Chapter 14: Mass-Storage Systems

- Logical Disk Structure
- Disk Scheduling
- Disk Management
- RAID Structure

Logical Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer.
- In the simplest arrangement, 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.

Logical Disk Structure (Cont.)

- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue to access logical blocks 0-199.
- Head pointer currently at block 53

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and high disk bandwidth.
- Access time has two major components
  - Seek time is the time for the disk arm to move the heads to the cylinder containing the desired sector.
  - Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time = seek distance
- Disk bandwidth is the total number of bytes transferred divided by time to transfer the data

FCFS

Illustration shows total head movement of 640 cylinders.

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.
- Illustration shows total head movement of 236 cylinders.
SSTF (Cont.)

- The head starts at 53, and moves through the disk.
- Head movement is shown for various requests.

SCAN

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- Sometimes called the elevator algorithm.
- Illustration shows total head movement of 208 cylinders.

SCAN (Cont.)

- 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.

C-SCAN

- 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.

C-LOOK

- 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

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk.
- Either SSTF or LOOK is a reasonable choice for the default algorithm.
- Performance depends on the number and types of requests.
- Requests for disk service can be influenced by the file-allocation method.
- The disk-scheduling algorithm is written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary.

Physical Disk Management

- Low-level formatting, or physical formatting — Dividing a disk into sectors that the disk controller can read and write.
- 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.
  - Logical formatting or “making a file system”.
- Boot block initializes system.
  - The bootstrap is stored in ROM.
  - Bootstrap loader program.

MS-DOS Disk Layout

- Sector 0: Boot block
- Sector 1: FAT
  - Root directory
  - Data blocks (subdirectories)

RAID Structure

- RAID — multiple disk drives provides reliability via redundancy.
- RAID is arranged into six different levels.

RAID (cont)

- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively.
- Disk striping spreads the blocks in a file across multiple disks in certain patterns.
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data.
  - Mirroring or shadowing keeps duplicate of each disk.
  - Block interleaved parity uses much less redundancy.
RAID Levels

- **RAID 0 (Non-redundant)**
  - Stripes Data, but does not employ redundancy
  - Lowest cost of any RAID
  - Best Write performance - no redundant information
  - Any single disk failure is catastrophic
  - Used in environments where performance is more important than reliability.

- **RAID 1 (Mirrored)**
  - Uses twice as many disks as non-redundant arrays - 100% Capacity Overhead - Two copies of data are maintained
  - Data is simultaneously written to both arrays
  - Data is read from the array with shorter queuing, seek and rotation delays - Best Read Performance.
  - If a disk fails, mirrored copy is still available
  - Used in environments where availability and performance (IO rate) are more important than storage efficiency.

- **RAID 2 (Memory Style ECC)**
  - Uses Hamming code - parity for distinct overlapping subsets of data
  - # of redundant disks is proportional to log of total # of disks - better for large # of disks - e.g., 4 data disks require 3 redundant disks
  - If a disk fails, data in subset is used to regenerate lost data
  - Multiple redundant disks are needed to identify faulty disk
RAID 3 (Bit Interleaved Parity)
- Data is bit-wise over the data disks.
- Uses single parity disk to tolerate data failures. Overhead in I/O.
- Logically a single high capacity, high transfer rate disk.
- Reads source data disks only. Writes source both data and parity disks.
- Used in environments that require high BW (Scientific, Image Processing, etc.), and not high I/O rates.

RAID 4 (Block Interleaved Parity)
- Similar to bit-interleaved parity disk array; except data is block-interleaved (Striping Units).
- Read requests smaller than one striping unit, access one striping unit.
- Write requests update the data block, and the parity block.
- Generating parity requires 4 I/O accesses (RMW).
- Parity disk gets updates on all writes - a bottleneck.

RAID 5 (Block-Interleaved Distributed Parity)
- Eliminates the parity disk bottleneck in RAID 4 - Distributes parity among all the disks.
- Data is distributed among all disks.
- All disks participate in read requests - Better performance than RAID 4.
- Write requests update the data block, and the parity block.
- Generating parity requires 4 I/O accesses (RMW).
- Left symmetry v.s. Right Symmetry - Allows each disk to be traversed once before any disk twice.

RAID 6 (P + Q Redundancy)
- Uses Reed-Solomon codes to protect against up to 2 disk failures.
- Data is distributed among all disks.
- Two sets of parity P & Q.
- Write requests update the data block, and the parity blocks.
- Generating parity requires 6 I/O accesses (RMW) - update both P & Q.
- Used in environments that require stringent reliability requirements.
## Comparisons

- **RAID 0**: Provides the best write performance
- **RAID 1**: Provides the best read performance
- **RAID 1**: Most expensive - 100% capacity overhead - 2N disks
- **RAID 1**: Best read performance - 5 disks - no redundancy
- **RAID 2**: RAID 1 & RAID 5 needs N + ceil(log_2(N)) - 1 disks

## Preferred Environments

- **RAID 0**: Performance & capacity are more important than reliability
- **RAID 1**: High I/O rate, high Availability environments
- **RAID 2**: Large I/O data throughput
- **RAID 3**: High I/O Application (Scientific, Image Processing...)
- **RAID 5**: High I/O Application
- **RAID 5 & RAID 6**: Mixed Applications

## What RAID for which application

- **Fast Workstation**:
  - Caching is important to improve I/O rate
  - If large files are installed, then RAID 0 may be necessary
  - It is preferred to put the OS and swap files in separate drives from user drives to minimize movement between swap file area & user area.

- **Small Server**:
  - RAID 1 is preferred

- **Mid-Size Server**:
  - If more capacity is needed, then RAID 5 is recommended

- **Large Server**: e.g. Database Servers
  - RAID 5 is preferred
  - Separate different I/Os in mechanically independent arrays; place index & data files in databases in different arrays

## The table below, which shows Throughput per SS relative to RAID 0, assumes that G drives in an error correcting group

<table>
<thead>
<tr>
<th>RAID Level</th>
<th>Small Reads</th>
<th>Small Writes</th>
<th>Large Reads</th>
<th>Large Writes</th>
<th>Storage Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RAID 1</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>RAID 3</td>
<td>1/G</td>
<td>1/G</td>
<td>(G-1)/G</td>
<td>(G-1)/G</td>
<td>(G-1)/G</td>
</tr>
<tr>
<td>RAID 5</td>
<td>max(1/G, 1/4)</td>
<td>1</td>
<td>(G-1)/G</td>
<td>(G-1)/G</td>
<td>(G-1)/G</td>
</tr>
<tr>
<td>RAID 6</td>
<td>max(1/G, 1/6)</td>
<td>1</td>
<td>(G-2)/G</td>
<td>(G-2)/G</td>
<td>(G-2)/G</td>
</tr>
</tbody>
</table>

## Price per Megabyte of DRAM, From 1981 to 2000

[Graph showing price per megabyte of DRAM from 1981 to 2000]

## Price per Megabyte of Magnetic Hard Disk, From 1981 to 2000

[Graph showing price per megabyte of magnetic hard disk from 1981 to 2000]