DM510: Mass-Storage Structure

Lars Rohwedder

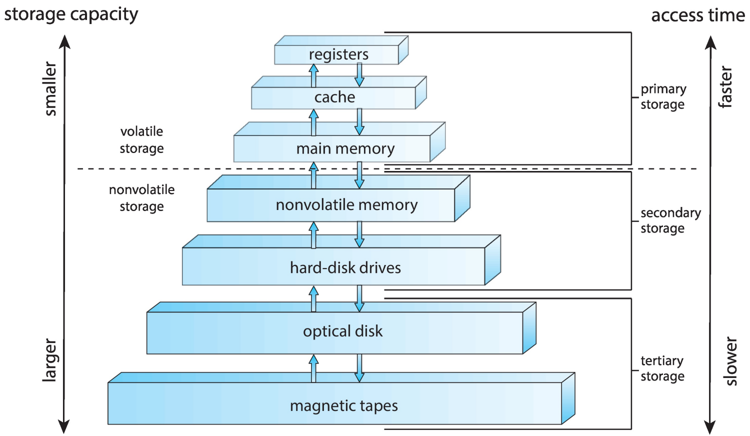
Disclaimer

These slides contain (modified) content and media from the official Operating System Concepts slides: https://www.os-book.com/OS10/slide-dir/index.html

Today's lecture

- Chapter 11 of course book

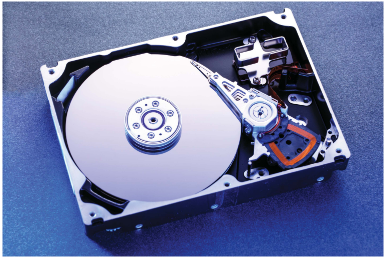
This concerns non-volatile storage:

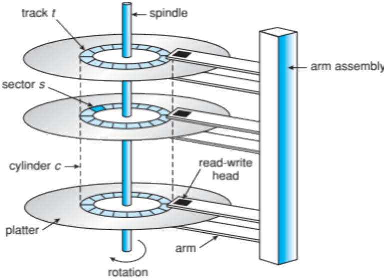


Hardware

Hard disk drives (HDD)

- **Platters** spin under **read-write head**
- Each platter has **tracks**. Each track has sectors. Head can move between **cylinders**, which are tracks of all platters with the same location
- **Transfer rate:** rate at which data is transferred between HDD and main memory
- **Positioning time (random-access time):** time to move head to correct cylinder (**seek time**) and time to move sector under head (**rotational latency**)
- Moving parts prone to permanent damage, e.g. **head crash** (collision with disk)





HDD performance

Typical speeds

Actual numbers vary between products

- **Total storage:** 30 GB $- 3$ TB
- **Transfer rate:** $0.1 - 1$ GB/s
- **Average seek time:** $3 - 9$ ms
- **Average rotational latency:** $2 - 6$ ms $(= 1/2 \cdot 1/\mathrm{RPM})$

$$\text{average access time} = \text{average seek time} + \text{average rotational latency}$$

$$\text{average I/O time} = \text{average access time} + \frac{\text{amount to transfer}}{\text{transfer rate}} + \text{controller overhead}$$

HDD performance

Typical speeds

Actual numbers vary between products

- **Total storage:** 30 GB $-$ 3 TB
- **Transfer rate:** $0.1 - 1$ GB/s
- **Average seek time:** $3 - 9$ ms
- **Average rotational latency:** $2 - 6$ ms $(= 1/2 \cdot 1/\mathrm{RPM})$

$$\text{average access time} = \text{average seek time} + \text{average rotational latency}$$

$$\text{average I/O time} = \text{average access time} + \frac{\text{amount to transfer}}{\text{transfer rate}} + \text{controller overhead}$$

Example

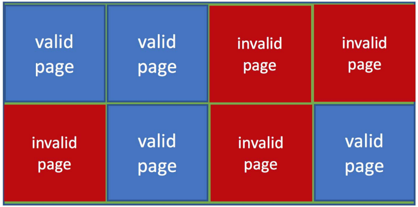
Transfer block of 4 KB, 7200 RPM, 5 ms average seek time, 0.1 GB/s transfer rate, 0.1 ms controller overhead: see blackboard

Nonvolatile memory devices (NVM)

- Includes SSDs (NVM used like HDD), USB drives, storage in mobile devices
- No moving parts $\rightsquigarrow$ more reliable, no seek time or rotational latency
- Compared to HDD: more expensive (per MB), lower capacity, faster random access
- Limited number of write-cycles $\rightsquigarrow$ potentially shorter life span

