



# Mass-Storage Systems

Amir H. Payberah  
payberah@kth.se  
2022



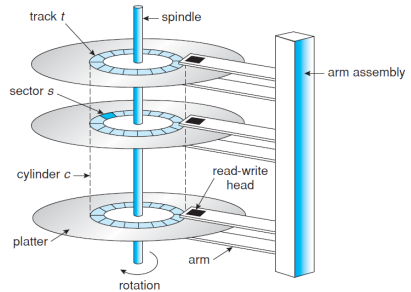


# Motivation

- ▶ Main memory is usually too small.
- ▶ Computer systems must provide secondary storage to back up main memory.

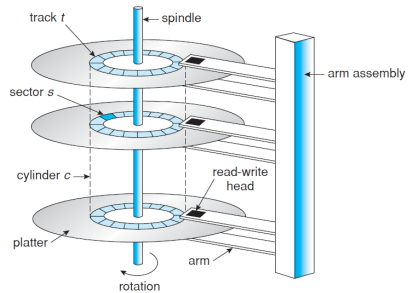
# Mass Storage Structure (1/2)

- ▶ Magnetic disks: bulk of **secondary storage**



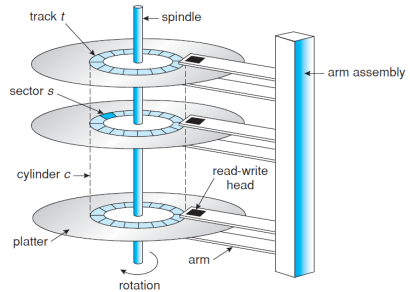
## Mass Storage Structure (1/2)

- ▶ Magnetic disks: bulk of secondary storage
- ▶ Disk platter is a flat circular shape, covered with a magnetic material.



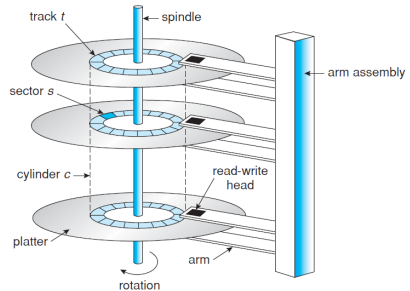
## Mass Storage Structure (1/2)

- ▶ Magnetic disks: bulk of secondary storage
- ▶ Disk platter is a flat circular shape, covered with a magnetic material.
- ▶ Heads are attached to a disk arm.



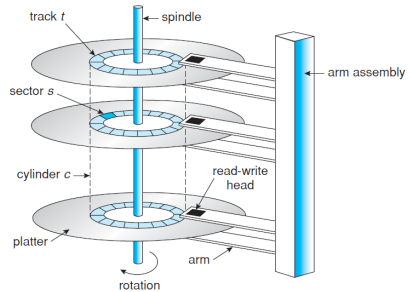
# Mass Storage Structure (1/2)

- ▶ **Magnetic disks**: bulk of **secondary storage**
- ▶ Disk **platter** is a **flat circular shape**, covered with a **magnetic material**.
- ▶ **Heads** are attached to a **disk arm**.
- ▶ The surface of a platter is **logically** divided into circular **tracks**, which are subdivided into **sectors**.



# Mass Storage Structure (1/2)

- ▶ **Magnetic disks**: bulk of **secondary storage**
- ▶ Disk **platter** is a **flat circular shape**, covered with a **magnetic material**.
- ▶ **Heads** are attached to a **disk arm**.
- ▶ The surface of a platter is **logically** divided into circular **tracks**, which are subdivided into **sectors**.
- ▶ The set of tracks that are at **one arm position** makes up a **cylinder**.





## Mass Storage Structure (2/2)

- ▶ Drives **rotate** at 60 to 250 times per second.





## Mass Storage Structure (2/2)

- ▶ Drives **rotate** at **60 to 250** times per second.
- ▶ **Transfer rate**: the rate at which data **flow between drive and computer**.



