Mass-Storage Structure

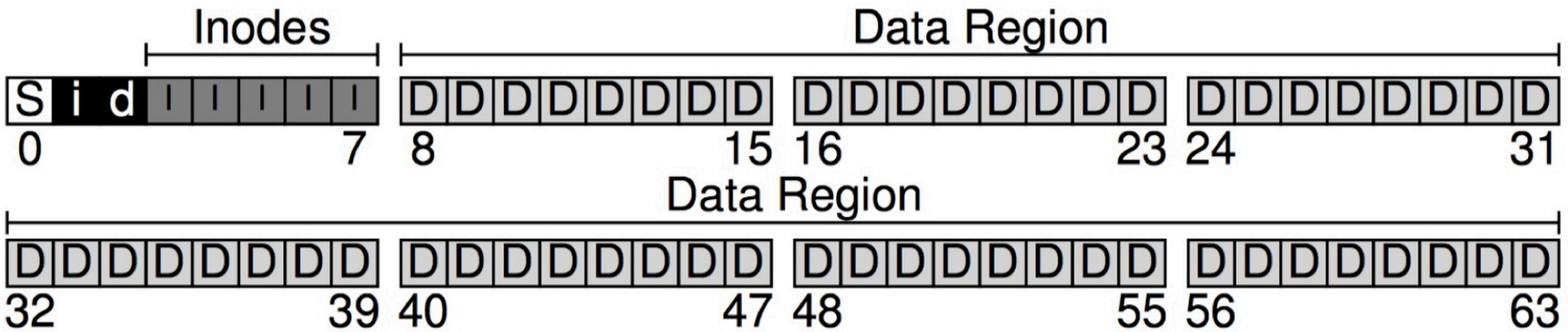
Yajin Zhou (http://yajin.org)

Zhejiang University

Acknowledgement: some pages are based on the slides from Zhi Wang(fsu).

Review

- On-disk structure about FS, how FS maps the calls to the structure
- Data block, inode tables, imap, dmap, super block



- Read a file /foo/bar -> read root inode, root data, foo inode, foo data, bar inode, bar data, write bar inode (Why?)
- Create and write /boo/bar-> read root inode, read root data, read foo inode, read foo data, read inode bitmap, write inode map, write foo data, read bar inode, write bar inode, write foo inode
- Caching and buffering

Content

- Overview of Mass Storage Structure
- Disk Structure
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure

Overview

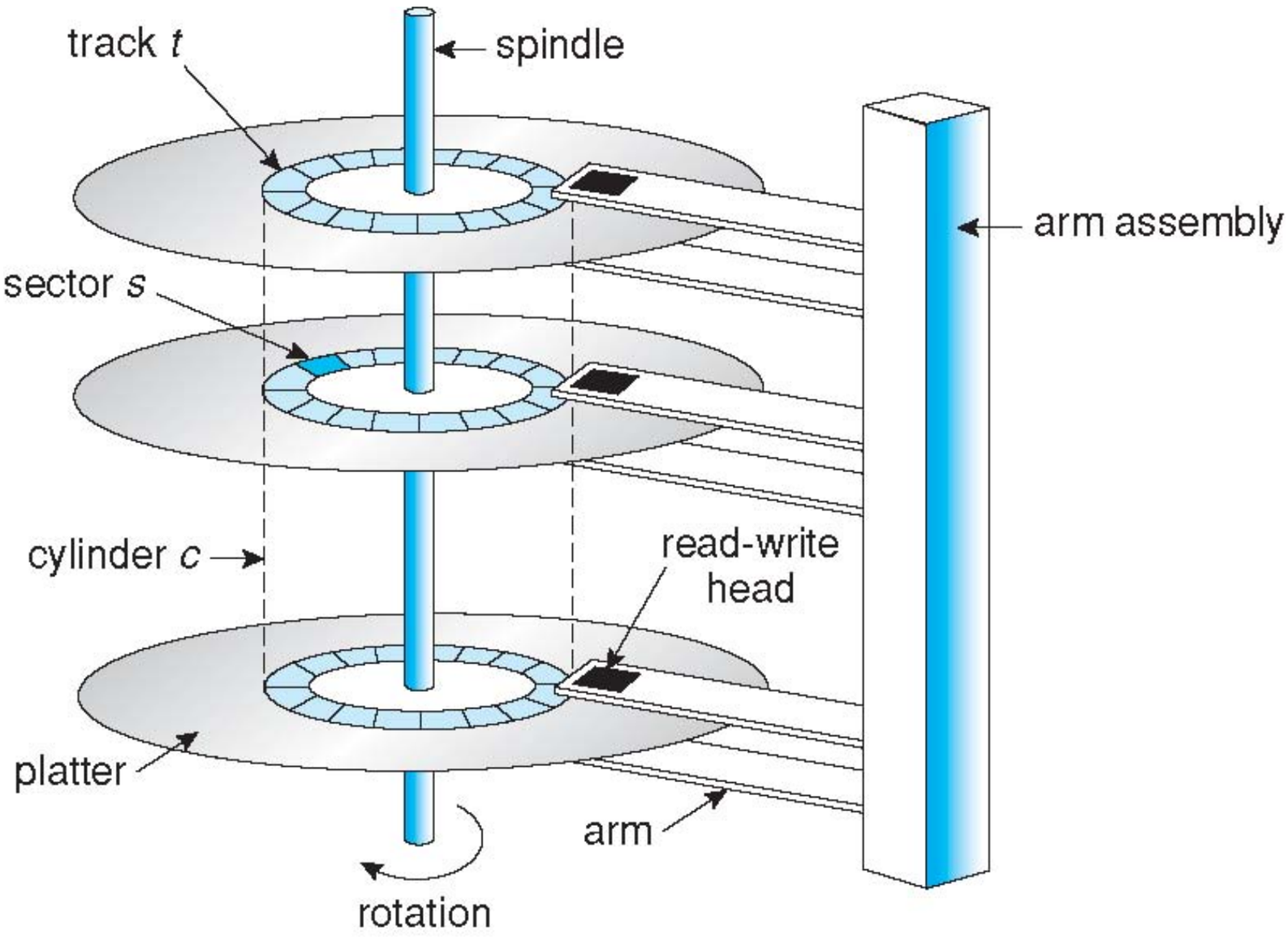
- Magnetic disks provide bulk of secondary storage of computer system
  - **hard disk** is most popular; some magnetic disks could be removable
    - driver attached to computer via I/O buses (e.g., USB, SCSI, EIDE, SATA…)
    - drives rotate at 60 to 250 times per second (7000rpm = **117rps**)
- Magnetic disks has platters, range from .85” to 14” (historically)
  - 3.5”, 2.5”, and 1.8” are common nowadays
- Capacity ranges from 30GB to 3TB per drive, and even bigger