Writing to NVM

- Cannot overwrite a page (similar to sector) in place, instead data is relocated and old page is invalidated
- Once block (of multiple pages) is mostly invalid, entire block is erased and can be reused
- Number of times a block can be erased is limited ($\approx 100000\times$)
  $\leadsto$ device controller should ensure different blocks are worn out evenly



Tertiary storage

Magnetic tapes

- Tape needs to be wound or rewound past read-write head. Moving to correct position can take minutes
- Similar transfer rates to HDD and large capacities (e.g. > 100 TB)
- Mainly for used for backup nowadays (tertiary storage)

Optical disks

- CD-ROM, CD-RW, DVD
- Sometimes, but not always rewritable
- Suitable for backup (tertiary storage)

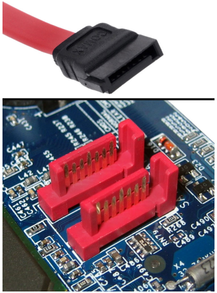
Interfacing with Disks

Storage device I/O

- Disks appear as arrays of logical blocks numbered $1, 2, \dots, n$
- Logical blocks mapped in straight-forward way to physical locations: first block to first track of outermost cylinder, then next blocks follow sequentially until end of track. Afterwards, next track, next cylinder, etc.

  $\rightsquigarrow$ proximity in logical addresss $\approx$ proximity in cylinder, section, etc.

- HDD connected via specialised bus, e.g. SATA
- Due to higher speed, SSDs often connected directly to PCI bus, via NVM express
- Device controller manages disk, writes and reads from main memory via direct memory access (DMA), interfaces with CPU via I/O requests and interrupts

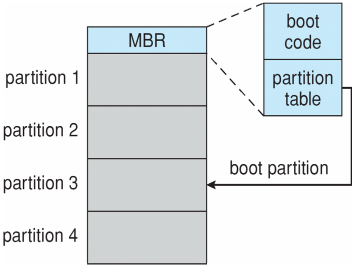


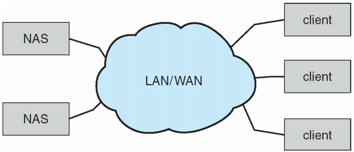
SATA port

Storage device management

Data structure on disk

- Cylinders can be grouped as **partitions** that act as logical disks
- **Formatting** creates a **file system** on partition
- Roles of partitions: boot partition (containing bootloader), swap space (for use in paging), root partition (contains file system with operating system), other mountable files systems, raw partition (e.g. for databases)





- Disks can be **mounted** or **unmounted**, which makes their file system accessible
- Also network/cloud devices can be mounted

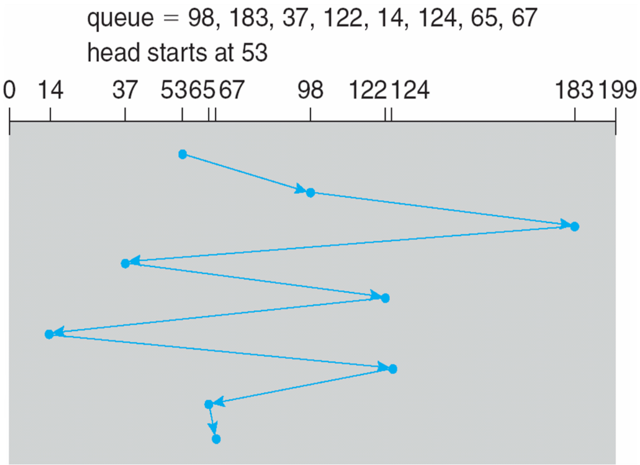
HDD Scheduling

Overview

- Goals: minimize seek time, fairness considerations
- Applicable when disk under heavy load and requests queue up
- Device controller has buffer to maintain queue and implements one of the following algorithms

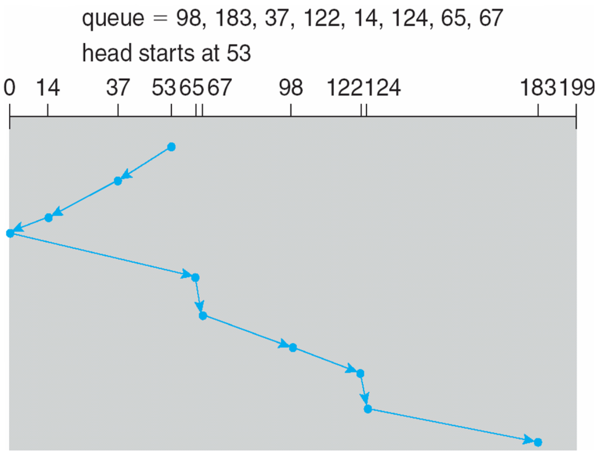
First-come-first-serve (FCFS)