## Mass Storage Structure (2/2)

- ▶ Drives **rotate** at **60 to 250** times per second.
- ▶ **Transfer rate**: the rate at which data **flow between drive and computer**.
- ▶ **Positioning time**: the time to **move disk arm to desired cylinder** (**seek time**) and time for **desired sector to rotate** under the disk head (**rotational latency**).

# The First Commercial Disk Drive

- ▶ IBM, 1956
- ▶ 5M
- ▶ Access time  $\leq 1$  second





## Solid-State Disks (SSDs)

- ▶ **Non-volatile memory** used like a hard drive.



## Solid-State Disks (SSDs)

- ▶ **Non-volatile memory** used like a hard drive.
- ▶ More **expensive** per MB.



## Solid-State Disks (SSDs)

- ▶ **Non-volatile memory** used like a hard drive.
- ▶ More **expensive** per MB.
- ▶ Maybe have **shorter life** span.



## Solid-State Disks (SSDs)

- ▶ **Non-volatile memory** used like a hard drive.
- ▶ More **expensive** per MB.
- ▶ Maybe have **shorter life** span.
- ▶ **Less capacity**, but much **faster**.



## Solid-State Disks (SSDs)

- ▶ **Non-volatile memory** used like a hard drive.
- ▶ More **expensive** per MB.
- ▶ Maybe have **shorter life** span.
- ▶ **Less capacity**, but much **faster**.
- ▶ **No moving parts**, so no seek time or rotational latency.





# Magnetic Tape

- ▶ Early secondary-storage medium.



# Magnetic Tape

- ▶ **Early** secondary-storage medium.
- ▶ **Relatively permanent** and holds **large quantities** of data.



# Magnetic Tape

- ▶ **Early** secondary-storage medium.
- ▶ **Relatively permanent** and holds **large quantities** of data.
- ▶ Access time **slow**.



# Magnetic Tape

- ▶ Early secondary-storage medium.
- ▶ Relatively permanent and holds large quantities of data.
- ▶ Access time slow.
- ▶ Random access  $\sim$  1000 times slower than disk.



# Magnetic Tape

- ▶ **Early** secondary-storage medium.
- ▶ **Relatively permanent** and holds **large quantities** of data.
- ▶ Access time **slow**.
- ▶ **Random access**  $\sim$  **1000 times slower** than disk.
- ▶ Mainly used for **backup**, storage of infrequently-used data.



# Magnetic Tape

- ▶ **Early** secondary-storage medium.
- ▶ **Relatively permanent** and holds **large quantities** of data.
- ▶ Access time **slow**.
- ▶ **Random access**  $\sim$  **1000 times slower** than disk.
- ▶ Mainly used for **backup**, storage of infrequently-used data.
- ▶ Once data under **head**, transfer rates comparable to disk.

# Disk Structure



## Disk Structure (1/2)

- ▶ Disk drives are addressed as large **1-dimensional arrays** of **logical blocks**.





## Disk Structure (1/2)

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



## Disk Structure (1/2)

- ▶ Disk drives are addressed as large **1-dimensional arrays** of **logical blocks**.
- ▶ The **logical block** is the **smallest unit of transfer**.
- ▶ **Low-level formatting** creates **logical blocks** on **physical media**.



## Disk Structure (2/2)

- ▶ The array of **logical blocks** is mapped into the **sectors** of the disk **sequentially**.



## Disk Structure (2/2)

- ▶ The 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**.



## Disk Structure (2/2)

- ▶ The 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 Structure (2/2)

- ▶ The 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**.
- ▶ **Logical to physical** address should be easy.



