Chapter 12: Mass-Storage Systems

Prof. Li-Pin Chang

CS@NYCU

Chapter 12: Mass-Storage Systems

- Magnetic Tape and Disk
- Disk Scheduling
- RAID Structure
- Solid State Disks

Objectives

- Describe the physical structure of secondary and tertiary storage devices and the resulting effects on the uses of the devices
- Explain the performance characteristics of mass-storage devices and performance optimization techniques
- Discuss operating-system services provided for mass storage, such as RAID
- Introduction to SSD

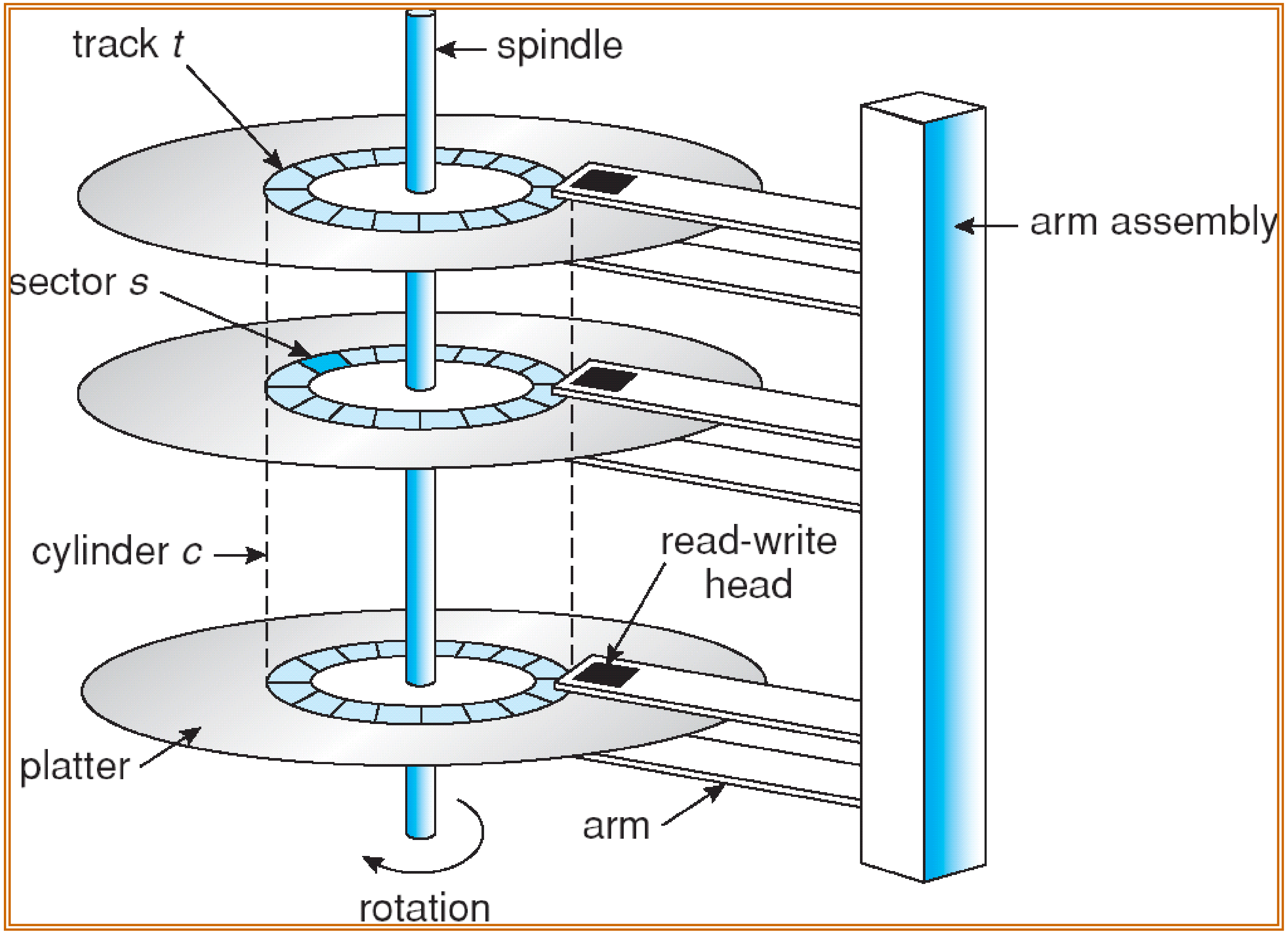
Magnetic Tape

- Very fast sequential read-write, achieving 400 MB/s (LTO-9); extremely slow on random access
  - Kept in spool and wound or rewound past read-write head
  - Sequential transfer rate comparable to disk
  - Very high capacity, 18 TB per cassette (LTO-9)
- Near-permanent storage for data backup

Magnetic Disks

- Provide bulk of secondary storage of modern computers
- Transfer rate is rate at which data flow between drive and computer
  - SATA3: 600 MB/s; typical HDD~100MB/s
- Positioning time (random-access time) is time to move disk arm to desired cylinder (seek time) and time for desired sector to rotate under the disk head (rotational latency)
  - Typically 5400 rpm (laptop) or 7200 rpm (desktop)
  - Typically 5ms~7ms average seek time
- Head crash results from disk head making contact with the disk surface, causing physical damage

Moving-Head Disk Mechanism



Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer.
- 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.

Disk Attachment

- ATA/IDE interface
  - the primary disk interface for personal computers
  - Parallel ATA (PATA) and Serial ATA (SATA)
- SCSI bus
  - Up to 16 devices (disks, printers, etc) on one cable
- FC (Fiber Channel) is high-speed serial architecture
  - The basis of Storage Area Networks (SANs) in which many hosts attach to many storage units
  - Infiniband, 10km per hop

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.
- Access time has two major components
  - Seek time is the time for the disk are to move the heads to the cylinder containing the desired sector.
  - Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Disk scheduling optimizes seek time, which is proportional to the total seek distance

The Need for Disk Scheduling

- Because of
  - 1) multiprogramming and
  - 2) write buffering,

  there are multiple pending disk requests

- How to select the next request to serve?
  - has impacts on response and throughput

Disk Scheduling (Cont.)

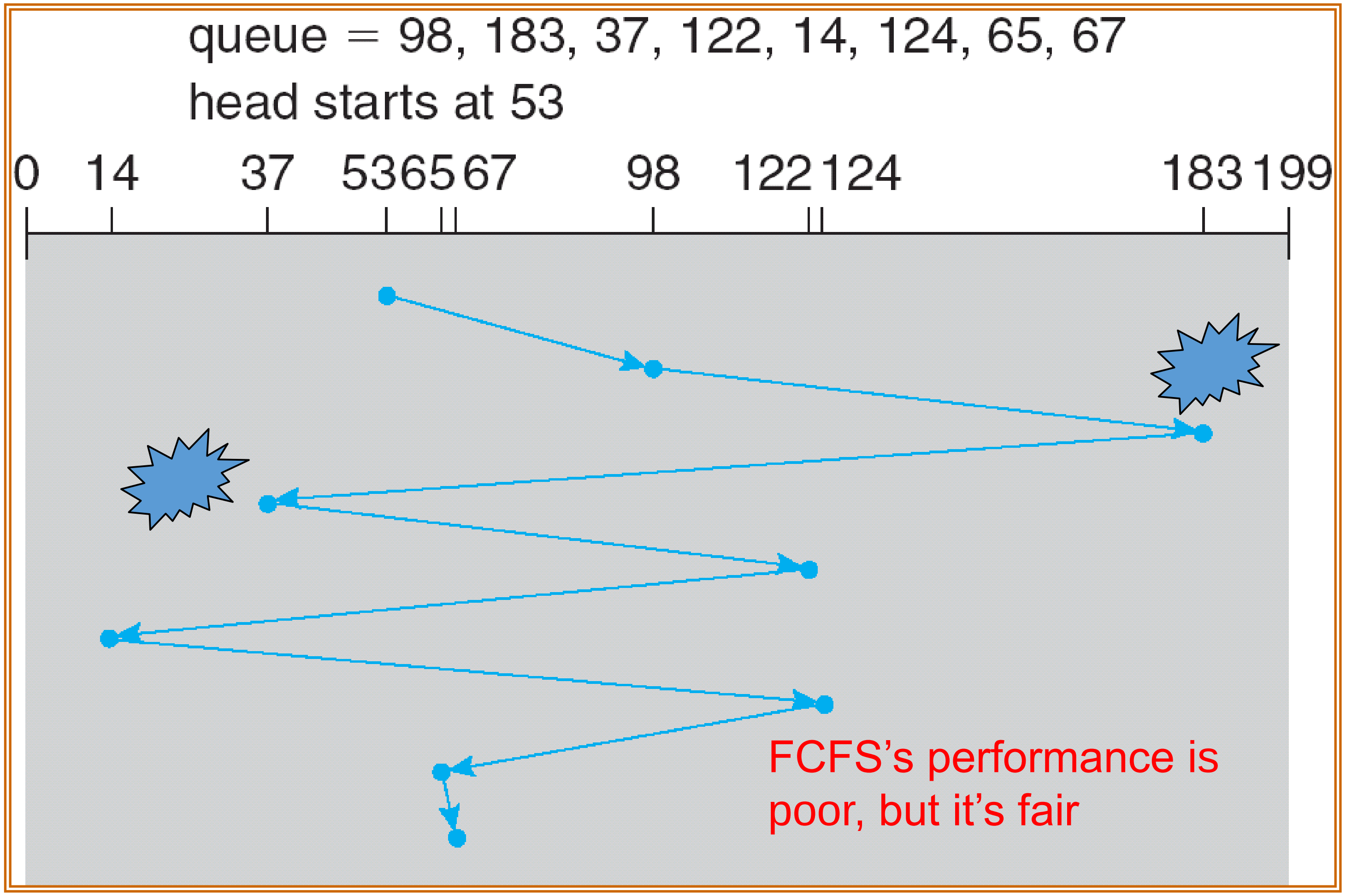
- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

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

- Head pointer 53

FCFS Disk Scheduling

Illustration shows total head movement of 640 cylinders.

