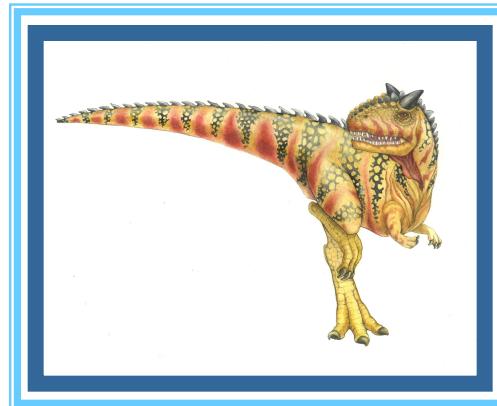


Mass-Storage Systems

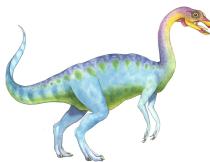




Chapter 11: Mass-Storage Systems

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





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





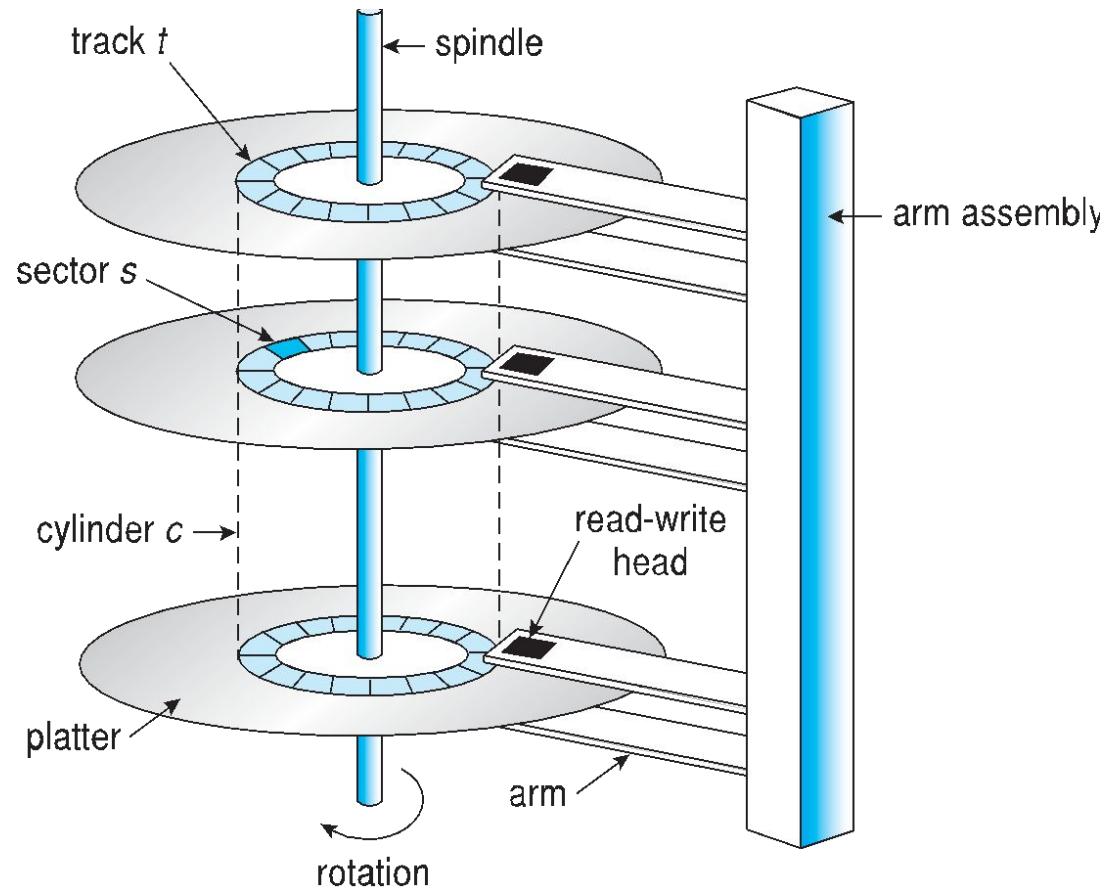
Overview of Mass Storage Structure

- Bulk of secondary storage for modern computers is **hard disk drives (HDDs)** and **nonvolatile memory (NVM)** provided by devices **Magnetic disks**
- Conceptually, disks are relatively simple (Figure 10.1). Each **disk platter** has a flat **circular shape**, like a CD, with both surfaces covered with a magnetic material
- **HDDs** spin platters under moving read-write heads
- 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 thousands of concentric circular **tracks forming a cylinder(read-write head)**, which are subdivided into hundreds of **sectors**.
- The set of tracks that are at one arm position makes up a **cylinder**.
- The storage capacity of common disk drives is measured in gigabytes.
- Disks can be removable allowing mounting
- Other forms of removable disks include CDs, DVDs, and Blu-ray discs as well as removable flash-memory devices known as **flash drives** (type of solid-state drive).





Moving-head Disk Mechanism





Disk Speed

- Drive motor rotates at 60 to 250 times per second
- Disk speed has two parts.
 - The transfer rate is the rate at which data flow between the drive and the computer.
 - The positioning time, or random-access time, consists of two parts:
 - the time necessary to move the disk arm to the desired cylinder, called the seek time,
 - and the time necessary for the desired sector to rotate to the disk head, called the rotational latency.
- Typical disks can transfer several megabytes of data per second, and they have seek times and rotational latencies of several milliseconds.
- Head crash - As the disk head flies on an extremely thin cushion of air, there is a danger that the head will make contact with the disk surface. Although the disk platters are coated with a thin protective layer, the head will sometimes damage the magnetic surface. This accident is called a head crash





Hard Disks Attachment

- A disk drive is attached to a computer by a set of wires called an **I/O bus**.
- Several kinds of buses are available, including **advanced technology attachment (ATA)**, **serial ATA (SATA)**, **eSATA**, **universal serial bus (USB)**, and **fibre channel (FC)**.
- Most common is SATA
- The data transfers on a bus are carried out by special electronic processors called **controllers**.
 - The **host controller** is the controller at the computer end of the bus.
 - A **disk controller** is built into each disk drive.
- To perform a disk I/O operation,
 - 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 disk controller
 - the disk controller operates the disk-drive hardware to carry out the command.
- Disk controllers usually have a built-in cache.
- Data transfer at the disk drive happens between the cache and the disk surface, and data transfer to the host, at fast electronic speeds, occurs between the cache and the host controller.





Nonvolatile Memory Devices

- **solid-state disks (SSDs)** - nonvolatile memory that is used like a hard drive with many variations, from DRAM with a battery to allow it to maintain its state in a power failure through flash-memory technologies like single-level cell (SLC) and multilevel cell (MLC) chips.
- Features
 - consume less power
 - Can be more reliable than HDDs as they have no moving parts so no seek time or rotational latency and so much faster
 - More expensive per MB
 - Maybe have shorter life span – need careful management
 - Less capacity
 - Busses can be too slow -> connect directly to PCI for example
- Use –
 - storage arrays for holding file-system metadata that require high performance.
 - In some laptop computers to make them smaller, faster, and more energy-efficient.





- Other forms include **USB drives** (thumb drive, flash drive), DRAM disk replacements, surface-mounted on motherboards, and main storage in devices like smartphones
- standard bus interfaces can cause a major limit on throughput of SSD. Some SSDs are designed to connect directly to the system bus (PCI, for example).





