Improving Performance and Reliability of Solid State Disk by Exploiting the Trade-Off between Heterogeneous Flash Memories

Ahmed Izzat Alsalibi, Putra Sumari, and Moh'd Khaled Yousef Shambour

***Abstract*—As a novel storage technology, SSD provide efficient features and challenges. The physical semiconductor characteristics of SSD result into high performance, power consumption, light size, shock resistance, and low noise. With all the benefits of SSD, replacing the conventional HDD drives with pure SSD may not be an efficient option for large-scale storage systems. This is mainly due to many disadvantages of SSD e.g. limited lifespan, small density, and high cost. Thus, a more practical solution is to use hybrid SSD (i.e., SSD consisting of SLC, MLC, and TLC) in hierarchy of storage system, such that the features of this technology are best utilized. In this research, an efficient scheme was proposed. The prosed scheme called Cost, Performance, and Reliability Concern Scheme (CPRCS) for hybrid SSD. CPRCS aim to exploit the trade-off between performance, reliability, cost, density, and power consumption. The eligibility of the CPRCS was proved by using widely used SSD simulation tools: the DiskSim and SSD extension from Microsoft. The results reveal that the proposed scheme is very competitive to state-of-the-art schemes in terms of effectiveness and efficiency.**

***Index Terms*—Hybrid solid state disk, flash memory, performance, reliability.**

I. Introduction

Recently, flash memory-based Solid State Drive (SSD) have received much attention to replace the HDD in the computer system. SSD use semiconductor chips instead of magnetic platters to store data. Semiconductor chips have novel technical features, e.g. low power consumption, shock resistance, high density, no noise, small size, not affected by magnetism, and high random access performance [1], [2]. These unique features can overcome the shortcomings of traditional magnetic disks. However, SSD remain a complicated storage device with it is properties. Flash memory, which is the basic unit of SSD has various unique properties that cause many challenges. Unlike HDD, flash memory does not allow in-place-updating for the block to overwrite [3]. Moreover, each block has a limited number of erase cycles (lifespan), after which the blocks become invalid. Economically, SSD are much more expensive than traditional HDD.

Considering the decrease in the price of SSD, the price gap between SSD and HDD is not likely to disappear in the near future. The cost of SSD is seven to eight times more expensive ($/GB) than that of HDD [4]. With all the benefits of SSD, replacing the conventional HDD drives with pure SSD may not be an efficient option for large-scale storage systems. Thus, a more practical solution is to use hybrid SSD (i.e., SSD consisting of SLC, MLC, and TLC) in hierarchy of storage system, such that the features of this technology are best utilized. The key challenges are to decide what role should each flash memory have in the storage hierarchy, and what data should be stored in it.

Nowadays, there are three types of flash memory that can be found in the marketplace: (i) Single-Level Cell (SLC) where a single bit is stored per cell, (ii) Multiple-Level Cell (MLC) where two bits are stored per cell, and (iii) Triple-Level Cell (TLC) where multiple bits are stored per cell. These three types developed one after the other. Initially the SLC is designed and used for SSD. However, it has two main limitations related to the storage density and price. Consequently, the research community of storage system have shifted their attention toward designing a proper SSD type capable of making up for the shortages in SLC. The new type called MLC, where the storage density has been manipulated through multiple bits to be stored per each cell at a cheap price. However, two shortcomings of the MLC as compared to the SLC are identified: lower performance and reliability of the lifespan. Notably, the lifespan of SLC is 10 times longer than MLC. On the other hand, the novel idea of designing the TLC has been devoted by Samsung, TLC flash memory provide high density and less expensive than SLC and MLC flash memory, which makes it appealing for consumer devices. The main disadvantages of TLC flash memory is the reliability, because it has less number of erase cycles comparing to SLC and MLC.

In this research, we come out with a novel scheme, known as Cost, Performance, and Reliability Concern Scheme (CPRCS) for Hybrid SSD consisting of SLC, MLC, and TLC flash memory, in order to exploit the trade-off between performance, reliability, density, cost, and power consumption. For evaluation purpose of CPRCS scheme, DiskSim [5] with SSD extension from Microsoft was used. The Workloads, which were used, are Financial1, Financial2 [6], and IOzone [7]. The CPRCS scheme is compared with recently proposed schemes: a Round-Robin Frozen Data Collection Algorithm (RRFDCA) scheme [8] and Hybrid SSD (HySSD) [9].

Manuscript received October 20, 2016; revised January 23, 2017.

A. I. Alsalibi, P. Sumari are with the School of Computer Sciences, Universiti Sains Malaysia, Pulau Penang 11800, Malaysia (e-mail: ahmed.salibi@ gmail.com, putras@usm.my).

