Chapter 6 - External Memory

Luis Tarrataca
luis.tarrataca@gmail.com

CEFET-RJ

Table of Contents I

Table of Contents I

Motivation

Computers have a set of internal memory mechanisms

- registers, RAM, etc

Computers also need to interact with peripherals

- Some of these are storage devices;
- *A.k.a* external memory.

How does such an external memory work?

How is such an external memory organized?

Some examples of external memory include:

- Magnetic disks;
- Redundant Array of Independent Disks (RAID);
- Solid State Drives (SSD);
- Optical Memory;
- Magnetic tapes.

Lets take a closer look at some of these.

Magnetic Disks

Circular platter coated with a magnetizable material.

Figure: Interior of a magnetic hard drive

Data are recorded on and later retrieved from the disk via a head:

- Most common design: a read head and a write head;
- During a read or write operation:
  - Head is stationary while the platter rotates beneath it.

Write Mechanism

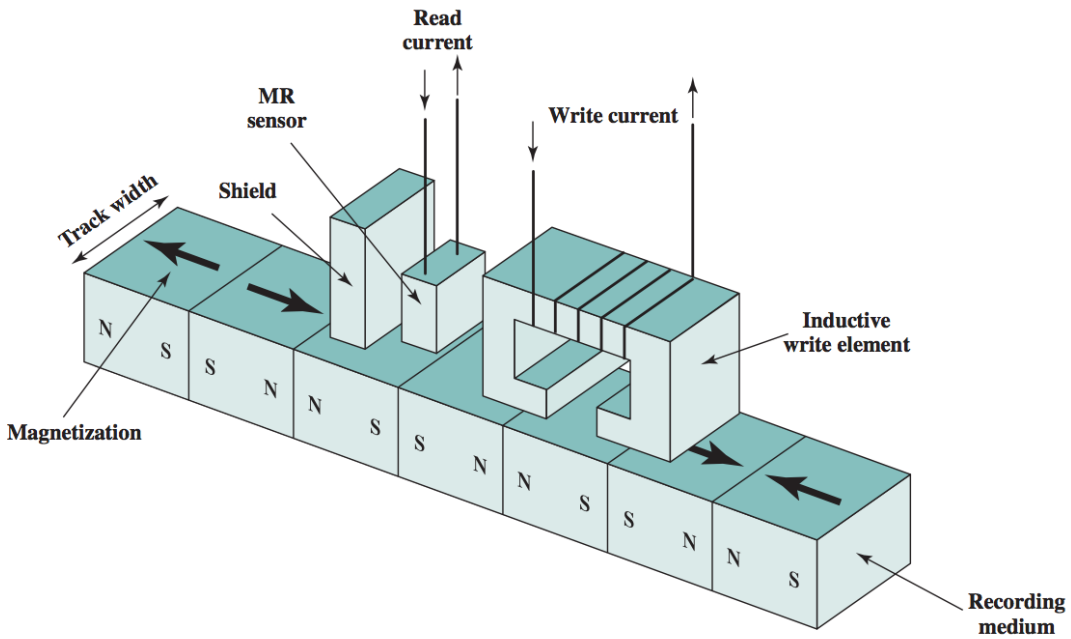

Figure: Inductive Write / Magnetoresistive Read Head. (Source: (Stallings, 2015))

Write Mechanism:

- Electric pulses are sent to the write head;
- Resulting magnetic patterns are recorded on the surface below:

Read Mechanism

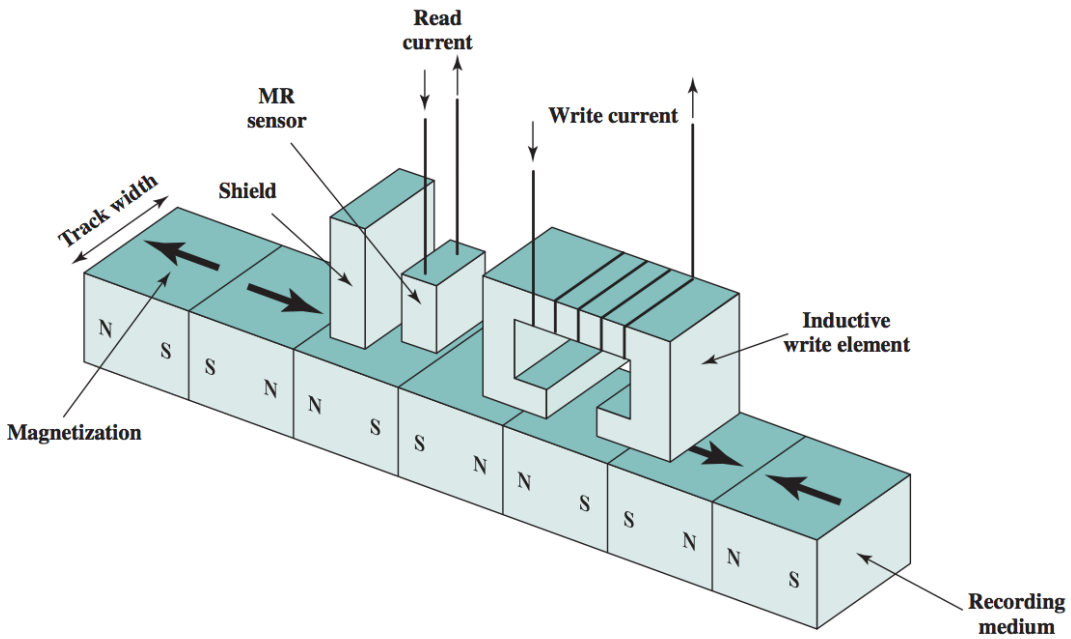

Figure: Inductive Write / Magnetoresistive Read Head. (Source: (Stallings, 2015))

Read Mechanism:

- Disk surface passes under the read head;
- Generating a current of the same polarity as the one recorded.

Contemporary Read Mechanism

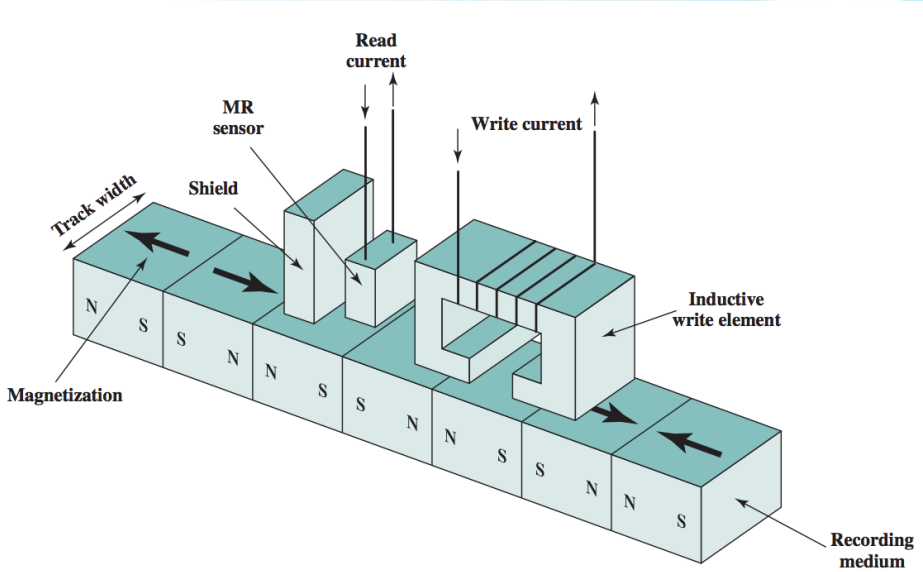

Figure: Inductive Write / Magnetoresistive Read Head. (Source: (Stallings, 2015))

Contemporary disk systems use a separate read head:

- Head consists of a magnetoresistive (MR) sensor;
- Resistance of MR material depends on the direction of the magnetization;
- Sensor detects resistance changes as voltage signals.

