Operating Systems
[11. Mass-Storage Systems]

Chung-Wei Lin
cwlin@csie.ntu.edu.tw
CSIE Department
National Taiwan University

From Operating System Concepts (10th Edition)

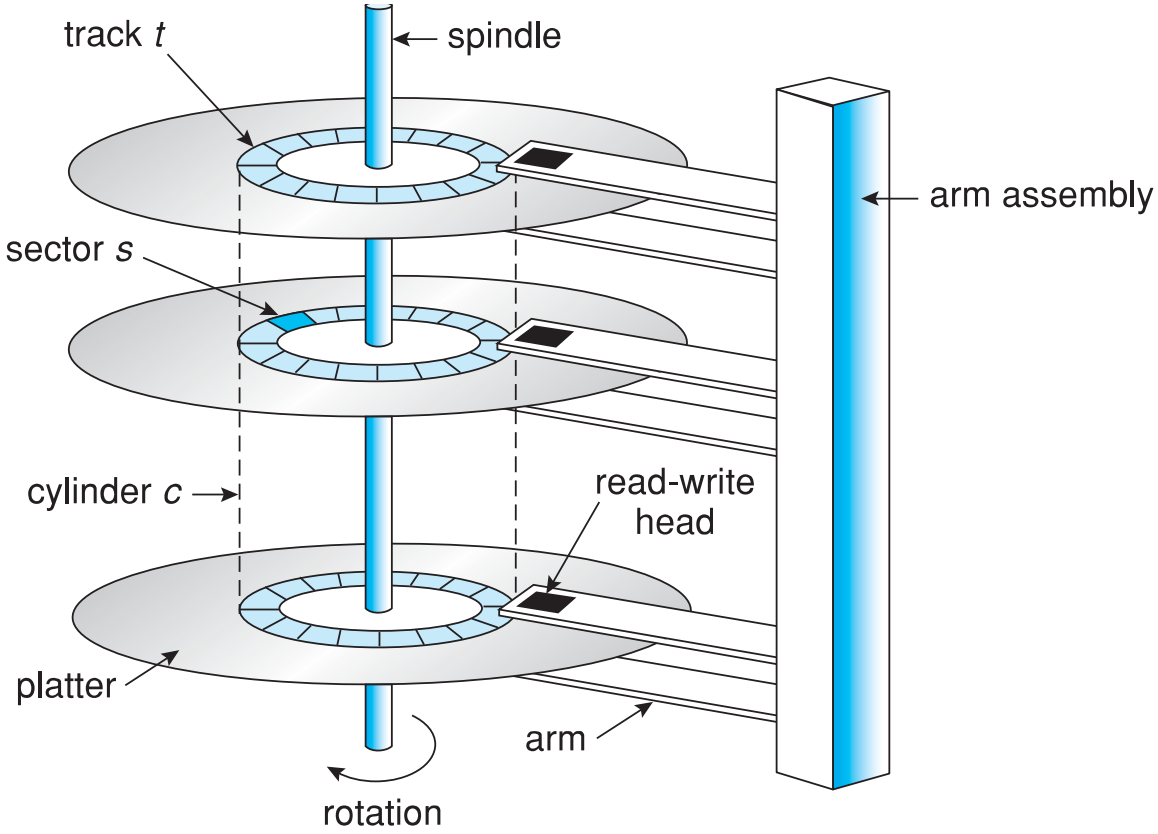
Objectives

- Describe the physical structure of secondary storage devices and the effect of a device's structure on its uses
- Explain the performance characteristics of mass-storage devices
- Evaluate I/O scheduling algorithms
- Discuss operating-system services provided for mass storage, including RAID

Outline

- **Overview of Mass Storage Structure**
  - **Hard Disk Drives (HDDs)**
  - Nonvolatile Memory (NVM)
  - Others
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

HDD Moving-Head Disk Mechanism



HDD Moving-Head Disk Mechanism

- Each disk **platter** has a flat circular shape
  - Common platter diameters range from 1.8 to 3.5 inches
  - The two surfaces of a platter are covered with a magnetic material
    - Store information by recording it magnetically on the platters
    - Read information by detecting the magnetic pattern on the platters
- A read-write head "flies" just above each surface of every platter
  - The heads are attached to a **disk arm** that moves all the heads as a unit
  - The surface of a platter is logically divided into circular **tracks**, which are subdivided into **sectors**
    - Each sector has a fixed size and is the smallest unit of transfer
    - The sector size was commonly 512 bytes until around 2010
    - At that point, many manufacturers start migrating to 4KB sectors
  - The set of tracks at a given arm position make up a **cylinder**

HDD Performance (1/2)

- Most drives rotate 60 to 250 times per second
  - Common 5,400, 7,200, 10,000, and 15,000 rotations per minute (RPM)
- **Transfer rate**
  - The rate at which data flow between the drive and the computer
- **Positioning time** (random-access time)
  - **Seek time**: time to move the disk arm to the desired cylinder, plus
  - **Rotational latency**: time for the desired sector to rotate to the disk head
- Typical disks can
  - Transfer tens to hundreds of megabytes of data per second
  - Have seek times and rotational latencies of several milliseconds
- **Head crash**
  - Result from disk head making contact with the disk surface

HDD Performance (2/2)

- Access time
  - Seek time + rotational latency
- I/O time
  - Access time + transfer time + controller overhead
- Example
  - Transfer a 4KB block on a 7200 RPM disk with a 5ms seek time, 1Gb/sec transfer rate, and 0.1ms controller overhead
  - Access time = 5ms + 4.17ms = 9.17ms
    - 4.17ms = 0.00417s = [1 / (7200 / 60)] * 0.5
  - Transfer time = 0.031ms
    - 0.031ms = 0.00031s = (4 * 8 / 1024 / 1024) / 1
  - I/O time = 9.17ms + 0.031ms + 0.1ms = 9.301ms

Outline

- **Overview of Mass Storage Structure**
  - Hard Disk Drives (HDDs)
  - **Nonvolatile Memory (NVM)**
  - Others
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

Nonvolatile Memory Devices (1/2)

- Flash-memory-based NVM is frequently used in a disk-drive-like container, called a solid-state disk (SSD)
  - Also take the form of a USB drive or a DRAM stick
  - Also surface-mounted onto motherboards as the main storage in devices like smartphones
- Advantages
  - More reliable (than HDDs) because they have no moving parts
  - Faster because they have no seek time or rotational latency
  - Less power consumption
- Disadvantages
  - More expensive per megabyte (dropping quickly)
  - Less capacity than the larger hard disks (increasing faster HDD)

