Machine Learning Assisted OSP Approach for Improved QoS Performance on 3D Charge-Trap Based SSDs

Zongwei Zhu[1], Chao Wu[2(✉)], Cheng Ji[2], and Xianmin Wang[3]

[1] Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou 215123, China
[2] Department of Computer Science, City University of Hong Kong, Hong Kong 999077, China
chaowu6-c@my.cityu.edu.hk
[3] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China

Abstract. 3D charge-trap based SSDs have become an emerging storage solution in recent years. One-shot-programming in 3D charge-trap based SSDs could deliver a maximized system I/O throughput at the cost of degraded Quality-of-Service performance. This paper proposes RLOSP, a reinforcement learning based approach to improve the QoS performance for 3D charge-trap based SSDs. By learning the I/O patterns of the workload environments as well as the device internal status, the proposed approach could properly choose requests in the device queue, and allocate physical addresses for these requests during one-shot-programming. In this manner, the storage device could deliver an improved QoS performance. Experimental results reveal that the proposed approach could reduce the worst-case latency at the 99.9^{th} percentile by 37.5–59.2%, with an optimal system I/O throughput.

Keywords: 3D charge-trap based SSD · One-shot-programming · Reinforcement learning · I/O throughput · QoS performance

1 Introduction

With the rapid development of high-density NAND flash technology, 3-dimensional (3D) SSDs have been employed as the prevalent storage solution in the market. There are two mainstream 3D SSD technologies, floating-gate (FG) technology and charge-trap (CT) technology [21]. CT-based technology has been deemed as a pre-dominant candidate since it eliminates the cell-to-cell interference [21]. Thus, multiple pages in a word line could be programmed simultaneously, which is called one-shot-programming (OSP), e.g., 3 pages for TLC-based SSDs. This technique can maximize the system I/O throughput, thus improving the user experience significantly.

The original version of this chapter was revised: Subfigures (g) and (h) in Fig. 7 have been removed. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-62463-7_51

X. Chen et al. (Eds.): ML4CS 2020, LNCS 12488, pp. 86–99, 2020.
https://doi.org/10.1007/978-3-030-62463-7_9

While OSP technique is recommended for superior system I/O throughput, this study identifies that OSP could significantly degrade the quality-of-service (QoS) performance. QoS demands all I/O latency to be constrained within a specific range [20]. Although SSDs provide a faster accessing speed comparing with HDDs, they suffer from large I/O performance variation and long worst-case latency owing to various reasons, e.g., garbage collection, read/write interference, process variation [6,20]. The enlarged I/O performance variation and prolonged worst-case latency could violate the QoS demand and impact the user experience. As all programmed pages share one completion time, OSP could spread one prolonged worst-case latency to the latency of all programmed pages. Moreover, as the storage device is oblivious to the process information of I/O requests, I/O from different processes might be programmed simultaneously. The prolonged worst-case I/O latency could be spread to different processes by OSP. As a result, OSP could significantly enlarge the I/O performance variation, thus incurring unpredictable I/O latency. Among these I/Os, some might have to experience long worst-case latency, thus violating the QoS demand and impacting the user experience.

Prior works fail to address the QoS performance incurred by OSP on CT-based SSDs. Some works propose to improve the system I/O performance by enhanced internal parallelism [15,21], which does not consider the QoS performance. There are also works proposed to improve the QoS performance by revised garbage collection [2,4,14], or mitigate the impaired storage lifetime incurred by process variation [1,5,17]. In addition, there are ways to optimize HDFS I/O performance in deep learning cloud computing platforms [25] or reduce system I/O latency by offloading computing power at the edge [24]. Since these works do not take account of the unique feature of OSP technique, there still exists a demanding need to enhance the efficacy when leveraging the OSP operations on 3D SSDs.

In this paper, a reinforcement learning based approach, RLOSP, is proposed. Through learning the I/O patterns and the device internal status, RLOSP properly decides which requests in the device queue to be programmed with OSP and the target physical addresses of these requests at each time. The considered I/O patterns consist of process ID and host I/O information of requests in the device queue, and the considered device status include the busy/free status of each flash chip and the space utilization of each plane. In this manner, the storage device could deliver reduced worst-case latency and improved QoS performance. Experimental results show that RLOSP could reduce the standard deviation of I/O latency by 51.8%, the worst-case I/O latency at the 99.9^{th} percentile by 37.5–59.2%, meanwhile delivering an optimal system I/O throughput (1.3% lower) comparing with existing approach. To the best of the author's knowledge, this is the first work proposed to improve the QoS performance of the 3D CT-based SSDs by OSP optimization.