Data Organization and Formatting

Based on the read / write head mechanism how are disks organized?
Any ideas?

Data Organization and Formatting

Based on the read / write head mechanism how are disks organized? Any ideas?

Impractical to have a large tape moving under the head:

- Old magnetic tapes;
- Substantial seek time times;

Circular Information Storing:

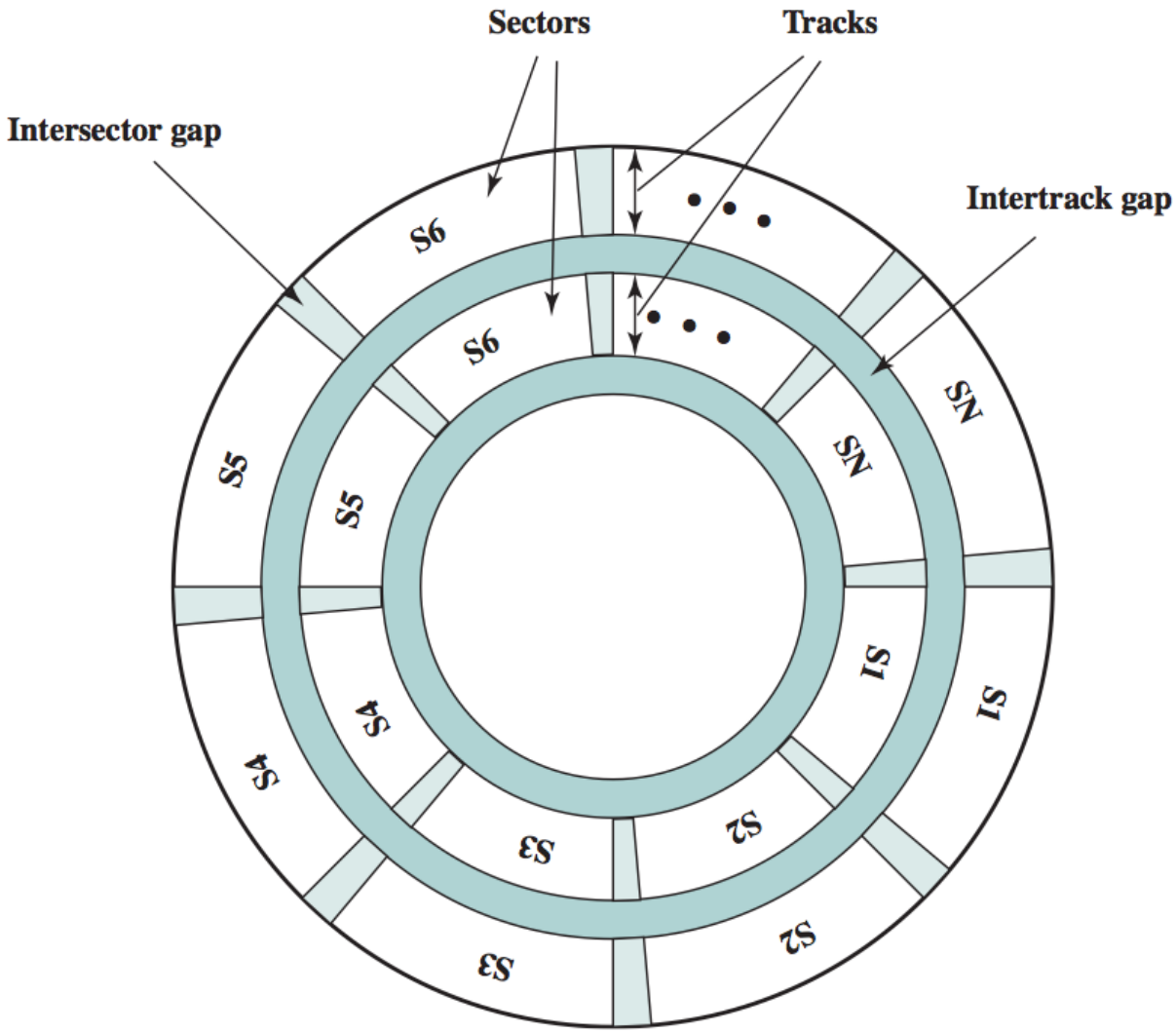

Figure: Disk Data Layout. (Source: (Stallings, 2015))

Each platter has a concentric set of rings, called **tracks**:

- Each track is the same width as the head.
- There are thousands of tracks per surface.
- Adjacent tracks are separated by gaps in order to prevent;
  - Misalignment of the head;
  - Magnetic field interference.

Data are transferred to and from the disk in **sectors**:

- There are typically hundreds of sectors per track;
- These may be of either fixed or variable length
  - Nowadays 512 bytes is the universal sector size.
- Adjacent sectors are separated by intersector gaps.

Now that we know about the existence of tracks and sectors:

How can the head find these elements within the disk?

How can the head find these elements within the disk?

Head needs to locate sector positions within a track, requiring knowing:

- Starting point on the track;
- Start and end of each sector;

How can the head find these elements within the disk?

Head needs to locate sector positions within a track, requiring knowing:

- Starting point on the track;
- Start and end of each sector;

How can this be performed? Any ideas?

How can this be performed? Any ideas?

- Location data needs to be recorded on the disk:
  - Disk is formatted with extra data;
  - This data is used only by the drive
    - Not accessible to the user.
  - Reason why:
    - Space seen as available by OS $\neq$ to physical disk space

Example (1/3)

Lets look at an example:

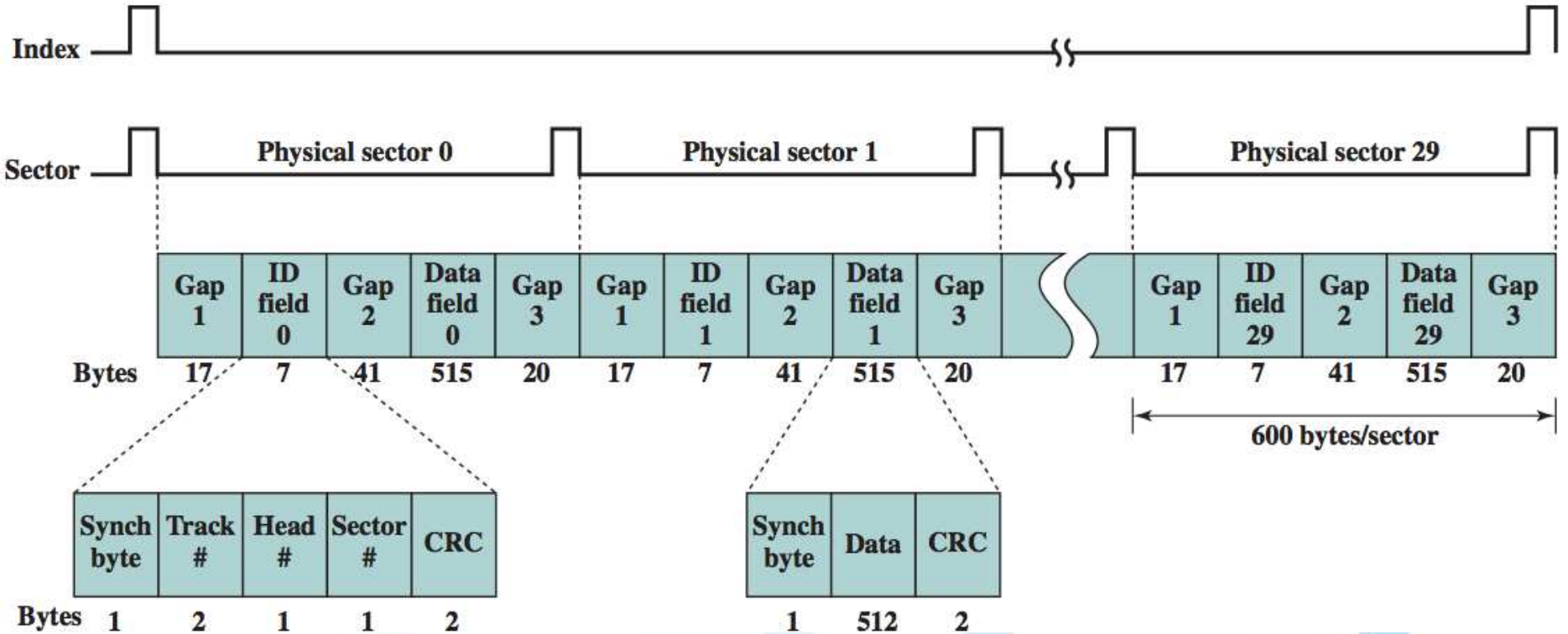

Figure: Winchester Disk Format. (Source: (Stallings, 2015))

Example (2/3)

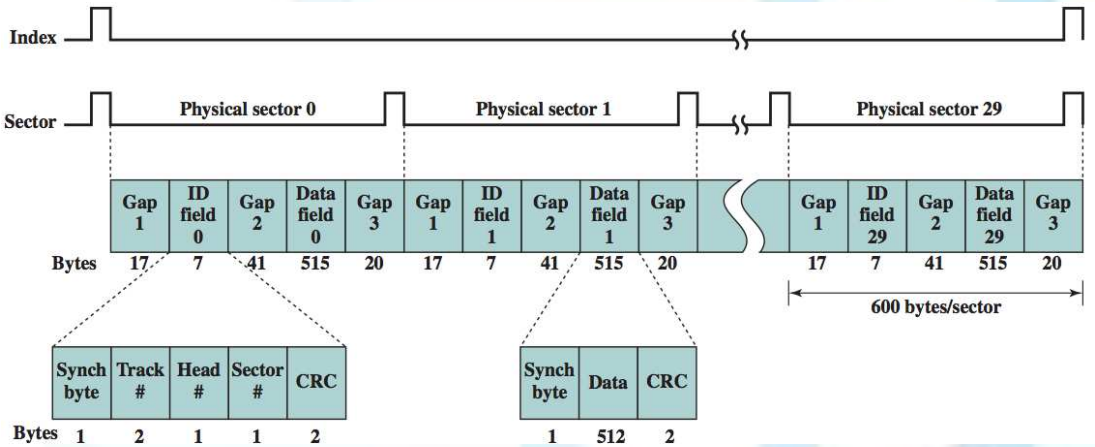

Figure: Winchester Disk Format. (Source: (Stallings, 2015))

Each track contains 30 fixed-length sectors of 600 bytes each.

- Each sector holds 515 bytes of data plus other control information;
- This means that only $515/600 \approx 85\%$ is available for data...

Example (3/3)

Lets look at a specific example:

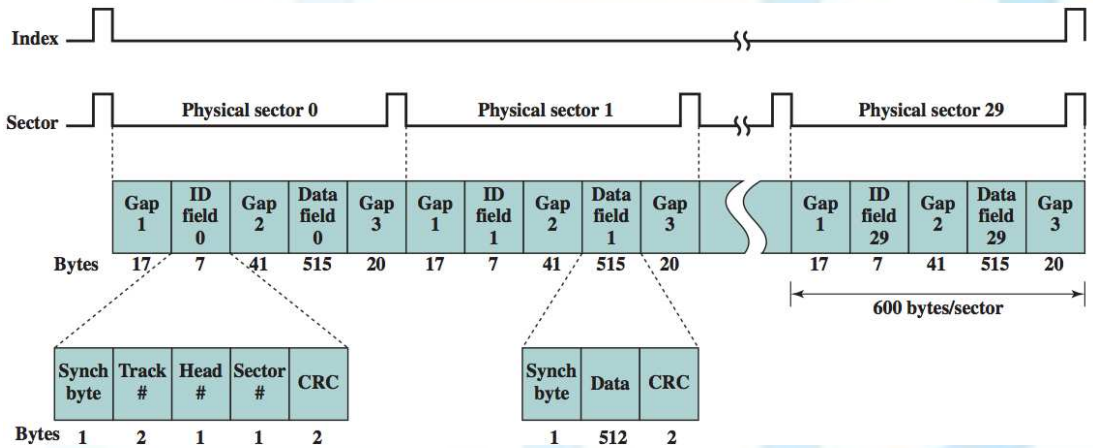

Figure: Winchester Disk Format. (Source: (Stallings, 2015))

The ID field uniquely identifies a sector containing:

- SYNCH byte is a special bit pattern that delimits the beginning of the field;
- Track number
- Head / Surface number for disks with multiple surfaces;

Components of a Disk Drive

One way of organizing the components of a disk drive:

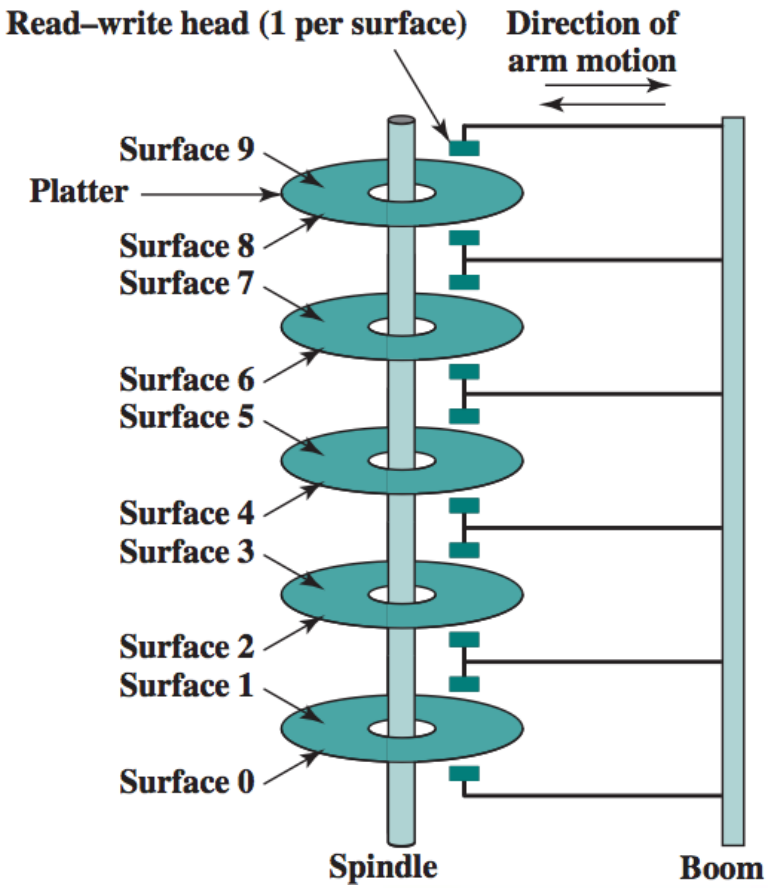

Figure: Components Of A Disk Drive. (Source: (Stallings, 2015))

Figure: Interior of a magnetic hard drive

Disk Performance Parameters

So, what do you think are some variables that influence the performance of a disk?

Disk Performance Parameters (1/4)

- **Seek time** - the time it takes to position the head at the track;
- **Rotational delay** - once the track is selected:
  - The sector still needs to line up with the head;
  - *E.g.:* an hard disk rotating at 20000 rpm:
    - 20000 rpm $\rightarrow$ one revolution per 3 ms;
    - On average we will have to wait for half the plate to spin;
    - rotational delay = 1.5 ms;
- **Access time** = SeekTime + Rotational Delay

Disk Performance Parameters (2/4)

- **Transfer time** - Data transfer portion of the operation;

$$T = \frac{b}{rN}$$

  - **b** - number of bytes to transfer;
  - **r** - rotation speed per second (rps)
  - **N** - number of bytes on a track;

Disk Performance Parameters (3/4)

- **Total average time**:

$$T = T_{\text{seek}} + T_{\text{rotational delay}} + T_{\text{transfer time}}$$

$$T = T_{\text{seek}} + \frac{1}{2r} + \frac{b}{rN}$$

Disk Performance Parameters (4/4)

There are also delays associated with I/O operation:

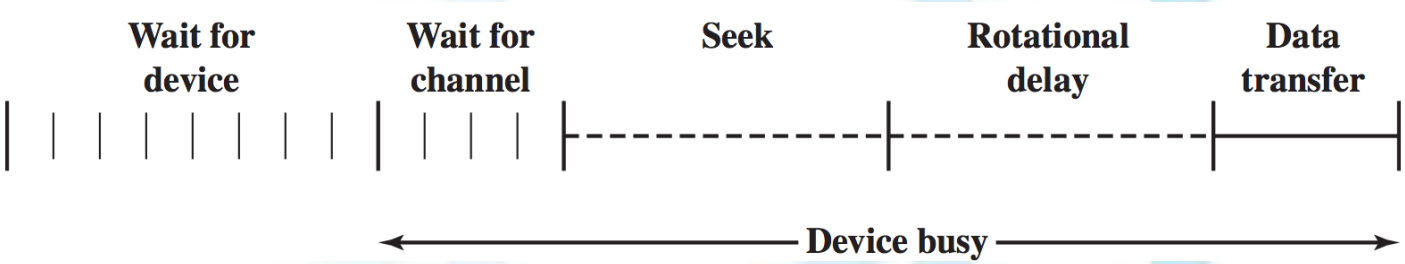

Figure: Timing of a disk I/O transfer (Source: (Stallings, 2015))

1. Wait for device:
   - A process must wait in a queue for a device to be available;
2. Eventually, the device is assigned to the process;
3. Wait for I/O channel:
   - If the device shares the I/O channel it must wait for it to be available;
4. Only after this point do we proceed with the head seek.

Characteristics	Constellation ES.2	Seagate Barracuda XT	Cheetah NS	Momentus
Application	Enterprise	Desktop	Network attached storage, application servers	Laptop
Capacity	3 TB	3 TB	400 GB	640 GB
Average seek time	8.5 ms read 9.5 ms write	N/A	3.9 ms read 4.2 ms write	13 ms
Spindle speed	7200 rpm	7200 rpm	10, 075 rpm	5400 rpm
Average latency	4.16 ms	4.16 ms	2.98	5.6 ms
Maximum sustained transfer rate	155 MB/s	149 MB/s	97 MB/s	300 MB/s
Bytes per sector	512	512	512	4096
Tracks per cylinder (number of platter surfaces)	8	10	8	4
Cache	64 MB	64 MB	16 MB	8 MB

Figure: Typical Hard Disk Driver Parameters (Source: (Stallings, 2015))

Redundant Array of Independent Disks (RAID)

Now that we know about how hards disks works:

How can we deal with hard disk failure? Any ideas?

Redundant Array of Independent Disks (RAID)

Now that we know about how hards disks works:

How can we deal with hard disk failure? Any ideas?

Redundancy is very important in computation:

- If a disk dies we do not want to lose everything;
- RAID is a form of redundancy used to deal with hard disk failure.

RAID has seven design architecture levels sharing these characteristics:

1. RAID is a set of physical disk drives:
   - Viewed by the operating system as a single logical drive;
2. Data are distributed across the physical drives of an array:
   - Scheme known as **striping**;
3. Redundant disk capacity is used to store parity information
   - Guaranteeing data recoverability in case of a disk failure. (RAID $\geq$ 2)

RAID strategy employs multiple disk drives:

- Data are distributed:
  - Enable simultaneous access from multiple drives;
  - Separate I/O requests can be handled in parallel;
  - Thereby improving I/O performance;
- Same concept of parallelism used throughout computation
  - Performance bottleneck? Parallelise the components...

Lets have a look at the different RAID levels....

RAID 0

Good for high I/O request rate:

- Disk array provides high I/O execution rates:
  - This is done by balancing the load across multiple disks.

RAID 0 has no redundancy:

- Therefore, not a true member of the RAID family....

