Report on the Second International Workshop on Flash-Based Database Systems (FlashDB 2012)

Xiaofeng Meng[†], Bingsheng He[‡], Wei Cao[†], Jianliang Xu[*]

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

[‡]School of Computer Engineering, Nanyang Technological University, Singapore

*Department of Computer Science, Hong Kong Baptist University

1. INTRODUCTION

Nowadays, as OLTP and OLAP applications' data volume grows into "big data" scale, requirements such as high performance, low latency, high availability, and low power consumption etc. become more and more critical and challenging. This trend gives rise to the advent of new types of storage media and storage devices, such as flash-based Solid State Drives and Phase Change Memory, which are competitive rivals of traditional magnetic disks and main memory.

Being the pioneers in the storage innovation market, flash-based devices have prevailed in consumer electronics because of non-volatility, low-cost, small size, shock resistance, and low-power consumption. But features like out-place updates, asymmetric read/write/erasure latencies, and limited life span and capacity etc. pose challenges in directly using flash-based devices as data storage devices. Storage subsystems in data management software should be carefully redesigned considering the specific characteristics of the underlying storage devices.

The Second International Workshop on Flash-based Database Systems (FlashDB 2012) was held in conjunction with DASFAA 2012 conference on 15 April in Busan, South Korea. Researchers and scholars from South Korea, Mainland China, Germany, Iceland, Poland, and United States in academia and industry shared their new insights, ideas and findings in this area. This half-day event carried on the advances in the open topics in flash-based database systems: hybrid-storage (1 paper), indexes (2 papers), buffer management (1 paper). Besides, a proposal of new FTL designs, a column-wise storage model and space reclamation algorithms for flash-based SSDs were discussed.

The workshop was moderated by PC Co-Chair Dr. Jianliang Xu, associate professor from Hong Kong Baptist University. This workshop featured an invited talk from the industry perspective and three research sessions from the academia perspective. Attendees of FlashDB 2012 workshop enjoyed the in-depth technical conversations between scholars and product vendors.

2. INVITED TALK

The invited talk was given by Bumsoo Kim, currently President of Semiconductor Division (INDILINX) at OCZ Technology Group, Inc. His talk gave a picture of OCZ's commercial SSD products technology whose sophisticated designs realize the potentials of flash SSDs in terms of high performance, high reliability and low cost. The talk was entitled "**Commercial SSD Products – Status Quo and Next**".

The first part was a brief introduction of OCZ and its products. This San Jose-based global company develops, produces, and distributes solid-state storage devices and high-performance computer components. It currently offers a rich portfolio of enterprise SSDs in a range of form factors and interfaces including SATA, PCIe, and SAS. Specifically it focuses on high transactional performance with low cost MLC and enterprise-MLC drives exceeding 800K random write IOPS. By adopting SSDs in data storage, current data placement policy is changed to a data tiering model, with the frequently-used "hot" data stored in higher tiers on PCIe based SSDs and less accessed "cold" data stored on traditional HDDs. Three dimensions of value are recognized as goals set for the product family, which are high reliability (especially for enterprise application), high performance (especially for performance critical application), and low Price/GB (especially for mainstream applications).

In the second part of the talk, the speaker introduced the product series designated for the three categories in terms of SATA, SAS and PCI-e interfaces.

In the third part of the talk, the speaker detailed two specific technologies called VCA (Virtualized Controller Architecture) and VXL (Virtualized Flash Cache) respectively. VCA virtualizes solid state storage devices into a massively parallel array of memory by combining the OCZ SuperScale™ controller technology and OCZ developed software stack. Since it provides SCSI over PCIe functionality, drives appear to the OS as a SCSI device and standard SCSI-SATA commands are also supported. To increase the performance, VCA makes use of algorithms like Tagged Command Queuing (TCQ), Native Command Queuing (NCQ) and OCZ Advanced Queue Balancing Algorithm (QBA). It also allows for monitoring, analyzing and reporting of the status of the device. PCIe SSDs with OCZ VCA technology, e.g., the Z-Drive R4-RM88 model with full height reaches a maximum read/write bandwidth of 2.8 GB/s, and 4KB-sized write rate of 410,000 IOPS. PCIe generation 3 series featured with VCA, i.e., Z-Drive R5 can reach 2.5 million IOPS and a read/write bandwidth of 7GB/s.

