Chapter 12: Mass-Storage Systems

Prof. Li-Pin Chang

National Chiao Tung University

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
- Discuss operating-system services provided for mass storage, such as RAID

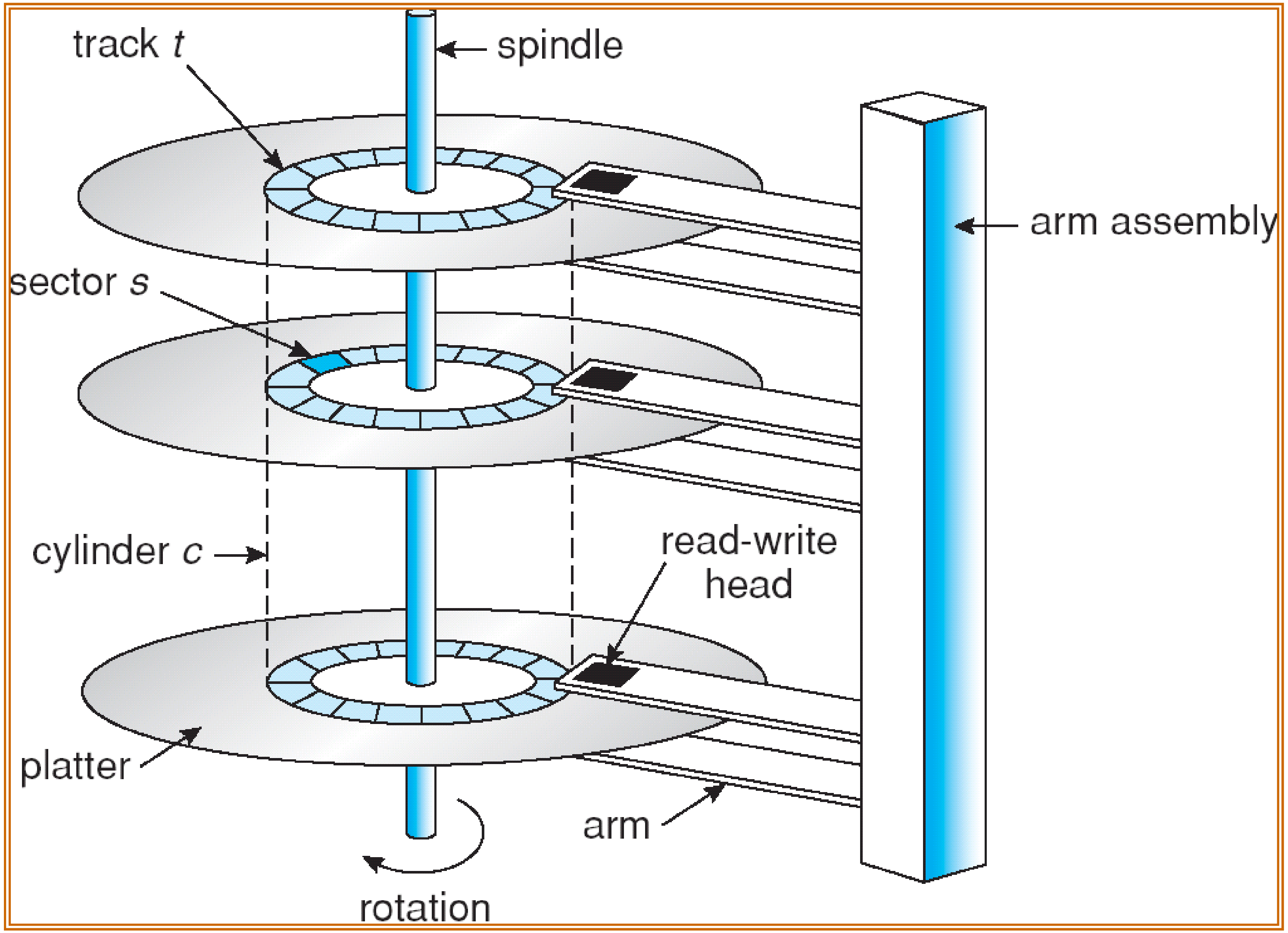
Magnetic Tape

- 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
  - Once data under head, transfer rates comparable to disk
- Relatively permanent and holds large quantities of data
  - 18 TB per cassette (LTO-9)
  - 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
- 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

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.
- Minimize seek time
- Seek time is proportional to 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.

The Need for Disk Scheduling

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

  there might be a number of pending disk requests

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

Disk Scheduling (Cont.)

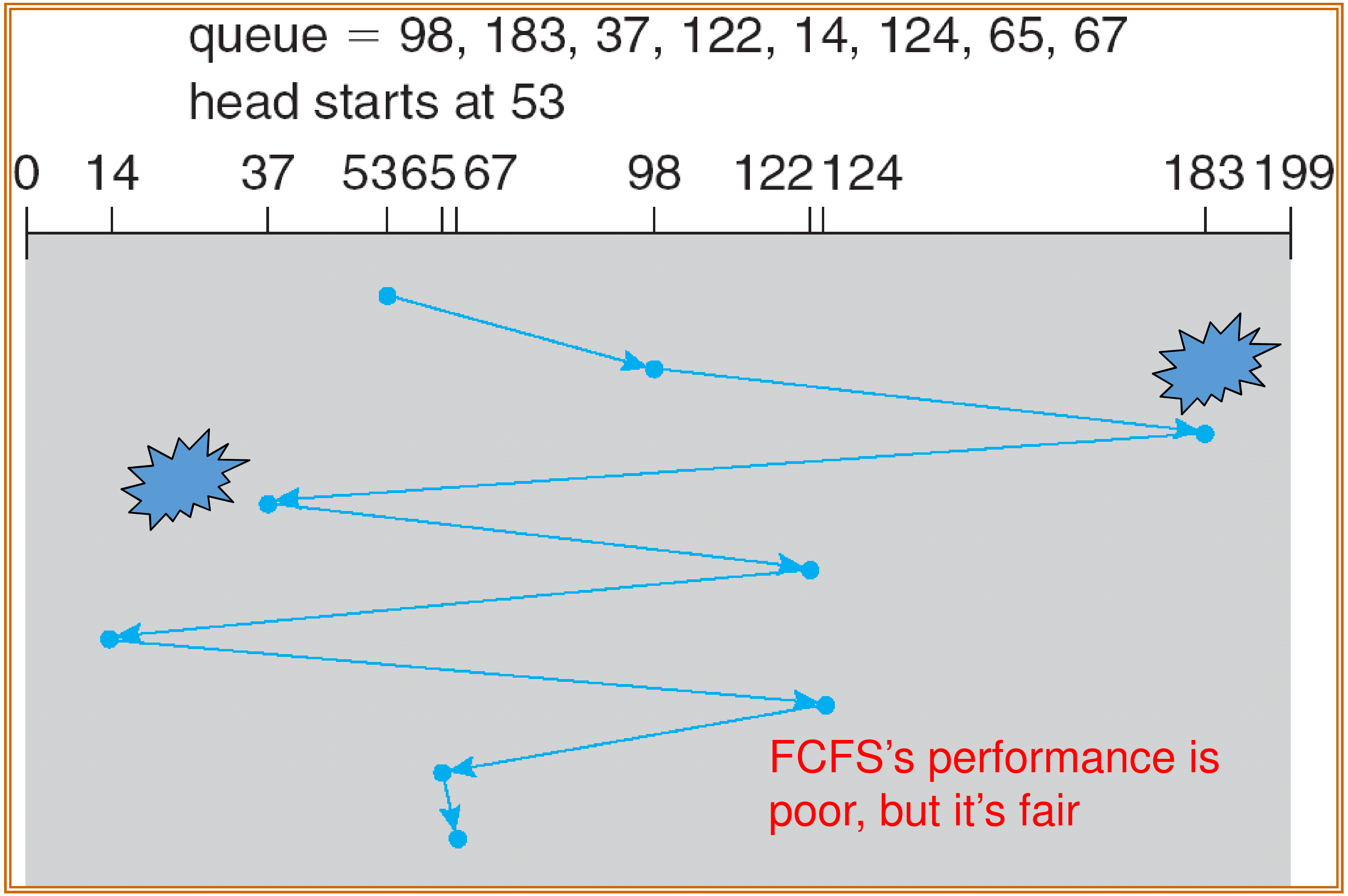
- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