Data are striped across the available disks:

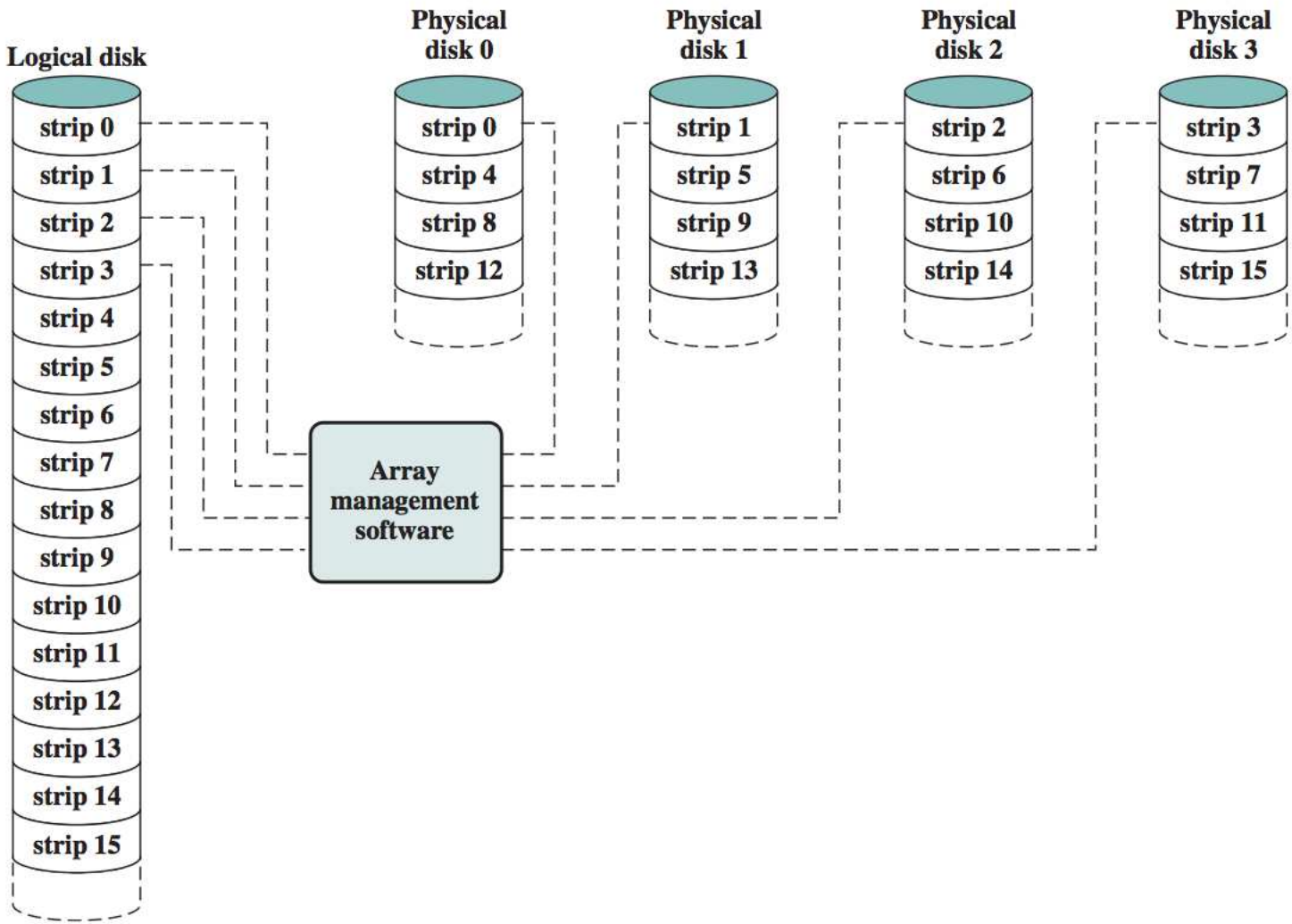

Figure: Data Mapping for a RAID Level 0 Array (Nonredundant). (Source: (Stallings, 2015))

Data are striped across the available disks:

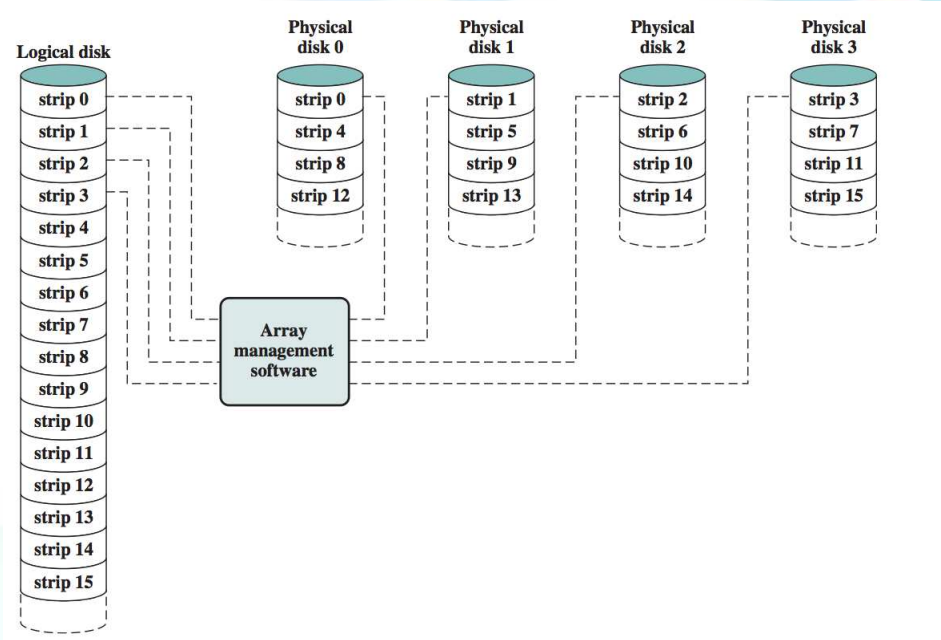

Figure: Data Mapping for a RAID Level 0 Array (Nonredundant). (Source: (Stallings, 2015))

Data are viewed as being stored on a logical disk:

- Logical disk is divided into strips:
  - Strips may be physical blocks, sectors, or some other unit
- Strips are mapped round robin to consecutive disks in the array.

RAID 0 advantage:

- A single I/O request consists of multiple logically contiguous strips:
  - Up to *n* strips for that request can be handled in parallel;
  - Greatly reducing the I/O transfer time.
- Recall that total read / write time is a function of seek and rotational delay:
  - By distributing across multiple disks we are diminishing these times;
  - Instead of having multiple seek and rotational delays for a single hard disk.

RAID 1

Very simple:

- Redundancy is achieved by duplicating all the data.

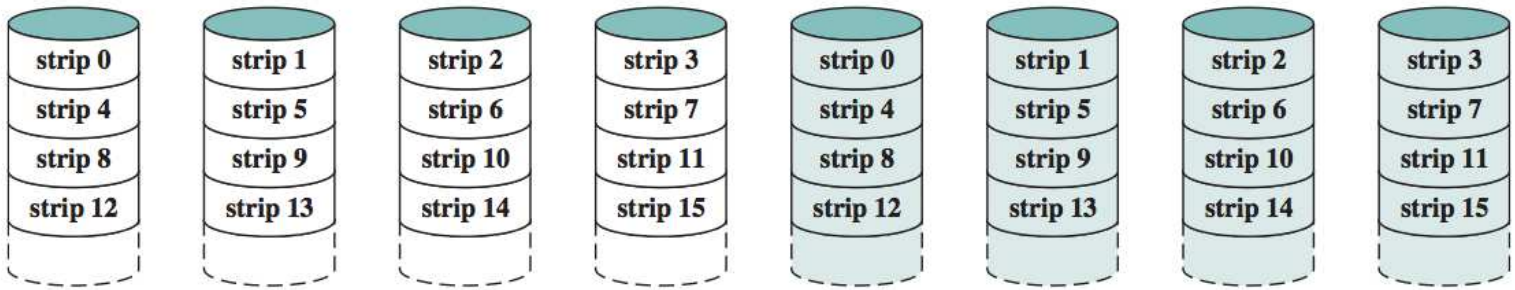

Figure: Data Mapping for a RAID Level 1 Array (Mirrored). (Source: (Stallings, 2015))