# Disk Attachment



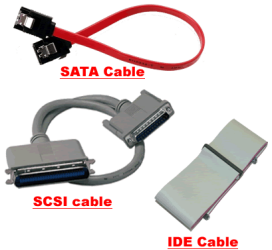
# Disk Attachment

- ▶ Host-attached storage
- ▶ Network-attached storage (NAS)
- ▶ Storage-area network (SAN)



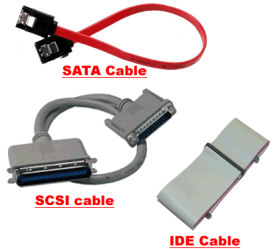
# Host-Attached Storage

- ▶ **Host-attached storage** accessed through I/O ports talking to I/O buses.



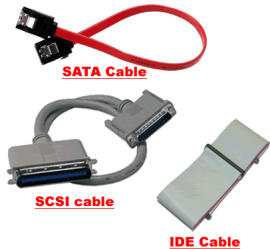
# Host-Attached Storage

- ▶ **Host-attached storage** accessed through **I/O ports** talking to **I/O buses**.
- ▶ **IDE or SATA** support max. **two drives** per I/O bus.



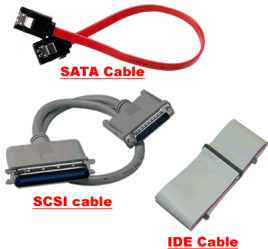
# Host-Attached Storage

- ▶ **Host-attached storage** accessed through **I/O ports** talking to **I/O buses**.
- ▶ **IDE or SATA** support max. **two drives** per I/O bus.
- ▶ **SCSI**, up to **16 devices** on one cable.



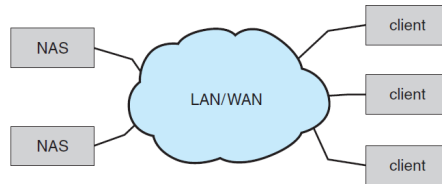
# Host-Attached Storage

- ▶ **Host-attached storage** accessed through **I/O ports** talking to **I/O buses**.
- ▶ **IDE or SATA** support max. **two drives** per I/O bus.
- ▶ **SCSI**, up to **16 devices** on one cable.
- ▶ **Fiber Channel (FC)** is high-speed serial architecture.



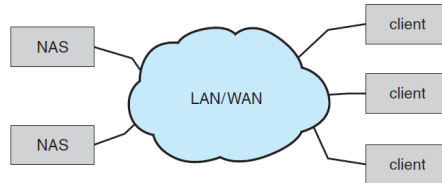
# Network-Attached Storage (NAS)

- ▶ **Network-attached storage** is storage made available over a network.



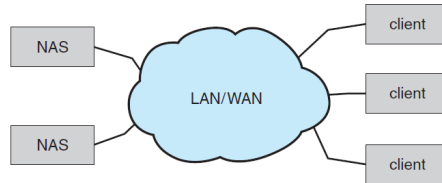
# Network-Attached Storage (NAS)

- ▶ **Network-attached storage** is storage made available over a network.
- ▶ **Remotely** attaching to file systems.



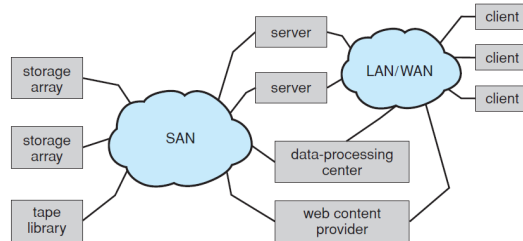
# Network-Attached Storage (NAS)

- ▶ **Network-attached storage** is storage made available over a network.
- ▶ **Remotely** attaching to file systems.
- ▶ **FTP**, **NFS** and **SMB** are common protocols.



# Storage-Area Network (SAN)

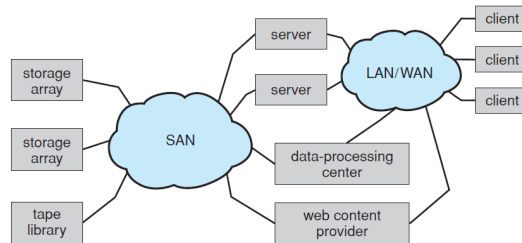
- ▶ **Storage-area network** is common in **large storage environments**.