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

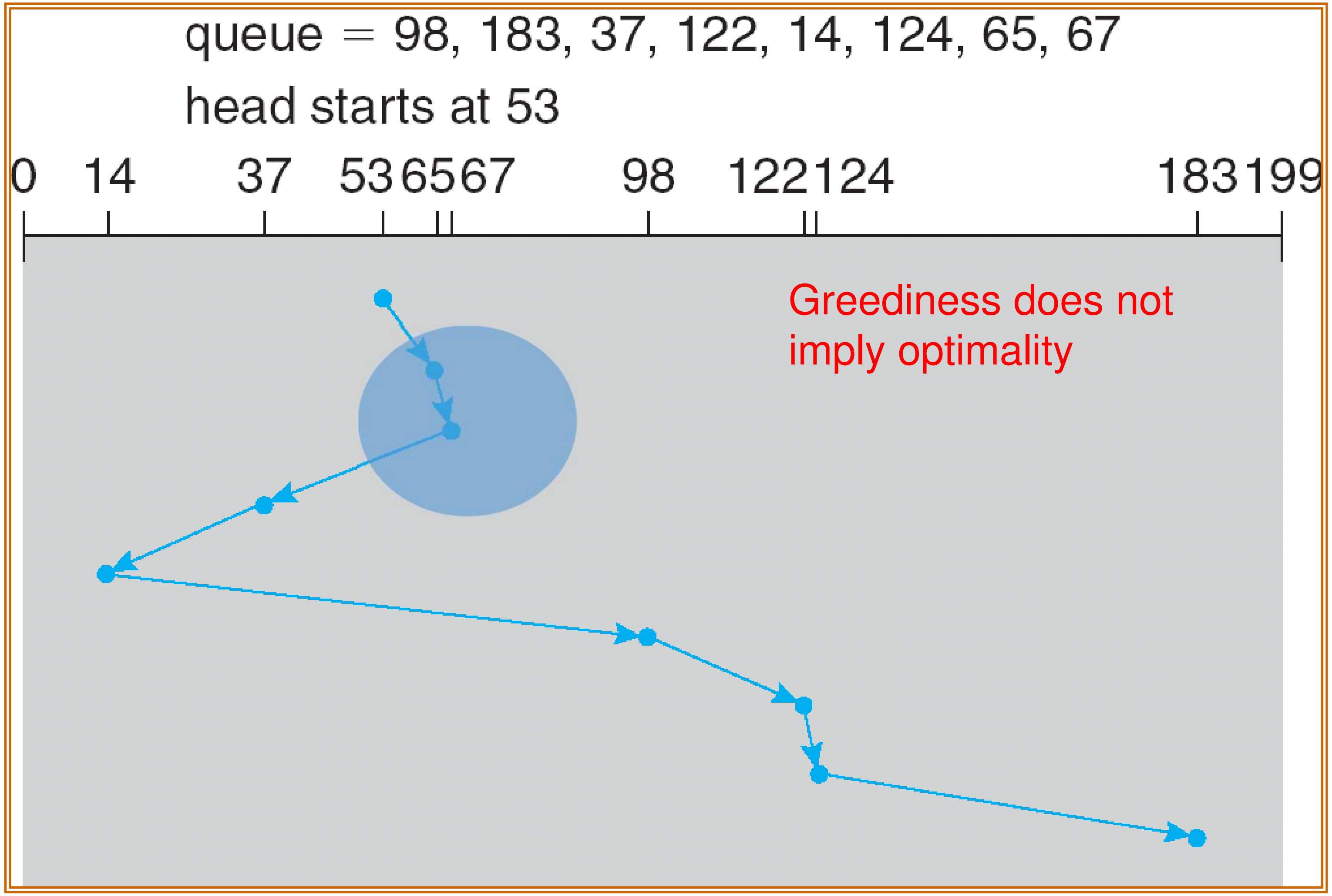
head starts at 53

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

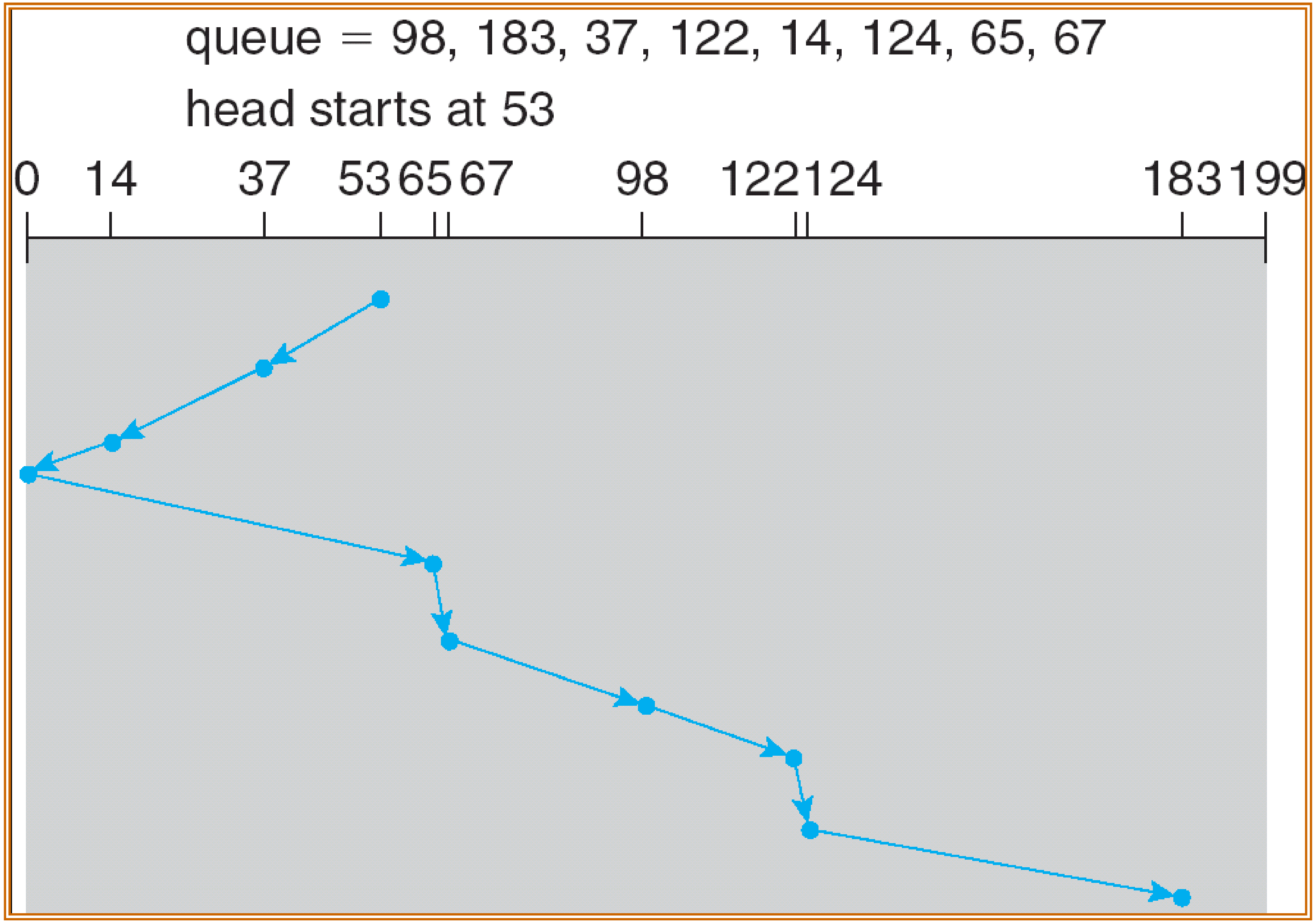
head starts at 53

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



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

head starts at 53

SCAN

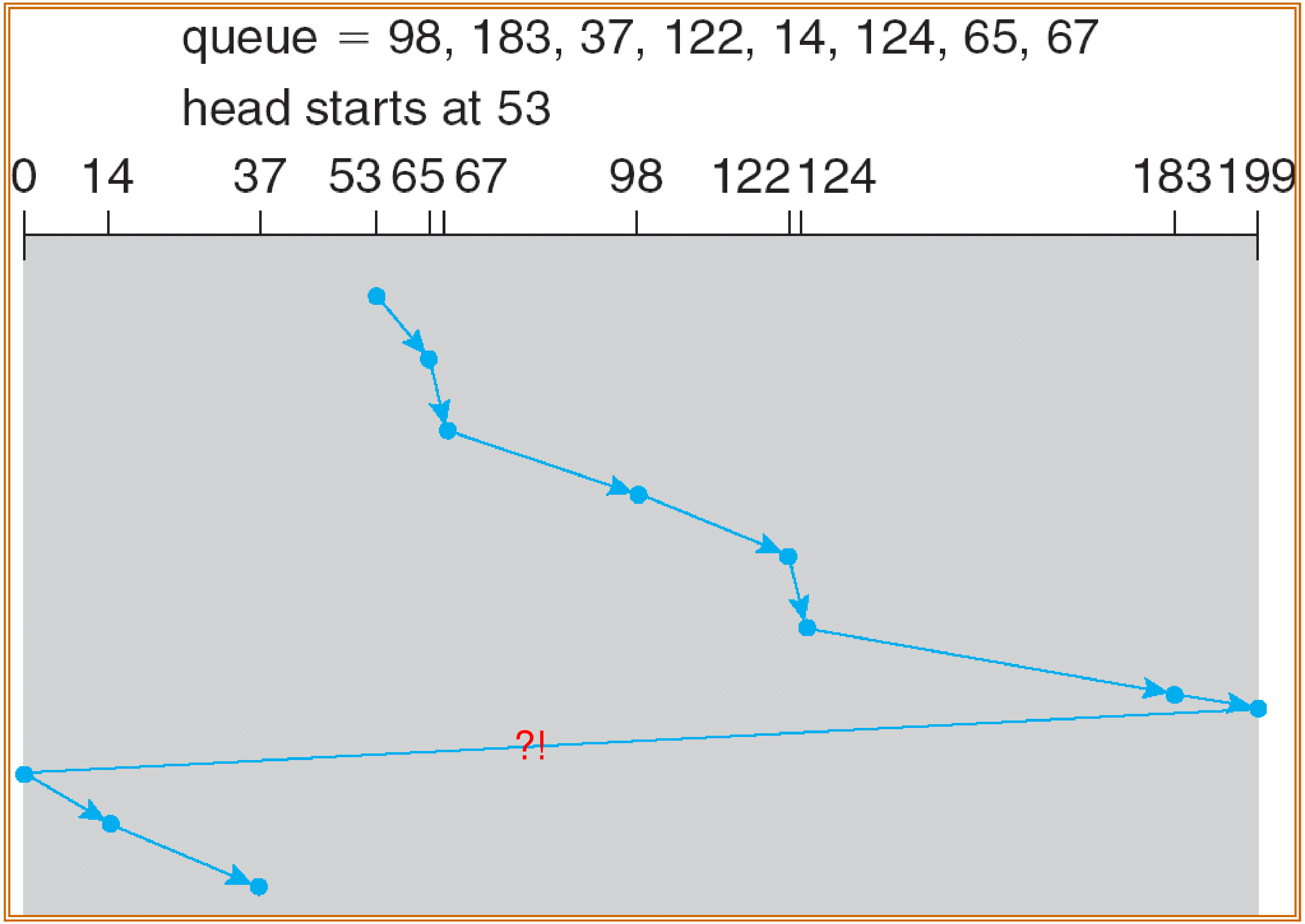
- A fair algorithm
- In the worst case, a request has to wait for 2 full strokes