- Read requests can be serviced by the lowest access-time disk;
- A write request requires updating in parallel both strips:
  - Write performance is dictated by the slower of the two writes

- Recovery from a failure is simple:
  - When a drive fails, the data may still be accessed from the second drive.
- **Disadvantage**:
  - Principal disadvantage of RAID 1 is the cost:
    - Requires twice the disk space of the logical disk that it supports.
- **Advantage**:
  - Achieves high I/O request rates if the bulk of the requests are reads.
    - Performance of RAID 1 can approach double of that of RAID 0.

RAID 2

- Makes use of a parallel access technique:
  - All member disks participate in the execution of every I/O request;
  - Spindles of the individual drives are **synchronized:**
    - Each disk head is in the same position on each disk at any given time.

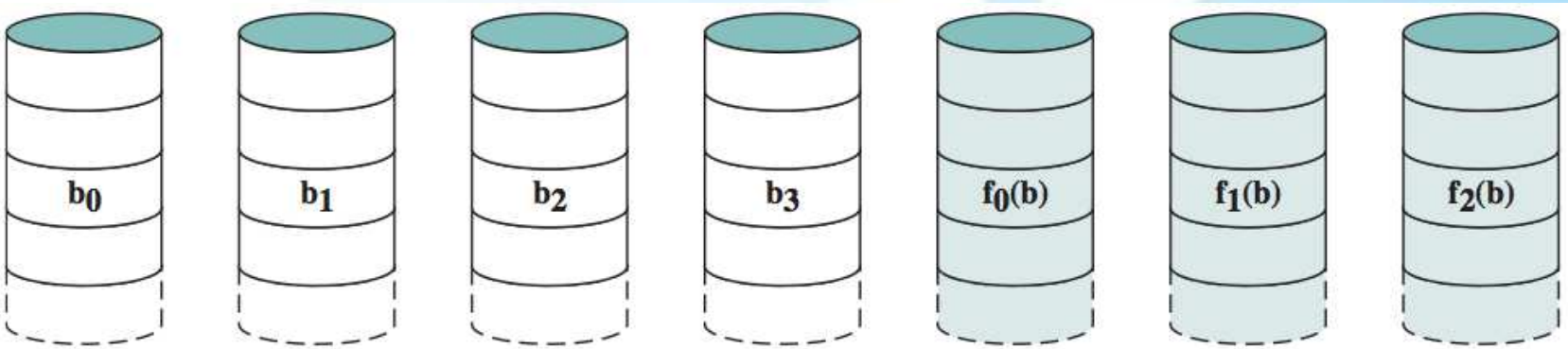

Figure: Data Mapping for a RAID Level 2 Array (Hamming code). (Source: (Stallings, 2015))

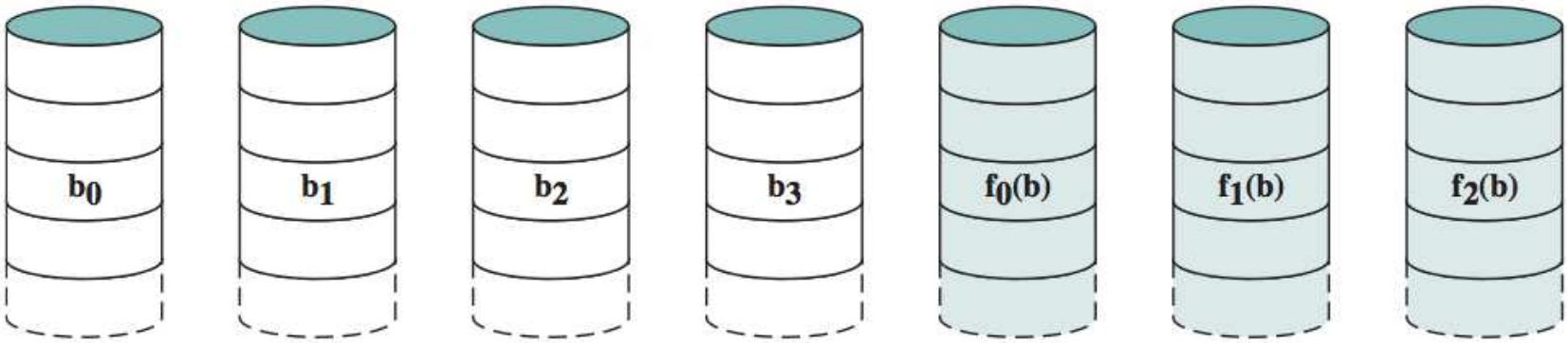

Figure: Data Mapping for a RAID Level 2 Array (Hamming code). (Source: (Stallings, 2015))

- Typically: Hamming code error-correction is used:
  - Stripping occurs at the bit-level;
  - Code bits are stored in the corresponding bit positions on multiple parity disks;
- Rarely used in practice!
  - Currently all hard disk drives implement internal error correction:
  - Complexity of an external Hamming code offers little advantage...

RAID 3

Similar fashion to RAID 2:

- Difference:
  - Requires only a single redundant disk, no matter how large the disk array.

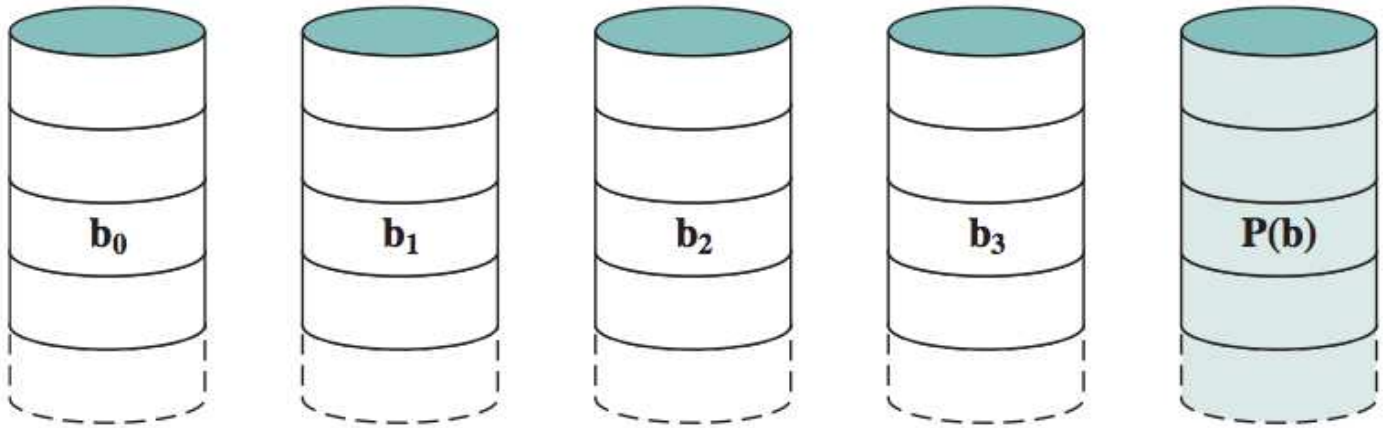

Figure: Data Mapping for a RAID Level 3 Array (Bit-interleaved parity). (Source: (Stallings, 2015))

- Parity bit is computed for the set of individual bits in the same position on all of the data disks.

$$X_4(i) = X_0(i) \oplus X_1(i) \oplus X_2(i) \oplus X_3(i)$$

  - X_0, X_1, X_2, X_3 represent the data hard disks;
  - X_4 is the redundancy hard disk
  - i represents the i-th bit.

