 192	N/A	Refresh
PCM	4—6	200—300	≈800	10^6-10^8	10
NAND flash	2—4	25 000	19.2	10^4	10

架构设计或改进传统的操作系统和数据管理系统，有望最终实现高性能的数据存储与管理系统，满足大数据存储与管理的要求.

2 PCM 持久存储技术

相对于闪存而言，PCM 具有众多优点，有望成为新兴持久数据存储设备. PCM 一方面可以作为类似于固态硬盘(solid state drive, SSD)或磁盘(hard disk driver, HDD)的二级存储设备；另一方面，由于其优越的访问性能，PCM 在存储体系结构中的位置可以更靠近 CPU. 因此 PCM 持久数据存储管理的研究可以分为外存持久存储技术和主存持久存储技术.

2.1 基于 PCM 的外存持久存储技术

基于 PCM 的外存持久存储技术是将 PCM 看作是与 SSD 或 HDD 一样的二级存储设备，并以此为前提研究关于 PCM 的数据存储组织技术. 图 1 显示了 PCM 作为二级存储设备的基本系统架构[15]，其中 PCM 存储器通过 I/O 接口与主机相连，与 SSD 和 HDD 类似.

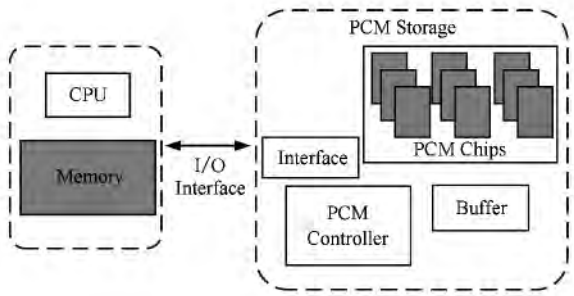

Fig. 1 The basic architecture of PCM storage.

图 1 PCM 二级存储基本架构

1) 基于 PCM 的外存文件系统

Lee 等人分别于 2012 年和 2013 年提出了基于 PCM 的 Shortcut-JFS[16] 和 On-demand Snapshot[17] 文件系统管理方法. 两种方法均利用 PCM 字节可寻址及其位可变性的特点，以 block 为主要管理单位. Shortcut-JFS 文件系统的数据块根据更新比特范围选择是“块日志”还是“差分日志”，其中差分日志(differential logging)技术只将与原始数据不同的字节写入日志区；使用原位校验(in-place checkpoint)技术在检查点到来时只需修改一定的指针数据，即可将日志区中的块数据直接转换为数据块，避免日志数据的大量复制，从而减少大量的 PCM 写操作，实验结果表明 Shortcut-JFS 比 ext3 文件系统平均提升了 40% 的性能；而 On-demand Snapshot 则是一个版本控制文件系统，打破了传统的数据更新从数据节点到根节点都需异地更新的情形，原位更新索引树结构中的指针信息，并通过一个包含时间戳的快照链表将原始数据及索引树中相应的原始指针信息链接起来，以便当需要访问前一版本的文件数据时构造快照树，该方法的 I/O 吞吐率比典型的版本控制文件系统 ZFS 平均提高了 144%. 随后 Lee 等人又详细分析了文件数据块更新前后的变化情况，实验结果显示平均超过 35% 的写操作对原数据块改动仅仅不到 10%，从而提出了可靠的双树文件系统(dual-tree file system, DTFS)[15]，使用每个数据块的前一版本存储的数据来执行该数据块新的更新操作，但是该系统不能应对数据页的新旧版本存储单元由于频繁更新而快速磨损的问题. 另外，PCM 的访问延迟与 DRAM 接近，与二级存储设备差异很大，当主存未命中时访问二级存储设备需要经过 I/O 调度、文件系统等一系列程序，该过程造成的延迟对于传统慢速二级存储设备来说微不足道，但对 PCM 等较快速的存储设备所造成的影响却极大.

2) 空间管理

由于 PCM 作为二级存储与 SSD 有诸多共同点，已有的 SSD 内部算法研究如闪存转换层算法(flash translation layer, FTL)、磨损均衡算法等对 PCM 存储器内部算法的设计有一定的借鉴作用，如 Choi 等人在 2012 年提出类似于 FTL 的相变存储转换层算法 (PRAM translation layer, PTL)[18]，通过有效的空间管理和分配技术解决基于 PCM 的存储子系统中的磨损均衡问题. 磨损均衡技术在过去基于闪存的研究中得到了长足发展，然而基于闪存的磨损均衡技术并不能直接用于 PCM 的管理，其主要原因是闪存具有写前擦除的特性，磨损均衡的基本处理单位是一个擦除块大小(包含多个页面)，常常存在“写放大(write amplification)”的问题，而 PCM 可以在更细粒度的地址空间上进行，并且可原位更新无需擦除. Im 等人于 2014 年提出了面向磨损均衡的差异型空间分配方法(differentiated space allocation, DSA)[19]，针对二级存储 PCM，摒弃已有的基于闪存的磨损均衡技术，采用多粒度的数据管理方式，为频繁更新的数据分配更多的存储空间. DSA 的主要管理单位为段(segment)，DSA 维护一个从逻辑段号到物理段号之间的映射表，而一个段包含多个块(chunk)，DSA 对每个块的写操作计数，若某个块被频繁更新，使得写次数超过一定

的阈值,则在保留段池(reserved segment pool)中为该块分配一个新空间,直到保留段池的段被耗尽需重新选取保留段时,才将其中的块数据重新写回原来段的相应存储位置,从而保证段内的各块之间的磨损程度能被控制在一定的范围内;而保留段被耗尽后随机从普通段空间中选择新段使之成为保留段的做法,对段与段之间的磨损均衡起到一定的作用.与一般的基于交换的磨损均衡算法相比,DSA 这种多粒度的空间管理方法大大减小了写放大比例,但维护每个块的更新计数以及其他映射表等元数据信息需要较大时间和空间开销.

2.2 基于 PCM 的主存持久存储技术

1) 基于 PCM 的主存文件系统

由于 PCM 拥有众多与 DRAM 相似的特性,为最有效地利用其优点,早在 2009 年,Condit 等人就提出可字节寻址的主存级文件系统原型(byte-addressable persistent file system, BPFS)[20]. 它是基于 PCM 直接连接在内存总线上,接收来自 L2 高速缓存层数据的 DRAM/PCM 混合主存架构实现,但 PCM 作为持久存储设备,以文件系统的管理方式来操作. 为了解决 PCM 数据持久化过程中的数据一致性问题,BPFS 改进硬件设计,提供 64 b 原子写操作以及 L2 高速缓存层数据写回时写有序等功能,并利用 DRAM 来存储复杂数据结构,结合 PCM 字节可寻址、非易失等特性,设计了一种简单的树形文件存储架构,由 3 种文件类型组成:索引节点文件、目录文件以及数据文件. 该系统管理固定大小的数据块,并提出短路影子页方法(short-circuit shadow paging, SCSP),对小于或等于 64 b 的数据进行原位更新,如图 2(a)所示;对超过 64 b 的数据更新则采取部分写时复制的方式(如图 2(b)所示),从更新的数据节点开始向根节点回朔,对所有大于 64 b 的更新数据都需复制到新分配的块中进行更新操作,直到回朔过程中某个块的数据更新可保证 64 b 原子写操作;得益于硬件设计的改进,SCSP 方法保证了数据的一致性.

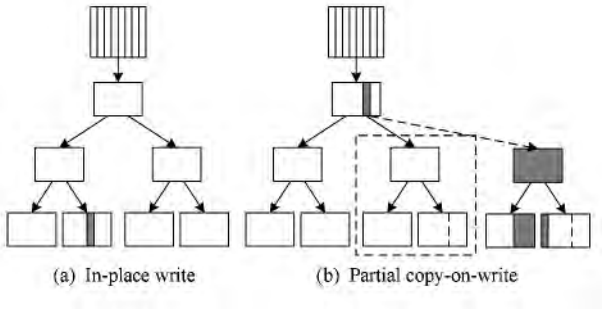

Fig. 2 Approaches to updating the BPFS file.

图 2 BPFS 文件数据更新方法

2) 基于 PCM 的软件系统

Coburn 等人提出的 NV-heaps (nonvolatile memory heaps)[21]是一个轻量级高性能持久对象系统,它基于文献[20]的硬件设计结构,为程序员提供了包括对象、指针、主存分配等原语,实现指针安全(防止非易失存储器上的指针指向易失性存储介质的情况发生)、ACID 事务处理、传统 API 服务、高性能及可靠的功能,以便在发生系统崩溃、断电或其他常见错误时保护非易失存储器上数据的正确性. Volos 等人提出的轻量级主存系统 Mnemosyne[22] 在 PCM 设备能保证 64 b 原子写的假设下,修改系统库函数,为程序员设计了一个直接访问非易失存储器的接口,对于开发过程中想要持久存储的数据,只需用特定的数据类型 pstatic 声明即可. Mnemosyne 提供了一种快速的存储简单持久数据的方法.

2.3 PCM 持久存储技术小结

表 2 显示了基于 PCM 的持久数据存储组织主要算法对比:

Table 2 The Comparison of PCM-Based Persistent Storage Technologies

表 2 基于 PCM 的持久存储技术对比

Algorithms	Type	PCM Location	Key Technology	Other Features
Shortcut-JFS	Journaling File System	Second Storage	differential logging, in-place checkpointing	Reduce writes
On-demand Snapshot	Versioning File System	Second Storage	Timestamp-based snapshot list	Reduce writes, but need time to reconstruct snapshot tree
DTFS	File System	Second Storage	dual-tree, shadow version	Reduce redundant writes, but the bits frequently updated will be badly worn
DSA	Space Management	Second Storage	multi-granularity, reserved segment	Wear leveling, but counting the write operations spends time and space
BPFS	Memory File System	Memory	SCSP, 64 bit Atomic updates	Need hardware support

表 2 中显示,多粒度是基于 PCM 的持久数据存储技术的主要思想,以固定大小的块为主要管理单位,但数据更新则更细粒度地执行,充分利用 PCM 非易失、可字节寻址及原位更新的特点,此类研究的主要目的是减少 PCM 的写次数和写操作范围.

