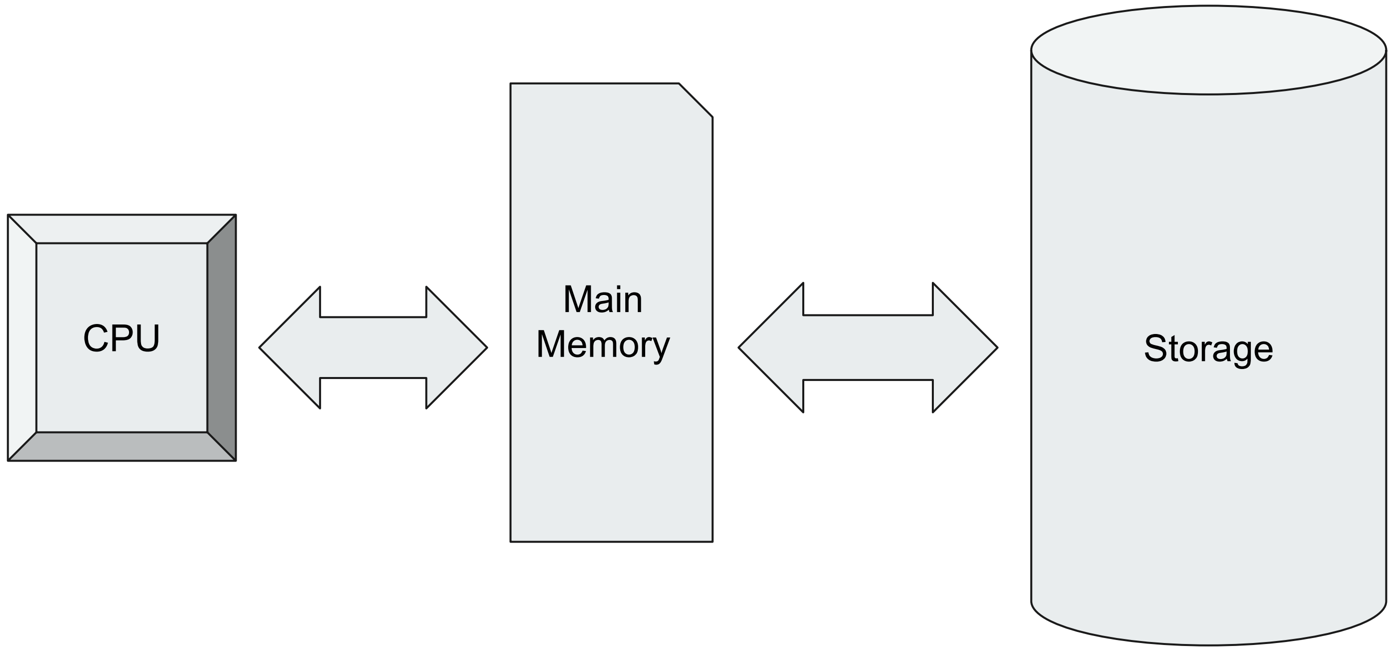
Mass storage and RAID

Dr. Naser Al Madi

Learning objectives

- Understand how mass storage works.
- Learn the different RAID configurations.

Mass-Storage Systems


CPU
Main Memory
Storage

The First Commercial Disk Drive

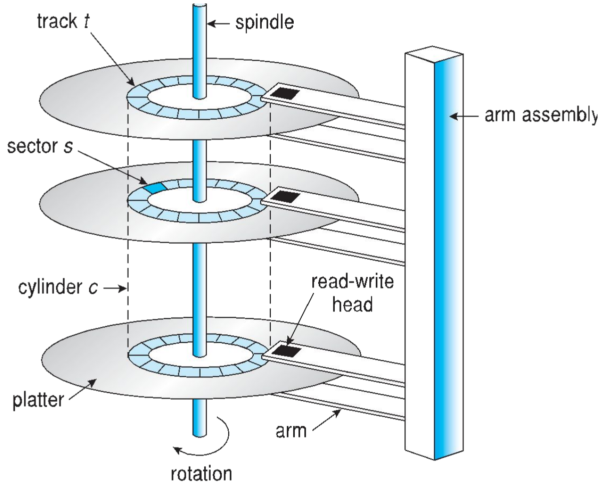
1956
IBM RAMDAC computer included the
IBM Model 350 disk storage system

5M (7 bit) characters
50 x 24” platters
Access time = < 1 second

Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 250 times per second
  - Transfer rate is rate at which data flow between drive and computer
- Drive attached to computer via I/O bus
  - Busses vary, including EIDE, ATA, SATA, USB, Fibre Channel, SCSI, SAS, Firewire

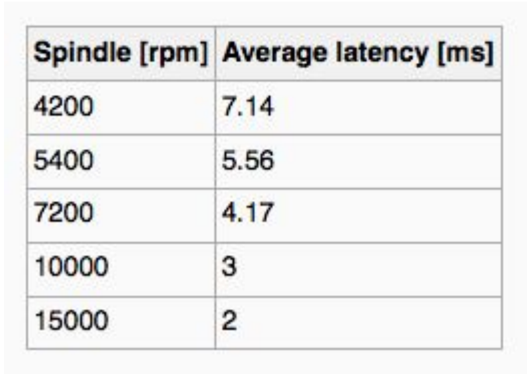
Moving-head Disk Mechanism



- Positioning time (**random-access time**) is: time to move disk arm to desired cylinder (**seek time**) + time for desired sector to rotate under the disk head (**rotational latency**)

Hard Disks

- Platters range from .85” to 14” (historically)
    - Commonly 3.5”, 2.5”, and 1.8”
- Range from 30GB to 3TB per drive



Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

(From Wikipedia)

Hard Disk Performance

- Access Latency = Average access time = average seek time + average latency
  - For fastest disk 3ms + 2ms = 5ms
  - For slow disk 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

Hard Disk Performance

For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:

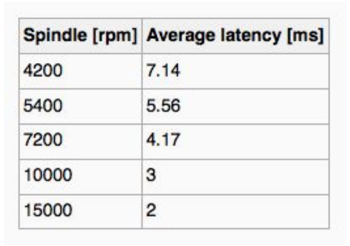
Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

Hard Disk Performance

For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:

Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

average access time = average seek time + average latency = 5ms + 4.17 ms = 9.27ms



Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

Hard Disk Performance

For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:

Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

average access time = average seek time + average latency = 5ms + 4.17 ms = 9.27ms

(amount to transfer / transfer rate) = 4kB / (1Gb/s) = 4000 * 8 / (1,000,000,000/s) = 0.031 ms

Hard Disk Performance

For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:

Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

average access time = average seek time + average latency = 5ms + 4.17 ms = 9.27ms

(amount to transfer / transfer rate) = 4kB / (1Gb/s) = 4000 * 8 / (1,000,000,000/s) = 0.031 ms

controller overhead = .1ms

Hard Disk Performance

For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:

Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead

average access time = average seek time + average latency = 5ms + 4.17 ms = 9.27ms

(amount to transfer / transfer rate) = 4kB / (1Gb/s) = 4000 * 8 / (1,000,000,000/s) = 0.031 ms

controller overhead = .1ms

Average I/O time for 4KB block = 9.27ms + .031ms + .1ms = 9.301ms

Solid-State Disks

- Nonvolatile memory used like a hard drive
    - Many technology variations
- More expensive per MB
- May have shorter life span
- Less capacity
- But much faster
- Busses can be too slow -> connect directly to PCI for example
- No moving parts, so no seek time or rotational latency

Intel 670p Series M.2 2280 1TB PCIe NVMe 3.0 x4 QLC Internal Solid State Drive (SSD)

Magnetic Tape

- Was early secondary-storage medium
  - Evolved from open spools to cartridges
- Relatively permanent and holds large quantities of data
- Access time slow
- Random access ~1000 times slower than disk
- Mainly used for backup, storage of infrequently-used data, transfer medium between systems
- 200GB to 1.5TB typical storage

Disk Structure

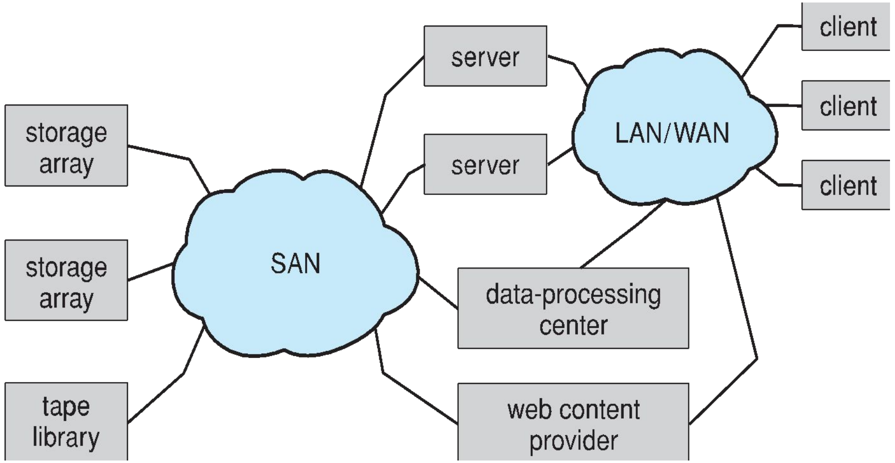
- Disk drives are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer
  - Low-level formatting creates logical blocks on physical media
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially
  - Sector 0 is the first sector of the first track on the outermost cylinder
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost
  - Logical to physical address should be easy
    - Except for bad sectors
    - Non-constant # of sectors per track via constant angular velocity

Storage Array

- Can just attach disks, or arrays of disks
- Storage Array has controller(s), provides features to attached host(s)
  - Ports to connect hosts to array
  - Memory, controlling software (sometimes NVRAM, etc)
  - A few to thousands of disks
  - RAID
  - Shared storage -> more efficiency
  - Features found in some file systems
    - Snapshots, clones, thin provisioning, replication, etc

Storage Area Network

- Common in large storage environments
- Multiple hosts attached to multiple storage arrays - flexible

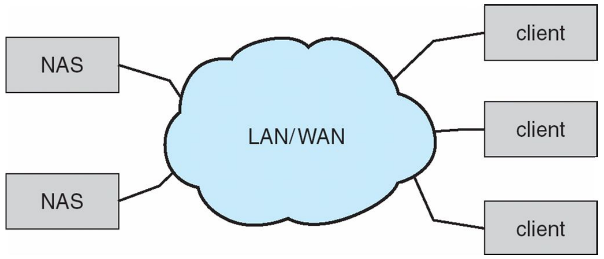


Storage Area Network (Cont.)

- SAN is one or more storage arrays
  - Connected to one or more Fibre Channel switches
- Hosts also attach to the switches
- Easy to add or remove storage, add new host and allocate it storage
  - Over low-latency Fibre Channel fabric
- Why have separate storage networks and communications networks?

Network-Attached Storage

- Network-attached storage (NAS) is storage made available over a network rather than over a local connection (such as a bus)
    - Remotely attaching to file systems
- NFS and CIFS are common protocols
- Implemented over typically TCP on IP network



Differences

Storage Area Network (SAN)

- Very very fast (fibre optics)
- Professional use
- Expensive

Network-Attached Storage (NAS)