VXL is a solution serving as data center flash cache, which empowers virtualization by increasing the number of vms per server and lowering total storage costs. Based on OCZ Z-Drive R4 CloudServ™ SSDs and SANRAD VXL acceleration and virtualization techniques, it improves performance through application optimized caching with dynamic run-time adjustment. In addition, SANRAD virtual v-switch component works with vMotion to make sure that the cached data are not lost during virtual machines' migration. SANRAD VXL also supports mirroring for high availability and can create SAN-less VMWare clusters, which can be managed through integration into vCenter or through standalone OCZ StoragePro management station.

Finally the speaker envisioned future challenges for new SSDs. At the NAND chip level, while sub-20 nm technology could offer better density and data capacity, reliability issues will surface which encourages rethink of techniques such as ECC and read-retry, as well as flash disk arrays (RAID). High speed circuit design poses challenges of signal integrity and power integrity problems. TLC and 3D NAND technologies are accelerating the NAND evolution in terms of more capacity and reduced cost. But reliability and performance issues still exist. At the interface level, challenges lie in how to make full utilization of next host interface, e.g. PCI-e Gen3, in addition to multi-core CPUs and firmware architectures. And the higher level integration should tackle scalability and other open issues in scenarios like multiple PCIe SSDs on a server, massive use of SSDs in data centers, or, even system software for better use of SSDs.

3. RESEARCH SESSIONS

3.1 Session 1: Buffer Management Revisit

An effective buffer management policy is important to the performance of database systems. Flash memory has many properties different from magnetic disks which make traditional buffer replacement strategies not suitable for flash-based DBMS. This session featured two research papers that exploit the characteristics of flash devices to effectively enhance the ability of buffer manager in a database environment.

The paper entitled "h-Buffer: An Adaptive Buffer Management Scheme for Flash-Based Storage Devices" [6] proposed a novel buffer management policy using a hybrid clustering granularity to address some existing problems caused by adopting exclusively page-clustering algorithms (PCA) or group-clustering ones (GCA). PCA is good at distinguishing hot pages and cold pages easily and is good for the simple replacement policy. But PCA's performance is worse than that of GCA when the buffer size is big enough because of the flush-outs by pages. GCA works well for write requests but when the buffer size is limited, it causes more writes and erasure operations. h-Buffer manages hot pages at page-level, and cold dirty pages at group-level. Specifically, the authors divided the buffer pool into three areas: hot list (HL), cold clean list (CCL) and cold dirty group list (CDGL) and employed three policies, namely, a replacement policy, a write-back policy, and a hot list compensating policy. The evicting in HL and CCL operates in the unit of page. In CDGL, the strategy selects a dirty page cluster to flush to flash memory. The policies provided in this paper are flexible in taking into account the access pattern of workload, the size of buffer pool and the FTL environment. Besides, the authors designed an optimal scheme to reduce the number of write and erase operations on flash memory by logging or padding when evicting a victim dirty group. The experiment result based on a simulator showed that h-buffer gained a significant performance improvement which is up to 50% reduction on erasure count, read count, write count and run time.

The paper entitled "Improving Database Performance Using a Flash-Based Write Cache" [7] used flash memory as an extension of dynamic random access memory (DRAM) to improve the performance of buffer management. Compared to hard disk drives (HDDs), the authors argued that flash-based solid-state drives (SSDs) cannot replace HDDs completely in the short term due to their disadvantages in capacity and price. The authors selected SSDs as a write cache for a database system stored on HDDs to reduce the disk I/O count and execution time. Along this idea, the authors proposed a three-tier storage architecture. The flash-based write cache is a middle-tier between the RAM-based buffer pool and the database stored on HDDs. Admit-on-Read (AOR) and Admit-on-Write (AOW) are used to serve as the cache admission strategies. As for cache-writing strategies, the authors proposed two kinds of schemes which are random cache write (RCW) and sequential cache write (SCW) and discussed their difference. Furthermore, in order to reduce the cost of disk seeking, the authors exploited a track-aware algorithm called coalesced flushing (CF). CF keeps tracks of the mapping information between SSD and HDD and flush flash-resident data pages in batch. The experiment result based on trace-driven simulation showed that both execution time and seeking cost have a significant reduction.

3.2 Session 2: Flash Memory System Internals

Space reclamation and flash translation layer (FTL) are two key problems of flash memory system internals. The first paper presentation in this session addressed the issue of space reclamation; the second discussed flash translation layer (FTL) with dual granularity.

