I/O—Redundant Arrays of Inexpensive Disks

Dr. Yitzhak Birk
EE Department, Technion
Advantages of Disk Arrays

- Improved volumetric density (smaller disks)
- Improved seek times and higher rpm (smaller radius and lower centrifugal force)
- Less expensive to manufacture (yield + wasted material + machining)
- Incremental growth (add a single small disk)
- Easier to offer fault tolerance
- Better access-to-capacity ratio!
Redundant Arrays of Disks

- Files are "striped" across multiple spindles
- Redundancy yields high data availability
  Disks will fail
  Contents reconstructed from data redundantly stored in the array
  - Capacity penalty to store it
  - Bandwidth penalty to update

Techniques:
- Mirroring/Shadowing (high capacity cost)
- Horizontal Hamming Codes (overkill)
- Parity & Reed-Solomon Codes
- Failure Prediction (no capacity overhead!)
Array Reliability

• Reliability of N disks = Reliability of 1 Disk / N

\[
\text{500,000 Hours 70 disks = 7000 hours}
\]

Disk system MTTF: Drops from 55 years to 10 months!

• Arrays without redundancy are too unreliable to be useful, especially for critical or time-critical applications!

Hot spares support reconstruction in parallel with access: very high media availability can be achieved.
Terminology and Error Model

• **Reliability**: measured by the mean time between situations in which there is something that can be fixed.

• **Availability**: measured by the fraction of time during which a system is available (“up”).

• **Fault-tolerance**: the ability to remain available despite being unreliable (operating in the presence of a failure).

• **Reparability**: measured by the mean time to repair (MTTR).

• **Disk error model**: operates correctly or declares failure ==> all “system” redundancy can be used for error correction.

• **Remark**: MTTF, fault-tolerance and MTTR jointly determine availability
Redundant Arrays of Disks (RAID) Techniques

- **Disk Mirroring, Shadowing**
  Each disk is fully duplicated onto its "shadow"
  Logical write = two physical writes
  100% capacity overhead

- **Parity Data Bandwidth Array (RAID 3)**
  Parity computed horizontally
  Logically a single high data bw disk

- **High I/O Rate Parity Array (RAID 5)**
  Interleaved parity blocks
  Independent reads and writes
  Logical write = 2 reads + 2 writes
  Parity + Reed-Solomon codes
Problems of Disk Arrays:
Small Writes

RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes

D0' D0 D1 D2 D3 P

new data old data old parity

(1. Read) (2. Read)

XOR

D0' D1 D2 D3 P'

(3. Write) (4. Write)
XOR in Disk Drive: RAID - S

• Drive can execute XOR between new and current blocks:
  – write = old, (old $\oplus$ new), or new
  – return = old or (old $\oplus$ new)

• Small write:
  – drive with old data:
    » write = $D_{\text{new}}$
    » $x = \text{return} = (D_{\text{old}} \oplus D_{\text{new}})$
  – drive with old parity:
    » write = ($x \oplus P_{\text{old}}$)

• Advantages:
  – two full bus transactions instead of four
  – three blocks traverse bus instead of four
Redundant Arrays of Disks
RAID 1: Disk Mirroring/Shadowing

- Each disk is fully duplicated onto its "shadow"
  Very high availability can be achieved

- Bandwidth sacrifice on write:
  Logical write = two physical writes

- Reads may be optimized (read “nearest” and/or by next available arm)

- Most expensive solution: 100% capacity overhead
  Targeted for high I/O rate, high availability environments
Redundant Arrays of Disks RAID 3: Parity Disk

- Parity computed across recovery group to protect against hard disk failures
- 33% capacity cost for parity in this configuration
- Wider arrays reduce capacity costs, decrease expected availability, increase reconstruction time

- Arms logically synchronized, spindles rotationally synchronized
- Logically a single high capacity, high transfer rate disk

Targeted for high bandwidth applications: Scientific, Image Processing
**Redundant Arrays of Disks RAID 5+: High I/O Rate Parity**

A logical write becomes four physical I/Os. Independent writes possible because of interleaved parity. Reed-Solomon Codes ("Q") for protection during reconstruction.