Nonvolatile Memory Devices (2/2)

- Standard bus interfaces can cause a major limit on throughput
- Read and written in a "page" increment (similar to sector) but data cannot be overwritten
  - The cells have to be erased first
    - Occur in a "block" increment that is several pages in size
    - Take much more time than a read (the fastest operation) or a write (slower than read, but much faster than erase)
  - After approximately 100,000 program-erase cycles, the cells no longer retain data
  - The life span is measured in **drive writes per day (DWPD)**
    - A 1 TB NAND drive with a 5 DWPD rating is expected to have 5 TB per day written to it for the warranty period without failure

NAND Flash Controller Algorithms

- The controller maintains a **flash translation layer** (FTL) to track which logical blocks contain valid data
  - Without overwrite, there are pages containing invalid data
- **Garbage collection**
  - Good data could be copied to other locations, freeing up blocks that could be erased and could then receive the writes
    - Allocates **overprovisioning** to provide working space for garbage collection
  - The over-provisioning space can also help with **wear leveling**
    - The controller tries to place data on less-erased blocks so that subsequent erases will happen on those blocks

Valid Page	Valid Page	Invalid Page	Invalid Page
Invalid Page	Valid Page	Invalid Page	Valid Page

NAND block with valid and invalid pages

Outline

- **Overview of Mass Storage Structure**
  - Hard Disk Drives (HDDs)
  - Nonvolatile Memory (NVM)
  - **Others**
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

Volatile Memory

- RAM drives (RAM disks)
  - Device drivers carve out a section of the system's DRAM and present it as it is a storage device
  - These "drives" can be used as raw block devices, but more commonly, file systems are created on them for standard file operations
- Computers have buffering and caching, so why RAM drives?
  - Caches and buffers are allocated by the programmer or operating system
  - RAM drives allow the user to place data
    - Found in all major operating systems: Linux **`/dev/ram`**, macOS **`diskutil`** to create them, Linux **`/tmp`** of file system type **`tmpfs`**
- High-speed temporary storage

Connection Methods (1/2)

- A secondary storage device is attached to a computer by the system bus or an I/O bus
  - Advanced technology attachment (ATA)
  - Serial ATA (SATA)
  - eSATA
  - Serial attached SCSI (SAS)
  - Universal serial bus (USB)
  - Fibre channel (FC)
  - NVM express (NVMe) especially for NVM devices which are much faster than HDDs
    - NVMe directly connects the device to the system Peripheral Component Interconnect (PCI) bus, increasing throughput and decreasing latency

Connection Methods (2/2)

- Data transfers on a bus are carried out by special electronic processors, called **controllers** (or host-bus adapters (HBA))
  - The **host controller** is the controller at the computer end of the bus
  - A **device controller** is built into each storage device
- The computer places a command into the host controller, typically using memory-mapped I/O ports
  - The host controller then sends the command via messages to the device controller
  - Device controllers usually have a built-in cache
    - Data transfer at the drive happens between the cache and the storage media
    - Data transfer to the host occurs between the cache host DRAM via direct memory-access (DMA)

Address Mapping

- Storage devices are addressed as large one-dimensional arrays of **logical blocks**
  - The logical block is the smallest unit of transfer
  - Each logical block maps to a physical sector or semiconductor page
- Example
  - Sector 0 is the first sector of the first track on the outermost cylinder
  - Mapping proceeds in order
    - Through that track
    - Through the rest of the tracks on that cylinder
    - Through the rest of the cylinders from outermost to innermost
- A **logical block address** (LBA) is easier for algorithms to use than a sector, cylinder, head tuple or chip, block, or page tuple
  - Constant linear velocity (CLV) vs. constant angular velocity (CAV)

Outline

- Overview of Mass Storage Structure
- **HDD Scheduling**
  - FCFS Scheduling
  - SCAN Scheduling
  - C-SCAN Scheduling
  - Selection of a Disk-Scheduling Algorithm
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

HDD Scheduling (1/3)

- The operating system is responsible for using hardware efficiently
  - For HDDs, minimize access time and maximize data transfer bandwidth
- Access time
  - Seek time
    - Time to move the disk arm to the desired cylinder, plus
  - Rotational latency
    - Time for the desired sector to rotate to the disk head
- **Bandwidth**
  - The total number of bytes transferred, divided by
  - The total time between the first request for service and the completion of the last transfer

HDD Scheduling (2/3)

- Whenever a process needs I/O to or from the drive, it issues a system call to the operating system
  - Whether this operation is input or output
  - The open file handle indicating the file to operate on
  - What the memory address for the transfer is
  - The amount of data to transfer
- The existence of a queue of requests to a device that can have its performance optimized
  - If the desired drive and controller are available, the request can be serviced immediately
  - If the drive or controller is busy, any new request will be placed in the queue of pending requests for that drive

HDD Scheduling (3/3)