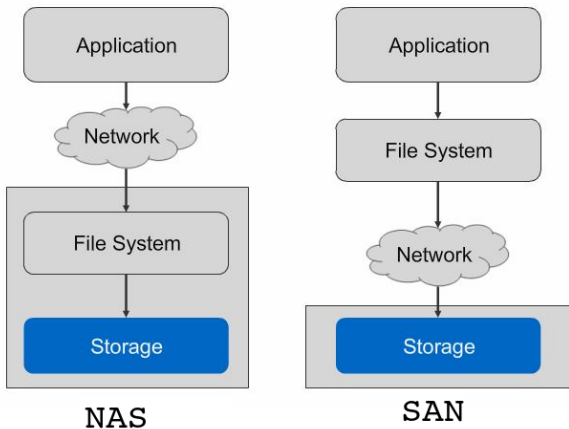


# Storage-Area Network (SAN)

- ▶ **Storage-area network** is common in **large storage environments**.
- ▶ **Multiple hosts** attached to **multiple storage arrays**.



# NAS vs. SAN



# Disk Management



## Disk Formatting (Physical)

- ▶ **Physical formatting:** dividing a disk into **sectors** that the **disk controller** can read and write.



## Disk Formatting (Physical)

- ▶ **Physical formatting**: dividing a disk into **sectors** that the **disk controller** can read and write.
- ▶ Each **sector** can hold **header** information, **data**, and **error correction code**.



## Disk Formatting (Physical)

- ▶ **Physical formatting**: dividing a disk into **sectors** that the **disk controller** can read and write.
- ▶ Each **sector** can hold **header** information, **data**, and **error correction code**.
- ▶ To use a disk to **hold files**, the OS needs to record **its own data structures** on the disk.



## Disk Formatting (Logical)

- ▶ Logical formatting or making a **file system**.



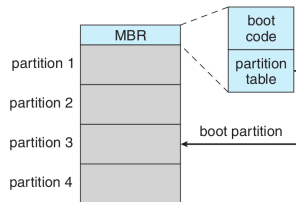
## Disk Formatting (Logical)

- ▶ Logical formatting or making a **file system**.
- ▶ **Partition**: one or more groups of **cylinders**, each treated as a **logical disk**.



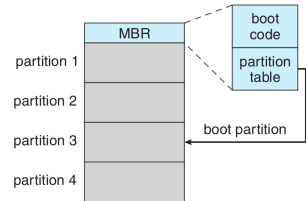
# Boot Block

- ▶ The **bootstrap program**: **initializes** a computer when it is powered up and starts the OS.



# Boot Block

- ▶ The **bootstrap program**: **initializes** a computer when it is powered up and starts the OS.
- ▶ The **bootstrap program** is stored in the **boot blocks** at a **fixed location on the disk**.





# Disk Scheduling



# Disk Scheduling

- ▶ There are many sources of **disk I/O request**, e.g., OS, system processes, users processes.



# Disk Scheduling

- ▶ There are many sources of **disk I/O request**, e.g., OS, system processes, users processes.
- ▶ OS maintains **queue of requests**, per disk or device.



# Disk Scheduling

- ▶ There are many sources of **disk I/O request**, e.g., OS, system processes, users processes.
- ▶ OS maintains **queue of requests**, per disk or device.
- ▶ **Idle disk** can immediately work on **I/O request**, **busy disk** means work must **queue**.



