• Magnetic Disk Characteristics
• I/O Connection Structure
• Types of Buses
• Cache & I/O
• I/O Performance Metrics
• I/O System Modeling Using Queuing Theory
• Designing an I/O System
• RAID (Redundant Array of Inexpensive Disks)

(Chapter 7.5)
RAID (Redundant Array of Inexpensive Disks)

- The term RAID was coined in a 1988 paper by Patterson, Gibson and Katz of the University of California at Berkeley.
- In that article, the authors proposed that large arrays of small, inexpensive disks --usually SCSI, IDE support just started-- could be used to replace the large, expensive disks used on mainframes and minicomputers.
- In such arrays files are "striped" and/or mirrored across multiple drives.
- Their analysis showed that the cost per megabyte could be substantially reduced, while both performance (throughput) and fault tolerance could be increased.

- The Catch: Array Reliability without any redundancy:
  
  \[
  \text{Reliability of } N \text{ disks} = \frac{\text{Reliability of 1 Disk}}{N} \\
  \text{50,000 Hours} \div 70 \text{ disks} = 700 \text{ hours}
  \]
  
  - Disk system MTTF: Drops from 6 years to 1 month!
  - Arrays (without redundancy) too unreliable to be useful!
Manufacturing Advantages of Disk Arrays

Conventional: 4 disk form factors
3.5” 5.25” 10” 14”

Low End → High End

Disk Array: 1 disk form factor
3.5”
RAID Subsystem Organization

- Host
- Host adapter
- Array controller
- Single board disk controller
- Single board disk controller
- Single board disk controller

- **Control, buffering, parity logic**
- **Physical device control**

**Striping software off-loaded from host to array controller**

- No application modifications
- No reduction of host performance

*often piggy-backed in small format devices*
Basic RAID Organizations

• Non-Redundant (RAID Level 0)
• Mirrored (RAID Level 1)
• Memory-Style ECC (RAID Level 2)
• Bit-Interleaved Parity (RAID Level 3)
• Block-Interleaved Parity (RAID Level 4)
• Block-Interleaved Distributed-Parity (RAID Level 5)
• P+Q Redundancy (RAID Level 6)
• Striped Mirrors (RAID Level 10)
Non-Redundant Striped (RAID Level 0)

- RAID 0 simply stripes data across all drives (minimum 2 drives) to increase data throughput but provides no fault protection.
  - Sequential blocks of data are written across multiple disks in stripes, as follows:

```
Disk 0
  Block 1
  Block 6
  Block 11
  Block 16
  Block 21

Disk 1
  Block 2
  Block 7
  Block 12
  Block 17
  Block 22

Disk 2
  Block 3
  Block 8
  Block 13
  Block 18
  Block 23

Disk 3
  Block 4
  Block 9
  Block 14
  Block 19
  Block 24

Disk 4
  Block 5
  Block 10
  Block 15
  Block 20
  Block 25
```

- The size of a data block, which is known as the "stripe width" or unit, varies with the implementation, but is always at least as large as a disk's sector size.

- This scheme offers the best write performance since it never needs to update redundant information.

- It does not have the best read performance.
  - Redundancy schemes that duplicate data, such as mirroring, can perform better on reads by selectively scheduling requests on the disk with the shortest expected seek and rotational delays.
Optimal Size of Data Striping Unit
(Appplies to RAID Levels 0, 5, 6, 10)

• Lee and Katz [1991] use an analytic model of non-redundant disk arrays to derive an equation for the optimal size of data striping unit.
• They show that the optimal size of data striping is equal to:

\[
\sqrt{\frac{P X (L - 1) Z}{N}}
\]

• Where:
  – P is the average disk positioning time,
  – X is the average disk transfer rate,
  – L is the concurrency, Z is the request size, and
  – N is the array size in disks.
• Their equation also predicts that the optimal size of data striping unit is dependent only the relative rates at which a disk positions and transfers data, PX, rather than P or X individually.
• Lee and Katz show that the optimal striping unit depends on request size; Chen and Patterson show that this dependency can be ignored without significantly affecting performance.
• Typical optimal values of striping unit = 16 - 128 KBytes
Mirrored (RAID Level 1)