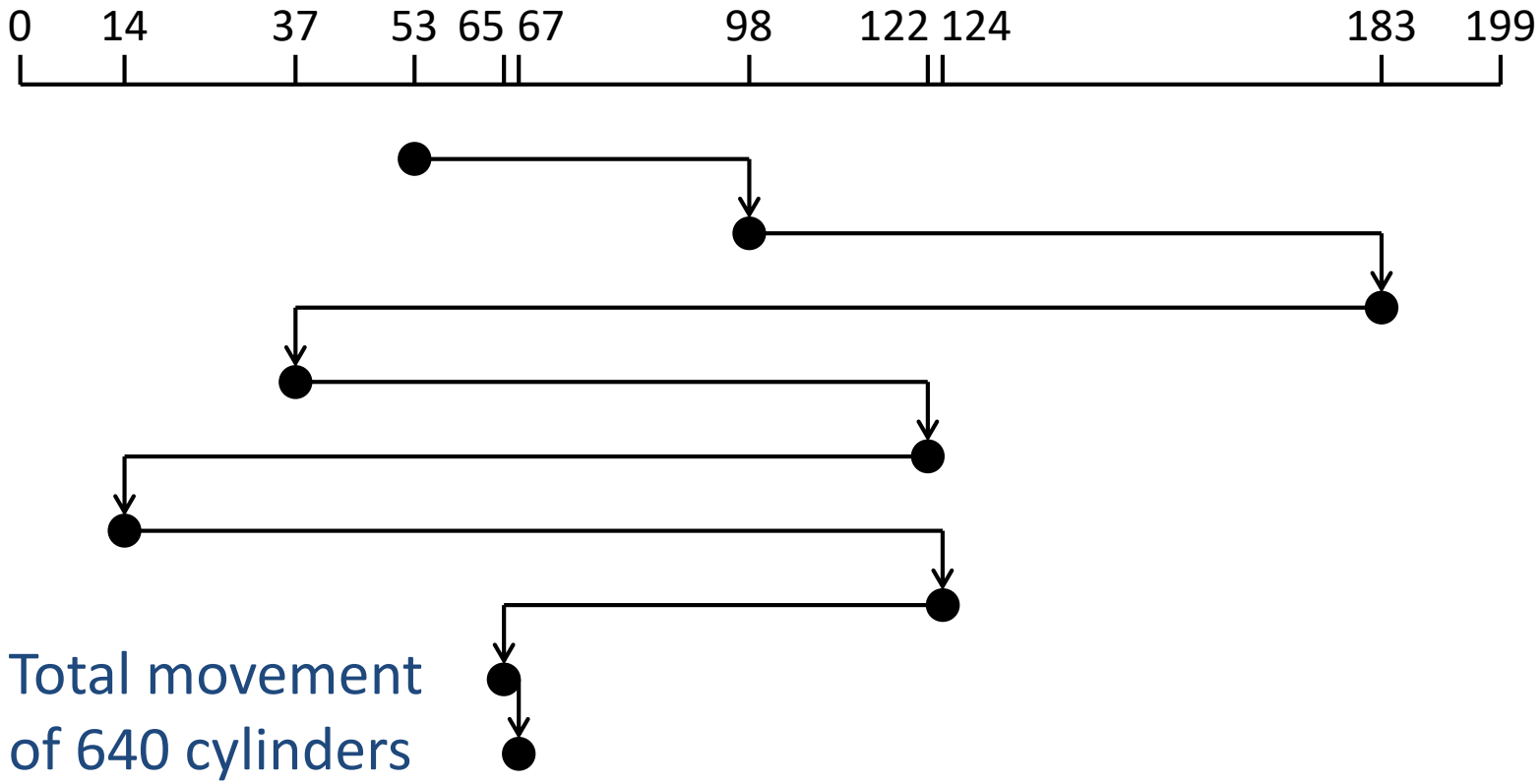
- Absolute knowledge of head location and physical block/cylinder locations is generally not possible on modern drives
  - As a rough approximation, algorithms can assume that
    - Increasing LBAs means increasing physical addresses
    - LBAs close together equate to physical block proximity

Outline

- Overview of Mass Storage Structure
- **HDD Scheduling**
  - **FCFS Scheduling**
  - SCAN Scheduling
  - C-SCAN Scheduling
  - Selection of a Disk-Scheduling Algorithm
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

FCFS Scheduling

- A request queue (0-199): 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer: 53



- Total movement of 640 cylinders

Outline

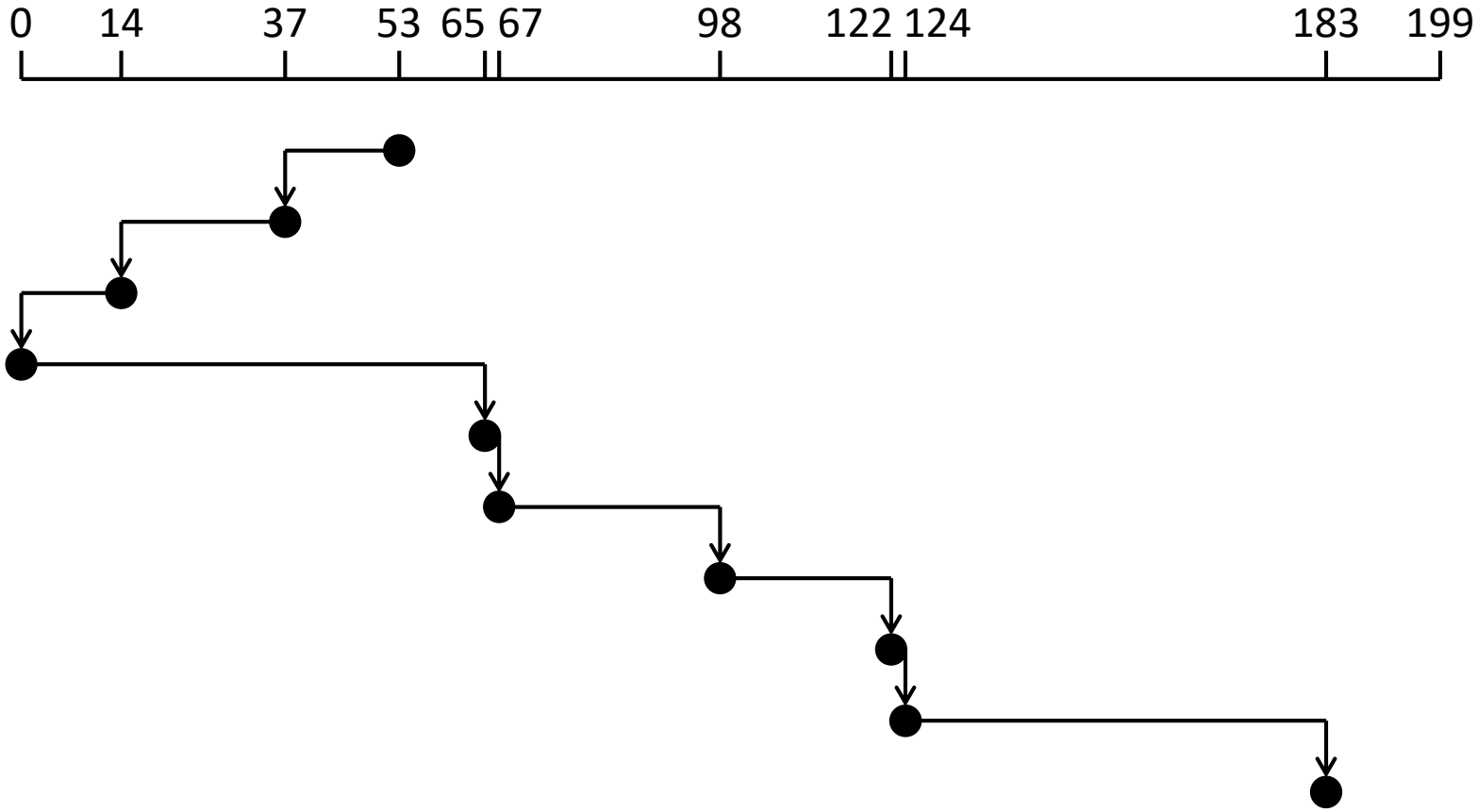
- Overview of Mass Storage Structure
- **HDD Scheduling**
  - FCFS Scheduling
  - **SCAN Scheduling**
  - C-SCAN Scheduling
  - Selection of a Disk-Scheduling Algorithm
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

