Chapter 12: Mass-Storage Systems

Chapter 12: Mass-Storage Systems

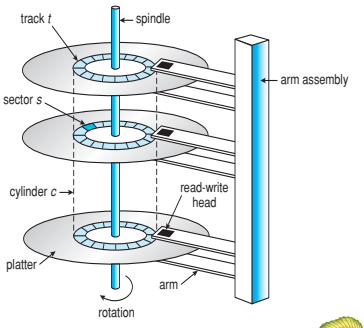
- Overview of Mass Storage Structure
- Disk Structure
- Disk Attachment
- Disk Scheduling
- Disk Management
- RAID Structure

Objectives

- To describe the physical structure of secondary storage devices and its effects on the uses of the devices
- To explain the performance characteristics of mass-storage devices
- To evaluate disk scheduling algorithms
- To discuss operating-system services provided for mass storage, including RAID

Moving-head Disk Mechanism

- Each disk **platter** has a flat circular shape with diameters 1.8 to 3.5 inches.
- Two surfaces of a platter covered with a magnetic materials for storing information
- A read-write head "flies" just above each surface of every platter
- The heads are attached to a disk arm that move all heads as a unit
- The surface of a platter is logically divided into circular **tracks**, which are subdivided into hundreds of **sectors**.
- The set of tracks that are at one arm position makes up a **cylinder**.
- There could be thousands of concentric cylinders in a disk drive



Overview of Mass Storage Structure

- **Magnetic disks** provide bulk of secondary storage for modern computers
  - Disk drives rotate at 60 to 250 times per second, or specified in rotations per minute (**RPM**)
  - **Positioning time** (**random-access time**) is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
  - **Transfer rate** is rate at which data flow between drive and computer
  - **Head crash** results from disk head making contact with the disk surface. This normally cannot be repaired; the entire disk must be replaced.
- A disk can be removable, allowing different disks to be mounted as needed.
  - Removable magnetic disks generally consist of one platter.
  - Other forms of removable disks include CDs, DVDs, Blue-ray discs as well as **flash drives**
- Drive attached to computer by a set of wires called an **I/O bus**
  - Busses vary, including **EIDE**, **ATA**, **SATA**, **USB**, **Fibre Channel**, **SCSI**, **SAS**, **Firewire**
  - The computer place commands into a **host controller** (controller at the computer end of the buses), typically using memory-mapped I/O ports, which then sends commands via messages to **disk controller** (built into disk drive). Disk controller operates the disk-drive hardware to carry out the commands
  - 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 occurs between the cache and the host controller

Magnetic Disks

- 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
  - RPM typically, 5,400, 7,200, 10,000 and 15,000
  - Latency based on spindle speed
    - 60/(RPM
  - Average latency = ½ latency

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

(From Wikipedia)

Magnetic Disk Performance

- **Access Latency** = **Average access time** = average seek time + average latency
  - For fastest disk 3ms + 2ms = 5ms
  - For slow disk 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (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 =
  - 5ms + 4.17ms + 4KB / 1Gb/sec + 0.1ms =
  - 9.27ms + 4 / 131072 sec =
  - 9.27ms + .12ms = 9.39ms
  - This implies it takes an average 9.39ms to transfer 4KB, thus effective bandwidth is 4KB/9.39 = 13.63 Mb/sec only (with a transfer rate at 1 Gb/sec given the overhead).

Solid-State Disks (SSDs)

- An SSD is nonvolatile memory used like a hard drive
  - There are many variations of this technology, from DRAM with a battery to maintain its state in a power failure, through flash-memory technologies like single-level cell (SLC) and multilevel cell (MLC) chips
- They can be more reliable than HDDs because they have no moving parts
- They are much faster because they have no seek time or rotation latency.
- They consumes less power
- But they are more expensive per MB, have less capacity, and may have shorter life span
- Their uses are somewhat limited
  - One use is in storage-arrays, where they hold file-system metadata that require high performance
  - SSDs also used in laptop to make them smaller, faster and more energy-efficient
- Because they are much faster than magnetic disk drives, standard bus interface can be too slow, causing a major limit on throughout
  - Some connect directly to system bus (PCI, for example)
  - Some use them as a new cache tier, moving data between magnetic disk, SSDs, and memory to optimize performance