- *Utilizes mirroring or shadowing* of data using twice as many disks as a non-redundant disk array.
- Whenever data is written to a disk the same data is also written to a redundant disk, so that there are always two copies of the information.
- When data is read, it can be retrieved from the disk with the shorter queuing, seek and rotational delays.
- If a disk fails, the other copy is used to service requests.
- Mirroring is frequently used in database applications where availability and transaction rate are more important than storage efficiency.
Memory-Style ECC (RAID Level 2)

- RAID 2 performs data striping with a block size of one bit or byte, so that all disks in the array must be read to perform any read operation.
- A RAID 2 system would normally have as many data disks as the word size of the computer, typically 32.
- In addition, RAID 2 requires the use of extra disks to store an error-correcting code for redundancy.
  - With 32 data disks, a RAID 2 system would require 7 additional disks for a Hamming-code ECC.
  - Such an array of 39 disks was the subject of a U.S. patent granted to Unisys Corporation in 1988, but no commercial product was ever released.
- For a number of reasons, including the fact that modern disk drives contain their own internal ECC, RAID 2 is not a practical disk array scheme.
Bit-Interleaved Parity (RAID Level 3)

- One can improve upon memory-style ECC disk arrays (RAID 2) by noting that, unlike memory component failures, disk controllers can easily identify which disk has failed. Thus, one can use a single parity disk rather than a set of parity disks to recover lost information.

- As with RAID 2, RAID 3 must read all data disks for every read operation.
  - This requires synchronized disk spindles for optimal performance, and works best on a single-tasking system with large sequential data requirements. An example might be a system used to perform video editing, where huge video files must be read sequentially.
Block-Interleaved Parity (RAID Level 4)

- RAID 4 is similar to RAID 3 except that blocks of data are striped across the disks rather than bits/bytes.
- Read requests smaller than the striping unit access only a single data disk.
- Write requests must update the requested data blocks and must also compute and update the parity block.
  - For large writes that touch blocks on all disks, parity is easily computed by exclusive-or’ing the new data for each disk.
  - For small write requests that update only one data disk, parity is computed by noting how the new data differs from the old data and applying those differences to the parity block.
- This can be an important performance improvement for small or random file access (like a typical database application) if the application record size can be matched to the RAID 4 block size.

Correction:
Blocks not bits
Block-Interleaved Distributed-Parity (RAID Level 5)

- The block-interleaved distributed-parity disk array eliminates the parity disk bottleneck present in RAID 4 by distributing the parity uniformly over all of the disks.

- An additional, frequently overlooked advantage to distributing the parity is that it also distributes data over all of the disks rather than over all but one.

- RAID 5 has the best small read, large read and large write performance of any redundant disk array.
  - Small write requests are somewhat inefficient compared with redundancy schemes such as mirroring however, due to the need to perform read-modify-write operations to update parity.
Problems of Disk Arrays: Small Writes

RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes

(1. Read) (2. Read) (3. Write) (4. Write)
P+Q Redundancy (RAID Level 6)

- An enhanced RAID 5 with stronger error-correcting codes used.
- One such scheme, called $P+Q$ redundancy, uses Reed-Solomon codes, in addition to parity, to protect against up to two disk failures using the bare minimum of two redundant disks.
- The P+Q redundant disk arrays are structurally very similar to the block-interleaved distributed-parity disk arrays (RAID 5) and operate in much the same manner.
  - In particular, P+Q redundant disk arrays also perform small write operations using a read-modify-write procedure, except that instead of four disk accesses per write requests, P+Q redundant disk arrays require six disk accesses due to the need to update both the ‘P’ and ‘Q’ information.
A logical write becomes four physical I/Os

Independent writes possible because of interleaved parity

Reed-Solomon Codes ("Q") for protection during reconstruction

Targeted for mixed applications

Increasing Logical Disk Addresses

Stripe

Stripe Unit

Disk Columns
RAID 10 (Striped Mirrors)

- RAID 10 (also known as RAID 1+0) was not mentioned in the original 1988 article that defined RAID 1 through RAID 5.
- The term is now used to mean the combination of RAID 0 (striping) and RAID 1 (mirroring).
- Disks are mirrored in pairs for redundancy and improved performance, then data is striped across multiple disks for maximum performance.
- In the diagram below, Disks 0 & 2 and Disks 1 & 3 are mirrored pairs.

