

Off-line Deduplication Method for Solid-State Disk Based on Hot and Cold Data

Xin YE, Zhengjun ZHAI, Xiaochang LI

Abstract: Solid-state disk (SSD) deduplication refers to the identification and deletion of duplicate data stored in an SSD. The reliability of SSDs is improved by deduplication. At present, the common data deduplication of SSDs is based on online data deduplication with Field Programmable Gate Array (FPGA) acceleration. The disadvantage is that FPGA, which has a complex structure. An off-line deduplication method for the SSD based on hot and cold data was proposed in this study to simplify the structure of an SSD deduplication, reduce the cost, and improve the efficiency of deduplication and access performance of SSDs. First, the wear-leveling algorithm was employed in the SSD to divide the data into cold and hot. Then, the corresponding fingerprint was generated for the cold data. Second, the fingerprint was compared, and the cold data with the same fingerprint were deleted. Finally, the cold and hot data were exchanged after deduplication. Results demonstrate that the duplicate recognition rate of the proposed method is 5% - 38%, which is close to that of the online deduplication method. In terms of access performance, the performance of SSDs using the proposed method is improved by 20% compared with that of traditional SSDs and is near the access performance of SSDs using online deduplication. This study provides certain reference for improving the reliability of existing SSDs.

Keywords: cold data and hot data; deduplication; fingerprint; off-line; solid-state disk

1 INTRODUCTION

Compared with hard disk drive (HDD), solid-state disk (SSD) with NAND flash as the storage medium has the advantages of high performance, low power consumption, small volume, non-volatility, and good shock resistance [1]. As the price of NAND flash continues to drop, the demand for SSDs rapidly increases. At present, SSDs have begun to replace HDDs as the main data storage device. Simultaneously, the research on SSD-related technologies has also made remarkable progress. An increasing number of enterprises and colleges are participating in the research of SSD-related technologies. Samsung, a South Korean company, developed a 128 - layer stacked 3D NAND flash, with a Non-Volatile Memory express (NVMe) interface SSD capacity of 15 TB.

However, the number of program/erase (P/E) cycles in the NAND flash is limited. For example, the number of P/E cycles of NAND flash in the SLC, MLC, TLC and QLC is approximately 100 K [2], 10 K [3], 1 K and 0.5 K, respectively. SSDs will perform P/E operation on the NAND flash [4, 5], erase the original invalid data in the NAND flash data block, and then write the new data into the data page. Therefore, the reliability of SSDs gradually decreases with the increase in SSDs usage time.

Therefore, scholars have conducted numerous studies on the improvement of SSDs reliability and obtained significant results. These results include wear-leveling, ECC technology, bad block management method, and on-chip redundant arrays of independent drives [6-10]. The applications of these achievements have remarkably improved the reliability and service life of SSDs. However, these methods do not fundamentally solve the reliability reduction problem of SSDs; that is, the actual number of written data has not decreased. Therefore, reducing the actual number of written data is the key to improving the reliability of SSDs.

This study designs an off-line data deduplication method. Identifying and deduplicating the cold data stored in SSDs aim to further reduce the written data into SSDs, decrease the P/E cycles of the NAND flash, and provide a new idea and method for improving the reliability of SSDs.

2 STATE OF THE ART

The main challenges of data deduplication include ensuring that the performance of SSDs is not degraded after deduplication and identifying additional duplicate data. At present, researchers have conducted many works on SSD deduplication. These works mainly focus on improving the efficiency of data deduplication and the recognition of repeated data.