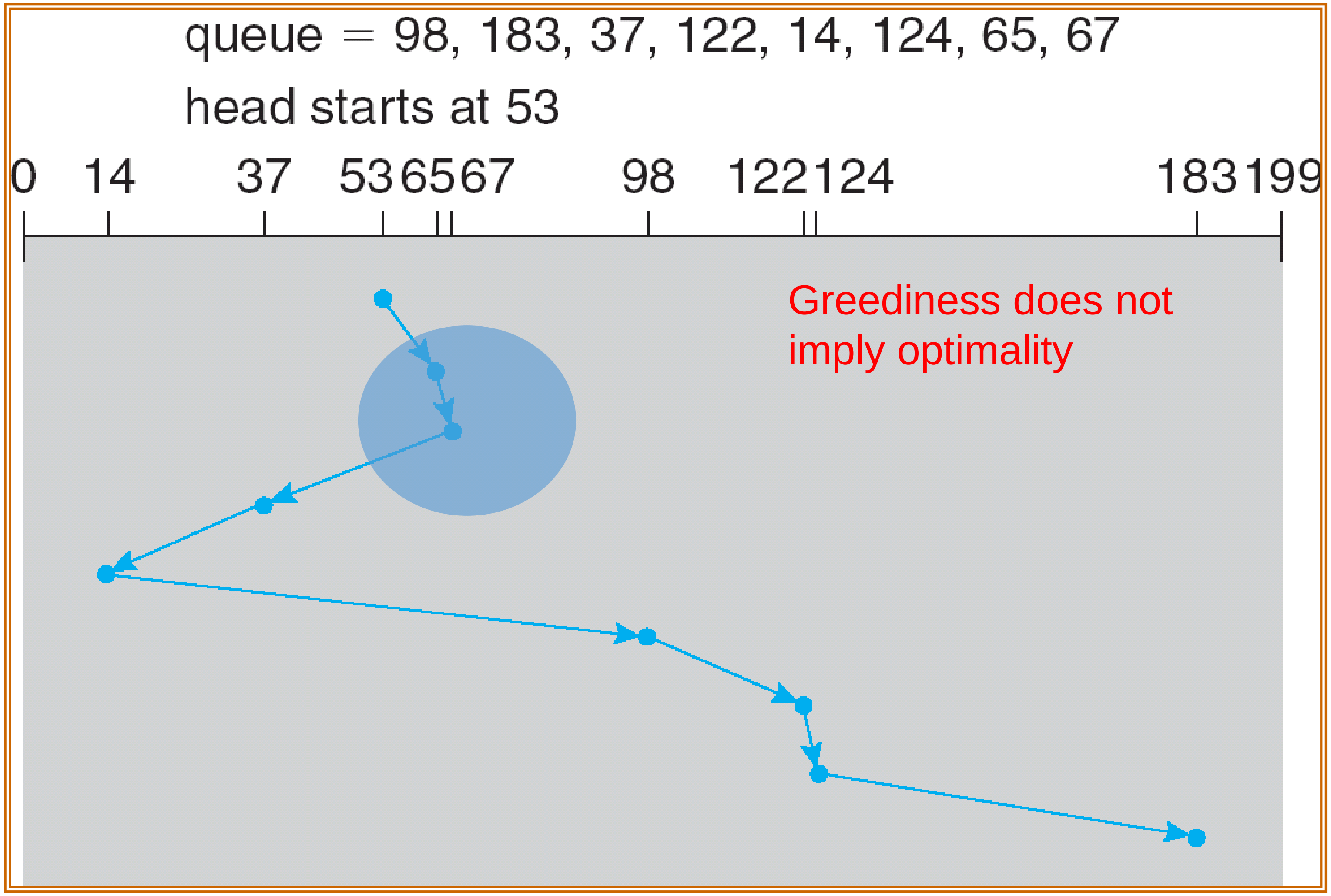


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199
FCFS's performance is poor, but it's fair

SSTF Scheduling

- 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
  - How to avoid starvation?
- Illustration shows total head movement of 208 cylinders

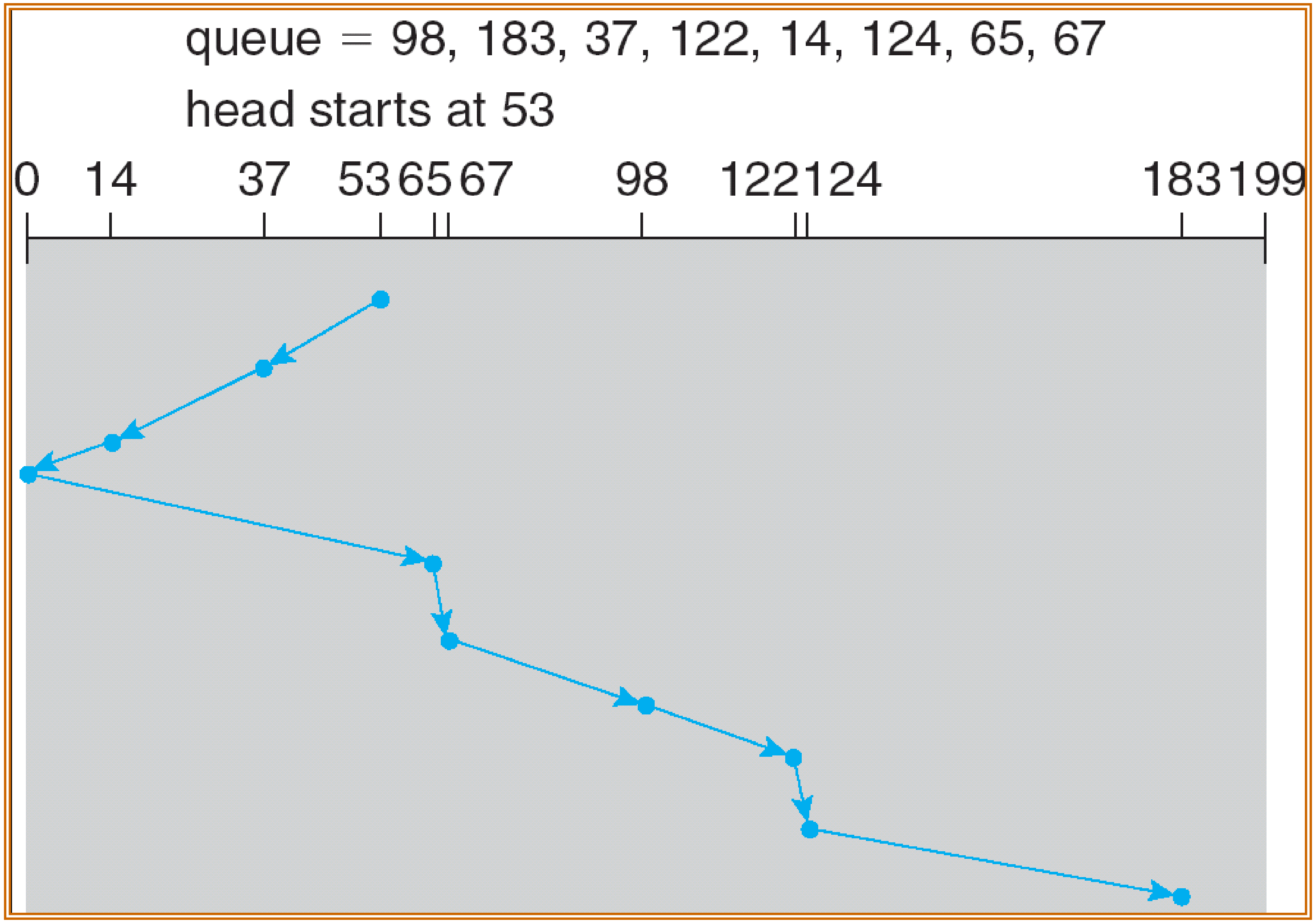
SSTF Disk Scheduling


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199
Greediness does not imply optimality

SCAN Scheduling

- 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
- Sometimes it is called the elevator algorithm
- Illustration shows total head movement of 236 cylinders

SCAN Disk Scheduling



SCAN

- Much less prone to starvation
- Waiting time of at different cylinders
    - At the outermost or the innermost cylinder:

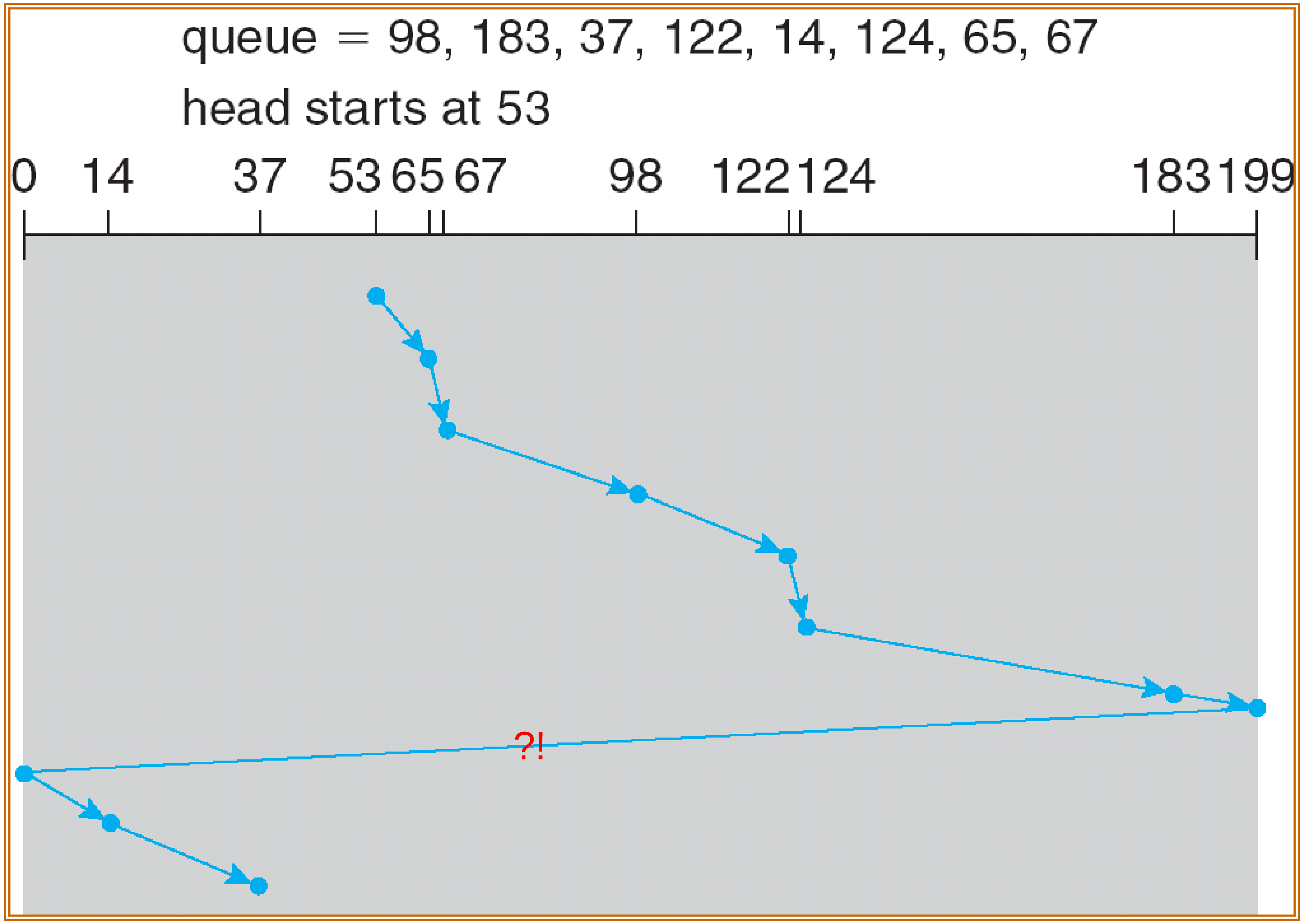
      Max 2 full strokes and 1 reverse
    - At the middle of the disk:

      Max 2 half full strokes and 1 reverse

C-SCAN Scheduling

- 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

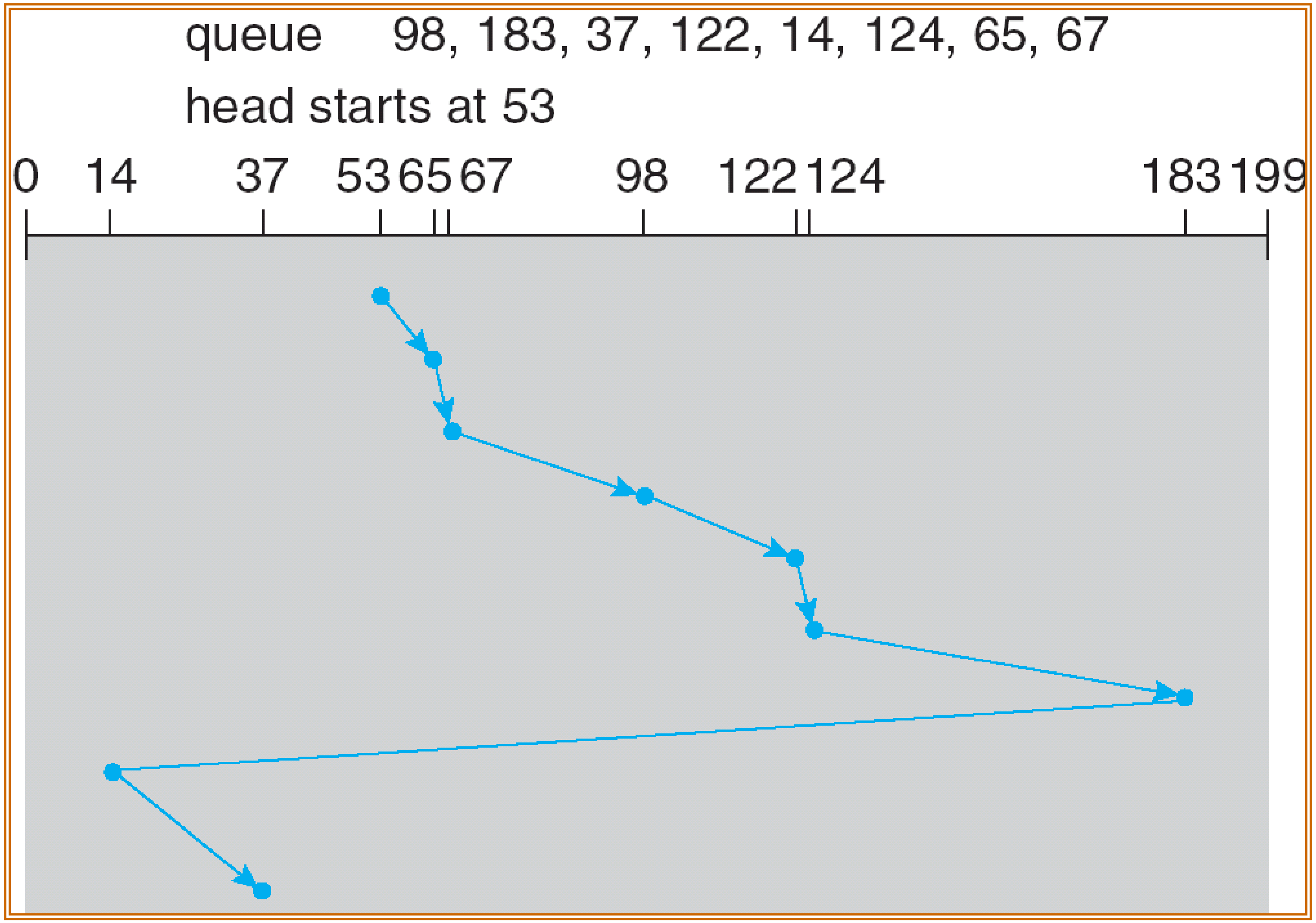
C-SCAN Disk Scheduling



C-LOOK

- Variation 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.

C-LOOK DISK Scheduling



Selection of a Disk-Scheduling Algorithm

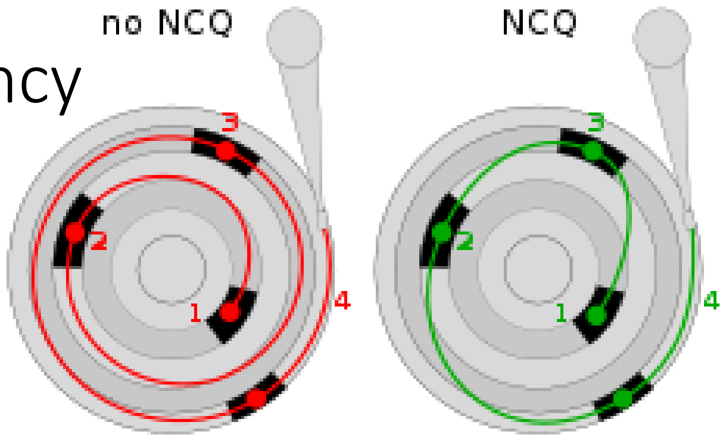
- SSTF has a natural appeal, but it risks starvation
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk; either one is a reasonable default choice
- Requests for disk service can be influenced by the file-allocation method

Disk Seek Optimization

- Disk scheduling problem is NP-hard
  - All the methods mentioned above are not optimal (in terms of the total seek distance)
- Hard to optimize rotational delay from operating systems
  - The HDD firmware knows the current rotation angle better
  - Delegating disk scheduling to the HDD firmware
  - New HDDs accepts a number of pending requests and then reorder them internally
  - E,g., SATA NCQ (Native Command Queuing)

Native Command Queuing (NCQ)

- The hard drive (or any SATA storage device like SSD) accepts multiple outstanding requests and manages them using an internal queue
- The hard drive decides which one to be serviced
- Pros: Taking internal states into consideration, e.g., the angular information
- Cons: Increasing the request latency



https://en.wikipedia.org/wiki/Native_Command_Queuing

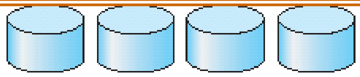
Linux CFQ Scheduler

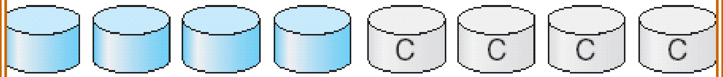
- Implement the concept of time slicing, i.e., each process receives a time share for I/O processing
  - Only for synchronous I/Os
- If no more synchronous I/Os, asynchronous I/Os (in a global queue) are serviced using SCAN

RAID Structure

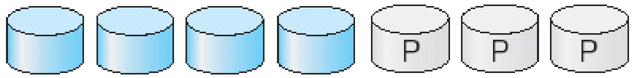
- RAID – Redundant Array of Inexpensive Disks
  - Performance improvement through parallelism
  - Reliability improvement through redundancy
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data.
  - Mirroring or shadowing keeps duplicate of each disk.
  - Block interleaved parity uses much less redundancy.

RAID Levels

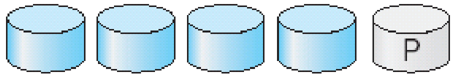

(a) RAID 0: non-redundant striping.



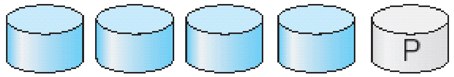
(b) RAID 1: mirrored disks.