虽然大数据处理的数据量巨大,但有价值的数据是有限的,利用 PCM 持久存储重要的数据信息,有助于提高大数据处理性能. 但是传统操作系统 I/O 体系结构设计不能最大程度地发挥 PCM 的低访问延迟优势;另一方面,传统的主存管理机制针对的是易失性 DRAM,其中的页面置换算法并不适用于持久存储设备的数据管理,传统分配策略对 PCM 的磨损也不友好. 研究者虽已开始寻求新的存储管理方案,如对 DRAM 和非易失内存分别采取不同的管理方法[23-24],提出基于对象的存储级主存(storage class memory, SCM)原型系统[25]等,但仍未有集快速、持久、耐用等诸多优势于一身的数据存储管理方案出现. PCM 持久存储设备无论是与磁盘、SSD 等二级存储设备同级还是直接与内存总线相连,都对传统系统架构设计提出了挑战,对大数据的存储和管理有着重要的意义.

3 基于 PCM 的主存系统

在大数据时代,人们对大容量的主存系统需求越来越大,由于 DRAM 不能长久存储数据,需定期刷新电流,所消耗的能量在整个主存系统能耗上占很大比例,所以低能耗的 PCM 作为主存有望解决大数据的能耗和性能等问题. 近年来国际上一些存储公司和研究机构不仅致力于研究提高 PCM 寿命的方法[26-29],也逐渐开始从主存系统架构的角度来研究相变存储器[14,30-32]. 基于 PCM 的主存系统架构大致分为 3 种[33]:1) PCM 替代 DRAM;2) DRAM 和 PCM 同级混合主存;3) DRAM 作为 PCM 缓冲.

3.1 PCM 替代 DRAM

根据 PCM 的特性以及对大容量主存的应用需求,PCM 完全替代 DRAM 作为主存存储器是最早讨论的架构方法,如图 3 所示. PCM 相对于 DRAM 而言,最大的不同之处在于只有有限的寿命,因此这种架构下如何延长 PCM 的寿命成为研究的热点. 关于解决 PCM 寿命的问题,当前主要研究重点集中在磨损均衡以及如何尽量减少对 PCM 的写操作上[34-37].

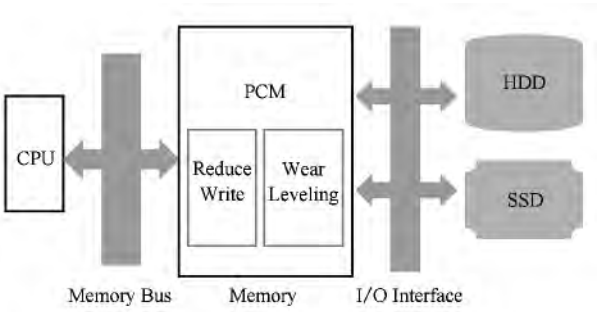

Fig. 3 PCM-based memory system architecture[35].

图 3 基于 PCM 的主存系统架构[35]

1) 磨损均衡

磨损均衡算法离不开数据交换、空间分配等操作,基于闪存的磨损均衡算法的交换、分配操作是由闪存转换层完成的;而 PCM 替代 DRAM 作为主存架构下的 PCM 物理空间分配、回收等操作都是由操作系统通过内存管理单元控制;因此该主存架构下的 PCM 磨损均衡算法需与操作系统内存管理相结合. Chen 等人在 PCM 作为主存的前提下,设计适用于 PCM 的页面管理方案,提出了基于桶和基于数组的两种磨损均衡方法[34]. 作者首先通过页面磨损计数器记录每个页面的写操作次数,其中基于桶的磨损均衡算法维护两个桶链表,分别管理空闲页面和非空闲页面,而同一个桶所包含的页面的磨损程度相近,如图 4(a)所示. 每次页面空间分配以及页面交换需要的空闲页面从写次数最少(即磨损最轻)的桶(即图中的"基桶"b)中获得,若空闲基桶为空,则从非空闲链表的基桶选取页面,将其中存储的数据转移到磨损最严重的空闲页面中. 而基于数组的磨损均衡算法则采用更简单的数据结构——一个指针和两个全局计数器来管理 PCM 物理页面,当某个页面的写次数超过一定阈值时,将其与邻近中心指针的 K 个页面中最年轻的页面进行交换,如图 4(b)所示.

2) 减少 PCM 写操作

在减少 PCM 写操作的研究上,主要方法是利用 PCM 可字节寻址的特点,减少写操作覆盖范围,数据更新只修改部分比特位. PCM 读操作的能耗和时间开销都比写操作少很多,因此写前读对整个写操作的延迟和能耗影响较小,通过这种方式对要写入的数据与原数据比较,仅修改有变化的比特位 0,可大大减少写操作涉及的存储单元数;从实验结果[35]可看出,该方法可延长 SLC PCM 的 4.5 倍寿命,对 MLC PCM 也延长了 3 倍以上的寿命. Cho 等人提出 Flip-N-Write[37] 方法对每组存储单元都增加一

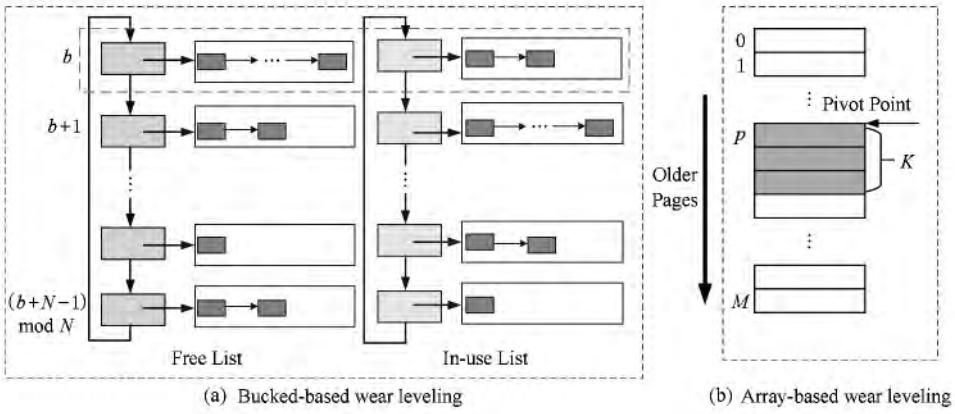

Fig. 4 The structure of two wear leveling algorithms.

图 4 两种磨损均衡算法数据结构

个比特,更新时先将旧数据读出与新数据进行比较,当该组存储单元中有超过一半的位变化时,增加的比特位被置为"1",此时反转没有发生变化的位,读取该组存储单元的数据时需要对所有位反转才能获取正确的数据;Flip-N-Write 方法保证每次数据单元的更新位数都不大于其位宽的一半,达到减少写的目的;实验显示,存储单元位宽从 2～32 b 变化,Flip-N-Write 方法可减少 56%～63%的位更新.

作为主存设备,性能仍然是计算机系统最为关心的问题. PCM 的读写延迟仍然大于 DRAM,因此除了延长 PCM 使用寿命以外,也有研究以提升 PCM 读写速度为目的优化方法——写取消和写中止[38];设备内部正在进行的写操作可以被取消以响应刚到达的读请求,降低读延迟,而当写操作处于迭代写阶段时可以被中止,等待执行完读操作后再回复到原状态继续执行迭代写过程. 实验证明,通过写取消和写中止方法,可消除 75%读请求引起的延迟增长,并且提升 46%系统性能.

综上所述,针对 PCM 替代 DRAM 的研究主要集中在 PCM 存储器内部结构以及算法设计上,以磨损均衡和减少 PCM 写操作为目标. 主要问题在于缺乏对运行在主存 PCM 上的写优化算法研究,对 PCM 替代 DRAM 带来的性能影响还缺少理论分析模型和实验研究. PCM 替代 DRAM 的主存架构虽然可以防止大数据存储与管理过程中数据丢失的情况发生,但由于 PCM 读写延迟大于 DRAM,该架构对计算机性能影响比较大,而大数据处理大量针对的是海量数据流(data stream),数据读写频繁,因此该结构容易降低 PCM 的使用寿命.

3.2 PCM 与 DRAM 同级混合主存

PCM 的读写性能虽比磁盘、闪存等设备要好很多,但和 DRAM 相比仍有不足,因此出现了 DRAM/PCM 混合主存架构,其中 DRAM 与 PCM 作为同级设备,共同承担主存数据存储管理任务是混合主存研究的重要分支之一,如图 5 所示:

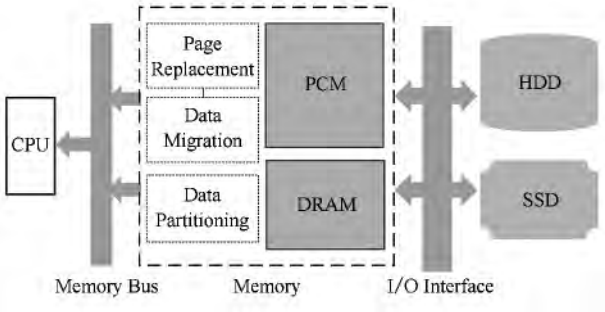

Fig. 5 PCM/DRAM hybrid memory architecture.

图 5 PCM/DRAM 同级混合主存架构

基于该架构的研究工作主要是缓冲区管理方法研究,包括页面分配、置换和不同存储介质之间的数据迁移等. 与传统主存系统不同,该架构下每种主存介质的性质差异较大,导致在主存页面管理上必须将不同存储介质的页面区别对待,因此在这类研究中合适的数据分类显得尤为重要. 除了采用传统主存数据访问频繁性、冷热的划分方法[39-42]以外,针对 PCM 的特性,已有研究工作热衷于探讨数据读写倾向性划分方法[43-47].

3.2.1 数据冷热划分

数据的访问频率和最近访问间隔共同反映了数据的冷热性,热数据表示在最近一段时间内访问频率高,相反,冷数据在近期不太可能被频繁访问. 考虑到 DRAM 读写延迟较 PCM 低、PCM 寿命有限等因素,基于 DRAM/PCM 混合主存的缓冲区管理算法通常将热数据存放在 DRAM 中,冷数据存放在 PCM 中.