- Requests are served in the order they arrive
- In example below the total movement is 640 cylinders



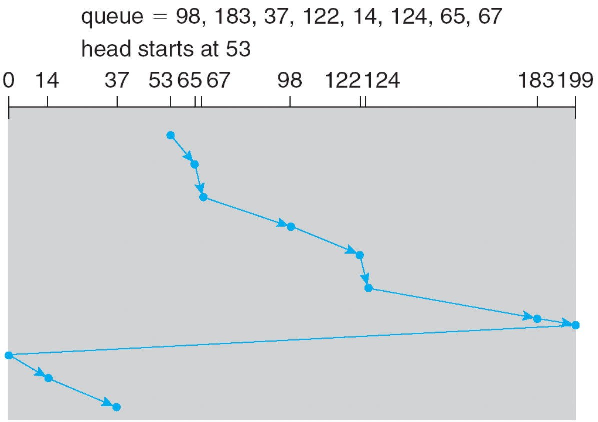
SCAN

- Starting at one end, head moves towards other end and services all request on the way. Once last request is reached, direction is reversed
- In example the total movement is 208 cylinders
- Requests at either end tend to wait longer than those in the middle



C-SCAN

- Like SCAN, but when end is reached, move all the way to the beginning again
- In example the total movement is 383 cylinders
- Serves requests more uniformly than SCAN



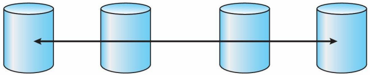
Other scheduling algorithms

- **Shortest-seek-time-first (SSTF)**: service request closest to current head position next
- Several queues for read and write: useful to give priority to reads because they are more likely to lead to blocking
- Several queues against starvation: use "unfair" algorithm on one queue, but move to a FCFS queue if request has not been served for too long

RAID

Redundant array of inexpensive disks (RAID)

- To hedge against data loss, when disks may (permanently) fail
- To increase performance via parallelism



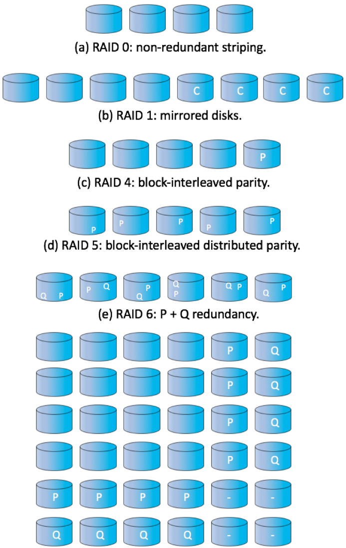
Data striping

- Several physical disks combined to one logical disk (of larger size)
- Example: the ith disk ($i = 1, 2, \ldots, n$) stores logical blocks $i, i+n, i+2n, \ldots$
- Leads to load balancing:

  Large accesses: when we read $k \cdot n$ consecutive blocks, we only need to read k block from each disk $\rightsquigarrow$ up to $n\times$ higher throughput.

  Small accesses: single block accesses are distributed over disks

RAID levels



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID 0

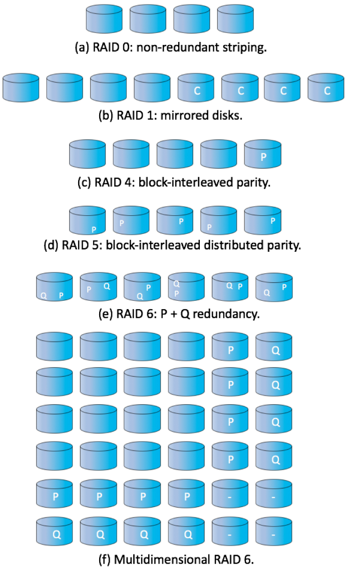
- No redundancy, failure leads to data loss

RAID levels

RAID 0

RAID 1

- Every disk is duplicated
- Can recover from single disk failure
- Requires $2\times$ more disks