M. Kh. Shambour is with the Umm Al-Qura University, Mecca, Saudi Arabia (e-mail: shambour@yahoo.com).

DOI: 10.7763/IJCTE.2017.V9.1151

II. Cost, Performance, and Reliability Concern Scheme (CPRCS): The Proposed Scheme

Fig. 1 represents the architecture design of the proposed scheme. CPRCS consists of four parts: (i) *Size Organizer*, (ii) *Data Clustering*, (iii) *Data* Monitor *and Migrator*, and (iv) *Garbage Collector*. The following sections explained the task of each part with further details.

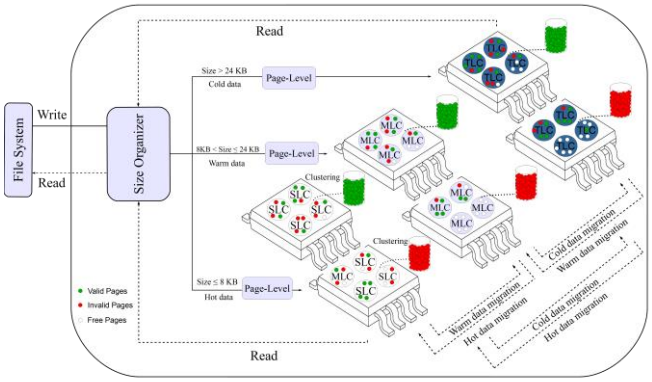


Fig. 1. Storage architecture of CPRCS.

A. Size Organizer

The main idea of *Size Organizer* is to check the size of each write request that has been forwarded from file system to SSD. Based on the previous study of Chang scheme [10]. The write request with small size always refers to hot data and the write request with large size always refers to cold data. Based on this assumption, the write request will be forwarded to SSD portions as follows: (i) In case the request size is smaller than or equal 8 KB, the *Size Organizer* will pass the request to SLC portion. (ii) In case the request size is more than 8 KB and smaller than or equal 24 KB, the *Size Organizer* will pass therequest to MLC portion. (iii) In case the request size is more than 24 KB, the *Size Organiz*er will pass the request to TLC portion.

B. Data Clustering

In rare cases, the size of write request may be larger than 8 KB and the destination is hot data. In contrast, the size of write request may be smaller than 8 KB and its destination is cold data. Since the lifespan of MLC and TLC is smaller than SLC, serving the hot data inside this portions will degrade the lifespan of these portions because hot data is frequently updated by the users, which mean more erase operations will be conducted inside this portion. As a result, the lifespan of hybrid SSD will be degraded.

To deal with such scenario as well as to avoid these hot data to reside in MLC and TLC portions, all the blocks inside these portions will be clustered in idle time by checking the status of the block; either hot or cold periodically. The block with number of valid data more than number of invalid data will be considered as cold and will be gathered to cluster 1.

In contrast, block with number of invalid data more than number of valid data will be considered as hot and will be gathered to cluster 2. Moreover, in order to make the organization of data more accurate and beneficial, each cluster will be divided into two sub-clusters based on the number of erase cycles for each block. Blocks with number of erase cycles more than the average of erase cycles of all the blocks inside Cluster1 will be gathered to sub-cluster 1, and blocks with number of erase cycles smaller than the average of erase cycles of all blocks inside cluster1 will be gathered to sub-cluster 2. The same operation will be applied to cluster2.

C. Data Monitor and Migrator

As mentioned before, the main idea of clustering operations is to avoid residing the hot data into MLC and TLC portions, and cold data into SLC portion. For this purpose, we classify the data inside SSD into three types as follow: (i) Hot: refers to data that is frequently updates by the system, (ii) Warm: refers to data that is normal update by the system (iii) Cold: refers to data that is seldom update by the system.

Data Monitor and Migrator will be invoked in idle time periodically to guarantee that all the data inside SLC, MLC, and TLC is hot, warm, and cold respectively. *Data Monitor and Migrator* takes the benefit of data clustering that was explained in the previous section. Since the blocks inside sub-cluster 2 in cluster 1 are considered as cold blocks as they have less number of erase cycles, thus selecting the block from this sub-cluster will increase the accuracy of selecting the most suitable block because all the blocks inside this sub-cluster are considered as cold blocks. The Block will be migrated to MLC portion in case the block is marked as warm. On the other hand, the block will be migrated to TLC in case the block is marked as cold. To check the status of the blocks (either warm or cold), Equation 1 will be applied to each block inside sub-cluster 2 in cluster 1.

$$SLC.Age.Block_{Cluster1} = \sum_{i=1}^{s} age_i \quad (1)$$

where, *s* refers to the number of valid pages inside a block, *age* refers to the elapsed time of the valid page since the latest update. If the *SLC.Age.Block_Cluster1* > *SLC.Age.Average_Cluster1*, the block will be marked as cold and will migrated to TLC.