Magnetic Tape

- Magnetic tape was an early secondary-storage medium
- It is relatively permanent and can hold large quantities of data
- Its access time is slow
- Random access ~1000 times slower than magnetic disk, so not very useful for secondary storage
- Mainly used for backup, storage of infrequently-used data, or as a medium of transferring information from one system to another
- Tape capacities vary greatly, depending on the particular kind of tape drive, with current capacities exceeding several terabytes, and typically between 200GB and 1.5TB

Disk Structure

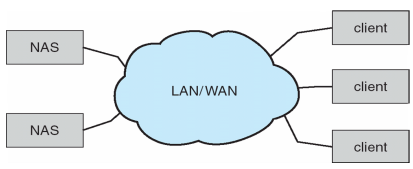
- Disk drives are addressed as large one-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer
  - The size of a logical block is usually 215 bytes
  - Low-level formatting creates **logical blocks** on physical media
- The one-dimensional array of logical 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
  - Logical to physical address should be easy in theory, except
    - Defective sectors
    - Non-constant # of sectors per track via constant angular velocity

Disk Attachment

- Host-attached storage accessed through I/O ports. These ports use several technologies
- The typical desktop PC uses an I/O bus architecture called IDE or ATA, or a new version SATA
- **Fibre channel** (**FC**) is a high-speed serial architecture. It has two variants:
  - A large switching fabric with 24-bit address space – the basis of **storage area networks** (**SANs**) in which many hosts and many storage devices attached to the fabric, allowing great flexibility in I/O communications
  - An **arbitrated loop** (**FC-AL**) with address 126 devices (drives and controllers)
- A wide variety of storage devices are suitable for use as host-attached storage, hard disk drives, RAID arrays, CD, DVD, and tape drives.

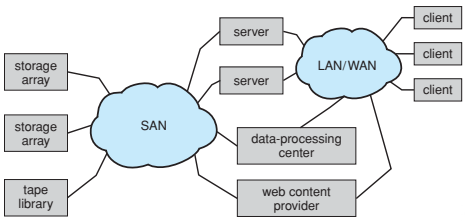
Network-Attached Storage

- A network-attached storage (**NAS**) device is a special-purpose storage system that is accessed remotely over a network
- Implemented via remote procedure calls (RPCs) between host and storage over typically TCP or UDP on IP network, such as NFS for UNIX systems and CIFS for Window machines
- **iSCSI** is the latest NAS protocol, which essentially uses IP network to carry the SCSI protocol
- This provides a convenient way for users on a LAN to share a pool of storage, but tends to be less efficient and have lower performance that some direct-attached storage options



Storage Area Network

- A storage-are network (**SAN**) is a private network (using storage protocols rather networking protocols like in NAS) connecting servers and storage units.
- SAN is one or more storage arrays, connected to one or more Fibre Channel (FC) switches
- The power of SAN is its **flexibility** - multiple hosts and multiple storage arrays can attach to the same SAN, an storage can be dynamically allocated to hosts.



Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having fast access time and large disk bandwidth
- The **seek time** is the time for the disk head arm to move the heads to the cylinder containing the desired sector, which can be measured by the seek distance in term of number cylinders/tracks.
- The **rotational latency** is the additional time for the disk to rotate the desired sector to the disk head.
- 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.
- We can improve both access time and the bandwidth by managing the order in which disk I/O requests are serviced.

Disk Scheduling (Cont.)

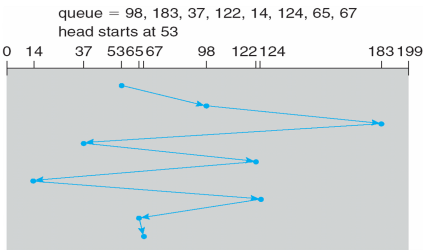
- There are many sources of disk I/O requests, from OS, system and user processes
- I/O request includes input/output mode, disk address, memory address, number of sectors to transfer
- The operating system maintains a queue of requests per disk or device. For a multiprogramming system with many processes, the disk queue may often have several pending requests.
- When one request is completed, which pending request to select to service next – **disk scheduling**
- We illustrate scheduling algorithms with a request queue (0-199), 0-199 are cylinder numbers.