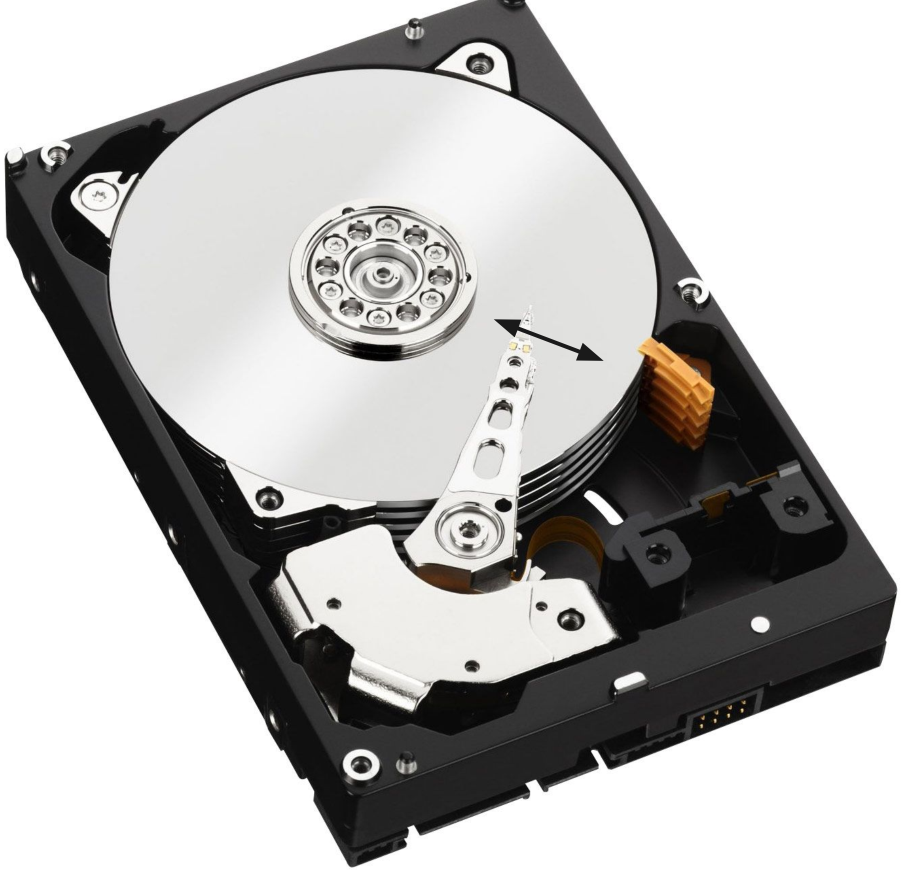
- Network speed
- Home/small business use
- Less expensive

Read more: https://www.backblaze.com/blog/whats-the-diff-nas-vs-san/

Disk Scheduling

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
- Minimize seek time
- Seek time ≈ seek distance
- Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer



Disk Scheduling (Cont.)

- There are many sources of disk I/O request
    - OS
    - System processes
    - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
    - Optimization algorithms only make sense when a queue exists

Disk Scheduling (Cont.)

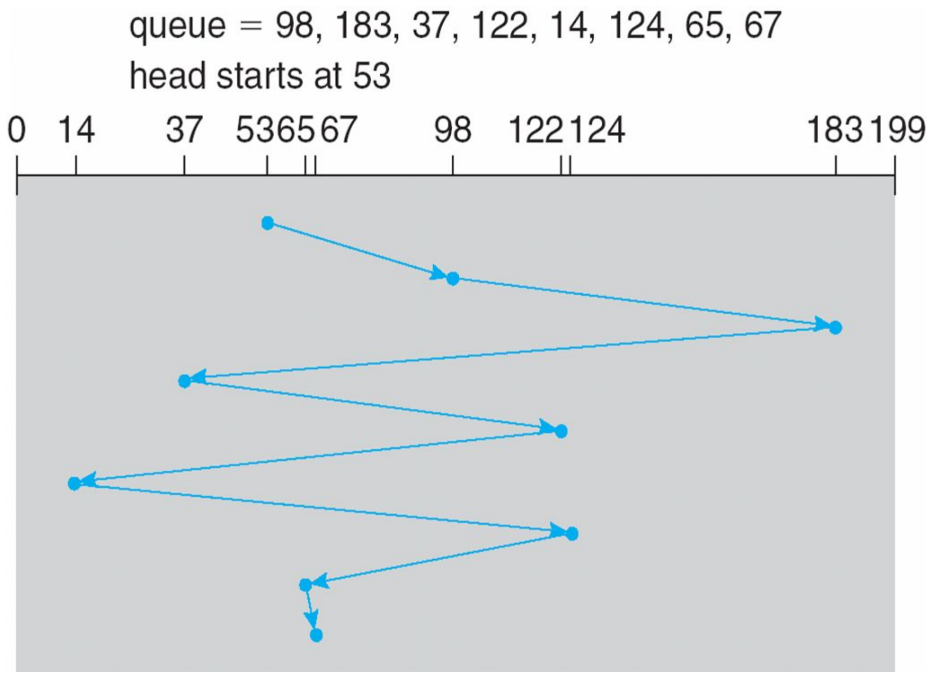
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying "depth")
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53