- The waiting time of each cylinder is not uniform
    - 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

- Motivation
  - Provides a more uniform wait time than SCAN
  - In addition, on head reversal at an end of disk, density of requests near this end is low
- 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

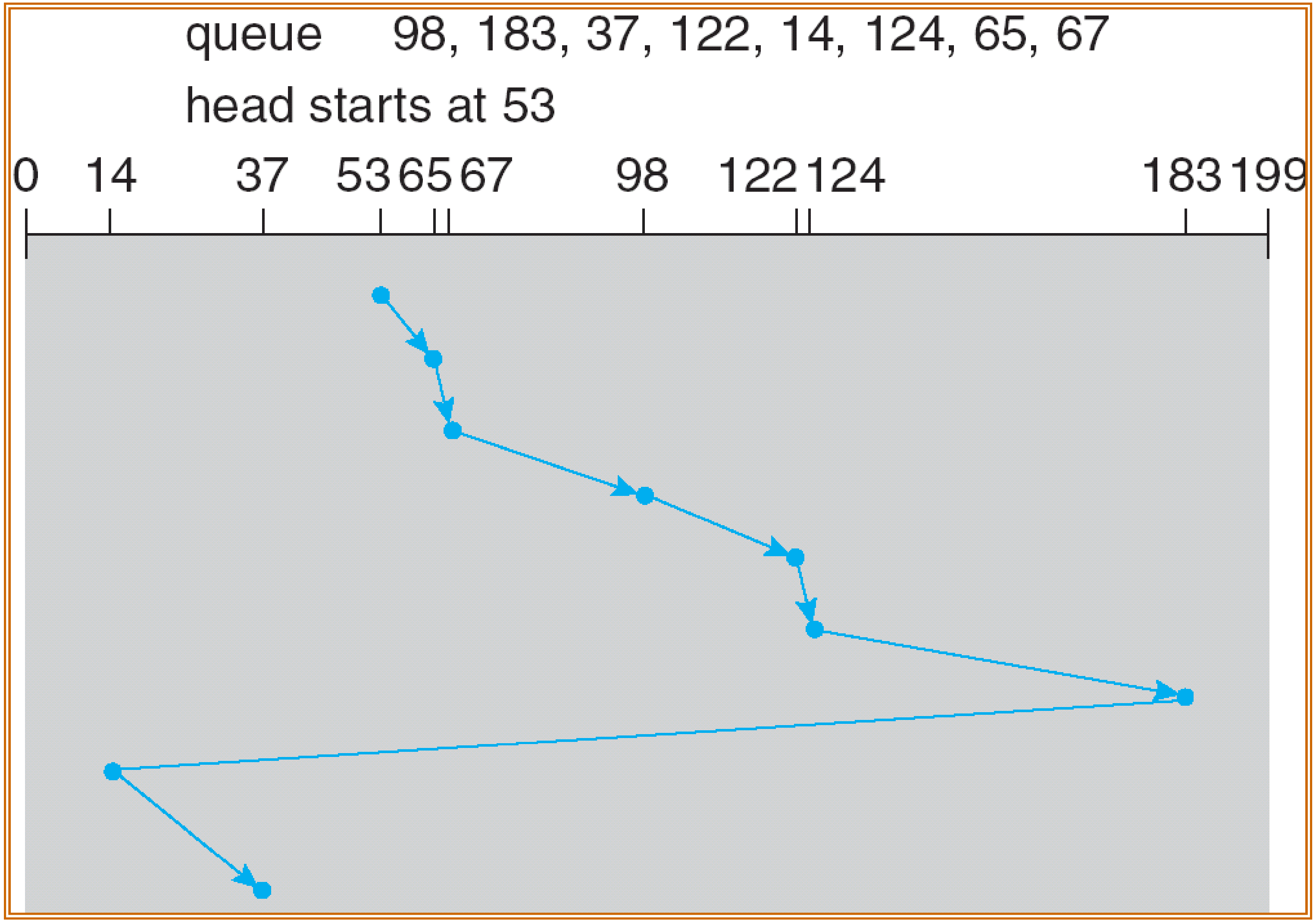
C-SCAN Disk Scheduling



C-LOOK

- 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.

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

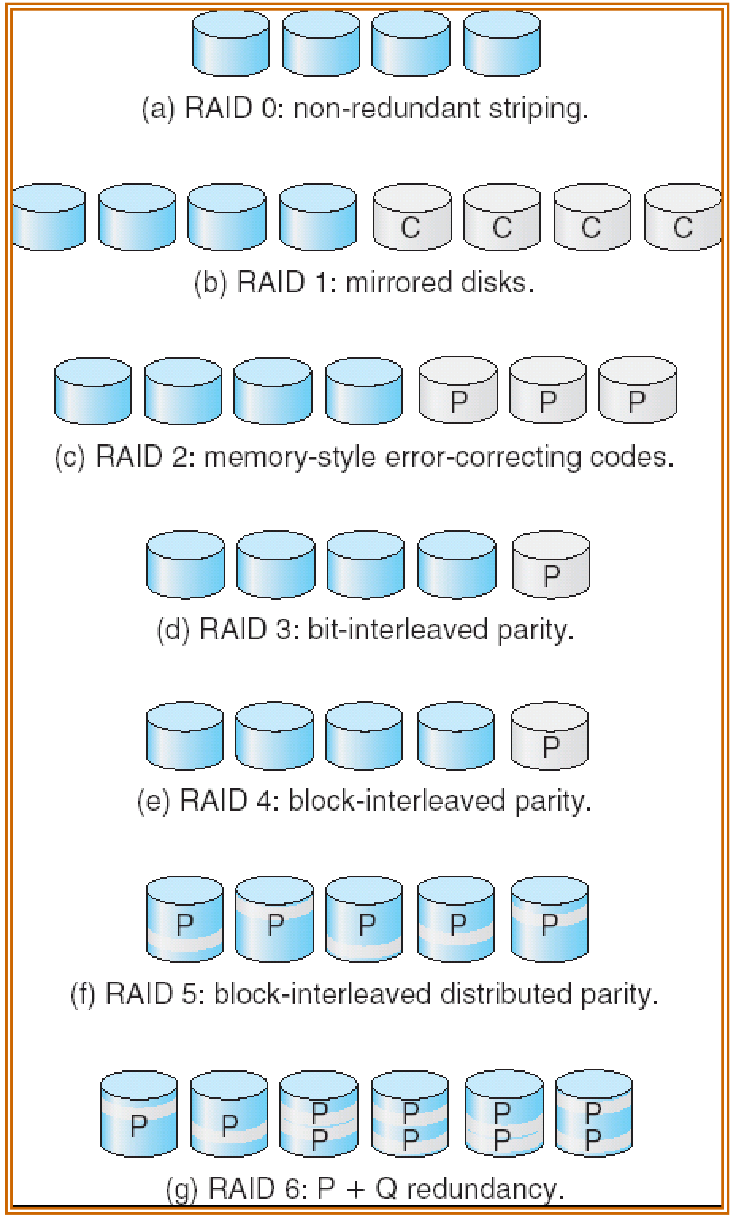
- Disk scheduling problem is NP-hard
  - All the methods mentioned above are not optimal (in terms of the total seek distance)
- Hard to optimize rotational delay from operating systems
  - The HDD firmware knows the current rotation angle better
  - Delegating disk scheduling to the HDD firmware
  -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

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.

