Introduction to Solid State Drive (NAND flash memory based)

Yaobin Qin
CSCI 8980 Advanced Storage System
Improvement of SSD based on the Issues

- Basic Context of SSD
- Physical Layer of SSD
- IO Operation of SSD
- Issues of SSD
- Improvement of SSD based on the Issues
Basic Context of SSD

Hard Disk Drive

• A HDD is a data storage device used for storing and retrieving digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.

• The platters are paired with magnetic heads arranged on a moving actuator arm, which read and write data to the platter surfaces.

http://www.dijitaler.com/bilgisayar-bilesenleri-4-ssd-vs-harddisk/
How do the HDD to organize and manage the data on the disk?

Logically HDD can be consisted of four main components:

- **Platter**

A hard disk drive platter (or disk) is the circular disk on which magnetic data is stored in a hard disk drive. Hard drives typically have several platters which are mounted on the same spindle. A platter can store information on both sides, requiring two heads per platter.
How do the HDD to organize and manage the data on the disk?

Logically HDD can be consisted of four main components:

- **Track**

A disk drive track is a circular path on the surface of a disk on which information is magnetically recorded and from which recorded information is read.
How do the HDD to organize and manage the data on the disk?

Logically HDD can be consisted of four main components:

- **Cylinder**
  - The cylinder is constructed by a single track location on all the platters.
  - IO operation of Data is based on the cylinder. That means the head reads/writes the data from the top platter to the bottom platter in the same track location.
**Basic Context of SSD**

**Hard Disk Drive**

How do the HDD to organize and manage the data on the disk?

Logically HDD can be consisted of four main components:

- **Sector**
  - The sector is an arc region in the platter. In general, each sector can be stored 512B data. (but it is necessary)
  - Sector is the smallest unit for IO operation.

[Link to faculty.cs.niu.edu/~berezin/463/lec/07io/hd]
Basic Context of SSD

Hard Disk Drive

What information we need to know if the system want to read/write a data on the HDD?

- Seeking the data on which cylinder (need longest time)
  - All the heads are attached to a single head **actuator**, also called an **actuator arm**, that moves the heads around the platters to seek the targeted track.

- Seeking the data on which platter
  - The head is driven by the electronic switch.

- Seeking the data on which sector
  - So far the maximum spinning speed of HDD is around 15000RPM
Solid State Drive

- A solid-state drive (SSD) is a flash memory or DRAM that uses integrated circuit assemblies as memory to store data persistently.
- SSD contains no actual disk, nor a drive motor to spin a disk.
- Unlike disk, SSD can save the cost of track seeking. Because of this factor, SSD has a great improvement in read performance.

[Source: http://www.cnet.com/how-to/digital-storage-basics-part-4-ssd-explained/ by Dong Ngo]
Solid State Drive

- Introduction of Floating Gate Transistor

  - In flash memory, each memory cell resembles a standard MOSFET, except that the transistor has two gates instead of one.

  - On top is the control gate (CG), as in other MOS transistors, but below this there is a floating gate (FG) insulated all around by an oxide layer.

  - The FG is interposed between the CG and the MOSFET channel. Because the FG is electrically isolated by its insulating layer, electrons placed on it are trapped until they are removed by another application of electric field.
Solid State Drive

- SLC MLC TLC Devices
  - Flash memory stores information in an array of memory cells made from floating-gate transistors.
  - In traditional single-level cell (SLC) devices, each cell stores only one bit of information.
  - Some newer flash memory, known as multi-level cell (MLC) devices, including triple-level cell (TLC) devices, can store more than one bit per cell by choosing between multiple levels of electrical charge to apply to the floating gates of its cells.

http://skynetisthefuture.blogspot.com/2015/09/secondary-storage.html
Solid State Drive

- Single Level Cell Device (Programming)

  - A SLC in its default state is logically equivalent to a binary “1” value.
  
  - An elevated on-voltage (typically >5 V) is applied to the CG
  
  - The channel is now turned on, so electrons can flow from the source to the drain (assuming an NMOS transistor).
  
