IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Performance Analysis of Solid State Drives

Chhatragun L. Shinde[1], Shefali P. Sonavane[2]

Walchand College of Engineering, Sangli, India[1, 2]

Abstract: Performance as well as reliability are the important factors to consider while designing and deploying solid state drives (SSD) in storage systems. Instead of directly manufacturing the SSD; first needs to get insight into performance analysis using SSD simulation tools and comparing its results. If obtained results are as comparable with real SSD performance results then that is proper way to proceed for manufacturing the SSD. SSDs have pronounced write-history sensitivity, which implies that they have unique requirements for precisely measuring their performances. Primary objectives of performance measurement are to compare and analyze the read/ write ratios, various workloads, interface, stability and security. DiskSim simulator, FlashSim, VSSIM and SSD player tools are used to evaluate the performance of SSD devices employing a variety of Flash Translation Layer (FTL) schemes. This paper compares the performance of SSD with different parameters. The results of comparison show that write operation is more weighted than the read operation.

Keywords: Flash memory; Storage; SSD; Simulation; Performance.

I. INTRODUCTION

Because of NAND flash's faster performance, higher reliability and lower power compared to Hard Disk Drives (HDDs), additionally its ever-decreasing bit cost, the role of SSDs is extending in both portable mobile and enterprise systems. SSDs emerge to be the next-generation storage medium. Today's SSDs mostly build on NAND flash memories and provide few improvements over hard disks including better I/O performance, higher shock resistance and lower energy consumption. As SSD prices continue to drop nowadays, they have been widely deployed in desktops and large-scale data centers [1].

The study on modeling and simulation has dependably been an essential instrument to give the knowledge in the internal behaviours of the storage device like SSDs and HDDs. It is necessary to understand that the design trade-offs of SSDs are to narrow down the design spaces and to decrease the prototyping efforts. It is hard to choose whether particular SSD is having great execution on various platforms with various configurations. Better way is to deal with the execution parameters on the same platforms with the same configurations.

II. SSD BASICS

Firmware, flash memory and controller are the important components in SSDs.

A. Flash memory

Flash memory is the essential part of the SSD. Flash memory has the following characteristics: First one is the unit for read and write operations is a page and the unit for an erase operation is a block. Hence, the speeds in the various operations are incredibly different, as shown in Table I. Second, the same physical page in the memory can be written upon just once after each erase operation. Third, each block has a limited number of erase cycles [2] [3].

When data is written to the flash memory, the amount of free pages turns out to be little; therefore the flash controller must erase a block to recycle free pages. Prior to the controller erases a block, the controller must copy the legitimate data in the block to other empty block; this operation is called as a garbage collection (GC). Table I demonstrates that the block erase time is ten times longer than the page program time. Therefore, the write, erase block operation and GC take extensive number of cycles to complete. As a result, the method for decreasing the number of write operations is very important for the SSDs. If the number of write operations is reduced, the life-time of an SSD is extended and the input output (I/O) processing speed is improved.

TABLE I SPECIFICATIONS OF NAND FLASH MEMORY [9]

Hynix 32GB NAND Flash Chip	
Data Integrity	100,000 erase cycles
Page Read	0.025 ms
Page Program Time	0.2 ms
Block Erase Time	2 ms

B. Read / Write operations on SSD

Solid state drives have exceptionally constrained write endurance in their entire life cycle [4] compared with traditional cache device – RAM. SSD's restricted write endurance which makes the hardware assumption of cache replacement algorithms no longer applicable. Traditional cache replacement (e.g. LRU) schemes prefer to cache hot data in a short run and obviously, they cannot keep write-efficient data long in SSD cache. Different innovative schemes are proposed in recent years to restrict the writes to SSD cache through a data filter. For instance, SieveStore [5] just allows the blocks with more miss counts than a threshold to enter SSD. Flash-based SSD caching is a best solution for boosting today's storage systems. A classical method proposed by Kgil et al. [6] is a two-level cache composed of RAM and a flash memory secondary cache, in this case the flash-based disk cache is divided into isolated read and write regions to improve I/O performance. Read operations on SSDs are generally quicker than write operations. Because of the way that a NAND memory location cannot be overwritten in a single IO operation (as HDDs can overwrite a single logical block address), a NAND flash write operation can take some steps performed by the SSD controller [7].