J. Kim [11] et al. first proposed a theoretical model of an SSD data deduplication to prove the feasibility of data deduplication on an SSD theoretically. Simultaneously, a complete SSD deduplication architecture was designed. They realized hardware and software fingerprint generation in SSDs through this architecture. However, they did not optimize the SSD data further, and the recognition of repeated data was still low. Feng Chen [12] et al. designed a content-aware flash translation layer (CAFTL) by improving the FTL. They established a secondary mapping table in the FTL to index fingerprints and improve the efficiency of deduplication. However, the size of the mapping table increased due to the introduction of the secondary mapping table. A. Gupta et al. [13, 14] used the value locality inside SSDs to reduce their write load while realizing deduplication. However, evaluating the efficiency of deduplication is impossible because they did not discuss fingerprint generation. Mohammadamin Ajdari et al. [15, 16] maximally reduced the overhead of system hardware and software resources and finally realized fast identification and removal of duplicate data in SSDs by dynamically adjusting the workload between the FPGA and the CPU. Ajdari proposed an online data deduplication method that could be employed to improve the reliability of SSDs. However, the proposed method has a complex system structure. A. Shimi [17] investigated the operating system layer. The data of more than 2000 users on 15 file servers of a multinational company were analyzed. On the basis of the analysis results, the architecture based on the main data for blocking and removing duplicate data was designed. The architecture could identify a large amount of duplicate data and has a low resource utilization rate. However, this system worked

at the operating system level and was not fully compatible with the existing main file system. Taejin Kim [18] et al. used variable-size deduplication technology to change the original fixed-size deduplication to variable-size deduplication. Additional duplicate data could be identified through this technology. However, this method increased system overhead. M. Jeonghyeon [19] utilized a GPU and CPU to compress and de-duplicate data simultaneously. The method integrated and coordinated multi-core CPU and GPU to improve data deduplication efficiency. Similar to the method proposed by A. Shimi, this method has high cost and poor universality could not be applied to all systems. R. Kapla [20] used a resistive content-addressable memory (ReCAM) and found that ReCAM was used as a cache for deduplication. The deduplication efficiency was close to that of the SSD with online deduplication, and the throughput rate was increased by 100 times. This method used ReCAM, which is expensive and unsuitable for cache in SSDs. Jin Li et al. [21] proposed a storage system based on the chunk stash. The system comprised three parts: RAM, flash, and HDD storage areas. The metadata in the disk was stored in a flash memory, and the retrieval speed was improved by utilizing the high performance of the flash memory. Qian Kai [22] used the secure hash function SHA-3 for fingerprint calculation during data de-duplication and designed a two-level index strategy to improve the search speed and accuracy. However, this method was not only computationally intensive but also caused a rapid increase in the size of fingerprint data. Therefore, this method is unsuitable for SSDs. Zhengyu Yang [23] proposed a deduplication framework applicable to the multi-layer cloud storage architecture comprising SSDs and HDDs. The deduplication parameter was adjusted dynamically to improve efficiency. The complexity of this architecture is not beneficial for improving the reliability of storage systems. Wang Tonglei analyzed the operating system layer. A flash cache system based on data deduplication optimization was designed. However, this system works at the operating system level and is incompatible with the existing mainstream file system; thus, the versatility of the system is weak. Binqi Zhang et al. [24] built a distributed file system based on Hadoop using the CAFTL file system. The system performance was improved by the new routing algorithm. However, this method is only suitable for clustered file systems and could not be used for SSDs. Renhai Chen et al. [25] proposed a resilient aware deduplication algorithm, which could dynamically increase memory resources to improve the overall deduplication performance. This algorithm had the advantages of high duplicate identification rate and small occupation of system resources. However, this method was also unsuitable for embedded systems. Mark Lillibridge [26] focused on solving the problem of excessive fragmentation caused by deduplication and improved system efficiency by caching and other methods. This method is suitable for data backup systems but not for SSDs.

The above studies mainly focused on reducing the data written to SSDs through data deduplication technology to improve the reliability of SSDs. However, although deduplication can reduce data writing, it also decreased the SSD efficiency. Meanwhile, the proposed method has a

complex structure and poor versatility. Therefore, the present study adopted a deduplication method based on hot and cold data. The method is an off-line data deduplication method. Due to the on-line deduplication using FPGA for acceleration, SSD becomes complex, cost increases and reliability decreases. The off-line data deduplication adopted in this study does not require FPGA, which can overcome the disadvantage of on-line deduplication. After the SSD worked normally for a period, hot and cold data could be distinguished by the wear-leveling algorithm in the FTL. Duplicate data could be easily identified by deduplication because cold data generally account for 80% [27-29] of the data stored on SSDs. This condition results in the reduction of the SSD write amplification during cold and hot data exchange. Fewer duplicate data are available due to the frequent updates of hot data; therefore, hot data were not duplicated.

