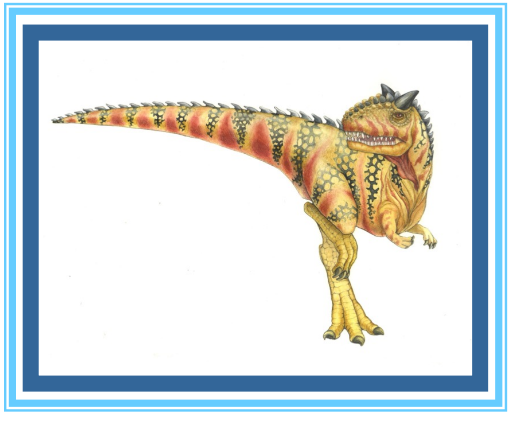
Lecture 10: Mass-Storage Systems



Lecture 10: Mass-Storage Systems

- Overview of Mass Storage Structure
- Disk Structure
- Disk Attachment (SAN and NAS)
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure
- Stable-Storage Implementation

Objectives

- To Describe Physical Structure of Secondary Storage Devices and its Effects on Uses of Devices
- To Explain Performance Characteristics of Mass-Storage Devices
- To Evaluate Disk Scheduling Algorithms
- To Discuss OS Services Provided for Mass Storage, Including RAID

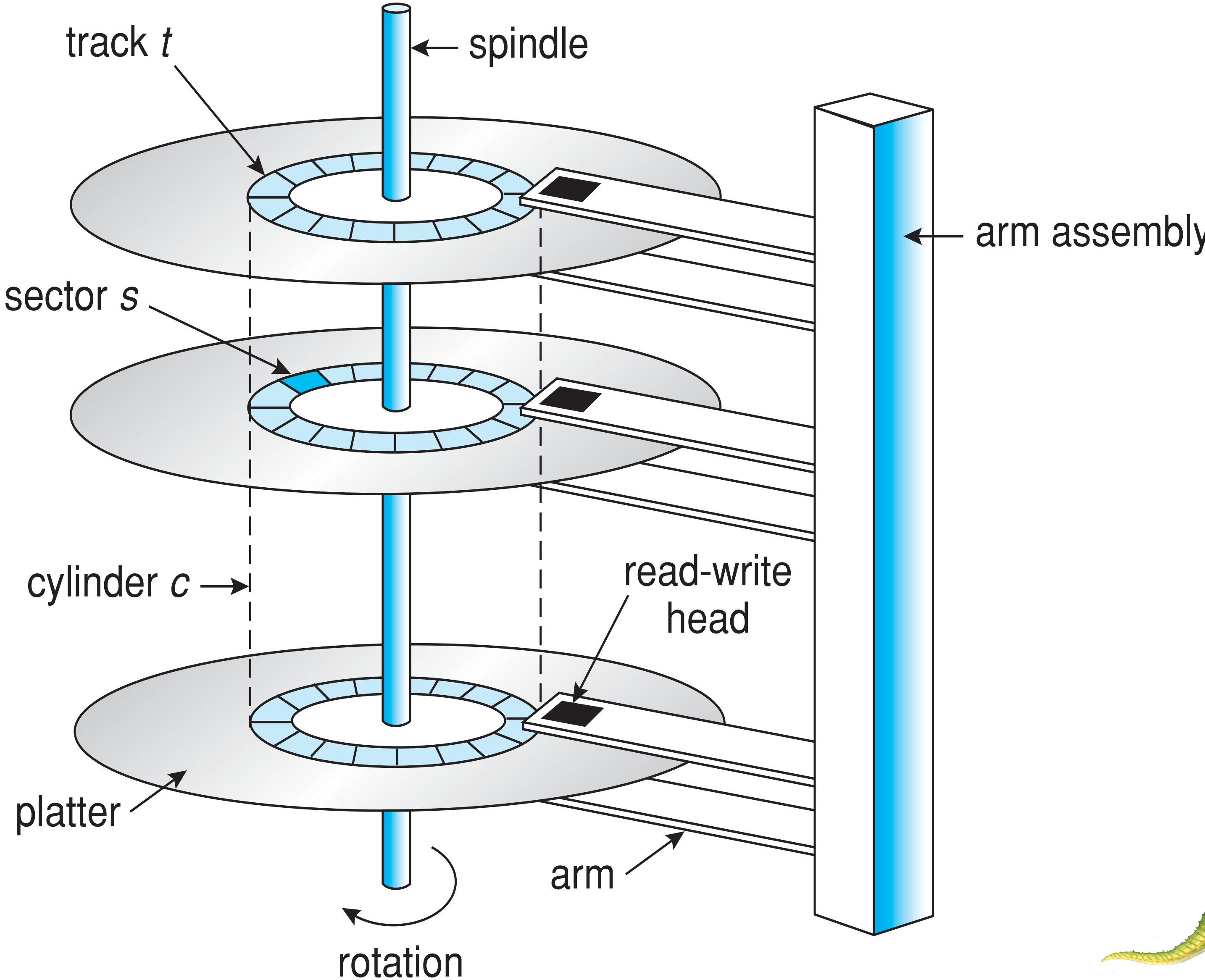
Overview of Mass Storage Structure

- **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
  - **Positioning time** (**random-access time**)
    - Time to move disk arm to desired cylinder (**seek time**)
    - Time for desired sector to rotate under disk head (**rotational latency**)
  - **Head crash** results from disk head making contact with disk surface -- That's bad

Overview of Mass Storage Structure (cont.)

- Disks can be Removable
- Drive Attached to Computer via **I/O bus**
  - Busses vary, including **EIDE**, **ATA**, **SATA**, **USB**, **Fibre Channel**, **SCSI**, **SAS**, **Firewire**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array

Moving-Head Disk Mechanism



Hard Disks

- Platters Range from 0.85" to 24" (Historically)
  - Commonly 3.5", 2.5", and 1.8"
- Range from 300GB to 16TB per drive
- Performance
  - Transfer Rate – theoretical – 6 or 12 Gb/sec
  - Effective Transfer Rate – real – 1Gb/sec
  - Seek time from 3ms to 12ms – 9ms common for desktop drives
  - Avg seek time measured or calculated based on 1/3 of tracks
  - Latency based on spindle speed
    - 1 / (RPM / 60) = 60 / RPM
  - Average latency = ½ latency

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 rotational latency
  - For fastest disk: 3ms + 2ms = 5ms
  - For slow disk: 9ms + 5.56ms = 14.56ms
- Avg I/O time = avg access time **+** (amount to transfer / transfer rate) **+** controller overhead
- E.g.: to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a 0.1ms controller overhead =
  - 5ms + 4.17ms + 0.1ms + transfer time =
  - Transfer time = 4KB / 1Gb/s * 8Gb / GB * 1GB / 1024^2KB = 32 / (1024^2) = 0.031 ms
  - Average I/O time (4KB)= 9.27ms + .031ms = 9.301ms

First Commercial Disk Drive

1956
IBM RAMDAC computer included
IBM Model 350 disk storage system

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

Solid-State Disks

- Non-Volatile Memory Used like a Hard Drive
  - Many technology variations
- Can be More Reliable than HDDs ☺
- More Expensive per GB ☹
- Maybe have Shorter Life Span
- Less Capacity ☹
- Much Faster ☺
  - No moving parts → no seek time or rotational latency
- Busses can be too slow → connect directly to PCI for example (e.g., NVMe SSDs)

Magnetic Tape

- Was Early Secondary-Storage Medium
  - Evolved from open spools to cartridges
- Relatively Permanent and Holds Large Quantities of Data
- Access Time Very Slow
- Random access ~1000X slower than HDD
- Mainly Used for Backup, Storage of Infrequently-used data, Transfer Medium between Systems

Magnetic Tape (cont.)

- Kept in spool and wound or rewound past read-write head
- Once Data under Head, Transfer Rates Comparable to Disk
  - 140MB/sec and greater
- 200GB to 30+TB Typical Storage
- Common technologies are LTO-{3,4,5} and T10000

Disk Structure

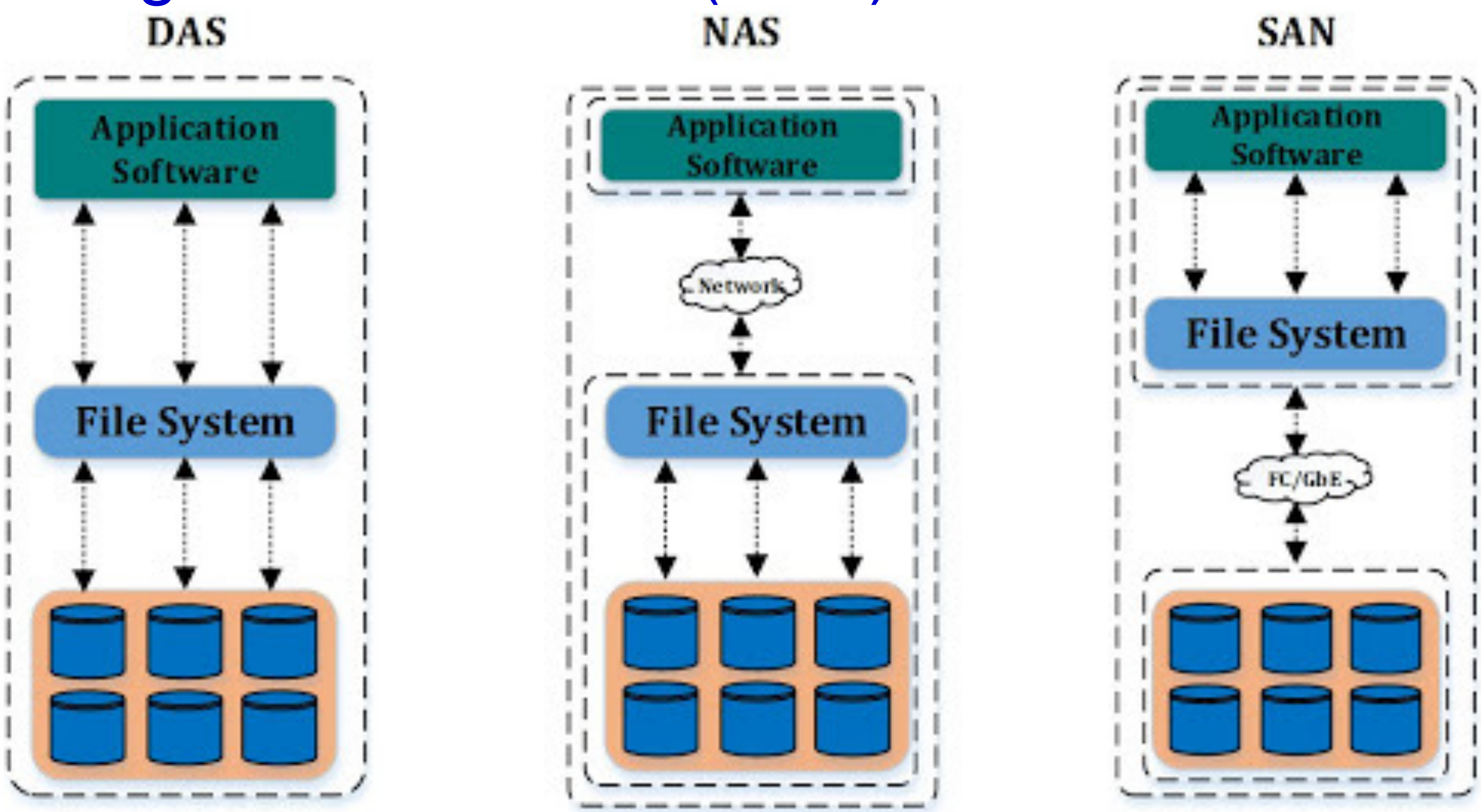
- Disk Drives are Addressed as Large 1-dim Arrays of **Logical Blocks**
  - where logical block is smallest unit of transfer
  - Low-level formatting creates **logical blocks** on physical media
- 1-dim array of logical blocks mapped into sectors of disk sequentially
  - Sector 0: first sector of first track on outermost cylinder

Disk Structure (cont.)