1) CLOCK-DWF

Lee 等人通过大量的实验数据分析,认为页面写操作的时间局部性和频繁性更适用于近期的写访问预测,因而提出了混合主存置换算法 CLOCK-DWF(CLOCK with dirty bits and write frequency)[40]. 该算法根据请求类型为未命中的数据分配物理空间(读请求数据存储到 PCM,写请求数据存储到 DRAM 上);当写请求未命中或者在 PCM 上命中时,会将数据载入到 DRAM 上;若 DRAM 满时会将 DRAM 上写冷的页面换出到 PCM 中,而当 PCM 满时,则根据传统的 CLOCK 算法替换策略替换页面. 该算法使所有的写请求均在 DRAM 上执行,DRAM 上页面替换兼顾了写频繁性和局部性,在不到 8%的性能损失情况下能有效减少 PCM 上的写. 然而该算法最大的不足之处在于,在 PCM 上页面更新时,必须将它迁移到 DRAM 上执行,若此时 DRAM 已满,则必须选取一个 DRAM 的冷页面与之交换,此时,PCM 仍然还要承受一次写操作;如接下来的写请求命中刚交换的冷页面,又得重新将它换入 DRAM,形成"颠簸效应". 因此在处理操作页面不集中且写密集的数据集时,将引起 DRAM 与 PCM 之间大量的数据迁移操作,增加额外读写操作.

2) RaPP

文献[39]提出了一种基于排列的页面管理方法(rank-based page placement, RaPP),使用改进的多队列置换算法(multi-queue, MQ)管理所有主存页面. MQ 算法主要思想是根据页面不同的访问频率决定其在主存中保存的时间长短,算法维护若干个队列,同一队列里面的页面访问频繁度或热度相似,但不同队列页面热度不同,队列级别越高其中的页面访问频繁度越高. 而改进的 MQ 算法使用时间参数来控制页面从高级队列向低级队列调整,但若 DRAM 页面连续两次降级,且期间没有被访问,则会被踢出队列,成为与 PCM 页面交换的候选;若某个被频繁访问的 PCM 页面所属队列级别达到一定阈值,则会触发迁移操作. 其中 MQ 队列的更新、数据页面迁移等操作借助主存控制器(memory controller, MC)完成,以达到性能最优. 然而出于磨损均衡的目的,RaPP 算法每一个 PCM 页面向 DRAM 迁移都会引起 3 个页面存储位置的变化,会增加额外写操作次数.

3) APG

自适应页面分组方法(adaptive page grouping, APG)[41]是一种数据迁移算法,针对的是在数据迁移中,若每次迁移数据量少(一次一个页面)可能会频繁触发数据迁移操作[39-40,42],处于迁移条件边缘的数据可能在两种介质间来回变动,增加不必要的读写操作问题而设计. 在 Linux 操作系统内存管理中,伙伴系统是采用一次性分配一大段连续空间的分配模式,Shin 等人[41]对一定的数据集进行了实验分析,证明在这种情况下主存页帧物理地址邻近的页面访问次数相近,所以他们提出的 APG 将物理邻近的页面聚类为一个组,根据每个页面访问统计信息计算读写热度,用组内页面热度平均值判断组页面冷热程度,触发迁移条件后一次性迁移整个组页面;实验显示,APG 可有效减少迁移带来的额外读写操作,对系统的性能影响也极小,取得最大 42%的能耗降低.

3.2.2 数据读写倾向性划分

一个页面的读操作多,写操作少,则被认为具有读倾向;若一个页面大部分的访问类型都是写访问,则认为它是写倾向页面. 本文第 2 节已提到,PCM 读延迟可与 DRAM 读写延迟相娉美,但 PCM 读写不均衡,且写操作会缩短 PCM 使用寿命,因此出于系统性能和 PCM 寿命方面的考虑,越来越多的研究开始尝试对数据进行读写倾向性划分,从而使读倾向的数据存储在 PCM 中,写倾向的数据存储于 DRAM 中.

1) LRU-WPAM

Seok 等人[44-45]提出的基于预测和迁移的页面置换算法(LRU with prediction and migration, LRU-WPAM),对每个页面进行监控,首先通过式(1)计算每个页面的权值,页面权值大小代表其读写倾向性:

$$W_{cur} = \alpha \cdot W_{pre} + (1-\alpha) \cdot RT, \quad (1)$$

其中,RT 代表页面的访问类型("1"表示写,"-1"表示读). 分别将 DRAM 和 PCM 的页面划分到读倾向链表和写倾向链表中,如图 6 所示,当 PCM 中写倾向页面的权值超过一定阈值时则将其迁移到 DRAM 上,若 DRAM(或 PCM)空间已满则从 DRAM 读倾向队列末尾选择页面换出;DRAM 向 PCM 迁移操作与上述过程相似,只是迁移的是 DRAM 中读倾向的页面. 算法还维护一个 LRU 链表管理所有 DRAM 和 PCM 的页面,用于页面替换算法中选择被替换的页面. 实验结果表明,该算法在保证一定命中率的情况下可降低 PCM 写次数,最大达到 52.9%. 但该页面置换算法若迁移操作触发时,迁移目的地没有空闲的存储空间供迁移数据存储,则会从相应

的读链表或写链表中选取页面换出主存,命中率可能因此而降低.

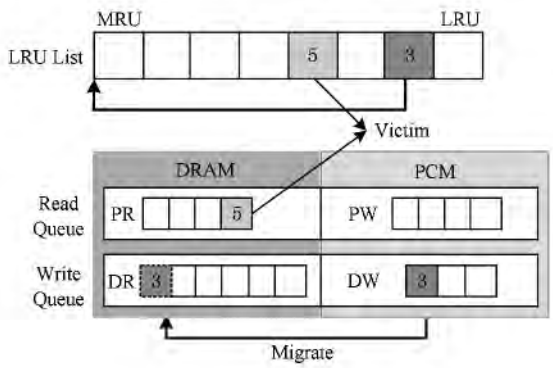

Fig. 6 Example of LRU-WPAM replacement algorithm.

图 6 LRU-WPAM 置换算法示例

2) MHR-LRU

针对 LRU-WPAM 算法命中率降低的问题,Chen 等人在 2014 年提出页面替换算法 MHR-LRU(Maintain-hit-ratio LRU)[46].该算法由一个总的 LRU 链表管理所有 DRAM 和 PCM 页面,只用于页面置换时选择换出的页面;同时该算法还维护一个 DRAM 写链表(DRAM write-aware LRU,DWL)来管理 DRAM 的页面,只有当 DRAM 页面写命中时才会引起链表中页面位置的改变,即命中的页面移到 DWL 链表 MRU 端,因此位于 DWL 链表 MRU 端的页面写更热,相反 LRU 端的页面写更冷,而触发迁移操作时会选择将 DWL 链表 LRU 端的页面从 DRAM 迁移到 PCM.由于该算法采取 CLOCK-DWF 相同的页面分配策略,根据读写访问类型为新页面分配存储空间,因此可能出现为新写操作页面换出一个 PCM 空间的页面,此时就会触发 DRAM 到 PCM 的页面迁移,从而在保证命中率的同时减少 PCM 上的写操作.然而该算法两种存储介质之间的移动是单向的,即只有 DRAM 向 PCM 迁移页面,而 PCM 上写倾向的页面不能迁移到 DRAM 中,因此不能达到有效减少 PCM 写的目的.

3) APP-LRU

与上述算法以数据读写操作类型作为页面分配依据不同[40,44-46],APP-LRU(access-pattern-prediction-based LRU)[47]算法维护了一个历史元数据信息表,存储的是历史上访问过的磁盘页面的读写访问比例;当这些页面被再次载入内存时,算法可根据它的历史读写访问比例信息判断该页面将来的读写倾向性,从而为读倾向的页面分配 PCM 空间,为写倾向的页面分配 DRAM 空间;由于算法页面置换与 MHR-LRU 相同,由一个 LRU 链表控制,因此也会出现换出的页面空间与新数据的读写倾向性不匹配的情况,此时则需要 PCM 与 DRAM 之间数据进行迁移,以便获取与读写倾向性相匹配的存储空间.该算法的目的在于保证命中率的同时使用最少的迁移操作来达到减少 PCM 写操作的目的.由于子链表 DRAM-List 或 PCM-List 中的页面是以局部读次数或局部写次数来排序的,没有时间衰减机制,可能造成冷页面迁移的情况发生.另外该算法对每个页面的读写操作计数会增加时间和空间开销.

3.2.3 实验对比

图 7 显示了基于 DRAM/PCM 同级主存架构的 CLOCK-DWF 算法相对于传统纯 DRAM 主存架构的时间开销(包括内存访问时间、磁盘 I/O 等),混合主存架构下的 DRAM 与 PCM 的容量比例为 1∶9,其中横坐标表示主存总容量大小,数值表示主存大小与数据集中涉及到的所有数据大小比值,比如 100%表示内存能容纳数据集中涉及到的所有数据.图 7 显示混合主存架构相对于传统 DRAM 主存架构的性能损失低于 8%,在某些情形下混合主存置换算法命中率高于传统 DRAM 主存架构下的置换算法,导致时间开销优于后者;图 7(a)和图 7(b)分别是 PCM 不同读延迟情形下的时间开销,但是两者之间的时间开销接近,这意味着影响系统的性能的主要因素是磁盘 I/O,不是 PCM 的读延迟.因此,PCM 作为主存并不会影响系统的性能;相对于 DRAM,廉价 PCM 作为主存具有非常高

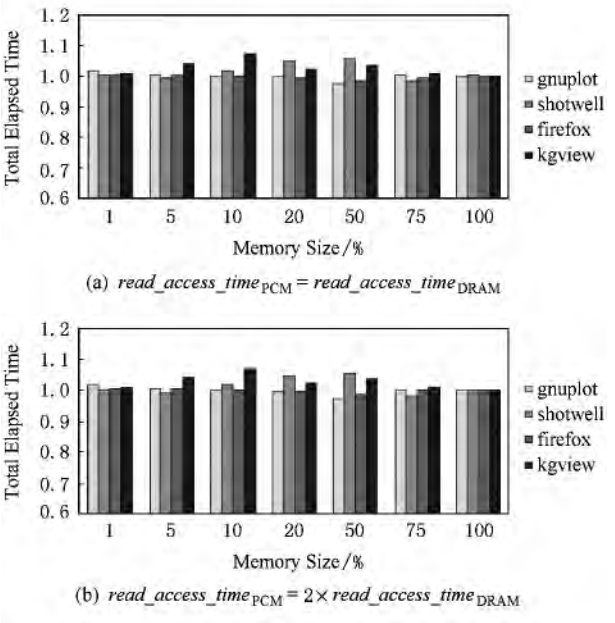

Fig. 7 Elapsed time of CLOCK-DWF[40].