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

  Head pointer 53

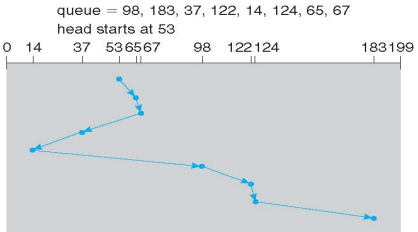
FCFS first-come-first-serve

- FCFS is intrinsically fair, but it generally does not provide fast service
- Illustration shows total head movement of 640 cylinders



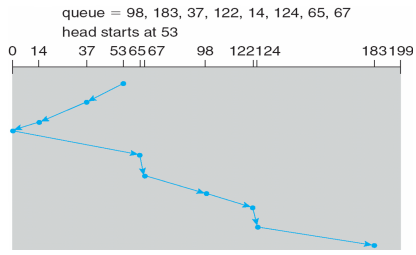
SSTF shortest-seek-time-first

- The **Shortest Seek Time First** (**SSTF**) selects the request with the least seek time from the current head position, i.e., choose the pending request closest to the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests, as requests may arrive at any time dynamically
- Illustration shows total head movement of 236 cylinders



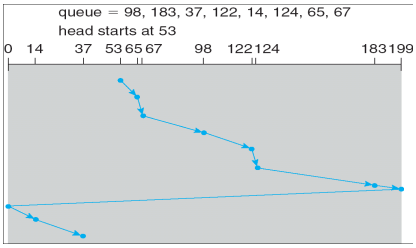
SCAN Scheduling

- 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. This sometimes called the elevator algorithm
- Note if requests are uniformly distributed across cylinders, the heaviest density of requests are at other end of disk and those wait the longest
- Illustration shows total head movement of 208 cylinders



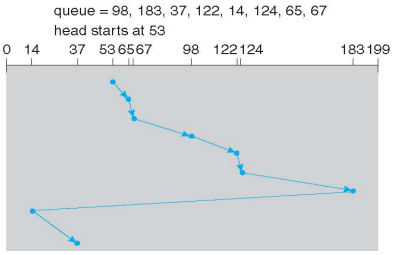
C-SCAN

- **C-SCAN**, Circular-SCAN, a variant of SCAN, provides a more uniform waiting 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? - 382



C-LOOK

- LOOK a version of SCAN, C-LOOK a version of C-SCAN
- Disk 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
- Total number of cylinders? – 322



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal as it increases performance over FCFS. SCAN and C -SCAN perform better for systems that place a heavy load on the disk, because they are less likely to cause starvation problem.
- Performance depends on the number and types of requests. If only one request, all scheduling behave the same
- Requests for disk service can be influenced by the file-allocation method
  - Contiguous allocated file will generate several requests close together on the disk, resulting in limited head movement, while a linked or indexed file may include blocks widely scattered on the disk, resulting in greater head movement.
- The location of directories and index blocks are also important, which are accessed frequently.
  - For instance
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary. Either SSTF or LOOK is a reasonable choice for the default algorithm.

Disk Management

- **Low-level formatting**, or **physical formatting** (done by disk manufacturer) — dividing a disk into sectors that the disk controller can read and write
  - Each sector holds data with a header and trailer containing information such as a sector number and an error- correction code (**ECC**)
- 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 separate disk
  - **Logical formatting** or creation of a file system – stores initial file-system data structures on the disk
- To increase efficiency most file systems group blocks into **clusters**
  - Disk I/O done in blocks, and file I/O done in clusters
- Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example) – raw I/O bypasses all file-system services such as space allocation, file names and directories.
- A **boostrap** program initializes all aspects of the system, from CPU registers to device controllers
  - The bootstrap is stored in ROM
- Methods such as **sector sparing** (low-level formatting sets aside spare sectors invisible to the operating system) used to handle **bad blocks**