In summary, this paper makes the following contributions:

– Identified that OSP in 3D CT-based SSDs could degrade the QoS performance of the storage device;

– Proposed RLOSP, a reinforcement learning assisted technique to make programming decisions adaptive to various I/O patterns and device internal status;
– Evaluated the proposed approach and verified that the proposed approach manages to effectively improve the QoS performance of the storage devices.

The remainder of this paper is as follows. Section 2 reviews related background and prior works. Section 3 states the problem of OSP technique in 3D CT-based SSDs. Section 4 presents the proposed RL assisted OSP approach. Section 5 evaluates the proposed approach and analyzes the experiment results. Finally, Sect. 6 concludes this paper.

2 Related Work

3D flash storage technology has attracted much attentions during past years. Liu et al. [15] propose to enhance the device internal parallelism by implementing the block-level parallelism and partial page accessing in the storage hardware architecture. Wu et al. [21] propose a distance-aware round robin page allocation scheme for improving the utilization of internal parallelism and thus improving the read performance. There are also works proposed to mitigate the detrimental effects of garbage collection activities [2,4,14], or mitigate the retention error and degraded system performance incurred by the process variation [1,3,6]. However, few of these works target on the degraded QoS performance incurred by the OSP in 3D CT-based SSDs.

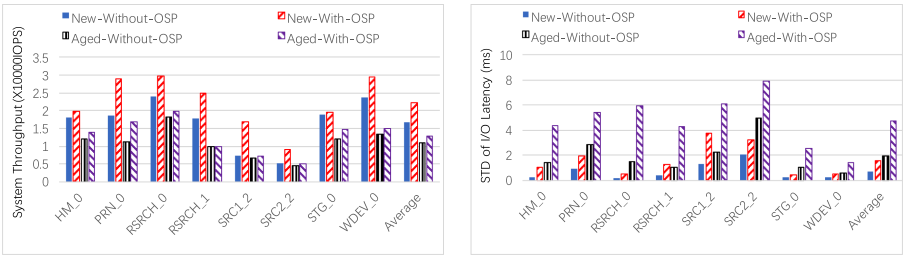
There are also works proposed to improve the QoS performance of SSDs. Some works propose to decrease the I/O performance variation and reduce the worst-case latency by garbage collection oriented optimizations [11,19,22,23]. Gugnani et al. [8] propose a set of strategies for providing QoS performance guarantee with NVMe SSDs on virtual environment. Wu et al. [20] propose a reinforcement learning-assisted I/O merging technique for improved QoS performance on SSDs. However, few considers the QoS violation incurred by OSP in 3D CT-based SSDs.

3 Motivation

To study the advantage and disadvantage of One-Shot-Programming (OSP) on system performance of CT-based SSDs, experiments are evaluated with SSDsim [9]. The simulator is configured to simulate a new TLC SSD and a TLC SSD device aged by 70%. The experimental results are compared with a device without OSP. The configuration of the simulator is described in Sect. 5.

Figure 1 shows the system throughput and I/O performance variation results. By writing three sub-requests into three consecutive pages, OSP can benefit system throughput significantly, as shown in Fig. 1(a). On average, system throughput is improved by 25.1% for a new device with OSP compared with a new device without OSP. For SRC1_2, the improvement of system throughput is 56.3%. However, the benefit of OSP on system throughput decreases as the aging status of the device increases. For a device aged by 70%, system throughput is

improved by OSP by only 14.1% compared with a device without OSP. The reason is three-fold. First, the enlarged programming page number caused by OSP could incur garbage collection and block I/O requests, especially for aged devices with less physical space [10,12]. Then, the I/O requests grouped by OSP operation increases the possibility of read/write interference [13,18]. Finally, the grouped I/O requests may spread the prolonged latency of all blocked I/O requests, which degrades system performance further.



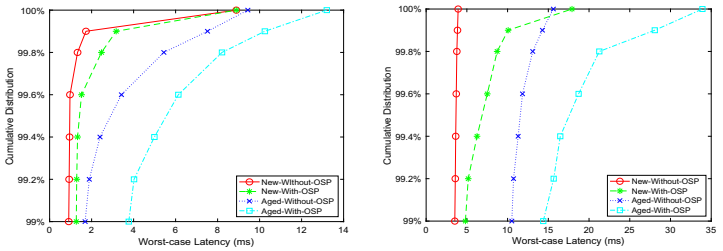
(a) System Throughput (b) Standard Deviation of I/O latency

Fig. 1. Comparison Results of 3D SSD with and without One-Shot-Programming (OSP). New-Without-OSP means a new SSD without OSP, New-With-OSP means a new SSD with OSP, Aged-Without-OSP and Aged-With-OSP mean a SSD with and without OSP with 70% physical space used respectively.