While, if *SLC.Age.Average_Cluster1* ≥ *SLC.Age.Block_Cluster1* ≥ *SLC.Age.Average_(Cluster1)/2,* the block will be marked as warm and will be migrated to MLC. Otherwise, the blocks will considered as hot and will reside in the SLC portion. *SLC.Age.Average_Cluster1* defined by Equation 2.

$$SLC.Age.Average_{Cluster1} = \sum_{i=1}^{n} age_i \quad (2)$$

where, *n* refers to the number of valid blocks inside sub-cluster 2 in cluster 1, *age* refers to the elapsed time of the block since the latest update. It is worth mentioning that the same steps will be applied with different conditions when migrated the data from MLC portion to SLC and TLC portion and vice versa.

D. Garbage Collector

Number of Free blocks should always be available to receive new write requests. Receiving the write request without availability of free blocks will degrade the performance of hybrid SSD, as all the I/O requests will be blocked until the Garbage Collector (GC) is invoked to create new free blocks. Consequently, in CPRCS, reserved amount of free blocks is always available to serve the upcoming write requests. Predefined threshold is defined to invoke the GC individually at each portion to guarantee that the reserved

amount of free blocks is always available. When the number of free blocks are less than 10 % in any portions of hybrid SSD (either SLC, MLC, or TLC), the GC is invoked to create a free block. Similar to the *Data Monitor and Migrator*, the GC takes the benefits of *Data Clustering* as explained in section *B*. The GC is accomplished based on the concern of the hybrid SSD. When performance (i.e., cleaning cost) is considered, the scheme prefers the victim blocks located in the sub-cluster 1 in cluster 2, and block with large amount of invalid data will be selected for erase. Thus, some valid data is moved to other free blocks before the erase operation, and the cleaning cost is decreased. However, when the hybrid SSD is more aware of wear leveling (i.e., number of erase cycles) than performance, which in this case is less important, the scheme will select the victim block from the sub-cluster 2 in cluster 1, and the block with less number of erase cycles will be selected for erase. The question that arises at this point is: how do we know the awareness of hybrid SSD? This can be calculated based on the wearing of flash memory portions using Equation 3.

$$\Delta_{\varepsilon} = \varepsilon_{\max} - \varepsilon_{\min} \tag{3}$$

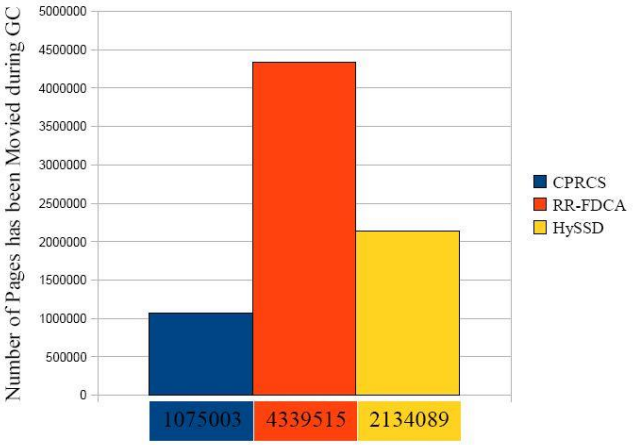
Equation (3) defines Δ_{ε} as the variance between the block with the maximum erasures $\varepsilon_{\max}$ and the block with the least erasures $\varepsilon_{\min}$. Thus, when the wear is high (i.e., variance between $\varepsilon_{\max}$ and $\varepsilon_{\min}$ is not equal 0). That means hybrid SSD is more wear sensitive. Therefore, the block will be selected from sub-cluster 2 in cluster 1. However, if the wear is low (i.e., variance between $\varepsilon_{\max}$ and $\varepsilon_{\min}$ is equal 0). Subsequently, the hybrid SSD becomes more sensitive to performance and weighs it more heavily than WL. In this case, the block will be selected from sub-cluster 1 in cluster 2. CPRCS scheme features a garbage collection method that considers performance and reliability based on the current condition of the hybrid SSD. Another advantage of the victim block score equation is the way it intelligently adjusts between reliability and performance.