The remainder of this study is organized as follows. The third section describes the principle and implementation of the off-line deduplication method for SSDs based on hot and cold data. The fourth section discusses the relevant tests conducted on the method and obtains the deduplication efficiency and access performance of the method under different workloads. The last section presents the conclusions.

3 METHODOLOGY

The deduplication method designed in this study belongs to an off-line deduplication method. When the SSD recognizes the cold and hot data, it only judges and removes the duplicate data from the cold data.

3.1 General Framework

In deduplication, the MD5 or SHA-1 hash algorithm is usually used for data fingerprint generation. The SHA-1 algorithm is employed in the proposed method. Compared with the common architecture, the SSD adds a cache to the system in the proposed method. The cache is used for storing hash values generated by cold data. The system architecture is shown in Fig. 1.

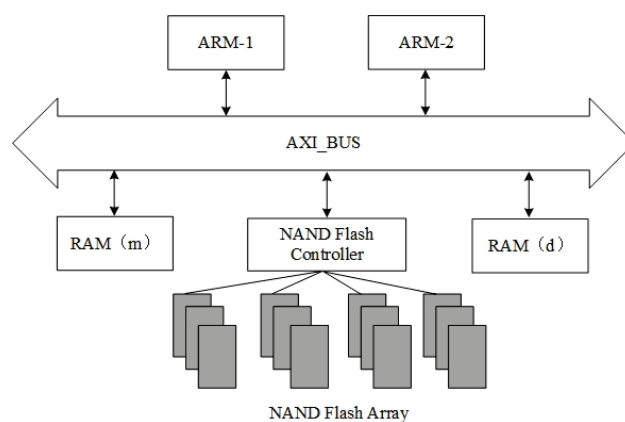


Figure 1 General framework

Fig. 1 shows two groups of RAM: RAM (m) is used to store address mapping tables and RAM (d) is used to store hash values generated from cold data. ARM-1 is a CPU that manages the FTL, including address mapping, wear-leveling, and garbage collection. ARM-2 is mainly used for

cold data fingerprint generation and data comparison. When deduplication is required, ARM-1 stops working and ARM-2 begins to work. After completing fingerprint generation, ARM-2 stops working, and ARM-1 carries out corresponding address mapping according to the result generated by fingerprint. The general framework also includes an interface protocol module and an NAND flash control module.

3.2 Principle

This section introduces the principle of the off-line deduplication method based on hot and cold data from four aspects. First, the structure of the address mapping table in the deduplication method based on cold and hot data is presented. Second, the data deduplication in the SSD is discussed. Third, special cases are handled with deduplication. Fourth, the realization of an efficient garbage collection using the proposed method is discussed.

3.2.1 Address Mapping Table

The address mapping table of the SSD must be modified to meet the requirements of this study because cold data must be deduplicated. The modified address mapping table is shown in Tab. 1

Table 1 Structure of address mapping table

LBA	PBA1	PBA2	FTBA	EC	ST	P
-----	------	------	------	----	----	---

The address mapping table adopts a page-level mapping method. The address mapping table contains information, such as physical block address (PBA), logical block address (LBA), data state (ST), footprint block address (FTBA), the erase count of the block (EC), and data dependency.

LBA represents the logical page address of the data transmitted by the host. Through the corresponding address mapping algorithm, the SSD converts the LBA into the corresponding PBA1. PBA2 is a sequential and constant value. ST indicates the state of the physical page corresponding to PBA2, including free, valid, and invalid states. The physical page corresponding to PBA2 is in the free state with the absence of data. When PBA2 writes data to the corresponding physical page, the page is in valid state. When the physical page data corresponding to PBA2 is updated or deduplicated, the value is in the invalid state. The physical pages in the invalid state are finally collected and returned to the free state. EC records the number of times each block is erased. Parameter k is defined here. When EC is larger than k , the data on the physical page corresponding to EC is considered hot data. Otherwise, it is cold data. Before deduplication, a hash value is generated for each cold data. When the hash value is judged to be equal, the data corresponding to the hash value is duplicate data. The hash value of cold data is stored in the over-provisioning (OP) of the NAND flash. Thus, the corresponding mapping relation must be established. The SSD cannot compare hash values of all cold data together because cold data is larger than hot data. Given that the duplicate data is close in time (9), parameter P, which records the correlation of data per page, is introduced. Tab. 2 is the address mapping table before deduplication.