Magnetic Tape

Magnetic tape was used as an early secondary-storage medium. Although it is nonvolatile and can hold large quantities of data, its access time is slow compared with that of main memory and drives. In addition, random access to magnetic tape is about a thousand times slower than random access to HDDs and about a hundred thousand times slower than random access to SSDs so tapes are not very useful for secondary storage. Tapes are used mainly for backup, for storage of infrequently used information, and as a medium for transferring information from one system to another.

A tape is kept in a spool and is wound or rewound past a read–write head. Moving to the correct spot on a tape can take minutes, but once positioned, tape drives can read and write data at speeds comparable to HDDs. Tape capacities vary greatly, depending on the particular kind of tape drive, with current capacities exceeding several terabytes. Some tapes have built-in compression that can more than double the effective storage. Tapes and their drivers are usually categorized by width, including 4, 8, and 19 millimeters and 1/4 and 1/2 inch. Some are named according to technology, such as LTO-6 (Figure 11.5) and SDLT.



Figure 11.5 An LTO-6 Tape drive with tape cartridge inserted.





HDD Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and large disk bandwidth
- Minimize seek time
- Seek time \approx seek distance
- Disk **bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer
- **Objective** - improve both the access time and the bandwidth by managing the order in which disk I/O requests are serviced
- sources of disk I/O request – OS, System processes, Users processes
- Whenever a process needs I/O to or from the disk, it issues a system call to the operating system. Information specified in requests include –
 - Whether this operation is input or output
 - What the disk address for the transfer is
 - What the memory address for the transfer is
 - What the number of sectors to be transferred is

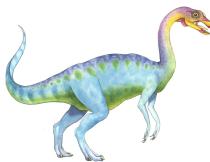




Disk Scheduling (Cont.)

- 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
- In the past, operating system responsible for queue management, disk drive head scheduling
 - Now, built into the storage devices, controllers
 - Just provide LBAs, handle sorting of requests
 - 4 Some of the algorithms they use described next





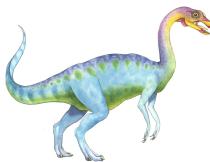
Disk Scheduling (Cont.)

- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The 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



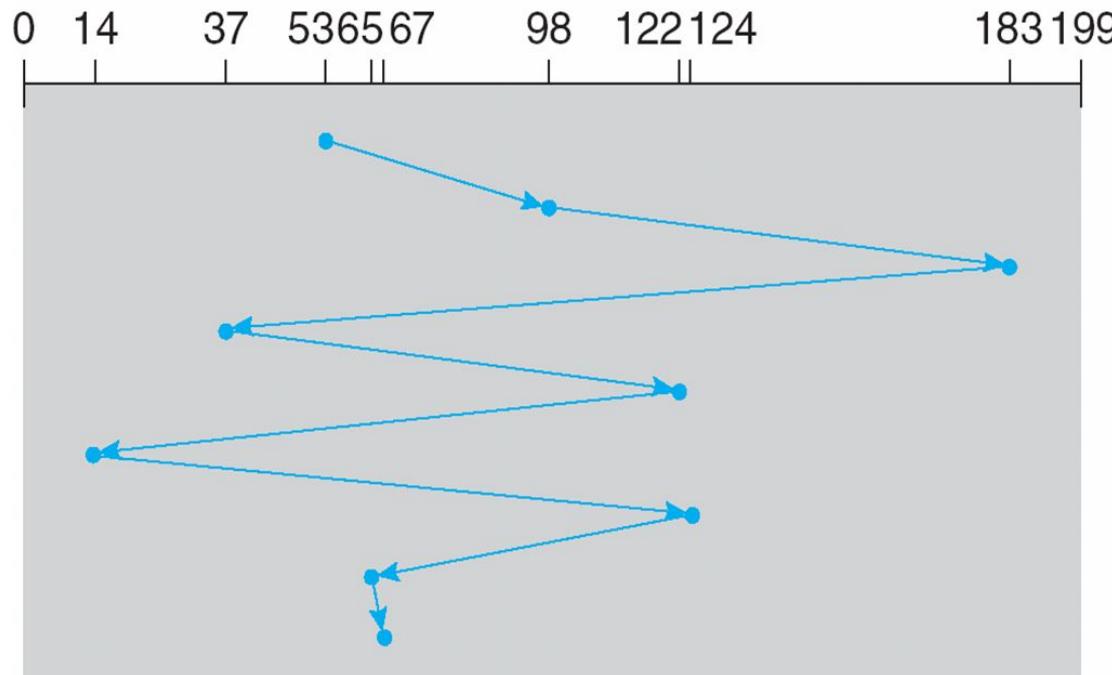


FCFS

Illustration shows total head movement of 640 cylinders

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

head starts at 53



The wild swing from 122 to 14 and then back to 124 illustrates the problem with this schedule. If the requests for cylinders 37 and 14 could be serviced together, before or after the requests for 122 and 124, the total head movement could be decreased substantially, and performance could be thereby improved

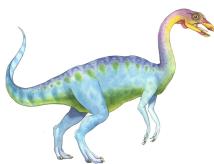




SSTF Scheduling

- It seems reasonable to service all the requests close to the current head position before moving the head far away to service other requests.
- The **Shortest Seek Time First algorithm** selects the request with the least seek time from the current head position. It chooses the pending request closest to the current head position.
- SSTF scheduling is essentially a form of shortest-job-first (SJF) scheduling; and like SJF scheduling, it may cause starvation of some requests.
- Remember
- As the requests may arrive at any time.
 - two requests in the queue, for cylinders 14 and 186, and while the request from 14 is being serviced, a new request near 14 arrives. This new request will be serviced next, making the request at 186 wait. This could cause the request for cylinder 186 to wait indefinitely.

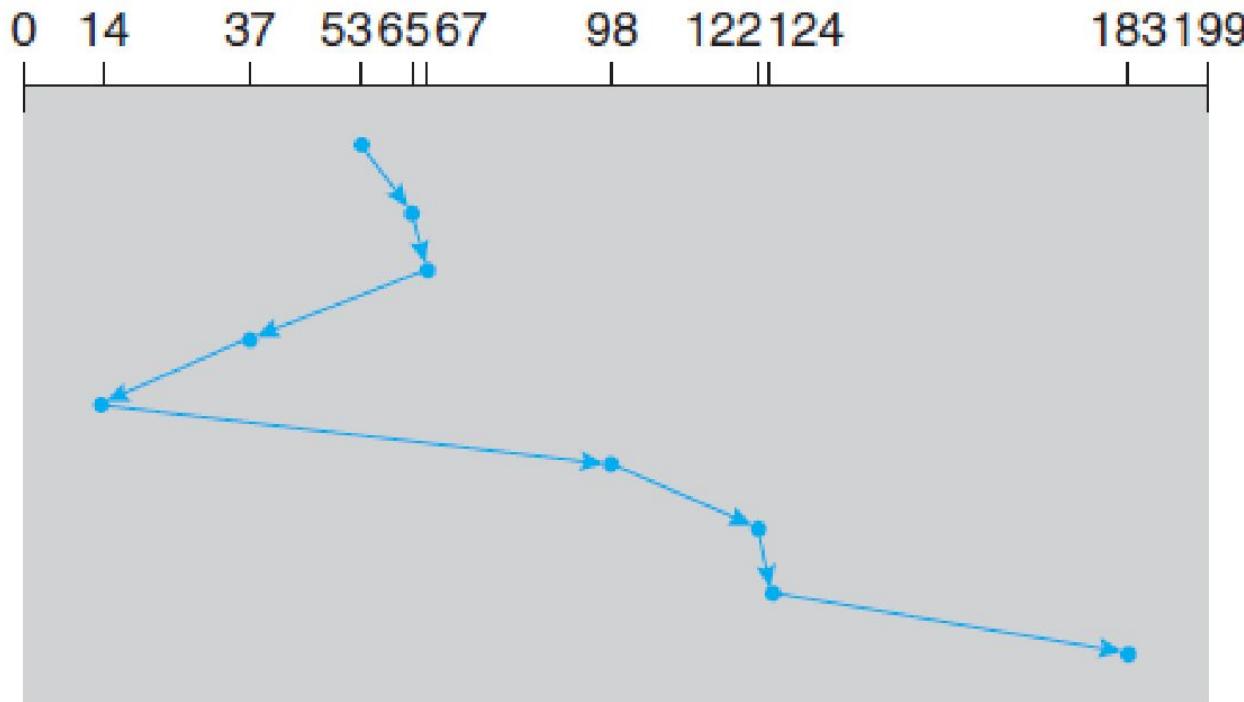




SSTF Scheduling Contd.

