COMP 7970 Storage Systems

Dr. Xiao Qin

Department of Computer Science and Software Engineering
Auburn University
http://www.eng.auburn.edu/~xqin
xqin@auburn.edu
Problems of Disk Arrays

Small Writes

*RAID-5: Small Write Algorithm*

1 Logical Write = 2 Physical Reads + 2 Physical Writes

Diagram:

1. Read (old data)
2. Read (old parity)
3. Write (new data)
4. Write (old data)
RAID 6: P + Q Redundancy

• An extension to RAID 5 but with two-dimensional parity.
• Each row has P parity and each row has Q parity. (Reed-Solomon Codes)
• Has an extremely high data fault tolerance and can sustain multiple simultaneous drive failures
• Rarely implemented

More information, please see the paper:

A tutorial on Reed-Solomon Coding for Fault Tolerance in RAID-like Systems

COMP 7970, Auburn University
Berkeley History: RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software

- Today RAID is $24 billion dollar industry, 80% nonPC disks sold in RAIDs
Summary: RAID Techniques: Goal was performance, popularity due to reliability of storage

- **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
  - Logical write = 2 reads + 2 writes
Definitions

• Examples on why precise definitions so important for reliability

• Is a programming mistake a fault, error, or failure?
  – Are we talking about the time it was designed or the time the program is run?
  – If the running program doesn’t exercise the mistake, is it still a fault/error/failure?

• If an alpha particle hits a DRAM memory cell, is it a fault/error/failure if it doesn’t change the value?
  – Is it a fault/error/failure if the memory doesn’t access the changed bit?
  – Did a fault/error/failure still occur if the memory had error correction and delivered the corrected value to the CPU?
IFIP Standard terminology

- Computer system **dependability**: quality of delivered service such that reliance can be placed on service
- **Service** is observed **actual behavior** as perceived by other system(s) interacting with this system’s users
- Each module has ideal **specified behavior**, where **service specification** is agreed description of expected behavior
- A system **failure** occurs when the actual behavior deviates from the specified behavior
- failure occurred because an **error**, a defect in module
- The cause of an error is a **fault**
- When a fault occurs it creates a **latent error**, which becomes **effective** when it is activated
- When error actually affects the delivered service, a failure occurs (time from error to failure is **error latency**)
Fault v. (Latent) Error v. Failure

- An error is manifestation *in the system* of a fault, a failure is manifestation *on the service* of an error

- Is If an alpha particle hits a DRAM memory cell, is it a fault/error/failure if it doesn’t change the value?
  - Is it a fault/error/failure if the memory doesn’t access the changed bit?
  - Did a fault/error/failure still occur if the memory had error correction and delivered the corrected value to the CPU?

- An alpha particle hitting a DRAM can be a fault
- if it changes the memory, it creates an error
- error remains latent until effected memory word is read
- if the effected word error affects the delivered service, a failure occurs
Fault Categories

1. Hardware faults: Devices that fail, such as alpha particle hitting a memory cell
2. Design faults: Faults in software (usually) and hardware design (occasionally)
3. Operation faults: Mistakes by operations and maintenance personnel
4. Environmental faults: Fire, flood, earthquake, power failure, and sabotage
   - Also by duration:
     1. Transient faults exist for limited time and not recurring
     2. Intermittent faults cause a system to oscillate between faulty and fault-free operation
     3. Permanent faults do not correct themselves over time
Fault Tolerance vs Disaster Tolerance

- **Fault-Tolerance (or more properly, Error-Tolerance):** mask local faults
  (prevent errors from becoming failures)
  - RAID disks
  - Uninterruptible Power Supplies
  - Cluster Failover

- **Disaster Tolerance:** masks site errors
  (prevent site errors from causing service failures)
  - Protects against fire, flood, sabotage,..
  - Redundant system and service at remote site.
  - Use design diversity

From Jim Gray's "Talk at UC Berkeley on Fault Tolerance" 11/9/00
Mean Time to System Failure (years) by Cause

<table>
<thead>
<tr>
<th>Year</th>
<th>Software</th>
<th>Hardware</th>
<th>Maintenance</th>
<th>Operations</th>
<th>Environment</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1985</td>
<td>2</td>
<td>29</td>
<td>45</td>
<td>99</td>
<td>142</td>
<td>8</td>
</tr>
<tr>
<td>1987</td>
<td>53</td>
<td>91</td>
<td>162</td>
<td>171</td>
<td>214</td>
<td>20</td>
</tr>
<tr>
<td>1990</td>
<td>33</td>
<td>310</td>
<td>409</td>
<td>136</td>
<td>346</td>
<td>21</td>
</tr>
</tbody>
</table>

Problem: Systematic Under-reporting

From Jim Gray's "Talk at UC Berkeley on Fault Tolerance" 11/9/00

COMP 7970, Auburn University
Is Maintenance the Key?