Table 2 Address mapping table before deduplication

LBA	PBA1	PBA2	FTBA	EC	ST	P
0001	0001	0001	0000	128	Valid	1
0002	0002	0002	0000	251	Valid	1
0003	0003	0003	0000	100	Valid	2

LBA is the data corresponding to 0001 and 0002, and their corresponding P is 1, which indicates that the data on the two physical pages is close in time. FTBA is 0000, which means that the corresponding physical data has not generated hash values.

3.2.2 Deduplication

The SSD will not perform deduplication during data storage in the initial state. The writing and reading are the same as those in ordinary SSDs. When an SSD works for a time, each block has different numbers of erasures. At this time, the wear-leveling algorithm in SSD can already identify hot and cold data according to the different erasure numbers of each block. When the number of erasures exceeds the pre-set K value, the SSD starts the corresponding static wear-leveling algorithm. After this algorithm is started, instead of directly exchanging the hot and cold data, the cold data, which is close in time, is read out from the NAND flash according to the P value and inputted into the fingerprint generation module to generate hash. Then, the generated hash values are compared. If the hash values are the same, then the data are deduplicated.

For example, Tab. 3 is an address mapping table for the SSD that has been in operation for a period. The EC corresponding to data with LBA of 0004 and 0005 is larger than K (assuming that k is 280), indicating that they are thermal data while the rest is cold data.

Table 3 Address mapping table that works for a period

LBA	PBA1	PBA2	FTBA	EC	ST	P
0001	0001	0001	0000	217	Valid	1
0002	0002	0002	0000	251	Valid	1
0003	0003	0003	0000	209	Valid	1
0004	0004	0004	0000	290	Valid	2
0005	0005	0005	0000	330	Valid	3

At this time, according to the P value, the SSD first reads the data in PBA1 of 0001, 0002, and 0003 into the fingerprint generator to generate hash. Given that the data of PBA1, 0001 and 0002 is the same, the hash values are also similar. After the fingerprint comparison is completed, the address mapping table is modified for the first time. The modified address mapping table is shown in Tab. 4.

Table 4 First modified address mapping table

LBA	PBA1	PBA2	FTBA	EC	ST	P
0001	0001	0001	1001	217	Valid	1
0002	0001	0002	0000	251	Invalid	0
0003	0003	0003	1002	209	Valid	1
0004	0004	0004	0000	290	Valid	2
0005	0005	0005	0000	330	Valid	3

The modified contents in the mapping table include the following. 1) PBA1 corresponding to LBA 0002 is modified to 0001 such that two different logical pages correspond to the same physical page. 2) Fingerprint information is generated for the data with LBA of 0001 and 0003 such that the physical address corresponding to fingerprint information is added to FTBA. 3) The state of

the physical page corresponding to PBA2 is modified to Invalid because the data of the physical page are detected to be duplicate data. Hot and cold data can be exchanged after completing the first mapping table modification. The exchange is divided into two parts. The first part is the second modification of the address mapping table. The second modified address mapping table is shown in Tab. 5. The second part is the exchange of NAND flash data. The process of NAND flash data exchange is not discussed in this study

Table 5 Address mapping table that works for a period

LBA	PBA1	PBA2	FTBA	EC	ST	P
0001	0004	0001	1001	0	Valid	1
0002	0004	0002	0000	0	Invalid	0
0003	0005	0003	1002	0	Valid	1
0004	0001	0004	0000	0	Valid	2
0005	0003	0005	0000	0	Valid	3

3.2.3 Special Case Handling

Under normal circumstances, an SSD performs static wear-leveling and cold data deduplication only when it is idle. However, whether the host will send a reading or writing request to the SSD during the above work is uncertain. Therefore, the SSD must deal with the above special cases. The following are discussions under the condition of static wear-leveling and cold data deduplication during idle states of the SSD.

(1) When the host performs thermal data processing. First, the SSD transfers all cold data on the bus to the fingerprint generator. Second, fingerprint generation simultaneously yields information for the existing data and updates the thermal data. Finally, the deduplication is completed after updating the hot data.