图 7 CLOCK-DWF 时间开销[40]

的性价比,且可以扩大主存容量提高命中率,反而能提升系统性能.

表 3 列出了目前 DRAM/PCM 同级混合主存架构下的主要内存管理算法,这类算法除了不降低系统性能以外,另一设计目标是减少对 PCM 的磨损. 本文实现了 LRU, CLOCK-DWF, LRU-WPAM, MHR-LRU 和 APP-LRU 等替换算法,以对比它们之间的差异. 实验设置:仿真系统采用统一的编址模式,DRAM 占用低端地址空间,PCM 占用高端地址空间,数据页大小设置为 2 KB,DRAM 与 PCM 的空间大小比例设置为 1∶4. 实验所用数据集包括 6 个仿真数据集(T1982, T1955, T5555, T5582, T9182 和 T9155)和一个真实数据集 OLTP. 每个仿真数据集都代表了不同的读写比例以及访问局部性,例如数据集 T9182 表示该数据集的读写操作比例为 90%/10%,以及 80%的操作请求集中在 20%的页面上. 真实数据集 OLTP 截取自一个银行系统 CODASYL 数据库访问,包含 470677 个读操作和 136 713 个写操作. 实验主要分析 PCM 的写次数和算法运行过程中的迁移次数,结果如图 8 所示.

Table 3 Comparison of PCM/DRAM hybrid Memory Management Policies

表 3 PCM/DRAM 同级混合主存管理策略对比

Algorithms	Data Partitioning	Key Technologies	Main Features
CLOCK-DWF	Cold or hot	CLOCK-based, page replacement, migration	advantage: Effectively reduce the write operations on PCM; drawback: pages with poor write locality need lots of page migrations
RaPP	Cold or hot	MQ-based, page replacement, migration, hardware support	advantage: page migration and list management needs Memory Controller hardware support, but spends less time; drawback: each migration will cause three pages' location change, increase read and write operations
APG	Cold or hot	Page group, migration	advantage: avoid frequent data migration and effectively reduce the extra reading and writing; drawback: cannot distinguish adjacent pages that have different access frequency, may migrate cold pages
LRU-WPAM	Read or write tendency	LRU-based, page replacement, migration	advantage: estimate read or write tendency based on the weight of page; drawback: may reduce the hit ratio because of migration
MHR-LRU	Read or write tendency	LRU-based, page replacement, unidirectional migration	advantage: reduce PCM write count under the premise of guarantee hit ratio; drawback: cannot migrate write-tendency page from PCM to DRAM
APP-LRU	Read or write tendency	LRU-based, page replacement, migration	advantage: reduce PCM write count by a small amount of migration and guarantee hit ratio; drawback: storing and updating metadata increase the time and space overhead, cold page migration may occur

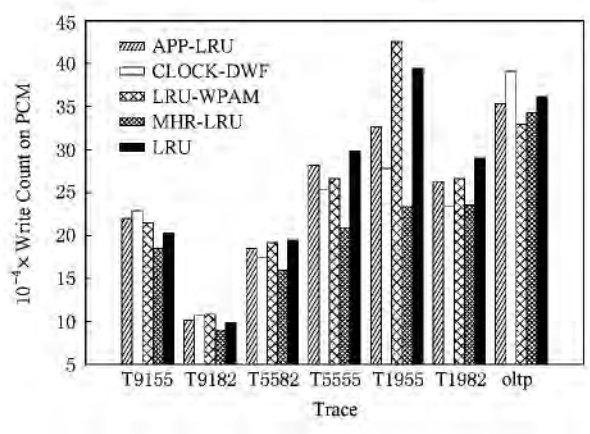

Fig. 8 Performance of buffer replacement algorithms.

图 8 缓冲区替换算法性能对比

基于 PCM/DRAM 同级主存的缓冲区替换算法均是以减少 PCM 写次数为目的,从图 8 可以看出,MHR-LRU 算法在各种类型的数据集上都能取得较好实验效果. 与 LRU 算法相比,基于 PCM/DRAM 同级主存的缓冲区替换算法因为考虑了 PCM 的读写不对称特性,对数据进行了读写冷热或倾向性划分,在多数情况下均能有效减少 PCM 的写次数;但对于数据集 T9155 和 T9182,各类算法优势不明显,这是因为这类数据集写操作只占 10%, PCM 上的写多数是由页面未命中时从磁盘读入 PCM 造成的.

3.2.4 PCM 与 DRAM 同级混合主存小结

总体而言,目前的 PCM 与 DRAM 同级主存架构

的研究以页面置换算法为主,虽然主存架构与传统不一样,但命中率仍然是置换算法优劣的首要评判标准,因此既能不降低或提升命中率又能减少 PCM 写次数是该类研究的主要目的.另外,该架构下的页面管理算法往往还涉及 PCM 与 DRAM 之间的页面迁移,迁移时机的把握以及如何选取迁移页面是主要关注点.但在基于 PCM 的大数据存储与管理中,数据价值评定却是重要的研究内容,因为海量数据中的高价值信息密度低,如何从中快速识别出高价值信息并存储在 PCM 上,以便在数据处理过程中被快速获取是主要难点.

3.3 DRAM 作为 PCM 缓冲

DRAM 作为 PCM 缓冲的二级主存系统架构是混合主存研究的另一个重要分支,如图 9 所示.它的出现一方面为系统提供大容量的存储空间,另一方面解决了 PCM 本身读写延迟方面的缺陷带来的系统性能下降问题.

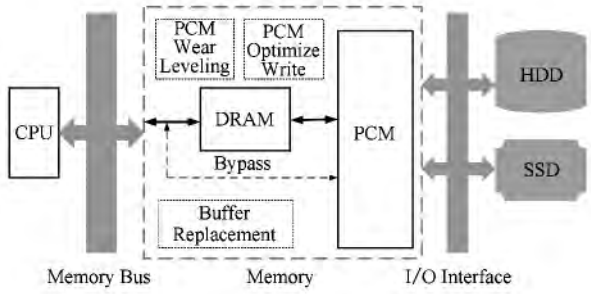

Fig. 9 DRAM-cache based PCM memory architecture.

图 9 DRAM 作为 PCM 缓冲的主存系统架构

Qureshi 等人[14]指出,PCM 存储密度高,虽然有着较大的存储容量,但是它的存取速度仍然比 DRAM 慢,且寿命有限,因此 PCM 不会完全替代 DRAM 而单独作为主存.他提出了一种基于 PCM 的混合存储结构,以 PCM 作为主存,一个较小的 DRAM 作为 PCM 的缓冲,使用 Lazy-Write Organization 方法延迟对 PCM 的写操作,另外用 Line-Level Write 方法减小页面写操作的粒度,以达到减少对 PCM 写次数的目的.研究实验结果表明,这种混合存储结构可在仅 13%的额外存储开销的情况下延长寿命 3~9.7 年.Qureshi 等人还基于上述架构提出了一种磨损均衡算法 Start-Gap[48],避免传统磨损均衡算法需要跟踪每个单元的写次数带来大量时间和空间开销的问题.Start-Gap 使用代数映射的方式完成逻辑地址到物理地址的映射,利用 Start 和 Gap 寄存器来保存用于计算地址映射的相关数据,Gap 始终指向空间中特定的未被使用的行(line),系统定期将与 Gap 相邻的行数据迁移到 Gap 所指的空行中.该方法无需大量的逻辑地址到物理地址的映射表,对时间和空间的开销都较小,但相邻多个行均出现写集中情形时,通过 Start-Gap 算法会导致一个写集中的行迁移后仍然有写集中的问题,Qureshi 等人在 Strat-Gap 基础上提出随机化 Start-Gap 算法,解决了这一难题,大幅提高了 PCM 的使用寿命.

文献[49]亦使用 DRAM 缓冲 PCM 数据,并维护基于行的 LRU 算法执行 DRAM 中的行数据替换操作,同时 DRAM 上还维护着记录每个 PCM 行写次数的 Hash 表,以便快速确定磨损严重和磨损较轻的页面,以便定期对页面内容进行交换,该作者随后对其工作作了扩展与改进[50],将 DRAM 上的干净数据行与脏数据行分开管理,并提出了基于缓冲的多数据交换移位策略.上述研究利用 DRAM 来缓冲数据;一方面是为了结合 DRAM 来设计适合 PCM 的磨损均衡算法,另一方面,则是为了减少 PCM 写.而减少 PCM 写,除了上述以减少总的 PCM 写次数为目的的内容外,还包括以缩小写操作覆盖范围为目的的研究工作,对于后者通常的解决方案是利用 PCM 位可变特性,以更细粒度的单位进行数据更新,免去不必要的数据重写,如 Lee 等人提出的 row buffer 方法和 partial write 方法[51];Ferreira 等人提出的 Read-Write-Read 和 page partitioning 技术[52].

除了以延长 PCM 使用寿命为目的的研究外,利用 PCM 的高存储密度和无需提供刷新电流的低能耗特性方面,Park 等人[53]以降低能耗为目标,设计出 DRAM 与 PCM 组成的二级主存架构模型,抓住 DRAM 中被访问的行在一定时间内不需要刷新电流的特点,赋予每个行一个随着时间衰减的"C"域,数据被访问时赋值,衰减至 0 时行数据被回收,通过这种方式减少对 DRAM 行的刷新电能消耗;同时在两级存储之间使用 DRAM Bypass 技术,对于 DRAM 未命中的读请求页面,首次访问时不直接进入一级缓冲 DRAM 中,更进一步降低 DRAM 刷新能耗.实验数据表明通过这种方式,在低微的性能损耗下使得能耗降低 23.5%~94.7%.