![Diagram of RAID 10](image)

- Obviously, RAID 10 uses more disk space to provide redundant data than RAID 5. However, it also provides a performance advantage by reading from all disks in parallel while eliminating the write penalty of RAID 5.
## RAID Levels Summary

<table>
<thead>
<tr>
<th>RAID Level</th>
<th>Minimum number of disk faults survived</th>
<th>Example Data Disks</th>
<th>Corresponding Check Disks</th>
<th>Widely Used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Non-Redundant Striped</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>1 Mirrored</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>No</td>
</tr>
<tr>
<td>10 Striped Mirrors</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>Yes</td>
</tr>
<tr>
<td>2 Memory-Style ECC</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>3 Bit-Interleaved Parity</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>4 Block-Interleaved Parity</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>5 Block-interleaved</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Distributed Parity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 P + Q Redundancy</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>No</td>
</tr>
</tbody>
</table>
### RAID Levels Comparison: Throughput Per Dollar Relative to RAID Level 0.

<table>
<thead>
<tr>
<th>RAID level</th>
<th>Small Read</th>
<th>Small Write</th>
<th>Large Read</th>
<th>Large Write</th>
<th>Storage Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID level 0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RAID level 1</td>
<td>1</td>
<td>1/2</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>RAID level 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 level 5</td>
<td>1</td>
<td>max(1/G,1/4)</td>
<td>1</td>
<td>(G-1)/G</td>
<td>(G-1)/G</td>
</tr>
<tr>
<td>RAID level 6</td>
<td>1</td>
<td>max(1/G,1/6)</td>
<td>1</td>
<td>(G-2)/G</td>
<td>(G-2)/G</td>
</tr>
</tbody>
</table>

**Table 3: Throughput Per Dollar Relative to RAID Level 0.** This table compares the throughputs of various redundancy schemes for four types of I/O requests. Small here refers to I/O requests of one striping unit; large refers to I/O requests of one full stripe (one stripe unit from each disk in an error-correction group). G refers to the number of disks in an error-correction group. In all cases, the higher the number the better. The entries in this table account for the major performance effects but not some of the second-order effects. For instance, since RAID level 1 stores two copies of the data, a common optimization is to dynamically read the disk whose positioning time to the data is smaller.
RAID Levels Comparison: Throughput Per Dollar Relative to RAID Level 0.

- **Small Reads**
  - RAID 1
  - RAID 3
  - RAID 5 & 6

- **Small Writes**
  - RAID 1
  - RAID 3, 5 & 6
  - RAID 5
  - RAID 6
  - RAID 3
RAID Levels Comparison:
Throughput Per Dollar Relative to RAID Level 0.
RAID Levels Comparison: Throughput Per Dollar Relative to RAID Level 0.
RAID Reliability

- Redundancy in disk arrays is motivated by the need to overcome disk failures.
- When only independent disk failures are considered, a simple parity scheme works admirably. Patterson, Gibson, and Katz derive the mean time between failures for a RAID level 5 to be:

\[
\frac{MTTF\ (disk)^2}{N \ (G - 1) \ MTTR(disk)}
\]

- where \( MTTF(disk) \) is the mean-time-to-failure of a single disk,
- \( MTTR(disk) \) is the mean-time-to-repair/replace of a single disk,
- \( N \) is the total number of disks in the disk array
- \( G \) is the parity group size

- For illustration purposes, let us assume we have:
- 100 disks that each had a mean time to failure (MTTF) of 200,000 hours and a mean time to repair of one hour. If we organized these 100 disks into parity groups of average size 16, then the mean time to failure of the system would be an astounding 3000 years!
- Mean times to failure of this magnitude lower the chances of failure over any given period of time.
System Availability: Orthogonal RAIDs

Array Controller

- String Controller
- String Controller
- String Controller
- String Controller
- String Controller

Data Recovery Group: unit of data redundancy

Redundant Support Components: power supplies, controller, cables
System-Level Availability

Host

I/O Controller

Fully dual redundant

I/O Controller

Array Controller

Array Controller

Recovery Group

with duplicated paths, higher performance can be obtained when there are no failures

Goal: No Single Points of Failure