International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Review and analysis of block allocation strategies in SSDs

Shruthi.B.B, Manas M N

Department of Computer Science and Engineering, R.V College of Engineering, Bangluru, India
Professor, Department of Computer Science and Engineering, R.V College of Engineering, Bangaluru, India

---

Abstract - *Flash memory based SSDs have wide range of applications in the data storage industry, taking over the large part of market by replacing the hard disk drives because of low power consumption, faster access, better performance and higher shock resistance. The performance of the flash drive differs for write and erase operations as the erase operation is slower because of the flashing method of the blocks that allows the erase of the complete block at once rather than the individual pages. SSD controllers use a mapping table with a specific allocation strategy to map logical addresses to physical page addresses and vice-versa within storage space. There are many factors that affect the memory allocation in these SSDs, some of which are the wear-leveling, static vs dynamic Parallelism, abstraction, degree of freedom, power consumption, access time, cost etc. One of the challenges that remains is to increase the capacity/yield of the memory which is used in these devices. This problem has various approaches including SLC, MLC, and TLC storage in the NAND units which allow one-bit, two-bit and three-bit storage in the same memory space by manipulating the voltages used in the flash memory unit respectively.*
This paper examines all the different parameters that effect the allocation of the memory in flash memory allocation. It compares and summarizes various strategies involved in allocation and concludes with the advantages and limitations of each of these approaches.

***Key Words*: logical block allocation, NAND flash memory, SSDs, Garbage collector, parallelism, dynamic/static strategy etc.**

1. INTRODUCTION

NAND flash drives are rapidly growing in the data storage industry because of their performance, faster access time, no physical moving parts and cost. There are many factors that effect a performance in a flash memory drive. These drives are made up of a controller and a memory unit. The memory unit are made up of NAND MOSFETs that are arranged page wise. The horizontal programming is done according to a word-line and the vertical programming is done according to the bit-line. A NAND flash drive can be programmed page wise where are the erase function can only be done block wise, hence the name Flash drive. The erase operation is done in the form of flashing a block of data altogether. There are various storage methods in a NAND flash drive that includes SLC (storing of one bit in one flash memory unit (fmu)), MLC (storing two bits per fmu), TLC (three bits per fmu) and so on. The programming for this is done by using the voltage ranges and splitting them into two, four, and eight and so on ranges that allows the programming to be done at various levels. The NAND flash drive is also classified into 2D and 3D drives where a 2D drive is a plane of NAND units and the 3D drive is these planes stalked one upon each other.

There are many strategies for memory allocation in SSDs. Few of the parameters that affect the allocation strategy are the wear-leveling, the amount of bits stored in each of the cells such as SLC, MLC and TLC. The mapping of the logical and physical mapping will also be affected by the parallelism involved in the system. There are many papers that propose different allocation strategies and this paper will compare the strategies and classify them according to their advantages and disadvantages.

2. OVERVIEW

There are many approaches that can be used to achieve different levels of parallelism in the block allocation process. This paper summarizes the concept of parallelism and how the allocation strategy differs because of that. The parallelism can be achieved at various levels such as plane level, die level, channel level parallelism etc that allows the system to perform at various levels. There are various other parameters such as the garbage collector, the static or dynamic allocation strategy and the type of NAND memory partitioning such as SLC, MLC which also determine the performance. The allocation strategy can vary on physical parameters which can create a hybrid memory base for the memory to be accessed. This system will give a different performance and memory usage patterns. The paper compares all these parameters and reviews all the types of allocation strategy adopted in the current market.

2.1 Different Page Allocation strategies

In the paper, An Evaluation of Different page allocation Strategies [1] there are multiple level parallelisms that can be adopted in a system. 1) System-Level Parallelism. 2) Flash-Level Parallelism. 3) Request-Level parallelism.
At system-Level parallelism the I/O requests can be striped over multiple channels called as channel striping. Unlike channels, way-level activities should be serialized because the multiplexed interfaces of each flash chip are shared within a channel. There are four different parallelism that can be achieved channel level, die level, page level and system level parallelism. These parallelism can be compared to one another in terms IOPS and latency numbers. These values are not normalized.
In order to compare all the four parallelism methods we need to adopt a cycle-accurate NAND flash simulator. The simulator lets the system architecture to be designed differently, the package type of this flash is designed be MLC flash with dual die. The different palloc strategies are measured in terms of the contribution of channel, die and plane levels.

This approach evaluates all page allocation (palloc) strategies using a cycle-accurate SSD simulator. The experimental results reveal that the channel-first palloc strategies are not the best from a performance perspective, when all levels of parallelism are considered. Further, our results show that flash-level parallelism can be interfered by channel-first palloc schemes, and internal resources are significantly underutilized with most data access methods. We believe our results and observations can be used for selecting the ideal palloc schemes, given a target workload.