  - the source-drain current is sufficiently high to cause some high energy electrons to jump through the insulating layer onto the FG, via a process called hot-electron injection. (“0” is written)
Solid State Drive

- Single Level Cell Device (Erasing)
  - To erase a flash cell (resetting it to the "1" state), a large voltage of the opposite polarity is applied between the CG and source terminal, pulling the electrons off the FG through quantum tunneling.
  - Modern NOR flash memory chips are divided into erase segments (often called blocks or sectors). The erase operation can be performed only on a block-wise basis; all the cells in an erase segment must be erased together.

https://en.wikipedia.org/wiki/Flash_memory
Electrons can be stored in the floating gate forever after the power off? (in general, a couple of months)

- Charge and Recharge
  - The process of hot electron injection is called the floating gate is charged.
  - However, the voltage of the floating gate will decay because of leakage.
  - So the floating gate need to be recharged before the data loss.
The Structure of NAND Flash

- NAND stores data in a large serial array of transistors. Each transistor can store data. NAND Flash arrays are grouped first and written into pages.

- Cells are grouped with a world line called a page.

- Pages are grouped again into blocks of 64 to 128 pages, which is the smallest erasable unit, thus with a SSD, we only erase in blocks.
The Structure of NAND Flash 2010

- The size of page is 8K together with 448B ECC.
- 128 pages are grouped as 1 block (erasable unit).
- In general, A SSD is divided into two plans based on the odd and even number of blocks. Each block is consisted of 2048 blocks.

Figure 4: 32Gb Array Organization of 25nm Micron SLC NAND Flash (2010)

http://www.qdpma.com/storage/SSD.html
Page Reading

- Hint: Charged Cell has lower leakage current.
  - Apply high voltage (5v) to the unselected page. (not very high)
  - 0v is applied to the selected page.
  - For the targeted page
    - Charged cell has lower leakage current $\rightarrow$ higher $V_s \rightarrow 0$
    - Uncharged cell has higher leakage current $\rightarrow$ lower $V_s \rightarrow 1$
Page Writing (programm)

- Assuming all the cell are in the default stage “1”
- Apply higher voltage to the word line of targeted page (20v).
- Strings being programmed are grounded. Others are at 10v
- “0” is written at 0v.

Can we write “0” and “1” to the page at the same time?

- If the cell in the default state, there is no circuit operation needed to write “1”
- But if the cell is charged (“0”), we can not erase the cell when in the same Page Writing period.
- So a solution to do that is to prepare free space for user through erasing operation. In this case, only “0” need to be truly written.
Page Writing (programm)

- Can we divide a Page Writing Operation into two periods? (to write “1” and “0” in two period)
  - One of issues of SSD -- Erased Before Overwrite
- Need to balance the life of the cells in all chips
  - One of issues of SSD – Wear Off
- Reduce the writing performance due to double the writing period at each writing operation.
  - Garbage Collection
NOR Flash Memory

In the internal circuit configuration of NOR Flash, the individual memory cells:
- Connect in parallel
- Have individual ground line

✓ Enable to independent addressing for each cell
✓ Achieve higher reading performance (particular in random read).

- lower density
- Slower write performance
- Higher price

**Figure 4: NAND and NOR Flash Operating Specifications**
Erase Before Overwrite

“In order to balance the life of all cells in the SSD and improve write performance, the erase unit is a block.”

Does the SSD need to erase a block if the user just wants to modify the page in this block?

1. All the data in the erasing block need to store to the RAM Buffer of SSD before erasing operation.
2. Erasing the block.
3. Updated the new page in the RAM Buffer.
4. Write the block with updated page to the SSD.

- That is why the size of cache in SSD need to be large.
- Such process (Copy On Write) will cause a lot of latency.
- That is a high cost. Because a large data without any update need to write to the buffer and rewrite to the SSD. This cost or penalty is called Write Amplification.
Issues of SSD

Wear Off

• Another limitation is that flash memory has a finite number of program–erase cycles (typically written as P/E cycles).

• Most commercially available flash products are guaranteed to withstand around 100,000 P/E cycles.

Wear leveling

<table>
<thead>
<tr>
<th></th>
<th>SLC</th>
<th>MLC</th>
<th>TLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits per cell</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>P/E Cycles</td>
<td>100,000</td>
<td>3,000</td>
<td>1,000</td>
</tr>
<tr>
<td>Read Time</td>
<td>25 μs</td>
<td>50 μs</td>
<td>~75 μs</td>
</tr>
<tr>
<td>Program Time</td>
<td>200-300 μs</td>
<td>600-900 μs</td>
<td>~900-1350 μs</td>
</tr>
<tr>
<td>Erase Time</td>
<td>1.5-2 ms</td>
<td>3 ms</td>
<td>4.5 ms</td>
</tr>
</tbody>
</table>

Improvement of SSD based on the Issues

Garbage Collection

- Writing the page in the free space avoid Copy On Write Operation.
- Updated page is also written into the free space. And the old page is marked Garbage.
- Process the Copy On Write Operation. (copy the valid page to the RAM Buffer and write to the new free block.
- Erasing the garbage block.