The paper entitled "A Study of Space Reclamation on Flash-based Append-only Storage Management" [8] mainly addressed the problem of space reclamation of flash-based append-only storage management, which works for both relational database systems and key-value database systems built on flash devices. Record-level space reclamation relies on two kinds of record layouts: version link list-based record layout (VLL-RL) and delete record-based record layout (DR-

RL). And then based on these structures, the authors proposed three space reclamation algorithms, namely version link list-based, sorted array-based and bitmap index-based. Simulation experiments showed that compared to other techniques, bitmap index-based space reclamation has 32X spatial performance improvement and 23X-44X temporal performance improvement.

In the second paper entitled “Dual-Grained FTL for Flash Memory” [9], the authors proposed a novel FTL, DGFTL (Dual-Grained FTL), which was used to cut down the overhead of small random writes and merge operations, especially full merges. In the proposed algorithm, flash memory is divided into two regions, namely a page region and a block region. Write buffer in main memory was used to improve the performance by replacing small random writes with sequential ones. For the two regions of flash memory, there are two mapping tables and a bitmap table for managing the logical addresses and physical addresses. Based on these data structures, the authors proposed page read and page write algorithms. The experiments verified that the count of erase operations was reduced down to less than 50% of other existing FTLs.

3.3 Session 3: Flashing Up Access Methods

Compared to magnetic disks, flash memory performs much better on IO speed. However, due to its characteristic electronic nature, flash devices have distinctive features, such as the discrepancy between read and write rate, erasure before write constraints etc. Access methods like B+ trees, R-trees and so on play an important role in DB systems, which accelerate the access of the data. How would storage systems influence the access methods’ performance in a balancing way between the “good side” and the “bad side” of flash-based SSDs? How to design access methods that are optimized specifically for SSDs? What would it be like when we adopt a more radical change in storage subsystems: from row-based storage model to column-wise model? The three research papers in this session addressed these problems respectively.

The paper entitled “Impact of Storage Technology on the Efficiency of Cluster-based High-Dimensional Index Creation” [4] sets itself in the context of content retrieval in multimedia data collections, where high-dimensional indices facilitate retrieval by pruning the search space. To inspect the performance impact of different types of secondary storage, the authors designed two different implementations for the creation of extended Cluster Pruning (eCP) algorithm — a cluster-based high dimensional indexing algorithm optimized for the case when data volume is too large to reside in main memory [2][3]. Specifically, the two implementations adopt temporary files (TF) policy and chunk files policy (CF) respectively during both the assignment phase, when vectors processed in main memory buffer are flushed out to secondary storage, and the merging phase, when the final index file is created. TF policy uses one temporary file for each cluster and entails large sequential reads and many small random writes during assignment phase, and cluster-sized sequential reads and large sequential writes during the merging phase [4]. CF policy follows a sort-merge style: conducting in-memory sorting and flushing out to newly created chunk files during the assignment phase, and merges all the sorted chunk files using “a typical secondary storage merging process” to get the final file during merging phase. Hence, CF policy induces large sequential reads and large sequential writes during assignment phase and many small random reads and large sequential write during merging phase. Experiments using “a real data collection made of more than 110 million local SIFT descriptors [5] computed on real images randomly downloaded from Flickr” demonstrated the impact on index construction performance of magnetic disks, flash-based SSDs and NAS solution. The results show that in single drive setup, where the raw collection, the temporary files/chunk files and the final file were all stored on a single disk, TF policy’s performance varied greatly across different secondary storage devices. CF policy has the best performance on SSDs compared to CF on all the other kinds of devices and TF on all the devices. The Intel SSD excels at handling random reads and writes better than the SSDs from other vendors. In two-drive setup, where the raw data collection and the final file were kept in one disk and temporary files/chunk files were stored on the other disk, a hybrid magnetic-SSD strategy (where intermediate files were stored on flashed-based SSDs) and an all-SSD strategy were investigated and the result showed both TF and CF policies exhibited considerable performance improvement over the single-drive setup. CF policy running with the all-SSD strategy had the best performance.

The short paper entitled “Implementation of the Aggregated R-Tree over Flash Memory” [10] proposed a new method to extend aR-trees (aggregated R-trees - storing aggregated values along with the entries inside R-tree nodes) on flash memory. In the proposed method, PARM, index items, which are stable, and aggregated values, which are subjected to frequent changes, are separately stored in different sectors of flash memory to avoid modifying the whole index item when the aggregated values are changed. The authors analyzed the numbers of reads and writes caused by updates of aggregated values. Simulation experiments showed the effect that PARM method trades more read operations for fewer write operation, which are very expensive on flash memory. The authors concluded that the proposed PARM method outperformed the original aR-tree method over flash memory.

