Special Issue of Engineering and Scientific International Journal (ESIJ)
Technical Seminar & Report Writing - Master of Computer Applications - S. A. Engineering College (TSRW-MCA-SAEC) - May 2016
ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

Improving the Write Performance of Solid State Drives

N. Angeline Monica[#1], R. Sridhar[*2]

[1]*Master of Computer Applications, S.A. Engineering college, Chennai-77.*
Angelinemonica.n@gmail.com
[2]*Head, Department of Computer Applications, S.A. Engineering College, Chennai-77.*
sridhar@saec.ac.in

Abstract— Solid State drives are currently used in market for data storage. These SSD drives are performing higher than conventional hard disks which are electro mechanical in nature. In order to know performance of a SSD, we must understand the working of a NAND Flash which forms the base of the pen drives and SSD. Once we know about all the above, we will have clear idea about the limitations of SSD drive and the increment of its write performance.

Keywords— FLASH Memory; NAND Gate; NOR Gate; Floating Gate MOSFET; P/E Cycle; Solid State Drives; Tunneling Effect; Over-provisioning.

1. Introduction

The SSD drives which are made up of mostly flash memory is most faster than all other Electro mechanical devices. Flash memory may be made up of NAND Gate and NOR Gate where NOR based Flash devices are not widely as they are unstable. These drives have limitations which do not allow us to use the full performance of the SSD drives. It is already clearly observed that the SSD drives have a limited P/E cycle which when exceeds damages the FLASH cell. A FLASH cell can be only programmed only after an erase operation. Due to this there is a Lag in write operation than the read. Hence read is faster than write in SSD. There are numerous ways to match the write performance with that of read.

2. Solid State Drives

A SSD drives has no macroscopic moving parts that wear out and limit performance. D-RAM type of memory with power backup as Solid state storage devices connected to Li batteries or external input power and backup drives have been used for decades and has some good performance, though extremely high cost per bits has made these to slow to gain market sales. Without backup power or batteries, the data is lost if power is lost.

2.1 Flash Memory

A single flash memory storage cell is based on the same kind of transistor used in modern digital integrated circuits, but with the added twist of a floating gate that is not electrically connected to anything. Because of the extreme thinness of the dielectric layer beneath the floating gate, appropriate voltage levels across the floating gate can energize electrons enough that they pass through the floating gate and so gets trapped while writing or programming, or move out of the floating gate (erasing).

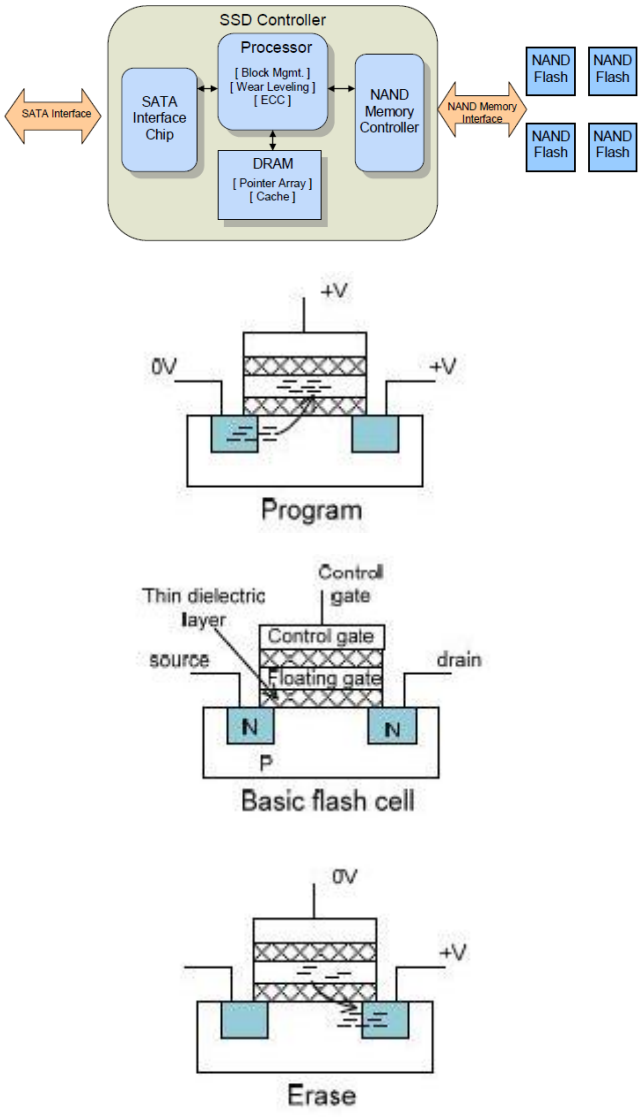

Fig.1: Flash memory and its operations

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

2.2 Flash memory –Wear out

The movement of the energetic electrons across the dielectric layer slowly damages this layer. The compensation attains a position where the suspended gate can no longer grasp an indict, or the electrons cannot shift crosswise the blockade at the planned rate. This causes the phenomenon called carry. Flash memories can only support a certain number of write and erase cycles before the cells 'wear out.' When a cell wears out, it can no longer be used to reliably store data.