DRAM 作为 PCM 缓冲的主存架构通过 DRAM 保证系统性能,隐藏 PCM 读写速度慢的特性,并通过 PCM 扩展主存容量提高命中率,减少磁盘 I/O,在大数据存储与管理中对性能提升和能耗降低方面都有

较大促进作用,基于该架构的研究多以行为粒度单位,细粒度的管理策略带来的首要问题就是维护元数据的空间复杂度高.另外,DRAM 的存储空间通常小于 PCM,若任何数据类型都经过 DRAM 缓冲,将会导致 DRAM 与 PCM 之间数据交换增多,无论是系统的性能还是 PCM 的寿命都会受到影响.因此在 DRAM 作为 PCM 缓冲的存储架构下,有选择地缓冲数据,设计适当的缓冲区管理算法和适合 PCM 的数据管理方法,仍是当前的主要关注点.

3.4 基于 PCM 的主存系统小结

综上所述,目前 PCM 的主存系统研究仍然以延长 PCM 寿命、减少 PCM 写或磨损均衡算法为主,但基于 PCM 的性能提升、降低能耗的方法也不可忽视.近几年,面向闪存的研究发展迅速,混合存储研究早已成为闪存存储领域的一个重要研究内容,主要研究工作集中于基于磁盘和固态硬盘 SSD 的混合存储,其中"主存-SSD-磁盘"模型[54-55]和"主存-SSD/磁盘"模型[56]中 SSD 和磁盘的角色与 DRAM/PCM 混合主存架构中 PCM 和 DRAM 的角色具有共同点,因此基于磁盘和 SSD 的混合存储研究成果对基于 PCM 的混合主存研究具有一定的指导和借鉴作用.当今大数据时代,闪存、PCM 等新型存储介质与 DRAM、磁盘等传统存储介质共存,并且在较长时间内并不能完全替代传统存储介质的地位,所以适合于高效能大数据存储和管理的新型存储架构是基于分层存储的多介质混合存储架构,相应地,多介质的存储架构设计及算法改进方法也变得至关重要.

4 PCM 应用研究

目前,由于 PCM 容量还相对较小,还没有真正完全基于纯 PCM 的存储设备出现,但已经有一些工作在相关领域中应用 PCM 进行了探索性研究.

本节主要讨论 PCM 在特定领域中的应用研究,包括 PCM 在混合存储设备中的应用(如 4.1 节所述)、PCM 在传统 DBMS 中的应用(如 4.2 节所述)、PCM 在实时与并行系统中的应用(如 4.3 节所述)、PCM 在嵌入式系统中的应用(如 4.4 节所述).

4.1 PCM 在混合存储设备中的应用

闪存等二级存储设备为大数据处理提供海量数据存储的功能,其技术的进步既有效延长设备的使用寿命,也为大数据处理带来高存取性能.闪存具有写前擦除的特性,因此异位更新是目前 NAND 闪存算法研究的主流思想.但是异位更新使得每次数据更新都会伴随着元数据的更新,而实际发生变化元数据大小远远小于 NAND 闪存的写操作粒度,从而带来不必要的磨损和开销.考虑到 NAND 闪存本身特性的限制以及随机写操作对 NAND 闪存的性能影响,研究者引入 DRAM 来缓冲 NAND 闪存的读写信息(包括普通数据和元数据);但是 DRAM 是易失性存储设备,当掉电、系统崩溃等突发情况发生时不能保证数据的可靠性.因此近几年已有研究开始在 NAND 闪存的研究中引入 PCM,主要解决元数据信息的持久存储与更新问题.图 10 显示了 PCM 与 NAND 闪存混合的基本存储架构:

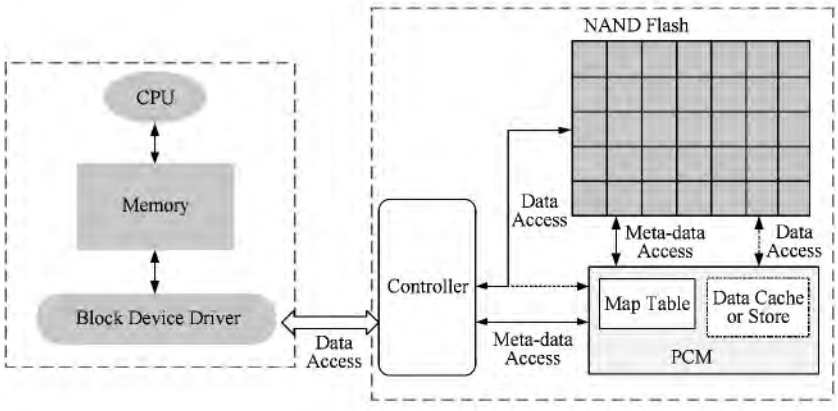

Fig. 10 PCM/NAND Flash hybrid storage architecture.

图 10 PCM 和 NAND 闪存混合存储架构

PCM 存储的元数据包括文件系统索引信息[31,57]、闪存 FTL 逻辑页-物理页的映射表[30,58-60]、描述页面状态的位图信息[29]等.PCM 与闪存混合旨在减轻闪存的磨损和提高性能,但是 PCM 也是寿命有限的存储设备,因此在基于 PCM 的闪存算法设计中,基本都会采取一定措施来降低 PCM 磨损,

如在为新写请求分配空间时,考虑到原映射关系的变化情况,会选择引起映射记录位改变最少的页面号分配(PCM-FTL[58]),或者为频繁更新的映射表信息添加小型缓冲区(cooling-pool[59]),或者延迟执行闪存数据块的合并操作(lazy-merge[59])等.文献[30]提出文件系统元数据分离技术(file system metadata separation, FSMS),针对 FAT 文件系统将文件元数据存储于固定位置的特点,修改块设备驱动程序,使之能够通过地址区别文件元数据和其他数据,一旦识别出元数据写请求则将其发送到 PCM 过滤器中,采用写前读策略将更新的数据与原始数据进行比较,从而只更新发生改变的数据.Park 等人提出的 PFFS 文件系统[31],是将元数据与普通数据分开存储,其中文件系统的元数据信息保存于 PCM 上;由于文件读写访问、重命名或文件位置移动都将引起元数据信息更新,加剧 PCM 的磨损,因此作者给出了多粒度的磨损均衡方法,包括段级(segment-level)交换方法和字级(word-level)移位方法.然而闪存芯片集成规模不断扩大,PFFS 文件系统必须随着闪存可存储的数据量的增长,增加 PCM 容量以容纳所有元数据信息.Park 等人针对元数据信息访问具有局部性和 PCM 存储空间有限的情形,在 2011 年对 PFFS 作了改进,实现了 PFFS2 文件系统[57],仅用 PCM 存储最近频繁使用到的元数据,较冷的元数据则仍存储在 NAND 闪存中;PCM 与 NAND 闪存之间的换入换出操作保证热元数据能及时存储到 PCM,降低性能影响和闪存的磨损.实验结果显示,对小文件读写来说,PFFS 和 PFFS2 分别比 YAFFS2 性能提高 25%和 38%.

由于 PCM 无需擦除,且读写延迟优于 NOR 闪存,有研究用 PCM 替换 NOR 闪存,存储代码信息[30,57,60];另外,PCM 还被用于存储一些普通数据,如文献[61]将用于重复数据检测的指纹数据和数据更新日志存储于 PCM 等.

4.2 PCM 在传统 DBMS 中的应用

目前,直接基于 PCM 的数据库层面的研究工作还比较少.但是,也有一些研究者探索了将 PCM 应用于传统 DBMS 的方法,例如利用 PCM 改进数据库恢复性能[62-63]、利用 PCM 优化数据库算法性能[33]等.

文献[62]提出了一种利用 PCM 改进数据库事务管理的方法——PCMLogging.PCMLogging 的基本思想是将 PCM 作为 DRAM 和磁盘之间的缓存,将脏数据和事务状态等原数据存储在 PCM 上,利用 PCM 的非易失性和较高读写性能,加快数据库的恢复过程,实验数据显示 PCMLogging 方法比传统数据缓冲和日志模式节省 40%磁盘 I/O 和 97%写冲突.文献[63]提出了一种提升数据库事务管理执行效率的方案,利用 PCM 写速度远高于磁盘的特点,提出在传统的存储体系中引入 PCM,专门用于记录日志数据,以解决传统的 ARIES 日志管理机制频繁写磁盘造成的性能瓶颈.在事务执行时,将日志信息全部写在 PCM 上,并异步更新至磁盘.为解决故障发生时可能一条日志尚未写完的问题,该方案修改了 ARIES 机制的日志结构,在日志结构的头部增加了日志长度,日志结构的尾部增加了一个终止位,以确保恢复时使用的是完整的日志记录.实验数据显示,比起传统的 ARIES 日志管理方案,该方案将数据库事务执行速度提升了 7 倍,当事务全部是写事务时性能最高可提升 29 倍.

文献[33]基于高速缓存数据置换写回主存的架构,讨论了 PCM 算法设计分析问题.提出 PCM 算法不仅仅要考虑计算复杂度,还要考虑其是否对高速缓冲友好(cache-friendly).研究表明,高速缓冲中行参数以及算法中读写操作类型等因素对算法运行时间、PCM 的磨损、以及能耗等都有影响.该文献以数据库中重要的 B^+-Tree 索引和散列连接算法为例,以行为读写单位,通过优化算法运行中高速缓存对主存的读写方式,讨论了算法分析中不同因素的作用.这一工作验证了 PCM 上的算法分析需要分别考虑算法中的读写操作,但是它仅仅立足在高速缓存对主存的读写角度来分析算法,关注的仍然是计算机系统中的缓存设计问题,在算法分析中没有考虑 PCM 的重要指标——磨损度量.