- Mapping Proceeds in Order through
  - Track and then rest of tracks in that cylinder
  - Then through rest of 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

DAS vs. NAS vs. SAN

- Direct Attached Storage (DAS)
- Network Area Storage (NAS)
- Storage Area Network (SAN)



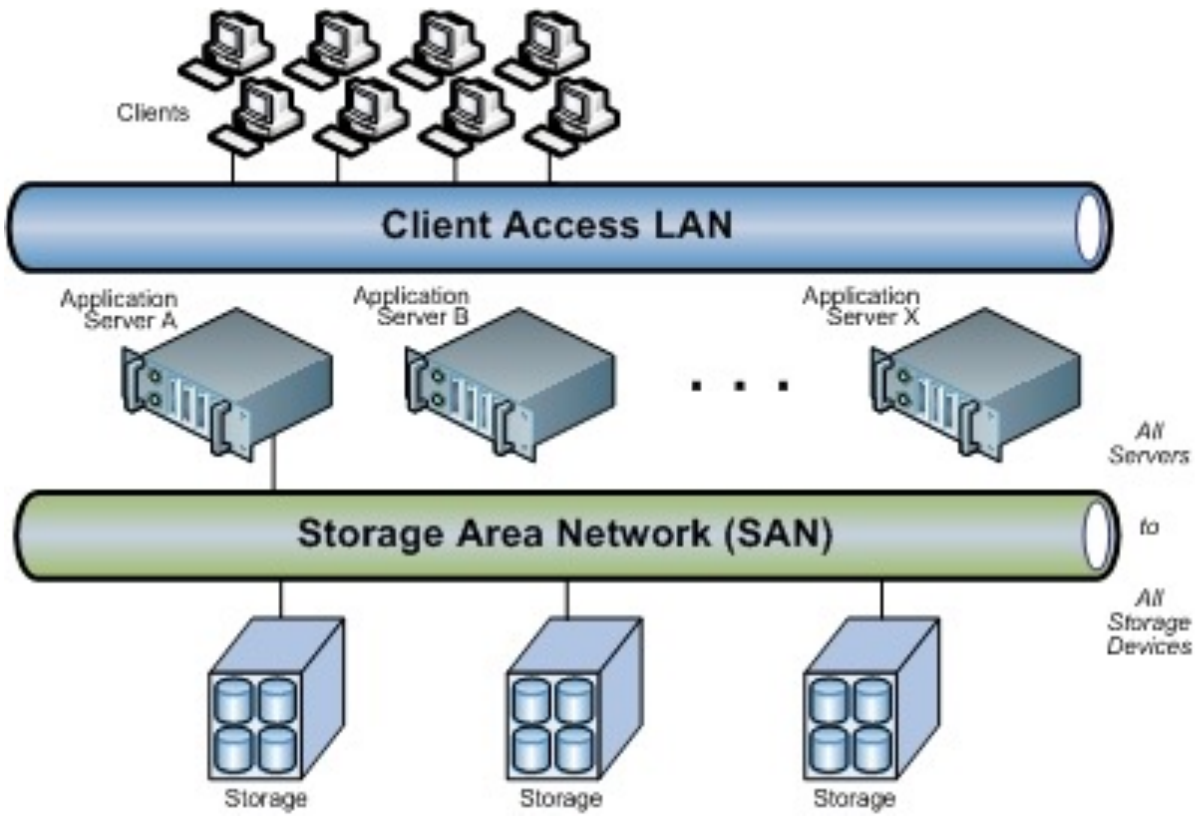
Disk Attachment

- Host-Attached Storage accessed through I/O ports talking to I/O busses
- SCSI itself is a bus, up to 16 devices on one cable, **SCSI initiator** requests operation and **SCSI targets** perform tasks
  - Each target can have up to 8 **logical units** (disks attached to device controller)
- FC is High-Speed Serial Architecture
  - Can be switched fabric with 24-bit address space – the basis of **storage area networks (SANs)** in which many hosts attach to many storage units
- I/O directed to bus ID, device ID, logical unit (LUN)

Storage Array

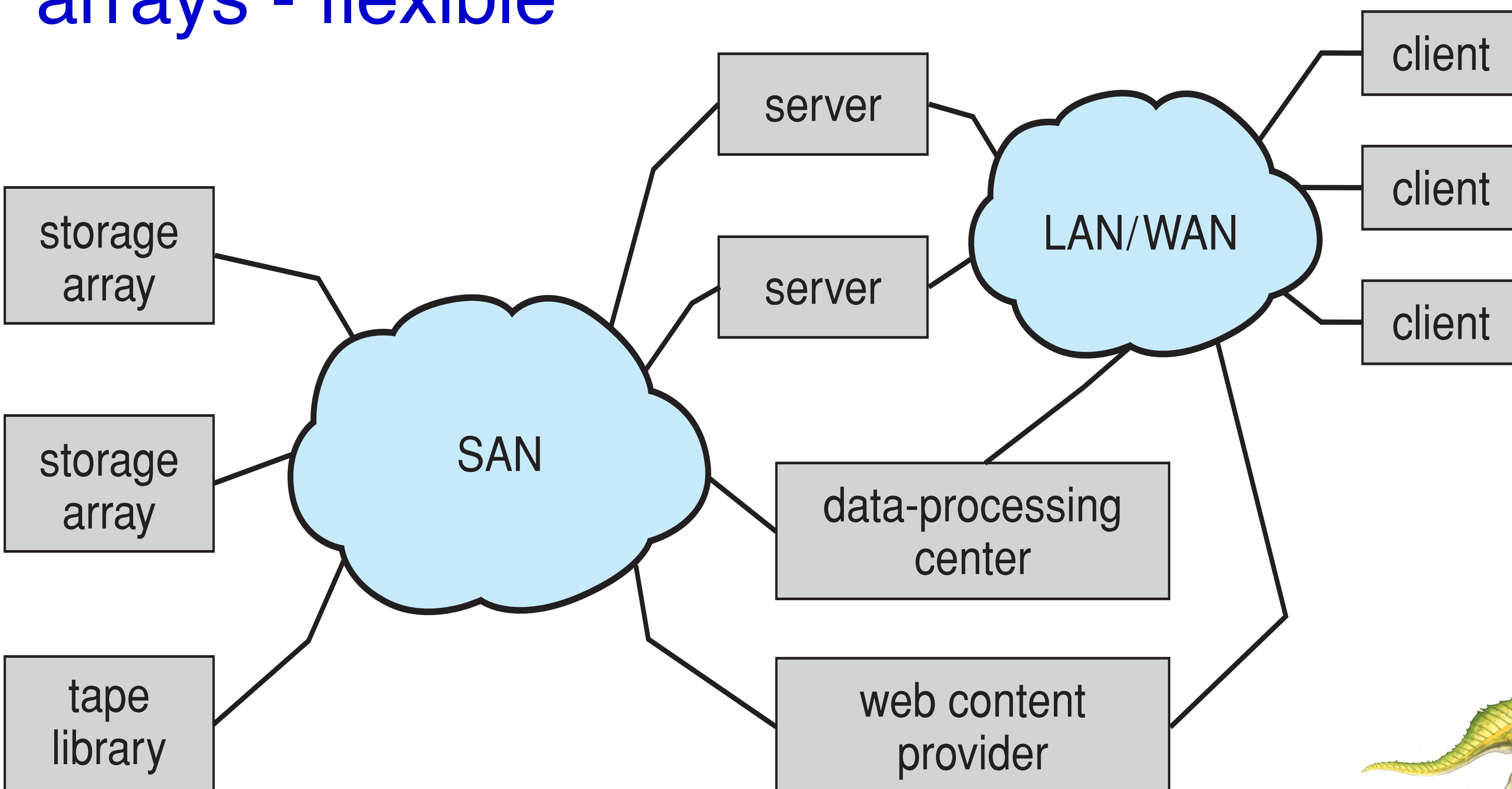
- Can just attach disks, or arrays of disks
- Storage Array has controller(s), provides features to attached host(s)
  - Ports to connect hosts to array
  - Memory, controlling software (sometimes NVRAM, etc)
  - A few to thousands of disks
  - RAID, hot spares, hot swap (discussed later)
  - Shared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

Storage Area Network



Storage Area Network

- Common in Large Storage Environments
- Multiple hosts attached to multiple storage arrays - flexible



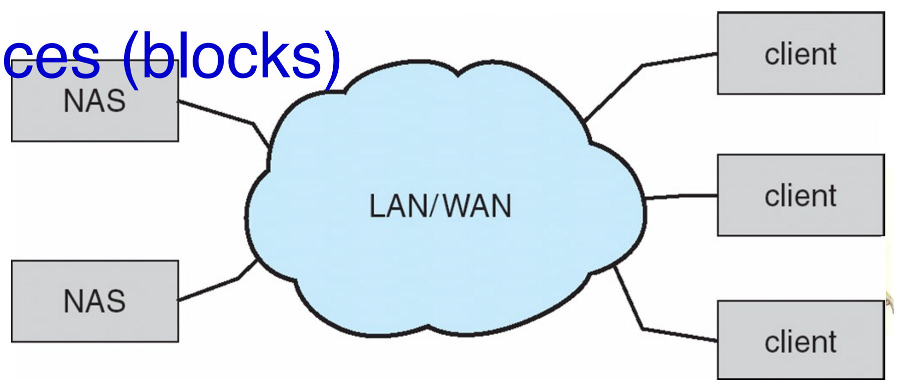
HDS

Storage Area Network (cont.)