III.PERFORMANCE PARAMETERS IN SSDS

Following are the some important performance parameters.

A. Device Interface

Agrawalet. al. provides a detailed study on performance of SSD. They provide detailed discussion on design tradeoffs for NAND flash SSDs by performing simulations using Microsoft Research's SSD extension to 50 DiskSim. They performed analysis on different SSD organizations using synthetic workloads and enterprise traces and concluded that serial interface to flash memory is a bottleneck for performance [8]. While SATA Express/AHCI has the advantage of legacy software compatibility, the AHCI interface does not provide ideal performance when communicating to a PCIe SSD. This is on account of AHCI is created during a period when the motivation behind the HBA in a system is to interface the CPU/Memory subsystem with the very slower rotating disk storage subsystem. This kind of interface has some inherent inefficiency when connected to SSD, which acts much more like DRAM than spinning media. NVMe planned starting from the earliest stage to utilize the minimum latency of today's PCIe-based SSD's and the parallelism of today's CPU's, different platforms and applications. At a high stage, the important points of interest of NVMe over AHCI relate to the ability to attempt parallelism in host software and hardware, showed by variations depth command queues, interrupt handling, the quantity of un-cacheable register accesses etc. [9].

B. Stability

SSD performance is more workload sensitive. Performance varies substantially if write requests are sequential or random (their enterprise traces are largely random; their synthetic traces are largely sequential; the performance improvements appeared for the synthetic traces are much more noteworthy than those appeared for the actual traces).

Moreover, the workloads utilized as a part of study are read oriented with roughly a 2:1 read-to-write ratio, which helps to hide the problem of moderate writes in an SSD. In any case, in computer applications (user-driven workloads), there tends to be a much more proportion of writes: in workloads, author seesa 50:50 ratio, which would tend to uncover flash's write issue. User driven workloads are not one-sided towards sequential or random requests but rather give a mix of random and sequential writes at a given time interval [8].

C. Data Integrity

Following four methods can be used to maintain data integrity in enterprise-class SSDs [10] :

Error Correction Code (ECC): Which ensures against read errors as a consequence of hardware errors in the NAND flash memory? The drive controller screens the read process and can correct hardware read errors up to a specific level. If successful then ECC will enable the drive to provide the right data back to the user.

Cyclic Redundancy Check (CRC): Which gives "end-to-end" insurance by guaranteeing that the data written is the same data returned when read. As information comes in over the interface, the drive controller produces a CRC esteem and inserts it with the file's other metadata. At the point when data is recovered, the controller checks to guarantee that the proper CRC value is available. In any case, if data does not match, CRC can distinguish that the error or the mistake has happened. It can't correct it, but it does prevent "silent data corruption".

Correct Address Translation by means of Logical Block Address (LBA): Which guarantees that the data is recovered from the right location; this is the logical mapping of the flash memory blocks. In case of HDD this is the physical sector on the rotating disk.

Correct Version of Data: This is to ensure that the current version of data is returned, rather than the stale version in SSDs. This is not an issue for HDDs because they can directly overwrite the older data. Definition of Data Integrity is, maintaining and assuring the accuracy and consistency of data over its entire life-cycle which SSDs are capable to maintain. Important data integrity algorithms are ECC, CRC and correct address translation via logical block address [10].

Generic methods for the error detection and correction are:

• Knowing there was an error!

Point to point Integrity Checking, avoid silent data corruption, misdirected writes, internal ECC/parity and address corruption checks.

• Preventing/Correcting Errors

Robust Error Correction: low-density parity-check (LDPC) is an iterative coding technique requires more run-time, which gives better error correction but lower throughput [11].