4.3 PCM 在实时与并行系统中的应用

文献[64]基于 PCM 和 DRAM 混合主存架构的服务器集群应用方案,设计了一种数据划分算法,在实现集群全局负载均衡的同时,对 DRAM 和 PCM 之间数据也进行了合理分配,使混合系统表现出比单一存储介质系统更好的性能.刘金垒等人给出了一种基于 PCM 和磁盘阵列的层次式并行混合存储系统结构[13],通过在磁盘阵列上增加一层基于 PCM 的本地存储作为高速缓存,以增强计算处理系统应对大规模计算的处理能力.文献[65]以实时系统为应用背景,采用 PCM 作为主存、DRAM 作为 PCM 缓存的存储架构,提出了 3 种实时调度策略,降低了实时请求在截止时间(deadline)到达时未被响应的频率.另一些研究者在一个解码系统中将

PCM 作为二级存储构建了基于时间的数据存储系统[66]. 他们根据 PCM 物理特性和系统数据存储与访问特性,提出了自适应地调整解码强度的 ECC 解码方法.

4.4 PCM 在嵌入式系统中的应用

PCM 应用到嵌入式设备上,系统环境相对简单,存储的数据特征明显且变化性小,因此不需要太灵活的算法处理能力. 文献[67]中提出了一种多项式时间复杂度的软件磨损均衡算法,使用最佳数据分配算法(optimal data allocation, ODA)对代码数据作预处理,产生每个数据区域的预分配方式,并统计相应变量地址空间写次数,最后根据统计结果得到最适合的页面地址分配方案. 文献[68]提出的负载均衡算法设计根据特定的应用及其数据特点,将写操作多的页面聚类到热区域,写操作少的聚类到冷区域,通过 Full Curling 和 Partial Curling 方法进行数据迁移工作,将写均匀分布到 PCM 各个存储单元. 在工业界,三星研发的 65 nm 工艺的多芯片封装 512 MB 相变存储器也已经开始应用到手机产品中[69].

5 未来研究展望

由于 PCM 具有字节可寻址、按位更新、高性能、非易失、低能耗等特性,同时具备了 DRAM 和闪存的优点,因此 PCM 被认为是未来主存和持久存储介质的重要发展方向. 而且,基于 PCM 构建新型存储架构,进而建立基于 PCM 的大数据文件系统以及数据管理系统,也有望为大数据领域提供高效能大数据存储与管理的可行解决方案. 因此,基于 PCM 的大数据存储与管理将是未来存储与数据管理领域中一个极具前景的研究方向.

本节给出了基于 PCM 的大数据存储与管理未来的一些研究展望,包括基于 PCM 的大数据文件系统(如 5.1 节所述)、基于 PCM 的主存系统(如 5.2 节所述)以及基于 PCM 的大数据管理系统(如 5.3 节所述).

5.1 基于 PCM 的大数据文件系统

PCM 的出现及应用打破了传统的 HDD/SDD+DRAM 的存储架构,为适应 PCM 和 DRAM 共存的新型存储架构,需研究新型的可支持以内存访问形式访问各种文件数据的新型文件系统. 同时,由于大数据时代数据一般需要分布式存储与计算,因此在文件管理上还需要考虑对分布式环境的支持. 具体的研究方向包括单节点文件系统和分布式文件系统.

单节点文件系统是研制基于 PCM 的分布式文件系统的基础,它实现基于 PCM 的单个节点上的文件管理功能,需要解决的主要问题包括基于 PCM 的文件原位访问技术、文件系统管理与控制技术、基于 PCM 的内存管理机制等.

本地节点的数据访问仅能够提升应用程序访问本地数据时的效率. 分布式存储技术使得我们可以基于 PCM 搭建支持海量数据存储的分布式环境,从而满足大数据存储的容量需求. 因此,将单节点文件系统向多节点扩充,完成支持 PCM 的分布式文件系统是未来建立基于 PCM 的大数据文件系统的研究重点之一. 分布式文件系统需要解决的主要问题包括分布式文件系统虚拟访问接口、基于统一寻址的分布式文件管理技术、存储空间的全局划分和寻址技术等.

5.2 基于 PCM 的主存系统

目前,已有越来越多的计算机存储系统采用 DRAM,SSD 和磁盘共同构成的混合存储体系架构,而短时间内 PCM 替代上述 3 种存储介质的可能性很小,因此随着 PCM 技术的发展,计算机存储系统可能发展成由 DRAM,PCM,SSD,HDD 共同构成的局面,而非易失性可能使得 PCM 并不仅仅作为主存,PCM 还可能承担一定的持久数据存储工作.

传统主存系统的算法设计都是基于易失性 DRAM 主存架构设计,单一存储介质使得在为空间分配、回收算法设计时,无需考虑存储介质特性差异问题,也无需考虑主存存储架构方面的因素,而新主存介质 PCM 的加入改变了原有的存储架构设计,对存储架构改变敏感的主存系统设计提出了新的挑战,相关硬件设计也可能与传统设计方式产生差异. 下面是基于 PCM 的主存系统中未来值得研究的一些问题.

1) 编址模式与地址映射方法

现有的存储技术限制使得 PCM 与 DRAM 共存成为可能,因此在 PCM 与 DRAM 混合的主存系统中,空间编址是首先需要考虑的问题. 而单独编址或统一编址对现有的计算机硬件设计或系统软件设计均有重大影响,如系统总线布局、操作系统对 PCM 和 DRAM 物理地址空间的可见性、进程虚拟空间地址与物理地址之间的映射方式等. 因此对于混合主存架构的编址模式变化、硬件设计支持及地址映射改变是新主存架构研究的重要关注方向.

2) 空间管理

计算机系统最核心的概念是进程,计算机所有工作内容都是围绕着进程概念展开,而进程离不开内存空间的支持,不同的进程对响应速度、数据等的需求不同,并且同一个进程需要的数据之间也有明显的区别. 在主存介质异构的情况下,给进程分配空间也需将这些因素加以考虑. 因此,结合实际存储架构改进空间分配策略是基于 PCM 主存系统的研究内容.

3) 数据划分

对于 PCM 与 DRAM 混合主存系统架构而言,多种存储介质带来了另一个问题——数据划分;传统主存架构在单一存储介质的情况下,数据存储位置的地位一样,但在混合架构下,数据存储在 PCM 上或是在 DRAM 上带来的性能、能耗都不同;所以对数据进行划分,必要时的数据迁移策略是基于 PCM 的主存系统中的研究重点之一.

4) 数据存储组织

一方面 PCM 是主存存储器的候选者,另一方面 PCM 具有非易失的特点,且存储密度高,具备作为二级存储器的特点. 作为 SCM 技术的候选存储设备之一,PCM 使得主存与二级存储器之间明确的界限(易失与非易失,快速主存与低速磁盘)变得模糊,主存 DRAM 和 PCM 和磁盘就组成了一个混合存储系统,一方面是 DRAM、PCM 主存的混合,另一方面是 PCM 与磁盘的混合,两个混合又通过 PCM 连通. 因此,设计新的存储架构体系定位 PCM 在体系中的位置,根据具体情况开发 PCM 作为主存空间和作为二级存储空间之间的灵活转换机制,设计新的这类非易失快速主存设备的数据存储组织方法也越来越迫切,这在计算机存储领域具有深远的意义,将可能引起现有的存储体系和方法的变革.

5.3 基于 PCM 的大数据管理系统

PCM 对存储与计算架构带来了巨大的改变,也对传统的数据管理系统提出了新的挑战. 新的存储硬件技术的出现往往会带来数据管理技术的革命性变化,例如磁盘的出现奠定了现在的磁盘操作系统、磁盘数据库管理系统的硬件基础. 为此,针对基于 PCM 的新型存储架构,以及大数据管理方面的性能、可扩展等需求,如何构建一个高效的大数据管理系统是今后的一个重要研究方向.

基于 PCM 的新型存储架构不仅带来了数据存取操作和计算模式上的变化,也在数据存储层次、数据分配、数据缓存、查询处理流程等方面带来了极大的改变,同时还对传统的事务处理、日志管理等工作带来了新的优化可能. 因此,未来需要从大数据管理的角度开展基于 PCM 的核心架构与算法研究,目标是构建一个适合 PCM 的高性能、可扩展的大数据管理系统. 主要涉及的一些研究要点如下.

1) 基于 PCM 的数据存储管理

PCM 的引入使得未来计算机系统中很有可能出现 DRAM,PCM,SSD,HDD 等不同层次的存储介质共存的局面. 由于不同的存储介质在易失性、访问延迟、存取模式、容量等方面存在着很大的差别,如何发挥新型存储器件在访问性能、存取模式等方面的优势,是研究基于 PCM 的大数据管理系统所要解决的首要问题. 而存储管理机制对于能否有效发挥新型存储介质的优势有着至关重要的作用. 基于 PCM 的数据存储管理主要涉及的问题包括数据存储分配、数据存储调整等.

在数据存储分配算法方面,一种可能的策略是采用多粒度的分配算法. 所谓多粒度是指在存储分配时同时采用文件和页两种粒度:①在 PCM 与 SSD/HDD 之间进行数据分配时,我们将 PCM 作为持久存储介质,采用文件粒度进行数据分配;②在 DRAM 与 SSD/HDD 之间进行数据分配时,DRAM 作为缓存,采用页粒度进行数据分配;③日志和全局分配信息则以文件形式常驻于 PCM 中.

在数据存储调整方面,我们基于应用对数据的访问模式变化来自适应动态调整数据存储策略. 访问模式的度量可以基于数据的访问频度以及存取方式(读/写)两类因素,通过周期性考察的方法确定当前数据访问模式的变化程度,并基于访问模式的变化程度确定是否重新执行数据存储分配. 一旦确定了新的数据存储分配策略,则需要对相应的数据进行介质之间的迁移操作.

2) 基于 PCM 的数据缓存管理

从存储层次的角度看,引入 PCM 后形成了一个 DRAM,PCM,SSD 等构成的多级缓存结构. 在这种结构下,DRAM 可以发挥其高速计算的特点,但由于 DRAM 具有存储易失性,因此适合进行程序运行中的动态数据存储;PCM 适合存储需要快速访问的静态文件数据;SSD 则适合存储访问频度较低的文件数据. 所有的数据冷热特性判断、数据分配、数据迁移等操作都需要系统提供缓存支持. 传统的缓存管理仅考虑 DRAM 和外存(SSD,HDD 等)之间的数据交换,但在多级缓存结构中,数据交换方式存在着多种可能(如 DRAM 与 PCM,DRAM 与

SSD,PCM 与 SSD 等),交换的粒度也从传统的页面粒度变为页面与字节流共存的情形(如 DRAM 与 PCM 之间的字节流交换等、PCM 与 SSD 之间的页面式交换等),这些因素对数据缓存管理带来了新的挑战,也是未来值得研究的一个重要问题.