- Suppose that drive X_1 has failed. If we add $X_4(i) \oplus X_1(i)$ to both sides of the preceding equation, we get

$$X_1(i) = X_0(i) \oplus X_2(i) \oplus X_3(i) \oplus X_4(i)$$

- Now we just need to process each individual bit i.

RAID 4

- Makes use of an independent access technique:
  - Each member disk operates independently;
  - Separate I/O requests can be satisfied in parallel.
  - More suitable for applications that have high I/O request rates;
  - Less suitable for applications that require high transfer rates.

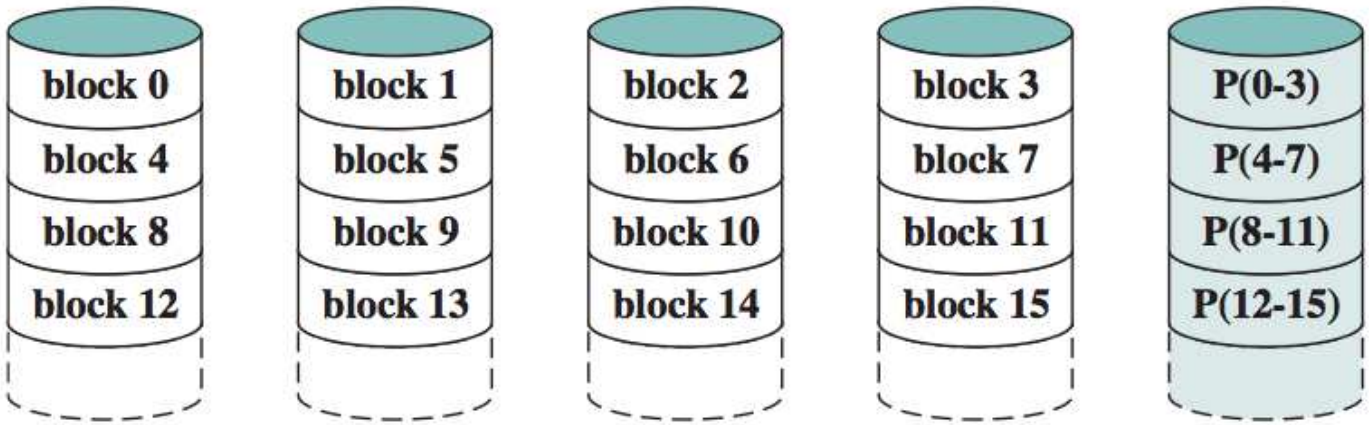

Figure: Data Mapping for a RAID Level 4 Array (Block-level parity). (Source: (Stallings, 2015))

Bit-by-bit parity calculated across corresponding strips on each disk:

- Parity bits are stored in the corresponding strip on the parity disk:

$$X_4(i) = X_0(i) \oplus X_1(i) \oplus X_2(i) \oplus X_3(i)$$

  - X_0, X_1, X_2, X_3 represent the data hard disks;
  - X_4 is the redundancy hard disk
  - i represents the i-th bit.

RAID 5

- Similar to RAID 4
  - Difference: RAID 5 distributes the parity strips across all disks.

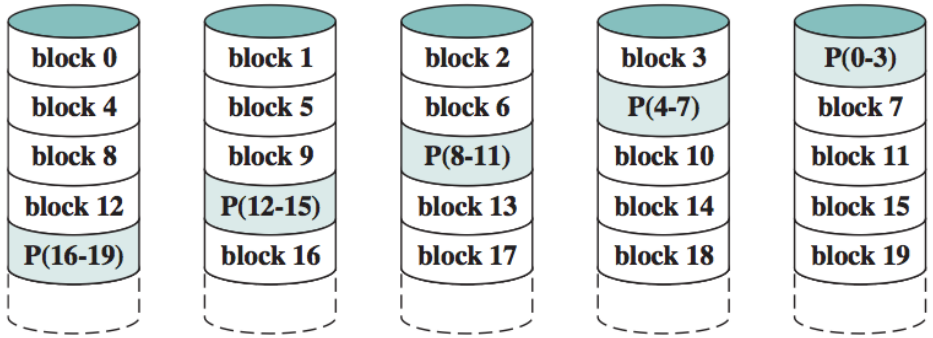

Figure: Data Mapping for a RAID Level 5 Array. (Source: (Stallings, 2015))

  - Distribution of parity across all drives avoids potential I/O bottle-neck:
    - Instead of having a single hard disk with the parity information...
    - Concurrent access to the parity bits are multiplexed;

RAID 6

- Two different parity calculations are carried out and stored in separate blocks on different disks.
  - Idea: Regenerate data even if two disks containing user data fail.

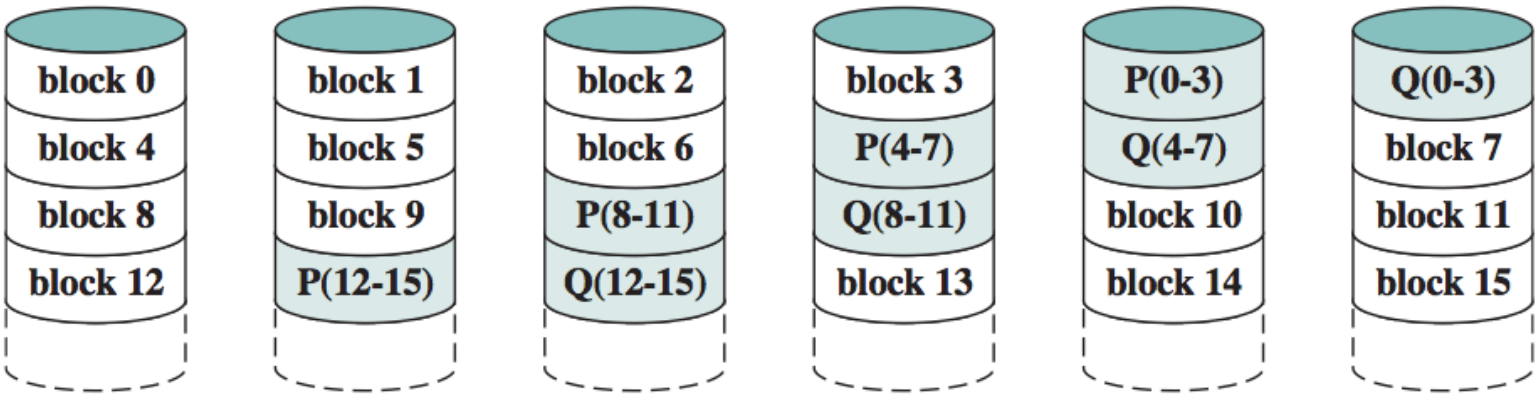

Figure: Data Mapping for a RAID Level 6 Array (Dual redundancy). (Source: (Stallings, 2015))