- SAN is one or more storage arrays
  - Connected to one or more Fibre Channel switches
- Hosts also attach to switches
- Storage made available via **LUN Masking** from specific arrays to specific servers
- 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?
  - Consider iSCSI, FCOE

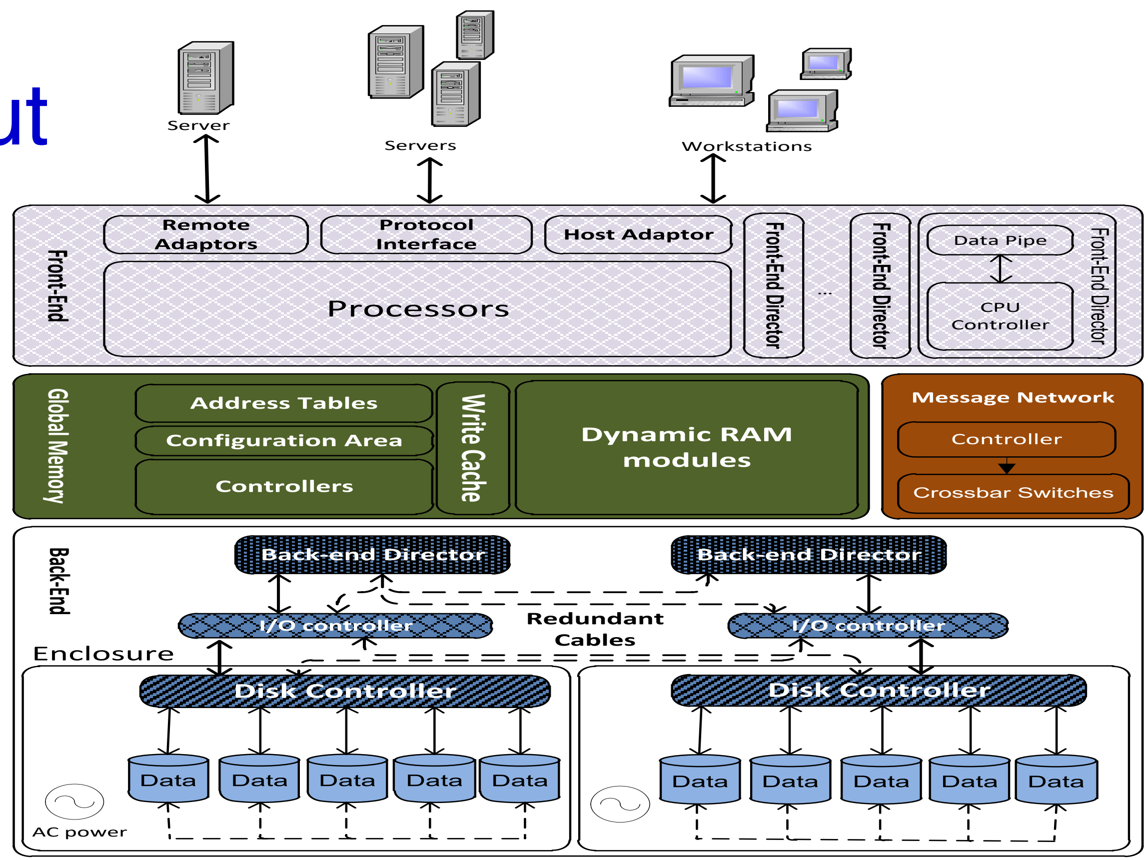
Network-Attached Storage

- Network-Attached Storage (**NAS**)
  - 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 via remote procedure calls (RPCs) between host and storage over typically TCP or UDP on IP network
- **iSCSI** protocol uses IP network to carry SCSI
  - Remotely attaching to devices (blocks)



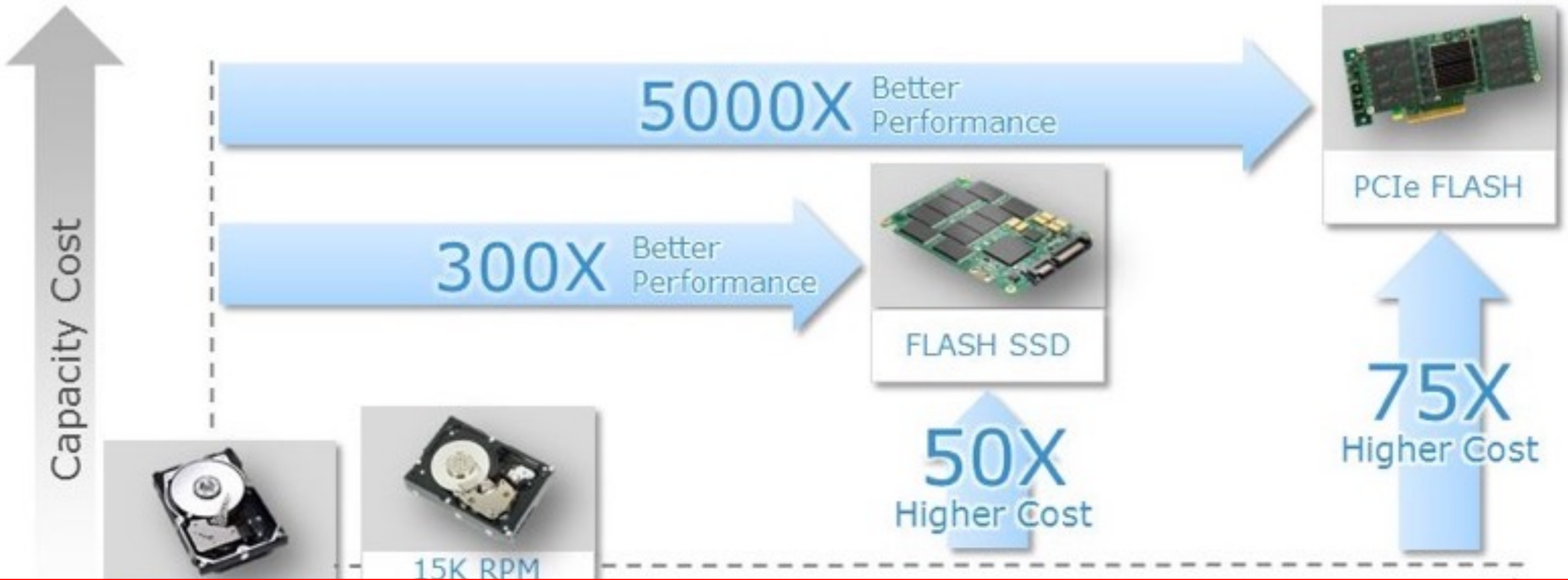
Why to use SAN instead of Local Drives?

- Easier & More Efficient Capacity Management
- High Availability
- High Reliability
- High Throughput
- Low Latency



- But More Expensive

Spectrum of Drives



HDD → Low Performance
SSD → High Cost

SSD → High Performance
HDD → Low Cost + Large Capacity

SSD + HDD → Feasible Cost and Performance

SSDs vs. HDDs

WD Black SATA HDD (4TB)

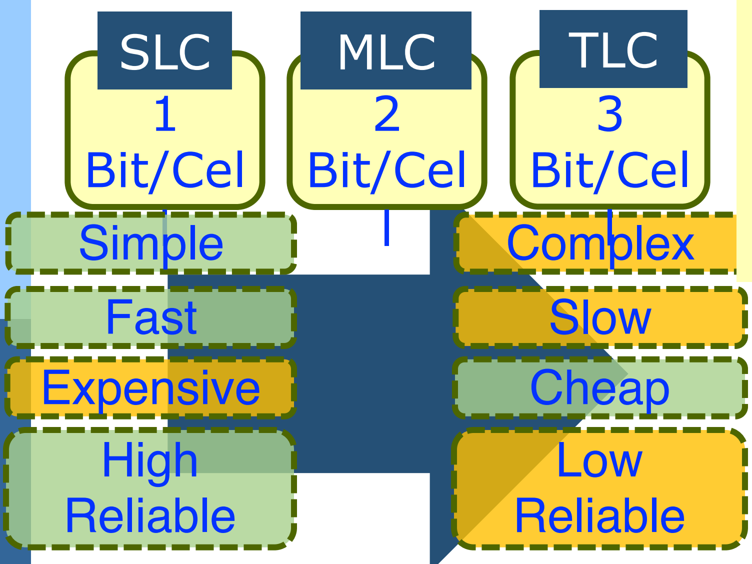
Slow (180 MB/s, IOPS: 140) ☹
Power Hungry (Active: 9w, Idle: 8w) ☹
Heavy (780g) ☹
Noisy ☹
MTTF < 1 Mh ☹
Inexpensive (0.05 $/G) ☺
$/IOPS (1.5 $) ☹
Unlimited Writes ☺

Samsung 863a SATA SSD (2TB)

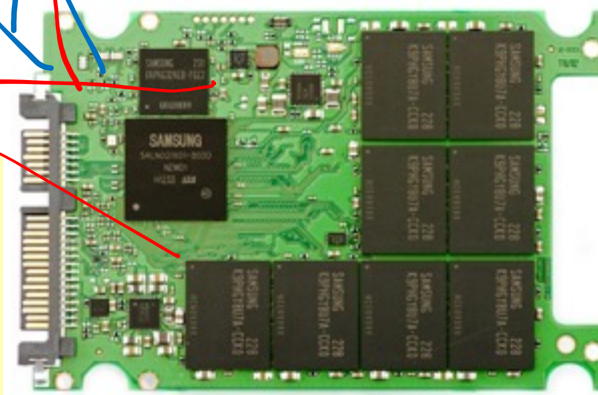
Fast (550 MB/s, IOPS > 90K) ☺
Low Power (Active: 5w, Idle: 60mw) ☺
Light (80g) ☺
Very Low Noise ☺
MTTF: 2 Mh ☺
Expensive (0.5 $/G) ☹
$/IOPS (0.1) ☺
Limited Writes ☹