3) 基于 PCM 的大数据索引与查询处理

在传统的基于"DRAM+SSD/HDD"的存储架构下,DRAM 与外存之间的 I/O 是影响系统查询处理性能的瓶颈. 在大数据应用中,由于数据的分布以及涉及的数据量过大,依靠传统的数据集划分、查询重写等查询优化机制不能从根本上解决问题. 因此,未来需要研究基于 PCM 的大数据索引技术,并在此基础上研究高性能的查询处理方法.

传统的索引机制如 B^+-Tree,R-Tree 等的应用场景与设计目的往往存在较大差异,而且传统索引机制既没有考虑 PCM 的特性,也没有考虑大数据分布式环境下的性能要求. 因此,研究基于 PCM 的新型索引机制,既需要满足大数据分布式应用场景的要求,还应减少存储在 PCM 上索引数据的更新、插入、删除操作.

4) 基于 PCM 的事务处理与优化

事务处理是传统数据库技术的一大贡献,为数据库技术的实用化奠定了基础. 但是,传统的事务处理技术适合在既定模型下一定数据量的业务流程,在大数据分布式环境中存在着严重的性能和技术问题. 例如,为了保证事务的 ACID 性质,我们不得不引入巨大的锁代价,并在海量的分布式节点之间进行低效的协调处理. 此外,事务日志的存储和管理在分布式和高并发的大数据应用环境下也面临着极大的性能考验. PCM 为我们探索新型的事务处理机制提供了新的机会,例如,我们可以利用 PCM 来作为事务日志的持久存储介质,在事务处理时直接将日志写入到 PCM 中,从而可以保证先写日志规则的有效性,并且避免事务处理时的 flush log 操作,提高事务处理的效率. 同时,由于事务日志存储在 PCM 中,因此在数据库恢复时可以直接从 PCM 中读取日志,利用 PCM 的高速存取特点提高恢复的性能.

6 结束语

新型存储介质 PCM 的出现及其快速发展,给传统的存储架构带来了新的变革,也为大数据存储与管理提供了新的研究契机. 本文重点讨论了新型存储器 PCM 的特点,并在此基础上对目前 PCM 存储架构及其关键技术进行了综述,包括 PCM 持久存储技术以及基于 PCM 的主存系统等,在此基础上总结了 PCM 在不同领域的应用现状,最后对基于 PCM 的大数据存储与管理未来发展方向进行了展望,以期能对这一领域的未来研究提供有价值的参考.

致谢 中国科学院上海微系统所陈小刚博士、中国人民大学信息学院曹巍博士参与了本文内容的前期讨论,在此表示感谢!

参 考 文 献

[1] Meng Xiaofeng, Ci Xiang. Big data management: Concepts, techniques and challenges [J]. Journal of Computer Research and Development, 2013, 50(1): 146-169 (in Chinese)
(孟小峰, 慈祥. 大数据: 概念、技术与挑战[J]. 计算机研究与发展, 2013, 50(1): 146-169)

[2] Konstantinou I, Tsoumakos D, Mytilinis I, et al. DBalancer: Distributed load balancing for NoSQL data-stores [C] // Proc of the 2013 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2013: 1037-1040

[3] Shen Derong, Yu Ge, Wang Xite, et al. Survey on NoSQL for management of big data [J]. Journal of Software, 2013, 24(8): 1786-1803 (in Chinese)
(申德荣, 于戈, 王习特, 等. 支持大数据管理的 NoSQL 系统研究综述[J]. 软件学报, 2013, 24(8): 1786-1803)

[4] Plattner H, Zeier A. In-Memory Data Management [M]. Translated by SAP. 2nd ed. Beijing: Tsinghua University Press, 2012 (in Chinese)
(Plattner H, Zeier A. 内存数据管理[M]. SAP 译. 2 版. 北京: 清华大学出版社, 2012)

[5] Wang Jiangtao, Lai Wenyu, Meng Xiaofeng. Flash-based database: Studies, techniques and forecasts [J]. Chinese Journal of Computers, 2013, 36(8): 1549-1567 (in Chinese)
(王江涛, 赖文豫, 孟小峰. 闪存数据库: 现状、技术与展望[J]. 计算机学报, 2013, 36(8): 1549-1567)

[6] Raoux S, Burr G W, Breitwisch M J, et al. Phase-change random access memory: A scalable technology [J]. IBM Journal of Research and Development (IBMRD), 2008, 52(4/5): 465-480

[7] Burr G W, Kurdi B N, et al. Overview of candidate device technologies for storage-class memory [J]. IBM Journal of Research and Development, 2008, 52(4/5): 449-464

[8] Lai S. Current status of the phase change memory and its future [C] // Proc of the IEEE Int Electron Devices Meeting (IEDM2003). Piscataway, NJ: IEEE, 2003: 10.1.1-10.1.4

[9] Freitas R. Storage class memory: Technology, systems and applications [OL]. [2014-10-08]. http://www.hotchips.org/wp-content/uploads/hc_archives/hc22/HC22.22.160-Freitas-Storage-Class-Memory.pdf

[10] Ding Sheng, Song Zhitang, Chen Houpeng, et al. Design of a 1 kb phase change memory chip [J]. Microelectronics, 2011, 41(6): 844-851 (in Chinese)
(丁晟, 宋志棠, 陈后鹏, 等. 一种 1 kb 相变存储芯片的设计[J]. 微电子学, 2011, 41(6): 844-851)

[11] Cai Dailin, Chen Houpeng, Wang Qian, et al. An 8 Mb phase change random access memory based on 0.13 μm technology [J]. Research & Progress of SSE, 2011, 31(6): 601-605 (in Chinese)
(蔡道林, 陈后鹏, 王倩, 等. 基于 0.13μm 工艺的 8 Mb 相变存储器 [J]. 固体电子学研究与进展, 2011, 31(6): 601-605)

[12] Deng Zhixin, Gan Xuewen. Introduction and prospects of phase change memory [J]. China Integrated Circuit, 2005, (4): 48-51 (in Chinese)
(邓志欣, 甘学温. 相变存储器简介与展望[J]. 中国集成电路, 2005, (4): 48-51)

[13] Liu Jinlei, Li Qiong. Application research on new non-volatile phase change memory PCM [J]. Journal of Computer Research and Development, 2012, S1: 90-93 (in Chinese)
(刘金垒, 李琼. 新型非易失相变存储器 PCM 应用研究[J]. 计算机研究与发展, 2012, S1: 90-93)

[14] Qureshi M K, Srinivasan V, Rivers J A. Scalable high performance main memory system using phase-change memory technology [C]// Proc of the 36th Annual ACM Int Symp on Computer Architecture. New York: ACM, 2009: 24-33

[15] Lee E, Jang J, Bahn H. DTFS: Exploiting the similarity of data versions to design a write-efficient file system in phase-change memory [C]// Proc of the 29th Annual ACM Symp on Applied Computing. New York: ACM, 2014: 1535-1540

[16] Lee E, Yoo S, Jang J E, et al. Shotcut-JFS: A write efficient journaling file system for phase change memory [C]// Proc of the 28th IEEE Symp on Mass Storage Systems and Technologies (MSST). Piscataway, NJ: IEEE, 2012: 1-6

[17] Lee E, Jang J E, et al. On-demand snapshopt: An efficient versioning file system for phase-change memory [J]. IEEE Trans on Knowledge and Data Engineering, 2013, 25(12): 2841-2853

[18] Choi G S, On B W, Choi K, et al. PTL: PRAM translation layer [J]. Microprocessors and Microsystems, 2012, 37(1): 24-32

[19] Im S, Shin D. Differentiated space allocation for wear leveling on phase-change memory-based storage device [J]. IEEE Trans on Consumer Electronics, 2014, 60(1): 45-51

[20] Condit J, Nightingale E B, Frost C, et al. Better I/O through byte-addressable, persistent memory [C]// Proc of the 22nd ACM SIGOPS Symp on Operating Systems Principles. New York: ACM, 2009: 133-146

[21] Coburn J, Caulfield A M, Akel A, et al. NV-Heaps: Making persistent objects fast and safe with next-generation, non-volatile memories [C]// Proc of the 16th Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2011: 105-118

[22] Volos H, Tack A J, Swift M M. Mnemosyne: Lightweight persistent memory [C]// Proc of the 16th Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2011: 91-104

[23] Oikawa S, Miki S. File-based memory management for non-volatile main memory [C]// Proc of the 37th IEEE Annual Computer Software and Applications Conf. Piscataway, NJ: IEEE, 2013: 559-568

[24] Chen Jianxi, Wei Qingsong, Chen Cheng, et al. FSMAC: A file system metadata accelerator with non-volatile memory [C]// Proc of the 29th IEEE Symp on Mass Storage Systems and Technologies (MSST). Piscataway, NJ: IEEE, 2013: 1-11

[25] Kang Y, Yang Jingpei, Miller E L. Object-based SCM: An efficient interface for storage class memories [C]// Proc of the 27th IEEE Symp on Mass Storage Systems and Technologies (MSST). Piscataway, NJ: IEEE, 2011: 1-12

[26] Zhou Ping, Zhao Bo, Yang Jun, et al. A durable and energy efficient main memory using phase change memory technology [C]// Proc of the 36th Annual ACM Int Symp on Computer Architecture. New York: ACM, 2009: 14-23

[27] Kong Jingfei, Zhou Huiyang. Improving privacy and lifetime of PCM-based main memory [C]// Proc of the 40th Annual IEEE/IFIP Int Conf on Dependable Systems and Networks. Piscataway, NJ: IEEE, 2010: 333-342

[28] Yoon D H, Muralimanohar N, Chang Jichuan, et al. FREE-p: Protecting non-volatile memory against both hard and soft errors [C]// Proc of the 17th IEEE Int Symp on High Performance Computer Architecture. Piscataway, NJ: IEEE, 2011: 466-477

[29] Qureshi M K. pay-as-you-go: Low-overhead hard-error correction for phase change memories [C]// Proc of the 44th Annual IEEE/ACM Int Symp on Microarchitecture. New York: ACM, 2011: 318-328