<table>
<thead>
<tr>
<th>D0</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>P</td>
<td>D7</td>
</tr>
<tr>
<td>D8</td>
<td>D9</td>
<td>P</td>
<td>D10</td>
<td>D11</td>
</tr>
<tr>
<td>D12</td>
<td>P</td>
<td>D13</td>
<td>D14</td>
<td>D15</td>
</tr>
<tr>
<td>P</td>
<td>D16</td>
<td>D17</td>
<td>D18</td>
<td>D19</td>
</tr>
<tr>
<td>D20</td>
<td>D21</td>
<td>D22</td>
<td>D23</td>
<td>P</td>
</tr>
</tbody>
</table>

Increasing Logical Disk Addresses

**Stripe**

**Stripe Unit**

Targeted for mixed applications

Disk Columns
Contiguous Data + Parity Striping

Data blocks are stored contiguously on same disk to the extent that any single access goes to a single disk:
- less work spent per task ==> higher throughput
- fewer resources assignable to single task ==> higher latency
- optimized for medium size read-only trans. with thpt more critical than latency.
- all writes are “small”

![Diagram showing data block storage and parity](image)
Perf. Measure: Work per Task

• Think of disk drives as workers

• Work per Task = total “man hours” spent by disk drives on behalf of the task.
  – Parity striping: one seek + one rotational latency + (request size / single-disk transfer rate)
  – RAID 3: M seeks + M rotational latencies + (“”)

• RAID 3 has the lowest latency at low loads
  – throws all the resources at the task ==> shortest service time, but
  – when the load is high, queuing delay dominates and is worst for RAID 3 due to the “waste” of work per task!
Another Reason for Striping: Load Balancing

- **Def.** uniform distribution of load among disks regardless of access pattern.
- Mirroring: excellent
- RAID 3: excellent
- RAID 5: OK
- Parity striping: OK for parity; may be a problem for data.
Subsystem Organization

- **host**
  - manages interface to host, DMA
  - control, buffering, parity logic
  - physical device control

- **host adapter**

- **array controller**
  - manages interface to host, DMA
  - physical device control

- **single board disk controller**
  - often piggy-backed in small format devices

- **single board disk controller**

- **single board disk controller**

- **single board disk controller**

- **single board disk controller**

- **single board disk controller**

- **single board disk controller**

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

- no applications modifications
- no reduction of host performance
**System Availability: Orthogonal RAIDs**

- **Array Controller**
  - **String Controller**
    - ... (replicating the structure)

**Data Recovery Group:** unit of data redundancy

**Redundant Support Components:** fans, power supplies, controller, cables

**End to End Data Integrity:** internal parity protected data paths
System-Level Availability

Fully dual redundant

I/O Controller

Array Controller

host

host

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

Goal: No Single Points of Failure

Recovery Group
Automatically Managed Removable Media

- The lowest level of the storage hierarchy
- “jukeboxes” of tape cartridges, CD-ROM, DVD, etc.
Data Compression Issues

Peripheral Manufacturer Approach:

System Approach:

Hints from Host

Compression Done Here

20:1

Data Specific Compression

2,3:1
Recent Trend: Networked Storage

• General idea:
  – Storage subsystem attached to the network, and available to multiple host computers.
  – permit data transfers to occur directly between the disk drive and the requesting computer, not via server.
  – Control, security etc. may still be handled by server

• Network-Attached Storage (Secure) Devices (NASD): boxes containing storage communicate with hosts using file semantics (like file servers)

• Storage-Area Networks (SAN):
  – boxes containing storage communicate with hosts using block (low-level) semantics.
  – a controller acts as a mediator, hiding details of the storage organization from the host computers.