2.3 Flash memory –SLC, MLC

SLC- A single cell represents one bit of data. An erased cell represents a '1' state, and a cell programmed with the appropriate charge trapped in the floating gate represents a '0' state. MLC - In the relentless march to improve densities and reduce cost, engineers found that they could store multiple charge levels in the floating gate. Today, commercially available parts can store four different charge levels in one cell, allowing the cell to store two or more bits of data.

	SLC	MLC	TLC
Bits per cycle	1	2	1
P/E cylces	100000	3000	1000
Read Time	25 µs	50 µs	~75 µs
Program Time	200-300 µs	600-900 µs	~900-1350 µs
Erase Time	1.5-2 ms	3 ms	4.5 ms

Higher density/Lower cost

Higher performance & endurance

Fig.2: Performance of Flash memory

STATES	SLC	MLC	TLC
	Empty	Empty	Empty
			1/7
		1/3	2/7
			3/7
	Full	2/3	4/7
			5/7
		Full	6/7
			Full

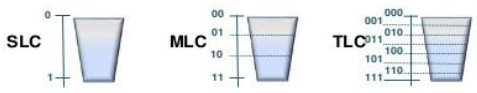

Fig.3: States of SLC,MLC,TLC

2.4 SLC VS MLC VS TLC as Explained with Glass of Water

The glass of water analogy show the advantages of SLC NAND Flash over MLC NAND Flash. When the glass is empty and full it's easier to read the correct fill status in SLC NAND Flash. In MLC NAND Flash it is difficult to read the fill status, when a glass is partially full, taking more time and energy.

3. Increase Performance And Reliability

Proprietary mixes of algorithms for managing wear, write allocation, data mapping, erase-ahead, garbage collection, all of which affect performance and wear life under different types of workloads and write size mixes.

4. Existing Methods: Over Provisioning Metho

All SSD's uses NAND flash as the storage medium which retains memory even without power. NAND flash memory cannot be overwritten, NAND must erase before it can write, and then can be rewritten. It puts the NAND flash cells to the background process to clear them. Essentially, there is never READ goes to those cells then it prepare those NAND cells for WRITE. It refers as the Over Provisioned; SSD has to write so much of data before they go to erase NAND flash cells. By UNMAP these blocks; we add cells to OP area of the drive, it gives the higher WRITE performance until all NAND cells has been written.

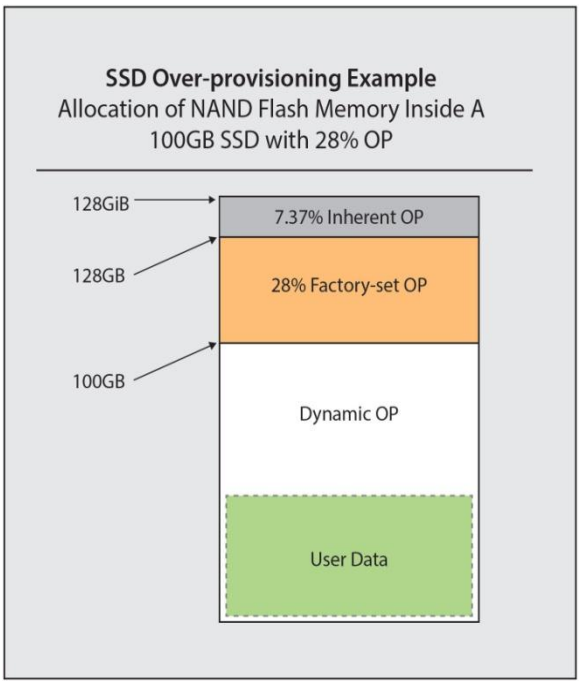

Fig.4: Allocation of NAND flash memory

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

Here, UNMAP/TRIM takes the previously mapped LBAs that map to the NAND flash go to the mapping table and check for the blocks where it is associated and reads those blocks. There is no location awareness (any NAND cells can be associated with any LBA). They keep on doing changes in mapping table and cycle the NAND cells and keep writing on same cells Over-provisioning is a technique to improve the WRITE performance of an SSD by increasing the size of its reserved space. If an SSD is new then the entire flash memory is in free pool. If host WRITE comes then drive allocates the enough memory to write. With higher random write access, NAND cells and blocks are written to more frequently and reclaimed in the background through erase process. Without this activity, the SSD cannot service host WRITES efficiently. SSD drives have over provisioned area but it is small in size. As soon as write request comes from the host, it writes to the entire block contain the memory location but must be erased before that location can be written again. Hence, next time when host write comes then it has to wait unless erase is not done which cause the WRITE performance to be slow. Now, Read/Modify/Write cycle in order to be completed before WRITE operation completes.

