Failure Analysis of Two Internet Services

Archana Ganapathithi

(archanag@cs.berkeley.edu)
Motivation

• 24x7 availability is important for Internet Services

• Unavailability = MTTR/(MTTR+MTTF)

• Determine reasons for failure and time to repair!
Recently in the News...

“Technicians were installing routers to upgrade the .Net Messenger service, which underlies both Windows Messenger and MSN Messenger. The technicians incorrectly configured the routers. Service was out from 9 a.m. to 2 p.m. Eastern time on Monday. Ironically, the routers were being installed to make the service more reliable.”

--posted on cnn.com techweb on Wednesday January 8th, 2003
Recently in the News...

“AN INTERNET ROUTING error by AT&T effectively shut off access from around 40 percent of the Internet to several major Microsoft Web sites and services on Thursday, Microsoft has said.

Access to Microsoft's Hotmail, Messenger, and Passport services as well as some other MSN services was cut for more than an hour after AT&T made routing changes on its backbone. The changes were made in preparation for the addition of capacity between AT&T and Microsoft that is meant to improve access to the services hit by the outage, said Adam Sohn, a spokesman for Microsoft.”

--posted on infoworld.com on Thursday January 9th, 2003
Approach

• Obtain real failure data from three Internet Services
  – Data in this talk is from two of them
• Validate/characterize failure based on post mortem reports
• Investigate failure mitigation techniques
# Internet Services

<table>
<thead>
<tr>
<th></th>
<th>Online Service/Internet Portal</th>
<th>Global Content Hosting Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hits/day</strong></td>
<td>~100 million</td>
<td>~7 million</td>
</tr>
<tr>
<td><strong>Collocation sites</strong></td>
<td>2 (500 machines)</td>
<td>15 (500 machines)</td>
</tr>
<tr>
<td><strong>Platform</strong></td>
<td>Solaris on SPARC &amp; x86</td>
<td>Open source x86 OS</td>
</tr>
<tr>
<td><strong># of months studied</strong></td>
<td>7 months</td>
<td>3 months</td>
</tr>
<tr>
<td><strong># of problems</strong></td>
<td>296</td>
<td>205</td>
</tr>
</tbody>
</table>
Internet Service Architecture

FE - forward user requests to BE and possibly process data that is returned

BE - data storage units

NET - interconnections between FE and BE
ROC techniques to mitigate failure
Types of Failures

• **Component Failure**
  - Not visible to customers
  - If not masked, evolves into a...

• **Service Failure**
  - Visible to customers
  - Service unavailable to end user or significant performance degradation
  - Always due to a component failure
Note: these are the statistics for 7 months of data
Note: these are the statistics for 3 months of data
Observations

- Hardware component failures are masked well in both services
- Operator induced failures are hardest to mask
- Compared to Online, Content is less successful in masking failures, especially operator-induced failures
  
  (Online: 33% unmasked, Content: 50% unmasked)
- Note: more than 15% of problems tracked at Content pertain to administrative/operations machines or services
## Service Failure Cause by Location

<table>
<thead>
<tr>
<th></th>
<th>Front End</th>
<th>Back End</th>
<th>Network</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online</strong></td>
<td>77%</td>
<td>3%</td>
<td>18%</td>
<td>2%</td>
</tr>
<tr>
<td><strong>Content</strong></td>
<td>66%</td>
<td>11%</td>
<td>18%</td>
<td>4%</td>
</tr>
</tbody>
</table>
Service Failure Cause by Component

Online

- Operator: 35%
- Hardware: 26%
- Software: 28%
- Unknown: 11%

Total: 61 failures in 12 months

Content

- Operator: 38%
- Hardware: 4%
- Software: 26%
- Unknown: 32%

Total: 56 failures in 3 months
## Time to Repair (hr)

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Median</th>
<th>Average</th>
<th>Total # of problems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ONLINE 8 months</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>0.25</td>
<td>50.9</td>
<td>4.5</td>
<td>12.4</td>
<td>13</td>
</tr>
<tr>
<td>SW</td>
<td>0.5</td>
<td>21.4</td>
<td>3.4</td>
<td>6.6</td>
<td>10</td>
</tr>
<tr>
<td>HW</td>
<td>1.2</td>
<td>25.8</td>
<td>2.5</td>
<td>5.7</td>
<td>7</td>
</tr>
<tr>
<td><strong>CONTENT 2 months</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>0.2</td>
<td>72.9</td>
<td>2.9</td>
<td>21.1</td>
<td>5</td>
</tr>
<tr>
<td>SW</td>
<td>0.5</td>
<td>9.6</td>
<td>1.4</td>
<td>2.7</td>
<td>6</td>
</tr>
<tr>
<td>HW</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
</tr>
</tbody>
</table>
Customer Impact

• Part of Customers Affected:
  - Part/All/None
  - Online: 50% affected part, 50% affected all customers (3 months of data)
  - Content: 100% affected part of customers (2 months of data)
Multiple Event Failures

Vertically Cascaded Failures:
chain of cascaded component failures that lead to service failure.

Horizontally Related Failures:
multiple independent component failures that contribute to service failure.
Multiple Event Failure Results

- **Online Service Failures (3 months of data)**
  - 41% Vertically Cascaded
  - 0% Horizontally Related

- **Content Service Failures (2 months of data)**
  - 0% Vertically Cascaded
  - 25% Horizontally Related
# Service Failure Mitigation Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online testing</td>
<td>26</td>
</tr>
<tr>
<td>Expose/monitor failures to reduce TTR</td>
<td>12</td>
</tr>
<tr>
<td>Expose/monitor failures to reduce TTD</td>
<td>11</td>
</tr>
<tr>
<td>Configuration checking</td>
<td>9</td>
</tr>
<tr>
<td>Redundancy</td>
<td>9</td>
</tr>
<tr>
<td>Online fault/load injection</td>
<td>6</td>
</tr>
<tr>
<td>Component Isolation</td>
<td>5</td>
</tr>
<tr>
<td>Pre-deployment fault/load injection</td>
<td>3</td>
</tr>
<tr>
<td>Proactive restart</td>
<td>3</td>
</tr>
<tr>
<td>Pre-deployment correctness testing</td>
<td>2</td>
</tr>
</tbody>
</table>

Total = 40 service failures
Conclusions/Lessons

• Operator errors are most impacting
  - Largest single cause of failure
  - Largest MTTR

• Most valuable failure mitigation techniques are:
  - Online testing
  - Better monitoring tools, and better exposure of component health and error conditions
  - Configuration testing and increased redundancy
Research Directions

• Improve classification based on Vertically Cascading and Horizontally Related service failures
• Further explore failure mitigation techniques
• Investigate failure models based on time of day
• Apply current statistics to develop accurate benchmarks
Related Work

• “Why do Internet services fail, and what can be done about it?” (Oppenheimer D.)
• “Failure analysis of the PSTN” (Enriquez P.)
• “Why do computers stop and what can be done about it?” (Gray J.)
• “Lessons from giant-scale services” (Brewer E.)
• “How fail-stop are faulty programs?” (Chandra S. and Chen P.M.)
• “Networked Windows NT System Field Failure Data Analysis” (Xu J., Kalbarczyk Z. and Iyer R.)