RAID levels

RAID 0

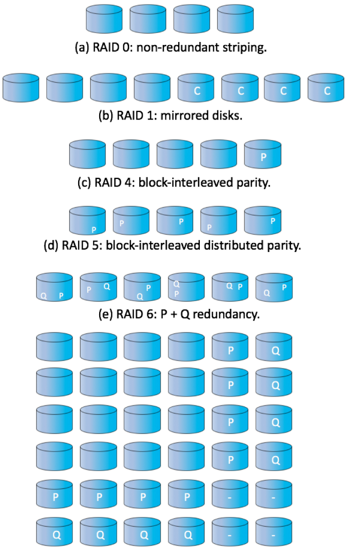
RAID 1

RAID 4

- One extra disk that stores parity:

$$P[i] = D_1[i] + D_2[i] + D_3[i] + \cdots \mod 2$$

  where $D_1, D_2, D_3, \ldots$ are disks, P is additional parity disk, i indexes block of data of each disk
- Can also recover from single disk failure, but requires only one extra disk



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

RAID levels

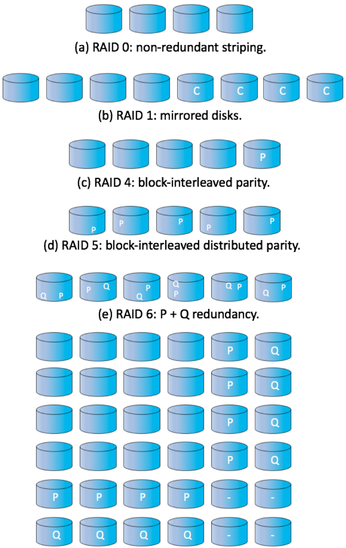
RAID 0

RAID 1

RAID 4

RAID 5

- like RAID 4, but for different blocks, different disks take role of parity
- more balanced accesses than RAID 4, where parity disk is accessed on every write



(a) RAID 0: non-redundant striping.
(b) RAID 1: mirrored disks.
(c) RAID 4: block-interleaved parity.
(d) RAID 5: block-interleaved distributed parity.
(e) RAID 6: P + Q redundancy.
(f) Multidimensional RAID 6.

RAID levels

RAID 0

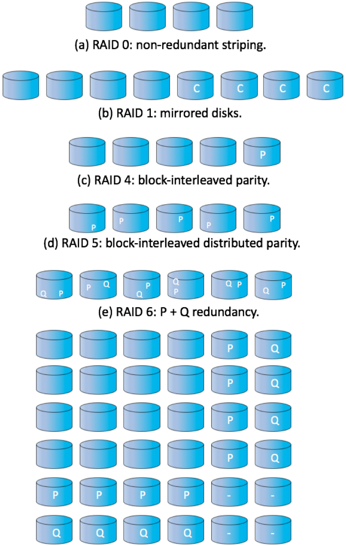
RAID 1

RAID 4

RAID 5

RAID 6

- Additional redundancy to be able to recover from multiple failures
- Details omitted



(a) RAID 0: non-redundant striping.
(b) RAID 1: mirrored disks.
(c) RAID 4: block-interleaved parity.
(d) RAID 5: block-interleaved distributed parity.
(e) RAID 6: P + Q redundancy.
(f) Multidimensional RAID 6.

RAID levels

RAID 0

RAID 1

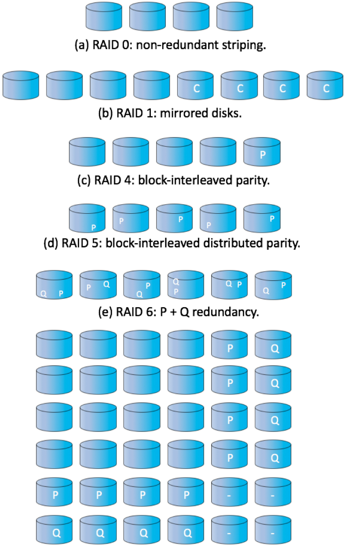
RAID 4

RAID 5

RAID 6

Multidimensional RAID 6

- Disks (virtually) aligned in matrix, redundancy on each column and row
- Few additional disks, high failure tolerance



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 4: block-interleaved parity.

(d) RAID 5: block-interleaved distributed parity.

(e) RAID 6: P + Q redundancy.

(f) Multidimensional RAID 6.

Mean time to data loss

How likely is data loss? (in expected number of years until data loss)

Assumptions

- For each disk we are given **mean time to failure**, i.e., how much time passes in expectation until disk fails
- We are also given the **mean time to repair:** time until a broken disk is replaced
- Failure event of each disk is independent

Example

- A disk is mirrored with RAID 1.
- Mean time to failure of each disk is $100000h \approx 11.4$ years
- Mean time to repair is 10 hours
- Then mean time to data loss $= \frac{100000^2}{2 \cdot 10} h = 500 \cdot 10^6 h \approx 57000$ years