- Balance the life of whole cells
- Reduce Write Amplification

1. Four pages (A-D) are written to a block (X). Individual pages can be written at any time if they are currently free (erased).
2. Four new pages (E-H) and four replacement pages (A’-D’) are written to the block (X). The original A-D pages are now invalid (stale) data, but cannot be overwritten until the whole block is erased.
3. In order to write to the pages with stale data (A-D) all good pages (E-H & A’-D’) are read and written to a new block (Y) then the old block (X) is erased. This last step is garbage collection.

https://en.wikipedia.org/wiki/Write_amplification
Based on the above slides, we realize that the life of SSD and writing performance depend on free space.

But the free space is different between the physical level and file system level.

- When user delete a file or data, the pointer to this file or data will be deleted in general. In this case, the file system know such file or data can be viewed as “Garbage” which can be physically erased. However, the physical level SSD does not know anything about that. So a firmware need to be designed to tell the SSD such information.

- TRIM (which, as a side note, is not an acronym) is a SATA command that enables the operating system to tell an SSD what blocks of previously saved data are no longer needed as a result of file deletions or using the format command.

- Based on the statistics, the TRIM can help SSD to improve the performance more than 95% and maintain the Write Amplification around 1.1 times.
Issues of SSD

Mapping Issue (FTL)

- Unlike magnetic disks, NAND flash memory is characterized by its erase-before-write operation.

- This inherently necessitates NAND flash management software know FTL (flash transition layer), which handles the algorithmic sequence of read, write and erase operations of NAND flash.

- The FTL receives read and write requests and maps a logical address to a physical address in NAND flash.

Abhishek Rajimwale+, Vijayan Prabhakaran*, John D. Davis+* University of Wisconsin, Madison *Microsoft Research, Silicon Valley
Mapping Issue (FTL)

- Typically FTL will logically divide the NAND flash into a meta data area (invisible data) and a user area (visible data).
  - The meta data includes Reserved Blocks for replacing initial or run-time bad blocks,
  - Map blocks for translating logical to physical address,
  - Write buffer blocks for temporarily storing the incoming write data.

- Three mapping schemes:
  - Page mapping
  - Block mapping
  - Hybrid mapping

A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications, Park et al., 2008
Mapping Issue (FTL)

- **Page Mapping**
  - a page-level mapping to map any logical page from the host to a physical page.

  - This mapping policy offers a lot of flexibility, but the major drawback is that the mapping table requires a large Map blocks, which can significantly increase the manufacturing costs.

  - 512MB SSD, a map table size of 1MB is required

![Diagram of page mapping](image-url)

---

Fig. 5. Page-mapping scheme.

A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications, Park et al., 2008
Mapping Issue (FTL)

- Page Mapping
  1. A write request to LPA 5 is inputted to the FTL, the FTL searches for the corresponding PPA 2.
  2. However, the PPN 2 is occupied with PPN 5 old data.
  3. Hence, the requested data should be written to the free page such as PPN 5. Then the data in PPN2 is marked as “Garbage”.
  4. Updated the pointer in Page Map Table from 2 to 5.

Fig. 5. Page-mapping scheme.

A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications, Park et al., 2008
Mapping Issue (FTL)

- **Block Mapping**
  - block-level has a huge improvement for space utilization.
  - 512MB SSD, a map table size of 16KB is required
  - However, the mapping still needs to be persisted on disk in case of power failure, and in case of workloads with a lot of small updates, full blocks of flash memory will be written whereas pages would have been enough. This increases the write amplification and makes block-level mapping widely inefficient.
Mapping Issue (FTL)

- **Block Mapping**
  1. A write request to LPA 5 is inputted to the FTL, the LPA is divided into LBN 1 and page offset 1.
  2. The PBN 0 for the corresponding LBN 1 is determined.
  3. The PPN 1 is determined by the page offset.
  4. However, the LPA 5 is already occupied with LPA 5 old data.
  5. Therefore, the data should be written to a free block such as PBN 2.
  6. Copy the data in PBN 0 with updated page into the PBN 2.
  7. The PBN 0 is marked as “Garbage”.

