Chapter 12: Mass-Storage Systems

Zhi Wang
Florida State University
Content

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

• Describe the **physical structure** of secondary and tertiary storage devices and the effects on the uses of the devices

• Explain the **performance characteristics** of mass-storage devices

• Discuss **operating-system services** provided for mass storage
Overview

- Magnetic disks provide bulk of secondary storage of computer system
  - hard disk is most popular; some magnetic disks could be removable
  - driver attached to computer via I/O buses (e.g., USB, SCSI, EIDE, SATA…)
  - drives rotate at 60 to 250 times per second (7000rpm = 117rps)
- Transfer rate is rate at which data flow between drive and computer
- **Positioning time** is time to move disk arm to desired sector
  - positioning time includes **seek time** and **rotational latency**
    - seek time: move disk to the target cylinder
    - rotational latency: for the target sector to rotate under the disk head
  - positioning time is also called random-access time
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
Magnetic Disks

- Magnetic disks have platters, range from .85” to 14” (historically)
  - 3.5”, 2.5”, and 1.8” are common nowadays
- Capacity ranges from 30GB to 3TB per drive
- Performance
  - **transfer rate**: theoretical 6 Gb/sec; effective (real) about 1Gb/sec
  - **seek time** from 3ms to 12ms (9ms common for desktop drives)
  - latency based on spindle speed: \( \frac{1}{\text{rpm}} \times 60 \)
    - average latency = \( \frac{1}{2} \) latency

<table>
<thead>
<tr>
<th>Spindle [rpm]</th>
<th>Average latency [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4200</td>
<td>7.14</td>
</tr>
<tr>
<td>5400</td>
<td>5.56</td>
</tr>
<tr>
<td>7200</td>
<td>4.17</td>
</tr>
<tr>
<td>10000</td>
<td>3</td>
</tr>
<tr>
<td>15000</td>
<td>2</td>
</tr>
</tbody>
</table>
Moving-head Magnetic Disk
Magnetic Disk

- **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 + (data to transfer / transfer rate) + controller overhead
  - e.g., to transfer a 4KB block on a 7200 RPM disk; 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead:
    
    $5 \text{ms} + 4.17 \text{ms} + 4 \text{KB} / 1 \text{Gb/sec} + 0.1 \text{ms} = 9.39 \text{ms}$ (4.17 is average latency)
Magnetic Tape

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

- Disk drives are addressed as a 1-dimensional arrays of logical blocks,
  - logical block is the smallest unit of transfer

- Logical blocks are mapped into **sectors** of the disk sequentially
  - sector 0 is the first sector of the first track on the outermost cylinder
  - mapping proceeds in order
    - first through that **track**
    - then the rest of the tracks in that **cylinder**
    - then through the rest of the cylinders from outermost to innermost
  - logical to physical address should be easy
    - except for bad sectors
Disk Attachment

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

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

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

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

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

• Disk scheduling chooses which pending disk request to service next
  • concurrent sources of disk I/O requests include OS, system/user processes
  • idle disk can immediately work on a request, otherwise OS queues requests
    • each request provide I/O mode, disk & memory address, and # of sectors
  • OS maintains a queue of requests, per disk or device
  • optimization algorithms only make sense when a queue exists
Disk Scheduling

• Disk scheduling usually tries to minimize seek time
  • rotational latency is difficult for OS to calculate

• There are many disk scheduling algorithms
  • FCFS
  • SSTF
  • SCAN
  • C-SCAN
  • C-LOOK

• We use a request queue of “98, 183, 37, 122, 14, 124, 65, 67” ([0, 199]), and initial head position 53 as the example
FCFS

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

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
**SSTF**

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

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
SCAN

- SCAN algorithm sometimes is called the *elevator* algorithm
  - disk arm starts at one **end** of the disk, and moves toward the **other end**
  - service requests during the movement until it gets to the other end
  - then, the head movement is reversed and servicing continues.
- Total head movement of 236 cylinders
C-SCAN

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

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

![Diagram showing head movements and queue numbers]
Selecting Disk-Scheduling Algorithm

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

- **Physical formatting**: divide disk into sectors for controller to read/write
  - each sector is usually 512 bytes of data but can be selectable
- OS records its own data structures on the disk
  - partition disk into groups of cylinders, each treated as a logical disk
  - **logical formatting** partitions to making a file system on it
    - some FS has spare sectors reserved to handle bad blocks
    - FS can further group blocks into clusters to improve performance
    - initialize the boot sector if the partition contains OS image
Windows 2000 Disk Layout
Swap Space Management

- **Swap space**: disk space used by virtual memory as an extension of the main memory
  - swap space can be carved out of normal FS, or a separate partition (raw)
  - less common now due to increased memory capacity
- Swap space management varies among OS
  - usually, kernel uses swap maps to track swap-space use
  - 4.3BSD allocates swap space when process starts
    - to hold text segment (the program) and data segment
  - Solaris 2 allocates it only when a dirty page is to be paged out
    - file data written to swap space until write to file system requested
    - other dirty pages go to swap space due to no other home
    - text segment pages thrown out and reread from the file system as needed
Linux Swap Space Management

![Diagram of Linux Swap Space Management]