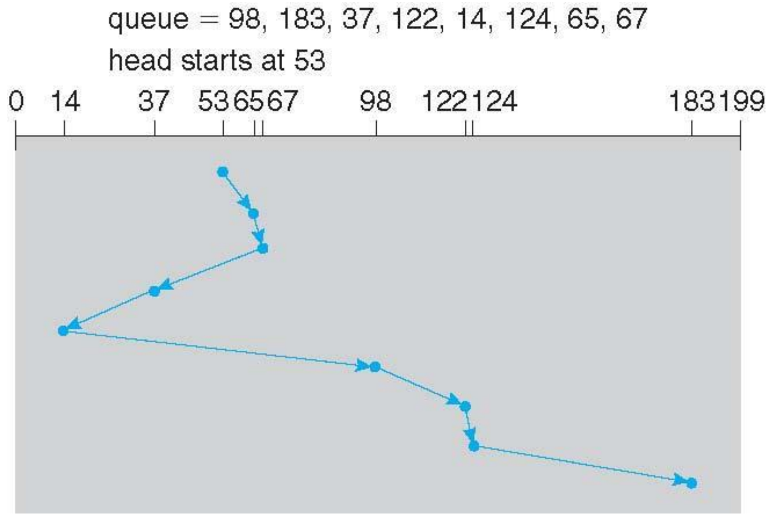
FCFS

Illustration shows total head movement of 640 cylinders



SSTF

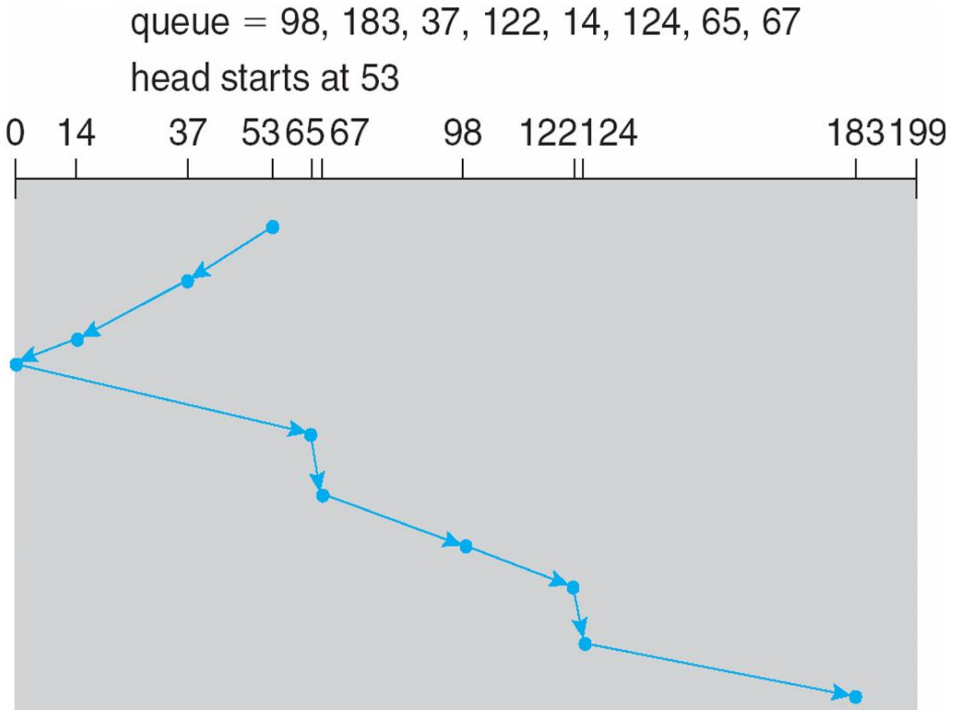
- Shortest Seek Time First selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests
- Illustration shows total head movement of 236 cylinders



SCAN

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- SCAN algorithm Sometimes called the elevator algorithm
- Illustration shows total head movement of 208 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

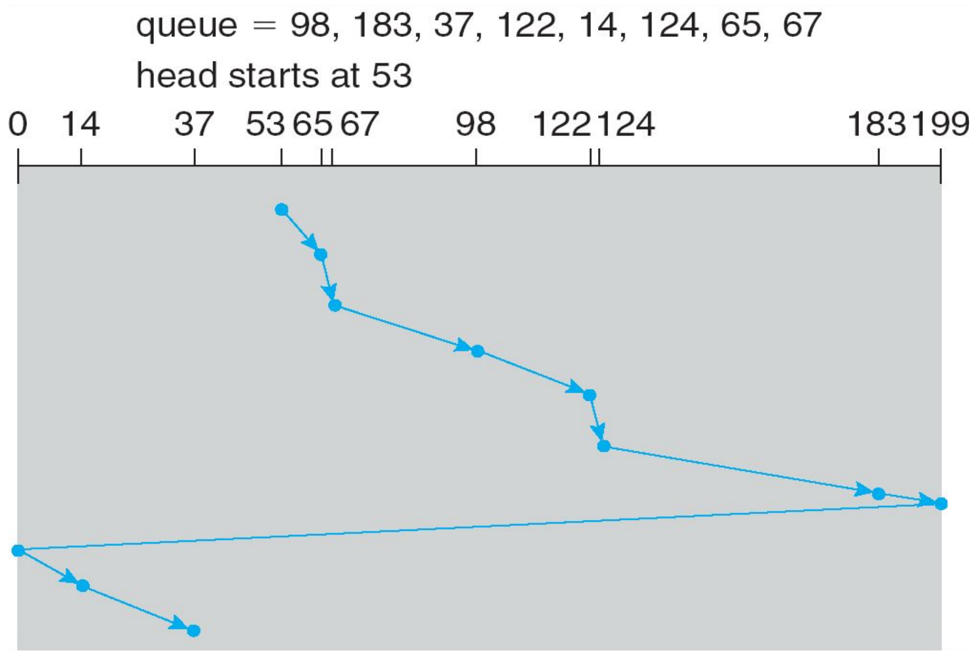
SCAN (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-SCAN

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other, servicing requests as it goes
    - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Total number of cylinders?

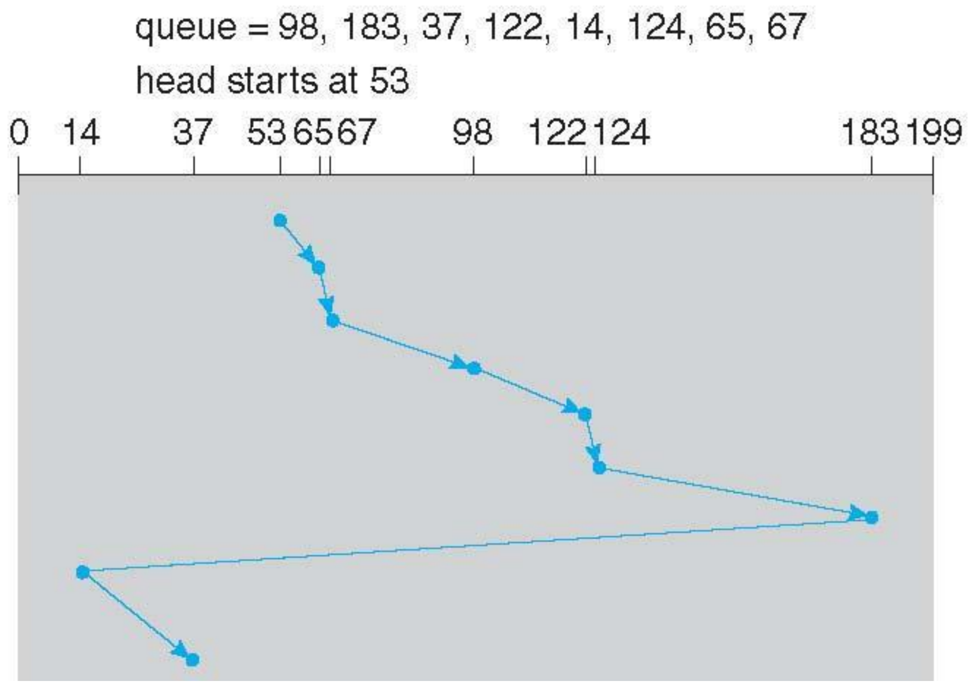
C-SCAN (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-LOOK

- LOOK a version of SCAN, C-LOOK a version of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk
- Total number of cylinders?