III. Experiment Configurations and Results

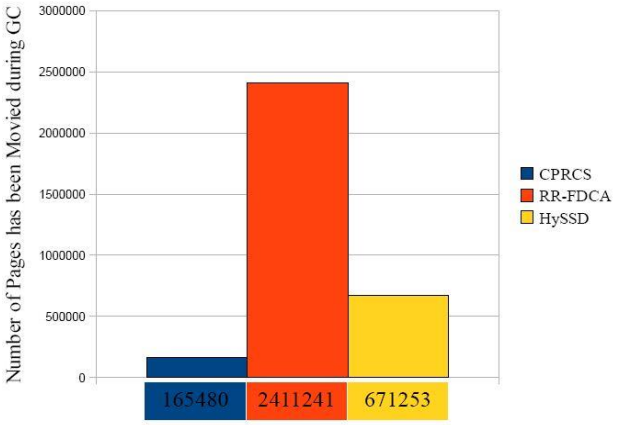
DiskSim [5] and SSD extension from Microsoft were used for evaluating the performance and validity of the CPRCS scheme. The trace files were used to evaluate the validity of the proposed scheme are Financial1, Financial2 [6], and IOzone [7]. Financial1 and Financial2 traces are I/O traces from UMass trace repository that run in two large financial companies, whereas IOzone is a synthetic workload generator. This benchmark creates a large file, and issues different kinds of read/write requests on this file.

A. Pages Moving Overhead during Garbage Collection

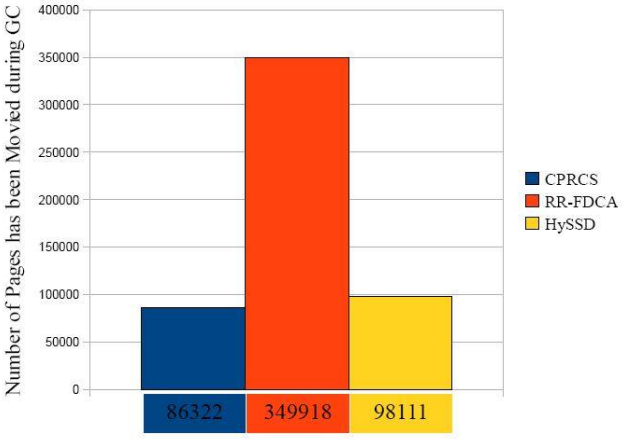
Predefined threshold is configured in order to monitor the number of free blocks inside each portions of hybrid SSD, which should always be available to serve the write requests. When the number of free blocks become smaller than 10% of any portion inside hybrid SSD, the GC will be invoked. The pages moving overhead refer to the valid pages, which were migrated from the victim block to other free blocks before the erase operation is applied to the victim block.



(a) Financial 1 workload.



(b) Financial 2 workload.



(c) IOzone workload.

Fig. 2. Number of pages has been migrated during GC.

Decreasing the number of migrated pages will increase the efficiency of GC operation as time that will be consumed by migrating little number of pages is smaller than the time that will be consumed when the number of migrated pages are more. Furthermore, since the migrated pages need to take place on the other free blocks inside SSD, decreasing the number of migrated pages will decrease the number of free blocks consumption, which in turn will increase the efficiency of serving the upcoming requests without any delay. The number of pages, which were migrated during GC, are 1075003 while using CPRCS, whereas the number of migrated pages amounts to 4339515 and 2134089 while

using RR-FDCA and HySSD respectively, with Financial1 workload as shown in Figure 2a. It means that the CPRCS decreases the number of migrated pages by 120.58% and 66 % as compared to RR-FDCA and HySSD respectively. Moreover, for Financial2 workload, as shown in Figure 2b, RR-FDCA and HySSD also perform worse than CPRCS in this context. 2411241 and 671253 pages were migrated while using RR-FDCA and HySSD respectively, whereas the number of migrated pages while using CPRCS amounts to 165480.

In addition, as shown in Figure 2c, when IOzone workload is used, the number of migrated pages for CPRCS, RR-FDCA, and HySSD amounts to 86322, 349918, and 98111 respectively. The percentage of enhancement in the favor of CPRCS over RR-FDCA and HySSD also amounts to 120.84% and 12.78% respectively. This enhancement is achieved because the data is organized well by CPRCS. All The blocks with smaller amount of valid data will be clustered together and the blocks with larger amount of invalid data will be clustered together. When the GC is invoked, the block with less amount of valid data will be selected, which leads to little number of migrations.