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

head starts at 53



Although the SSTF algorithm is a substantial improvement over the FCFS algorithm, it is not optimal.

For example moving the head from 53 to 37, even though the latter is not closest, and then to 14, before turning around to service 65, 67, 98, 122, 124, and 183. This strategy reduces the total head movement to 208 cylinders.

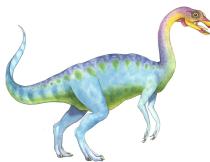




SCAN

- In the **SCAN algorithm**, the disk arm starts at one end of the disk and moves toward the other end, servicing requests as it reaches each cylinder, until it gets to the other end of the disk. At the other end, the direction of head movement is reversed, and servicing continues. The head continuously scans back and forth across the disk. **SCAN algorithm**
- Sometimes called the **elevator algorithm**
- If a request arrives in the queue just in front of the head, it will be serviced almost immediately; a request arriving just behind the head will have to wait until the arm moves to the end of the disk, reverses direction, and comes back.
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest
- Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction. At this point, relatively few requests are immediately in front of the head, since these cylinders have recently been serviced. The heaviest density of requests is at the other end of the disk. These requests have also waited the longest, so why not go there first? That is the idea of the next algorithm.





SCAN (Cont.)

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

head starts at 53

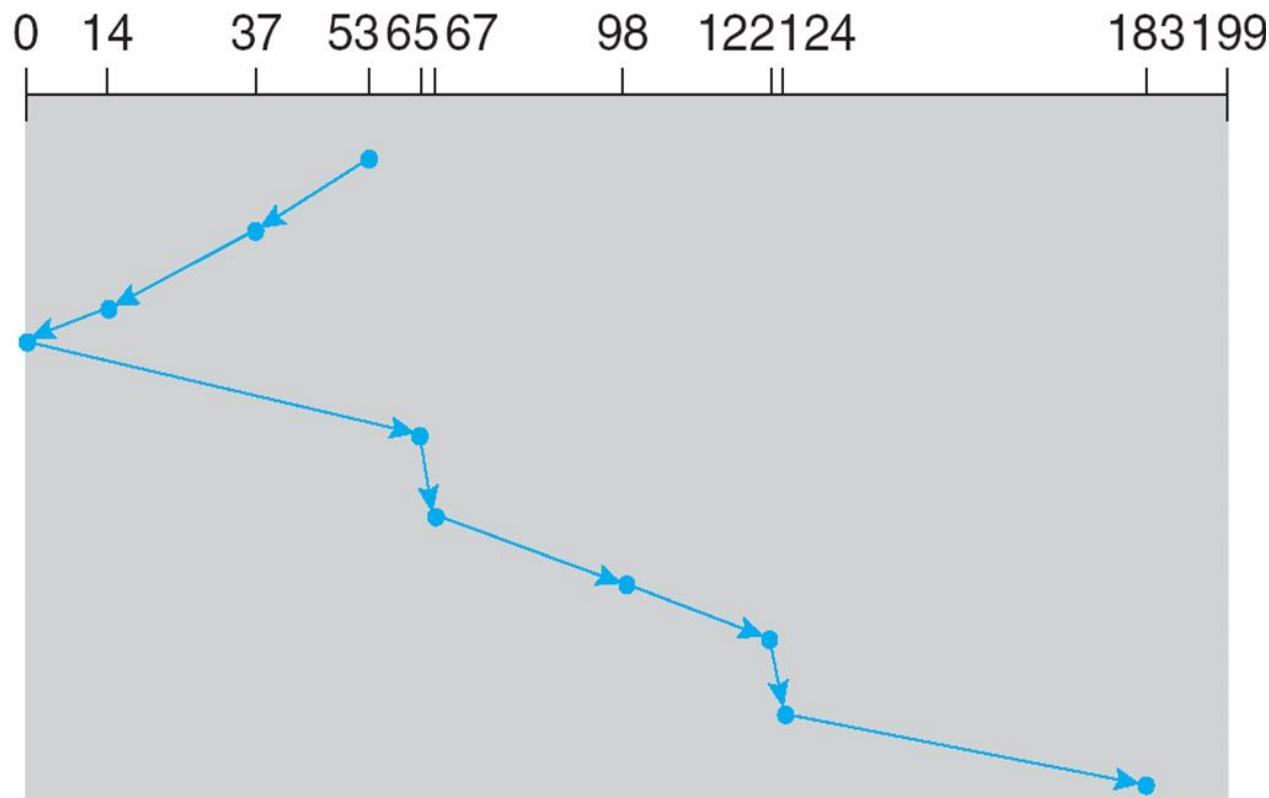


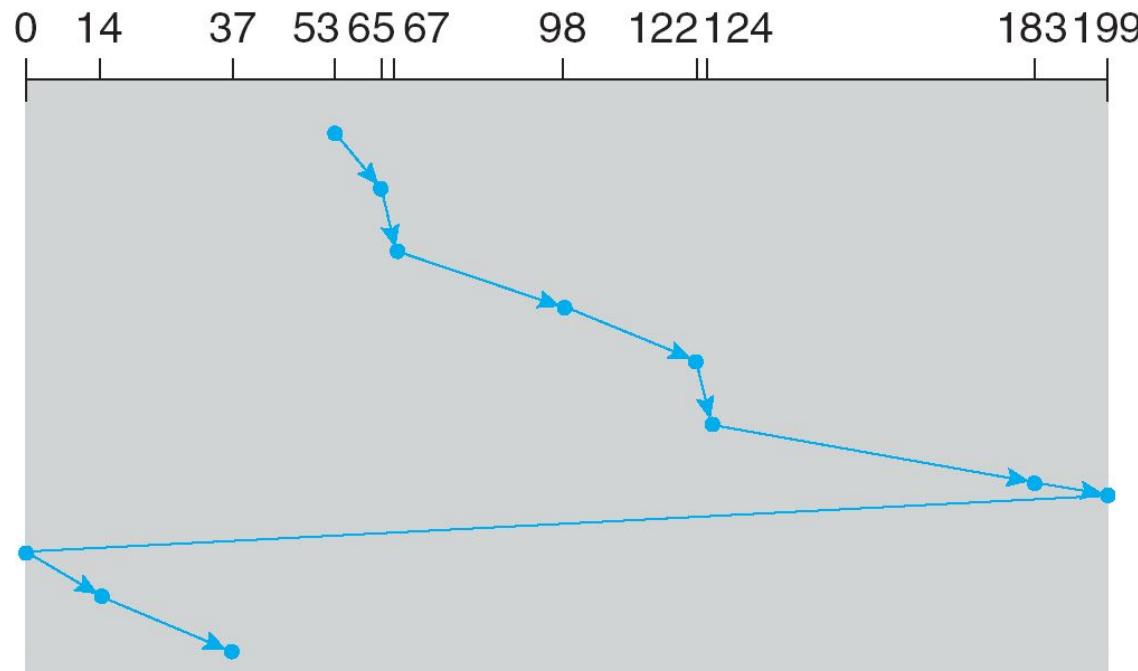
Illustration shows total head movement of 208 cylinders





C-SCAN

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other, servicing requests as it goes
 - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Total number of cylinders?
 - queue = 98, 183, 37, 122, 14, 124, 65, 67
 - head starts at 53





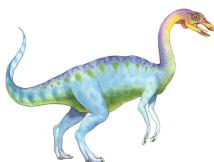
LOOK and C-LOOK

In practice, neither algorithm is often implemented this way.