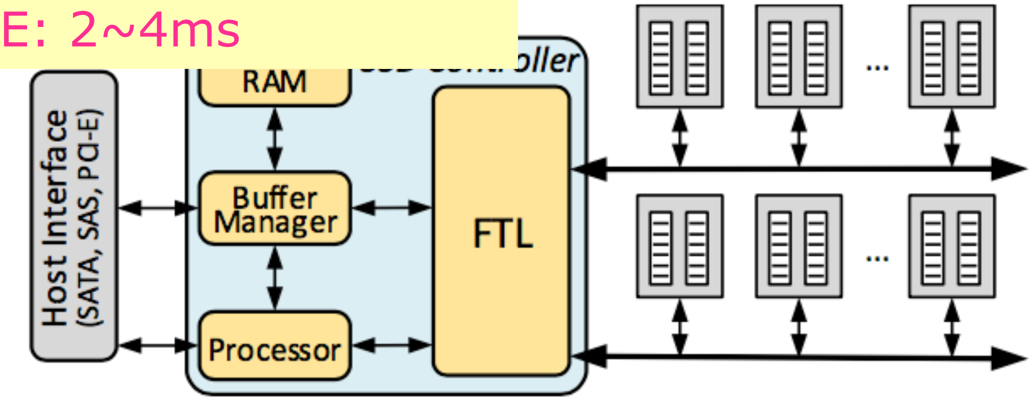
SSDs: Quick Introduction

- Flash-Based Solid State Drives
  - Non-volatile
  - Consist of NAND flashes



✓SLC
P: 200us,
E: 0.4~1ms
✓MLC
P: 0.3~2ms,
E: 2~4ms

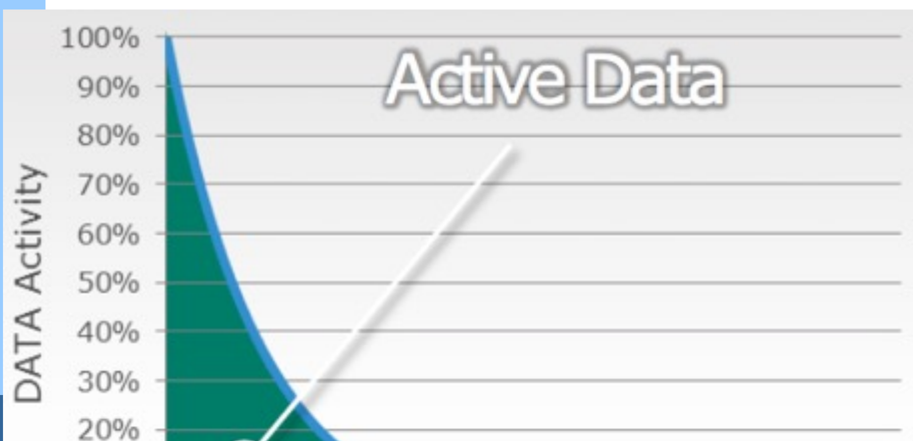


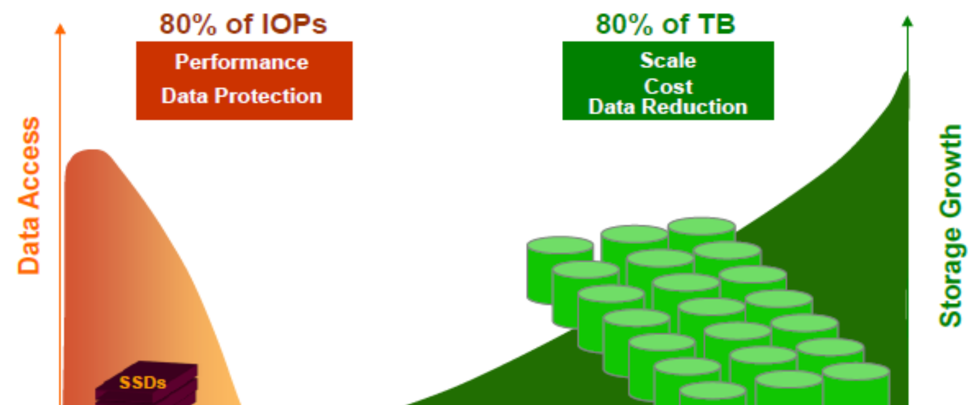


[1] Available: http://codecapsule.com/wp-content/uploads/2014/01/samsungssd840pro-02.jpg, Accessed: Sep. 2017.
[2] J. Kim, S. Seo, D. Jung, J.-S. Kim, and J. Huh, "Parameter-Aware I/O Management for Solid State Disks (SSDs)," IEEE TC, vol.61, no.5, pp.636–649, 2012.

Applications and Data Accesses

- Data is Highly Skewed
- At Any Given Time, Only a Small Percentage of Data is Active





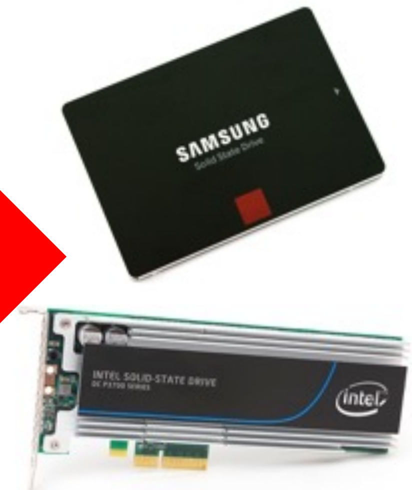
Small Size Hot Data → SSD
Large Size Cold Data → HDD

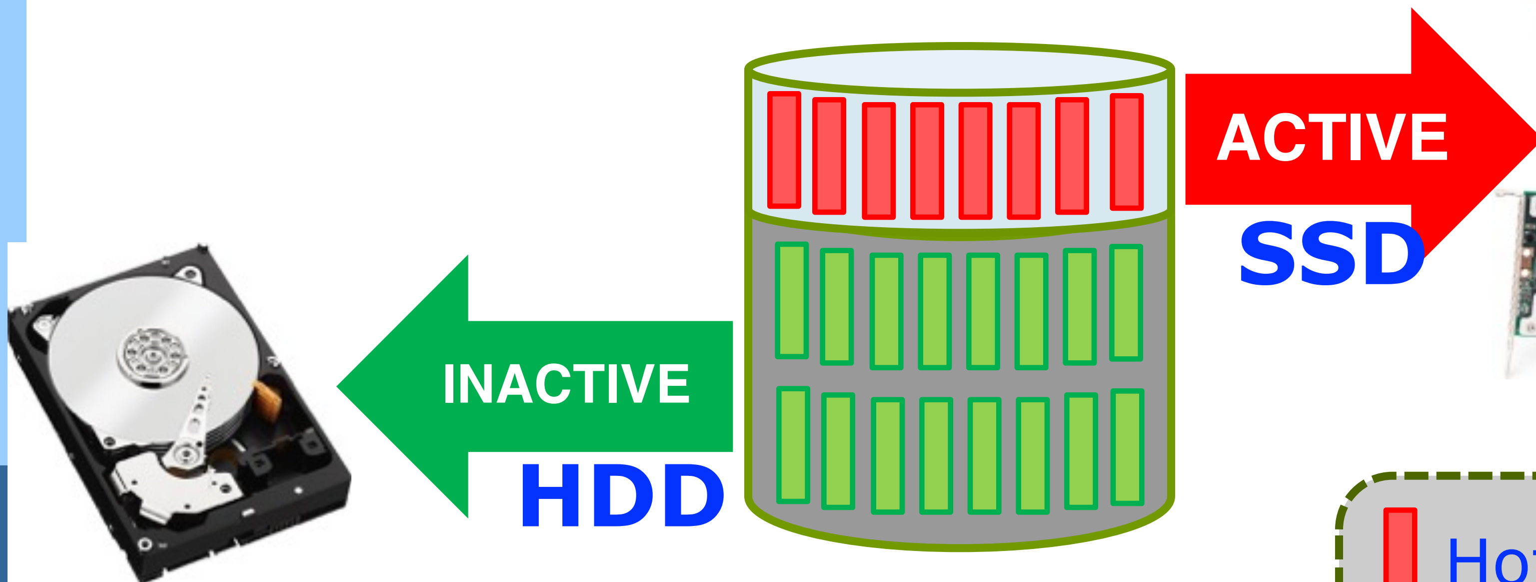
EMC Corp., 2015.
IMEX Research Cloud Infrastructure Report 2009-2011.

Hybrid Storages: I/O Caching

- ✓ Balance Workload
- ✓ Handle IO Bursts

Active Data on
High Performance SSD



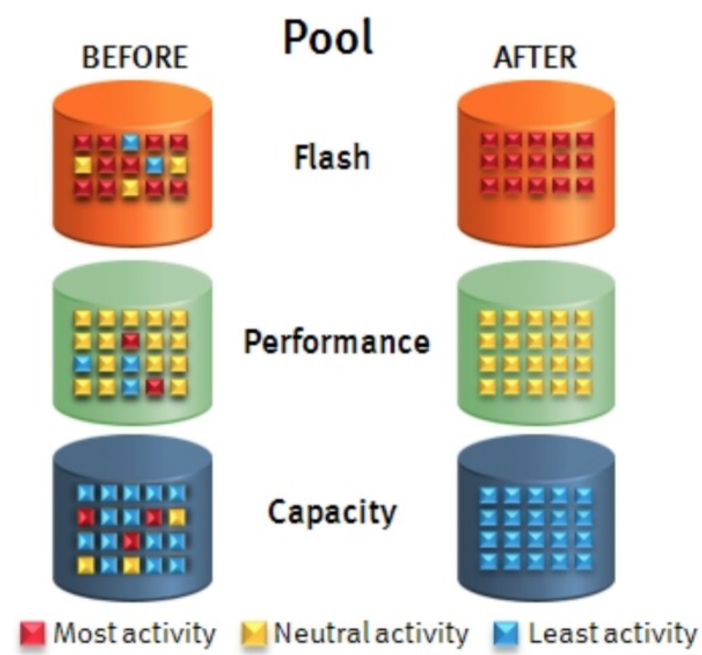


Inactive Data on
Low Cost HDD

Hot Data
Cold Data

Hybrid Storages: Tiering

- Move hot data to high-performance tier
- Move cold data to low-performance tier
- Common tiers (used in enterprise storages)
  - Extreme performance tier
    - Flash drives
  - Performance tier
    - SAS drives
  - Capacity tier
    - NL-SAS drives



Disk Scheduling

- OS is responsible for using HW Efficiently
  - For disk drives, this means having a fast access time and disk bandwidth
- → Minimize Seek Time
- Seek Time ≈ Seek Distance
- Disk **Bandwidth** is Total Number of Bytes Transferred, divided by Total Time between first Request for Service and Completion of 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 servicing of disk I/O requests
- 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