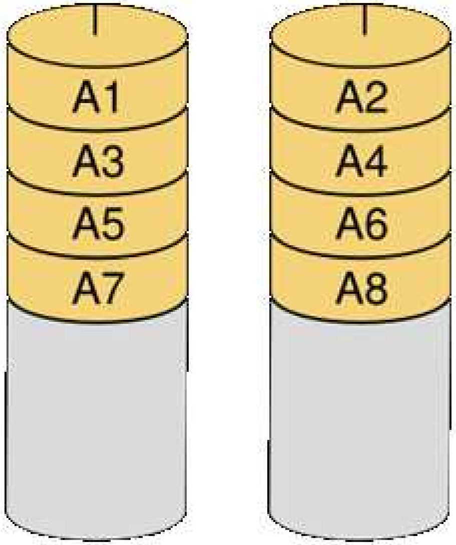
(c) RAID 2: memory-style error-correcting codes.

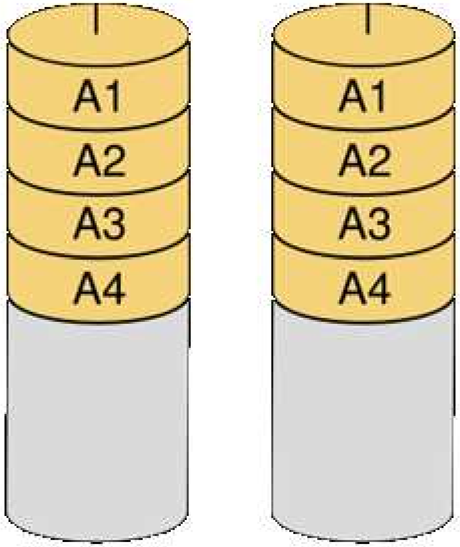
(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

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

(g) RAID 6: P + Q redundancy.



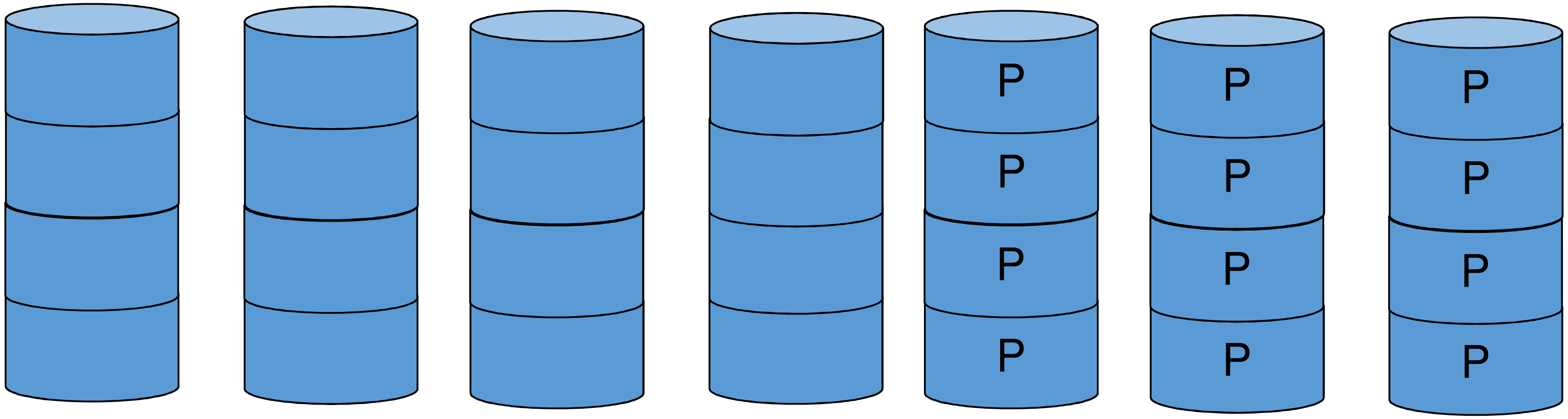


RAID0

RAID1

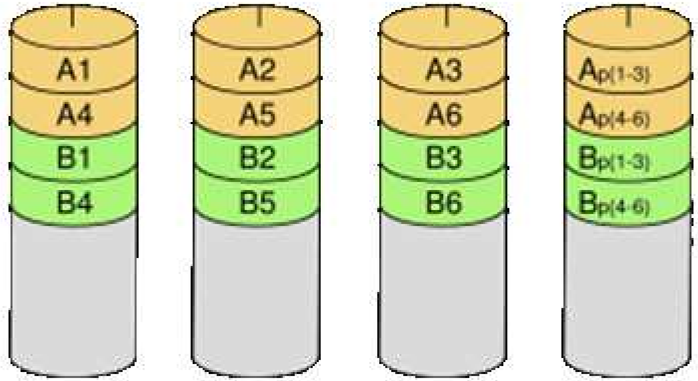
Striping. Aiming at parallelism

Mirroring, 100% redundancy



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

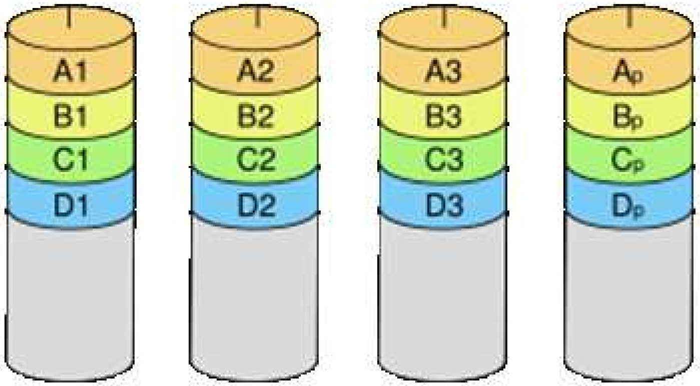
# of parity disks = log2(# of data disks)



RAID 3

Bit-interleaved
(or sub-block-interleaved)

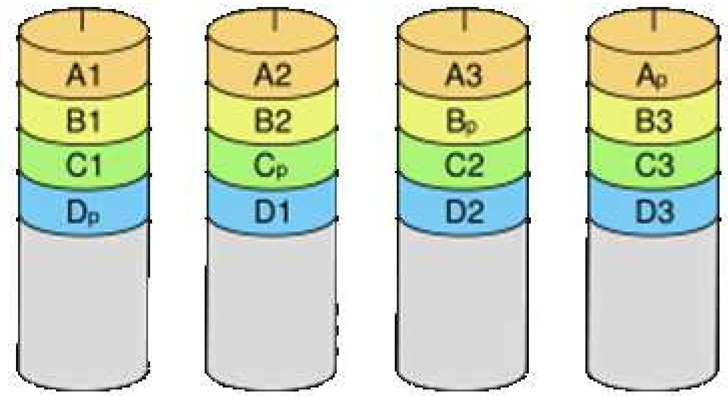
Fully interleaved, one R/W involves all disks



RAID4

Block-interleaved

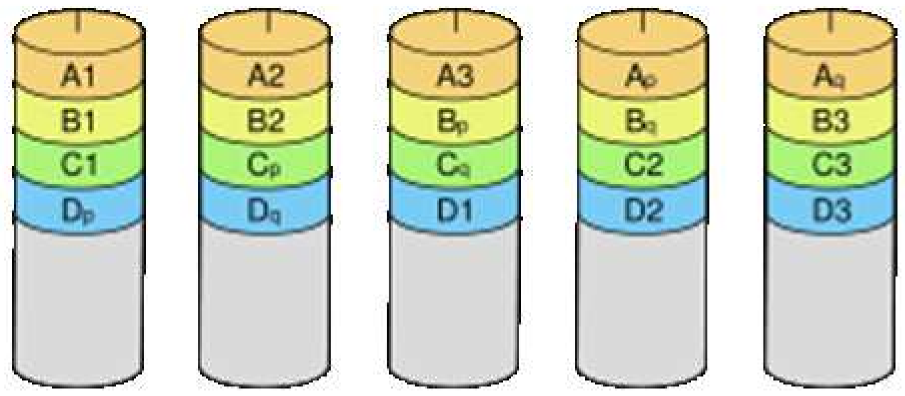
One R involves one disk
One W involves two disks
Parity disk → bottleneck