(2) When the host updates the cold and fast data. First, the SSD transfers all the cold data on the bus to the fingerprint generator. Second, whether the updated cold data are currently in the fingerprint generator is determined. If the data are not in the fingerprint generator, then the cold data in the NAND flash are directly updated. If the fingerprint is already generated, then updating the corresponding cold data in the fingerprint device and the NAND flash is necessary to avoid inconsistent data in the hot and cold data exchange.

(3) When the host reads data. The SSD transfers all the cold data on the first bus to the fingerprint generator because data updating is not involved in data reading. Then, the corresponding data from the NAND flash are read to the host.

3.2.4 Garbage Collection

Given that the proposed method is an off-line deduplication, the judgment and removal of duplicate data are performed only after the data have been written into the NAND flash. The method removes duplicate data by changing the address mapping relationship and marking the physical pages of duplicate data as invalid. Thus, the number of invalid blocks will increase, raising the frequency of garbage collection. A high deduplication rate of duplicate data leads to a high garbage collection frequency. This phenomenon will not only reduce the system performance but also increase the erasing times of

the NAND flash. The SSD OP is added herein to address the problem of frequent garbage collection caused by the marking of duplicate data as invalid pages and reduce the garbage collection frequency. Additional space is reserved.

4 RESULT ANALYSIS

The Altera SoC platform is used to conduct a practical test and compare the test results with those using the method designed by Jonghwa Kim and an ordinary SSD. An Altera A10 series SoC (model SX066N3F40I2LG) is adopted in this platform. CPU DDR capacity is 1 GB, and FPGA DDR capacity is 2 GB.

4.1 Duplication Rate Test

Three kinds of traces are considered to test the duplication rate of the proposed method accurately. Tab. 6 provides a detailed description of the three types of traces.

Table 6 Details of three trace categories

Trace Number	Trace Type	Trace quantity	Trace Size
Trace1	WinXP install	1	1.6 GB
Trace 2	Office Work	68	132 MB
Trace 3	Outlook	108	3.9 GB

Fig. 2 shows the test result of the algorithm proposed by Kim considering the recognition degree of duplicate data. The test results in Kim are directly used because his method cannot be completely reproduced. No comparison test will be conducted because ordinary SSDs do not have a deduplication function. The test results show that the recognition rate of this algorithm is lower than that of the method proposed by Kim because the proposed method does not deduplicate all data but only the cold data in the SSD.

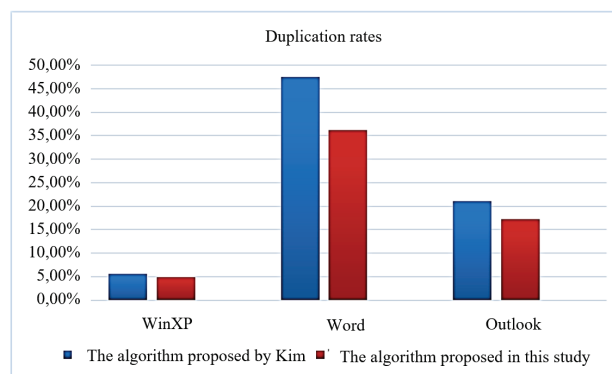


Figure 2 Duplication rates of traces

Given that the WinXP system is installed only once in Trace1, it will immediately become cold data if it does not perform numerous operations. Therefore, the duplication rate of the proposed method is close to that of Kim's algorithm in the Trace1 test.

Trace2 has many requests for office documents and fewer cold data. Therefore, the duplication rate of this algorithm will decrease.

In Trace3, Outlook files have a lower proportion of cold data than that of WinXP and a higher proportion than office documents. Therefore, the duplication rate also presents corresponding results.

4.1 Performance Test

The performance of SSDs may be reduced after adding deduplications because data fingerprint generation and comparison are included in deduplication. Fig. 3 presents the write latency performance of the proposed method, the method proposed by Jim, and an ordinary SSD without deduplication. The traces follow the traces in Tab. 5.