The blocked I/O requests may experience a long I/O latency, which incurs enlarged I/O performance variation [20]. As shown in Fig. 1(b), the standard deviation of I/O latency for a SSD with OSP is 55.3% larger than a device without OSP on average for a new device, while the value is 58.9% for a device aged by 70%. This enlarged I/O performance variation makes I/O latency unpredictable, while several I/O may experience a worst-case latency. The unpredictable and extremely long I/O latency could affect user experience and violates the quality-of-service (QoS) requirement.

Figure 2 shows the comparison results of worst-case latency of example traces between SSDs with and without OSP operation. At the 99.9% percentage, the worst-case latency for a device with OSP is 45.4% for HM_0 and 61.8% for PRN_0 higher than a device without OSP for a new device. On average among all 8 adopted traces, the worst-case latency is prolonged by OSP by 44.2–67.8%, 55.3% on average at the 99.9% percentage for a new device. For a device aged by 70%, the worst-case latency is prolonged by OSP by 26.6–80.3%, 56.1% on average at the 99.9% percentage. Experimental results verified that OSP operation significantly prolongs worst-case latency and degrades QoS performance.

In summary, OSP in 3D charge-trap based SSD could improve system I/O throughput at the cost of prolonged worst-case latency and degraded QoS performance. The key observation here is that there is a need to propose a new approach to mitigate the worst-case latency while delivering a maximum I/O throughput during the adoption of OSP.



(a) HM_0 (b) PRN_0

Fig. 2. Worst-case Latency of Example Traces. The horizontal axis is I/O latency in millisecond, the vertical axis is the cumulative distribution of I/O latency.

4 Methodology

This work proposes RLOSP, a reinforcement learning based approach to mitigate the worst-case latency issue incurred by one-shot-programming in CT-based SSDs. RLOSP learns from the I/O pattern and the storage internal status, and make OSP decisions accordingly. To avoid the worst-case I/O latency from spreading to different host I/Os and processes, the process ID (PID) and host I/O information of requests in the device queue is considered. The PID of requests is considered to avoid the worst-case I/O latency from spreading to various processes. Although current storage device is oblivious to this information, it is convenient to acquire this information from the host side with open-channel SSD technology [16] with trivial overhead. The host I/O information is considered to avoid the worst-case I/O latency from spreading to different host I/Os. To record the PID and host I/O information of each request in the device queue, a FIFO-list is maintained which is synchronized with the maintenance of all requests in the device queue. The considered storage internal status consists of busy/free status of flash chips and the space utilization of each plane. Requests programmed to the busy chips by OSP could be blocked until the chips are free, thus incurring a long I/O latency. Moreover, programming I/O requests to the planes with high space utilization might incur garbage collection, which could significantly degrade the I/O latency. The enlarged number of programmed pages in OSP increases the possibility of triggering garbage collection. Therefore, the busy/idle status of flash chips and the space utilization of each plane are considered in RLOSP.

Figure 3 describes the architecture of RLOSP. When a new request is inserted to the device queue (RQ1), the device controller splits this request into several one-page transactions. Then, the agent in RLOSP collects the PID and host I/O information of requests in queue from the storage device controller, as well as the busy/idle status of each flash chip and the space utilization of each plane ②. Accordingly, the agent identifies the current state S_i. After that, the agent refers to the $\varepsilon - Greedy$ policy to decide to perform exploration or exploitation ③. An action A_i is selected according to the current state in the Q-table ④,

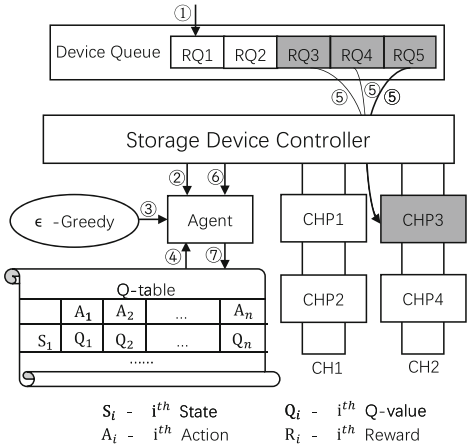


Fig. 3. Architecture of the proposed framework. CHP is the flash chips, CH is the device channel. RQ is the request in device queue.

which chooses requests in queue to be programmed with OSP (RQ3, RQ4 and RQ5, assume all are one-page requests) and assigns physical addresses for these requests. These requests are then programmed to the assigned addresses in the target flash chip (CHP3) with OSP ⑤. After the latency of the programmed requests is perceived by the storage device controller, the agent acquires the latency from the storage device controller ⑥ and calculates the reward R_i of the action A_i. Finally, the Q-value Q_i is calculated and updated into the corresponding state-action entry in the Q-table ⑦.