C-LOOK (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
    - Less starvation
- Performance depends on the number and types of requests
- Either SSTF or LOOK is a reasonable choice for the default algorithm

RAID: Redundant Array of Independent/Inexpensive Disks

Outline

- RAID 0
- RAID 1
- RAID 10 (1 + 0)
- RAID 3
- RAID 4
- RAID 5
- RAID 6

Why multi-disk systems?

A single storage device may not provide:

- enough storage capacity
- performance capacity
- reliability

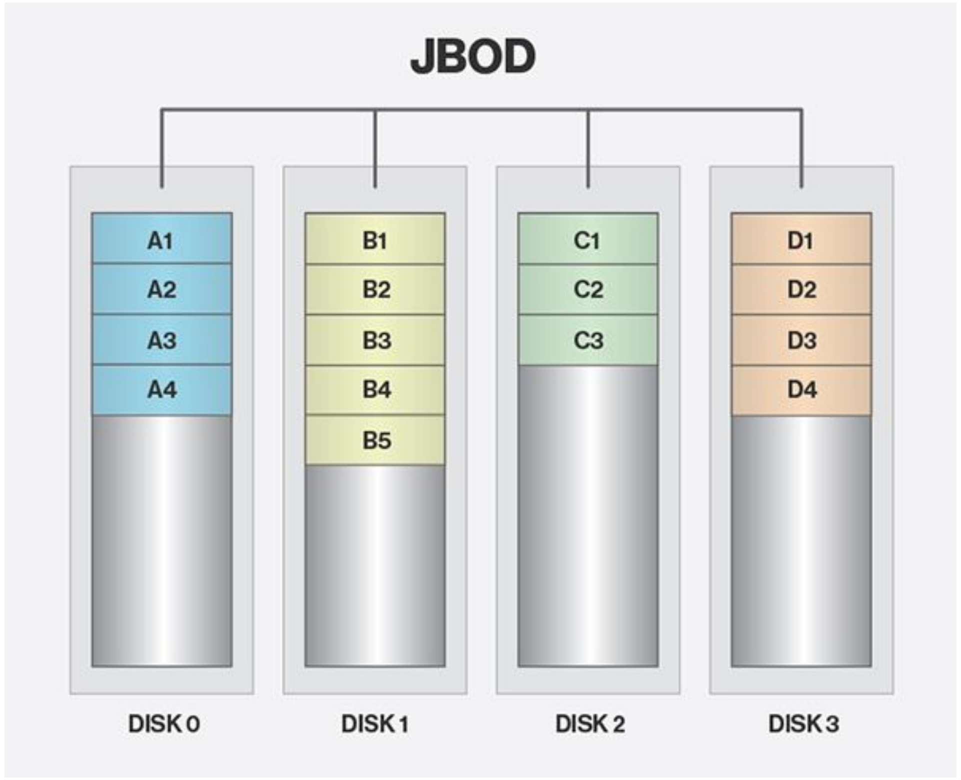
Why multi-disk systems?

A single storage device may not provide:

- enough storage capacity
- performance capacity
- reliability

So, what is the simplest arrangement?

Just a bunch of disks (JBOD)



Load Balancing

I/O requests are almost never evenly distributed

- Some data is requested more than other data
- Depends on the apps, usage, time, …

Load Balancing

I/O requests are almost never evenly distributed

- Some data is requested more than other data
- Depends on the apps, usage, time, …

What is the right data-to-disk assignment policy?

- Common approach: Fixed data placement (Your data is on disk X, period!)

Load Balancing

I/O requests are almost never evenly distributed

- Some data is requested more than other data
- Depends on the apps, usage, time, …

What is the right data-to-disk assignment policy?

- Common approach: Fixed data placement (Your data is on disk X, period!)
- Fancy: Dynamic data placement
    - If some of your files are accessed a lot, the admin (or even system) may separate the “hot” files across multiple disks
    - In this scenario, entire files systems (or even files) are manually moved by the system admin to specific disks

Load Balancing

I/O requests are almost never evenly distributed

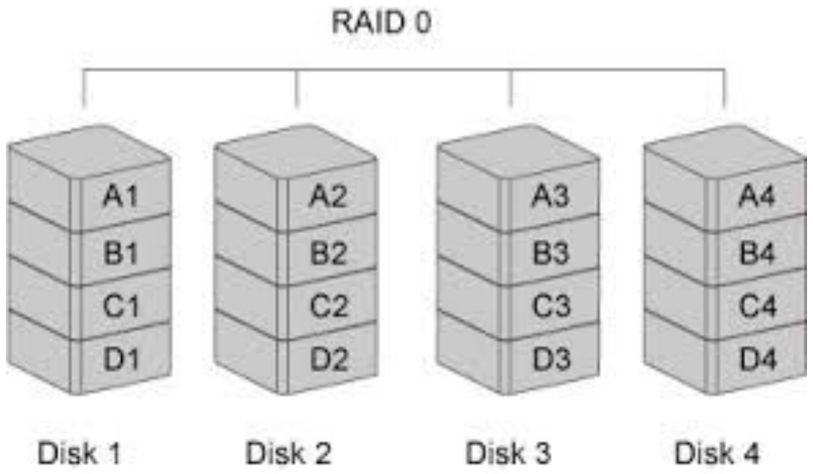
- Some data is requested more than other data
- Depends on the apps, usage, time, …

What is the right data-to-disk assignment policy?