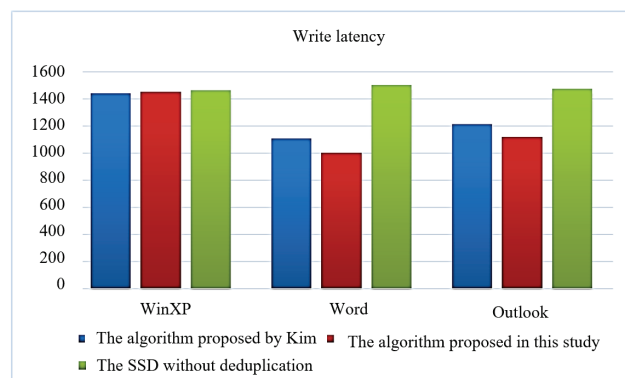


Figure 3 Write latency performance test results

According to the write latency performance test results, the write latency of the three methods is similar when the deduplication rate is low. A high deduplication rate leads to a low write latency of the proposed method. This phenomenon is due to the proposed method, which belongs to off-line deduplication and hardly affects the normal data read-write process. Thus, the performance of this method is higher than that of ordinary SSDs. Jim's method uses hardware acceleration, and its performance is higher than that of ordinary SSDs. As the SSD usage time increases, Jim's method must deduplicate all data with the increase in an SSD usage time. However, the hot data update frequency is high and the deduplication rate is low. The deduplication rate of cold data is high, and the efficiency can be improved for this part of deduplication. Overall, the proposed method is superior to Jim's algorithm in performance.

In the sequential write test, only the proposed method is compared with the ordinary SATA interface SSD because the sequential write performance test of Jim's algorithm cannot be obtained. The comparison results are shown in Tab. 7

Table 7 Result of sequential write test

Sequential write data size (MB)	256	512	1024	2048	10240
Ordinary SSD write Performance (MB/s)	451	448	455	455	460
Write performance of the proposed method (MB/s)	540	540	538	533	520

The comparison results indicate that when the amount of test data is small, the sequential writing performance of the proposed method is close to that of the ordinary SATA interface SSD. The sequential writing performance of the proposed method is reduced when the amount of data is large because duplication easily occurs and the algorithm must deduplicate the data. Therefore, continuous sequential performance is affected to some extent.

5 CONCLUSION

This study designed an off-line deduplication method for SSDs based on hot and cold data to simplify the structure of an SSD data deduplication, reduce costs, and improve data deduplication efficiency and the SSD performance. This proposed method uses the static wear-leveling algorithm in an SSD to identify cold data and discriminates and removes duplicate data. Meanwhile, the frequency of garbage collection due to data deduplication is reduced by increasing the OP space inside the SSD. The following conclusions could be drawn.

(1) The proposed method belongs to an off-line deduplication method. The method is similar to an SSD using online deduplication in deduplication rate.

(2) Compared with the traditional SSD, the read-write performance of the proposed method is improved and is close to the online deduplication of SSDs accelerated by FPGA.

This study provides a certain reference for improving the reliability of SSDs on the premise of ensuring the read-write performance of SSDs. The above test results were obtained without FPGA acceleration. Therefore, the proposed method for deduplication has the advantages of high deduplication efficiency and simple system structure. Meanwhile, further reduction of the excessive garbage collection operations caused by off-line deduplication is a key issue that must be addressed in the future.

Acknowledgements

This work was supported by the Shaanxi Science and Technology Coordination and Innovation Project (No. 2016TZC-G-8-3).