- Rule of Thumb: Maintenance 10X HW
  - so over 5 year product life, ~ 95% of cost is maintenance

VAX crashes ‘85, ‘93 [Murp95]; extrap. to ‘01
 Sys. Man.: N crashes/problem, SysAdmin action
  - Actions: set params bad, bad config, bad app install

HW/OS 70% in ‘85 to 28% in ‘93. In ‘01, 10%?
HW Failures in Real Systems: Tertiary Disks

- A cluster of 20 PCs in seven 7-foot high, 19-inch wide racks with 368 8.4 GB, 7200 RPM, 3.5-inch IBM disks. The PCs are P6-200MHz with 96 MB of DRAM each. They run FreeBSD 3.0 and the hosts are connected via switched 100 Mbit/second Ethernet.

<table>
<thead>
<tr>
<th>Component</th>
<th>Total in System</th>
<th>Total Failed</th>
<th>% Failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCSI Controller</td>
<td>44</td>
<td>1</td>
<td>2.3%</td>
</tr>
<tr>
<td>SCSI Cable</td>
<td>39</td>
<td>1</td>
<td>2.6%</td>
</tr>
<tr>
<td>SCSI Disk</td>
<td>368</td>
<td>7</td>
<td>1.9%</td>
</tr>
<tr>
<td>IDE Disk</td>
<td>24</td>
<td>6</td>
<td>25.0%</td>
</tr>
<tr>
<td>Disk Enclosure -Backplane</td>
<td>46</td>
<td>13</td>
<td>28.3%</td>
</tr>
<tr>
<td>Disk Enclosure - Power Supply</td>
<td>92</td>
<td>3</td>
<td>3.3%</td>
</tr>
<tr>
<td>Ethernet Controller</td>
<td>20</td>
<td>1</td>
<td>5.0%</td>
</tr>
<tr>
<td>Ethernet Switch</td>
<td>2</td>
<td>1</td>
<td>50.0%</td>
</tr>
<tr>
<td>Ethernet Cable</td>
<td>42</td>
<td>1</td>
<td>2.3%</td>
</tr>
<tr>
<td>CPU/Motherboard</td>
<td>20</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>
Does Hardware Fail Fast? 4 of 384 Disks that failed in Tertiary Disk

<table>
<thead>
<tr>
<th>Messages in system log for failed disk</th>
<th>No. log msgs</th>
<th>Duration (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Failure</strong> (Peripheral device write fault [for] Field Replaceable Unit)</td>
<td>1763</td>
<td>186</td>
</tr>
<tr>
<td><strong>Not Ready</strong> (Diagnostic failure: ASCQ = Component ID [of] Field Replaceable Unit)</td>
<td>1460</td>
<td>90</td>
</tr>
<tr>
<td><strong>Recovered Error</strong> (Failure Prediction Threshold Exceeded [for] Field Replaceable Unit)</td>
<td>1313</td>
<td>5</td>
</tr>
<tr>
<td><strong>Recovered Error</strong> (Failure Prediction Threshold Exceeded [for] Field Replaceable Unit)</td>
<td>431</td>
<td>17</td>
</tr>
</tbody>
</table>
High Availability System Classes
Goal: Build Class 6 Systems

<table>
<thead>
<tr>
<th>System Type</th>
<th>Unavailable (min/year)</th>
<th>Availability</th>
<th>Availability Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unmanaged</td>
<td>50,000</td>
<td>90.%</td>
<td>1</td>
</tr>
<tr>
<td>Managed</td>
<td>5,000</td>
<td>99.%</td>
<td>2</td>
</tr>
<tr>
<td>Well Managed</td>
<td>500</td>
<td>99.9%</td>
<td>3</td>
</tr>
<tr>
<td>Fault Tolerant</td>
<td>50</td>
<td>99.99%</td>
<td>4</td>
</tr>
<tr>
<td>High-Availability</td>
<td>5</td>
<td>99.999%</td>
<td>5</td>
</tr>
<tr>
<td>Very-High-Availability</td>
<td>.5</td>
<td>99.9999%</td>
<td>6</td>
</tr>
<tr>
<td>Ultra-Availability</td>
<td>.05</td>
<td>99.99999%</td>
<td>7</td>
</tr>
</tbody>
</table>

UnAvailability = MTTR/MTBF
Can cut it in ½ by cutting MTTR or MTBF

From Jim Gray’s “Talk at UC Berkeley on Fault Tolerance” 11/9/00

COMP 7970, Auburn University
How Realistic is "5 Nines"?

• HP claims HP-9000 server HW and HP-UX OS can deliver 99.999% availability guarantee “in certain pre-defined, pre-tested customer environments”
  – Application faults?
  – Operator faults?
  – Environmental faults?

• Collocation sites (lots of computers in 1 building on Internet) have
  – 1 network outage per year (~1 day)
  – 1 power failure per year (~1 day)

• Microsoft Network unavailable recently for a day due to problem in Domain Name Server: if only outage per year, 99.7% or 2 Nines