# Disk Scheduling Algorithms

- ▶ First Come First Serve (FCFS)
- ▶ Shortest Seek Time First (SSTF)
- ▶ SCAN
- ▶ C-SCAN
- ▶ C-Look



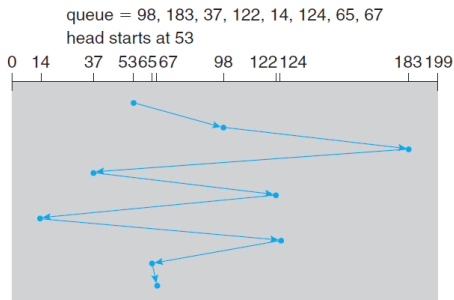
## FCFS

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



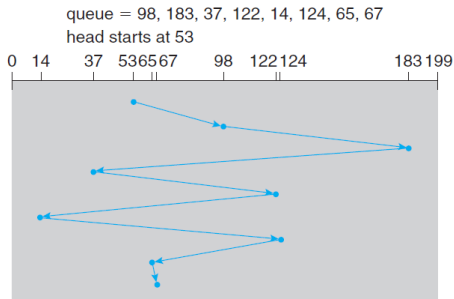
# FCFS

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



# FCFS

- ▶ Request queue (0-199): 98, 183, 37, 122, 14, 124, 65, 67
- ▶ Head pointer 53
- ▶ Total head movement: 640 cylinders





## SSTF

- ▶ Selects the request with the **minimum seek time** from the **current head position**.

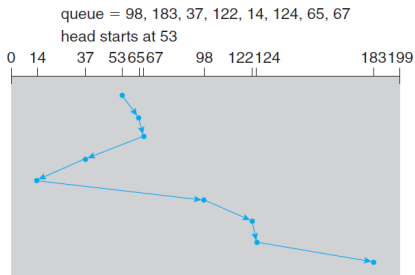


## SSTF

- ▶ 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.

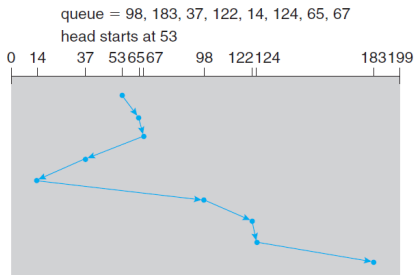
# SSTF

- ▶ 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.



# SSTF

- ▶ 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.
- ▶ Total head movement: **236 cylinders**.



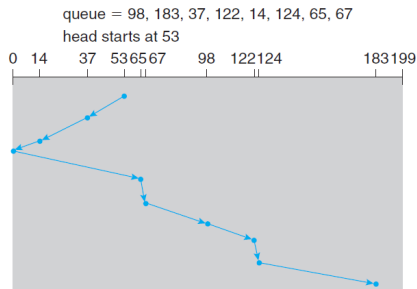


# SCAN

- ▶ Starts from **one end** of the disk, and moves toward **the other end**.
  - Servicing requests until it **gets to the other end** of the disk.
  - At the end of the dist, the head **movement is reversed**.

# SCAN

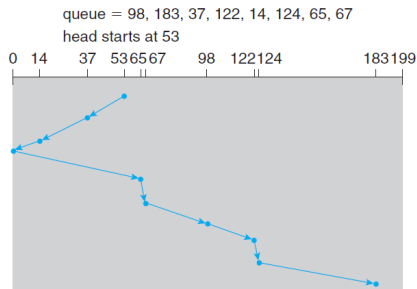
- ▶ Starts from **one end** of the disk, and moves toward **the other end**.
  - Servicing requests until it **gets to the other end** of the disk.
  - At the end of the dist, the head **movement is reversed**.





# SCAN

- ▶ Starts from **one end** of the disk, and moves toward **the other end**.
  - Servicing requests until it **gets to the other end** of the disk.
  - At the end of the dist, the head **movement is reversed**.
- ▶ Total head movement: **236 cylinders**





## C-SCAN

- ▶ Provides a **more uniform wait** time than SCAN.

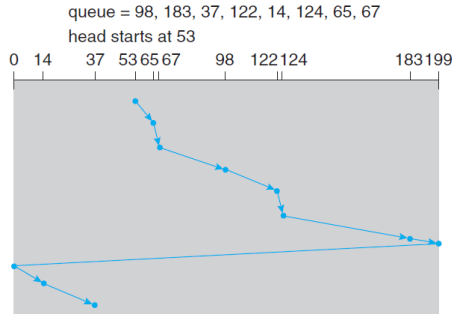


## C-SCAN

- ▶ Provides a **more uniform wait** time than SCAN.
- ▶ When it reaches the end, it immediately returns to the beginning of the disk **without servicing any requests** on the **return trip**.