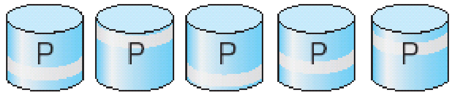
(c) RAID 2: memory-style error-correcting codes.



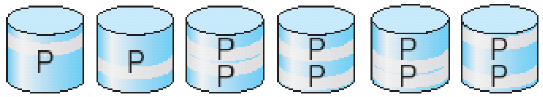
(d) RAID 3: bit-interleaved parity.



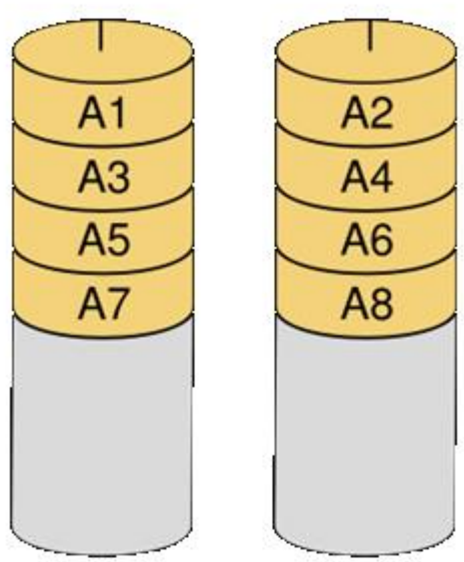
(e) RAID 4: block-interleaved parity.



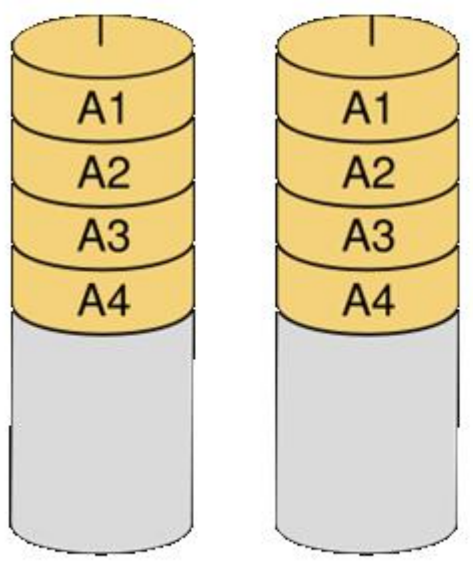
(f) RAID 5: block-interleaved distributed parity.



(g) RAID 6: P + Q redundancy.



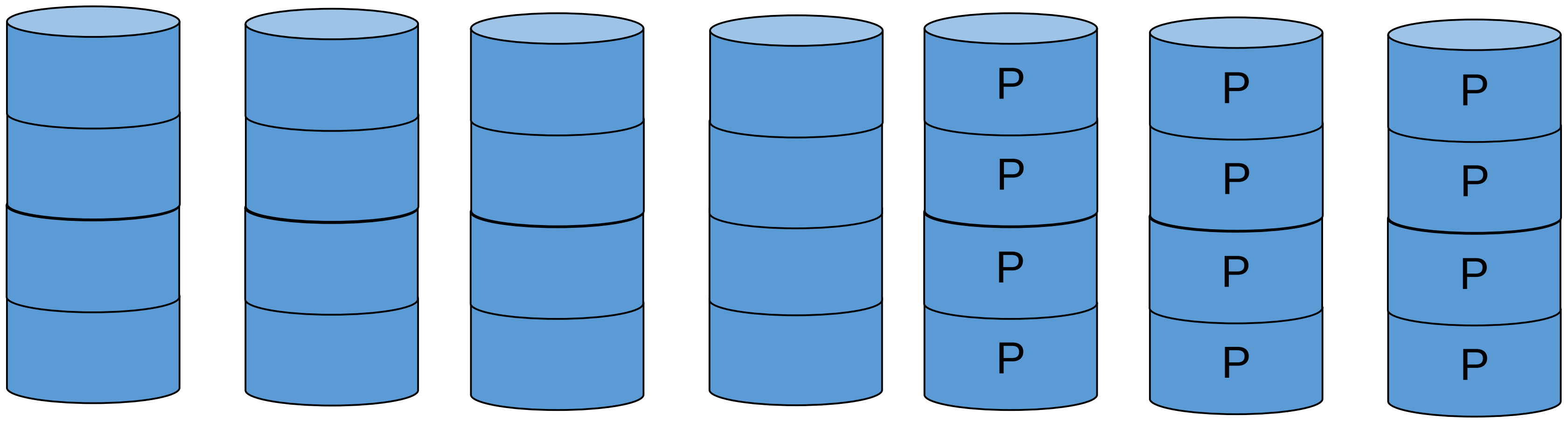
RAID0



RAID1

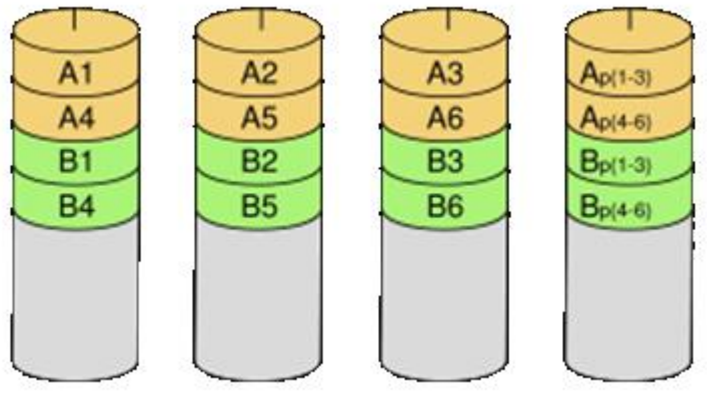
Striping. Aiming at parallelism

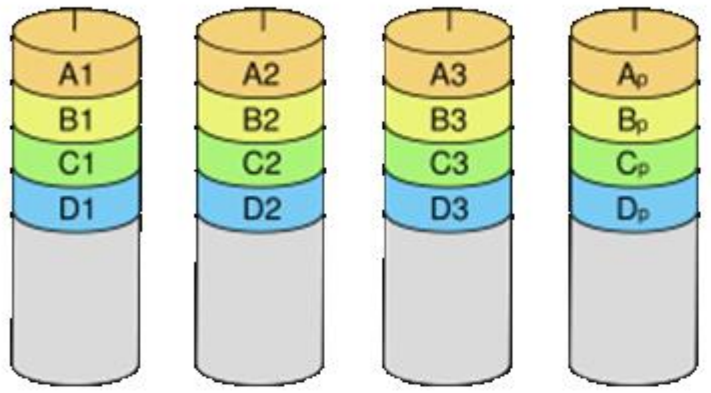
Mirroring, 100% redundancy



RAID-2: memory-style ECC, such as Hamming code

# of parity disks = log2(# of data disks)





RAID 3

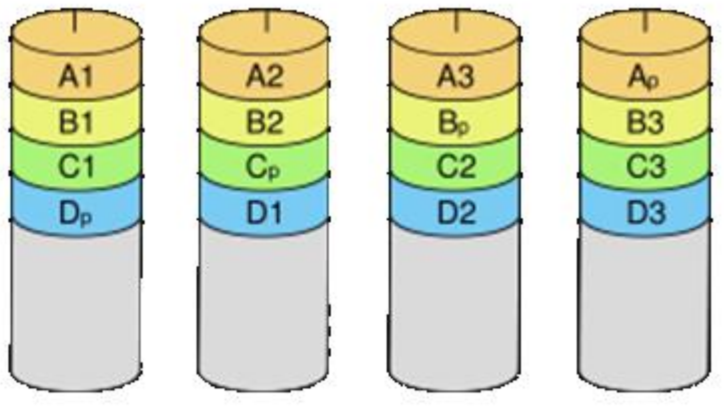
Bit-interleaved
(or sub-block-interleaved)

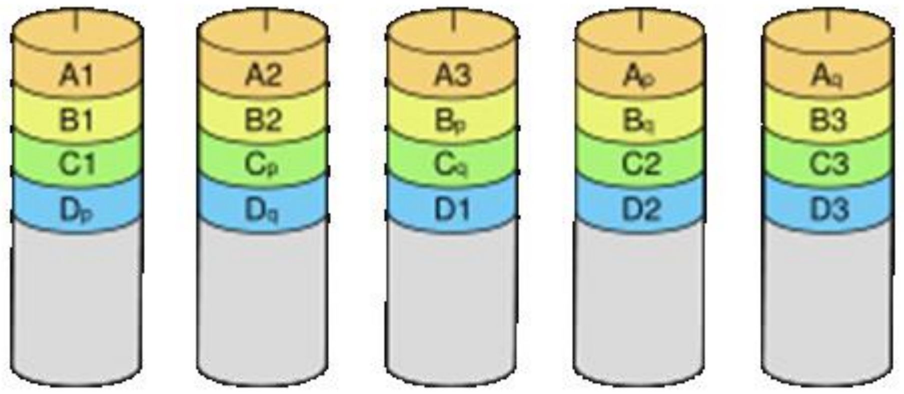
Fully interleaved, one R/W
involves all disks