6 REFERENCES

- [1] Matsui, C., Arakawa, A., & Takeuchi, K. (2017). Write Order-Based Garbage Collection Scheme for an LBA Scrambler Integrated SSD. *IEEE Transactions on Very Large Scale Integration System*, 25(2), 510-519. <https://doi.org/10.1109/TVLSI.2016.2594200>
- [2] Li, H., Gui, C., & Xiao, K. (2018). Heterogeneity-aware ECC designs for MLC SSD. *Journal of HuaZhong Normal University*, 52(4), 483-488. <https://doi.org/10.19603/j.cnki.1000-1190.2018.04.008>
- [3] Kim, J., Lee, E., Choi, J., Lee, D., & Noh, S.H. (2016). Chip-Level RAID with Flexible Stripe Size and Parity Placement for Enhanced SSD Reliability. *IEEE Transactions on Computers*, 65(4), 1116-1130. <https://doi.org/10.1109/TC.2014.2375179>
- [4] Lee, J., Ganesh, K., Lee, H. J., & Kim, Y. (2017). FeSSD: A Fast Encrypted SSD Employing On-Chip Access-Control Memory. *IEEE Computer Architecture Letters*, 16(2), 115-118. <https://doi.org/10.1109/LCA.2017.2667639>
- [5] Gupta, A., Pisolkar, R., Uргаonkar, B., & Anand, S. (2011, February). Leveraging Value Locality in Optimizing NAND Flash-based SSDs. *Proceedings of the 9th USENIX conference on File and Storage Technologies*, San Jose, California, NJ: USENIX.
- [6] Fan, N., Wu, X. B., Li, W. J., Wang, L., & Jiang, S. (2018). WOJ: Enabling Write-Once Full-data Journaling in SSDs by using weak-hashing-based deduplication. *Performance Evaluation*, 127(128), 56-69. <https://doi.org/10.1016/j.peva.2018.09.004>
- [7] Yang, L., Cao, X., & Geng, X. (2019). A novel intelligent assessment method for SCADA information security risk