The arm goes only as far as the final request in each direction. Then, it reverses direction immediately, without going all the way to the end of the disk.

This Versions of SCAN and C-SCAN that follow this pattern are called **LOOK** and **C-LOOK scheduling**, because they *look* for a request before continuing to move in a given direction





LOOK and C-LOOK Contd.

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

head starts at 53

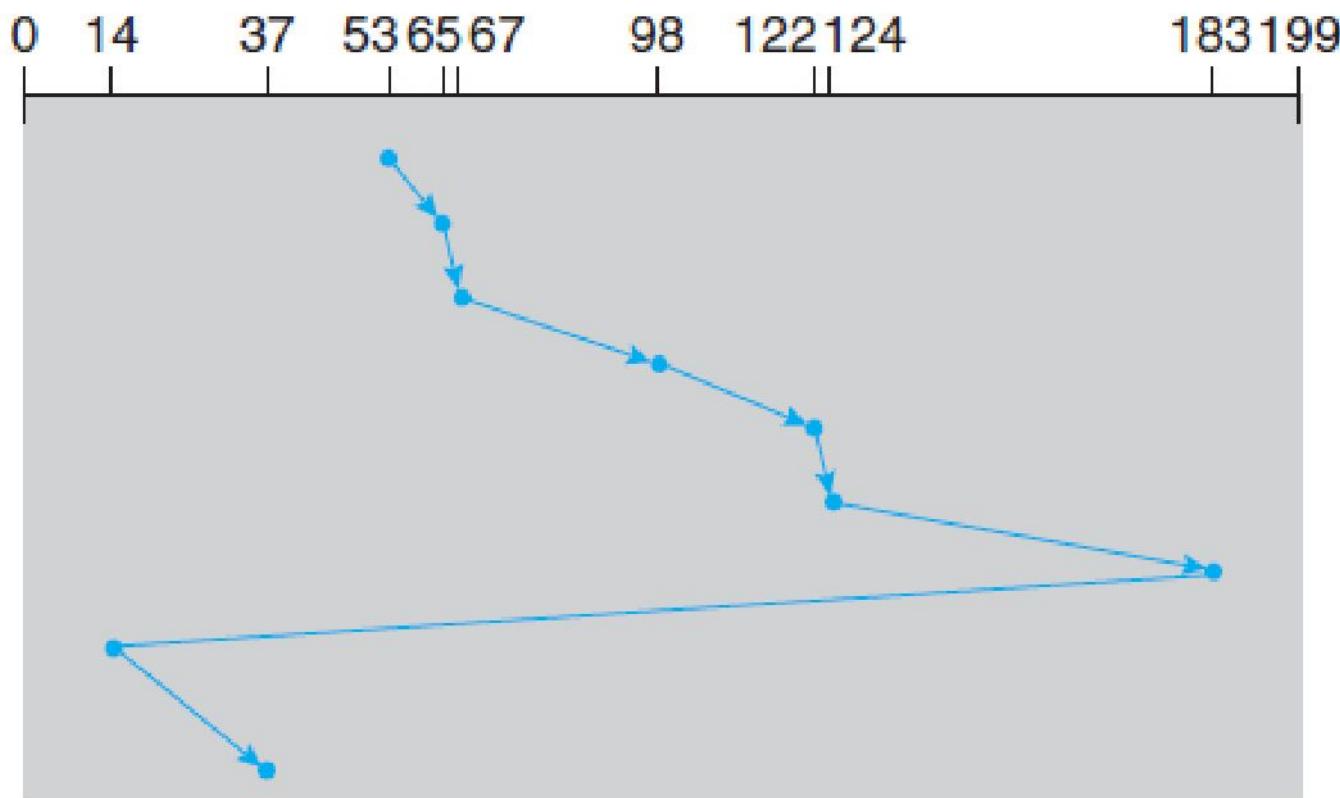


Figure 10.8 C-LOOK disk scheduling.





Question..

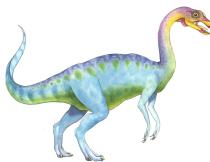




Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
 - Less starvation, but still possible
- To avoid starvation Linux implements **deadline** scheduler
 - Maintains separate read and write queues, gives read priority
 - 4 Because processes more likely to block on read than write
 - Implements four queues: 2 x read and 2 x write
 - 4 1 read and 1 write queue sorted in LBA order, essentially implementing C-SCAN
 - 4 1 read and 1 write queue sorted in FCFS order
 - 4 All I/O requests sent in batch sorted in that queue's order
 - 4 After each batch, checks if any requests in FCFS older than configured age (default 500ms)
 - If so, LBA queue containing that request is selected for next batch of I/O





NVM Scheduling

- No disk heads or rotational latency but still room for optimization
- In RHEL 7 **NOOP** (no scheduling) is used but adjacent LBA requests are combined
 - NVM best at random I/O, HDD at sequential
 - Throughput can be similar
 - **Input/Output operations per second (IOPS)** much higher with NVM (hundreds of thousands vs hundreds)
 - But **write amplification** (one write, causing garbage collection and many read/writes) can decrease the performance advantage





Address Mapping

- Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer
 - Low-level formatting creates **logical blocks** on physical media
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially
 - Sector 0 is the first sector of the first track on the outermost cylinder
 - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost
 - This mapping can convert a logical block number into an old-style disk address that consists of a cylinder number, a track number within that cylinder, and a sector number within that track. Can be sometimes difficult due to
 - 4 bad sectors
 - 4 Non-constant # of sectors per track via constant angular velocity - **the disk rotation speed can stay constant; the density of bits decreases from inner tracks to outer tracks to keep the data rate constant.** This method is used in hard disks and is known as **constant angular velocity (CAV)**.
 - Consider media that use constant linear velocity (CLV) - **the density of bits per track is uniform** - The drive increases its rotation speed as the head moves from the outer to the inner tracks to keep the same rate of data moving under the head. Used in CDs





