Lecture 1  Disks & Filesystems
  ▶ Revisions
  ▶ Performance
  ▶ Limitations and solutions

Lecture 2  RAID
  ▶ build server filestore from (inexpensive) PC parts

Lecture 3  Storage Systems and Virtualization
  ▶ Logical Volume Management
  ▶ Storage Area Networks
  ▶ Solid State Disks
Learning Objectives - Storage 1

- Review disk and file system characteristics
- Understand the operational limitations of conventional disk usage
- Introduce simple solutions using multiple disks
Characterisation

- Write Once, Read Many (times) – WORM
  - CD-ROM, DVD, Blu-ray Disc
  - Irreversible writes

- Write Many, Read Many
  - Hard disk drive, tape drive
  - Fully reversible writes (almost)

- Write (not too) Many, Read Many
  - CD/DVD±RW (100s to 1000s)
  - Flash Memory (1000s to …)
  - Mostly reversible writes – “wear”
HDD Internals – tinyurl.com/disk-video

Source: http://systemspro.blogspot.co.uk/2011/09/hard-disk-drive.html
Hard Disk Drive Storage Structure

Capacity

- 2TB platter (2012/13)
- 8TB HDD (Seagate 2014)
- 10TB (WD HGST 2015)
- 12TB (WD HGST 2017)
- 60TB (Seagate TBC - SSD!)
Price trends

Dollars per Gigabyte

Source: Gartner, Market Trends: Evolving HDD and SSD Storage Landscapes (October 2013)
Price trends

SSD Cost per Gigabyte over Time

USD/GB

Price trends

Backblaze Average Cost per Drive Size

By Quarter: Q1 2009 - Q2 2017
Hard Disk Drive Storage Structure

- **Capacity**
  - 2TB platter (2012/13)
  - 8TB HDD (Seagate 2014)
  - 10TB (WD HGST 2015)
  - 12TB (WD HGST 2017)
  - 60TB (Seagate TBC - SSD!)

- **Power consumption**
  - Spinning platters
  - Moving the heads (seek)
  - Reading/Writing
  - Controllers
  - Data transfer (I/O)

- **Rotation speed**
  - 5400/7200/10000/15000

Hard Disk Attributes – *Performance*

Seek time  Time for the **head** to reach the target **track**.

Search time  Time for the target **sector** to arrive under the **head**. Also called *rotational latency*.

Transfer rate  Amount of data that can be read / written per unit of time. Dependent on access patterns.

Aka. “sustained transfer rate” in contrast to “interface transfer rate”

\[
\text{Disk access time} = \text{seek time} + \text{search time} + \text{transfer time}
\]

Note: all values are average as they depend on many factors.
Disk access example

- Host initiates read sends a list of blocks to read
- Block schedule requested...
- ... may not be optimal
- and leads to extra revolutions
- HDD internal processor optimizes the schedule
- No direct mapping from block numbers to the sector/track/cylinder position (high-level interfaces like ATA / SCSI)
Example HDD specs

HGST Western Digital He6 HUS726060ALA640

- Capacity 6TB
- Power consumption: 7.3/5.3/3.7 W
- Rotational speed: 7200 RPM
- Seek time: 8.5 ms
- Sustained transfer rate: 177 MB/sec
- Interface transfer rate: 600 MB/sec (SATA)
- Data buffer: 64 MB
- MTBF: 2,500,000 hours
- Price: £250 to £400 (Q1 2015)

- Similar spec. Q1 2018: £155
  Seagate IronWolf 6 TB ST6000VN0033 - SATA 6Gb/s - 7,200 rpm
How long would it take on average to read / write a 512 byte sector on this disk?

**Disk access time = seek time + search time + transfer time**

seek time: 8.5 ms

search time: the disk must, on average, complete a half rotation

\[
\text{7200 RPM} \quad \Rightarrow \quad \frac{0.5 \text{ rotations} \cdot 60 \text{ sec}}{7200 \text{ RPM}} = 4.16 \text{ ms}
\]

transfer time: \[
\frac{512 \text{ B}}{177 \cdot 10^6 \text{ B/sec}} = 2.89 \mu s
\]

access time = 8.5 + 4.16 + 2.89 \cdot 10^{-3} = 12.66 ms
Example: disk access time (2)

How long would it take on average to read / write 512 MB on this disk? (assuming sectors are “contiguous”)

Disk access time = seek time + search time + transfer time

seek time: 8.5 ms

search time: the disk must, on average, complete a half rotation

\[
7200 \text{ RPM} \quad \Rightarrow \quad \frac{0.5 \text{ rotations} \cdot 60 \text{ sec}}{7200 \text{ RPM}} = 4.16 \text{ ms}
\]

transfer time: \[
\frac{512 \cdot 10^6 \text{ B}}{177 \cdot 10^6 \text{ B/sec}} = 2.89 \text{ s}
\]

access time = \[
8.5 \cdot 10^{-3} + 4.16 \cdot 10^{-3} + 2.89 = 2.9 \text{ s}
\]
File System Review

- **Naming service**
  - files
  - directories
  - links
- **Storage service**
  - “vector of bytes”
  - owners, permissions...
- **Data and metadata**
- **Space allocation**
  - contiguous
  - linked
  - indexed
- **Recovery**
  - chkdsk, fsck
Problems with disks

Small  Slow  Unreliable
Disks are (were?) too small

1956 first HDD IBM 350: \( \sim 3.5 \text{ MB} \) (enough to store one selfie!)
2015 first 10 TB disk: 1000s of times smaller, \( 3 \cdot 10^6 \times \) capacity

\( 10^{10} \) higher storage density in 60 years: is this enough?

Source: https://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html
If one disk is not enough ... 

Use multiple disks

- Independent disks
- Can we have a single volume with the combined capacity?
- Storage virtualization

Redundant Array of Independent Disks
Disks are too slow

Slow because of:

- High seek time
  - Reduce the number of times the head must move
  - Multiple platters $\Rightarrow$ more tracks/sectors per cylinder
- High search time (aka. rotational latency)
  - Increase the rotation speed (e.g., server disks up to 15000 RPM)
- Low sustained transfer rate
  - Increase rotation speed (physical limitations)
  - Increase the recording density (physical limitations)
  - Apply cache and prefetch principles
  - “Stripe” file system across multiple disks
Solution: Disk Striping (RAID 0)

- Split data evenly across multiple disks
- Distribute fixed-size “stripes” of a virtual volume
- Illusion of faster and larger disk

BUT lower reliability!
Disks are unreliable

- Mechanical components subject to wear
- Partial failure: sectors go bad
- Total failure: no data recoverable

- If reliability cannot be improved: **tolerate failures**
  - Fault-tolerance through redundancy
  - Disk “mirror”
Solution: Disk Mirroring (RAID 1)

- Use two (or more) redundant disks
- Write to each (same, replicated data)
- Read from either (possibly choose “nearest” for performance)
- If one fails: use the other and re-create a new copy (slowly)
Nested RAID: RAID 1+0 (aka. RAID 10)

- Operation continues in case of disk failure
- Can tolerate failures as long as no mirror loses all drives
Disks are too small
  ▶ Fixed: use multiple disks (possibly striped)

Disks are too slow
  ▶ Fixed: disk striping (RAID 0)

Disks are unreliable
  ▶ Fixed: disk mirroring (RAID 1)

Disks may be in the wrong place!
  ▶ What happens when we migrate a Virtual Machine?

Better solutions on Monday