Typically, SSD drives are Over Provisioned to some percentage. Host is not able to see this area. The OP operations take the number of NAND cells that are associated with LBA locations and disconnects and when you read those LBAs, it goes to the SSD lookup tables and doesn't find any mapping and return wherever they want to return. It puts the NAND flash cells to the background process to clear them. Essentially, there is never READ goes to those cells then it prepare those NAND cells for and returned to the free pool (where they will be erased until such that they are ready to be used again).

To enhance the WRITE performance the new OPO bit is introduced. OPO will be done for only first time during configuration of LUN. Here, we instruct controller that when SSD is present that supports TRIM/UNMAP with any capacity, the controller will kick off a small RPI while the configuration command is running; it will TRIM/UNMAP the entire drive. TRIM/UNMAP command will wipe out/erase the entire drive. The "set configuration" command won't be completed till the UNMAP command is successful, (the volume will not be OFFLINE as RPI because it is not created). Here, UNMAP takes the previously mapped LBAs that map to the NAND flash go to the mapping table and check for the blocks where it is associated and reads those blocks. There is no location awareness (any NAND cells can be associated with any LBA). They keep on doing changes in mapping table and cycle the NAND cells and keep writing on same cells.

WRITE - It refers as the Over Provisioned; SSD has to write so much of data before they go to erase NAND flash cells. By UNMAP these blocks; we add cells to OP area of the drive, it gives the higher WRITE performance until all NAND cells has been written. An SSD puts newly received data into freshly allocated space in the pool of freeflash. When this happens, old flash blocks that has the stale junk data are considered as invalid

5. Problems of Opo

The write performance is lower than that of read This is due to the extra erase and program process that is carried out by the NAND Flash. Even by using its slower because it works only with small data transfer .With large data movement, when all the reserved area is filled with data then again the write process will be slow

6. Implementation: Erase All Blocks During Portioning

As we have earlier seen we are still using the same principles of Over-provisioning. The area of the erase process is already known from the existing one. The write process of the data is as same as the OPO. In order to write or read data we have to create a partition within the drive, Consider we are going to create a partition with 15 blocks As soon as we get the request to create the partition we will specify the start and end block. Once the start and end blocks are specified, instead of creating the partition the TRIM command must be initiated to these blocks. The TRIM command will erase all the specified blocks. Only after all the block are erased the partition is created and presented to the operating system. The Reserved area will be always erased if data is not there (we have already seen before) and no partition cannot be created on reserved area.

7. Results

As for the first time the data is directly written without any erase process the write performance is very high. This can be compared with the drives that does not have a Over provisioning area and drives which has it. Anyhow, Our implementation is aimed only at first time very large data movement. So from second write, it will use the existing model of Over-provisioning process.

8. Conclusion

By using this process, the data transfer for the first time can be improved upto 50% as we all the erase process is avoided. This method helps during up-gradation of hardware and also during large data movement.

References

[1] M. Abraham, "NAND flash security", In Special Pre-Conference Workshop on Flash Security, August 2010, pp.144-157.

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

[2] U. S. Air Force, "Air Force System Security Instruction 5020", Vol.3,1998, pp.1022-1025.
[3] American National Standard of Accredited Standards Committee X3T13, Information Technology at Attachment-3 Interface, Vol.9, January 1997, pp.48-53.
[4] American National Standard of Accredited Standards INCITS T13. Information Technology - ATA/ATAPI Command Set - 2, June 2010.Apple Inc. Mac OS X 10.6, 2009.pp.1290-2000.
[5] U.S. Army.Army Regulation 1998, pp. 380-319.
[6] A. Birrell, M. Isard, C. Thacker, and T. Wobber, "A design for high-performance flash disks", Technical Report MSR-TR-2005-176, Microsoft Re- search, December 2005, pp.87.
[7] Blancco Oy Ltd, Blan. PC Edition 4.10.1, http://www.blancco.com.
[8] Bundesamts fr Sicherheit in der Informationstech- nik. Richtlinien zum Geheimschutz von Ver- schlusssachen beim Einsatz von Informationstech- nik, December 2004, pp.671.
[9] A. Defence Signals Directorate, Government Infor- mation Security Manual (ISM), 2006, pp.42.
[10] U. S. Defense Security Services, Clearing and San- itization Matrix, June 2007, pp.92.
[11] U. S. Department of Defense. 5220.22-M Na- tional Industrial Security Program Operating Man- ual, January 1995, pp.76-91.
[12] U. S. Dept. of the Navy. NAVSO P-5239-08 Net- work Security Officer Guidebook, March 1996.Eraser. http://eraser.heidi.ie/.
[13] E. Gal and S. Toledo, "Algorithms and data struc- tures for flash memories", ACM Comput. Surv., Vol. 37(2): 2005, pp.138–163.