D. Power efficiency

Advanced Host Controller Interface (AHCI) Link Power Management is a method where the SATA AHCI controller puts the SATA link connection to the internal SSD disk into a very low power mode when there's no IO (input/output) activity for some time. The SSD controller naturally returns the link back into active power state when there's real work to be done like I/O operation. This is performed to save power consumption by SSD.

Support for the setting of power models using the Set Features command as mentioned in SATA specification document. Management of various power states using the power models (which power model is set via Set Features command). Less power utilization prompts better thermal dissipation and reliability. To increase performance of SSD controller Adaptive Voltage Scaling (AVS) technology can also be used [12].

E. Security
Self-encrypting SSD drives keep your information safe regardless of the possibility that your drives are lost or stolen. Secure Erase keep all data on the hard drive hidden in less than a second through a cryptographic erase using data encryption key. So you can return, reuse or discard of the drive safely. Auto-Lock automatically locks the SSD drive and secures information the instant a drive is removed from a system, or when the drive or system is turned off [13]. The Opal Storage Specification is a set of specifications for storage devices and these features enhance their security. Also it defines specification for self-encrypting drives (SED). The specification is published by the Trusted Computing Group Storage Workgroup [14].

IV. METHODOLOGY FOR EXTRACTING PERFORMANCE PARAMETERS IN SSDS

A. Experiment Environment
This section elaborates more on simulation tools.
The DiskSim Simulation Environment: DiskSim is an efficient, precise and exceptionally configurable storage disk system simulator which is used to understand various aspects of storage subsystem architecture. It includes modulesand methodsthat simulate storage disks, intermediate controllers, request schedulers, buses, device drivers, disk block caches and disk array data organizations. Some parameters and its description are given in Table II [15]. These parameters are more related to the SSD design specifications for example cache memory, I/Os and synthetic workloads etc.

TABLE II DISKSIM PARAMETER FILE CONFIGURATION

Configuration	
Parameter	**Description**
disksim global	Global Block
disksim stats	Stats Block
disksimiosim	iosim Block
disksimiodriver	I/O Subsystem Component Specifications
disksim bus	Buses
disksimcachemem	Memory Caches
disksimcachedev	Cache Devices
disksimlogorg	Disk Array Data Organizations
disksimpf	Process-Flow Parameters
disksimsynthio	Traces and Synthetic Workloads

EagleTree: EagleTree addresses actual SSD configuration related issues and enables a principled investigation of SSD-Based algorithms. The demonstration scenario explains the design space for algorithms based on an SSD-based I/O stack and shows how researchers and practitioners can utilize EagleTree to perform tractable investigations of this complex design space. EagleTree configuration details are given in Table III [16]. Table III contains the basic SSD configurations which are considered while designing the SSDs. Some of these parameters are different timing delays, memory size like page, block size and controller specifications etc.

TABLE III EAGLETREE CONFIGURATION

Operation Timings (µs)	
BUS_CTRL_DELAY:	1
BUS_DATA_DELAY:	10
PAGE_READ_DELAY:	5
PAGE_WRITE_DELAY:	20
BLOCK_ERASE_DELAY:	60
SSD Architecture	
SSD_SIZE:	4
PACKAGE_SIZE:	2
DIE_SIZE:	1
PLANE_SIZE:	1024
BLOCK_SIZE:	128
PAGE_SIZE:	4096
MAX_SSD_QUEUE_SIZE:	16
OVER_PROVISIONING_FACTOR:	0.7
Controller	
BLOCK_MANAGER_ID:	0
GREED_SCALE:	2
MAX_CONCURRENT_GC_OPS:	8
MAX_REPEATED_COPY_BACKS_ALLOWED:	0
MAX_ITEMS_IN_COPY_BACK_MAP:	1024
WRITE_DEADLINE:	10000000
READ_DEADLINE:	10000000
ENABLE_WEAR_LEVELING:	0
Open Interface	
ENABLE_TAGGING:	0
Operating System	
OS_SCHEDULER:	0