RAID4

Block-interleaved

One R involves one disk
One W involves two disks
Parity disk → bottleneck





RAID 5

One R involves one disk
One W involves two disks
Parity is spread over all disks

Simple XOR-based parity

RAID 6

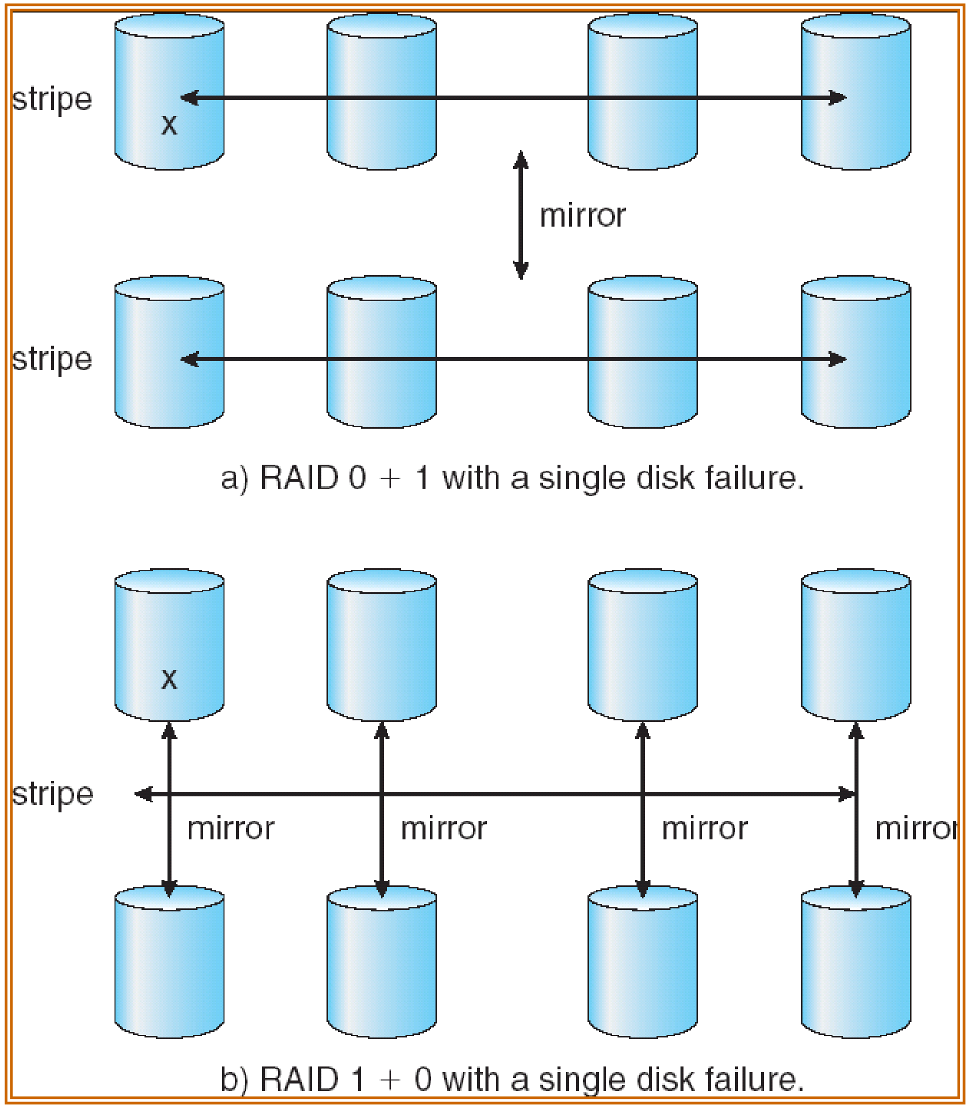
Choosing 4 blocks out of {1,2,3,4,p,q} sufficiently reconstruct {1,2,3,4}

Reed-Solomon code
EVENODD parity

RAID-5 Reliability

- Let the probability of 1-year up of a disk be *p*
- The probability of 1-year up of a RAID-0 of 4 disks:
  - p^4
  - ~0.96 if p=0.99
- The probability of 1-year up of a RAID-5 of 5 disks:
  - $p^5+(5,1)(1-p)*p^4$
  - ~0.999 if p=0.99

RAID 0 + 1 and 1 + 0

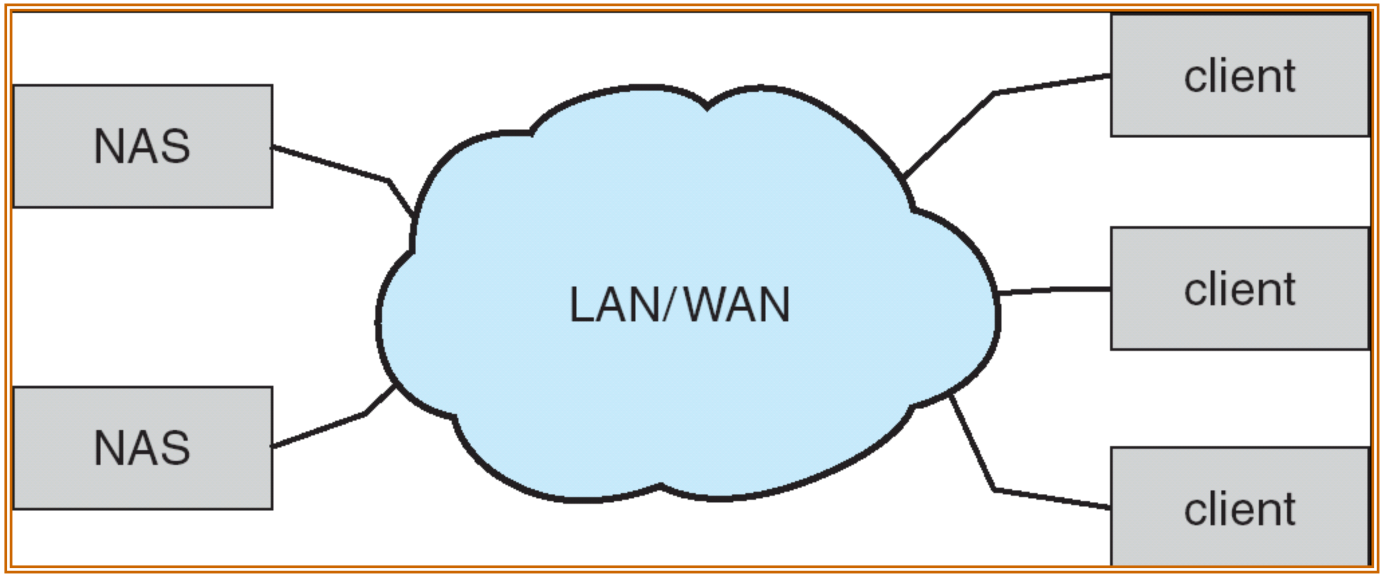


a) RAID 0 + 1 with a single disk failure.

b) RAID 1 + 0 with a single disk failure.

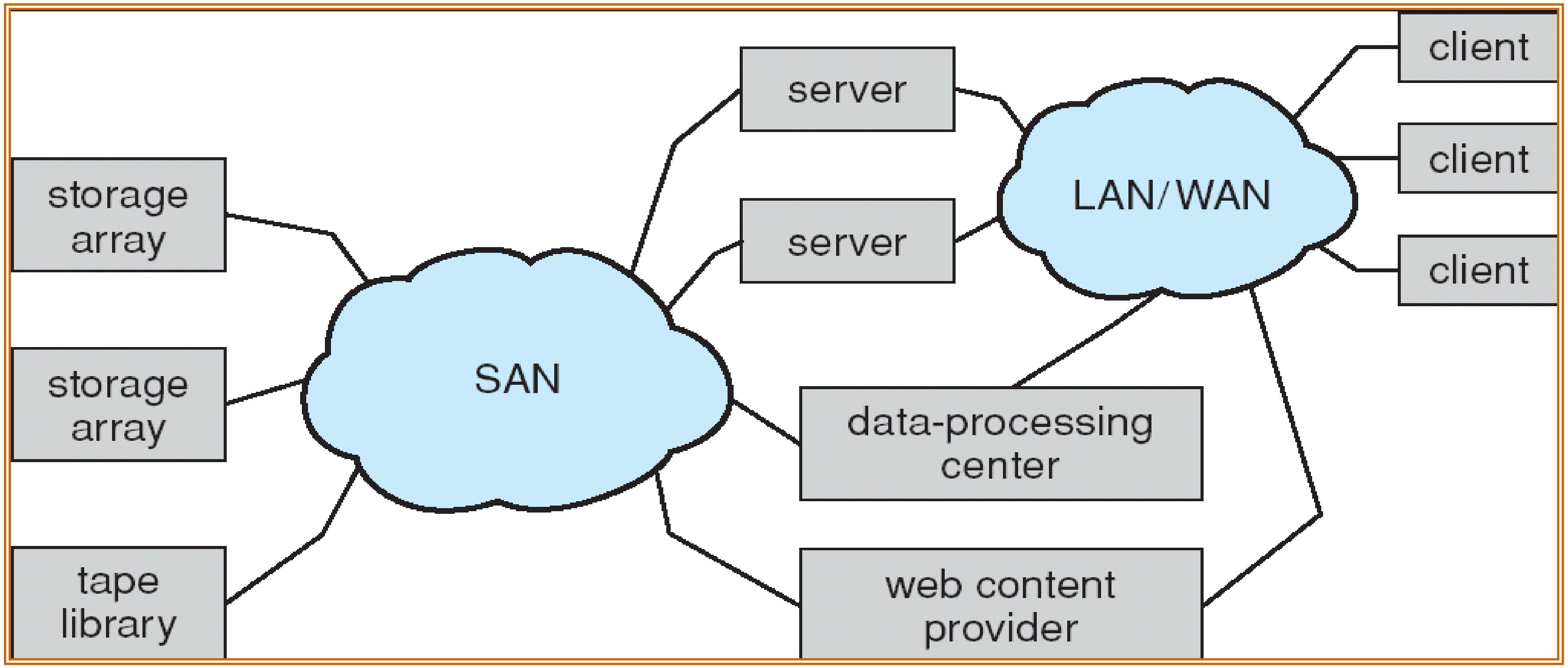
← Better survivability

Network-Attached Storage

- Network-attached storage (NAS) is storage made available over a network rather than over a local connection (such as a bus)
- NFS, CIFS, SAMBA are common protocols
- Implemented via remote procedure calls (RPCs) between host and storage
- New iSCSI protocol uses IP network to carry the SCSI protocol
- 

Storage-Area Network

- Common in large storage environments (and becoming more common)
- Multiple hosts attached to multiple storage arrays – flexible