4.1 Model Construct

State Space. The state space definition should provide sufficient information of the environment, so that the agent could fully learn from the environment and make OSP decisions accordingly. In RLOSP, the state definition considers storage internal status, including busy/idle status of each flash chip and the space utilization of each plane. In this manner, the OSP node could be assigned to a free chip and a plane with low space utilization, so as to boost the OSP node and improve the QoS performance. The space utilization of each plane is classified in to several regions (3 by default) in state definition.

The number of the state is illustrated as Eq. 1 shown.

$$S_n = 2^{CHP} \times SO^P \tag{1}$$

In which CHP is the number of flash chips. 2^{CHP} is the number of busy/idle status of all flash chips. P is the number of planes in the back-end storage device. SO^P means the space utilization region distribution of all planes.

Action Space. The action space is defined by choosing requests in the device queue to be programmed by OSP and assigning physical addresses for these requests. There are two rules in action selection. Firstly, transactions from the same host I/O are programmed with highest priority in the same OSP node to avoid the worst-case latency from spreading to different host I/O. Then, transactions with the same PID are programmed with moderate priority in the same OSP node to avoid the worst-case latency from spreading to various processes. Finally, transactions with different PID are programmed with lowest priority in the same OSP node. In this manner, the action is defined by choosing A transactions from I/O requests in the device queue, and assigning physical address for these transactions. A is the number of transactions programmed by OSP each time.

Reward. The reward in RL is defined to represent the correctness of the last action. In RLOSP, improper OSP operations could incur long worst-case latency thus violating the QoS demand and impacting the user experience. The reward is defined by the I/O latency of the transactions programmed by last OSP action.

Figure 4 describes the reward definition in RLOSP. The I/O latency of each OSP operation is classified into three regions, as shown in the figure. For OSP actions with low latency, a bonus is assigned as the reward. For actions with medium latency, the reward is none. A minus reward is assigned for actions with long latency.

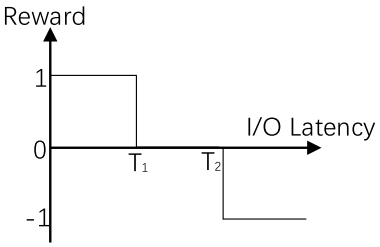


Fig. 4. Reward definition.

4.2 Algorithm

Algorithm 1 shows the algorithm of RLOSP. Four groups of elements are input in RLOSP, including the PID of each request in the queue $G(PID_1, .., PID_n)$, the host I/O information of each request in the device queue $G(HI_1, .., HI_n)$, busy/idle status of each chip $G(S_1, .., S_{CHP})$ and the space utilization of each plane $G(SO_1, .., SO_P)$. n is the number of requests in queue, P is the number of planes in the flash storage and CHP is the number of flash chips. When the device controller prepares for an OSP operation, the agent picks an action

Algorithm 1. RL-Based OSP Approach

Require:
$G(PID_1, .., PID_n), G(HI_1, .., HI_n)$,
$G(S_1, .., S_{CHP}), G(SO_1, .., SO_P)$
Ensure: A_t
if $Data/Eviction$ **then**
$S_t = state\_identify(G(PID_1, .., PID_n)$,
$G(HI_1, .., HI_n), G(S_1, .., S_{CHP}), G(SO_1, .., SO_P))$;
$A_t = \varepsilon - Greedy(S_t)$;
$Perform\ OSP$;
$Acquire\ I/O\ latency$;
$Reward = Cal\_Reward(T_{Lat})$;
$Q(s, a) = Q(s, a) + \gamma * Reward$;
end if

referring to $\varepsilon - Greedy$, which means deciding which requests in queue will be programmed with OSP and which physical address will be assigned for these requests. After that, the I/O latency of the last OSP operation is perceived and the Q-value $Q(s, a)$ is calculated and updated to the corresponding state-action (s, a) entry in the Q-table.

Figure 5 describes a walk-through example of the comparison between the current OSP technique and RLOSP. In the current OSP technique, the device controller simply selects three requests in the tail of the device queue (RQ3, RQ4 and RQ5) to perform OSP operation. As the storage controller is oblivious to the busy/idle status of each flash chip and the space utilization of each plane in physical address allocation, the selected requests might been issued to busy chips (CHP3) or planes with high utilization, thus incurring garbage collection. In this case, RQ3, RQ4 and RQ5 could be blocked thus incurring a long worst-case latency. When these requests are from different processes (PID1, PID2 and PID3), the long worst-case latency could be spread to different processes, thus blocking the latency of all involved processes and impacting the user experience significantly. As shown in the figure, the wait time of PID1, PID2 and PID3 blocks the latency of all processes, the latency of PID1 is prolonged to T5, and the latency of PID2 and PID3 are prolonged to T4.