RAID 5

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

Simple XOR-based parity



RAID 6

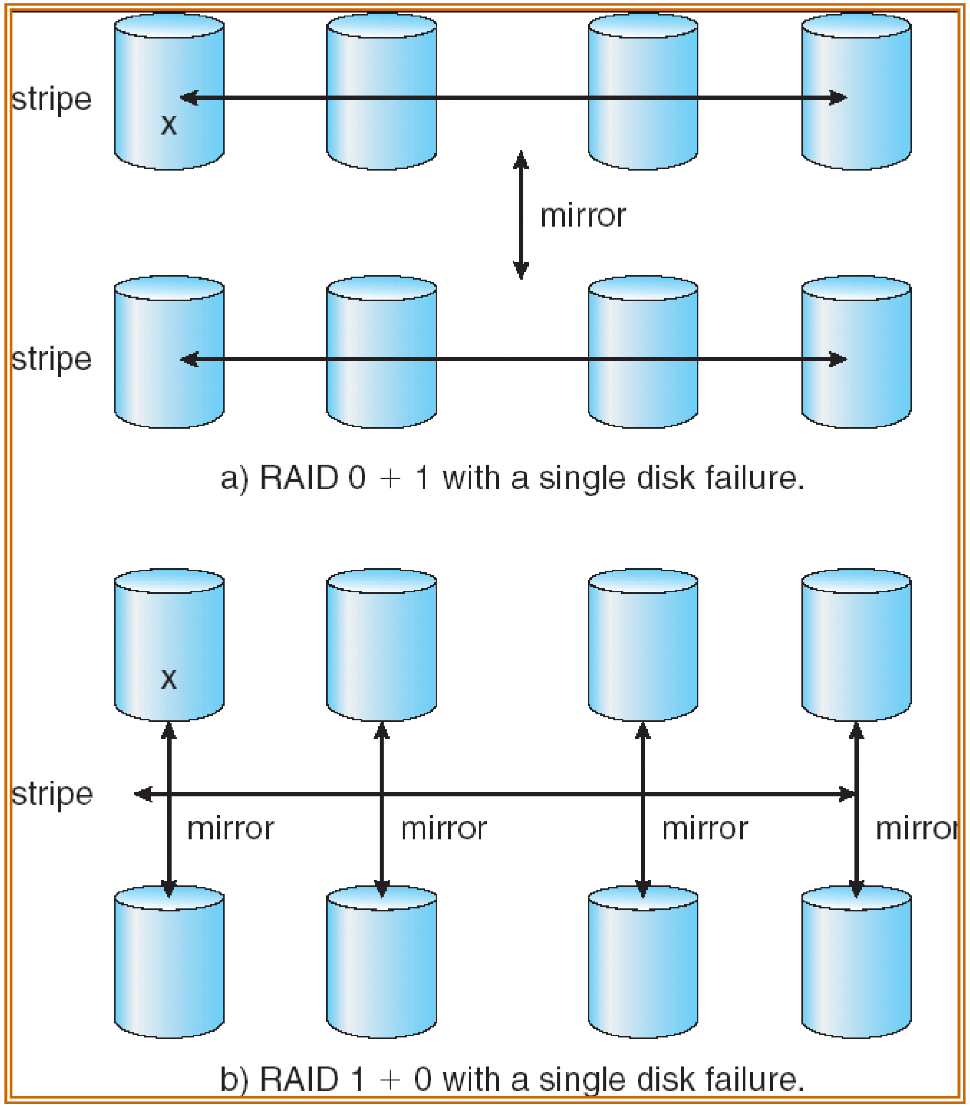
Choosing 4 blocks out of
{1,2,3,4,p,q} sufficiently
reconstruct {1,2,3,4}

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



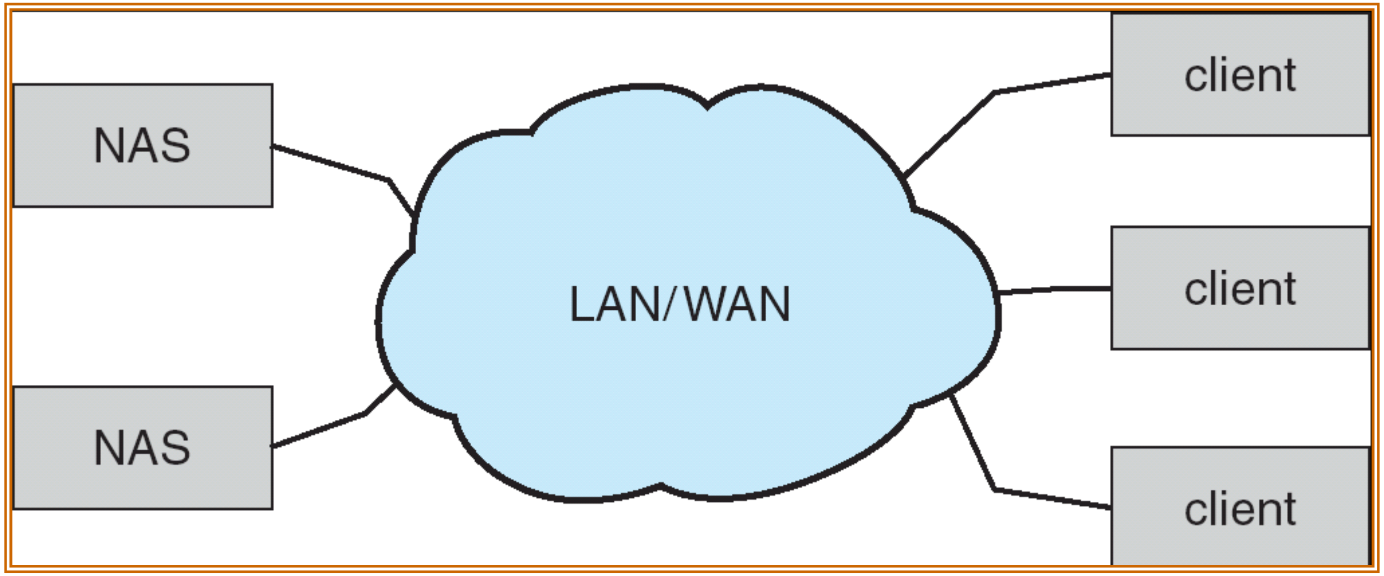
a) RAID 0 + 1 with a single disk failure.