- Common approach: Fixed data placement (Your data is on disk X, period!)
- Fancy: Dynamic data placement
  - If some of your files are accessed a lot, the admin (or even system) may separate the “hot” files across multiple disks
  - In this scenario, entire files systems (or even files) are manually moved by the system admin to specific disks
- Alternative: Disk striping - Stripe all of the data across all of the disks

RAID level 0: Disk striping

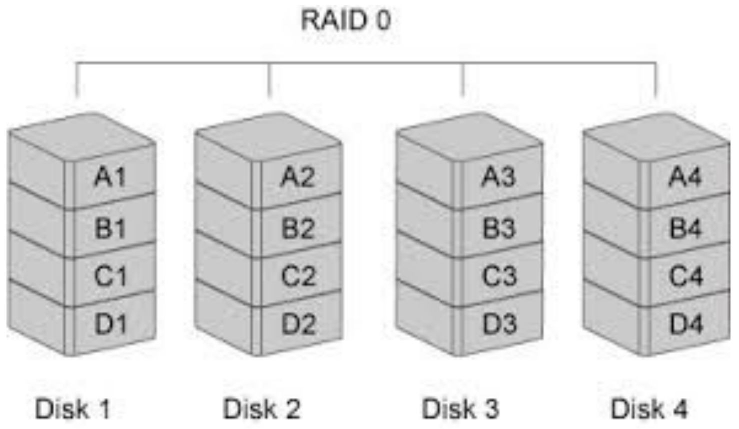
- Interleave data across multiple disks (striping)
- Large file streaming can enjoy parallel transfers



RAID level 0: Disk striping

- Interleave data across multiple disks (striping)
- Large file streaming can enjoy parallel transfers

1 disk transfer rate = 1Gb/s

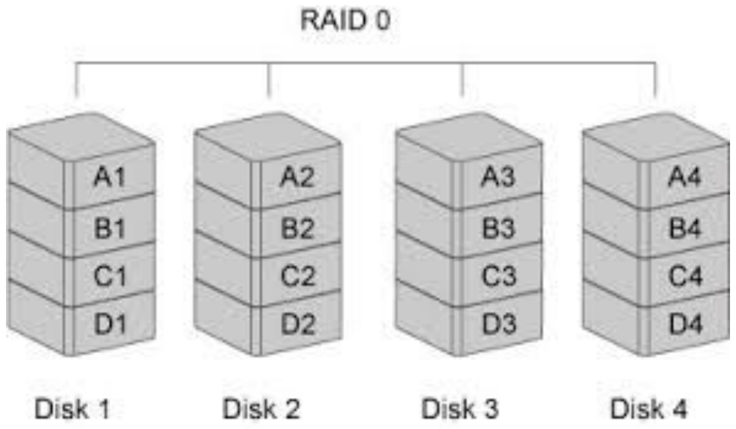


RAID level 0: Disk striping

- Interleave data across multiple disks (striping)
- Large file streaming can enjoy parallel transfers

1 disk transfer rate = 1Gb/s

4 disks in RAID = [up to] 4 Gb/s



Fine grained or coarse grained data interleaving

Fine grained disk arrays conceptually interleave data in relatively small units so that all I/O requests, regardless of their size, access all of the disks in the disk array. This results in very high data transfer rate for all I/O requests but has the disadvantages that only one logical I/O request can be in service at any given time and all disks must waste time positioning for every request.

Coarse grained disk arrays interleave data in relatively large units so that small I/O requests need access only a small number of disks while large requests can access all the disks in the disk array. This allows multiple small requests to be serviced simultaneously while still allowing large requests to see the higher transfer rates afforded by using multiple disks.

Fine grained or coarse grained data interleaving

For small files, coarse grained might be faster!

Example on next slide.

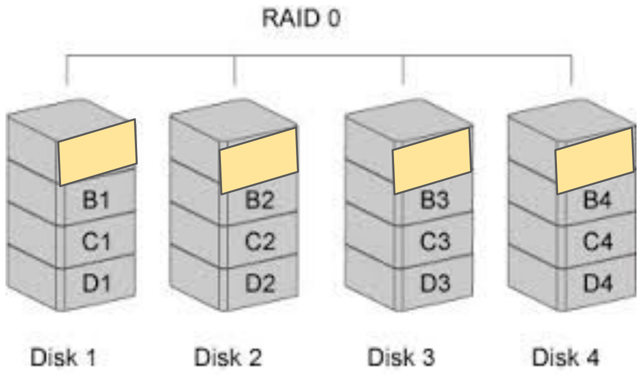
Fine grained or coarse grained example

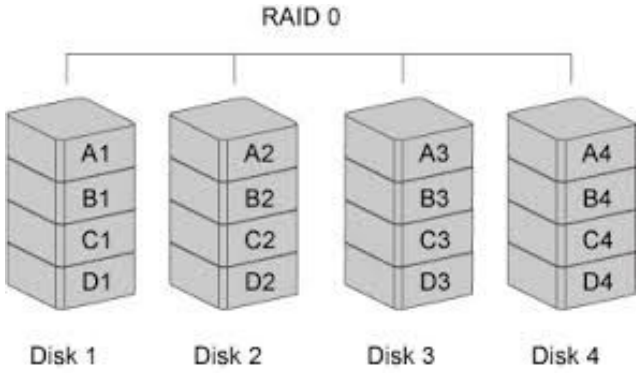
Fine grained: File is distributed on <u>many</u> disks