4.3 Walk-Through Example

For RLOSP, the agent considers the process ID and host I/O information of requests in queue, as well as the busy/idle status of each flash chip and space utilization of each plane in physical address allocation. The device controller firstly selects RQ1, RQ2 and RQ3 to serve, which are all from the same process (PID1). Then, an idle chip (CHP1) and plane with low space utilization will be assigned for these requests. Finally, other requests in queue (RQ4, RQ5) will be served simultaneously with OSP operation. In this manner, the wait time of PID1 process is eliminated, which could significantly reduce the worst-case latency of all processes and improve the QoS performance for SSDs. As shown

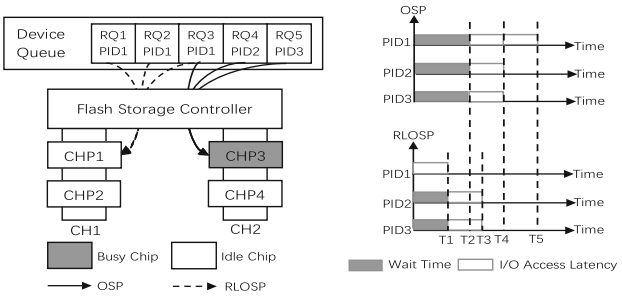


Fig. 5. Walk-through example of the comparison between the current OSP technique and RLOSP.

in the figure, the latency of PID1 for RLOSP is reduced into T1, the latency of PID2 and PID3 for RLOSP is reduced into T2.

5 Results and Analysis

5.1 Experiment Environment

Experimental Setup. In this work, experiments are conducted with a trace-driven simulator SSDsim [20] to verify the proposed approach. The simulator is configured as a new 3D SSD device and a 3D SSD device aged by 70% respectively. OSP operation is implemented in the simulator. The access latency is configured following a SAMSUNG NAND-Flash product [7]. The configuration of the simulator is described in Table 1.

Table 1. Configuration of SSDsim.

Channel number	2	Chip per channel	2
Die per chip	1	Plane per die	1
Block per plane	2048	Page per block	576
Page capacity	4 KB	Over provisioning ratio	10%
Garbage collection scheme	Greedy	Page read latency	90 us
Page write latency	900 us	Block erase	10 ms

Workload Environment. In the experiments, MSRC traces [20] are adopted, which are published by Microsoft Cambridge collected from production servers. The characteristics of the adopted traces are shown in Table 2. The size distribution of the adopted traces shows the diversity among all traces collected from various workload environments. Interval denotes the average arrival time interval between each two consecutive I/O requests. Write ratio means the percentage of write I/O requests in all I/O requests.

Table 2. Characteristics of I/O traces.

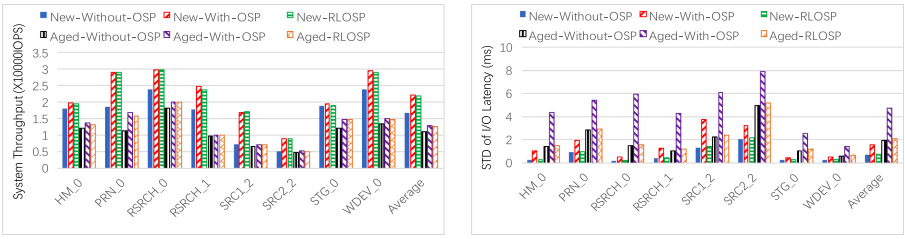
Trace	Size distribution			Write ratio	Interval (ms)
	4 KB	8–16 KB	>16 KB		
HM_0	11.1%	1.1%	87.8%	5.2%	1.45
PRN_0	72.2%	11.1%	16.7%	85.7%	2.14
RSRCH_0	69.6%	23.6%	6.8%	91.8%	1.63
RSRCH_1	5.3%	1.6%	93.1%	2.1%	1.60
SRC1_2	43.8%	3.0%	53.2%	12.7%	1.40
SRC2_2	74.5%	16.2%	9.3%	79.4%	1.56
STG_0	70.0%	20.0%	10.0%	50.0%	0.09
WDEV_0	70.0%	20.0%	10.0%	50.0%	0.09

5.2 Experimental Results

In the experiments, the evaluation results of RLOSP is compared with the performance of the current OSP technique in CT-based 3D SSDs. In addition, the performance of SSDs without OSP is also evaluated to show the advantage and disadvantage of OSP technique, and show the optimizing efficiency of RLOSP further.