A brief summary of all the info regarding RAID levels:

Category	Level	Description	Disks Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

Note: N = number of data disks; m proportional to log N

Figure: RAID levels (Source: (Stallings, 2015))

Solid State Drives (SSD)

- SSD is made with solid state components:
  - Use of flash memory (instead of magnetic disk option)
  - In recent years the cost and performance of flash memory has evolved.
- Increased use of solid state drives (SSDs) to complement or even replace hard disk drives (HDDs);
  - Dedicated SSD;
  - SSD + HD;
  - Hybrid HD (SSD + HD).
- Significant development in external memory;

What do you think are the advantages of using an SSD instead of a HDD?

SSD compared to HDD

SSDs have the following advantages over HDDs (1/2):

- Lower access times and latency rates. Why?
- High-performance input/output operations per second. Why?
- Durability. Why?

SSD compared to HDD

SSDs have the following advantages over HDDs (1/2):

- Lower access times and latency rates. Why?
  - Due to having no mechanical parts;
- High-performance input/output operations per second. Why?
  - Significantly increases performance I/O subsystems.
  - Again due to having no mechanical parts;
- Durability. Why?
  - Less susceptible to physical shock and vibration;
  - Again due to having no mechanical parts;

SSD compared to HDD

SSDs have the following advantages over HDDs (2/2):

- Longer lifespan. Why?
- Lower power consumption. Why?
- Quieter and cooler running capabilities. Why?

SSD compared to HDD

SSDs have the following advantages over HDDs (2/2):

- Longer lifespan. Why?
  - SSDs are not susceptible to mechanical wear.
- Lower power consumption. Why?
  - SSDs use as little as 2.1 watts of power per drive.
- Quieter and cooler running capabilities. Why?
  - Less floor space required, lower energy costs, and a greener enterprise.

What do you think are the disadvantages of using an SSD?

What do you think are the disadvantages of using an SSD?

- Cost per bit is high;
- Capacity is low;
- Not for long ;)

	NAND Flash Drives	Disk Drives
I/O per second (sustained)	Read: 45,000 Write: 15,000	300
Throughput (MB/s)	Read: 200+ Write: 100+	up to 80
Random access time (ms)	0.1	4–10
Storage capacity	up to 256 GB	up to 4 TB

Figure: Comparison of Solid State Drives and Disk Drives. (Source: (Stallings, 2015))

Disadvantage of mechanical vs. digital in terms of speed.

SSD contains the following components:

- **Controller**: Device level interfacing and firmware execution;
- **Addressing**: Logic to select the flash memory components;
- **Data buffer/cache**: To further increase performance;
- **Error correction**: Logic for error detection and correction;
- **Flash memory components**: Individual NAND flash chips.

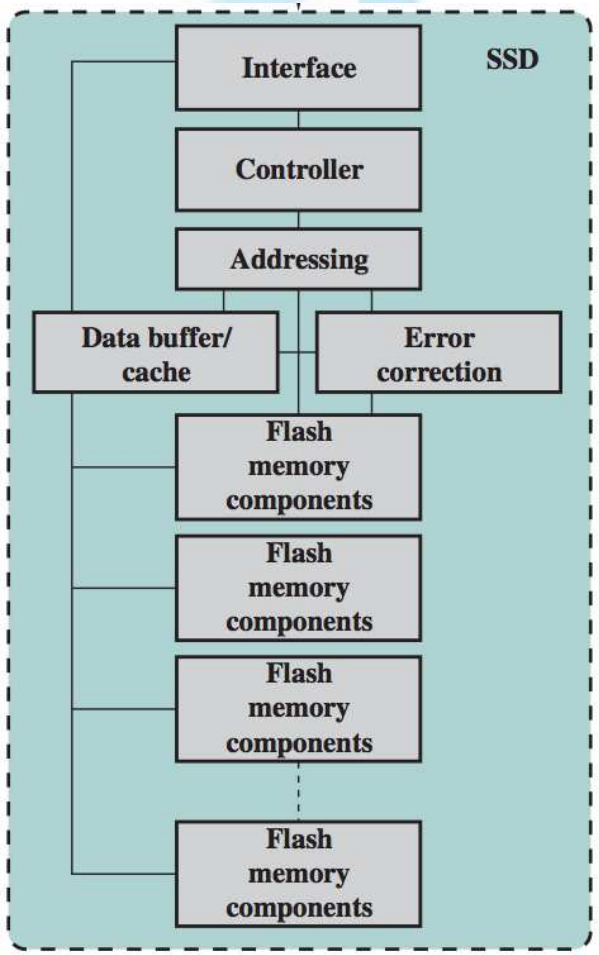

Figure: Solid State Drive Architecture. (Source: (Stallings, 2015))

SSD Practical Issues

Performance has a tendency to slow down as the device is used (1/2):

- Flash memory is accessed in blocks containing memory pages;
- Consider what must be done to write a page onto a flash memory:
  1. Block must be read from flash memory and placed in a RAM buffer;
  2. Appropriate page in the RAM buffer is updated;
  3. Before block can be written back:
     - Entire block of flash memory must be erased;
     - Not possible to erase just one page of the flash memory.
  4. Entire block from the buffer is now written back to the flash memory.

SSD Practical Issues

Performance has a tendency to slow down as the device is used (2/2):

- Over time files become fragmented:
  - File pages are scattered over multiple blocks;
  - This means that we will erase blocks to update few pages;
  - At the beginning with less fragmentation:
    - Multiple pages would be written for each block erasure;
- File pages not contiguously in memory $\rightarrow$ loss of efficiency;
- Several techniques exist to compensate for this property of flash memory;

- Worse than fragmentation issues:
  - Flash memory becomes **unusable** after a certain number of writes.
  - Depends on the quality of the flash memory (good estimate 10^5 writes)
  - There are also a variety of techniques to handle the life of flash memory:
    - A cache to delay and group write operations;
    - Wear-leveling algorithms: evenly distribute writes across block of cells;
    - Bad-block management techniques: when eventually blocks do *kaputz*!

Where to focus your study

After this class you should be able to:

- Understand the key properties of magnetic disks.
- Understand the performance issues involved in magnetic disk access.
- Explain the concept of RAID, importance of redundancy and mechanisms for redundancy;
- Compare and contrast hard disk drives and solid disk drives.
- Describe in general terms the operation of flash memory.

Less important to know:

- details of all RAID levels;
- details of specific magnetic disk formats;

Your focus should always be on the building blocks for developing a solution =)

References I

Stallings, W. (2015).

Computer Organization and Architecture.

Pearson Education.