# C-SCAN

- ▶ Provides a **more uniform wait** time than SCAN.
- ▶ When it reaches the end, it immediately returns to the beginning of the disk **without servicing any requests** on the **return trip**.





## Look

- ▶ LOOK is a version of SCAN, C-LOOK is a version of C-SCAN.

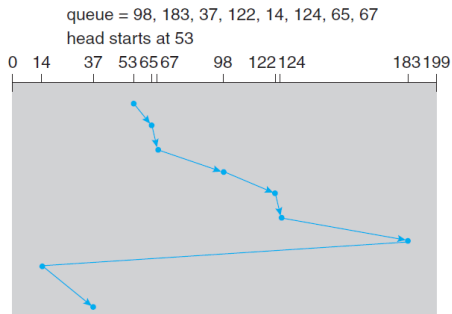


## Look

- ▶ LOOK is a version of SCAN, C-LOOK is a 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.

# Look

- ▶ LOOK is a version of SCAN, C-LOOK is a 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.





## Selecting a Disk-Scheduling Algorithm (1/2)

- ▶ Having a fast access time and disk bandwidth.





## Selecting a Disk-Scheduling Algorithm (1/2)

- ▶ Having a fast access time and disk bandwidth.
- ▶ Minimize seek time.



## Selecting a Disk-Scheduling Algorithm (1/2)

- ▶ Having a fast access time and disk bandwidth.
- ▶ Minimize seek time.
- ▶ Disk bandwidth is the total bytes transferred, divided by the total time between the first request and the completion of the last transfer.



## Selecting a Disk-Scheduling Algorithm (2/2)

- ▶ SSTF is common and has a natural appeal: good performance



## Selecting a Disk-Scheduling Algorithm (2/2)

- ▶ SSTF is common and has a natural appeal: **good performance**
- ▶ SCAN and C-SCAN perform better for systems that place a **heavy load** on the disk: **less starvation**



## Selecting a Disk-Scheduling Algorithm (2/2)

- ▶ SSTF is common and has a natural appeal: good performance
- ▶ SCAN and C-SCAN perform better for systems that place a heavy load on the disk: less starvation
- ▶ Performance depends on the number and types of requests.

# RAID Structure



# Failure and Reliability

- ▶ Multiple disk drives provides reliability via redundancy.



## Failure and Reliability

- ▶ Multiple disk drives provides reliability via redundancy.
- ▶ Increases the mean time to failure.





## Failure and Reliability

- ▶ Multiple disk drives provides reliability via redundancy.
- ▶ Increases the mean time to failure.
  - E.g., if the mean time to failure of a single disk is 100,000 hours.



# Failure and Reliability

- ▶ Multiple disk drives provides reliability via redundancy.
- ▶ Increases the mean time to failure.
  - E.g., if the mean time to failure of a single disk is 100,000 hours.
  - The mean time to failure of some disk in an array of 100 disks will be  $100,000/100 = 1,000$  hours, or 41.66 days



# Failure and Reliability

- ▶ Multiple disk drives provides reliability via redundancy.
- ▶ Increases the mean time to failure.
  - E.g., if the mean time to failure of a single disk is 100,000 hours.
  - The mean time to failure of some disk in an array of 100 disks will be  $100,000/100 = 1,000$  hours, or 41.66 days
  - It is not long at all.



# Mirroring

- ▶ The simplest approach to introducing **redundancy** is to **duplicate every disk**, called **mirroring**.



# Mirroring

- ▶ The simplest approach to introducing **redundancy** is to **duplicate every disk**, called **mirroring**.
- ▶ A **logical disk** consists of **two physical disks**, and every write is carried out on **both disks**.