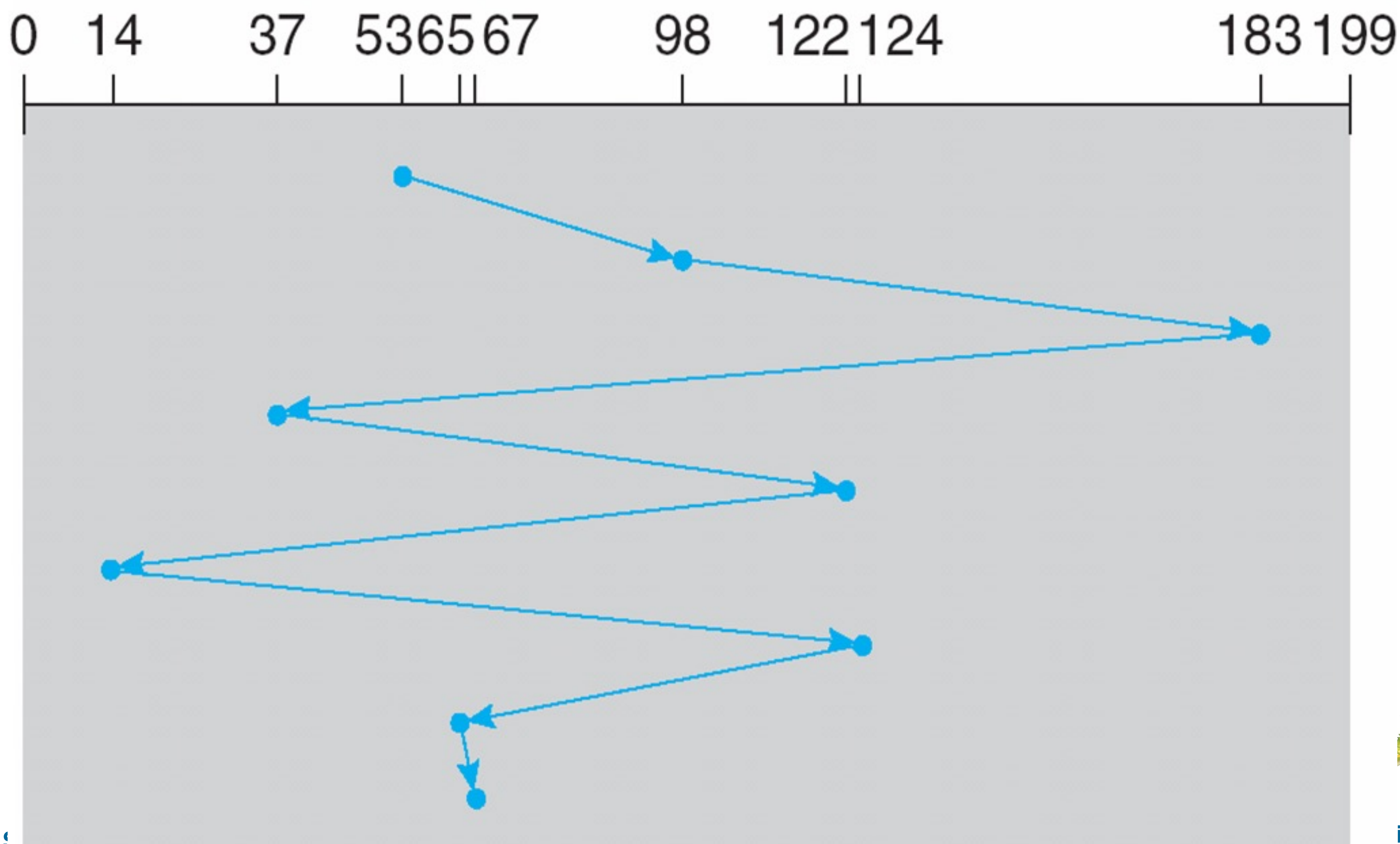
Head pointer 53

FCFS

Illustration shows total head movement of 640 cylinders

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

head starts at 53

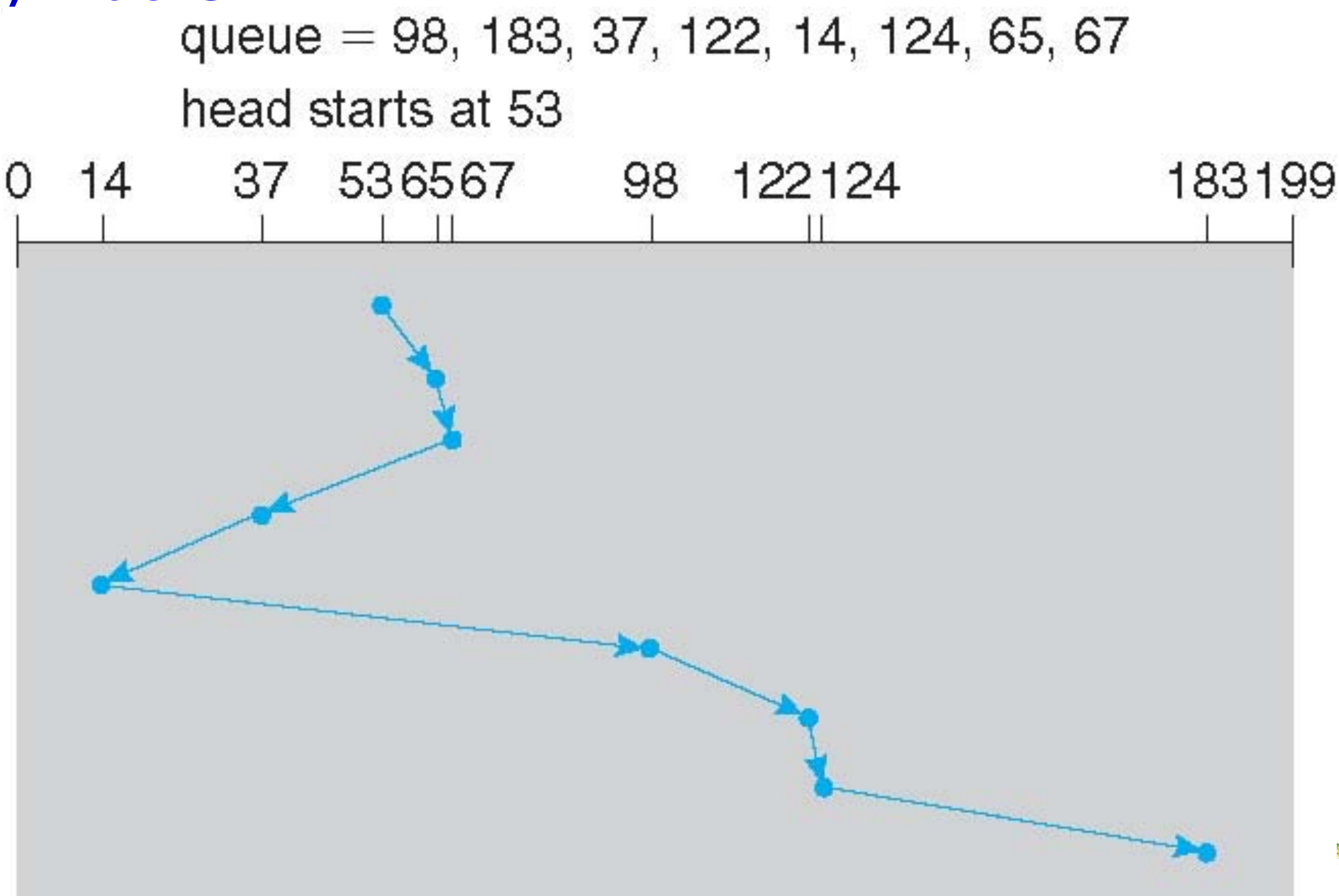


SSTF

- Shortest Seek Time First (SSTF) Scheduling
  - Selects request with minimum seek time from current head position
  - Is a form of SJF scheduling
  - May cause starvation of some requests
- Illustration shows total head movement of 236 cylinders

SSTF (cont.)

- Illustration shows total head movement of 236 cylinders



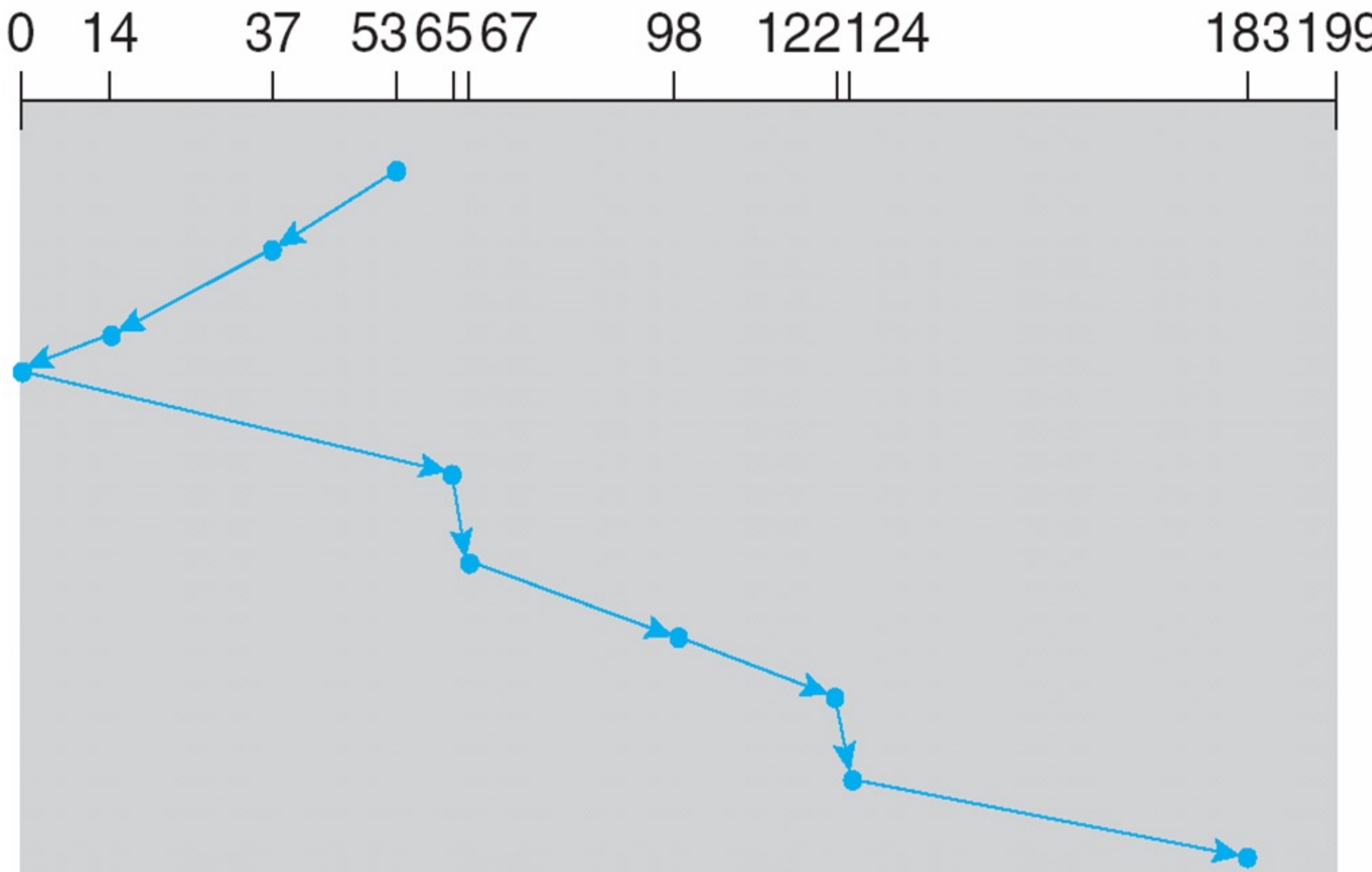
SCAN

- **SCAN Scheduling**
  - Disk arm starts at one end of 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 called **Elevator Algorithm**
  - Illustration shows total head movement of 208 cylinders
- **Note**
  - If requests are uniformly dense, largest density at other end of disk and those wait longest

SCAN (cont.)