A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications, Park et al., 2008
Mapping Issue (FTL)

- Hybrid Mapping

The tradeoff between page-level mapping and block-level mapping is the one of performance versus space. Some researchers have tried to get the best of both worlds, giving birth to the so-called “hybrid” approaches.

![Diagram of Hybrid Mapping Scheme](image.png)

Fig. 7. Hybrid mapping scheme.

A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications, Park et al., 2008
Mapping Issue (FTL)

- **Hybrid Mapping**
  1. A write request to LPA 5->7->7->5 is inputted to the FTL, the LPA is divided into LBN 1 and page offset 1.
  2. These four write requires are written into the Log Block Page Map Table. As a result, only the last requires 7,5 are valid for LBN 1 with PBN 100. (marked 7’’ and 5’’ in the Log Block)
    - LPN 7->Logical Page Offset 3
    - LPN 5->Logical Page Offset 1
  3. Updated Log Block Page Map Table
  4. Merge Operation need to be processed when the Log Block is full. After generating a new free block, the Log Block and original Data Block are erased to free space.
Issues of SSD

Fault Tolerant Issue

Many things cause errors on Flash!
• Write Disturb
  - Data Cells NOT being written to are corrupted.
  - Fixed with normal erase

• Read Disturb
  - Repeated reads on same page effects other pages on block
  - Fixed with normal erase

• Charge Loss/Gain
  - Transistors may gain or lose charge over time
  - Flash devices at rest or rarely accessed data
  - Fixed with normal erase

All of these issues are generally dealt with very well using standard ECC techniques.
Fault Tolerant Issue

Many things cause errors on Flash!

• **Write Disturb**
  - Data Cells NOT being written to are corrupted.
  - Fixed with normal erase

• **Read Disturb**
  - Repeated reads on same page effects other pages on block
  - Fixed with normal erase

• **Charge Loss/Gain**
  - Transistors may gain or lose charge over time
  - Flash devices at rest or rarely accessed data
  - Fixed with normal erase

All of these issues are generally dealt with very well using standard ECC techniques.
Issues of SSD

Fault Tolerant Issue

Many things cause errors on Flash!
• Write Disturb
  - Data Cells NOT being written to are corrupted.
  - Fixed with normal erase

• Read Disturb
  - Repeated reads on same page effects other pages on block
  - Fixed with normal erase

• Charge Loss/Gain
  - Transistors may gain or lose charge over time
  - Flash devices at rest or rarely accessed data
  - Fixed with normal erase

All of these issues are generally dealt with very well using standard ECC techniques. (Error Correction Coding)
• Improve the Write Amplification
• Reduce Copy On Write

Improve the wear off
<table>
<thead>
<tr>
<th>Solid NAND flash based</th>
<th>Mechanism type</th>
<th>Magnetic rotating platters</th>
</tr>
</thead>
<tbody>
<tr>
<td>64GB</td>
<td>Density</td>
<td>80GB</td>
</tr>
<tr>
<td>73g</td>
<td>Weight</td>
<td>365g</td>
</tr>
<tr>
<td>Read: 100MB/s, Write : 80MB/s</td>
<td>Performance</td>
<td>Read: 59MB/s, Write: 60MB/s</td>
</tr>
<tr>
<td>1W</td>
<td>Active Power consumption</td>
<td>3.86W</td>
</tr>
<tr>
<td>20G (10~2000Hz)</td>
<td>Operating Vibration</td>
<td>0.5G (22~350Hz)</td>
</tr>
<tr>
<td>1,500G for 0.5ms</td>
<td>Shock resistance</td>
<td>170G for 0.5ms</td>
</tr>
<tr>
<td>0°C to 70°C</td>
<td>Operating temperature</td>
<td>5°C to 55°C</td>
</tr>
<tr>
<td>None</td>
<td>Acoustic Noise</td>
<td>0.3 dB</td>
</tr>
<tr>
<td>MTBF &gt;2M hours</td>
<td>Endurance</td>
<td>MTBF &lt; 0.7M hours</td>
</tr>
</tbody>
</table>

[8] A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications, Park et al., 2008
Thank You

Questions?