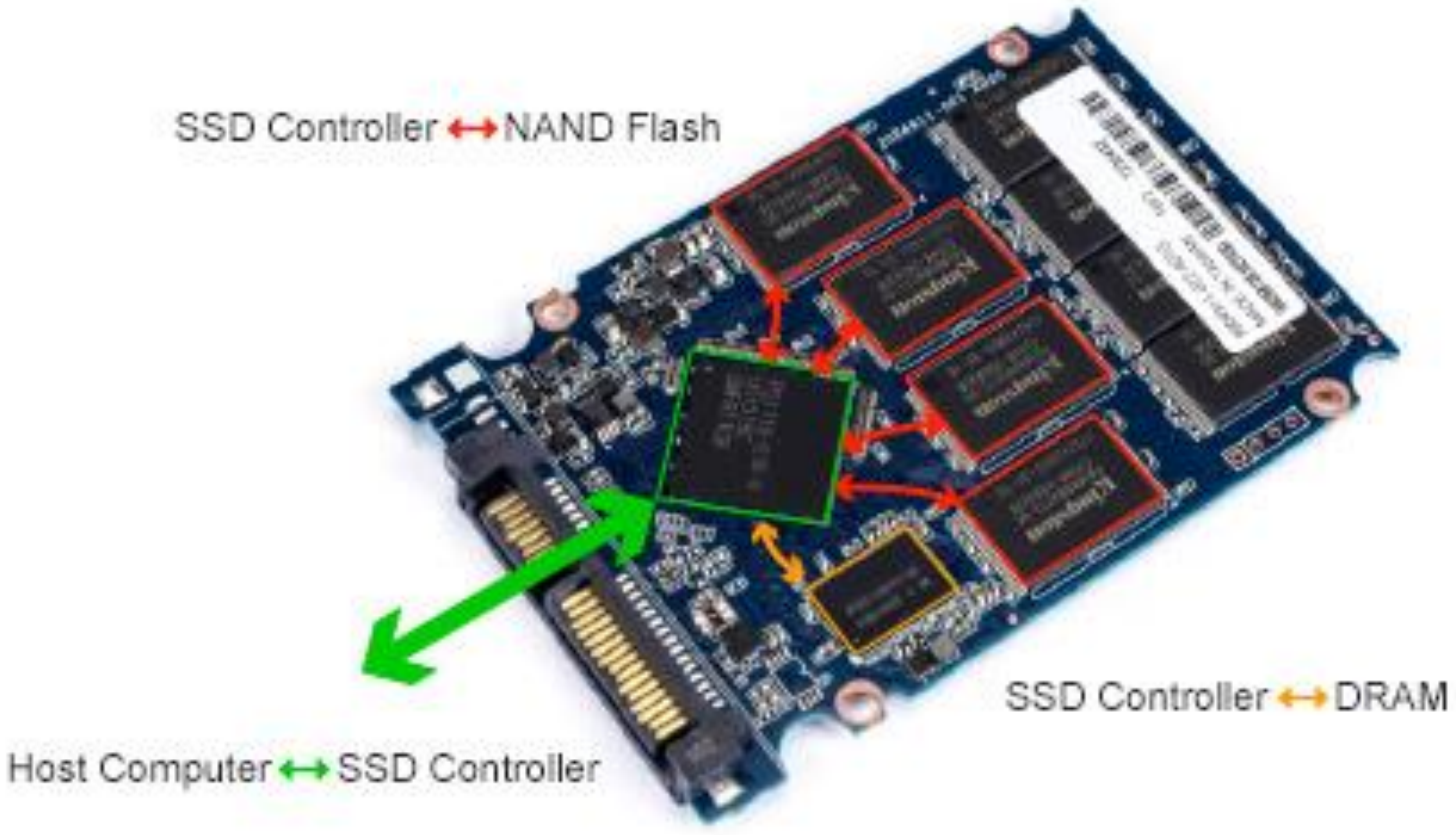
SAN vs. NAS

- SAN is a network dedicated for storage
  - Performance is the primary concern
  - Topology, bandwidth, cost…
- Storage resource in SAN is hidden from the client of SAN.
  - A volume may sit across many storage devices
- NAS may operate over legacy network
  - Interoperability is much more important

Solid-State Disks (SSDs)

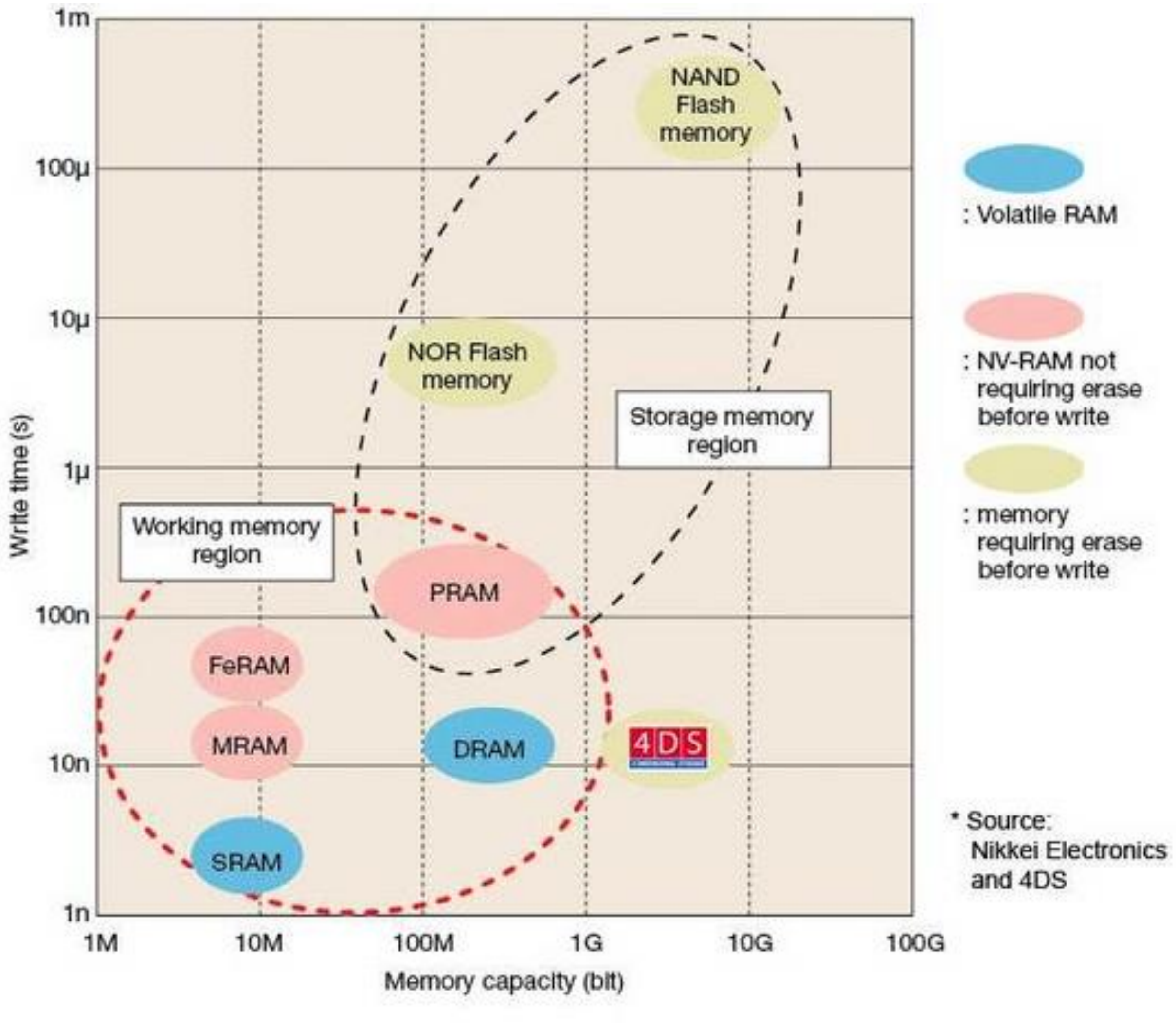
- Storage devices that emulate standard block devices using non-volatile memory
  - Flash memory or battery-backed RAM
  - The OS use the legacy I/O stack on top of SSDs
- Commodity Products
  - SD cards, USB drives, mobile storage (eMMC or UFS), SSDs (SATA or NVMe)
- Performance
  - RAM SSD > flash SSD >> HDD
- Applications
  - Cloud storage: tier storage, cache SSDs
  - Personal computer: HDD replacement, system drive
  - Embedded storage: Smartphones, tablets, laptops, wearables
  - Portable storage: SD cards, USB drives

SSD Internals



https://www.kingston.com/tw/ssd/data-protection

Non-Volatile Memory



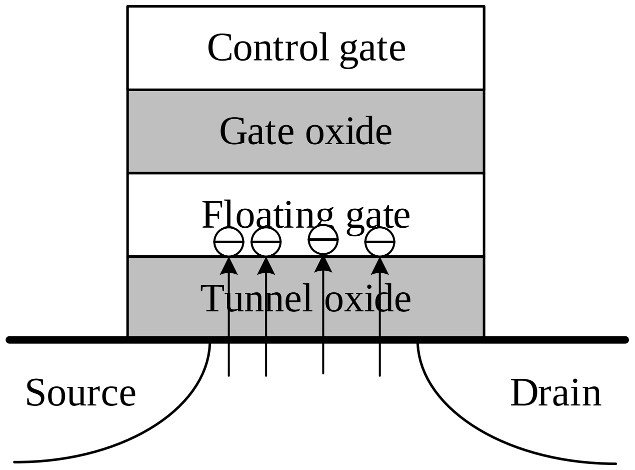
Faster write

Smaller capacity

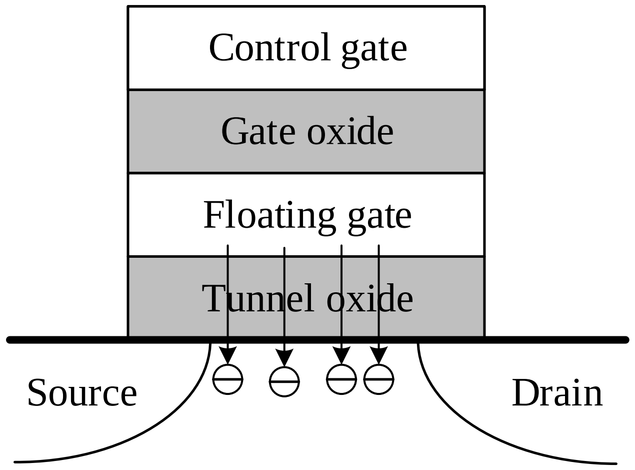
https://www.storagenewsletter.com/2009/11/10/4ds-start-up-profile/