SCAN Scheduling (1/2)

- The disk arm
  - Start at one end of the disk
  - Move toward the other end
  - Service requests until it gets to the other end of the disk
  - Reverse and continue
- **SCAN algorithm** sometimes called the **elevator algorithm**
  - Assume a uniform distribution of requests for cylinders and consider when the head reaches one end and reverses direction
    - The heaviest density of requests is at the other end of the disk
    - These requests have also waited the longest

SCAN Scheduling (2/2)

- A request queue (0-199): 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer: 53



Outline

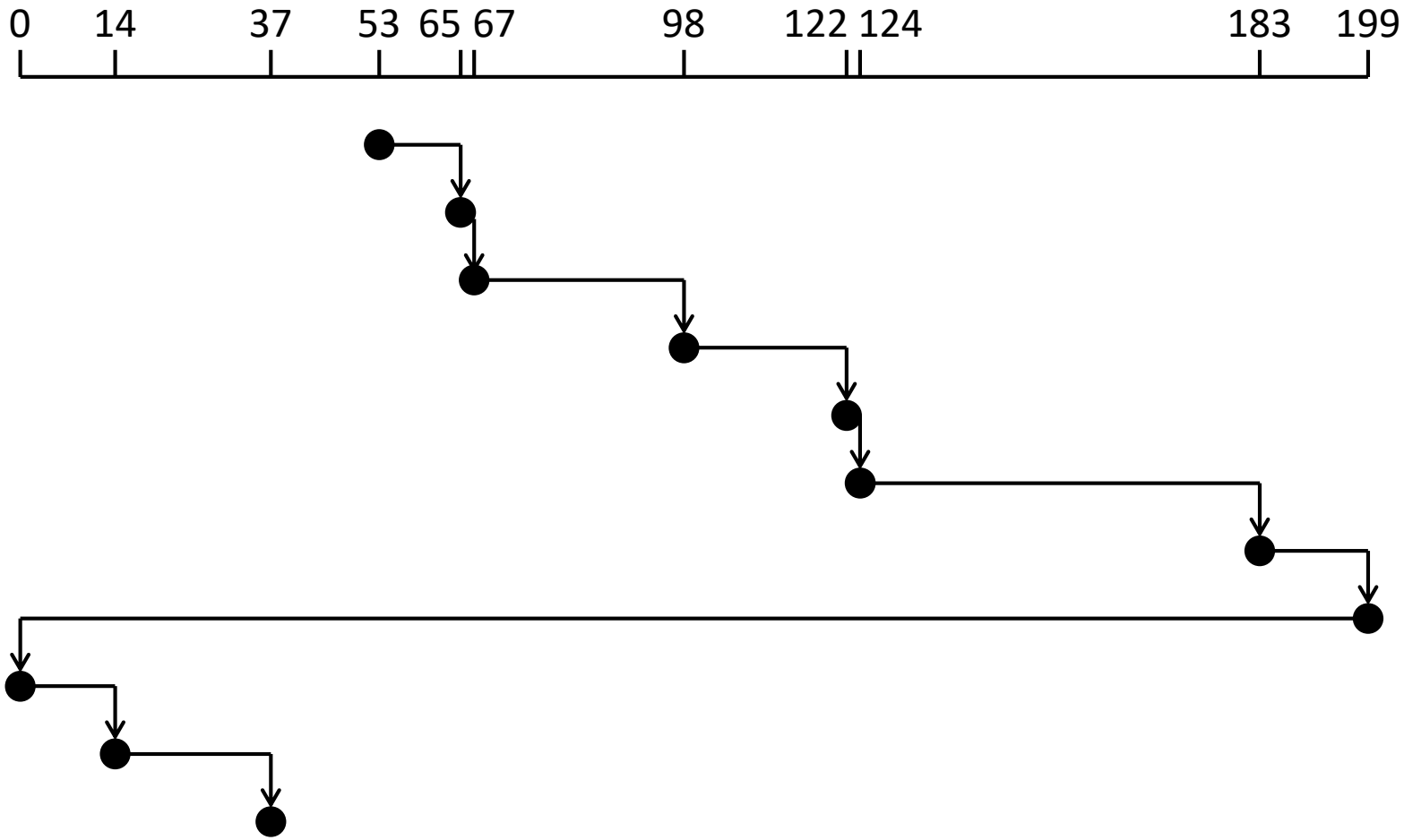
- Overview of Mass Storage Structure
- **HDD Scheduling**
  - FCFS Scheduling
  - SCAN Scheduling
  - **C-SCAN Scheduling**
  - Selection of a Disk-Scheduling Algorithm
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

C-SCAN Scheduling (1/2)

- The head moves from one end of the disk to the other, servicing requests along the way
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Provide a more uniform wait time than SCAN
- Treat the cylinders as a circular list that wraps around from the final cylinder to the first one

C-SCAN Scheduling (2/2)

- A request queue (0-199): 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer: 53



Outline

- Overview of Mass Storage Structure
- **HDD Scheduling**
  - FCFS Scheduling
  - SCAN Scheduling
  - C-SCAN Scheduling
  - **Selection of a Disk-Scheduling Algorithm**
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

Selecting a Disk-Scheduling Algorithm

- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation, but still possible
- To avoid starvation, Linux implements **deadline** scheduler
  - Maintain separate read and write queues and give read priority
    - Because processes more likely to block on read than write
  - Implement four queues
    - 1 read and 1 write queue sorted in LBA order
    - 1 read and 1 write queue sorted in FCFS order
    - After each batch, check if any request in FCFS older than configured age (default 500ms)
    - If so, the LBA queue containing that request is selected for next batch of I/O
- In RHEL 7, **NOOP** and **completely fair queueing scheduler** (CFQ) are also available

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- **NVM Scheduling**
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

NVM Scheduling

- NVM devices do not contain moving disk heads and commonly use a simple FCFS policy
  - Example: The Linux NOOP scheduler uses an FCFS policy but modifies it to merge adjacent requests
- Random-access I/O is much faster on NVM
  - Measured in **input/output operations per second** (IOPS)
  - An HDD can produce hundreds of IOPS
    - Sequential access is optimal for mechanical devices like HDD and tape
  - An SSD can produce hundreds of thousands of IOPS
- However, **write amplification** can greatly impact the write performance of the device
  - One write causes garbage collection and many reads/writes