The short paper entitled “A Flash-Based Decomposition Storage Model” [11] put forward a new column storage model for flash-based SSDs. Columnar storage has gained much popularity recently. Most related works adopted such magnetic storage devices as HDDs. However, those column stores might not exhibit their best advantages on HDDs for the requirement of frequent random reads. The authors suggested a new column storage model named as Flash-based Decomposition Storage Model (FBDSM), which adapts to the properties of SSDs. In FBDSM, every attribute has a Primary Table (PT) to store the inserted or loaded data and a Log Table (LT) to store updated or deleted data. These structures solved the dilemma of update with entry sequence in column storage on SSDs. Implementations of such operations as inserts, deletes, updates, data merge and query processing based on these structures were discussed. To evaluate the effectiveness of the proposed model, the traditional Decomposition Storage Model (DSM) [1] and the proposed FBDSM model were simulated on MySQL. The experiment result showed that the update latency of FBDSM was better than that of the traditional Decomposition Storage Model (DSM). And the fewer attributes got updated, the more reduction was the update latency of FBDSM than that of DSM. The authors also estimated the query processing performance by analyzing the join cost.

4. DISCUSSIONS

In the part of future challenges in the invited talk, three directions at different levels of SSD related technologies were discussed. At the chip level, the future NAND technology is featured with sub-20nm nodes, higher frequency and TLC/3D NAND techniques. At the interface level, techniques should seek to maximize utilization of host interface, such as PCIe Gen 3. And at the higher levels, server-level multiple PCIe SSD integration, data center level SSD’s massive use, and system software level SSD optimization, are practically significant research topics.

This inspiringly drives the flash-based database research community to reconsider the trends of commercial SSD products, including the potentials and limits, to fully exploit the advantage of high performance and other features. Interesting questions include, but are not limited to: (1) how to design a sound and flexible data storage system for data tiering which has high performance and low cost? (2) how to design data management systems for enterprise scenarios such as servers supporting multiple PCIe SSDs or data centers? How to design new DBMS strategies in the virtual machine environment? (3) Should new objectives like low power consumption become more and more equally eminent as performance? what and how can flash-based database systems re-adjust to embrace and contribute to those objectives?

5. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under grant Nos. 60833005 and 61073039, and Hong Kong Research Grant Council under grant No. HKBU211510.

6. REFERENCES

[1] G. Copeland, S. Khoshafian, A Decomposition Storage Mode, In Proc. of SIGMOD 1985, pp. 268-279, 1985.

[2] F. Chierichetti, A. Panconesi, P. Raghavan, M. Sozio, et al, Finding near neighbors through cluster pruning. In Proc. Of PODS 2007, pp. 103–112, 2007.

[3] G. Gudmundsson, B. Jónsson, L. Amsaleg, A large-scale performance study of cluster-based high-dimensional indexing. In: Proc. of the international workshop on Very-large-scale multimedia corpus, mining and retrieval 2010, pp. 31-36.

[4] G. Gudmundsson, L. Amsaleg, B. Jónsson, Impact of storage technology on the efficiency of cluster-based high-dimensional index creation, in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 53-64, 2012.

[5] D. G. Lowe, Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2) pp. 91-110, 2004.

[6] R. Wang, L. Yue, P. Jin, J. Wang, h-Buffer: An Adaptive Buffer Management Scheme for Flash-Based Storage Devices, in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 14 – 27, 2012.

[7] Y. Ou, T. Härder, Improving Database Performance Using a Flash-Based Write Cache, in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 2-13, 2012.

[8] Y. Fan, W. Cao, X. Meng, A Study of Space Reclamation on Flash-based Append-only Storage Management, , in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 28-39, 2012.

[9] J. Wang, L. Yue, P. Jin, R. Wang, A Dual-Grained FTL for Flash Memory, in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 40-52, 2012.

[10] M. Pawlik, W. Macyna, Implementation of the Aggregated R-Tree over Flash Memory, in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 65-72, 2012.

[11] Q. Cao, Z. Liang, Y. Fan, X. Meng, A Flash-Based Decomposition Storage Model, in Proc. of DASFAA’12 workshop, LNCS 7240, pp. 73-80, 2012.