2.2 Dynamic allocation Strategy in SSDs.

The paper, performance evaluation of Dynamic page allocation strategy in SSDs [2] speaks of past studies was on allocation strategies where the target of a flash operation is determined using simple mathematical equations. For example, the target channel, chip, die, and plane are determined based on the quotient and reminder of the successive divisions of the logical address in a predefined. Some of the main dependencies of the allocation strategy are the pattern of address spaces which also consider about deadlock situations in the shared memory space. The dynamic allocation strategy suggests the use of round-robin algorithm or the circular order of resource allocation. According to this parameter, the paper defines a degree of freedom as the number of parallelism levels where resources are allocated dynamically. The degree of freedom can vary from two to four and so on.
The paper also defines the various level parallelism, garbage collector and other factors. The paper maps all these parameters against each other and in different specification scenarios. The performance is found to be impacted by the effective utilization of resources. The paper suggests a new allocation strategy based on dynamic resource assignment to achieve better striping of flash transactions and lower probability of resource conflicts. The simulator used for this comparison shows the best result for the mid ranged or high-ranged SSDs with three degrees of freedom. The steady-state simulation showed that the dynamism greatly helps in the mitigate performance and endurance side effects of the garbage collector.

2.3 Hystor: A hybrid physical storage system

Hystor: Making the Best Use of Solid State Drives in High Performance Storage Systems [3]. There can be an approach which considers the physical storage architecture. There is a design and implementation of a practical hybrid storage system, called Hystor [3]. This system integrates the lost-cost HDDs and the high-speed SSDs as a single block device and isolates this architecture from all other components, this allows the OS kernel to be built without any complications. Hystor achieves an optimal objective of data management through a few factors. First, By analyzing the I/O traffic, hystor automatically learns the workload patterns and allows the system to identify the performance-critical blocks. These critical blocks are mapped to SSD instead of the HHDs. Second, hystor also exploits the high-level information and identifies semantically-critical blocks and offers them a higher priority to stay in SSDs. Third, incoming writes are buffered into the low-latency SSDs for increasing the performance of the write-intensive workloads. The hystor is found to have a better performance using a larger SSD.
This architecture provides a higher advantage in terms of performance and has a disadvantage in terms of cost as the hybrid SSD-HHD drive needs to have a different physical design and hence the interface between from the hardware to the software must be developed from scratch. The system must also take care of timely retraining performance- and semantically- critical data which can be effective as a write-back buffer for the incoming write requests.

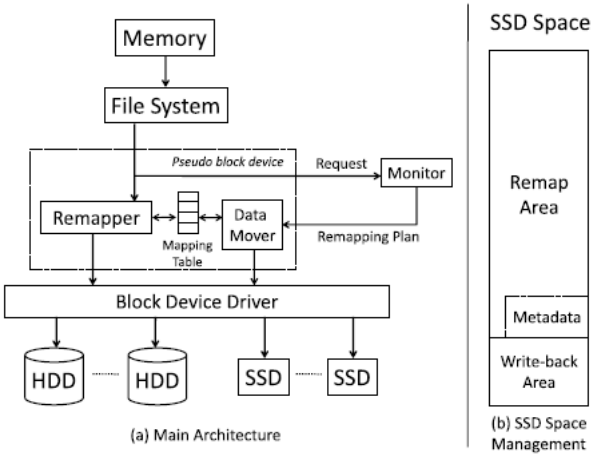

Figure 1: **Architecture of Hystor.**

2.4 ShadowGC: Cooperative Garbage Collection with Multi-level Buffer

In the paper, ShadowGC: Cooperative Garbage Collection with Multi-level Buffer for Performance Improvement in NAND flash-based SSDs [4], the garbage collector is developed with a memory factor called the *"shadows"*. There are advantages of using a ShadowGC along with buffers which are summarized as follows:

When the garbage collection process is on-going, the GC has dirty-copies of the host-side write buffer which are stored in the shadow pages, these pages are relocated to dedicated blocks using fast-write program mode. The fast-write program mode allows the system to resolve the write-amplification and latency issues. As shadow pages have a shorter lifespan which invalidates the dedicated pages when being reclaimed. This automatically increases the write operation speed and reduces the GC latency.

When the garbage collector has the dirty copies in the device-side write buffers, the system chooses to read the contents of the buffer which again allows the system to resolve the write amplification and latency issues. The number of writes are effectively reduced by using a write-back operation of the buffer with the page relocation operation in GC. By choosing to copy the data from the write buffer, the system skips the long latency flash read, ECC check and correction overhead which again leads to the reduction of the average GC latency.

This system has successfully reduced the write amplification by 16.2% and GC latency by 20.5% over the state-of-the-art.

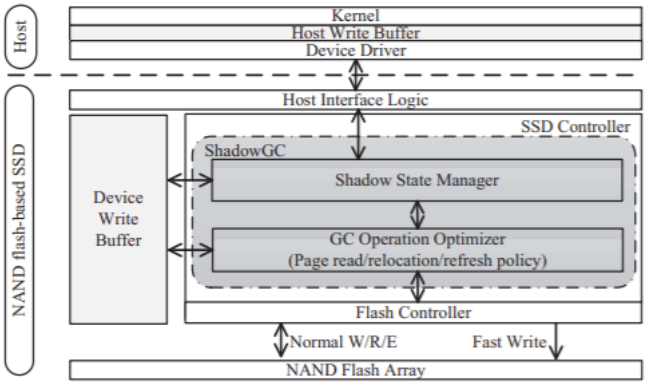

Figure 2: **System Overview with Shadow GC**

2.5 Block Management in Solid-State Devices

This paper summarizes the parameters that usually effect the allocation strategy in a given SSD.[5]

2.5.1 Sequential vs Random

Latency of the sequential access is found to be better as compared to the random access. But as SSDs use a log-structured FTL both sequential and random writes are likely to consume the same amount of time. File systems that run mainly on SSDs must consider the necessity for complex policies to achieve sequential block-level.