- random-access time X 4
  4 ms X 4 = 16 ms

- + file transfer time (4 ms in total = 1 ms per disk)
  1 ms X 4 = 4 ms

  Total = 16 ms + 4 ms = 20 ms





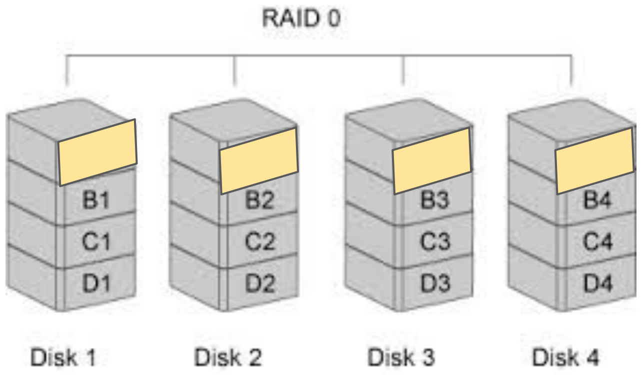
Fine grained or coarse grained example

Fine grained: File is distributed on many disks

- random-access time X 4
  4 ms X 4 = 16 ms

- + file transfer time (4 ms in total = 1 ms per disk)
  1 ms X 4 = 4 ms

  Total = 16 ms + 4 ms = 20 ms

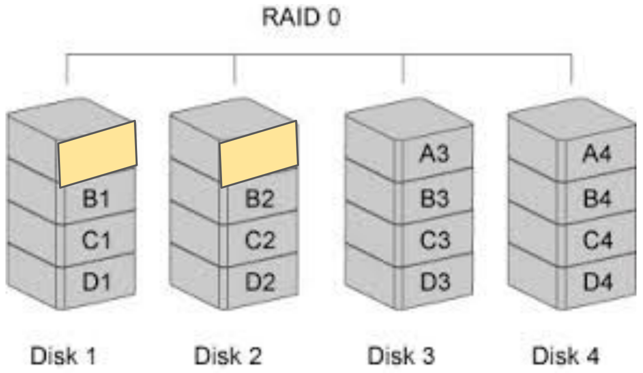


coarse grained: File is distributed on few disks

- random-access time X 2
  4 ms X 2 = 8 ms

- + file transfer time (4 ms in total = 2 ms per disk)
  2 ms X 2 = 4 ms

  Total = 8 ms + 4 ms = 12 ms



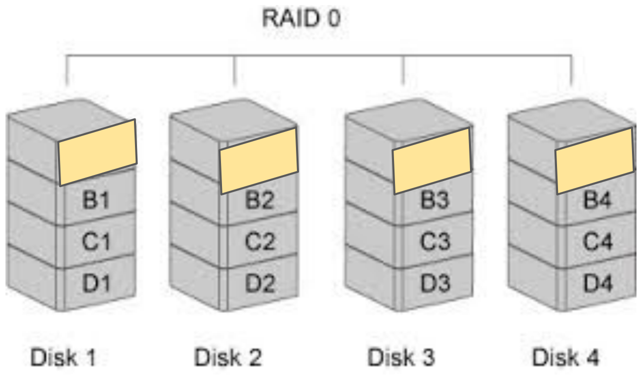
Fine grained or coarse grained example

Fine grained: File is distributed on <u>many</u> disks

- random-access time X 4
  4 ms X 4 = 16 ms

- + file transfer time (4 ms in total = 1 ms per disk)
  1 ms X 4 = 4 ms

  Total = 16 ms + 4 ms = 20 ms

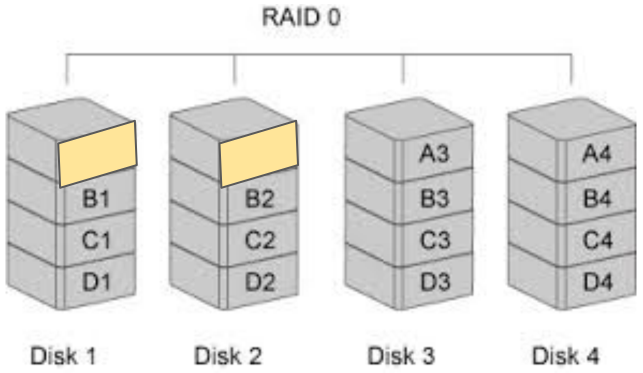


coarse grained: File is distributed on <u>few</u> disks

- random-access time X 2
  4 ms X 2 = 8 ms

- + file transfer time (4 ms in total = 2 ms per disk)
  2 ms X 2 = 4 ms

  Total = 8 ms + 4 ms = 12 ms

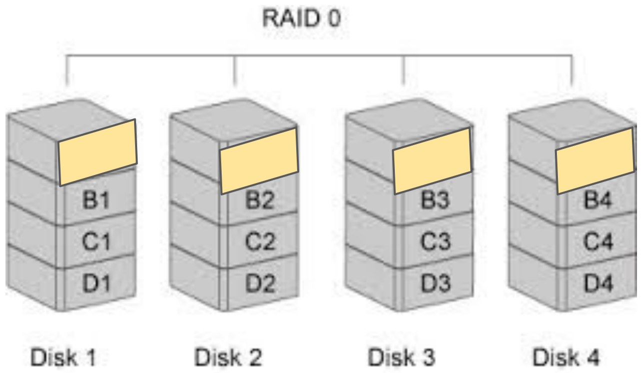


Fine grained or coarse grained example

Fine grained: File is distributed on many disks

- random-access time X 4
  4 ms X 4 = 16 ms
- + file transfer time (4 ms in total = 1 ms per disk)
  1 ms X 4 = 4 ms

  Total = 16 ms + 4 ms = 20 ms

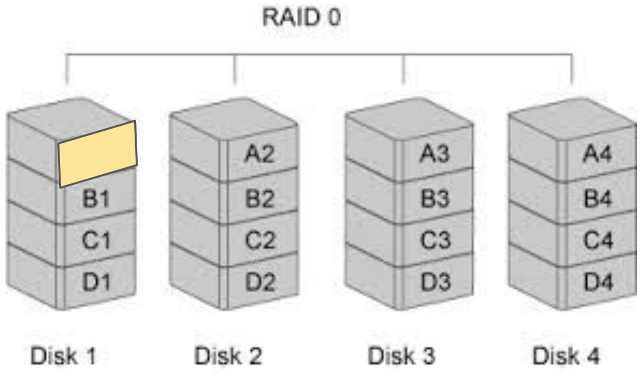


coarse grained: File is distributed on few disks

Even better:

- random-access time X 1
  4 ms X 1 = 4 ms
- + file transfer time (4 ms in total)
  4 ms X 1 = 4 ms

  Total = 4 ms + 4 ms = 8 ms



Disk striping details

How disk striping works?

- Breakup total space into fixed-size stripe units
- Distribute the stripe units among disks in round-robin
- Compute location of block B as follows:

  disk# = B % N (N = # of disks)

Now, What If A Disk Fails?

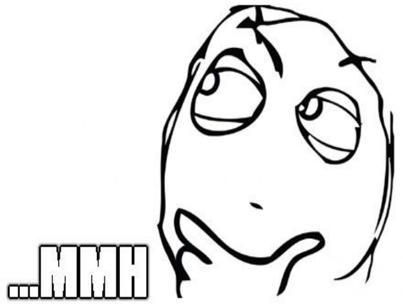
- In a JBOD (independent disk) system
  - one or more file systems lost
- In a striped system (Raid 0)
  - a part of each file system lost

- Backups can help, but backing up takes time and effort
- backup doesn’t help recover data lost during that day
- any data loss is a big deal to a bank or stock exchange

Now, What If A Disk Fails?