- **Swap Area**
- **Swap Partition or Swap File**
- **Swap Map**
  - 1
  - 0
  - 3
  - 0
  - 1
RAID

- RAID: use multiple disk drives to provide performance/reliability
  - reliability via mirroring or error correction code
  - performance via **disk striping**
    - segmenting logically sequential data, such as a file, and
    - store consecutive segments on different physical storage devices
- RAID is arranged into **six** different levels
  - **RAID 0**: splits data evenly across two or more disks without parity bits
    - aka. striped volume, it improves performance, but decrease MTTF
  - **RAID 1**: an exact copy (or mirror) of a set of data on two disks
  - **RAID 2**: stripes data at the bit-level; uses Hamming code for error correction
    - hamming code (4bit data+3bit parity) allows 7 disks to be used
RAID

- **RAID 3**: byte-level striping with a dedicated parity disk
  - require synchronized disk spinning (RAID 3 is usually not used)
- **RAID 4**: block-level striping with a dedicated parity disk
  - a single block request can be fulfilled by one disk
  - different disk can fulfill different block requests
- **RAID 5**: block-level striping with parity data distributed across all disks
- **RAID 6**: extends RAID 5 by adding an additional parity block
  - RAID 6 has block-level striping with 2 parity blocks
RAID Levels

(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

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

(g) RAID 6: P + Q redundancy.
RAID (0 + 1) and (1 + 0)

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

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

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

(a) Traditional volumes and file systems.

(b) ZFS and pooled storage.
Tertiary Storage Devices

- Tertiary storage: being third-level after memory and magnetic disks
  - e.g., floppy disks, CD-ROMs, DVDs...
  - usually removable
  - low cost
Magneto-optic Disks

- MO disk records data on a rigid platter coated with magnetic material
  - laser is used to record and read data
  - larger distance between head and disk surface (to shoot the laser)
  - optical disks don’t use magnetism; employs materials changeable by laser
- MO disk usually can be written many times
WORM Disks

- WORM disks can be written only once
  - WORM: write once, read many time (e.g., CD-ROM, DVD-ROM…)
  - usually a thin aluminum film sandwiched between two glass/plastic platters
  - to write a bit, drive uses laser to burn a small hole through the aluminum
    - information can be destroyed by not altered
  - relatively durable and reliable
Tapes

- Tape is less expensive and has higher capacity than disk
  - many cheap cartridges share a few expensive drives
    - e.g., dell PowerVault LTO-3: $2,056
- Tape is best for **sequential access**, random access is much slower
  - mostly used for backup or transfer of large volumes of data
  - large tape installation automates tape change and storage with robotic arms
OS-support for Tape

- Tapes are usually presented as a **raw** storage medium
  - normal disks can be accessed as either as raw media or with file systems
  - no file system on the tape, just array of blocks
  - tape drive is usually reserved for exclusive use of the application
  - the application decides how to use the array of blocks
    - other applications usually do not understand the format of it
- Tape drives are “**append-only**” devices
  - an EOT mark is placed after a block that is written
  - updating a block in the middle effectively erases everything beyond it
Speed of Tertiary Storage

- Two aspects of speed in tertiary storage are bandwidth and latency.

- **Bandwidth** is measured in bytes per second.
  - **sustained bandwidth**: average data rate during a large transfer
  - data rate when the data stream is actually flowing
  - **effective bandwidth**: average over the entire I/O time
    - including seek time or locate time, and cartridge switching
    - drive’s overall data rate

- **Access latency**: amount of time needed to locate data
  - access time for a disk: seek time + rotational latency; < 35 milliseconds
  - access time for tape: tens or hundreds seconds to wind the tape
    - thousands times slower than disk
Relative Reliability

- A fixed disk is likely to be more reliable than a removable disk or tape
  - Flash device is even more reliable as there are no moving parts
  - Though a head crash in a fixed hard disk generally destroys the data
  - Failure of tape drive or optical disk drive often leaves data undamaged
- An optical cartridge normally more reliable than magnetic disk or tape
Cost

- Main memory is much more expensive than disk storage
- Cost per megabyte of hard disk is comparable to tape if only one tape is used per drive
  - but tape drive is expensive and tape is cheap
  - cheap tape and hard disk have about same storage capacity
Cost Per Megabyte of DRAM

The graph illustrates the decreasing cost per megabyte of DRAM from 1982 to 2004. The cost has significantly declined over the years, with notable drops around 1988, 1992, and 1998, corresponding to increases in memory capacity. For example, the cost of 16 KB dropped from a high of $1280 to a low of $0.8 by 2004. Similar trends are observed for other memory sizes, such as 64 KB, 256 KB, 1 MB, and 32 MB, with gradual decreases in cost.
Cost Per Megabyte of Hard Disk

![Graph showing the decreasing cost per megabyte of hard disk from 1982 to 2004. The y-axis represents the cost per megabyte, ranging from 0.1 to 0.001. The x-axis represents the years from 1982 to 2004. Key points on the graph include 10 MB, 20 MB, 120 MB, 1.2 GB, 2 GB, 19 GB, 45 GB, and 80 GB. The trend line shows a significant decrease in cost as storage capacities increase.]
Cost Per Megabyte of Tape
End of Chapter 12