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

Moving-head Magnetic Disk



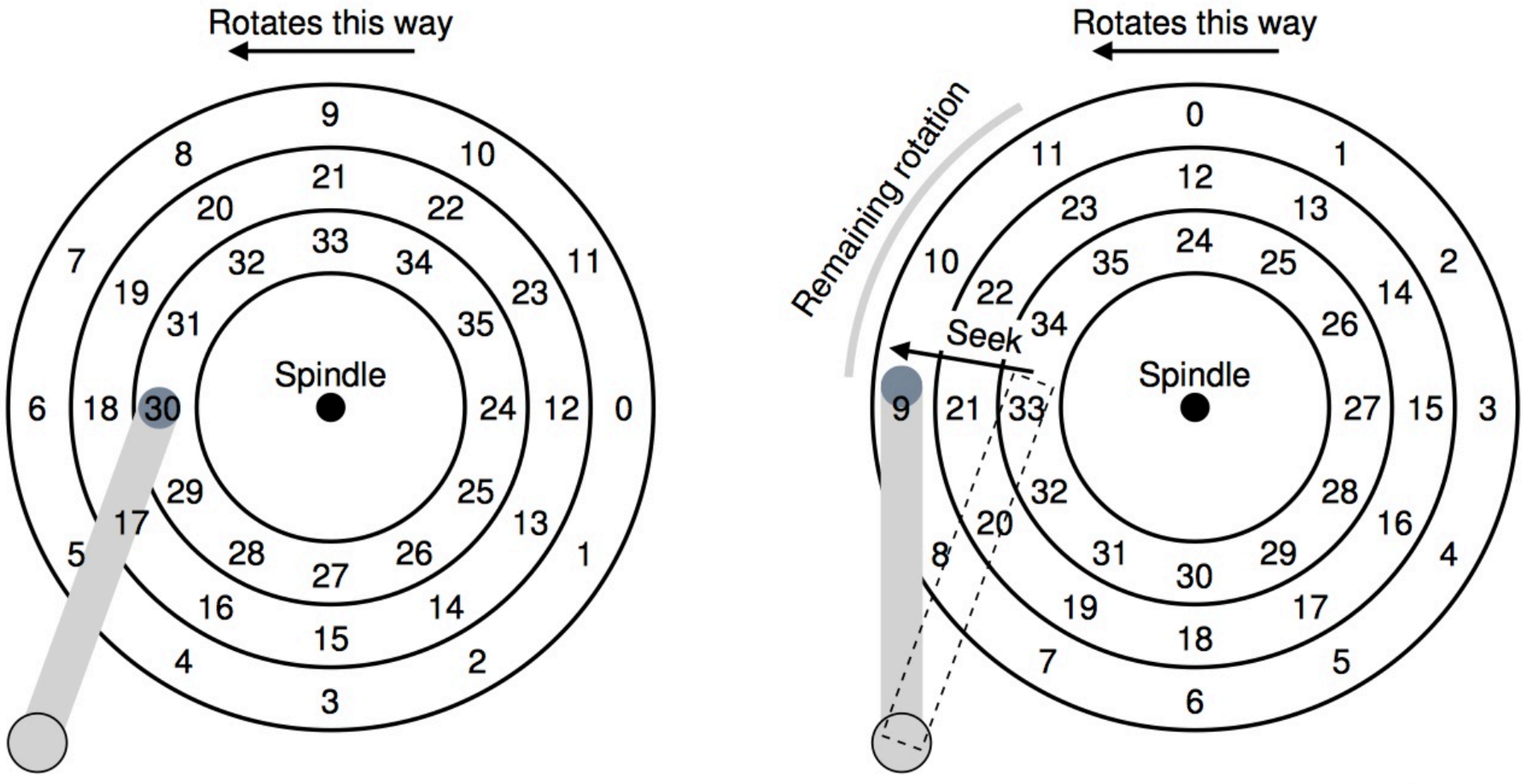


Figure 37.3: **Three Tracks Plus A Head (Right: With Seek)**

Magnetic Disk

- **Positioning time** is time to move disk arm to desired sector
  - positioning time includes **seek time** and **rotational latency**
    - seek time: move disk to the target cylinder
    - rotational latency: for the target sector to rotate under the disk head
  - positioning time is also called random-access time
- **Performance**
  - **transfer rate**: **theoretical** 6 Gb/sec; **effective** (real) about 1Gb/sec
    - Transfer rate is rate at which data flow between drive and computer
  - **seek time** from 3ms to 12ms (9ms common for desktop drives)
  - latency based on spindle speed: 1/rpm * 60
    - average latency = ½ latency

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

Magnetic Disk

- **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 + (data to transfer / transfer rate) + controller overhead
  - e.g., to transfer a 4KB block on a 7200 RPM disk; 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:

5ms + 4.17ms + 4KB / 1Gb/sec + 0.1ms = 9.39ms (4.17 is average latency)

Nonvolatile Memory Devices

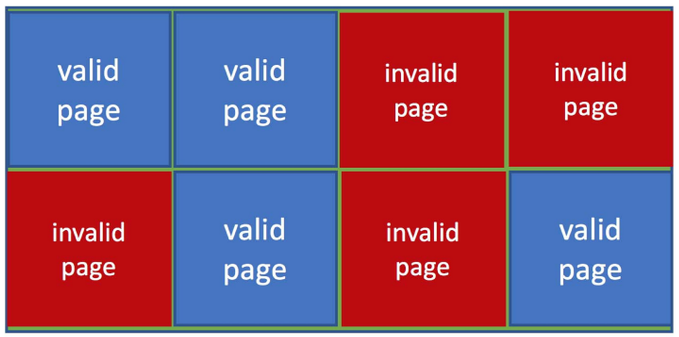
- If disk-drive like, then called solid-state disks (SSDs)
- Other forms include USB drives, and main storage in devices like smartphones
- Can be **more reliable** than HDDs
- More expensive per MB
- Maybe have shorter life span – need careful management
- Less capacity, but much faster
- Busses can be too slow -> connect directly to PCI for example
- No moving parts, so no seek time or rotational latency

Nonvolatile Memory Devices

- Have characteristics that present challenges
  - **Read and written in "page"** increments (think sector) but can't **overwrite** in place
    - Must first be **erased**, and erases happen in larger "**block**" increments
    - Block size: 64 128 256K, page size: 512, 2k or 4k
    - Can only be erased a limited number of times before worn out – ~ 100,000
    - Life span measured in **drive writes per day (DWPD)**
      - A 1TB NAND drive with rating of 5DWPD is expected to have 5TB per day written within warrantee period without failing

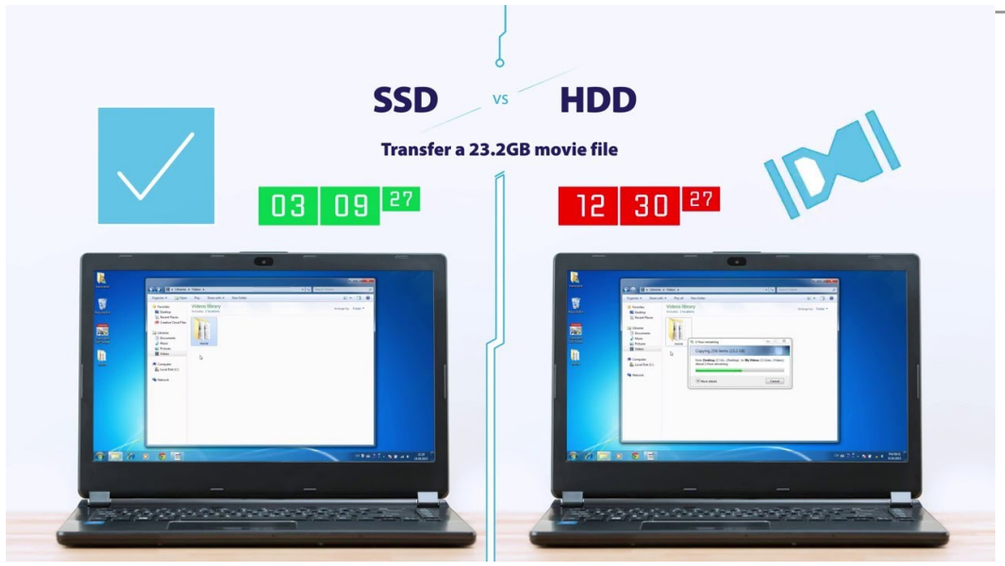
NAND Flash Controller Algorithms

- With no overwrite, pages end up with mix of valid and invalid data
- To track which logical blocks are valid, controller maintains **flash translation layer (FTL) table**
- Also implements **garbage collection** to free invalid page space - pages available but no free blocks
- Allocates over-provisioning to provide working space for GC
  - Copy good data to over-provisioning area, and erase the block for later use
- Each cell has lifespan, so wear leveling needed to write equally to all cells



NAND block with valid and invalid pages

Performance



京东精选 三星(SAMSUNG) 860 EVO 1TB 2.5英寸 SATAIII 固态硬盘（MZ-76E1T0B）

玩转速度 新一代V-NAND技术 性能强劲 智能兼容 多种接口选择

京 东 价 ￥1299.00 降价通知

促　　销 限购 购买1-10件时享受单件价￥1299，超出数量以结算价为准

增值业务 以旧换新，卖了换钱

配 送 至 浙江杭州市西湖区城区 有货 支持 京尊达 | 99元免基础运费(20kg内) | 次日达 | 自提

由 京东 发货, 供应商提供售后服务. 23:00前下单,预计**明天(01月03日)**送达

重　　量 0.08kg

选择颜色 860 EVO | 860 PRO

选择版本 SATA-3 | M.2 | MSATA

容量 250-256G | 500-512G | 1TB | 2TB | 4TB

西部数据(WD)蓝盘 1TB 5400转128M SATA6Gb/s 笔记本硬盘(WD10SPZX)

开机慢、打开软件时间长、运行不流畅？ >>> 你该换块新的笔记本硬盘了！

京 东 价 ￥329.00 降价通知

增值业务 以旧换新，卖了换钱 | 礼品包装

配 送 至 浙江杭州市西湖区城区 有货 支持 99元免基础运费(20kg内) | 次日达 | 自提

由 京东 发货, 并提供售后服务. 23:00前下单,预计**明天(01月03日)**送达

重　　量 0.14kg

选择颜色 游戏领地"黑盘" | 日常存储"蓝盘" | 监控领域"AV系列"

选择版本 【500G】 | 【1TB】 | 【2TB】 | 【15毫米厚2TB】

增值保障 全保换2年 ¥29 | 延长保2年 ¥15 | 上门安装 ¥99

Magnetic Tape

- Tape was early type of secondary storage, now mostly for backup
  - large capacity: 200GB to 1.5 TB
  - slow access time, especially for random access
    - seek time is much higher than disks
    - once data under head, transfer rates comparable to disk (140 MB/s)
    - need to wind/rewind tape for random access
- data stored on the tape are relatively permanent

