Disk Arrays: Reliability?

- MTTF of $N$ disks = MTTF of 1 Disk ÷ $N$
  
  $50,000$ Hours ÷ $70$ disks = $700$ hours
  
  Disk system MTTF: Drops from $6$ years to $1$ month!

- Arrays (without redundancy) too unreliable to be useful!

Redundant Arrays of (Inexpensive) Disks

- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  - Capacity penalty to store redundant info
  - Bandwidth penalty to update redundant info
RAID 1: Disk Mirroring/Shadowing

- Each disk is fully duplicated onto its “mirror”
- Very high availability can be achieved
- Bandwidth sacrifice on write:
  Logical write = two physical writes
- Reads may be optimized
- Most expensive solution: 100% capacity overhead

(RAID 2 not interesting, so skip)

RAID 3: Parity Disk

10010011
11001101
10010011

logical record

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Striped physical records

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

P contains sum of other disks per stripe mod 2 (“parity”)

If disk fails, subtract P from sum of other disks to find missing information
RAID 3

- Sum computed across recovery group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk: good for large transfers
- Wider arrays reduce capacity costs, but decreases availability
- 33% capacity cost for parity if 3 data disks and 1 parity disk

Inspiration for RAID 4

- RAID 3
  - Relies on parity disk to discover errors on Read

- RAID 4
  - But every disk sector has an error detection field (built-in)
  - To catch errors on read, rely on error detection field
  - Allows independent reads to different disks simultaneously
RAID 4: High I/O Rate Parity

Example: small read D0 & D5, large write D12-D15

Inspiration for RAID 5

- RAID 4 works well for small reads
- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk
  - Option 2: since P has old sum, compare old data to new data, add the difference to P
- Small writes are limited by Parity Disk: Write to D0, D5 both also write to P disk
RAID 5: High I/O Rate
Interleaved Parity

Independent writes possible because of interleaved parity

Example: write to D0, D5 uses disks 0, 1, 3, 4

RAID 6: Recovering from 2 failures

- Why > 1 failure recovery?
  - operator accidentally replaces the wrong disk during a failure
  - since disk bandwidth is growing more slowly than disk capacity, the MTT Repair a disk in a RAID system is increasing
  ⇒ increases the chances of a 2nd failure during repair since takes longer

- Like RAID 5, but 2 lots of parity
Summary: RAID Techniques

- **Disk Mirroring, Shadowing (RAID 1)**
  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