TABLE IV FLASHSIM SIMULATION PARAMETERS

Default simulation parameters	
Flash Type	Large Block
Page (Data)	2KB
Page (OOB)	64KB
Block	(128KB+4KB)
Interface	SATA
GC	Yes
Wear-leveling	Implicit/Explicit
FTL Type	Page/FAST/DFTL
Access Time	
Page Read	130.9 us
Page Write	405.9 us
Block Erase	1.5 ms
Energy Consumption	
Page Read	4.72uJ
Page Write	38.04uJ
Block Erase	527.68uJ

DOI 10.17148/IJARCCE.2016.55173

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

FlashSim: This simulation tool helps to get deep into specification parameters of SSD. Details of Default simulation parameters are given in Table IV [17]. These parameters are considered while designing the prototype of SSD. Some of the important parameters are memory size for example page and block size, access delays and energy consumption etc.

B. Assumptions on SSD performance

Following are the best practices for SSD performance measurement according to Micron Technology, Inc. document [17].

1) The enterprise SSD performance measurement assumptions:

- Drive fill state: The drive should be always 100% full.
- Accesses: It is being accessed 100% of the time (means the drive gets no interface idle time).
- Decisions: The enterprise market picks undertaking SSDs in view of their performance in steady state, full and most pessimistic scenario are not the same thing.
- Consequences of failure: Failure is catastrophic for different users.

2) Client SSD performance measurement assumptions:

- Drive fill state: The drive has less than half of its user space occupied.
- Accesses: This can be accessed a maximum of 8 hours a day, 5 days a week.
- Decisions: The client market picks enterprise SSDs in view of their performance in the fresh- out-of-box (FOB) state. In case of SSDs FOB means there were very less or no program/erase (P/E) cycles since the device was manufactured.
- Consequences of failure: Failure is catastrophic for a single user.

V. PERFORMANCE EVALUATION

A. Experimental Setup

Experimental setup needs some knowledge about programming languages like C and C++. Modify the configuration file according to requirement and execute the make command.

The details available for commercial SSDs are insufficient for modeling them precisely. The study on modeling and simulation has always been an important tool to give the knowledge in the internal behaviors of the storage device to comprehend the design trade-offs, to narrow down the design spaces and to reduce the prototyping endeavors. Instead of checking performance on various platforms with different configuration, look at the performance parameters on the same platforms with the same configurations. Two simulation tools EagleTree and FlashSim are used for read/write performance comparison analysis.

Details of simulation parameters are given in Table V, VI and corresponding graphs are shown in figure 1 and figure 2. In both graphs X axis represents the performance parameters and Y axis represents the values (in unit time) for those parameters.

In Table V, for Case 1 to 5; the values of two parameters PAGE READ DELAY and PAGE WRITE DELAY are changed and observed the write time. In case 5 the block size is changed from 64 to 4 which decreases the average read/write time. Corresponding graph is shown in figure 1. In Table VI, again changed the values of two parameters PAGE READ DELAY and PAGE WRITE DELAY then observed the write time and read time also. In case 5 the RAM WRITE DELAY is changed from 1 to 2, which increases the average write time. Corresponding graph is as shown in figure 2.

TABLE V EAGLETREESIMULATION RESULTS

Parameter	Case 1	Case 2	Case 3	Case 4	Case 5
PAGE_READ_DELAY	5	65	5	5	65
PAGE_WRITE_DELAY	20	35	20	20	35
Write time	33.767	102.191	33.767	33.767	107.992
Read Time	43.480	318.986	43.480	43.480	304.889

TABLE VI FLASHSIM SIMULATION RESULTS

Parameter	Case 1	Case 2	Case 3	Case 4	Case 5
PAGE_READ_DELAY	25	25	1	65	65
PAGE_WRITE_DELAY	300	30	1	35	35
Write time	325.04	55.04	2.04	142.04	100.04

B. Experimental Results and analysis

Experimental setup is run on CentOS 7 following graphs as shown in figure 1 and 2 are used for analysis of read and write time measurement.