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- **Error Detection and Correction**
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- RAID Structure

Error Detection and Correction

- Fundamental to many areas (memory, networking, storage)
- **Error detection** determines if a problem has occurred
  - Example: a bit flipping and parity bit (odd or even numbers of 1's)
  - The system can halt an operation, report the error, provide warning
- **Checksum**
  - Use modular arithmetic to compute, store, compare values on fixed-length words
- **Cyclic redundancy check (CRC)**, common in networking
  - Use a hash function to detect multiple-bit errors
- **Error-correction code (ECC)**
  - Not only detects the problem but also correct it
    - A **soft error** is recoverable
    - A **hard error** is signaled but noncorrectable

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- **Storage Device Management**
  - **Drive Formatting, Partitions, and Volumes**
  - Boot Block
  - Bad Blocks
- Swap-Space Management
- Storage Attachment
- RAID Structure

Formatting, Partitions, and Volumes

- **Low-level formatting** (or **physical formatting**)
  - Divide a disk into sectors that the disk controller can read and write
    - Each sector can hold a header, data, and a trailer (e.g., error correction code)
    - It is usually possible to choose among a few sector sizes (e.g., 512 bytes, 4KB)
- The operating system still needs to record its own data structures on the device to use a drive to hold files
  - **Partition** the device into one or more groups of blocks or pages
    - **Mounting** is to make the file system available for use by the system and users
  - **Volume** creation and management
  - **Logical formatting** (or creation of a file system)
    - Store the initial file-system data structures onto the device
- Most file systems group blocks into **clusters** for efficiency
  - Device I/O is done via blocks, but file system I/O is done via clusters

Outline

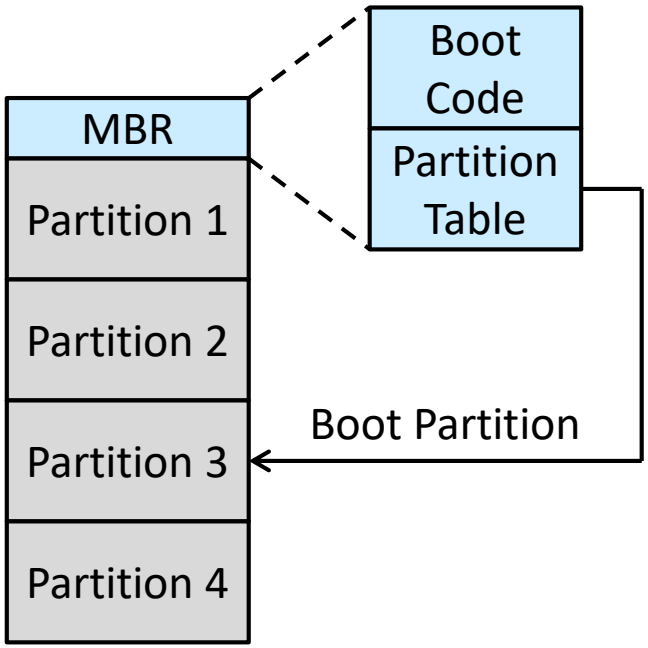
- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- **Storage Device Management**
  - Drive Formatting, Partitions, and Volumes
  - **Boot Block**
  - Bad Blocks
- Swap-Space Management
- Storage Attachment
- RAID Structure

Boot Block

- For most computers, the **bootstrap** is
  - Stored in NVM flash memory firmware on the system motherboard
  - Mapped to a known memory location
- A tiny bootstrap loader program can bring in a full bootstrap program from secondary storage
  - The full bootstrap program is stored in the "boot blocks" at a fixed location on the device
  - A device that has a boot partition is called a **boot disk** or **system disk**

Boot Process in Windows

- **Master boot record (MBR)** contains
  - Boot code in the first logical block on the hard disk or first page of the NVM device
  - A table listing the partitions for the drive
  - A flag indicating which partition the system is to be booted from
- Steps
  - Run the code in the system's firmware
  - Direct to read the boot code from the MBR
  - Read the first sector/page from the **boot partition** which directs to the kernel



Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- **Storage Device Management**
  - Drive Formatting, Partitions, and Volumes
  - Boot Block
  - **Bad Blocks**
- Swap-Space Management
- Storage Attachment
- RAID Structure

Bad Blocks

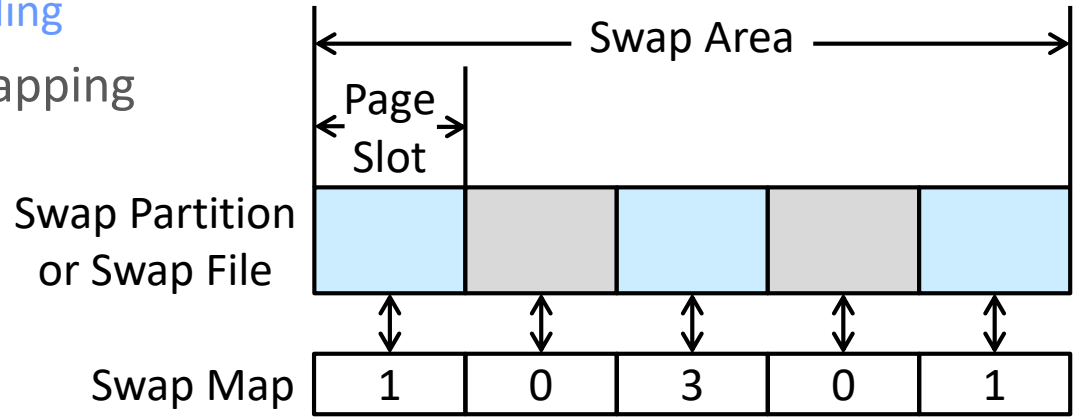
- Most disks even come from the factory with **bad blocks**
  - **Sector sparing** can be used to handle bad blocks
    - The operating system tries to read logical block 87
    - The controller calculates the error-correction code (ECC) and finds that the sector is bad
    - It reports this finding to the operating system as an I/O error
    - The device controller replaces the bad sector with a spare
    - After that, whenever the system requests logical block 87, the request is translated into the replacement sector's address by the controller
  - **Sector slipping**

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- **Swap-Space Management**
- Storage Attachment
- RAID Structure