Storage Attachment

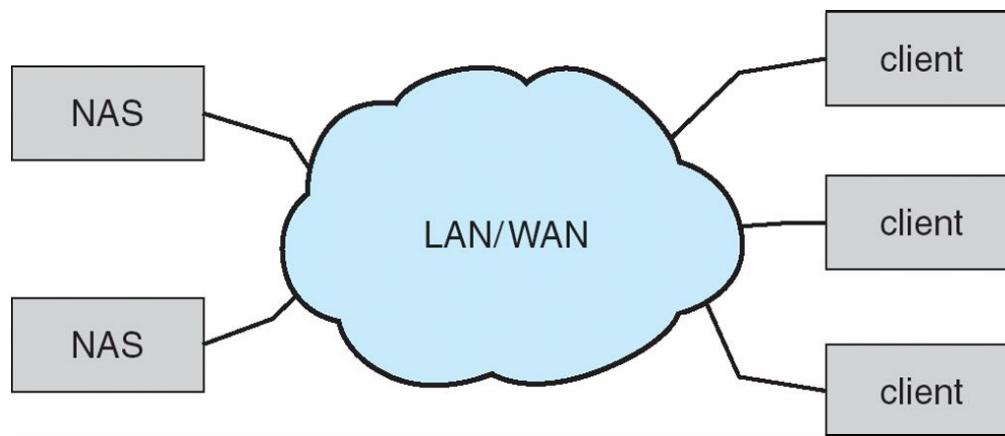
- Computers access storage in three ways
 - host-attached
 - network-attached
 - cloud
- Host attached access through local I/O ports, using one of several technologies
 - To attach many devices, use storage busses such as USB, firewire, thunderbolt
 - This architecture supports a maximum of two drives per I/O bus. A newer, similar protocol that has simplified cabling is SATA
 - High-end systems use **fibre channel (FC)**
 - 4 High-speed serial architecture using fibre or copper cables
 - 4 Because of the large address space and the switched nature of the communication, multiple hosts and storage devices can connect to the FC fabric





Network-Attached Storage

- Network-attached storage (**NAS**) is storage made available over a network rather than over a local connection (such as a bus)
 - Remotely attaching to file systems
- NFS (linux) and CIFS (Windows) 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 the SCSI protocol
 - hosts can treat their storage as if it were directly attached, even if the storage is distant from the host

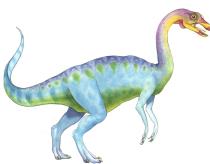




Network-Attached Storage Contd.

- NAS provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.
 - It tends to be less efficient and have lower performance
- NAS presented as just another file system, while cloud storage is API based, with programs using the APIs to provide access
 - Examples include Dropbox, Amazon S3, Microsoft OneDrive, Apple iCloud
 - Use APIs because of latency and failure scenarios (NAS protocols wouldn't work well)





Storage Array

- Can just attach disks, or arrays of disks
- Avoids the NAS drawback that the storage I/O operations consume network bandwidth, thereby increasing the latency of network communication.
- 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





RAID Structure

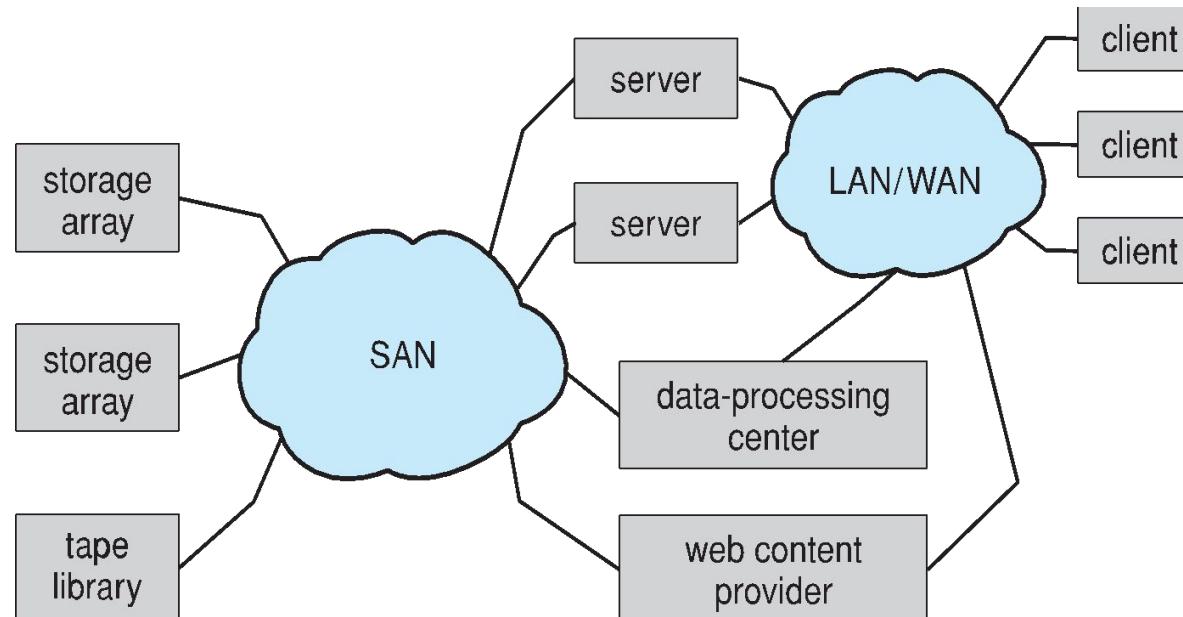
- RAID – redundant array of inexpensive disks
 - multiple disk drives provides reliability via redundancy
- Increases the mean time to failure
- Mean time to repair – exposure time when another failure could cause data loss
- Mean time to data loss based on above factors
- If mirrored disks fail independently, consider disk with 1300,000 mean time to failure and 10 hour mean time to repair
 - Mean time to data loss is $100,000^2 / (2 * 10) = 500 * 10^6$ 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

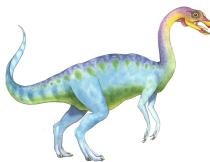




Storage Area Network

- A storage-area network (SAN) is a private network (using storage protocols rather than networking protocols) connecting servers and storage units,
- Common in large storage environments
- Multiple hosts attached to multiple storage arrays – flexible – for example, if a host is running low on disk space, the SAN can be configured to allocate more storage to that host.





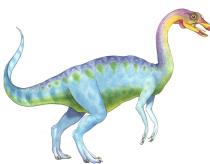
Storage Area Network (Cont.)

- SAN is one or more storage arrays
 - Connected to one or more Fibre Channel switches or **InfiniBand (IB)** network
- Hosts also attach to the 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
- SANs make it possible for clusters of servers to share the same storage
- Why have separate storage networks and communications networks?
 - Consider iSCSI, FCOE



A Storage Array





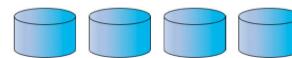
RAID (Cont.)

- 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 within a storage array can still fail if the array fails, so automatic **replication** of the 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

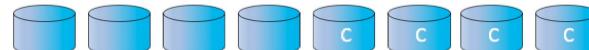




RAID Levels



(a) RAID 0: non-redundant striping.



(b) RAID 1: mirrored disks.



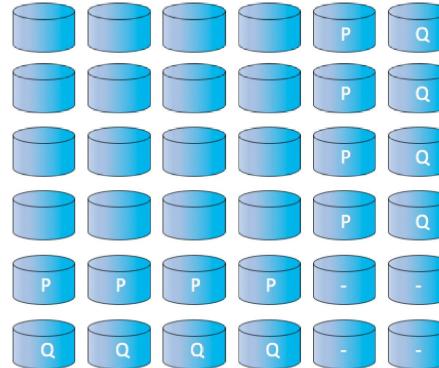
(c) RAID 4: block-interleaved parity.



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



(e) RAID 6: P + Q redundancy.



(f) Multidimensional RAID 6.