Disk Structure

- Disk drives are addressed as a 1-dimensional arrays of logical blocks (LBA)
  - logical block is the smallest unit of transfer
- Logical blocks are mapped into **sectors** of the disk sequentially
  - sector 0 is the first sector of the first track on the outermost cylinder
  - mapping proceeds in order
    - first through that **track**
    - then the rest of the tracks in that **cylinder**
    - then through the rest of the cylinders from outermost to innermost
- logical to physical address should be easy
  - except for bad sectors

Disk Attachment

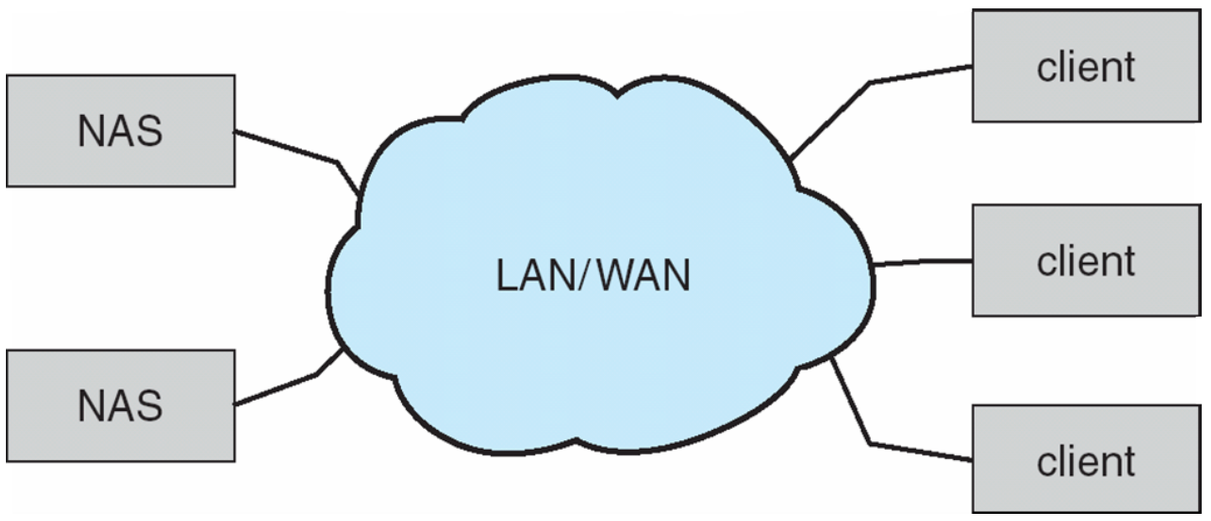
- Disks can be attached to the computer as:
  - **host-attached** storage
    - hard disk, RAID arrays, CD, DVD, tape…
  - **network-attached** storage
  - **storage area network**

Host-Attached Storage

- Disks can be attached to the computers directly via an **I/O bus**
  - e.g., SCSI is a bus architecture, up to 16 devices on one cable,
    - SCSI initiator requests operations; SCSI targets(e.g., disk) perform tasks
    - each target can have up to 8 logical units
- e.g., Fiber Channel is high-speed serial bus
  - can be switched fabric with 24-bit address space
  - most common storage area networks (SANs) interconnection

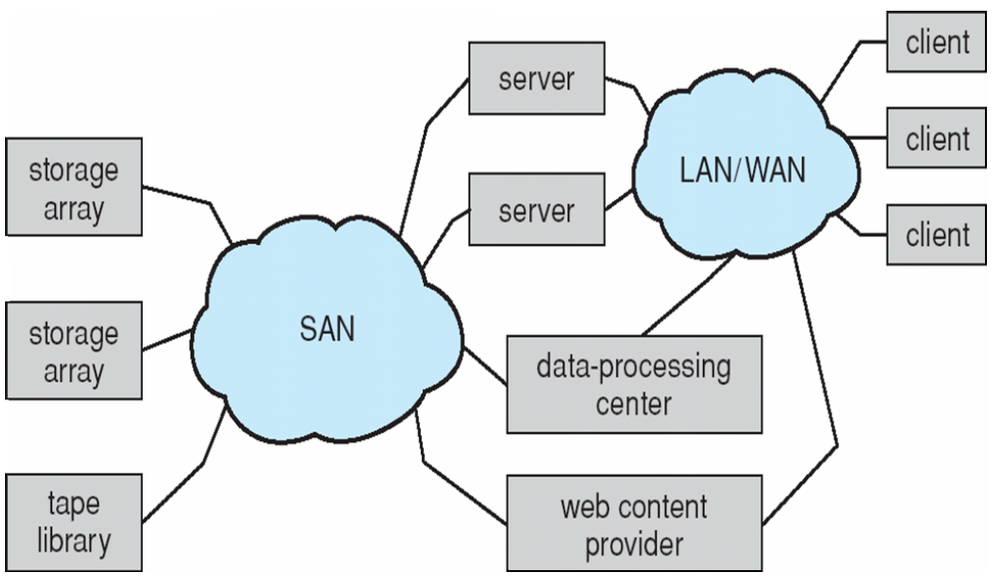
Network-Attached Storage

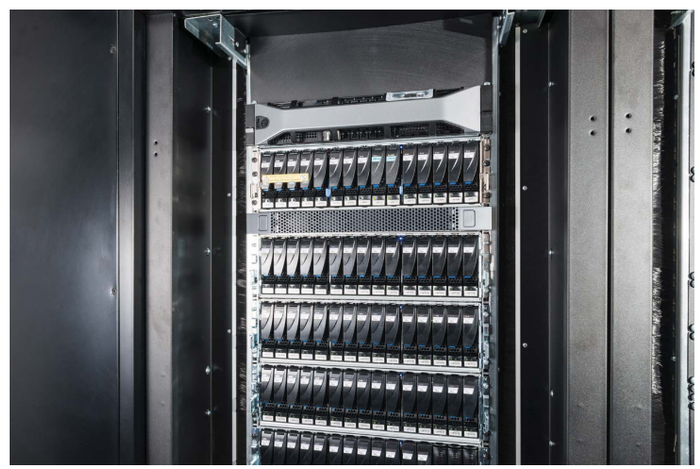
- **NAS** is storage made available over a network instead of a local bus
  - client can remotely attach to file systems on the server
  - NFS, CIFS, and iSCSI are common protocols
  - usually implemented via remote procedure calls (RPCs)
  - typically over TCP or UDP on IP network
    - **iSCSI** protocol uses IP network to carry the SCSI protocol



Storage Area Network

- **SAN** is a private network connecting servers and storage units
  - SAN consumes high bandwidth on the data network, separation is needed
  - TCP/IP stack less efficient for storage access
    - SAN uses **high speed interconnection and efficient protocols**
    - FC (Infiniband) is the most common SAN interconnection
  - multiple hosts and storage arrays can attach to the same SAN
    - a *cluster* of servers can share the same storage
- storage can be *dynamically* allocated to hosts





Disk Scheduling

- OS is responsible for using hardware efficiently
  - for the disk drives: a fast access time and high disk bandwidth
  - **access time**: seek time (roughly linear to seek distance) + rotational latency
- **disk bandwidth** is the speed of data transfer, data /time
  - data: total number of bytes transferred
  - time: between the first request and completion of the last transfer