System I/O Throughput. Figure 6(a) compares the system I/O throughput of RLOSP with existing approaches. OSP technique could significantly improve the system I/O throughput, while RLOSP could maintain the benefit brought by OSP. Compared with the performance of the current OSP technique, RLOSP reduces the system I/O throughput by only 1.3% on average on a new device. For a device aged by 70%, RLOSP reduces the system I/O throughput by 2.0% on average compared with the current OSP technique. Oriented at OSP optimization, RLOSP chooses proper requests to program with OSP in the device queue instead of canceling the OSP operation. Therefore, RLOSP could improve the QoS performance of the storage device without influencing the benefit gained by OSP technique.

I/O Performance Variation. The current OSP technique could enlarge the I/O performance variation, and part I/O might have to experience long worst-case latency thus degrading the QoS performance of the storage device. This issue is successfully mitigated by RLOSP. Figure 6(b) compares the standard deviation of I/O latency of RLOSP with existing approaches. On average, RLOSP could decrease the I/O performance variation by 51.8% compared with the current OSP technique on a new device. On a device aged by 70%, RLOSP could decrease the standard deviation of 56.2%. For HM_0 trace in the aged device, the I/O performance variation is decreased by RLOSP by 73.4%. RLOSP considers PID and host I/O information of requests in the device queue, thus avoiding the



(a) System Throughput (b) Standard Deviation of I/O latency

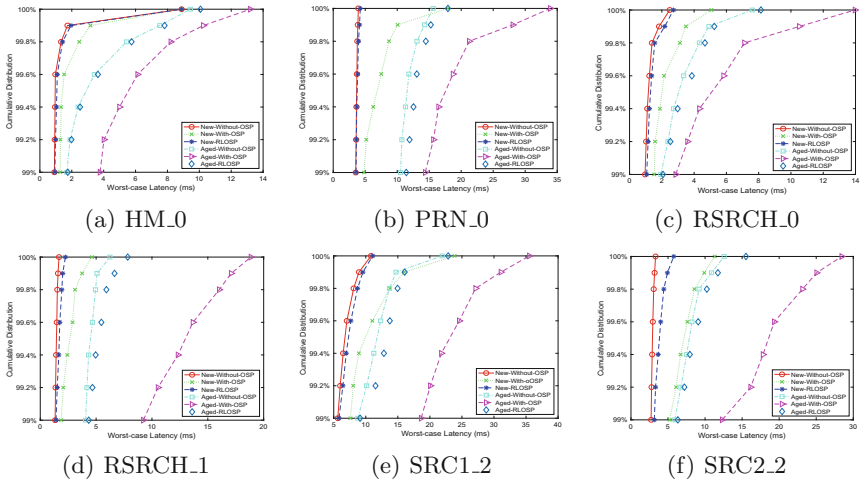
Fig. 6. Evaluation results of various approaches.

worst-case I/O latency from spreading to multiple I/O requests and different processes. In addition, the consideration of busy/idle status of flash chips and space utilization of planes in RLOSP could prevent the I/O requests from being blocked by busy chips or garbage collection. In this manner, RLOSP manages to eliminate the detrimental effects of OSP technique on QoS performance.

Worst-Case I/O Latency. Figure 7 describes the worst-case I/O latency of various approaches at the tail 1% percentile. OSP technique could significantly prolong the worst-case I/O latency, thus degrading the QoS performance of the device. For a new device, the worst-case latency is prolonged by OSP 26.0–54.7%, 35.0% on average at the 99% percentile, prolonged by 44.2–67.8%, 55.3% on average at the 99.9% percentile compared with a device without OSP. For a device aged by 70%, the worst-case latency is prolonged by OSP by 19.5–82.5%, 47.2% on average at the 99% percentile, by 26.6–80.3%, 56.1% on average at the 99.9% percentile. This detrimental effects of OSP technique is successfully eliminated by the proposed RLOSP approach. Among all adopted traces, RLOSP reduces the worst-case latency on a new device by 22.4–48.6%, 30.2% on average at the 99% percentile, by 37.5–59.2%, 46.7% on average at the 99.9% percentile compared with a device with OSP technique. For a device aged by 70%, RLOSP reduces the worst-case latency by 11.7–77.0%, 42.5% on average at the 99% percentile, by 23.5–78.7%, 52.0% on average at the 99.9% percentile compared with a device with OSP technique. By considering the I/O patterns and the storage internal status, RLOSP manages to choose the proper requests in the device queue and allocate right physical addresses for these requests, thus boosting the worst-case I/O latency significantly.