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

← Better survivability

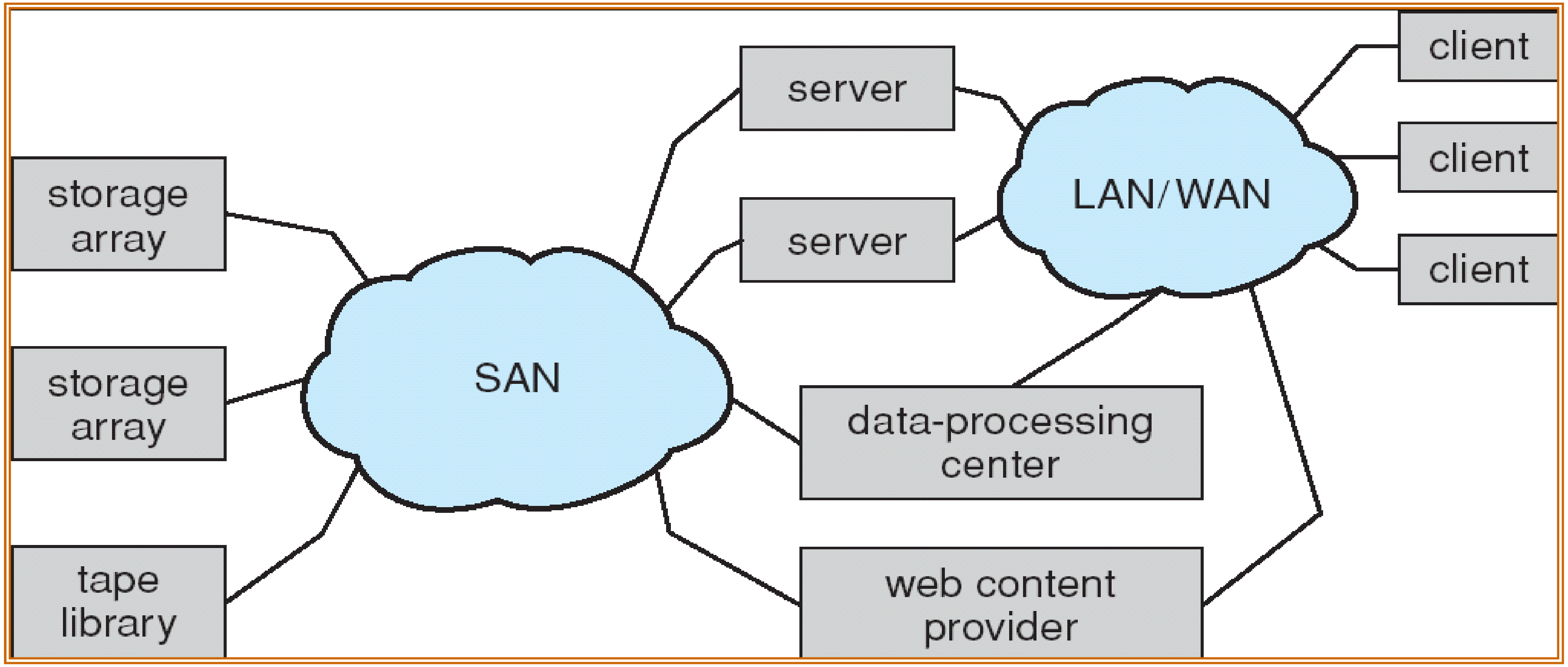
Network-Attached Storage

- 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



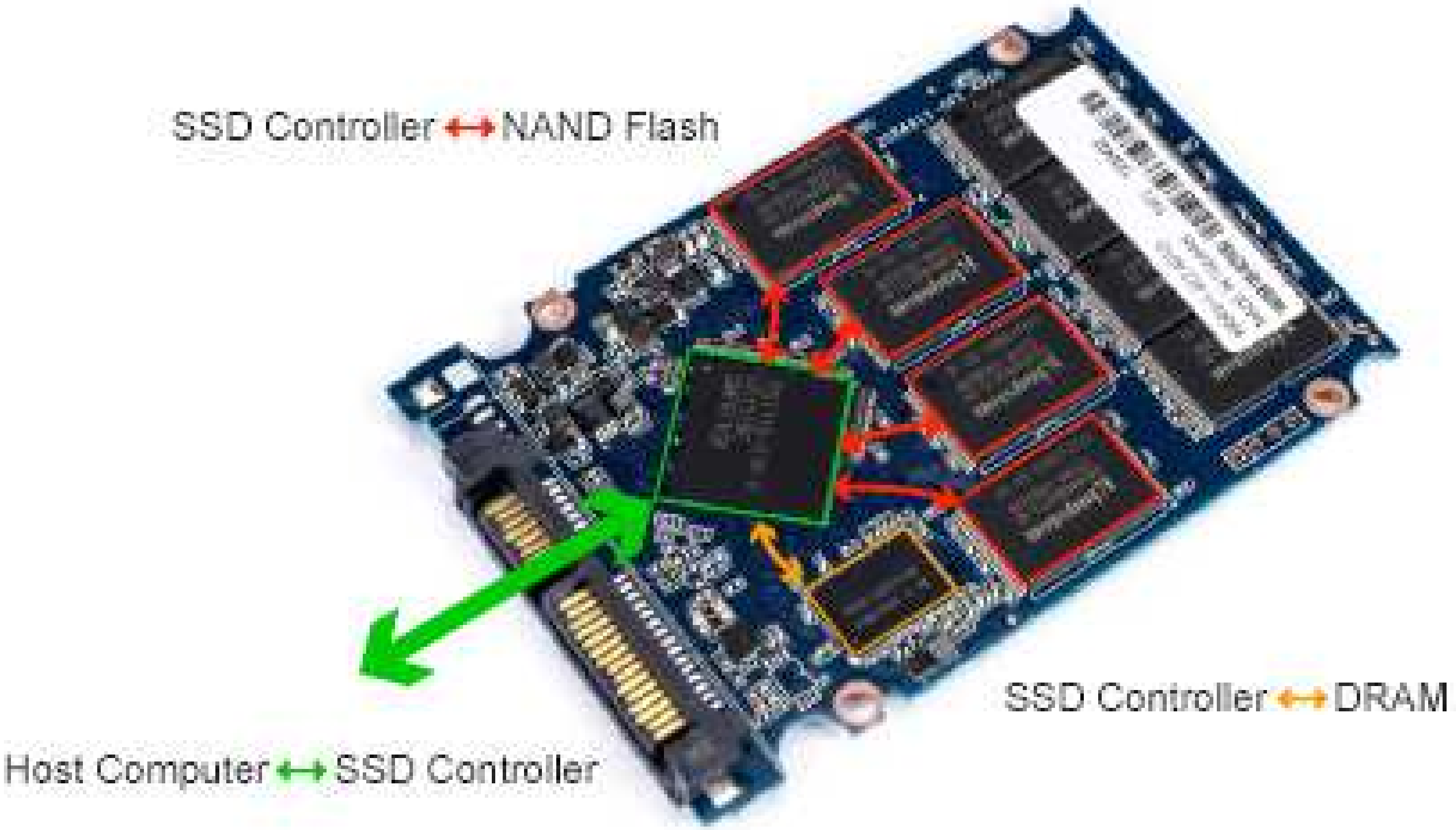
SAN vs. NAS

- 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)

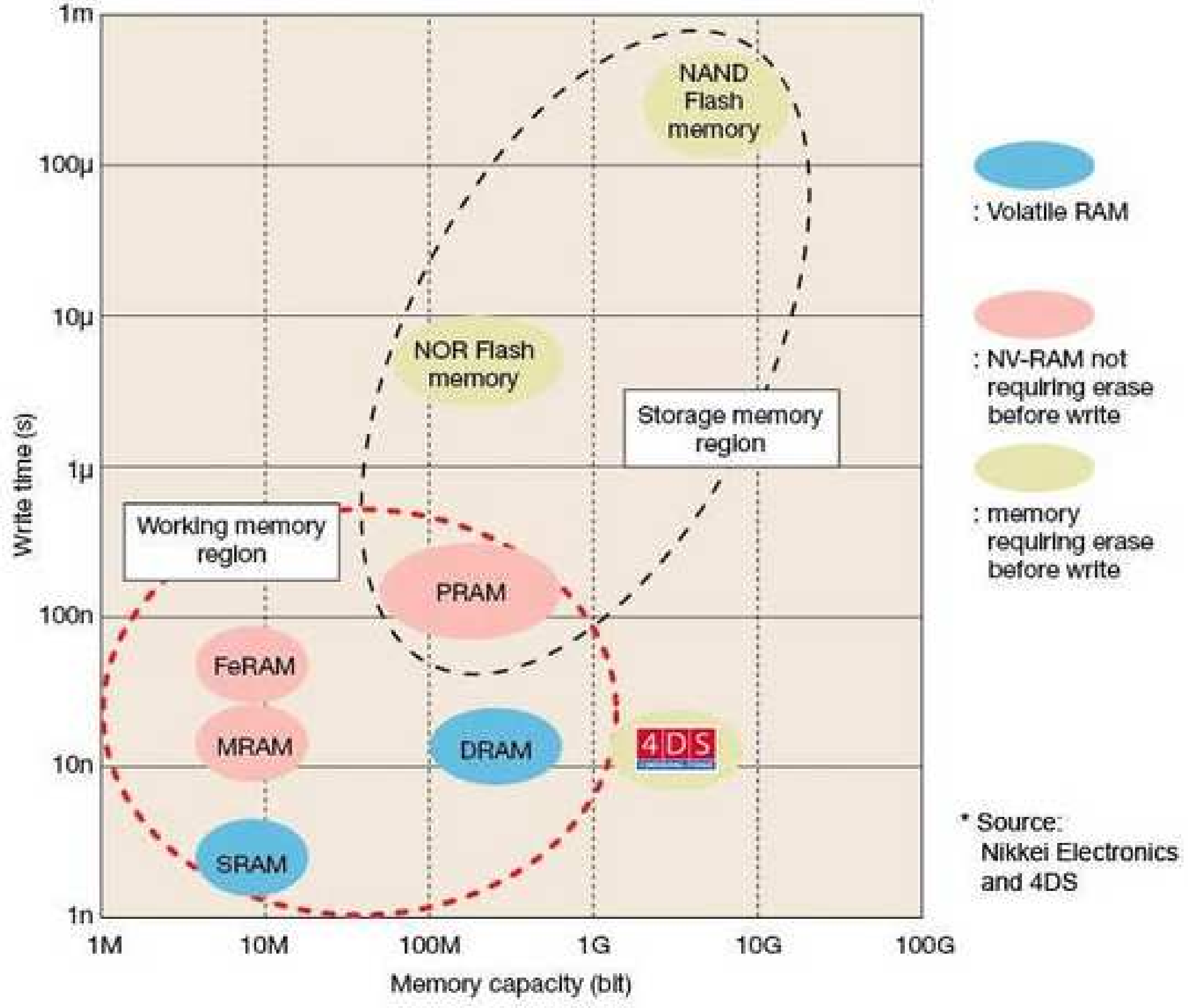
- Storage devices that emulate standard block devices using non-volatile memory
  - Flash memory or battery-backed RAM
  - The OS use the legacy I/O stack on top of SSDs
- Products
  - Embedded flash cards, SD cards, USB thumb drives, SSDs, PCI-e flash cards
- Performance
  - RAM disk > SSD >> HDD
-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

SSD Internal Organization



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

Non-Volatile Memory



* Source: Nikkei Electronics and 4DS

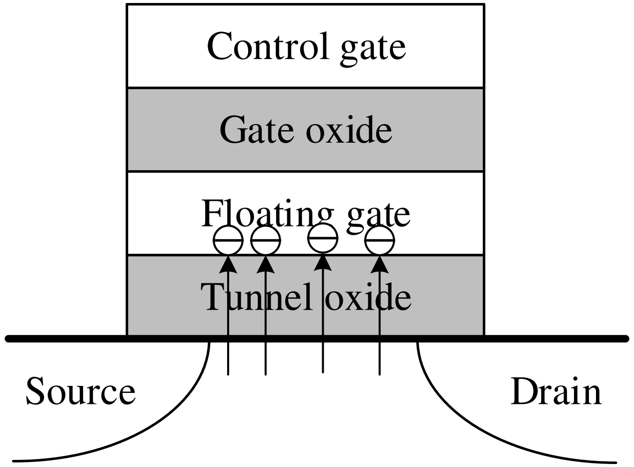
Faster write

Smaller capacity

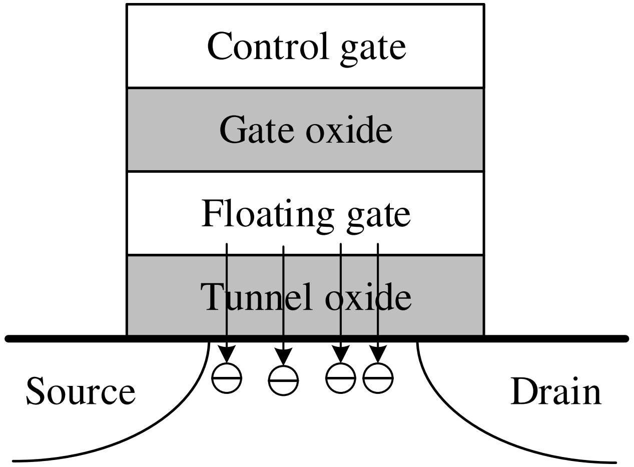
http://www.storagenewsletter.com/news/flash/4ds-start-up-profile?snc=7835