Disk Scheduling

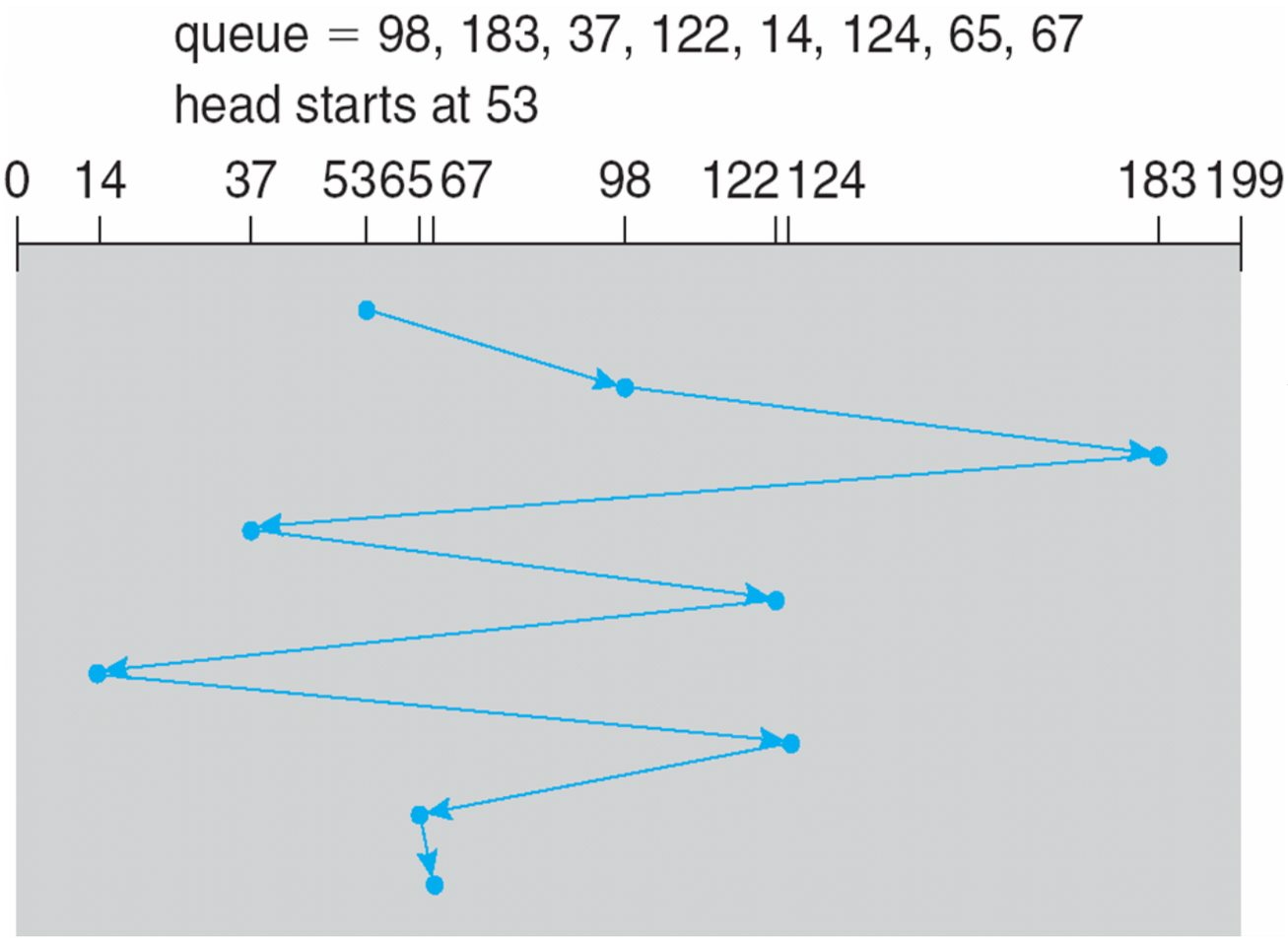
- **Disk scheduling** chooses which **pending disk request to service next**
  - concurrent sources of disk I/O requests include OS, system/user processes
  - idle disk can immediately work on a request, otherwise os queues requests
    - each request provide I/O mode, disk & memory address, and # of sectors
    - OS maintains a queue of requests, per disk or device
    - optimization algorithms only make sense when a queue exists
- In the past, operating system responsible for queue management, disk drive head scheduling
  - Now, **built into the storage devices, controllers - firmware**
  - Just provide **LBAs**, handle sorting of requests
    - Some of the algorithms they use described next

Disk Scheduling

- Disk scheduling usually tries to minimize **seek time**
  - rotational latency is difficult for OS to calculate
- There are many disk scheduling algorithms
  - FCFS
  - SSTF
  - SCAN
  - C-SCAN
  - C-LOOK
- We use a request queue of "**98, 183, 37, 122, 14, 124, 65, 67**" (**[0, 199]**), and initial head position **53** as the example

FCFS

- First-come first-served, simplest scheduling algorithm
- Total head movements of *640* cylinders



Advantage:

Every request gets a fair chance

No indefinite postponement

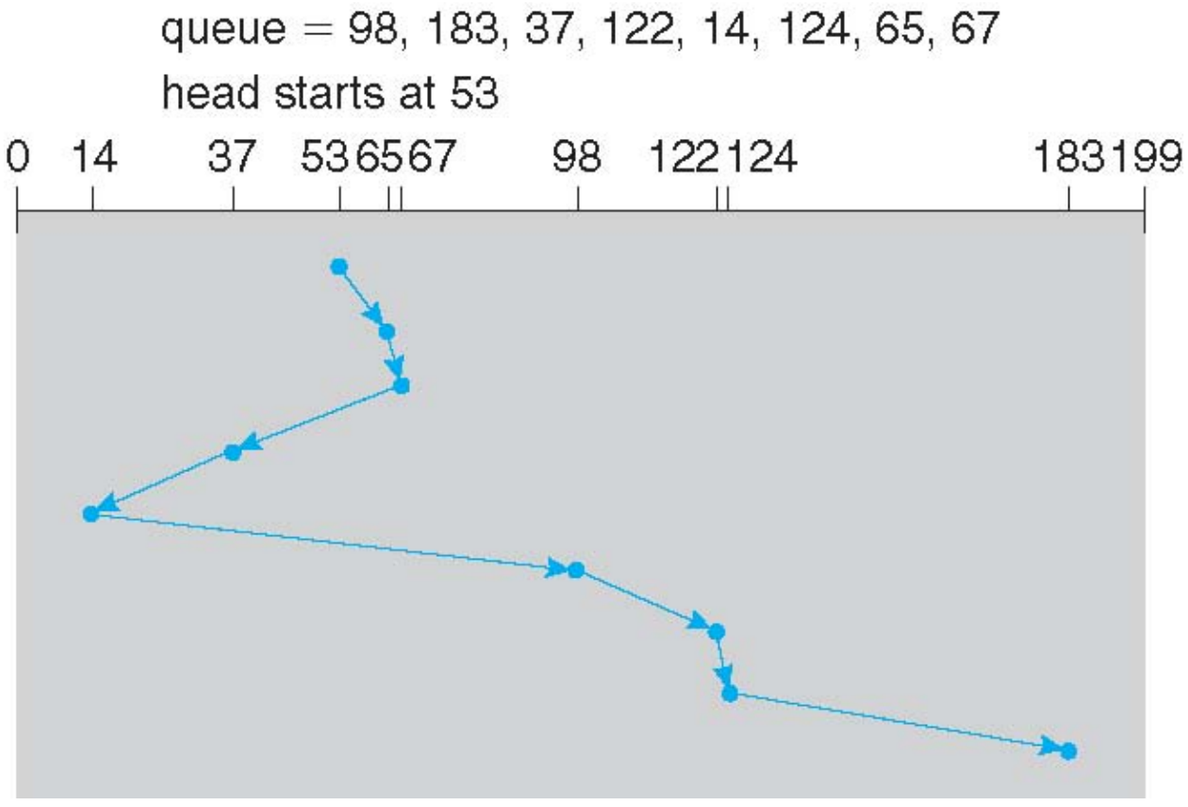
Disadvantages:

Does not try to optimize seek time

May not provide the best possible service

SSTF

- SSTF: shortest seek time first
  - selects the request with minimum seek time from the **current** head position
  - SSTF scheduling is a form of SJF scheduling, **starvation** may exist
    - unlike SJF, SSTF **may not** be **optimal**
- Total head movement of *236* cylinders



Advantage:

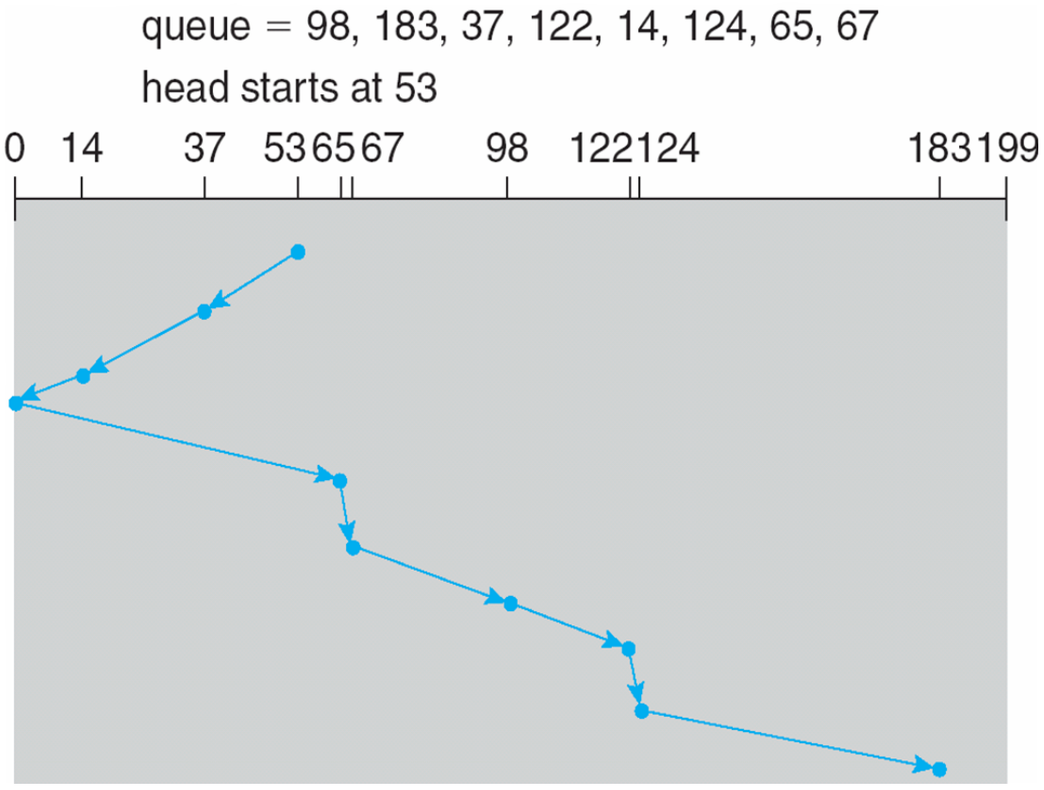
- Average Response Time decreases
- Throughput increases

Disadvantages:

- Overhead to calculate seek time in advance
- Can cause **Starvation** for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favors only some requests

SCAN

- SCAN algorithm sometimes is called the **elevator** algorithm
  - disk arm starts at one **end** of the disk, and moves toward the **other end**
  - service requests during the movement until it gets to the other end
  - then, the head movement is reversed and servicing continues.