In summary, the proposed RLOSP approach could eliminate the detrimental effects of the OSP technique on the QoS performance, while maintaining the benefits gained by OSP on system I/O throughput. In this manner, the 3D CT-based SSDs could deliver an optimal system throughput with significantly improved QoS performance as well as the user experience.

Overhead Analysis. RLOSP incurs trivial overheads. First, the computation overhead of RLOSP is trivial. The storage system acquires an action at the cost



(a) HM_0 (b) PRN_0 (c) RSRCH_0

(d) RSRCH_1 (e) SRC1_2 (f) SRC2_2

Fig. 7. Evaluation results of the worst-case I/O latency. The horizontal axis is I/O latency in millisecond, the vertical axis is the cumulative distribution of I/O latency.

of a table traversing operation, whose computation overhead is negligible. Moreover, the memory overhead of RLOSP is trivial. The major memory overhead of RLOSP is incurred by the maintenance of the Q-table. According to Eq. 1, there are 1296 states. In RLOSP, the action is allocating a plane for selected requests in the device queue. There are 4 actions in the experiments in this work. Therefore, the Q-table is a 1296×4 vector, which takes around 10 KB memory overhead. This memory overhead is trivial over current SSDs with a DRAM larger than 1 GB.

6 Conclusion

This paper proposes RLOSP, a reinforcement learning assisted approach for the optimization of the OSP technique on 3D CT-based SSDs. First, evaluation results reveal that OSP could maximize the system I/O throughput at the cost of degraded QoS performance. Then, RLOSP considers the I/O patterns and the storage internal status in each OSP operation. In this manner, RLOSP manages to eliminate the detrimental effects of OSP while keeping the benefits of OSP on system I/O performance. We expect that this work could help the designers in 3D SSDs oriented optimizations.

References

1. Chen, J., Wang, Y., Zhou, A.C., Mao, R., Li, T.: PATCH: process-variation-resilient space allocation for open-channel SSD with 3D flash. In: Teich, J., Fummi, F. (eds.) Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, 25–29 March 2019, pp. 216–221. IEEE (2019). https://doi.org/10.23919/DATE.2019.8715197
2. Chen, S., Chang, Y., Liang, Y., Wei, H., Shih, W.: An erase efficiency boosting strategy for 3D charge trap NAND flash. IEEE Trans. Comput. **67**(9), 1246–1258 (2018). https://doi.org/10.1109/TC.2018.2818118
3. Chen, S.H., Chen, Y.T., Wei, H.W., Shih, W.K.: Boosting the performance of 3D charge trap nand flash with asymmetric feature process size characteristic. In: 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6. IEEE (2017)
4. Chen, T., Chang, Y., Ho, C., Chen, S.: Enabling sub-blocks erase management to boost the performance of 3D NAND flash memory. In: Proceedings of the 53rd Annual Design Automation Conference, DAC 2016, Austin, TX, USA, 5–9 June 2016, pp. 92:1–92:6. ACM (2016). https://doi.org/10.1145/2897937.2898018
5. Ji, C., et al.: Inspection and characterization of app file usage in mobile devices. ACM Trans. Storage (TOS) **16**(4), 1–25 (2020)
6. Du, Y., Zhou, Y., Zhang, M., Liu, W., Xiong, S.: Adapting layer RBERs variations of 3D flash memories via multi-granularity progressive LDPC reading. In: Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019, Las Vegas, NV, USA, 02–06 June 2019, p. 37. ACM (2019). https://doi.org/10.1145/3316781.3317759
7. Samsung Electronics: K9F8G08UXM Flash Memory Datasheet, March 2007
8. Gugnani, S., Lu, X., Panda, D.K.: Analyzing, modeling, and provisioning QoS for NVMe ssds. In: Sill, A., Spillner, J. (eds.) 11th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2018, Zurich, Switzerland, 17–20 December 2018, pp. 247–256. IEEE Computer Society (2018). https://doi.org/10.1109/UCC.2018.00033
9. Hu, Y., Jiang, H., Feng, D., Tian, L., Luo, H., Zhang, S.: Performance impact and interplay of SSD parallelism through advanced commands, allocation strategy and data granularity. In: Proceedings of the International Conference on Supercomputing, pp. 96–107 (2011)
10. Jung, M., Choi, W., Srikantaiah, S., Yoo, J., Kandemir, M.T.: HIOS: a host interface I/O scheduler for solid state disks. In: ACM/IEEE 41st International Symposium on Computer Architecture, ISCA 2014, Minneapolis, MN, USA, 14–18 June 2014, pp. 289–300. IEEE Computer Society (2014). https://doi.org/10.1109/ISCA.2014.6853216
11. Jung, S., Song, Y.H.: Garbage collection for low performance variation in NAND flash storage systems. IEEE Trans. CAD Integr. Circ. Syst. **34**(1), 16–28 (2015). https://doi.org/10.1109/TCAD.2014.2369501
12. Kang, W., Shin, D., Yoo, S.: Reinforcement learning-assisted garbage collection to mitigate long-tail latency in SSD. ACM Trans. Embedded Comput. Syst. **16**(5s), 134:1–134:20 (2017). https://doi.org/10.1145/3126537
13. Lee, H., Lee, M., Eom, Y.I.: Mitigating write interference on SSD in home cloud server. In: IEEE International Conference on Consumer Electronics, ICCE 2018, Las Vegas, NV, USA, 12–14 January 2018, pp. 1–3. IEEE (2018). https://doi.org/10.1109/ICCE.2018.8326216