Device	Read			Write		
	Seq	Rand	Ratio	Seq	Rand	Ratio
HDD	86.2	0.6	143.7	86.8	1.3	66.8
$S1_{slc}$	205.6	18.7	11.0	169.4	53.8	3.1
$S2_{slc}$	40.3	4.4	9.2	32.8	0.1	328.0
$S3_{slc}$	72.5	29.9	2.4	75.8	0.5	151.6
$S4_{slc\_sim}$	30.5	29.1	1.1	24.4	18.4	1.3
$S5_{mlc}$	68.3	21.3	3.2	22.5	15.3	1.5

Table 1: **Ratio of Sequential to Random Bandwidth**

2.5.2 Logical-to-Physical Mapping

An experiment conducted with a new algorithm for SSDs, called the shortest wait time first (SWTF) which exploits the queue wait times for all the parallel elements in an SSD and schedules I/O that has the shortest wait time. The results of this experiment showed that the scheduling of I/O using SWTF improves by 8% as compared to the FCFS.

2.5.3 Interchangeable Address space

There comes a necessity for the logical address space to be spread uniformly over the device. This is invalidated in disks as the outermost tracks contain more logical addresses as compared to the inner most. Most of the SSDs developed today have a uniformity in their partitions such as SLC, MLC.

2.5.4 Write amplification

The OS of the drive usually assumes that the I/O operation time is proportional to the I/O size. This happens because logical pages is generally larger than the physical page size and the write operation is issued in-place using a read-modify-erase-write cycle. The bandwidth when measured against the write size is poor on small write sizes. The bandwidth is improved as the write size increases up to 1 MB and the bandwidth decreases from this point.

The write amplification can be reduced by merging writes and aligning time to stripe sizes.

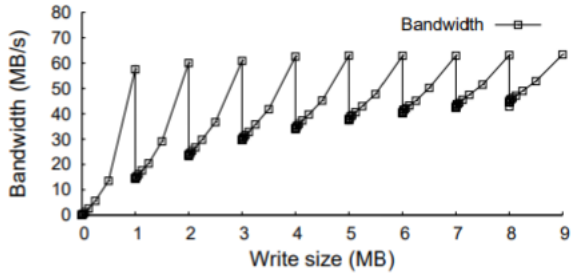

Figure 3: **Write Amplification.**

2.5.5 Other Block allocation strategy parameters

Some of the other parameters use the block-wear or the wear- leveling which allows the system to check the amount of times the block can be reused before it becomes read-only. The background activity also influences the performance of the block allocation in the system. The last parameter would be the Object-based storage which provides a completely different architecture and handling methods as compared to the normal allocation Strategies

3. CONCLUSIONS

This paper summarizes five different approaches that can influence the allocation strategy and performance of a SSD. There are no particular approach defined to give one best result as the system has many parameters and each of these parameters can be manipulated in with different restrictions to achieve a different kind of performance.

The paper started with different strategies of page allocation using parallelism which compared the latency with different level of parallelism. Then moved on to find an optimal solution for dynamic allocation in SSDs. The paper also spoke about a hybrid system which was achieved by manipulating the physical architecture and combining the SSD with HDD. The next approach spoke about how to reduce the average latency from the garbage collector by using shadow pages. Lastly, the paper discussed various parameters that can be manipulated in general to achieve a higher performance in block allocation in SSDs.

REFERENCES

[1] Jung, Myoungsoo & Kandemir, Mahmut. (2012).An evaluation of different page allocation strategies on high-speed SSDs. 9-9.

[2] Tavakkol, Arash & Mehrvarzy, Pooyan & Arjomand, Mohammad & Sarbazi-Azad, Hamid. (2016). Performance Evaluation of Dynamic Page Allocation Strategies in SSDs. ACM Transactions on Modeling and Performance Evaluation of Computing Systems. 1. 1-33. 10.1145/2829974.

[3] Chen, Feng & Koufaty, David & Zhang, Xiaodong (Frank). (2011). Hystor: Making the best use of solid state drives in high performance storage systems. Proceedings of the International Conference on Supercomputing. 22-32. 10.1145/1995896.1995902.

[4] J. Cui, Y. Zhang, J. Huang, W. Wu and J. Yang, "ShadowGC: Cooperative garbage collection with multi-level buffer for performance improvement in NAND flash-based SSDs," 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE).

[5] Rajimwale, Abhishek & Prabhakaran, Vijayan & Davis, John. (2009). Block management in solid-state devices. 21-21