IV. Conclusion

Combining different kinds of memory such as volatile and non-volatile could increase the complexity of managing the hybrid SSD. In contrast, combining different kinds of flash memory sharing the same pin definition and package dimension could be more efficient and simple to manage. As a result, in this research we designed hybrid SSD with three kinds of flash memory SLC, MLC, and TLC in order to exploit the trade-off between (i) performance, (ii) reliability, (iii) capacity, and (iv) cost. The performance and reliability will be increased by enlarging the size of SLC to receive heavy workload and the exact hot data workload which is considered in the first layer of the SSD hierarchy.

As a second layer in the hierarchy of SSD, a medium size of MLC is added and considered to serve the warm data. In the last layer, a medium size of TLC is considered to serve the cold data. As the cost of TLC type is less than that of MLC, adding medium size of this kind to the hierarchy of SSD will shrink the cost of adding extra size of SLC. Moreover, as TLC kind storing three bits per cell, the capacity of SSD will be increased. The objective of this research work is met through achieving its contributions. The main objective is to manage a hybrid SSD that is consists of SLC, MLC, and TLC. This objective is achieved by proposed scheme called CPRCS for hybrid SSD, which consists of SLC, MLC, and TLC. The obtained results of the CPRCS scheme are compared with two hybrid scheme that was recently proposed (i.e., RR-FDCA scheme and HySSD scheme). The experimental results of the DiskSim simulator with Microsoft extension show the superiority of the proposed scheme in terms of number of pages moving overhead during GC. This is an important research domain in the field of storage system that can drive on future developments of the storage architecture research communities to improve the overall performance of the current storage devices.

References

[1] G. Wu and X. He, "Delta-FTL: Improving SSD lifetime via exploiting content locality," In *Proc. of the 7th ACM European Conference on Computer Systems*, pp. 253-266, April 2012.

[2] R. Micheloni, A. Marelli, and K. Eshghi, "Inside solid state drives (SSD)," *Springer Science & Business Media*, vol. 37, 2012.

[3] M. L. Chiang and R. C. Chang, "Cleaning policies in mobile computers using flash memory," Journal of Systems and Software, vol. 48, no. 3, pp. 213-231, 1999.

[4] F. Chen, D. A. Koufaty, and X. Zhang, "Understanding intrinsic characteristics and system implications of flash memory based solid state drives," *In ACM SIGMETRICS Performance Evaluation Review* vol. 37, no. 1, pp. 181-192, 2009.

[5] J. S. Bucy, J. Schindler, S. W. Schlosser, and G. R. Ganger, "The disksim simulation environment version 4.0 reference manual (cmu-pdl-08-101)," *Parallel Data Laboratory*, 2008.

[6] Trace Repository. (2007). Financial1 and Financial2 Traces. [Online]. Available: http://traces.cs.umass.edu, accessed June. 2016.

[7] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. S. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," *In USENIX Annual Technical Conference*, pp. 57-70, 2008.

[8] S. Hachiya, K. Johguchi, K. Miyaji, and K. Takeuchi, "Hybrid triple-level-cell/multi-level-cell NAND flash storage array with chip exchangeable method," *Japanese Journal of Applied Physics*, vol. 53, no. 4S, 2014.

[9] Y. Oh, E. Lee, J. Choi, D. Lee, and S. H. Noh, "Hybrid solid state drives for improved performance and enhanced lifetime," *In 2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, pp. 1-5, 2013.

[10] L. P. Chang, "Hybrid solid-state disks: Combining heterogeneous NAND flash in large SSD," *In 2008 Asia and South Pacific Design Automation Conference*, pp. 428-433, 2008.

Ahmed Alsaibi was born in Gaza, Palestine, on June 19, 1986. He received a B.Sc degree in computer system engineering from Palestine Technical Collage in 2009, and M.S. degree in computer science from Universiti Sains Malaysia. Now, he is a Ph.D. candidate at school of computer sciences, Universiti Sains Malaysia. His research interests mainly focused on computer architecture and storage systems.

Putra Sumari Obtained his M.S. and Ph.D. from Liverpool University, England in 1997 and 2000, respectively. He currently serves as an associate professor, and deputy dean of School of Computer Sciences, Universiti Sains Malaysia. His research interests mainly focused on multimedia tools and applications, multimedia on demand & MPEG technology, image retrieval And compression.

Mohd Shambour received the BSc and MSc degrees in computer science from Yarmouk University and Philadelphia University, Jordan, in 2004 and 2009 respectively, and the Ph. D degree from the School of Computer Sciences, Universiti Sains Malaysia, in October 2014. He is currently an assistant professor with the Custodian of the Two Holy Mosques Institute for Hajj and Umrah Research, Department of Scientific Information and Services, Umm Al-Qura University, Saudi Arabia. His research interests mainly focused on scheduling, storage systems, timetabling and metaheuristic optimization techniques.