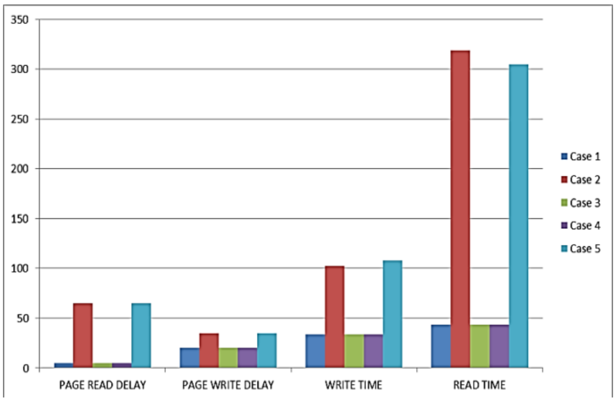

Fig. 1. Graph for EagleTree

The target parameters for SSDs performance are PAGE_READ_DELAY, PAGE_WRITE_DELAY, RAM_WRITE_DELAY and BLOCK_SIZE. The experimental results show that 1) sequential write is less costly than sequential read as both FlashSim and EagleTree use sequential read/write operations. 2) Write time is always more for both the values of PAGE READ DELAY and PAGE WRITE DELAY. 3) If there is an increase the value of any parameter PAGE READ DELAY or PAGE WRITE DELAY then it will increase the average page write time linearly.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

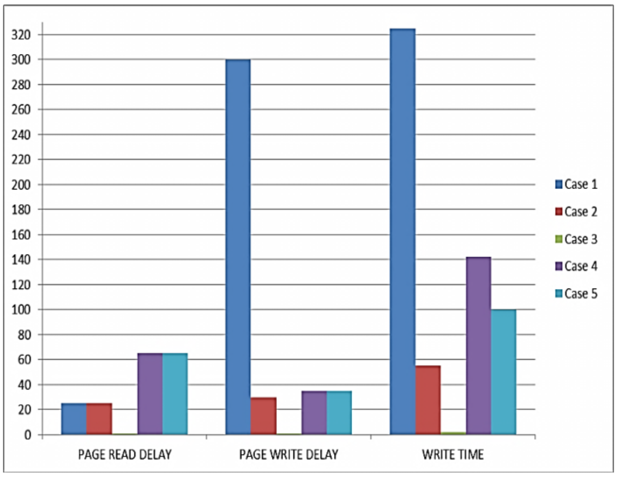

Fig. 2. Graph for FlashSim

VI. CONCLUSION

Solid state disks (SSDs) have numerous points of interest over hard disk drives, including better reliability, performance, durability, security and power efficiency. The characteristics of SSDs are more related to NAND flash memory. To accomplish the maximum performance improvement with SSDs, operating systems or applications; it is necessary to understand the critical performance parameters of SSDs to fine-tune their accesses. NVMe interface provides high performance than SATA because of parallelism in host hardware and software. Major performance parameters are read and write operations. Read operation is performed page by page and having high speed as compared to HDDs. Read performance is not affected due to location of data storage on SSD. Read operation is less costly (in terms of time and number of operations) than write because write operation first erases the data and then writes the user data. Erase operation is performed block by block. Sequential read/ write operation is faster than random read/ write because it uses the sequential physical addresses to perform these operations. How quicker a SSDs can perform I/O tasks is calculated in Input/Output Operations per Second (IOPS) and varies depending on the kind (sequential/ random) of I/O being performed. The greater the number of IOPS, the better the performance. ECC, CRC and LDPC are the techniques which are used to maintain the data integrity in SSDs. Different power management techniques like SATA link power management, device initiated power management, host initiated power management and advanced power management reduce the power consumption in SSD and hence increase performance of SSDs.