- In a JBOD (independent disk) system
  - one or more file systems lost
- In a striped system
  - a part of each file system lost

- Backups can help, but backing up takes time and effort
- backup doesn’t help recover data lost during that day
- any data loss is a big deal to a bank or stock exchange



Disk drive failure is top bugbear for IT pros

By: John Leyden
Posted: 13/03/2001 at 16:17 GMT

Hard drive crashes are the number one concern for systems administrators in charge of keeping storage systems up and running.

That's the conclusion of a survey of 900 IT professionals, 61 per cent of which rated hard-disk failure as their most pressing concern when it came to hard drive problems. Running out of disk drive space was cited as the second most important issue, and was rated as their top bug bear by 27 per cent of respondents to the survey.

The study, conducted by Survey.com for Executive Software, also reported that managers estimated their direct cost of a hard-drive failure at $15,000 per incident, a figure which doesn't include lost productivity and or the effects on sales while systems are down.

Reliability metric

Mean Time Between Failures (MTBF):

- Usually computed by dividing a length of time by the number of failures during that time (averaged over a large population of items)

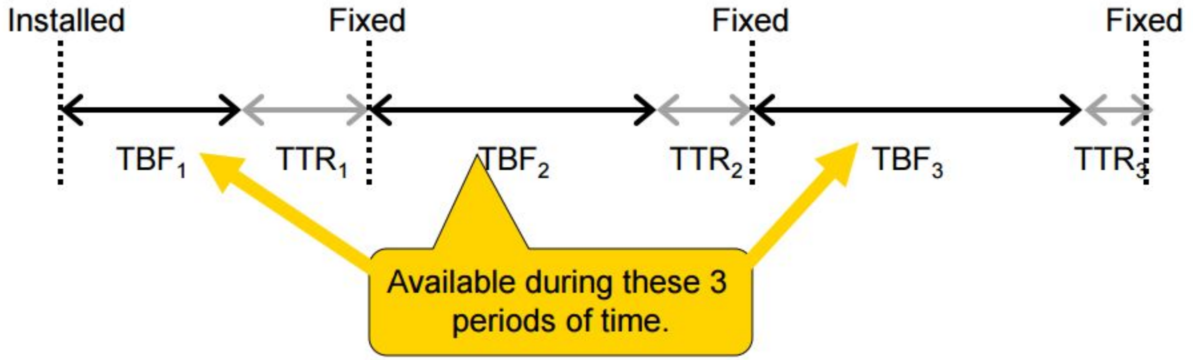
$$\text{MTBF} = \frac{\text{1000 days}}{\text{5 failures / 10 devices}} = \text{2000 days per device}$$

Availability metric

Fraction of time that server is able to handle requests

- Computed from MTBF and MTTR (Mean Time To Repair)

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} - \text{MTTR}}$$



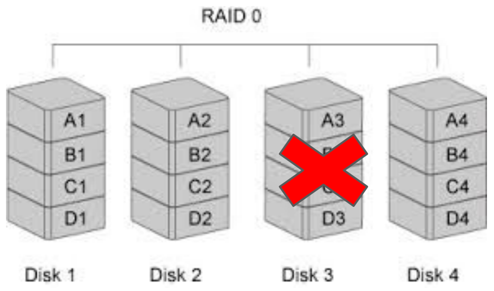
How often are failures?

MTBF (Mean Time Between Failures)

- MTBF disk ~ 1,200,000 hours (~136 years, <1% per year)
- MTBF multi-disk system = mean time to first disk failure
- which is MTBF disk / (number of disks)

For a striped array of 200 drives:

- MTBF array = 136 years / 200 drives = 0.65 years



Tolerating and masking disk failures

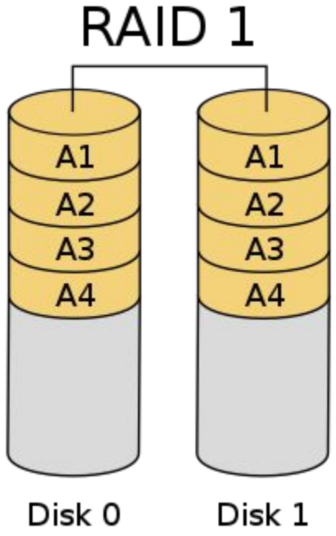
- If a disk fails, it’s data is gone
  - may be recoverable, but may not be
- To keep operating in face of failure
  - must have some kind of data redundancy

- Common forms of data redundancy
  - Replication
  - Parity codes
  - Error-correcting codes

RAID level 1: Redundancy via replicas

Two (or more) copies: mirroring, shadowing, duplexing, etc.

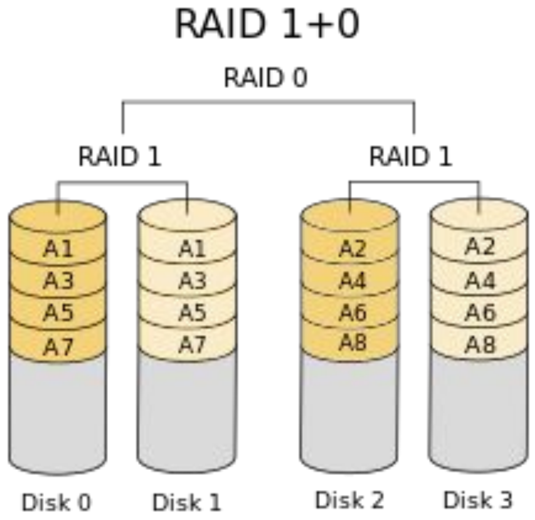
- No striping
- Low performance
- Capacity / 2
- Reliability is most important



RAID 1 + 0 or RAID 10: Mirroring & Striping

Mirror to 2 virtual drives, where each virtual drive is really a set of striped drives
Provides reliability of mirroring

Provides striping for performance (with write update costs)



Virtualbox and Final Project Options

References

- Operating systems concepts 9th edition
- http://www.ecs.umass.edu/ece/koren/architecture/Raid/striping.html
- https://www.cs.cmu.edu/~dga/15-440/F12/lectures/raid-guestlec.pdf
- https://en.wikipedia.org/wiki/Standard_RAID_levels
-