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

head starts at 53



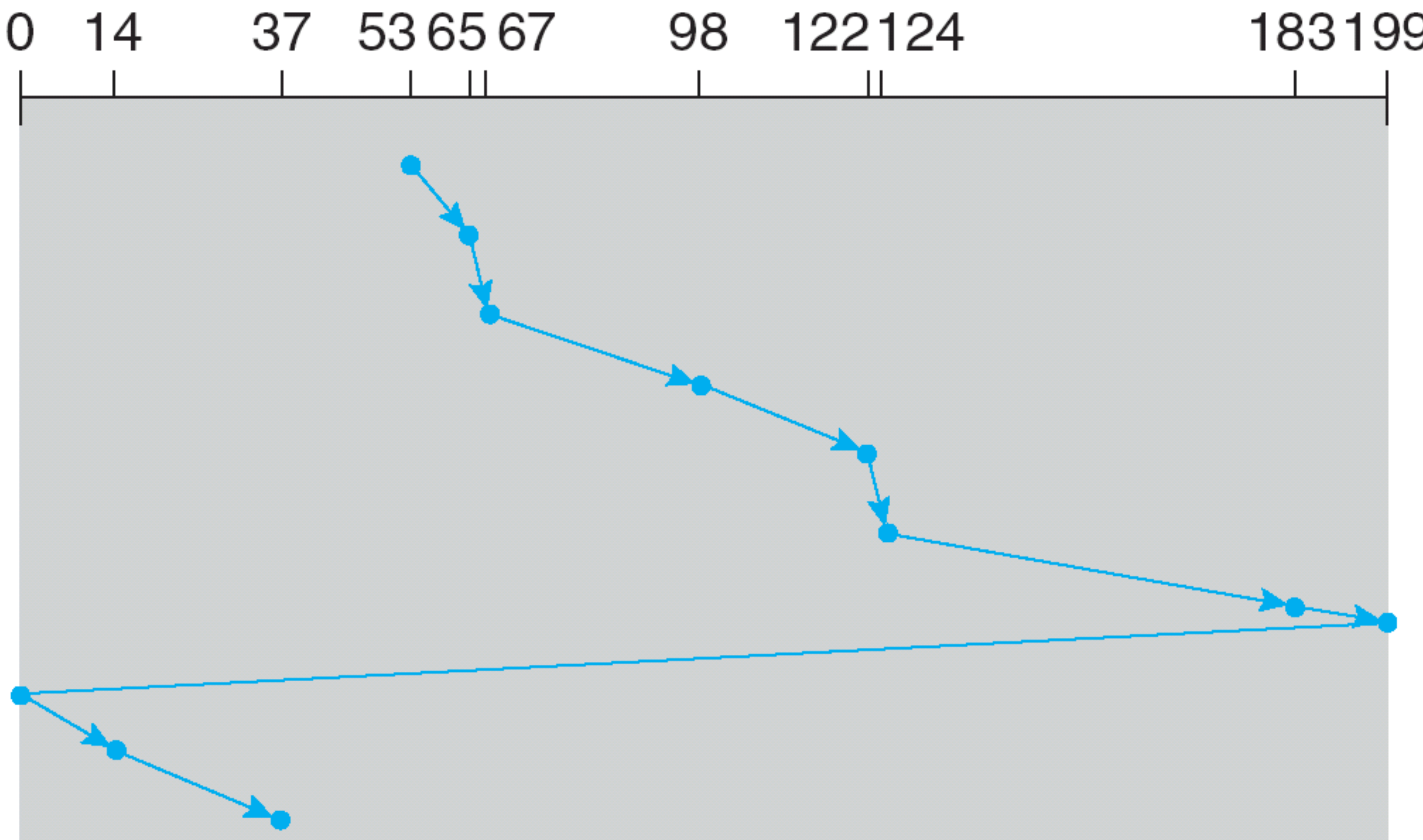
C-SCAN

- Circular SCAN (C-SCAN) Scheduling
  - Provides a more uniform wait time than SCAN
  - Head moves from one end of 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 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

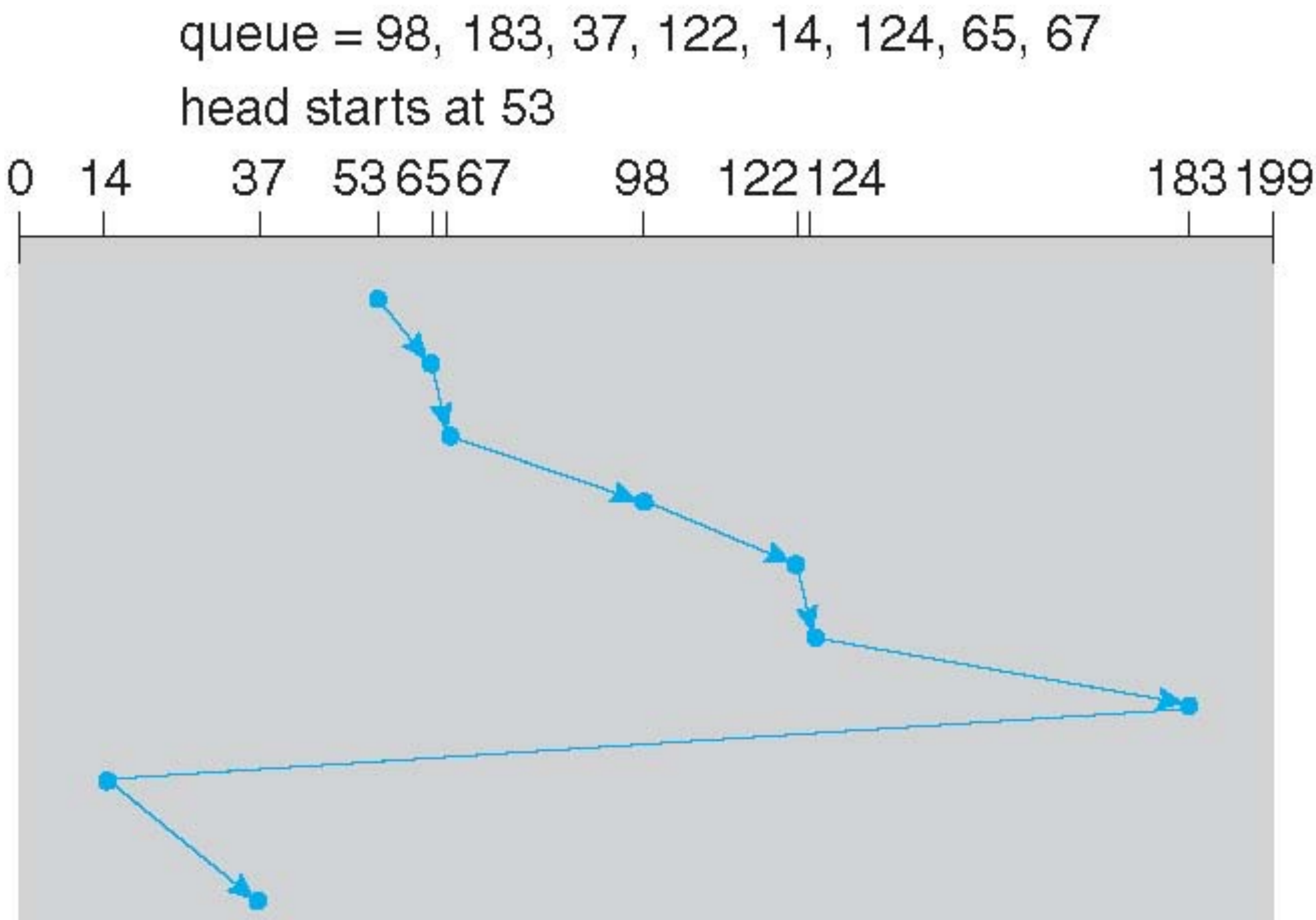
head starts at 53



C-LOOK

- LOOK a version of SCAN
- C-LOOK a version of C-SCAN
  - Arm only goes as far as last request in each direction, then reverses direction immediately
    - Without first going all the way to end of 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 disk
  - Less starvation
- Performance depends on number and types of requests
- Requests for disk service can be influenced by file-allocation method
  - And metadata layout

Selecting a Disk-Scheduling Algorithm (cont.)

- Disk-Scheduling algorithm should be written as a separate module of OS, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for default algorithm
- What about Rotational Latency?
  - Difficult for OS to calculate
- How does disk-based queuing effect OS queue ordering efforts?