Advantage:

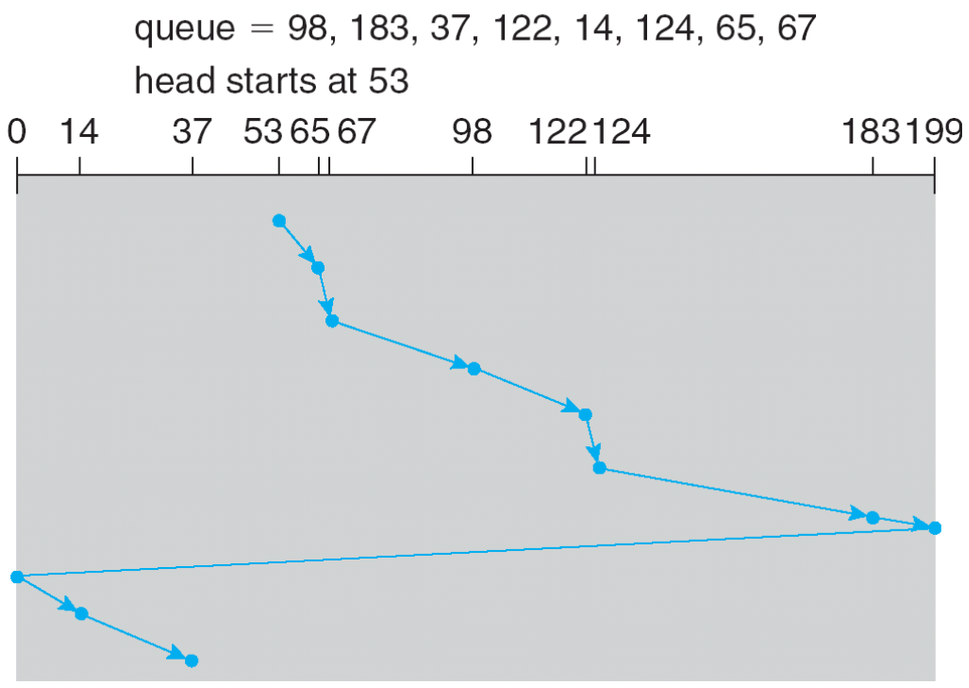
- High throughput
- Low variance of response time
- Average response time

Disadvantages:

- Long waiting time for requests for locations just visited by disk arm

C-SCAN

- Circular-SCAN is designed to provides a more uniform wait time
  - head moves from **one end** to **the other**, servicing requests while going
  - when the head reaches the end, it immediately returns to the beginning
    - **without** servicing any requests on the return trip
  - it essentially treats the cylinders as a circular list

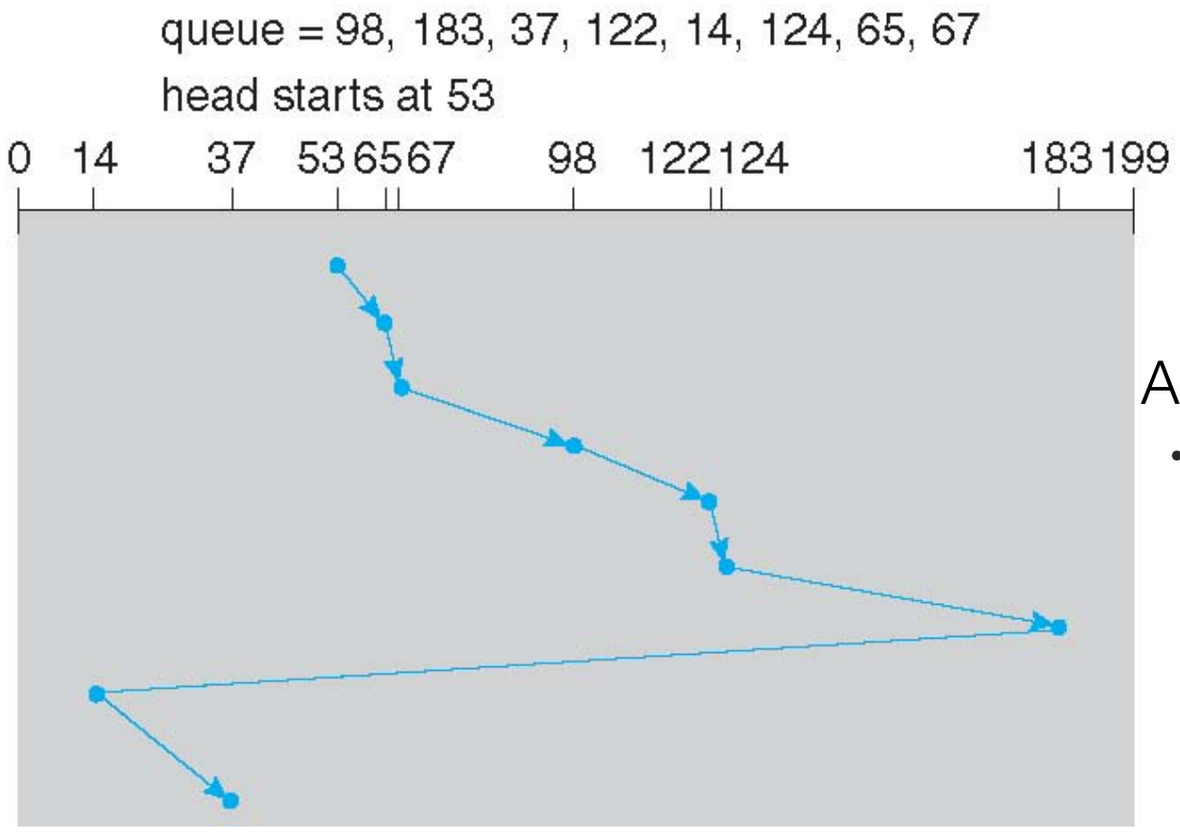


Advantage:

- Provides more uniform wait time compared to SCAN

LOOK/C-LOOK

- SCAN and C-SCAN moves head end to end, even no I/O in between
  - in implementation, head only goes as far as **last request** in each direction
  - **LOOK** is a version of **SCAN**, **C-LOOK** is a version of **C-SCAN**



Advantage:

- prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Selecting Disk-Scheduling Algorithm

- Disk scheduling performance depends on the # and types of requests
  - disk-scheduling should be written as a separate, replaceable, module
    - SSTF is common and is a reasonable choice for the default algorithm
    - LOOK and C-LOOK perform better for systems that have heavy I/O load
- disk performance can be influenced by file-allocation and metadata layout
  - file systems spend great deal of efforts to increase spatial locality

Disk Management

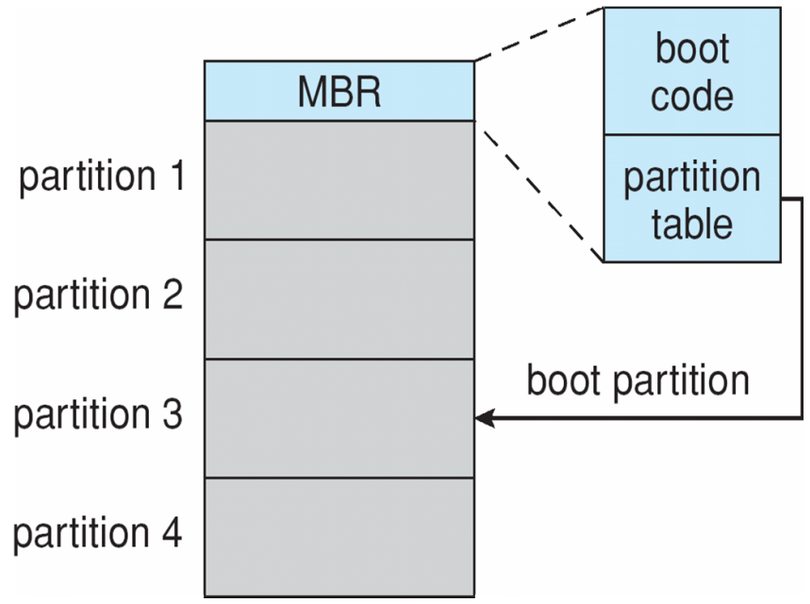
- **Physical formatting**: divide disk into sectors for controller to read/write
  - Each sector can hold header information, plus data, plus error correction code (ECC)
  - Usually 512 bytes of data but can be selectable
- OS records its own data structures on the disk
  - **partition disk** into groups of cylinders, each treated as a logical disk
  - **logical formatting** partitions to **make a file system** on it
    - some FS has spare sectors reserved to handle bad blocks
    - FS can further group blocks into clusters to improve performance
      - Disk I/O done in blocks
      - File I/O done in **clusters**
  - initialize the boot sector if the partition contains OS image