Hard Disk Drives

- Platters range from .85" to 14" (historically)
 - Commonly 3.5", 2.5", and 1.8"
- Range from 30GB to 3TB per drive
- Performance
 - Transfer Rate – theoretical – 6 Gb/sec
 - Effective Transfer Rate – real – 1Gb/sec
 - Seek time from 3ms to 12ms – 9ms common for desktop drives
 - Average seek time measured or calculated based on 1/3 of tracks
 - Latency based on spindle speed
$$4 \cdot 1 / (\text{RPM} / 60) = 60 / \text{RPM}$$
 - Average latency = $\frac{1}{2}$ latency





Hard Disk Performance

- **Access Latency = Average access time** = average seek time + average latency
 - For fastest disk $3\text{ms} + 2\text{ms} = 5\text{ms}$
 - For slow disk $9\text{ms} + 5.56\text{ms} = 14.56\text{ms}$
- **Average I/O time** = average access time + (amount to transfer / transfer rate) + controller overhead
- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead =
 - $5\text{ms} + 4.17\text{ms} + 0.1\text{ms} + \text{transfer time} =$
 - Transfer time = $4\text{KB} / 1\text{Gb/s} * 8\text{Gb / GB} * 1\text{GB} / 1024^2\text{KB} = 32 / (1024^2) = 0.031 \text{ ms}$
 - Average I/O time for 4KB block = $9.27\text{ms} + .031\text{ms} = 9.301\text{ms}$





The First Commercial Disk Drive



1956

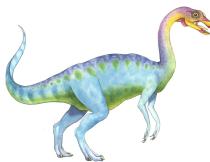
IBM RAMDAC computer
included the IBM Model
350 disk storage system

5M (7 bit) characters

50 x 24" platters

Access time = < 1 second





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
 - Can only be erased a limited number of times before worn out – $\sim 100,000$
 - Life span measured in **drive writes per day (DWPD)**
 - 4 A 1TB NAND drive with rating of 5DWPD is expected to have 5TB per day written within warranty period without failing





Volatile Memory

- DRAM frequently used as mass-storage device
 - Not technically secondary storage because volatile, but can have file systems, be used like very fast secondary storage
- RAM drives (with many names, including RAM disks) present as raw block devices, commonly file system formatted
- Computers have buffering, caching via RAM, so why RAM drives?
 - Caches / buffers allocated / managed by programmer, operating system, hardware
 - RAM drives under user control
 - Found in all major operating systems
 - 4 Linux /dev/ram, macOS diskutil to create them, Linux /tmp of file system type tmpfs
- Used as high speed temporary storage
 - Programs could share bulk data, quickly, by reading/writing to RAM drive





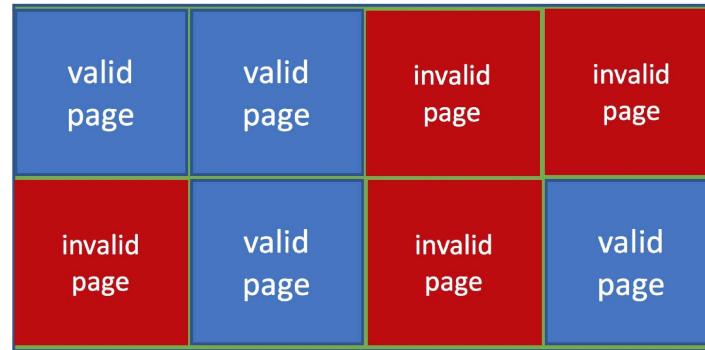
!!





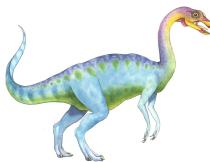
NAND Flash Controller Algorithms

- 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
- Allocates **overprovisioning** to provide working space for GC
- Each cell has lifespan, so **wear leveling** needed to write equally to all cells



NAND block with valid and invalid pages





Error Detection and Correction

- Fundamental aspect of many parts of computing (memory, networking, storage)
- **Error detection** determines if there a problem has occurred (for example a bit flipping)
 - If detected, can halt the operation
 - Detection frequently done via parity bit
- Parity one form of **checksum** – uses modular arithmetic to compute, store, compare values of fixed-length words
 - Another error-detection method common in networking is **cyclic redundancy check (CRC)** which uses hash function to detect multiple-bit errors
- **Error-correction code (ECC)** not only detects, but can correct some errors
 - Soft errors correctable, hard errors detected but not corrected





Storage Device Management

- **Low-level formatting**, or **physical formatting** — Dividing a disk into sectors that the 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, the operating system still needs to record its own data structures on the disk
 - **Partition** the 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**
 - 4 Disk I/O done in blocks
 - 4 File I/O done in clusters

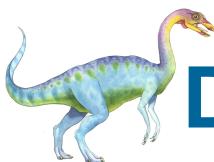




Storage Device Management (cont.)

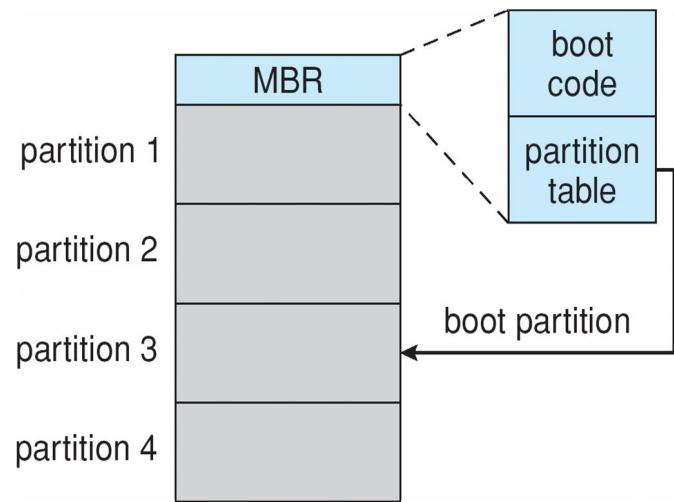
- Root partition contains the OS, other partitions can hold other Oses, other file systems, or be raw
 - Mounted at boot time
 - Other partitions can mount automatically or manually
- At mount time, file system consistency checked
 - Is all metadata correct?
 - 4 If not, fix it, try again
 - 4 If yes, 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





Device Storage 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
 - The bootstrap is stored in ROM, firmware
 - Bootstrap loader program stored in boot blocks of boot partition
- Methods such as sector sparing used to handle bad blocks



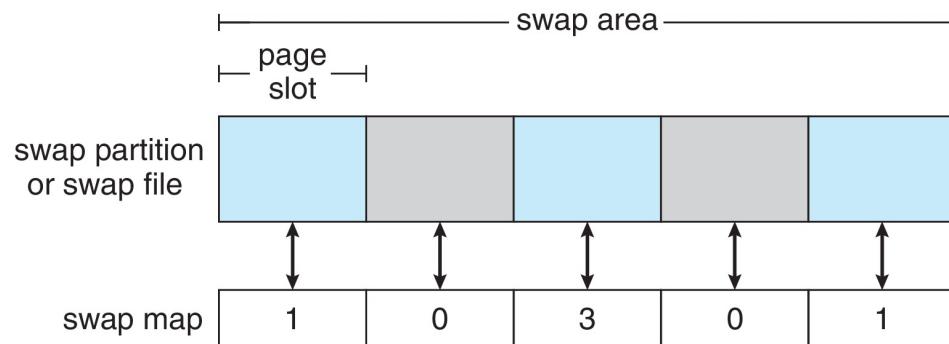
Booting from secondary storage in Windows

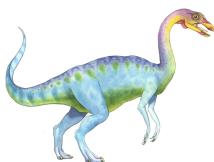




Swap-Space Management

- 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
 - Usually 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:

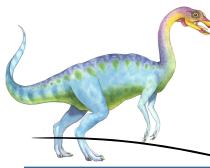




Storage Attachment

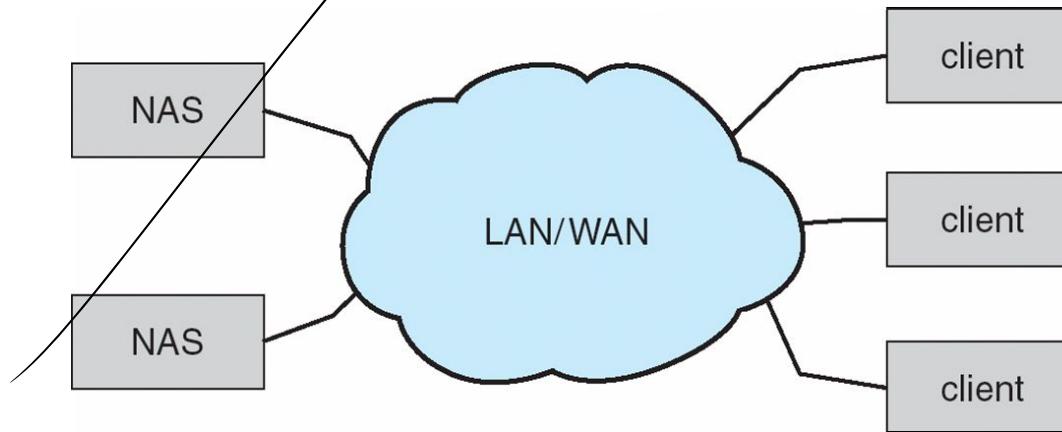
- Computers access storage in three ways
 - host-attached
 - network-attached
 - cloud
- Host attached access through local I/O ports, using one of several technologies
 - To attach many devices, use storage busses such as USB, firewire, thunderbolt
 - This architecture supports a maximum of two drives per I/O bus. A newer, similar protocol that has simplified cabling is SATA
 - High-end systems use **fibre channel (FC)**
 - 4 High-speed serial architecture using fibre or copper cables
 - 4 Because of the large address space and the switched nature of the communication, multiple hosts and storage devices can connect to the FC fabric

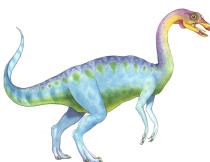




Network-Attached Storage

- Network-attached storage (**NAS**) is storage made available over a network rather than over a local connection (such as a bus)
 - Remotely attaching to file systems
- NFS (linux) and CIFS (Windows) 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 the SCSI protocol
 - hosts can treat their storage as if it were directly attached, even if the storage is distant from the host





Cloud Storage

- NAS provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.
 - It tends to be less efficient and have lower performance
- NAS presented as just another file system, while cloud storage is API based, with programs using the APIs to provide access
 - Examples include Dropbox, Amazon S3, Microsoft OneDrive, Apple iCloud
 - Use APIs because of latency and failure scenarios (NAS protocols wouldn't work well)





Storage Array

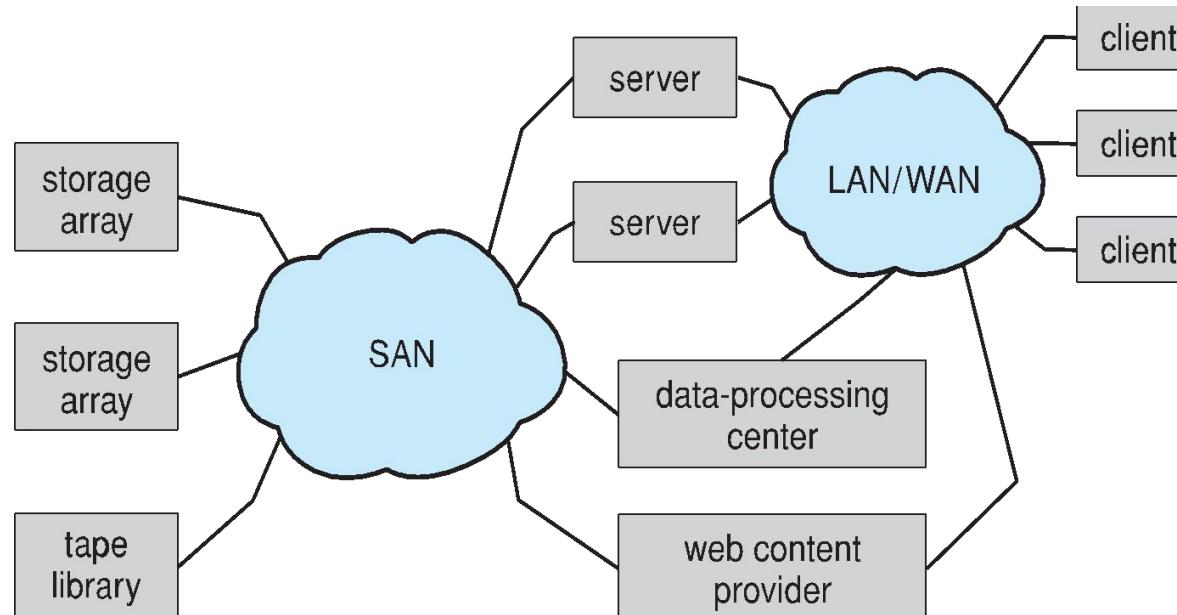
- Can just attach disks, or arrays of disks
- Avoids the NAS drawback of using network bandwidth
- Storage Array has controller(s), provides features to attached host(s)
 - Ports to connect hosts to array
 - Memory, controlling software (sometimes NVRAM, etc)
 - A few to thousands of disks
 - RAID, hot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





Storage Area Network

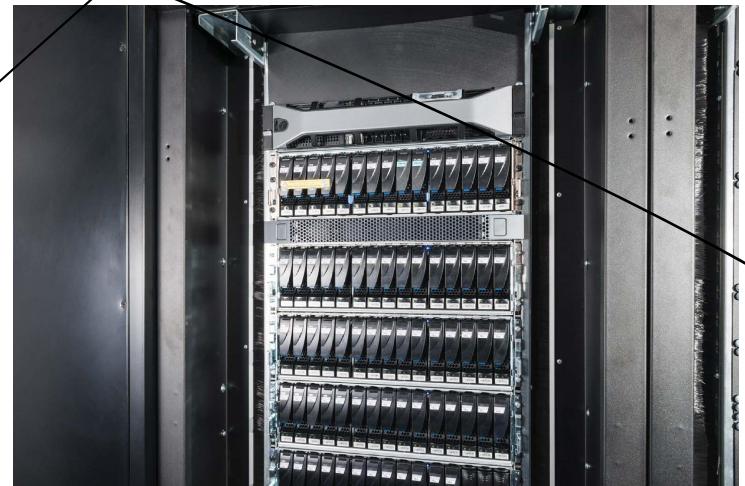
- Common in large storage environments
- Multiple hosts attached to multiple storage arrays – flexible





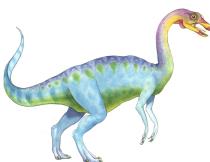
Storage Area Network (Cont.)