Floating Gate (FG) Flash Cell Structure

- Program (injection) & erase (discharge)
- P/E cycle limit

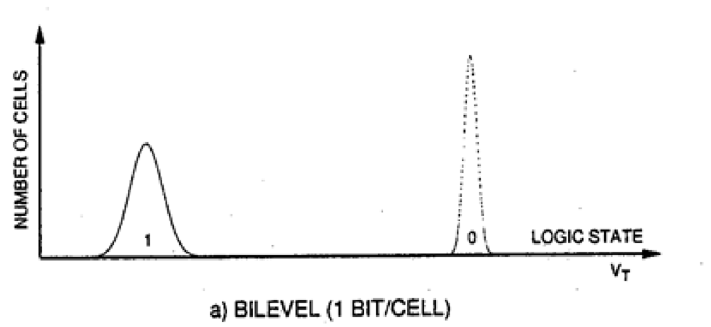


Program (write)

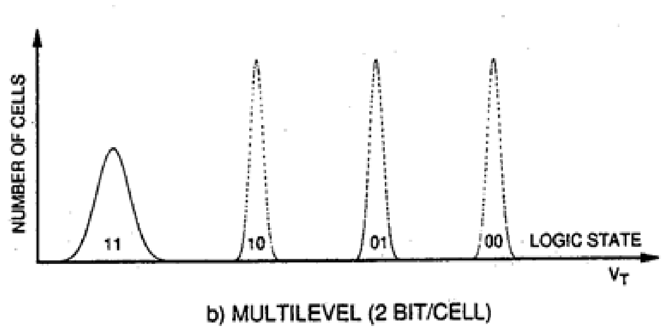


Erase

Logical Bits in Cells



Single-level cell



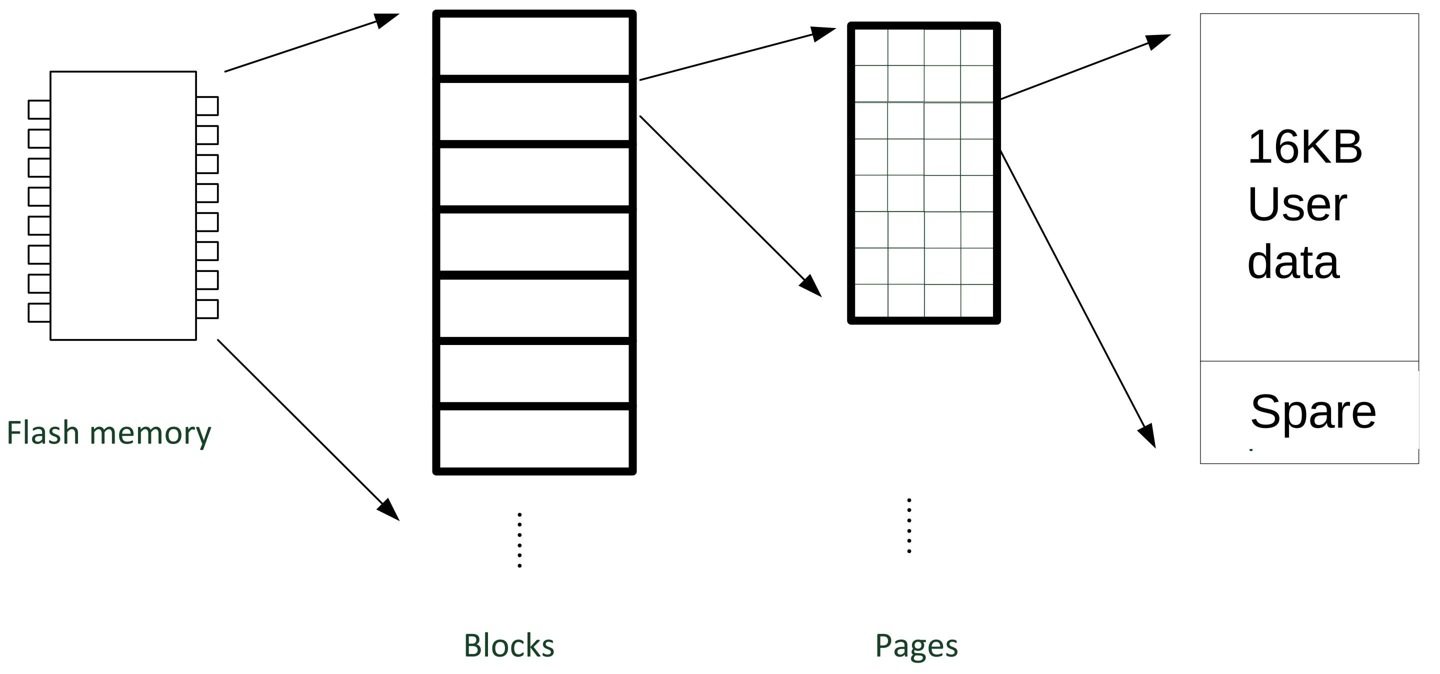
Multi-level cell

- SLC (1 bit) → MLC (2 bits) → TLC (3 bits) → QLC (4)

	Bit per cell	Write latency	P/E cycles	Cost
SLC	1	1	100K	1
MLC	2	1x~2x	3K	1/2 ~ 1/3

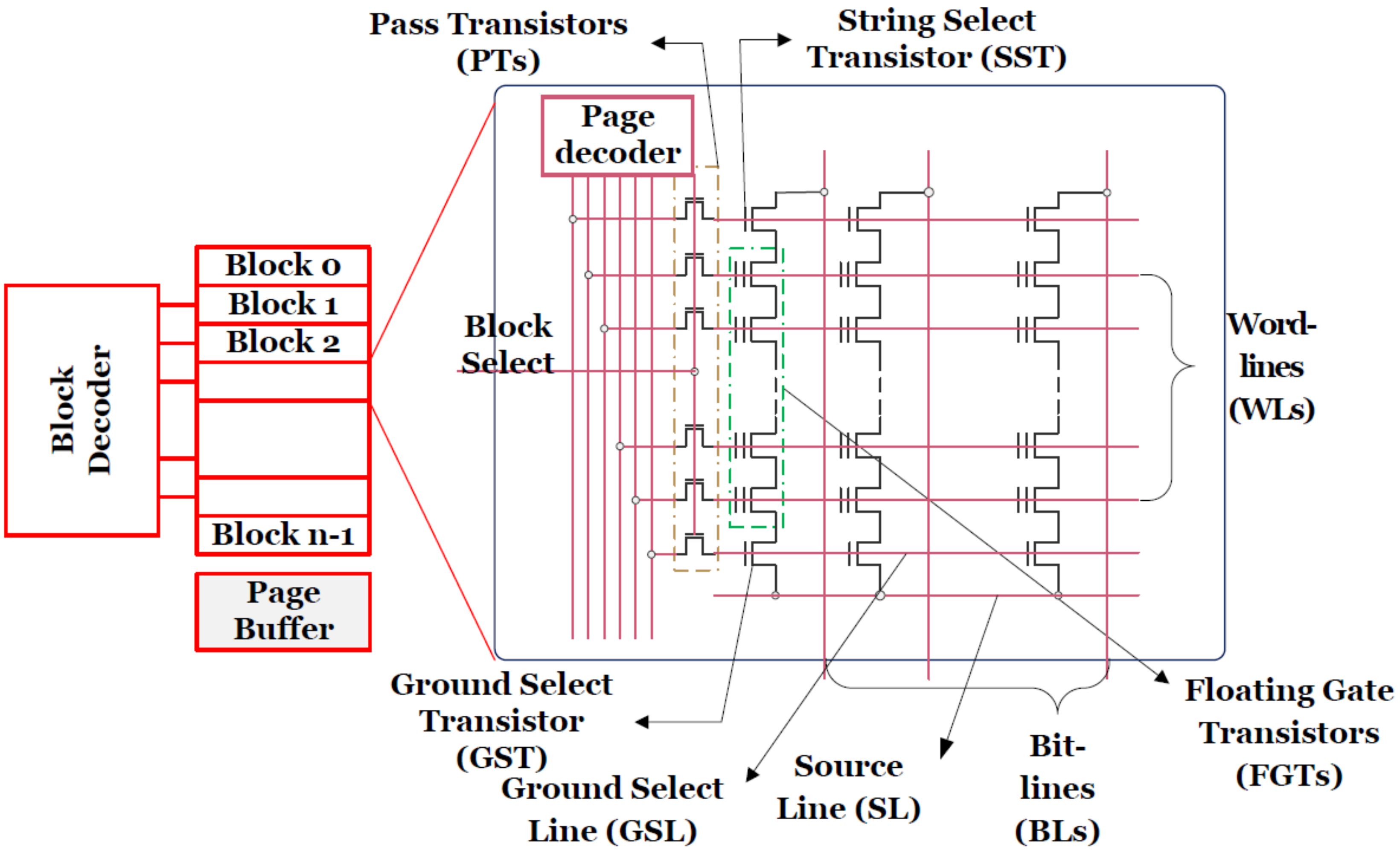
- TLC and QLC are under mass production

NAND Flash Geometry



- Units of operation
    - Read/write → page (16KB) ; Erase → blocks (2MB)