[30] Kim J K, Lee H G, Choi S, et al. A PRAM and NAND flash hybrid architecture for high-performance embedded storage subsystems [C]// Proc of the 8th ACM Int Conf on Embedded Software (EMSOFT). New York: ACM, 2008: 31-40

[31] Park Y, Lim S H, Lee C, et al. PFFS: A scalable flash memory file system for the hybrid architecture of phase change RAM and NAND flash [C]// Proc of the 2008 ACM Symp on Applied Computing. New York: ACM, 2008: 1498-1503

[32] Bivens A, Dube P, Franceschini M, et al. Architectural design for next generation heterogeneous memory systems [C]// Proc of the 4th IEEE Int Memory Workshop. Piscataway, NJ: IEEE, 2010: 1-4

[33] Chen Shimin, Gibbons P B, Nath S. Rethinking database algorithms for phase change memory [C/OL] // Proc of the 5th Biennial Conf on Innovative Data Systems Research. 2011: 21-31. [2014-10-08]. http: // www. cidrdb. org/cidr2011/Papers/CIDR11_Paper3. pdf

[34] Chen C H, Hsiu P C, Kuo T W, et al. Age-based PCM wear leveling with nearly zero search cost [C] // Proc of the 49th Annual ACM Design Automation Conf. New York: ACM, 2012: 453-458

[35] Zhou Ping, Zhao Bo, Yang Jun, et al. A durable and energy efficient main memory using phase change memory technology [C] // Proc of the 36th Annual ACM Int Symp on Computer Architecture. New York: ACM, 2009: 14-23

[36] Yang B D, Lee J E, Kim J S, et al. A low power phase-change random access memory using a data-comparison write scheme [C] // Proc of the 2007 IEEE Int Symp on Circuits and Systems. Piscataway, NJ: IEEE, 2007: 3014-3017

[37] Cho S, Lee H. Flip-N-Write: A simple deterministic technique to Improve PRAM write performance, energy and endurance [C] // Proc of the 42nd Annual IEEE/ACM Int Symp on Microarchitecture. New York: ACM, 2009: 347-357

[38] Qureshi M K, Franceschini M M, Andlastras-Montano L A. Improving read performance of phase change memories via write cancellation and write pausing [C] // Proc of the 16th IEEE Int Symp on High Performance Computer Architecture (HPCA). Los Alamitos, CA: IEEE, 2010: 1-11

[39] Ramos L, Gorbatov E, Bianchini R. Page placement in hybrid memory systems [C] // Proc of the Int Conf on Supercomputing. New York: ACM, 2011: 85-95

[40] Lee S, Bahn H, Noh S H. Characterizing memory write references for efficient management of hybrid PCM and DRAM memory [C] // Proc of the 19th Annual IEEE/ACM Int Symp on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. Piscataway, NJ: IEEE, 2011: 168-175

[41] Shin D J, Park S K, Kim S M, et al. Adaptive page grouping for energy efficiency in hybrid PRAM-DRAM main memory [C] // Proc of the 2012 ACM Research in Applied Computation Symp. New York: ACM, 2012: 395-402

[42] Park Y, Shin D J, Park S K, et al. Power-aware memory management for hybrid main memory [C] // Proc of the 2nd Int Conf on Next Generation Information Technology. Piscataway, NJ: IEEE, 2011:82-85

[43] Park Y, Park K H. Linux kernel support to exploit phase change memory [C/OL] // Proc of Linux Symp. 2010: 217-224. [2014-10-08]. http: // www. kernel. org/doc/ds/2010/ols2010-pages-217-224. pdf

[44] Seok H, Park Y, Park K H. Migration based page caching algorithm for a hybrid main memory of DRAM and PRAM [C] // Proc of the 26th Annual ACM Symp on Applied Computing. New York: ACM, 2011: 595-599

[45] Seok H, Park Y, Park K, et al. Park. Efficient page caching algorithm with prediction and migration for a hybrid main memory [J]. ACM SIGAPP Applied Computing Review, 2011, 11(4): 38-48

[46] Chen Kaimeng, Jin Peiquan, Yue Lihua. A novel page replacement algorithm for the hybrid memory architecture involving PCM and DRAM [C] // Proc of the 11th IFIP Int Conf on Network and Parallel Computing (NPC). Berlin: Springer, 2014, 108-119

[47] Wu Zhangling, Jin Peiquan, Yang Chengcheng, et al. APP-LRU: A new page replacement method for PCM/DRAM-Based hybrid memory systems [C] // Proc of the 11th IFIP Int Conf on Network and Parallel Computing (NPC). Berlin: Springer, 2014: 84-95

[48] Qureshi M K, Karidis J, Franceschini M, et al. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling [C] // Proc of the 42nd Annual IEEE/ACM Int Symp on Microarchitecture. New York: ACM, 2009: 14-23

[49] Park S K, Seok H, Shin D J, et al. PRAM wear-leveling algorithm for hybrid main memory based on data buffering, swapping, and shifting [C] // Proc of the 27th Annual ACM Symp on Applied Computing. New York: ACM, 2012: 1643-1644

[50] Park S K, Maeng M K, Park K, et al. Adaptive wear-leveling algorithm for PRAM main memory with a DRAM buffer [J]. ACM Trans on Embedded Computing Systems, 2014, 13(4): 88

[51] Lee B C, Ipek E, Mutlu O, et al. Architecting phase change memory as a scalable dram alternative [C] // Proc of the 36th Annual ACM Int Symp on Computer Architecture, New York: ACM, 2009: 2-13

[52] Ferreira A P, Zhou M, Bock S, et al. Increasing PCM main memory lifetime [C] // Proc of the 2010 Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2010: 914-919

[53] Park H, Yoo S, Lee S. Power management of hybrid dram/pram-based main memory [C] // Proc of the 48th ACM/EDAC/IEEE Design Automation Conf. New York: ACM, 2011: 59-64

[54] Canim M, Mihaila G A, Bhattacharjee B, et al. SSD Bufferpool Extensions for Database Systems [J]. VLDB Endowment, 2010, 3(1/2): 1435-1446

[55] Kang W, Lee S, Moon B. Flash-based extended cache for higher throughput and faster recovery [J]. VLDB Endowment, 2012, 5(11): 1615-1626

[56] Fisher N, He Z, McCarthy M. A hybrid filesystem for hard disk drives in tandem with flash memory [J]. Computing, 2012, 94(1): 21-68

[57] Park Y, Park K H. High-performance scalable flash file system using virtual metadata storage with phase-change RAM [J]. IEEE Trans on Computers, 2011, 60(3): 321-334

[58] Liu Duo, Wang Tianzheng, Wang Yi, et al. PCM-FTL: A write-activity-aware NAND flash memory management scheme for PCM-based embedded systems [C]// Proc of the 32nd IEEE Real-Time Systems Symp(RTSS). Piscataway, NJ: IEEE, 2011: 357-366

[59] Liu Duo, Wang Tianzheng, Wang Yi, et al. A block-level flash memory management scheme for reducing write activities in PCM-based embedded systems [C]// Proc of the 2012 Conf on Design, Automation and Test in Europe (DATE). San Jose, CA: EDA Consortium, 2012: 1447-1450

[60] Lee H G. High-Performance NAND and PRAM Hybrid storage design for consumer electronics [J]. IEEE Trans on Consumer Electronics, 2010, 56(1): 112-118

[61] Pathak S, Tay Y C, Wei Q. Power and Endurance Aware Flash-PCM Memory System [C]// Proc of the 2011 Int Green Computing Conf and Workshops (IGCC). Piscataway, NJ: IEEE, 2011: 1-6

[62] Gao S, Xu J, He B, et al. PCMLogging: Reducing transaction logging overhead with PCM [C]// Proc of the 20th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2011: 2401-2404

[63] Fang Ru, Hsiao Hui-I, He Bin, et al. High performance database logging using storage class memory [C]// Proc of the 27th IEEE Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2011: 1221-1231

[64] Ramos L, Bianchini B. Exploiting Phase-Change Memory in Cooperative Caches [C]// Proc of the 24th Int Symp on Computer Architecture and High Performance Computing. Piscataway, NJ: IEEE, 2012: 227-234

[65] Zhou Miao, Bock S, Ferreira A P. Real-Time Scheduling for Phase Change Main Memory Systems [C]// Proc of the 10th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2011: 991-998

[66] Wu Qi, Sun Fei, Xu Wei, et al. Using multilevel phase change memory to build data storage: A time-aware system design perspective [J]. IEEE Trans on Computers, 2013, 62(10): 2083-2095

[67] Liu Duo, Wang Tianzheng, Wang Yi, et al. Curling-PCM: Application-specific wear leveling for phase change memory based embedded systems [C]// Proc of the 18th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2013: 279-284

[68] Hu Jingtong, Zhuge Qingfeng, Xue C J, et al. Software enabled wear-leveling for hybrid PCM main memory on embedded systems [C]// Proc of Design, Automation and Test in Europe (DATE2013). Piscataway, NJ: IEEE, 2013: 599-602

[69] Krishnamurthy R. Comparing samsung NOR-Compatible PCM with samsung NOR [EB/OL]. [2014-10-08]. http://www.chipworks.com/en/technical-competitive-analysis/resources/blog/comparing-samsung-nor-compatible-phase-change-memory-pcm-with-samsung-nor-flash-used-as-memory-interchangeably-in-samsung-gt-e2550-gsm-phones/

Wu Zhangling, born in 1988. PhD candidate. Her main research interests include hybrid memory management and flash-based databases, and new database architecture(linglang@mail.ustc.edu.cn).

Jin Peiquan, born in 1975. PhD, associate professor. His main research interests include databases on new storage, spatial-temporal databases, and Web information retrieval(jpq@ustc.edu.cn).

Yue Lihua, born in 1952. Professor, PhD supervisor. Her main research interests include flash-based databases, spatial-temporal databases, and remote-sensing image processing(llyue@ustc.edu.cn).

Meng Xiaofeng, born in 1964. PhD. Professor and PhD supervisor. His main research interests include flash-based databases, moving object data management, Web data management, and cloud data management(xfmeng@ruc.edu.cn).