RAID - Improving Reliability via Redundancy

- **RAID – redundant arrays of independent disks**
  - In the past, RAID composed of small, cheap disks were viewed as a cost-effective alternative to large, expensive disks (once called redundant arrays of inexpensive disks)
  - Now RAIDs are used for higher reliability via redundancy and higher data-transfer rate (access in parallel)
- The chance that a disk out of N disks fails is much higher than the chance that a specific single disk fails. Suppose that 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.
- The data loss rate is unacceptable if we store only one copy of the data.
- The solution to the problem of reliability is to introduce redundancy; the simplest (but most expensive) approach is to duplicate every disk, called **mirroring**. Every write is carried out on two physical disks. Data will be lost only if the second disk fails before the first failed disk is replaced.
- The **mean time to repair** is the time it takes (on average) to replace a failed disk and to restore data on it – exposure time when another failure could cause data loss
- Suppose the mean time to failure of a single disk is100,000 hours and the mean time to repair is 10 hours. The **mean time to data loss** is 100, 000^2 / (2 ∗ 10) = 500 ∗ 10^6 hours, or 57,000 years!

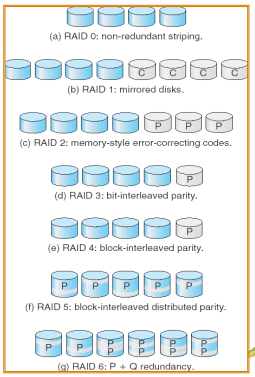
RAID – Improving Performance via Parallelism

- Parallelism in a disk system, via **data stripping**, has two main goals:
  - Increase the throughout of multiple small access by load balancing
  - Reduce the response time of large access
- **Bit-level stripping**
  - For example, if we have an array of 8 disks, we can write bit i of each byte to disk i. The array of 8 disks can be treated as a single disk with sectors that are 8 times the normal sector size. The access rate can be improved by 8 times!
- Bit-level stripping can be generalized to include a number of disks that either is a multiple of 8 or a divides 8.
  - For example, with an array of 4 disks, bit i and 4+i of each byte cane be stored in disk i
- The **block-level stripping**, blocks of a file are stripped across multiple disks
  - With n disks, block i of a file goes to disk (i mod n)+i

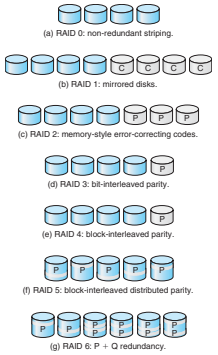
RAID Structure

- Mirroring provides high reliability, but expensive. Stripping provides high data-transfer rates, but does not provide reliability
- Numerous schemes to provide redundancy at lower cost by using stripping combined with "parity" bits.
- These schemes have different cost-performance trade-offs and classified into RIAD levels.
- In the RIAD levels, four disks' worth of data are stored. P indicates error-correcting bits and C indicates a second copy of the data

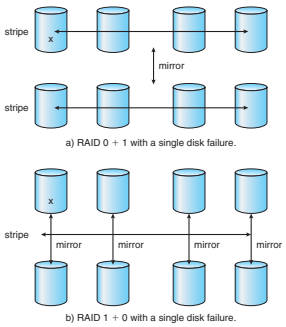
- Mirroring or shadowing (RAID 1) keeps duplicate of each disk
- Striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides high performance and high reliability
- Block interleaved parity (RAID 4, 5, 6) uses much less redundancy



RAID Levels



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



Other Features

- Regardless of where RAID implemented, other useful features can be added
- Snapshot is a view of file system before the last update took place (for recovery)
- Replication is automatic duplication of writes between separate sites for redundancy and disaster recovery. This can be synchronous or asynchronous
- A hot spare disk is not used for data but is configured to be used as a replacement in case of disk failure
  - For instance, a hot spare can be used to rebuild a mirrored pair should one of the disks in the pair fails. In this way, RAID level can be reestablished automatically, without waiting for the failed disk to be replaced/repaired.