Disk Management

- **Root partition** contains the OS, other partitions can hold other OSes, other file systems, or be raw
  - **Mounted** at boot time
  - Other partitions can mount automatically or manually
- At mount time, file system consistency checked
  - Is all metadata correct?
    - If not, fix it, try again
    - If yes, add to mount table, allow access
- **Boot block** can point to boot volume or boot loader set of blocks that contain enough code to know how to load the kernel from the file system
  - Or a boot management program for multi-os booting

Disk Management

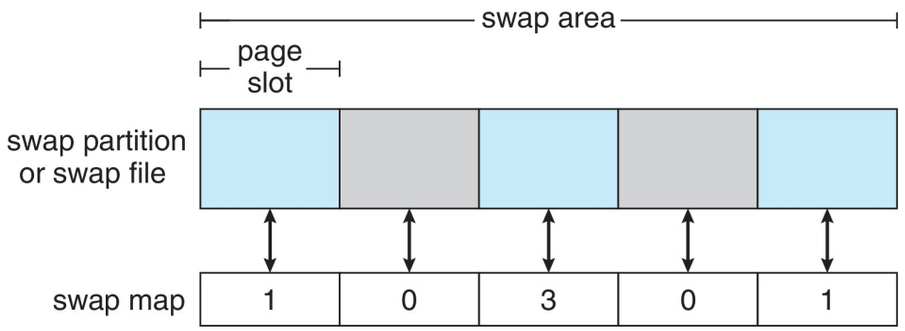
- Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example)
- Boot block initializes system
  - The bootstrap is stored in ROM, firmware
  - **Bootstrap loade**r program stored in boot blocks of boot partition
- Methods such as **sector sparing** used to handle bad blocks



Booting from secondary storage in Windows

Swap Space Management

- Used for moving entire processes (swapping), or pages (paging), from DRAM to secondary storage when DRAM not large enough for all processes
- Operating system provides **swap space management**
  - Secondary storage slower than DRAM, so important to optimize performance
  - Multiple swap spaces possible – decreasing I/O load on any given device
  - Best to have dedicated devices
  - Can be in raw partition or a file within a file system (for convenience of adding)
  - Data structures for swapping on Linux systems:



RAID

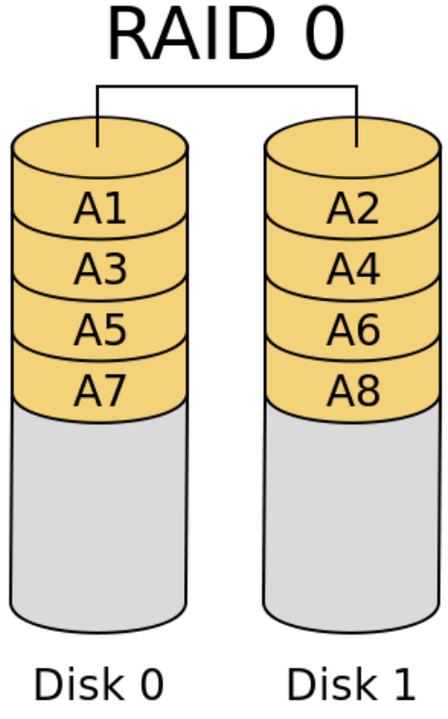
- **RAID – redundant array of inexpensive disks**
  - multiple disk drives provides reliability via **redundancy**
- Increases the **mean time to failure**
- **Mean time to repair** – exposure time when another failure could cause data loss
- **Mean time to data loss** based on above factors
- If mirrored disks fail independently, consider disk with 100,000 hours mean time to failure and 10 hour mean time to repair
  - Mean time to data loss is 100, 000x100, 000/(2 $*$ 10) = 500 $*$ 10^6 hours, or 57,000 years!
- Frequently combined with NVRAM to improve write performance
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively

RAID

- Disk **striping** uses a group of disks as one storage unit
- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliability of the 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
- **Block interleaved parity** (RAID 4, 5, 6) uses much less redundancy

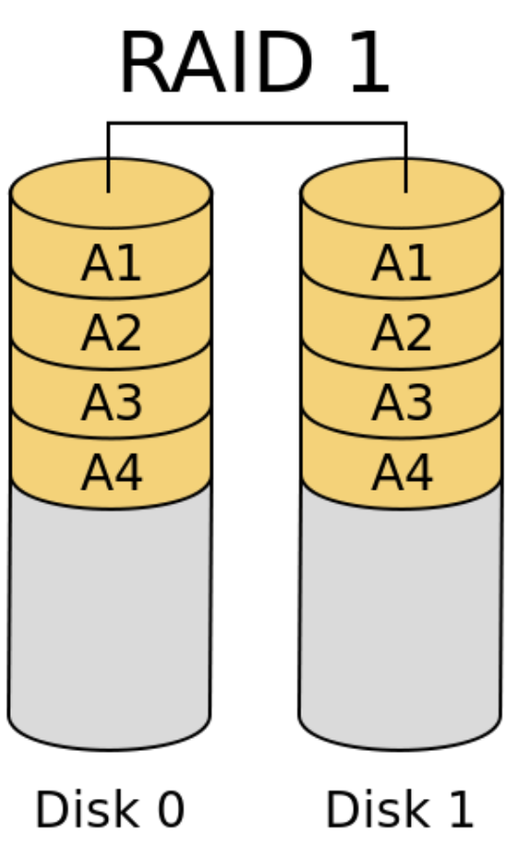
RAID 0

- **RAID 0**: splits data evenly across two or more disks without parity bits
  - read/write: N



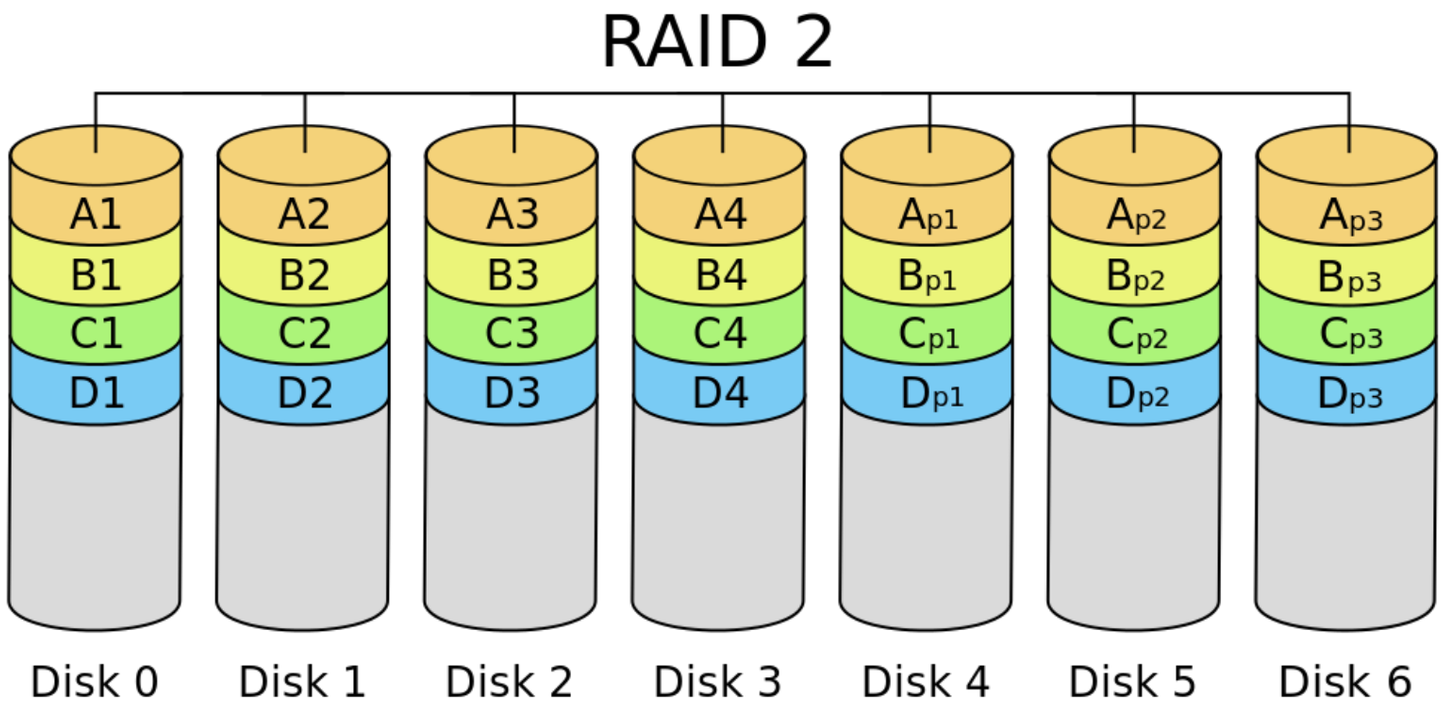
RAID 1

- an exact copy (or mirror) of a set of data on two disks
  - Read: N
  - Write: the slowest one