Flash Memory

- Cell structure, flash program and erase

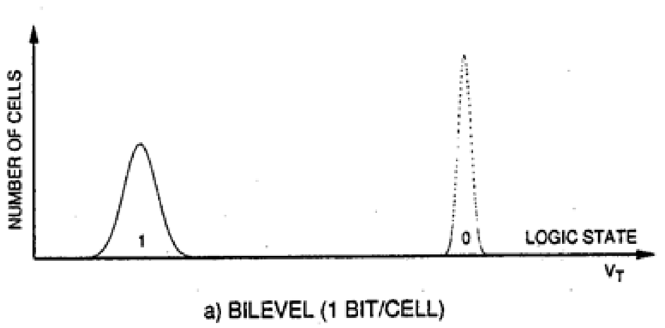


Program (write)

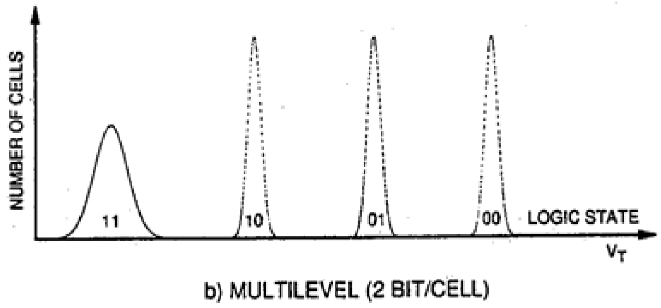


Erase

Multilevel Cells



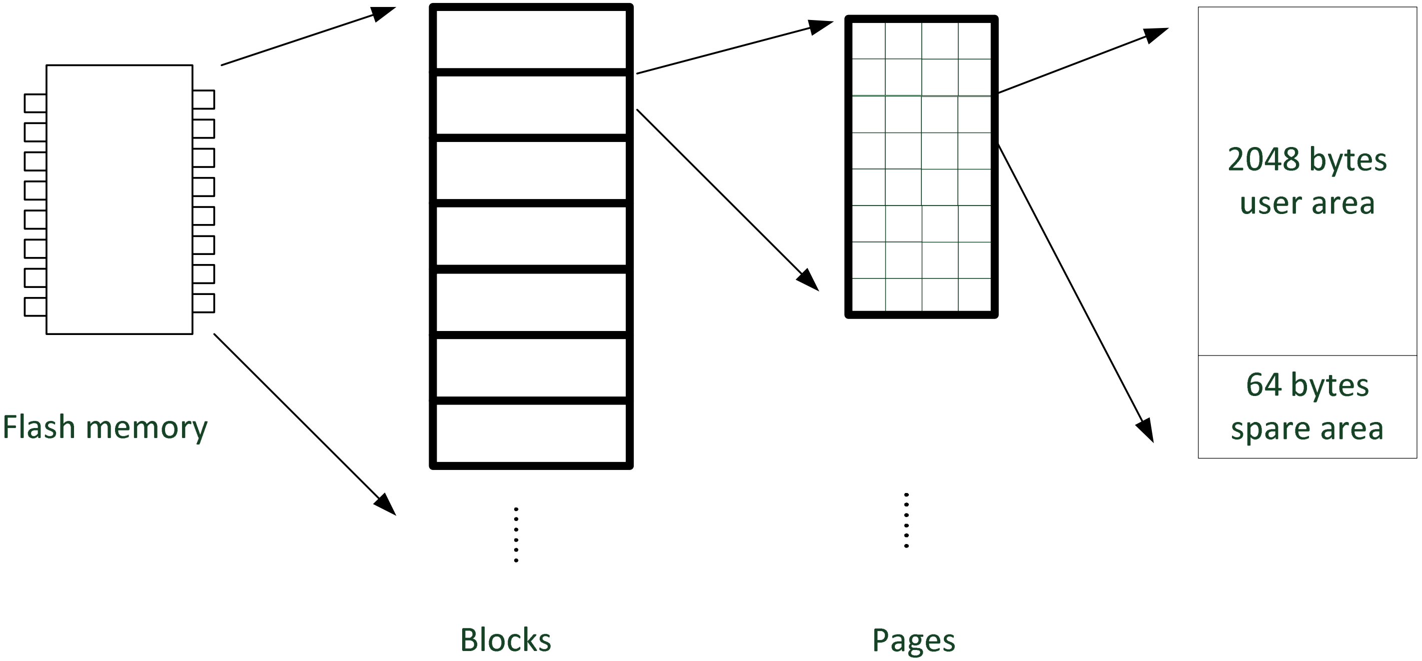
Single-level cell



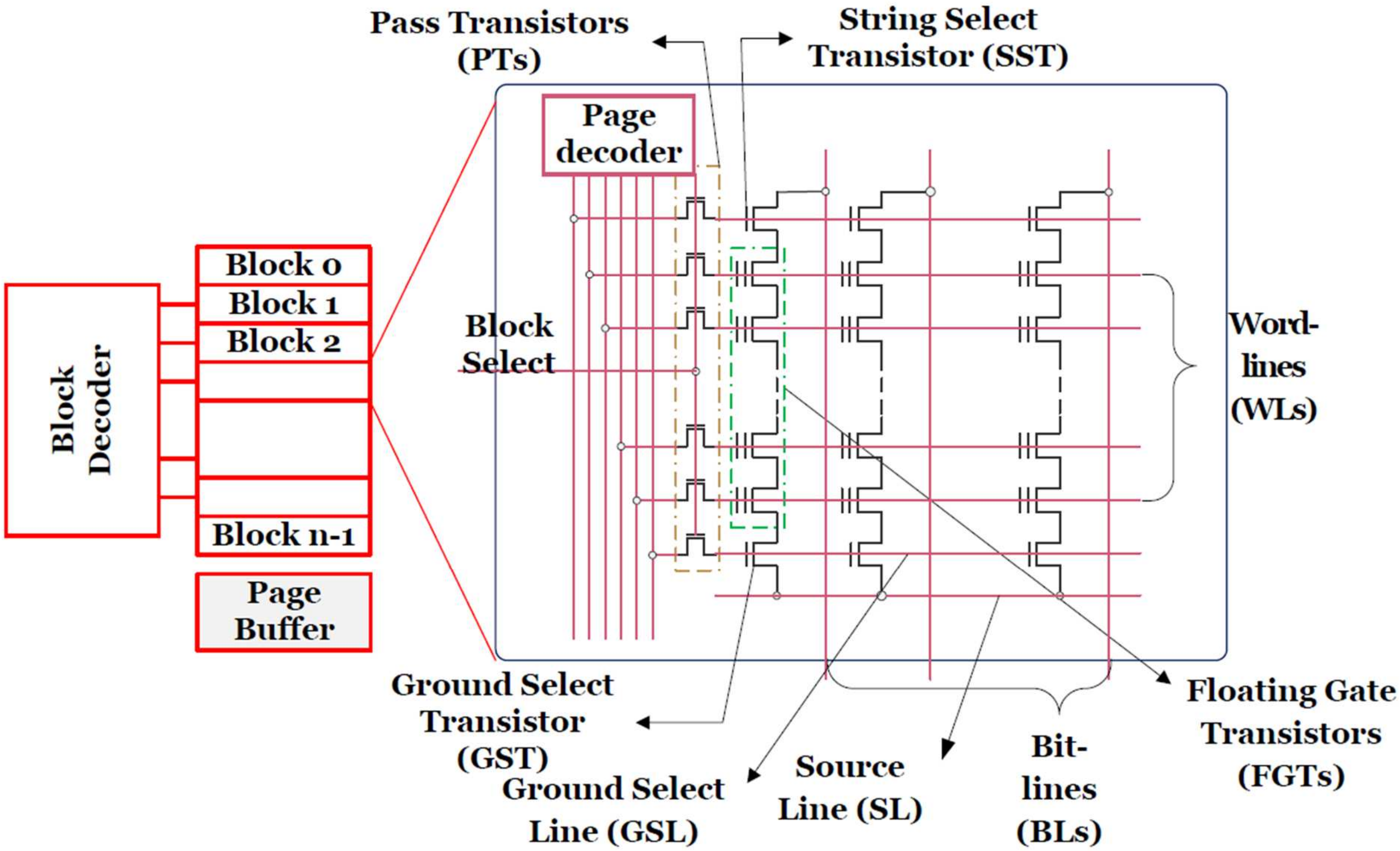
Multi-level cell

- SLC vs. MLC flash
  - Comparable read speed
  - SLC writes about 2x or 3x faster than MLC
  - P/E endurance: 5K cycles (MLC), 100K (SLC)
  - SLC is 2x or 3x more expensive than MLC (and increasing)
  - Hybrid SSDs, dynamic density SSDs
- Now TLC, QLC are in mass production