Swap-Space Management

- Used for moving entire processes (swapping) or pages (paging) from DRAM to secondary storage
  - When DRAM is not large enough for all processes
- Operating system provides swap-space management
  - Secondary storage is slower than DRAM
  - These swap spaces are usually placed on separate storage devices
    - The load placed on the I/O system by paging and swapping can be spread over the system's I/O bandwidth
  - It can be in raw partition or a file within a file system
    - For convenience of adding
  - Data structures for swapping on Linux systems



Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- **Storage Attachment**
  - **Host-Attached Storage**
  - Network-Attached Storage
  - Cloud Storage
  - Storage-Area Networks and Storage Arrays
- RAID Structure

Host-Attached Storage

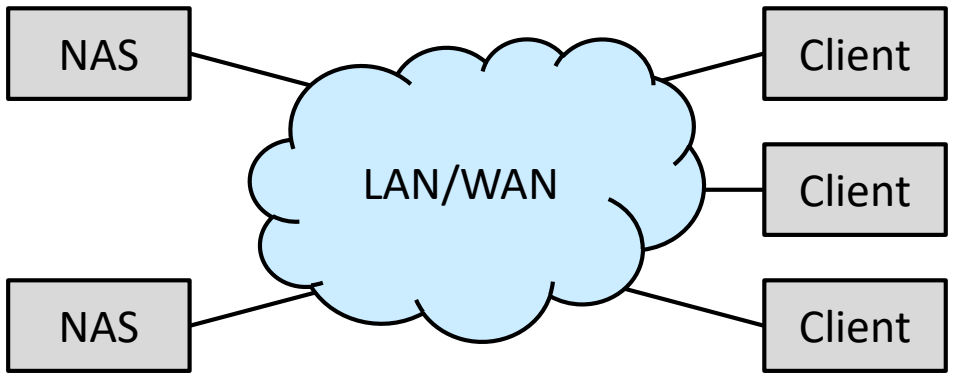
- **Host-attached storage** is storage accessed through local I/O ports
  - The most common is SATA
  - A typical system has one or a few SATA ports
- To gain access to more storage, USB FireWire or Thunderbolt ports and cables can be used
- High-end workstations and servers
  - Need more storage or need to share storage
  - Use more sophisticated I/O architectures
  - Example: **fibre channel** (FC)
    - A high-speed serial architecture that can operate over optical fiber or over a four-conductor copper cable

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- **Storage Attachment**
  - Host-Attached Storage
  - **Network-Attached Storage**
  - Cloud Storage
  - Storage-Area Networks and Storage Arrays
- RAID Structure

Network-Attached Storage

- **Network-attached storage** (NAS) provides access to storage across a network
  - Clients access network-attached storage via a remote-procedure-call interface
    - Example: NFS for UNIX and Linux, CIFS for Windows
  - The remote procedure calls (RPCs) are carried via TCP or UDP over an IP network
    - Usually the same local-area network (LAN) that carries all data traffic
  - iSCSI protocol uses the IP network protocol to carry the SCSI protocol
    - Hosts can treat their storage as if it were directly attached



WAN: Wide Area Network

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- **Storage Attachment**
  - Host-Attached Storage
  - Network-Attached Storage
  - **Cloud Storage**
  - Storage-Area Networks and Storage Arrays
- RAID Structure

Cloud Storage

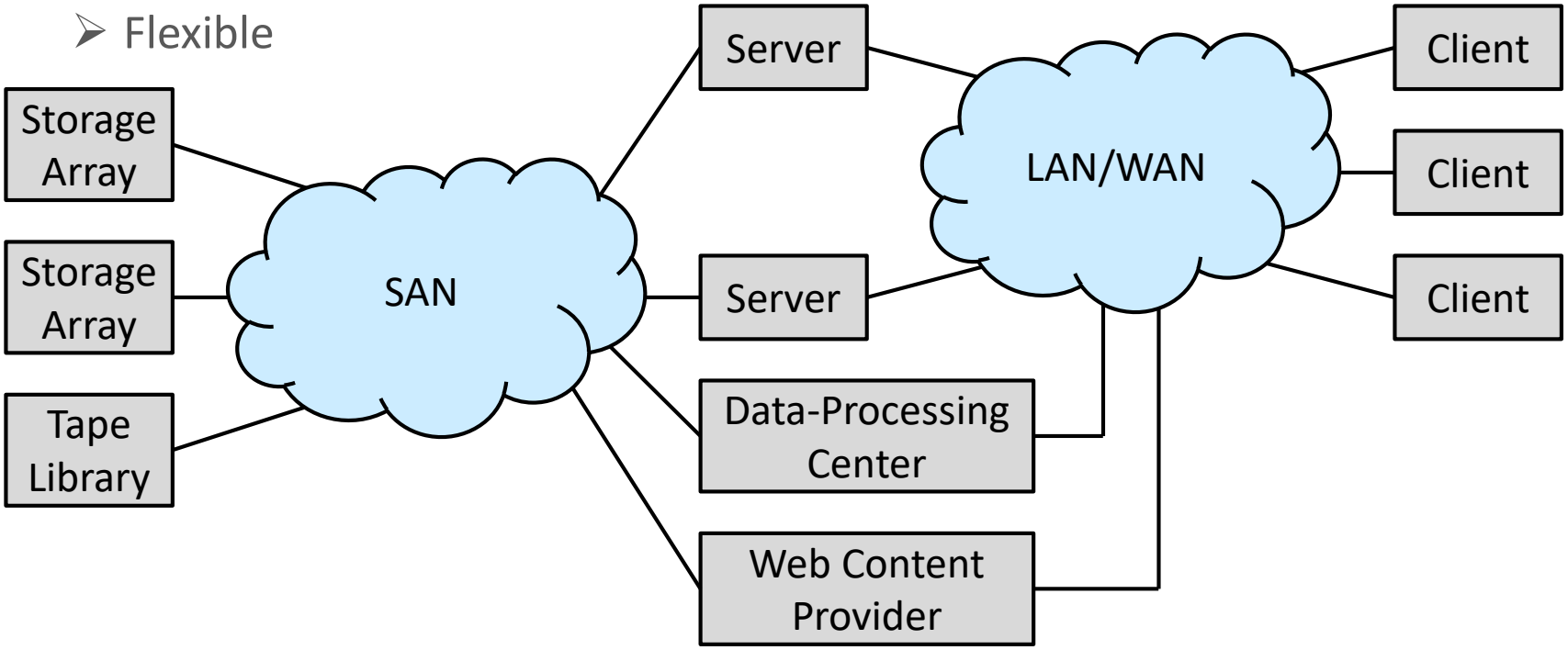
- Similar to NAS, cloud storage provides access to storage across a network
  - Unlike NAS, the storage is accessed over the Internet or another WAN to a remote data center that provides storage
- NAS is accessed as
  - Another file system if the CIFS or NFS protocols are used, or
  - A raw block device if the iSCSI protocol is used
- Cloud storage is API based
  - Programs use the APIs to access the storage
  - One reason: the latency and failure scenarios of a WAN
  - Examples: Amazon S3, Dropbox, Microsoft OneDrive, and Apple iCloud