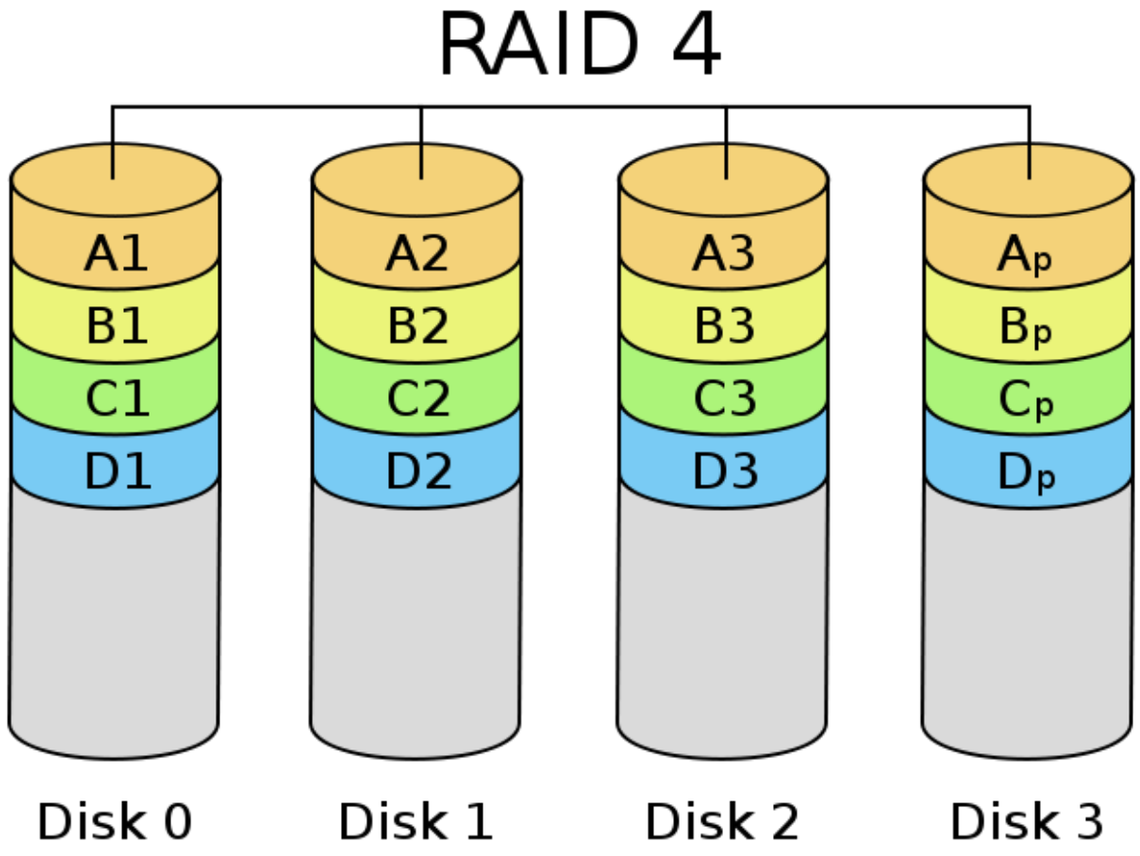
RAID2

- **RAID 2**: stripes data at the **bit**-level; uses Hamming code for error correction (not used)
  - hamming code (4bit data+3bit parity) allows 7 disks to be used
  - read: not N: since we have parity
  - Write: need to write to parity



RAID4

- **RAID 4**: block-level striping with a dedicated parity disk
  - a single block request can be fulfilled by one disk
  - different disk can fulfill different block requests
  - Read: N-1
  - Write: slower than RAID 0



RAID4

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

Figure 38.4: **RAID-4 with Parity**

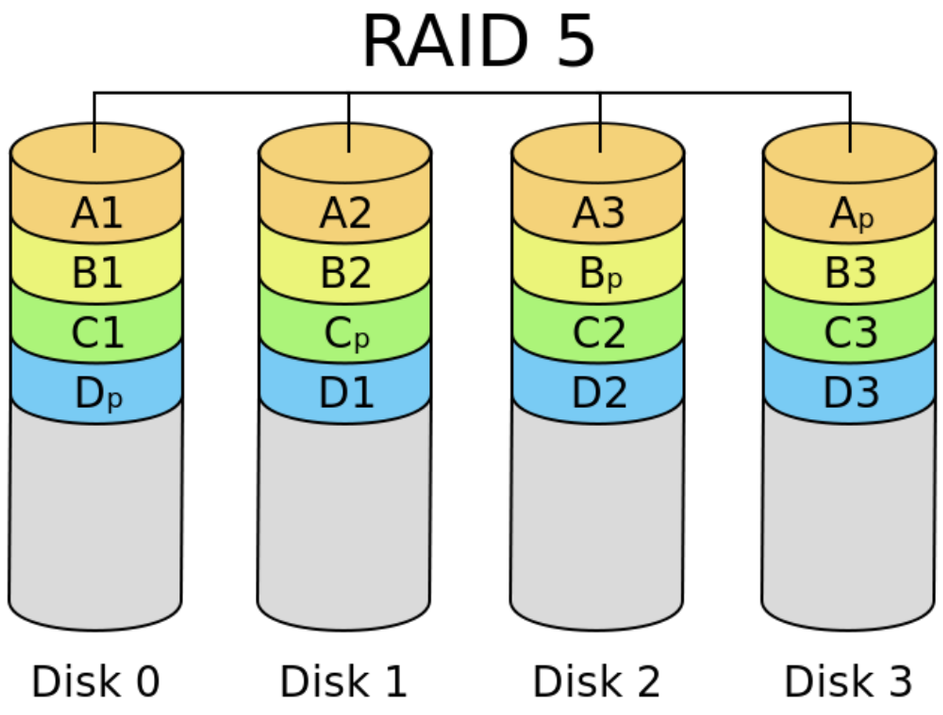
C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1) = 0
0	1	0	0	XOR(0,1,0,0) = 1