NAND Flash Geometry



- Unit size
  - Read/write: page
  - Erase: block

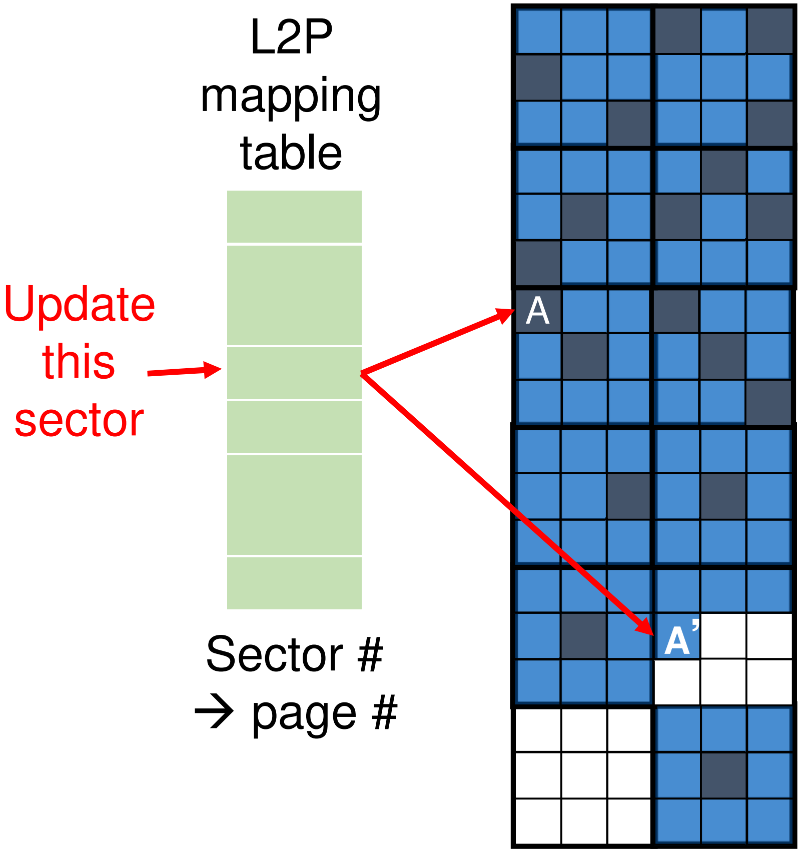

Pass Transistors (PTs)
String Select Transistor (SST)
Page decoder
Block Decoder
Block 0
Block 1
Block 2
Block n-1
Page Buffer
Block Select
Word-lines (WLs)
Ground Select Transistor (GST)
Ground Select Line (GSL)
Source Line (SL)
Bit-lines (BLs)
Floating Gate Transistors (FGTs)

Flash Translation Layer (FTL)

- A firmware layer inside of SSDs
  - Hiding flash memory physics from the host
- Provide block device emulation to the host
- Manage flash memory inside of SSDs
  - Logical-to-physical address translation
  - Garbage collection
  - Wear leveling

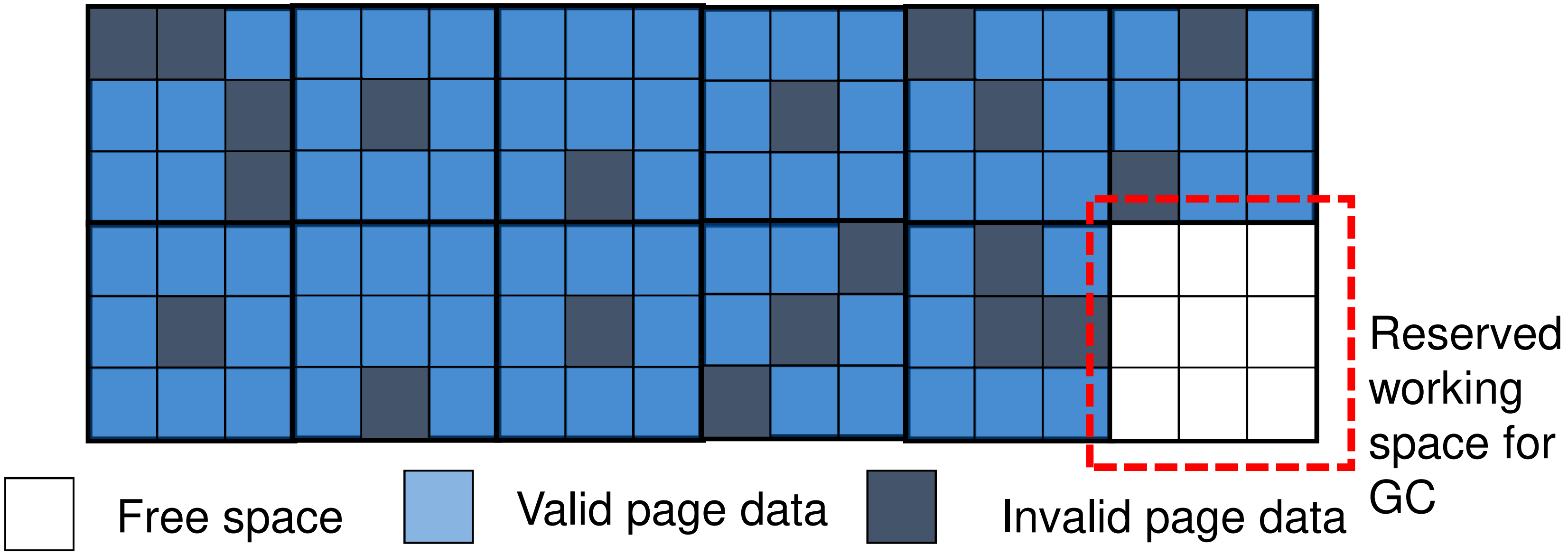
Logical-to-Physical Address Translation

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



Garbage Collection

- Recycle memory space occupied by invalid data through block erasure
- Victim selection
  - Minimize the page-copy overhead



Reserved working space for GC

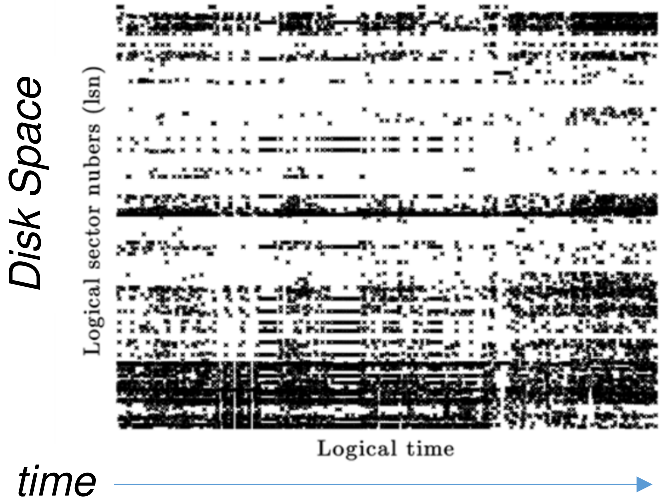
Free space

Valid page data

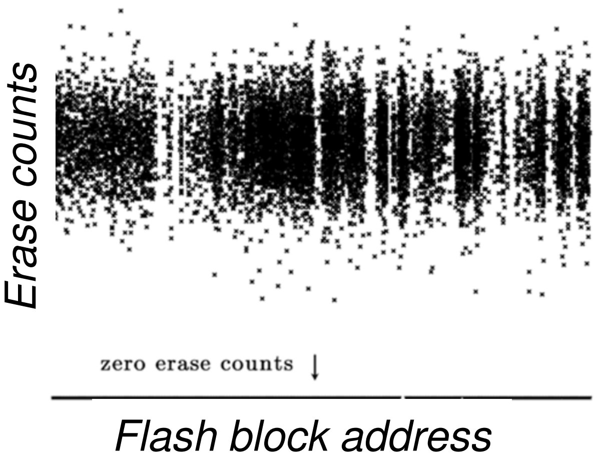
Invalid page data

Wear Leveling

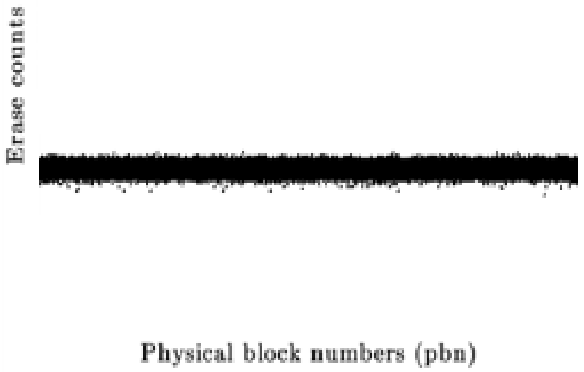
- Typically a (MLC) block endures 3000 cycles of program-erase operations (P/E cycles)
- Locality of write creates frequently written blocks
- Delay the first block retirement by migrating cold data



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