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- **Storage Attachment**
  - Host-Attached Storage
  - Network-Attached Storage
  - Cloud Storage
  - **Storage-Area Networks and Storage Arrays**
- RAID Structure

Storage Area Networks and Storage Arrays

- A network connecting servers and storage units
- Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts
  - Flexible



Storage Area Networks and Storage Arrays

- A storage array is a purpose-built device that includes SAN ports, network ports, or both
  - It also contains drives to store data and a controller to manage the storage
  - The controllers are composed of CPUs, memory, and software that implement the features of the array
    - Network protocols, user interfaces, RAID protection, snapshots, replication, compression, deduplication, and encryption
- FC is the most common SAN interconnect
  - The simplicity of iSCSI is increasing its use
  - Another SAN interconnect is **InfiniBan** (IB)
    - A special purpose bus architecture that provides hardware and software support for high-speed interconnection networks for servers and storage units

Outline

- Overview of Mass Storage Structure
- HDD Scheduling
- NVM Scheduling
- Error Detection and Correction
- Storage Device Management
- Swap-Space Management
- Storage Attachment
- **RAID Structure**

Improve Reliability via Redundancy

- **Redundant arrays of independent (inexpensive) disks** (RAID)
  - Provide reliability via **redundancy** with multiple disk drives
  - Increase the **mean time to data loss**
    - **Mean time between failures** (MTBF)
    - **Mean time to repair**: the time it takes (on average) to replace a failed drive and to restore the data on it
- Example
  - If the MTBF of a single drive is 100,000 hours and the mean time to repair is 10 hours
  - Then the mean time to data loss of a mirrored drive system is $100{,}000^2 / (2 \times 10) = 500 \times 10^6$ hours = 57,000 years

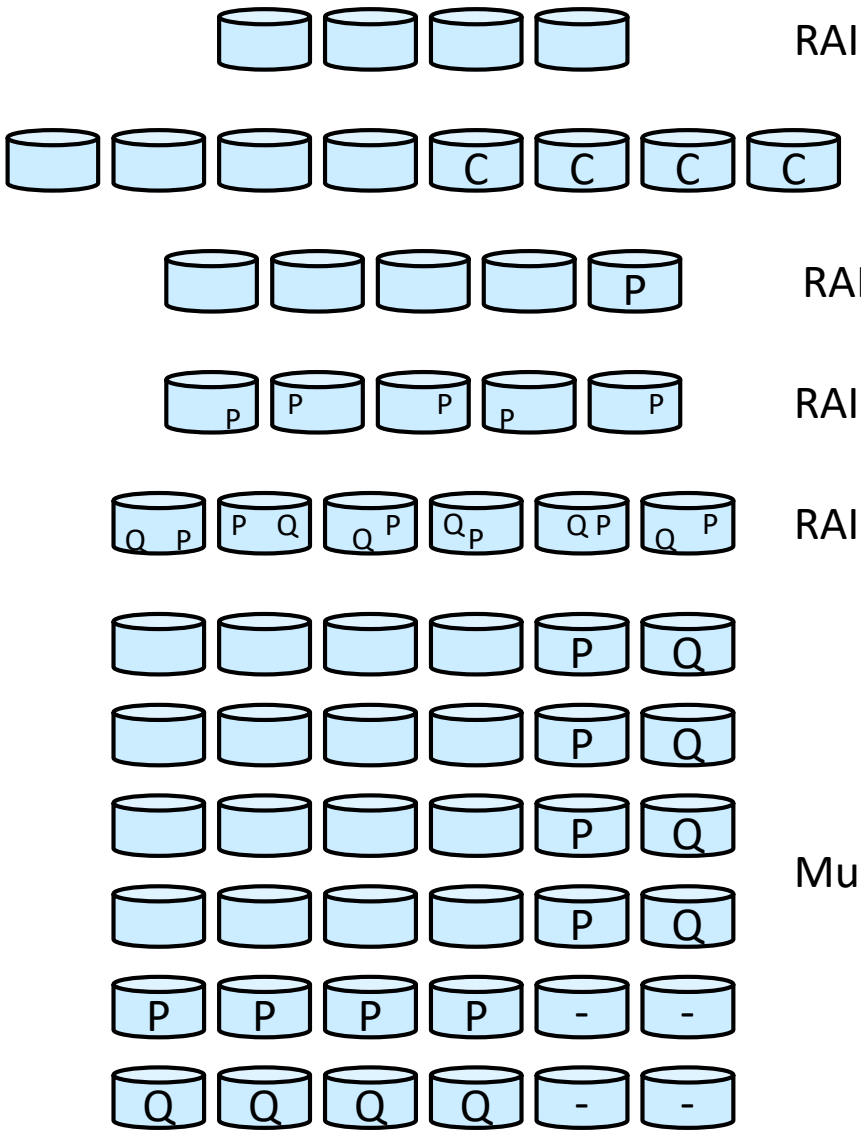
Improve Performance via Parallelism

- With mirroring, the rate at which read requests can be handled is doubled
  - Read requests can be sent to either drive
- **Data striping** consists of splitting the bits of each byte across multiple drives
  - Called **bit-level striping**
- In **block-level striping**, blocks of a file are striped across multiple drives
- Goals of parallelism in a storage system
  - Increase the throughput of multiple small accesses by load balancing
  - Reduce the response time of large accesses

RAID Levels (1/3)

- **Mirroring** or **shadowing (RAID 1)** keeps duplicate of each disk
- **Block interleaved parity (RAID 4, 5, 6)** uses much less redundancy
- Striped mirrors (**RAID 1+0**) or mirrored stripes (**RAID 0+1**) provides high performance and high reliability

RAID Levels (2/3)