14. Liu, C., Kotra, J., Jung, M., Kandemir, M.T.: PEN: design and evaluation of partial-erase for 3D NAND-based high density SSDs. In: Agrawal, N., Rangaswami, R. (eds.) 16th USENIX Conference on File and Storage Technologies, FAST 2018, Oakland, CA, USA, 12–15 February 2018, pp. 67–82. USENIX Association (2018). https://www.usenix.org/conference/fast18/presentation/liu
15. Liu, C., Kotra, J.B., Jung, M., Kandemir, M.T., Das, C.R.: SOML read: rethinking the read operation granularity of 3D NAND SSDs. In: Bahar, I., Herlihy, M., Witchel, E., Lebeck, A.R. (eds.) Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2019, Providence, RI, USA, 13–17 April 2019, pp. 955–969. ACM (2019). https://doi.org/10.1145/3297858.3304035
16. Lu, Y., Shu, J., Zhang, J.: Mitigating synchronous I/O overhead in file systems on open-channel SSDs. TOS **15**(3), 17:1–17:25 (2019). https://doi.org/10.1145/3319369
17. Luo, Y., Ghose, S., Cai, Y., Haratsch, E.F., Mutlu, O.: HeatWatch: improving 3D NAND flash memory device reliability by exploiting self-recovery and temperature awareness. In: IEEE International Symposium on High Performance Computer Architecture, HPCA 2018, Vienna, Austria, 24–28 February 2018, pp. 504–517. IEEE Computer Society (2018). https://doi.org/10.1109/HPCA.2018.00050
18. Nguyen, D.T., Zhou, G., Xing, G.: Poster: towards reducing smartphone application delay through read/write isolation. In: Campbell, A.T., Kotz, D., Cox, L.P., Mao, Z.M. (eds.) The 12th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2014, Bretton Woods, NH, USA, 16–19 June 2014, p. 378. ACM (2014). https://doi.org/10.1145/2594368.2601458
19. Wang, X., Li, J., Li, J., Yan, H.: Multilevel similarity model for high-resolution remote sensing image registration. Inf. Sci. 505 (2019). https://doi.org/10.1016/j.ins.2019.07.023
20. Wu, C., et al.: Maximizing I/O throughput and minimizing performance variation via reinforcement learning based I/O merging for SSDs. IEEE Trans. Comput. **69**(1), 72–86 (2020). https://doi.org/10.1109/TC.2019.2938956
21. Wu, F., Lu, Z., Zhou, Y., He, X., Tan, Z., Xie, C.: OSPADA: one-shot programming aware data allocation policy to improve 3D NAND flash read performance. In: 36th IEEE International Conference on Computer Design, ICCD 2018, Orlando, FL, USA, 7–10 October 2018, pp. 51–58. IEEE Computer Society (2018). https://doi.org/10.1109/ICCD.2018.00018
22. Xie, W., Chen, Y.: A cache management scheme for hiding garbage collection latency in flash-based solid state drives. In: 2015 IEEE International Conference on Cluster Computing, CLUSTER 2015, Chicago, IL, USA, 8–11 September 2015, pp. 486–487. IEEE Computer Society (2015). https://doi.org/10.1109/CLUSTER.2015.75
23. Yan, S., et al.: Tiny-tail flash: near-perfect elimination of garbage collection tail latencies in NAND SSDs. TOS **13**(3), 22:1–22:26 (2017). https://doi.org/10.1145/3121133
24. Zhu, Z., Han, G., Jia, G., Shu, L.: Modified DenseNet for automatic fabric defect detection with edge computing for minimizing latency. IEEE Internet Things J. **7**(10), 9623–9636 (2020)
25. Zhu, Z., Tan, L., Li, Y., Ji, C.: PHDFS: optimizing i/o performance of HDFS in deep learning cloud computing platform. J. Syst. Arch. **109**, 101810 (2020)