Circuitry of NAND Flash

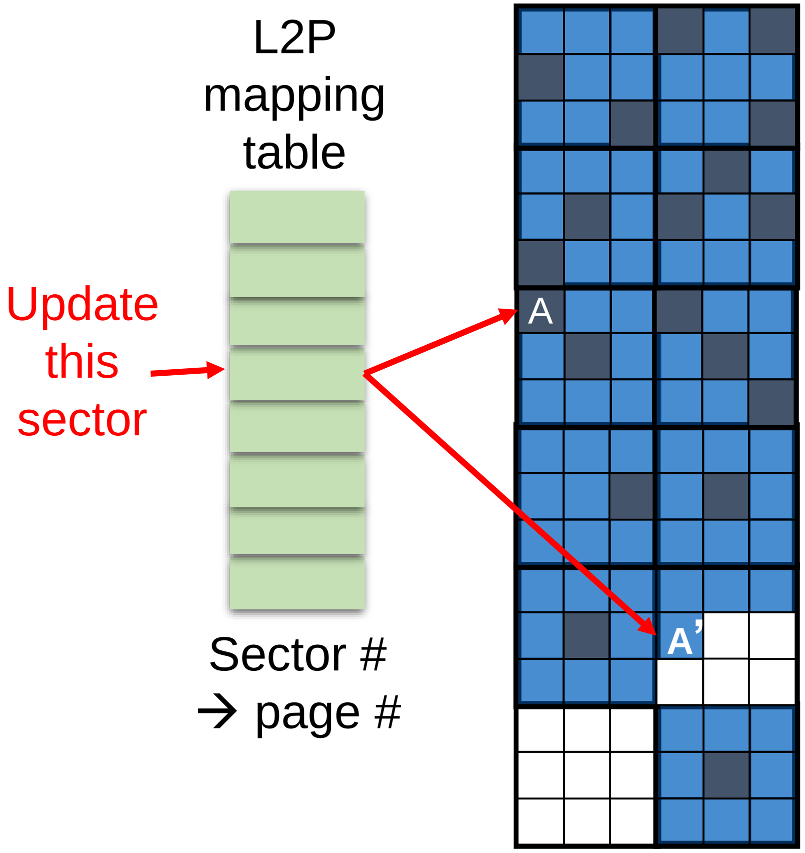


Flash Translation Layer (FTL)

- The firmware of SSDs
  - Hiding flash memory physics from the host
- Provide block device emulation to the host
- Flash memory management tasks
  - Logical-to-physical address translation
  - Garbage collection
  - Wear leveling

Logical-to-Physical Address Translation

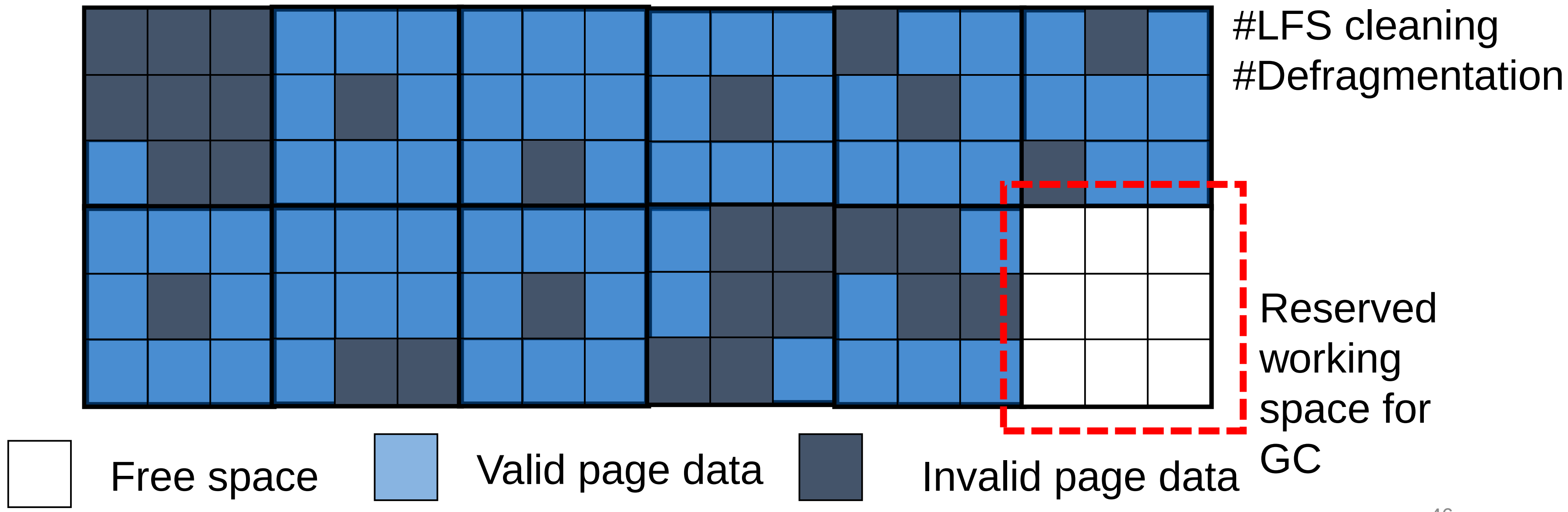
- Pages cannot be overwritten unless being erased
- Erase a block to update a page
  - Too inefficient
- Out-of-place update; mark old data (page) invalid
- Need logical-to-physical address translation
  - logical sector # → physical page #



#Page table
#Out-of-place update (LFS)

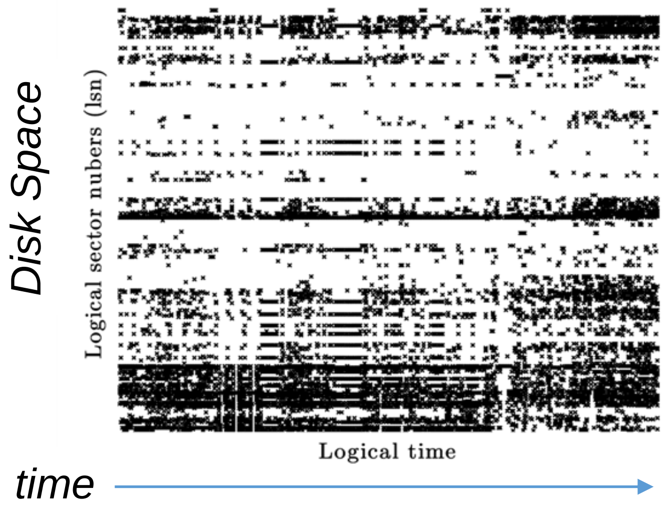
Garbage Collection

- Recycle memory space occupied by invalid data through block erasure
  - Copying valid data (overhead) and erasing block
- Victim selection: Minimize the page-copy overhead

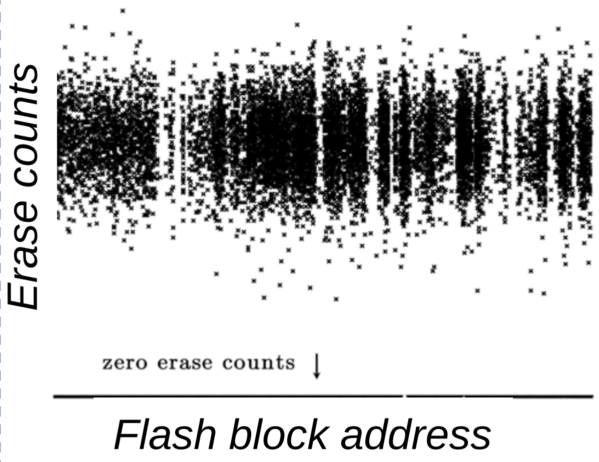


Wear Leveling

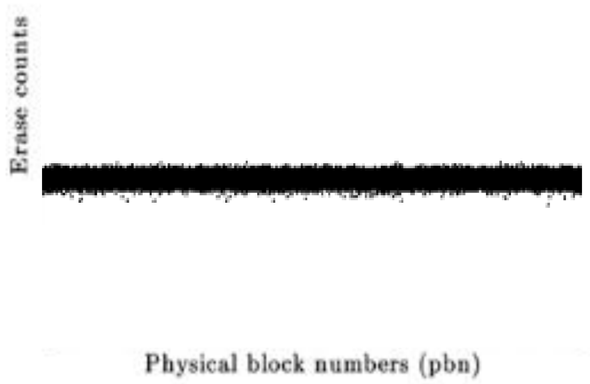
- Typically a (MLC) block endures 3000 cycles of program-erase operations (P/E cycles)
- Locality of write creates imbalance in block erase counts
- Delay the first block retirement by migrating cold data



Disk write pattern



Block ECs without WL



Block ECs with WL

Li-Pin Chang, Tung-Yang Chou, and Li-Chun Huang, "An Adaptive, Low-Cost Wear-Leveling Algorithm for Multichannel Solid-State Disks," ACM Transactions on Embedded Computing Systems, Volume 13, Issue 3, 2013.

End of Chapter 12

Review Questions

1. What are rotational latency and seek time? Why kernel disk schedulers optimize seek overhead but not rotational latency?
2. Redo the examples for disk scheduling using FCFS, SSTF, and SCAN
3. Linux disk schedulers handles synchronous I/Os and then asynchronous I/Os. Discuss the reason why.
4. FCFS performs poorly for hard drives, but it is preferable for SSDs. Why?
5. How does a RAID-5 disk array recover data when a single hard disk fails?
6. Discuss how and why out-of-place updating is used by Flash Translation Layer and Log-Structured File Systems
7. Briefly survey anticipatory scheduling for hard drives
8. Compare following a) SSD L2P mapping vs. virtual memory paging b) flash garbage collection vs. LFS cleaning