RAID 0: Non-Redundant Striping

RAID 1: Mirrored Disks

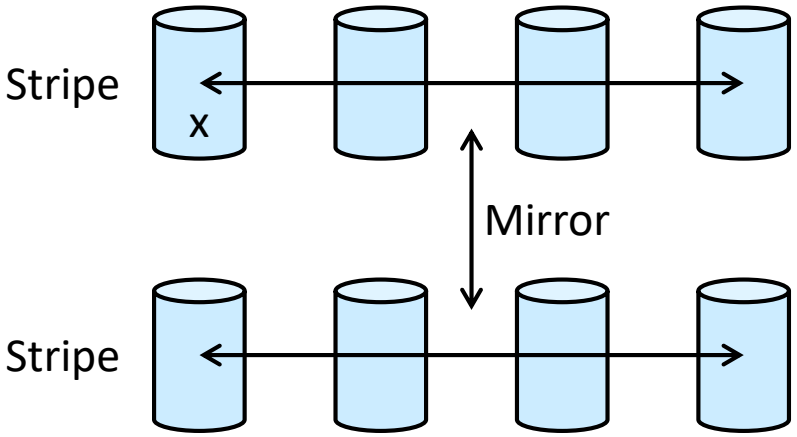
RAID 4: Block-Interleaved Parity

RAID 5: Block-Interleaved Distributed Parity

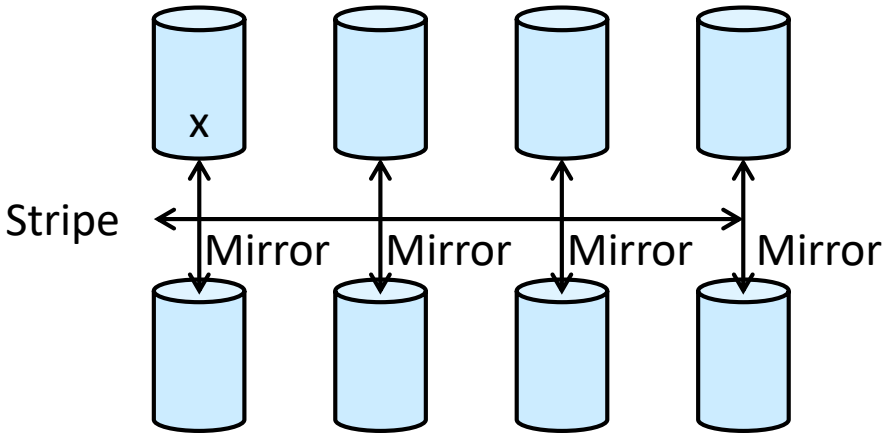
RAID 6: P + Q Redundancy

Multidimensional RAID 6

RAID Levels (3/3)



RAID 0 + 1 with a single disk failure



RAID 1 + 0 with a single disk failure

Other Features

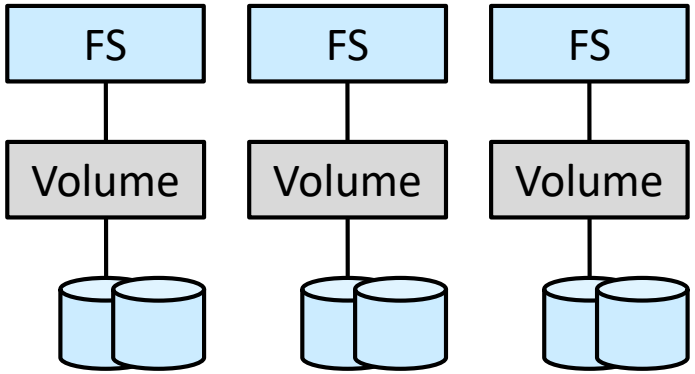
- A **snapshot** is a view of the file system before the last update took place
- **Replication** involves the automatic duplication of writes between separate sites for redundancy and disaster recovery
  - Replication can be synchronous or asynchronous
- A **hot spare** is not used for data but is configured to be used as a replacement in case of drive failure
  - The RAID level can be reestablished automatically, without waiting for the failed drive to be replaced

Problems with RAID (1/2)

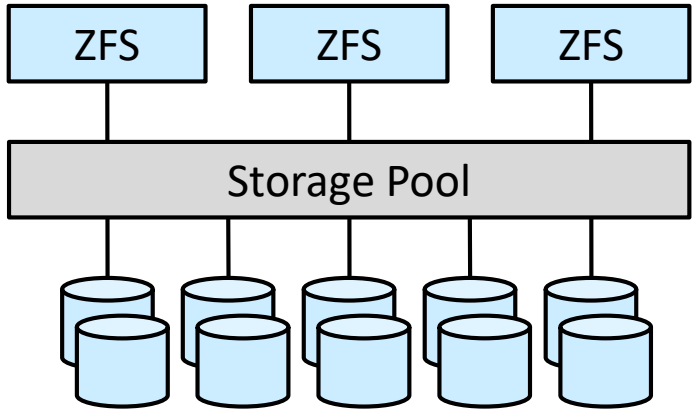
- RAID does not always assure that data are available for the operating system and its users
  - The Solaris ZFS maintains internal checksums of all blocks, including data and metadata
    - These checksums are not kept with the block that is being checksummed
    - Consider an **inode**, a data structure for storing file system metadata, with pointers to its data
    - Within the inode is the checksum of each block of data

Problems with RAID (2/2)

- Most RAID implementations lack flexibility
  - ZFS gathers drives, or partitions of drives, together via RAID sets into **pools** of storage
    - A pool can hold one or more ZFS file systems
    - ZFS uses the memory model of `malloc()` and `free()` to allocate and release storage for each file system



Traditional Volumes and File Systems



ZFS and Pooled Storage

Object Storage (1/2)

- General-purpose computers typically use file systems to store content for users
- Another approach: start with a storage pool and place objects in that pool
  - No way to navigate the pool and find those objects
  - Computer-oriented, not user-oriented
- A typical sequence
  - Create an object within the storage pool and receive an object ID
  - Access the object when needed via the object ID
  - Delete the object via the object ID

Object Storage (2/2)

- Object storage management software determine where to store the objects and manages object protection
  - Examples: Hadoop file system (HDFS) and Ceph
    - HDFS can store N copies of an object on N different computers
  - Typically by storing N copies, across N systems, in the object storage cluster
- **Horizontally scalable**
  - To add capacity to an object store, we simply add more computers with internal disks or attached external disks and add them to the pool
- **Content addressable**
  - Objects can be retrieved based on their contents
  - There is no set format for the contents, so what the system stores is **unstructured data**

Objectives

- Describe the physical structure of secondary storage devices and the effect of a device's structure on its uses
- Explain the performance characteristics of mass-storage devices
- Evaluate I/O scheduling algorithms
- Discuss operating-system services provided for mass storage, including RAID

Q&A