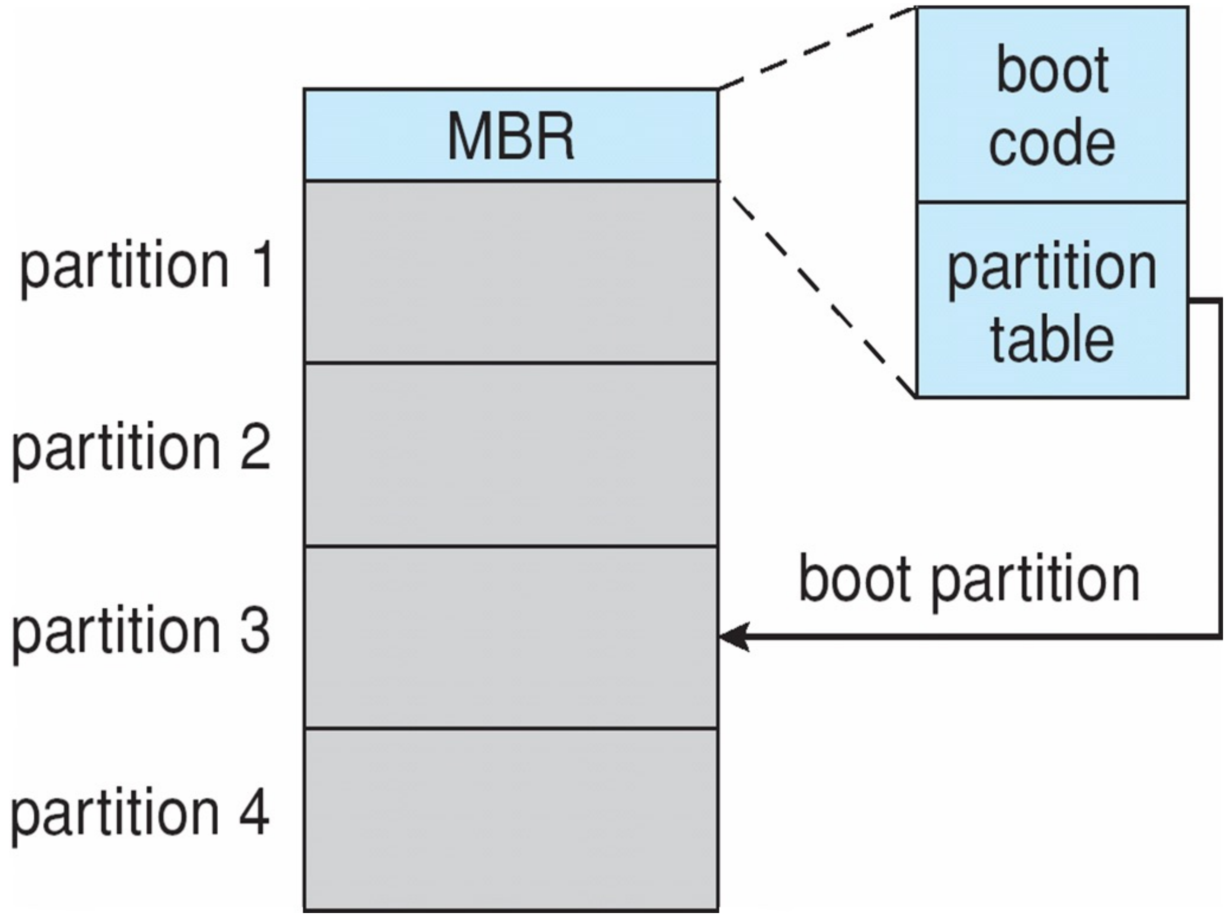
Disk Management

- **Low-level Formatting**, or **Physical Formatting** — Dividing a disk into sectors that disk controller can read and write
  - Each sector can hold header information, plus data, plus error correction code (**ECC**)
  - Usually 512 bytes of data but can be selectable
- To use a disk to hold files, OS still needs to record its own data structures on disk
  - **Partition** disk into one or more groups of cylinders, each treated as a logical disk
  - **Logical formatting** or "making a file system"
  - To increase efficiency most file systems group blocks into **clusters**
    - Disk I/O done in blocks
    - File I/O done in clusters

Disk Management (cont.)

- Raw Disk Access for apps that want to do their own block management, keep OS out of the way (databases for example)
- Boot Block Initializes System
  - Bootstrap is stored in ROM
  - **Bootstrap loader** program stored in boot blocks of boot partition
- Methods such as **Sector Sparing** used to handle bad blocks

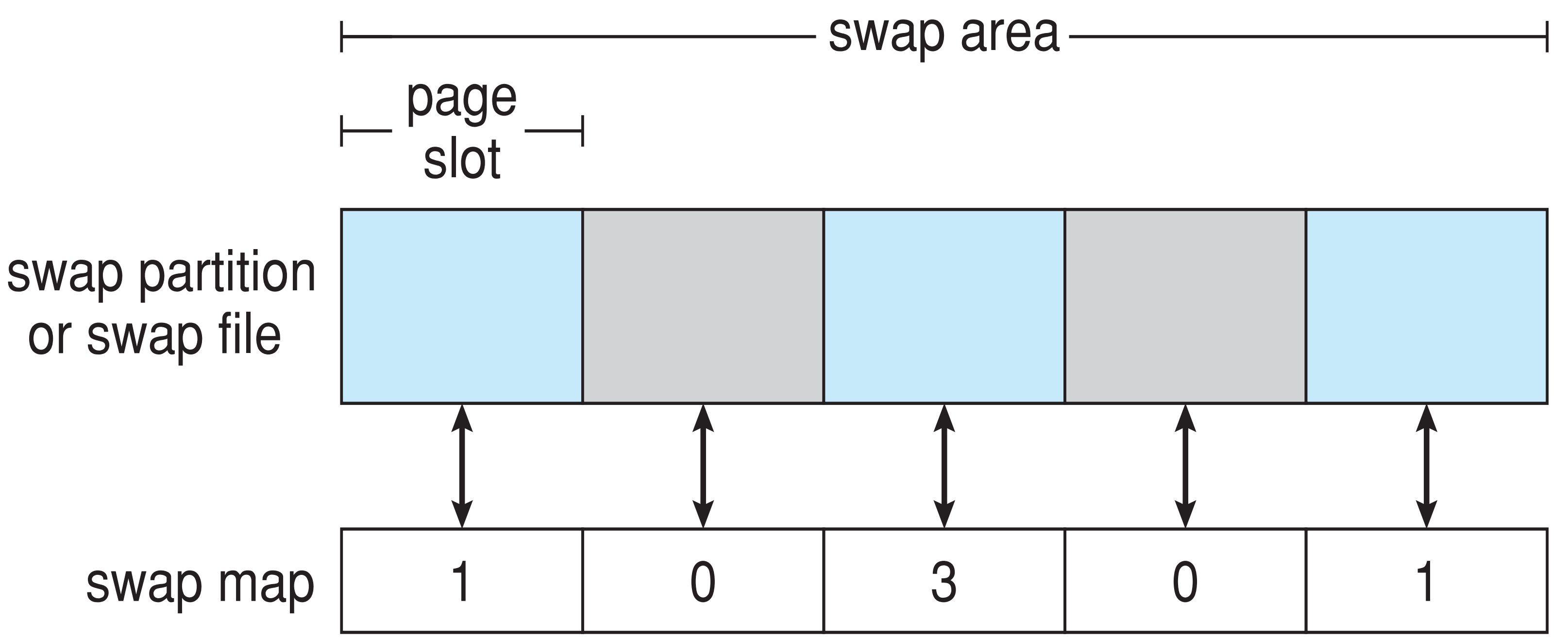
Booting from a Disk in Windows



Swap-Space Management

- Swap-Space — Virtual memory uses disk space as an extension of main memory
  - Less common now due to memory capacity increases
- Swap-space can be carved out of normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-Space Management
  - 4.3BSD allocates swap space when process starts; holds text segment (program) and data segment
  - Kernel uses **swap maps** to track swap-space use

Data Structures for Swapping on Linux Systems



RAID Structure

- Redundant Array of Inexpensive Disks
  - Multiple disk drives provides reliability via **redundancy**
- Increases **Mean Time To Failure**
- **Mean Time To Repair**
  - Exposure time when another failure could cause data loss
- **Mean Time To Data Loss**
  - Based on above factors

RAID

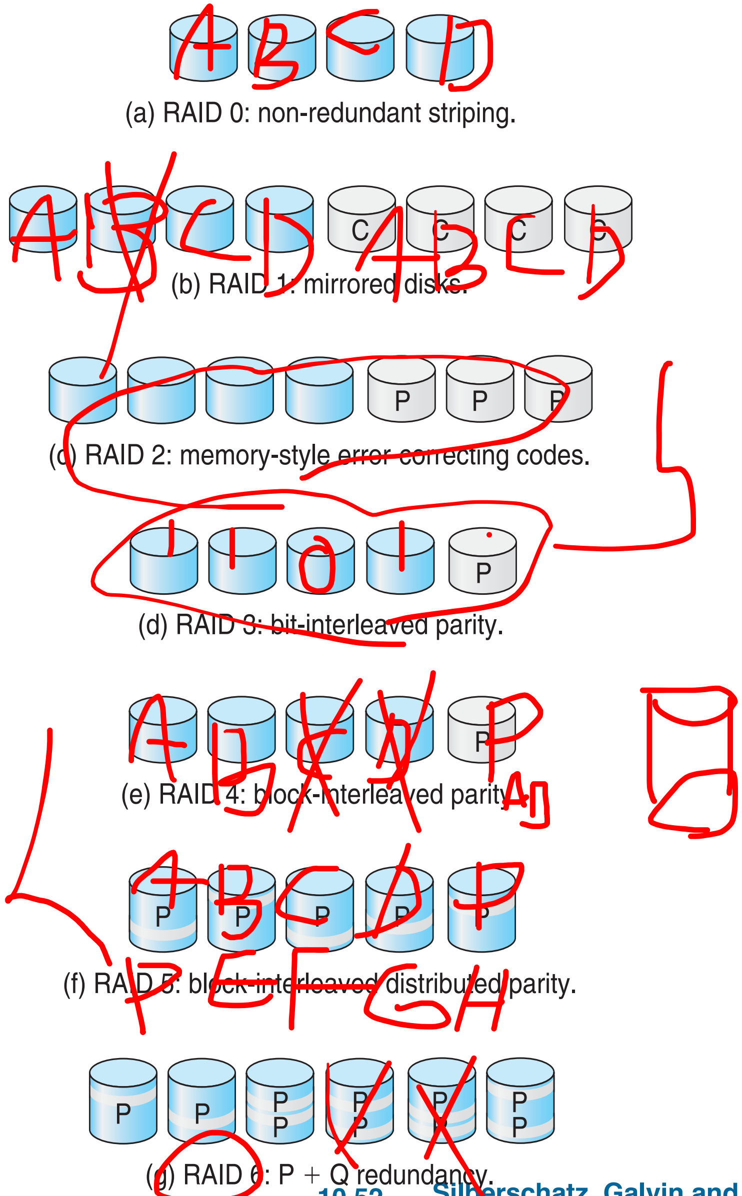
- Disk **Striping** uses a Group of Disks as one Storage Unit
- Arranged into Six Different Levels
- Improves Performance and Reliability of Storage System by Storing Redundant Data
  - **Mirroring** or **shadowing** (**RAID 1**) keeps duplicate of each disk
  - Striped mirrors (**RAID 1+0**) or mirrored stripes (**RAID 0+1**) provides high performance and high reliability

RAID (cont.)