# Mirroring

- ▶ The simplest approach to introducing **redundancy** is to **duplicate every disk**, called **mirroring**.
- ▶ A **logical disk** consists of **two physical disks**, and every write is carried out on **both disks**.
- ▶ If one of the disks in the volume **fails**, the data can be read **from the other**.



## Improvement in Performance via Parallelism

- ▶ **Disk striping** uses a **group of disks** as **one storage unit**.



## Improvement in Performance via Parallelism

- ▶ **Disk striping** uses a **group of disks** as **one storage unit**.
- ▶ **Bit-level striping**: **splitting the bits of each byte** across multiple disks.
  - E.g., with  $n$  disks, bit  $i$  of a file goes to disk  $(i \bmod n) + 1$ .





## Improvement in Performance via Parallelism

- ▶ **Disk striping** uses a **group of disks** as **one storage unit**.
- ▶ **Bit-level striping**: **splitting the bits of each byte** across multiple disks.
  - E.g., with  $n$  disks, bit  $i$  of a file goes to disk  $(i \bmod n) + 1$ .
- ▶ **Block-level striping**: **blocks of a file** are striped across multiple disks.



## RAID Levels

- ▶ **RAID**: redundant array of inexpensive disks



## RAID Levels

- ▶ **RAID**: redundant array of inexpensive disks
- ▶ RAID schemes improve **performance** and improve the **reliability** of the storage system by storing **redundant data**.

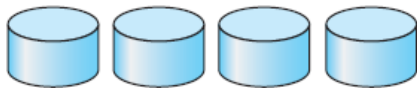


## RAID Levels

- ▶ **RAID**: redundant array of inexpensive disks
- ▶ RAID schemes improve **performance** and improve the **reliability** of the storage system by storing **redundant data**.
- ▶ RAID is arranged into **six different levels**.

## RAID Level 0

- ▶ Disk arrays with **striping at the level of blocks** but **without any redundancy**.



# RAID Level 1

- ▶ Disk mirroring



## RAID Level 4

- ▶ Block-level striping, as in RAID 0.



## RAID Level 4

- ▶ Block-level striping, as in RAID 0.
- ▶ Error-correcting code (ECC)





## RAID Level 4

- ▶ Block-level striping, as in RAID 0.
- ▶ Error-correcting code (ECC)
- ▶ Keeps ECC on a separate disk for corresponding blocks from N other disks.



## RAID Level 5

- ▶ Spreads **data and ECC** among all  $N+1$  disks, rather than storing data in  $N$  disks and parity in one disk.



## RAID Level 6

- ▶ Like RAID level 5 but stores **extra redundant information** to guard against **multiple disk failures**.



# Summary



## Summary

- ▶ Mass storage structure: platter, track, sector, cylinder



## Summary

- ▶ Mass storage structure: platter, track, sector, cylinder
- ▶ Disk attachment: host-attached, network-attached, storage-area-network



## Summary

- ▶ Mass storage structure: platter, track, sector, cylinder
- ▶ Disk attachment: host-attached, network-attached, storage-area-network
- ▶ Disk scheduling: FCFS, SSTF, SCAN, C-SCAN, C-Look



## Summary

- ▶ Mass storage structure: platter, track, sector, cylinder
- ▶ Disk attachment: host-attached, network-attached, storage-area-network
- ▶ Disk scheduling: FCFS, SSTF, SCAN, C-SCAN, C-Look
- ▶ Disk management: formatting, boot block





## Summary

- ▶ Mass storage structure: platter, track, sector, cylinder
- ▶ Disk attachment: host-attached, network-attached, storage-area-network
- ▶ Disk scheduling: FCFS, SSTF, SCAN, C-SCAN, C-Look
- ▶ Disk management: formatting, boot block
- ▶ RAID: RAID0-RAID6

# Questions?

## Acknowledgements

Some slides were derived from Avi Silberschatz slides.