- SAN is one or more storage arrays
 - Connected to one or more Fibre Channel switches or **InfiniBand (IB)** network
- Hosts also attach to the 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
- Why have separate storage networks and communications networks?
 - Consider iSCSI, FCOE



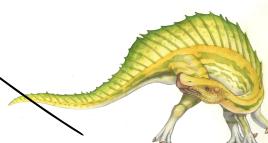
A Storage Array





RAID Structure

- 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 1300,000 mean time to failure and 10 hour mean time to repair
 - Mean time to data loss is $100,000^2 / (2 * 10) = 500 * 10^6$ 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





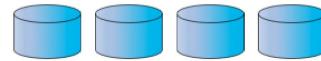
RAID (Cont.)

- 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 within a storage array can still fail if the array fails, so automatic **replication** of the 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

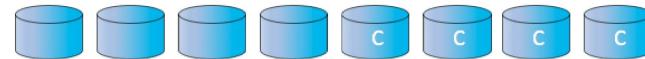




RAID Levels



(a) RAID 0: non-redundant striping.



(b) RAID 1: mirrored disks.



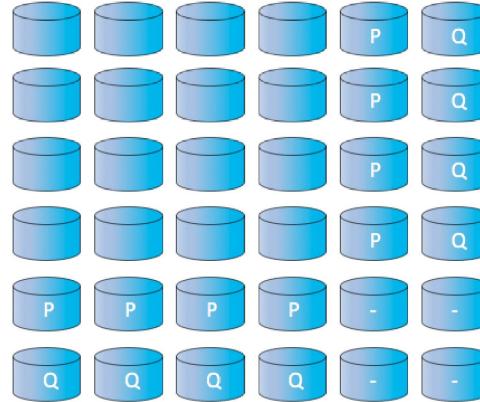
(c) RAID 4: block-interleaved parity.



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



(e) RAID 6: P + Q redundancy.

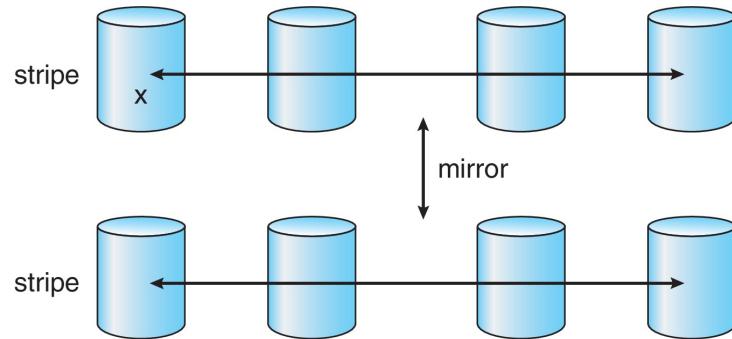


(f) Multidimensional RAID 6.

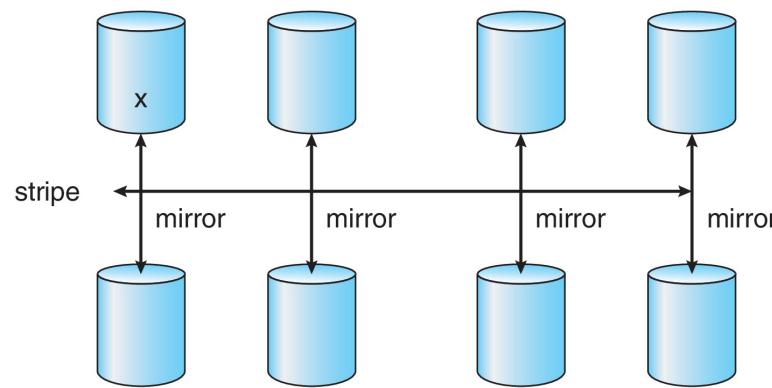




RAID (0 + 1) and (1 + 0)



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



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





Other Features

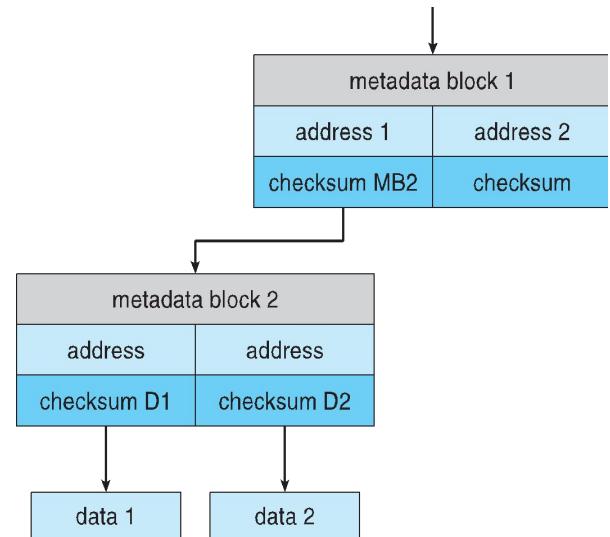
- Regardless of where RAID implemented, other useful features can be added
- **Snapshot** is a view of file system before a set of changes take place (i.e. at a point in time)
 - More in Ch 12
- Replication is automatic duplication of writes between separate sites
 - For redundancy and disaster recovery
 - 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





Extensions

- RAID alone does not prevent or detect data corruption or other errors, just disk failures
- Solaris ZFS adds **checksums** of all data and metadata
- Checksums kept with pointer to object, to detect if object is the right one and whether it changed
- Can detect and correct data and metadata corruption
- ZFS also removes volumes, partitions
 - Disks allocated in **pools**
 - Filesystems with a pool share that pool, use and release space like **malloc()** and **free()** memory allocate / release calls

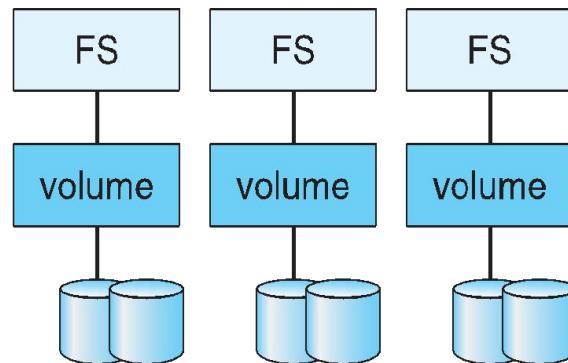


ZFS checksums all metadata and data

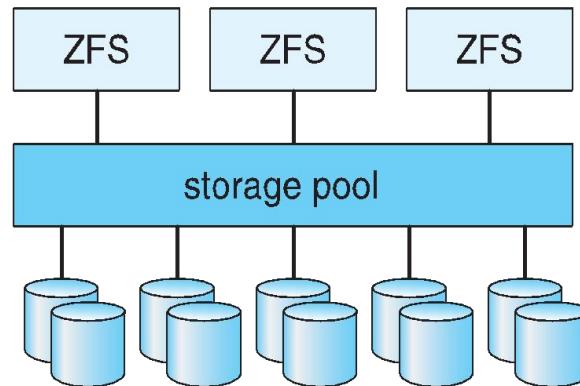




Traditional and Pooled Storage



(a) Traditional volumes and file systems



(b) ZFS and pooled storage.





Object Storage

- General-purpose computing, file systems not sufficient for very large scale
- Another approach – start with a storage pool and place objects in it
 - Object just a container of data
 - No way to navigate the pool to find objects (no directory structures, few services)
 - Computer-oriented, not user-oriented
- Typical sequence
 - Create an object within the pool, receive an object ID
 - Access object via that ID
 - Delete object via that ID
- Object storage management software like **Hadoop file system (HDFS)** and **Ceph** determine where to store objects, manages protection
 - Typically by storing N copies, across N systems, in the object storage cluster
 - **Horizontally scalable**
 - **Content addressable, unstructured**



End of Chapter 11