In any case, the internal hardware and software organizations change significantly among SSDs and thus, each SSD shows different parameters which impact the overall performance. Parameter-aware management leads to huge performance improvements for expansive file accesses by performing SSD-specific optimizations [2] [18].

REFERENCES

[1] K. Hess. 2011: Year of the SSD? [Online]. Available: http://www.datacenterknowledge.com/archives/2011/02/17/ 2011-year-of-the-ssd/, Feb 2011.
[2] H. Kim and U. Ramachandran, "FlashLite: A user-level library to enhance durability of SSD for P2P file sharing," in Proc. IEEE ICDCS, Jun. 2009, pp. 534–541.
[3] Best Practices for SSD Performance Measurement. [Online]. Available [May 2016]: https://www.micron.com/~/media/documents/products/technical-marketing -brief/brief_ssd_performance_measure.pdf
[4] S. Boboila and P. Desnoyers, "Write Endurance in Flash Drives: Measurements and Analysis", Proceedings of the 8th USENIX conference on File and Storage Technologies (FAST'10), Feb. 2010
[5] T. Pritchett, M. Thottethodi, "SieveStore: a highly-selective, ensemble-level disk cache for cost-performance", Proceedings of the 37th annual international symposium on Computer Architecture (ISCA'10), pp. 163-174, 2010
[6] T. Kgil, D. Roberts, and T. Mudge, "Improving NAND Flash Based Disk Caches", Proceedings of the 35th International Symposium on Computer Architecture (ISCA'08), pp. 327-338, Jun. 2008.
[7] SSD Performance – A Primer, An Introduction to Solid State Drive Performance, Evaluation and Test. [Online]. Available [May 2016]: http://www.snia.org/sites/default/files/SNIASSSI.SSDPerformance-APrimer2013.pdf
[8] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigraphy. "Design Tradeoffs for SSD Performance." In Proc. USENIX Annual Technical Conference (USENIX 2008), Boston, MA, June 2008.
[9] AHCI and NVMe as Interfaces for SATA Express™ Devices - Overview. [Online]. Available [May 2016]: https://www.sata-io.org/sites/default/files/documents/NVMe%20and%20AHCI%20as%20SATA%20Express%20Interface%20Options%20-%20Whitepaper_.pdf
[10] Data Integrity Challenges in flash SSD Design.[Online]. Available [May 2016]: http://www.storagesearch.com/sandforce-art1.html
[11] The Nibbles and Bits of SSD Data Integrity. [Online]. Available [May 2016]: http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2013/20130813_B12_Cohen.pdf
[12] Advanced Power Technologies for SSD Applications. [Online]. Available [May 2016]: http://www.flashmemorysummit.com/ English/Collaterals/Proceedings/2013/20130815_301B_Vainberg.pdf
[13] Drive Security & Disk Encryption. [Online]. Available [May 2016] : http://www.seagate.com/in/en/solutions/security/
[14] Developers Storage Specifications. [Online]. Available [May 2016]: http://www.trustedcomputinggroup.org/developers/storage/specifications
[15] The DiskSim Simulation Environment. [Online]. Available: http://www.pdl.cmu.edu/DiskSim/
[16] Martin KjærSvendsen, MatiasBjørling, Philippe Bonnet, Luc Bouganim, "EagleTree: exploring the design space of SSD-based algorithms", Proceedings of the VLDB Endowment Volume 6 Issue 12, Pp 1290-1293, August 2013
[17] Youngjae Kim; Tauras, B.; Gupta, A.; Urgaonkar, B., "FlashSim: A Simulator for NAND Flash-Based Solid-State Drives," in Advances in System Simulation, 2009. SIMUL '09. First International Conference on , vol., no., pp.125-131, 20-25 Sept. 2009
[18] Jaehong Kim; SangwonSeo; Dawoon Jung; Jin-Soo Kim; Jaehyuk Huh, "Parameter-Aware I/O Management for Solid State Disks (SSDs)," in Computers, IEEE Transactions on , vol.61, no.5, pp.636-649, May 2012