- based on causality analysis. *Cluster Computing*, 22(Suppl 3), 5491-5503. <https://doi.org/10.1007/s10586-017-1315-4>
- [8] McEwan, A. A. & Muhammed, Z. K. (2017). Reliability and performance enhancements for SSD RAID. *Microprocessors & Microsystems*, 52, 461-469. <https://doi.org/10.1016/j.micpro.2016.11.012>
- [9] Li, Y. K., Lee, P. C., & Lui, C. S. (2014). Analysis of Reliability Dynamics of SSD RAID. *IEEE Transactions on Computers*, 65(4), 1131-1144. <https://doi.org/10.1109/TC.2014.2349505>
- [10] Li Y., Cao X., & Li J. (2016). A new cyber security risk evaluation method for oil and gas SCADA based on factor state space. *Chaos Solitons & Fractals*, 89, 203-209. <https://doi.org/10.1016/j.chaos.2015.10.030>
- [11] Kim, J., Lee, C., & Lee, S. (2012, April). Deduplication in SSDs: Model and Quantitative Analysis. *28th Symposium on Mass Storage Systems and Technologies*, San Diego, CA, NJ: IEEE. <https://doi.org/10.1109/MSST.2012.6232379>
- [12] Chen, F., Luo, T., & Zhang, X. D. (2011, February). CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives. *Proceedings of the 9th USENIX conference on File and Storage Technologies*, San Jose, CA, NJ: USENIX.
- [13] Mao, B., Jiang, H., & Wu, S. Z. (2015). Leveraging Data Deduplication to Improve the Performance of Primary Storage Systems in the Cloud. *IEEE Transactions on Computers*, 65(6), 1175-1188. <https://doi.org/10.1109/TC.2015.2455979>
- [14] Neves, P., Ferreira, P., & Barreto, J. (2013, August). Leveraging Web Prefetching Systems with Data Deduplication. *12th International Symposium on Network Computing and Applications*, Cambridge, MA, NJ: IEEE. <https://doi.org/10.1109/NCA.2013.47>
- [15] Ajdari, M., Park, P., Kim, J., Kwon, D., & Kim, J. (2019, February). CIDR: A Cost-Effective In-line Data Reduction System for Terabit-per-Second Scale SSD Arrays. *25th IEEE International Symposium on High Performance Computer Architecture*, Washington, DC, NJ: IEEE. <https://doi.org/10.1109/HPCA.2019.00025>
- [16] Ajdari, M., Park, P., Kwon, D., & Kim, J. (2017). A Scalable HW-Based Inline Deduplication for SSD Arrays. *IEEE Computer Architecture Letters*, 17(1), 47-50. <https://doi.org/10.1109/LCA.2017.2753258>
- [17] El, S. A., Kalach, R., Kumar, A., Oltean, A., Li, J., & Sengupta, S. (2012, June). Primary Data Deduplication - Large Scale Study and System Design. *Proceedings of the 2012 USENIX conference on Annual Technical Conference*, Boston, MA, NJ: USENIX.
- [18] Kim, T., Kim, J., & Lee, S. (2017). FineDedup: A Fine-grained Deduplication Technique for Extending Lifetime of Flash-based SSDs. *Journal of Semiconductor Technology and Science*, 17(5), 648-659. <https://doi.org/10.5573/JSTS.2017.17.5.648>
- [19] Ma, J. & Park, C. (2017, September). Parallelizing Inline Data Reduction Operations for Primary Storage Systems. *14th International Conference on Parallel Computing Technologies*, Nizhny, Novgorod, NJ: PaCT. https://doi.org/10.1007/978-3-319-62932-2_29
- [20] Kaplan, R., Yavits, L., Morad, A., & Ginosar, R. (2016, September). Deduplication in Resistive Content Addressable Memory Based Solid State Drive. *26th International Workshop on Power and Timing Modeling, Optimization and Simulation*, Bremen, Germany, NJ: IEEE. <https://doi.org/10.1109/PATMOS.2016.7833432>
- [21] Li, J., Debnath, B., & Sengupta, S. (2010, June). ChunkStash: Speeding up Inline Storage Deduplication using Flash Memory. *Annual Technical Conference*, Boston, MA, NJ: USENIX.
- [22] Yan, C. & Qian, K. (2018). Similarity-Based Client-Server Data Deduplication Method in Cloud Storage. *Journal of Donghua University*, 44(1), 115-122. <https://doi.org/10.3969/j.issn.1671-444.2018.01.017>
- [23] Yang, Z. Y., Wang, Y. F., Bhamini, J., Tan, C. C., & Mi, N. F. (2018). EAD: elasticity aware deduplication manager for datacenters with multi-tier storage systems. *Cluster Computing*, 21(3), 1561-1579. <https://doi.org/10.1007/s10586-018-2141-z>
- [24] Zhang, B. Q., Wang, C., Zhou, B. B., & Albert, Y. Z. (2015, January). Inline Data Deduplication for SSD-Based Distributed Storage. *21st International Conference on Parallel and Distributed Systems*, Melbourne, VIC, NJ: IEEE. <https://doi.org/10.1109/ICPADS.2015.80>
- [25] Wang, Y. F. & Tan, C. C. (2014, December). Using Elasticity to Improve Inline Data Deduplication Storage Systems. *7th International Conference on Cloud Computing*, Anchorage, AK, NJ: IEEE. <https://doi.org/10.1109/CLOUD.2014.109>
- [26] Lillibridge, M., Eshghi, K., & Bhagwat, D. (2013, February). Improving restore speed for backup systems that use inline chunk-based deduplication. *Proceedings of the 11th USENIX conference on File and Storage Technologies*, San Jose, California, NJ: USENIX.
- [27] Chen, R. H., Wang, Y., Liu, D., Shao, Z. L., & Jiang, S. (2016). Heating Dispersal for Self-Healing NAND Flash Memory. *IEEE Transactions on Computers*, 66(2), 361-367. <https://doi.org/10.1109/TC.2016.2595572>
- [28] Panigrahi, K. S., Maity, C., & Gupta, A. (2014). A simple wear leveling algorithm for NOR type solid storage device. *CSI Transactions on ICT*, 2(1), 65-76. <https://doi.org/10.1007/s40012-014-0047-3>
- [29] Yang L., Geng X., Cao X. (2016). A novel knowledge representation model based on factor state space. *Optik - International Journal for Light and Electron Optics*, 127(12), 5141-5147. <https://doi.org/10.1016/j.ijleo.2016.02.074>

Contact information:

Xin YE, Master,
(Corresponding author)
School of Computer Science and Engineering,
Northwestern Polytechnical University Xi'an,
127 West Youyi Road, Beilin District,
Xi'an Shaanxi, 710072, P. R. China
E-mail: yexinisphere@163.com

Zhengjun ZHAI, Master,
School of Computer Science and Engineering,
Northwestern Polytechnical University Xi'an,
127 West Youyi Road, Beilin District,
Xi'an Shaanxi, 710072, P. R. China
E-mail: zhengjunzhai@139.com

Xiaochang LI, Master,
School of Computer Science and Engineering,
Northwestern Polytechnical University Xi'an,
127 West Youyi Road, Beilin District,
Xi'an Shaanxi, 710072, P. R. China
E-mail: lxcwilliam@gmail.com