- **Block interleaved parity** (**RAID 4, 5, 6**) uses much less redundancy

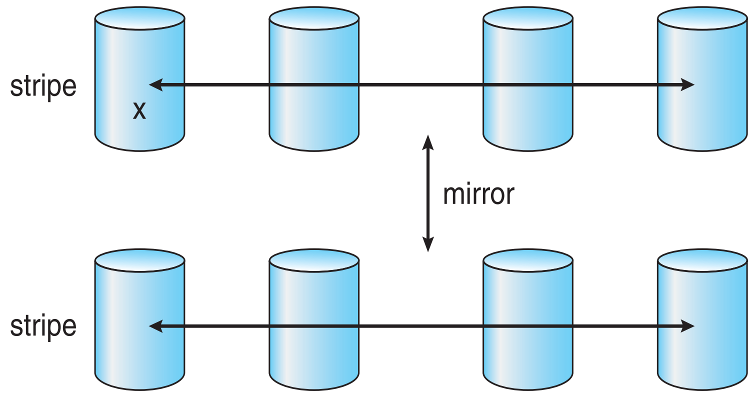
- RAID within a storage array can still fail if the array fails, so automatic **replication** of data between arrays is common
- Frequently, a small number of **hot-spare** disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them

RAID Levels

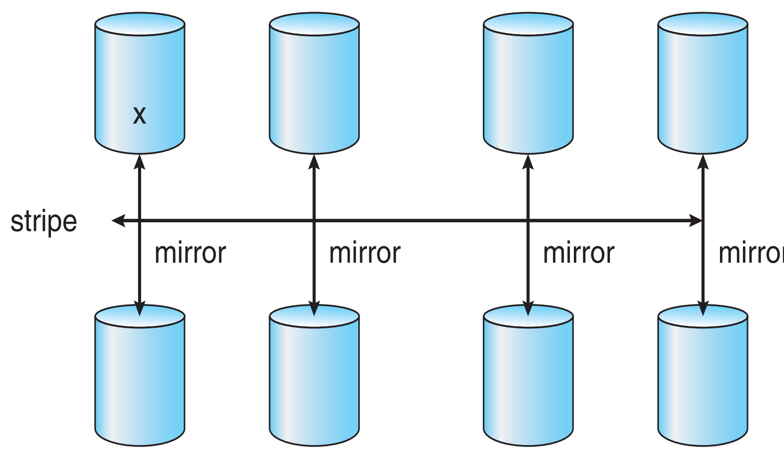


C
C
C
C
(a) RAID 0: non-redundant striping.
(b) RAID 1: mirrored disks.
(c) RAID 2: memory-style error-correcting codes.
(d) RAID 3: bit-interleaved parity.
(e) RAID 4: block-interleaved parity.
(f) RAID 5: block-interleaved distributed parity.
(g) RAID 6: P + Q redundancy.

RAID (0 + 1) and (1 + 0)

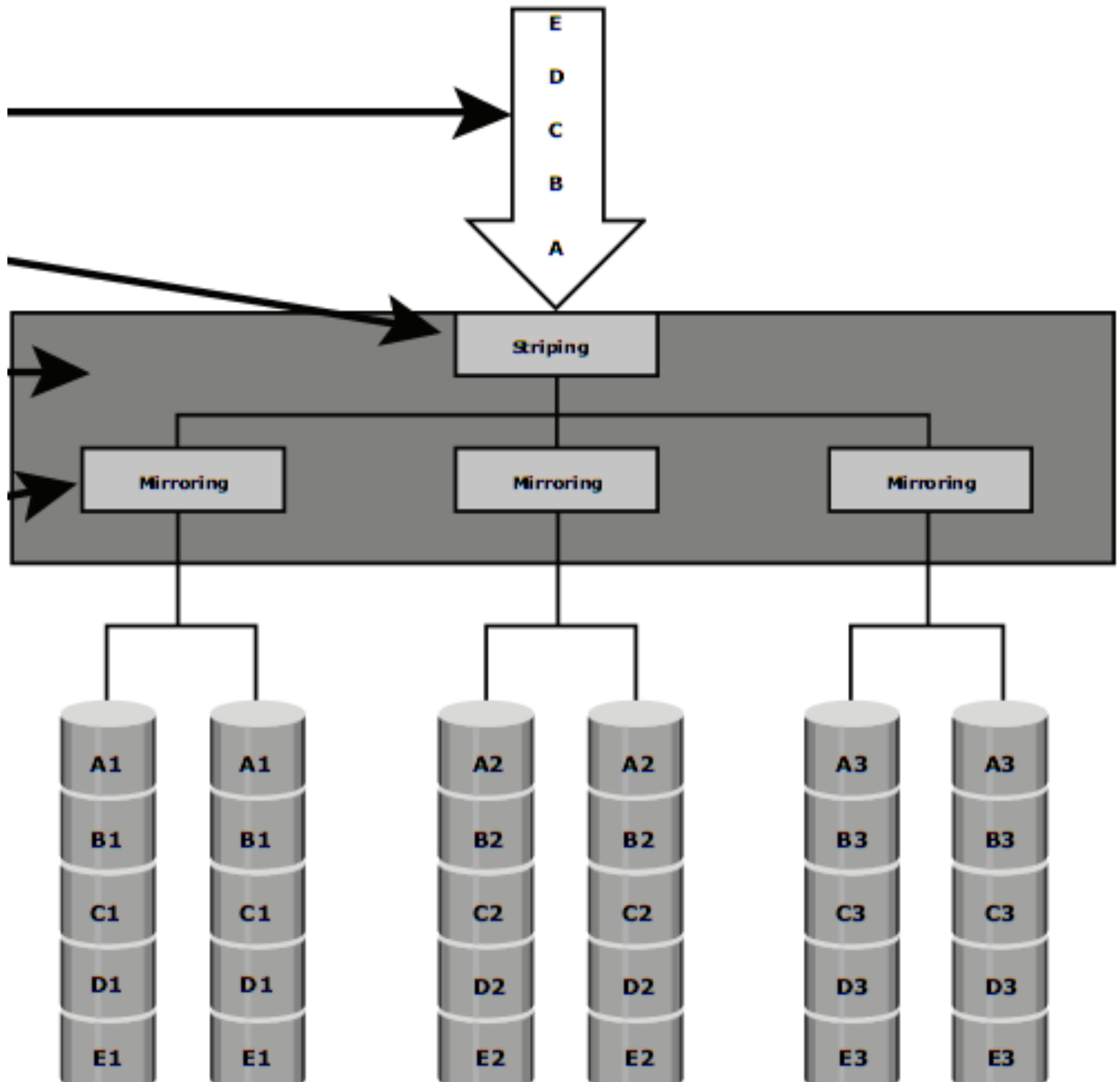


a) RAID 0 + 1 with a single disk failure.



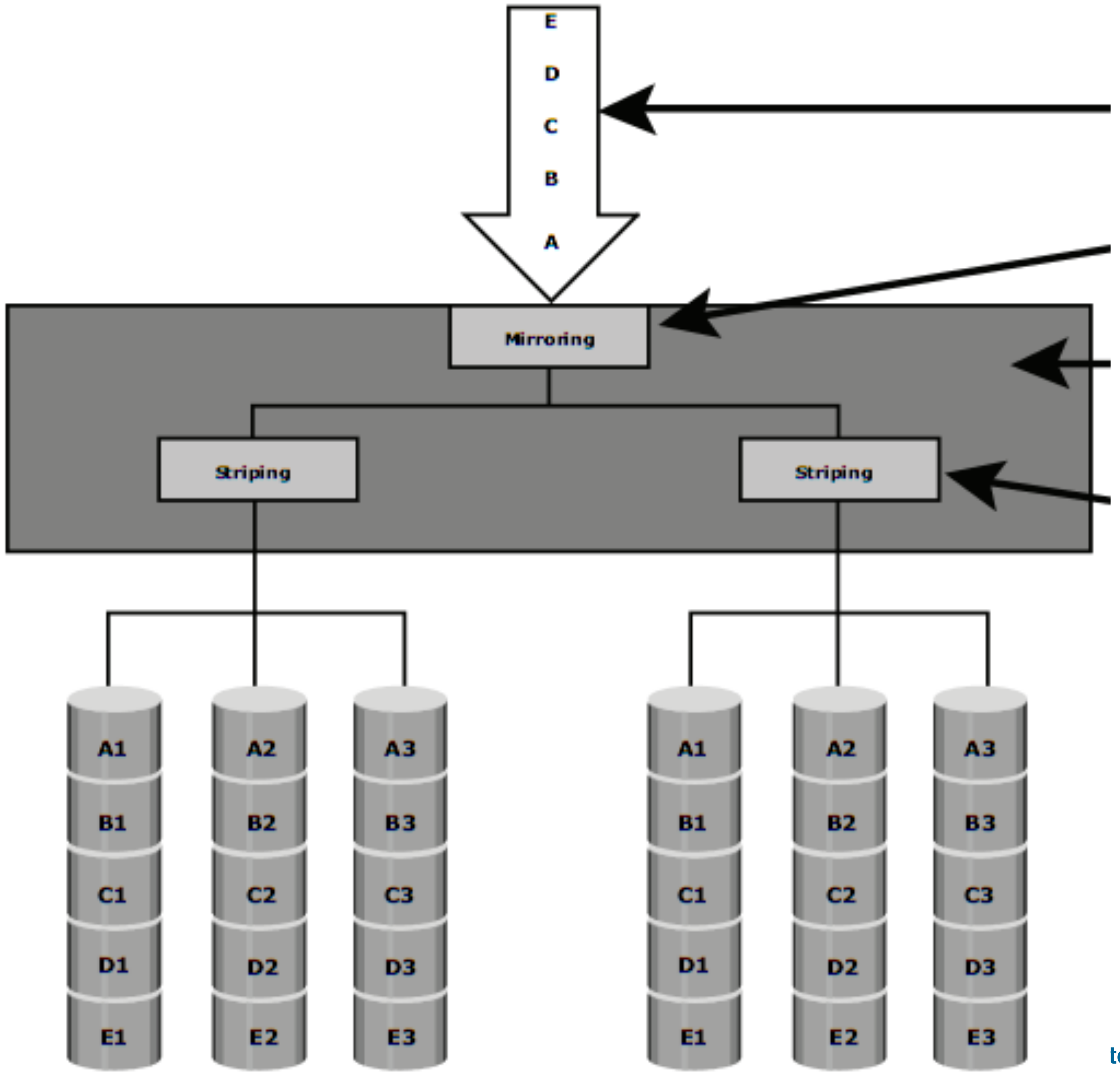
b) RAID 1 + 0 with a single disk failure.

RAID 0+1



E
D
C
B
A
Striping
Mirroring
Mirroring
Mirroring
A1
B1
C1
D1
E1
A1
B1
C1
D1
E1
A2
B2
C2
D2
E2
A2
B2
C2
D2
E2
A3
B3
C3
D3
E3
A3
B3
C3
D3
E3

RAID 1+0



E
D
C
B
A
Mirroring
Striping
Striping
A1
B1
C1
D1
E1
A2
B2
C2
D2
E2
A3
B3
C3
D3
E3
A1
B1
C1
D1
E1
A2
B2
C2
D2
E2
A3
B3
C3
D3
E3

Other Features

- Regardless of where RAID implemented, other useful features can be added
- **Snapshot** is a view of file system before a set of changes take place (i.e. at a point in time)
- Replication is automatic duplication of writes between separate sites
  - For redundancy and disaster recovery
  - Can be synchronous or asynchronous
- Hot spare disk is unused, automatically used by RAID production if a disk fails to replace the failed disk and rebuild the RAID set if possible
  - Decreases mean time to repair

End of Lecture 10