P: even number of 1

Block0	Block1	Block2	Block3	Parity
00	10	11	10	11
10	01	00	01	10

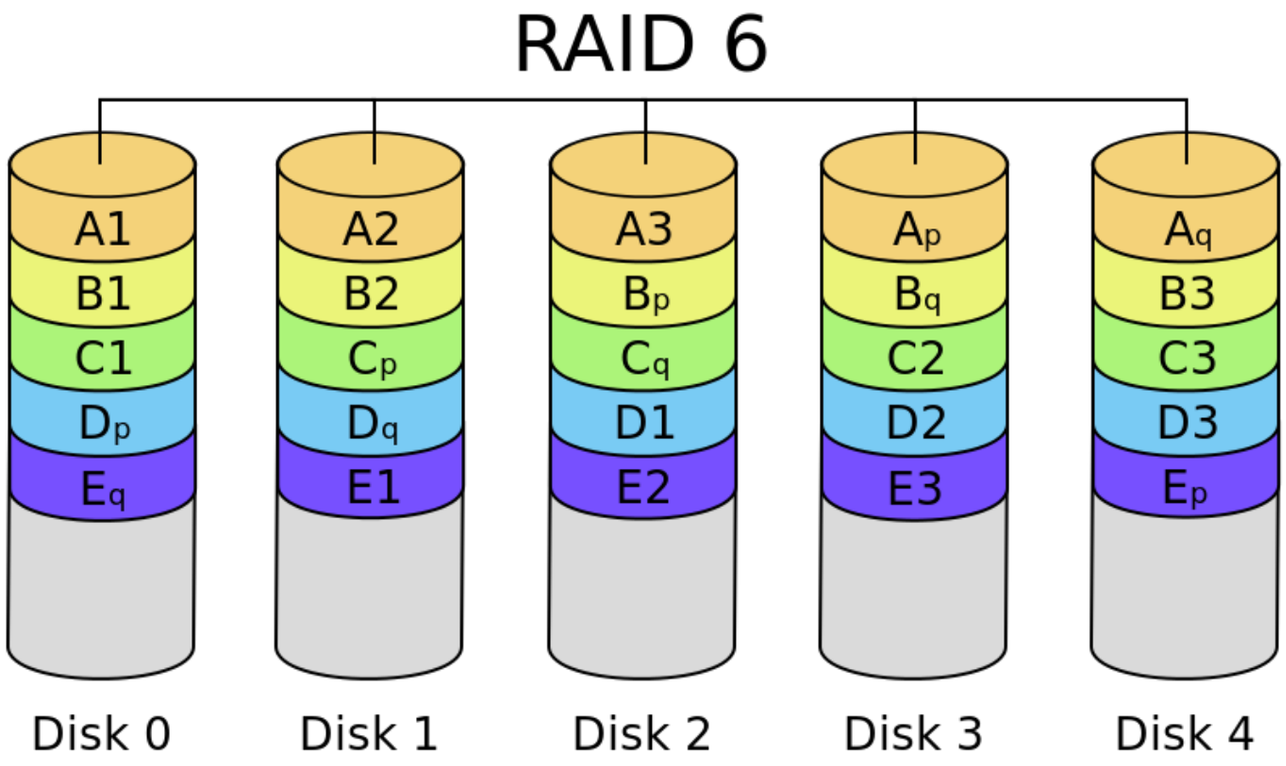
RAID5

- **RAID 5**: block-level striping with parity data distributed across all disks
  - Read: N
  - Write: slower than RAID 0



RAID6

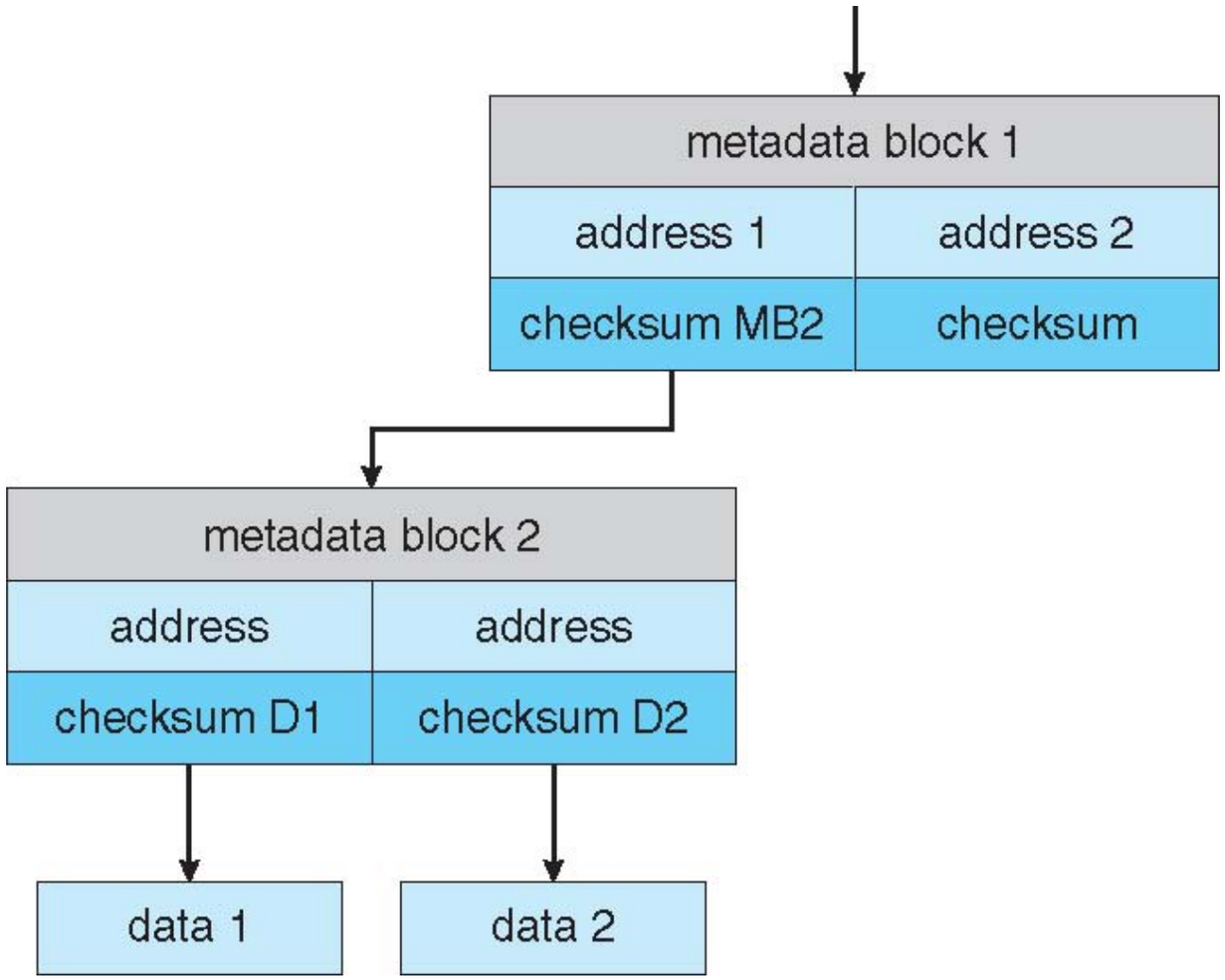
- **RAID 6**: extends RAID 5 by adding an additional parity block
  - RAID 6 has block-level striping with 2 parity blocks



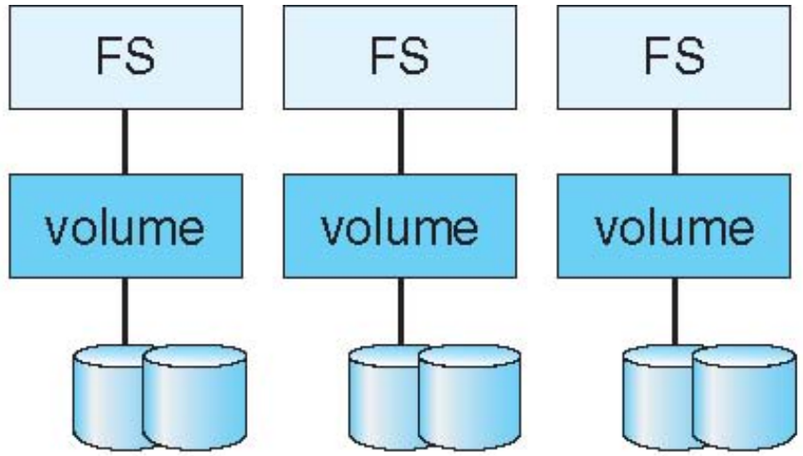
RAID and File Systems

- RAID can only detect/recover from **disk failures**
  - it **does not** prevent or detect **data corruption** or other errors
- File systems like Solaris ZFS add additional checks to detect errors
  - ZFS adds checksums to all **FS data and metadata**
    - checksum is collocated with pointer to the data/metadata
    - can detect and correct data and metadata corruption
- ZFS allocates disks in pools, instead of volumes or partitions
  - file systems within a pool share that pool, allocate/free space from/to pool

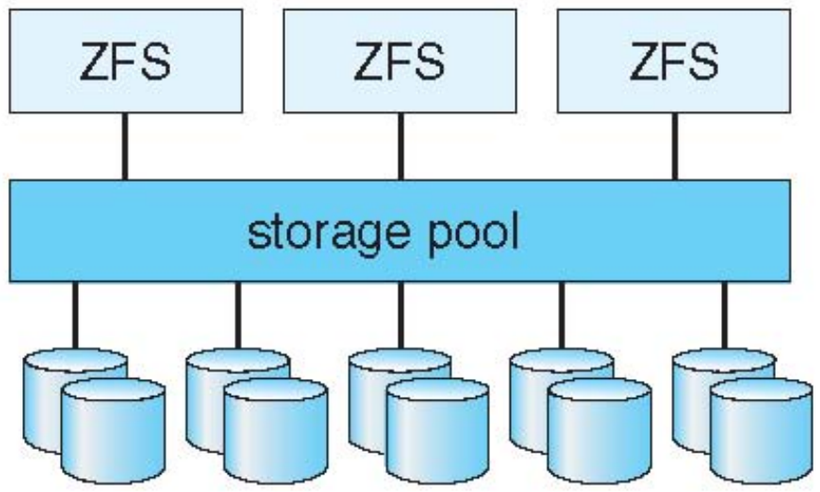
ZFS Checksums



Traditional and Pooled Storage



(a) Traditional volumes and file systems.



(b) ZFS and pooled storage.

HW is out!