Monitoring and Measuring Server Availability

Lee Liang Kang, Alex (3047809)
Supervisor: Dr Tim Moors
Assessor: Dr Vijay Sivaraman
Project Motivation

- Internet services not always available
- E.g. HTTP, DNS, SMTP, POP
- Telnet does not provide TCP level details
- Need mechanisms to measure server availability
- Inform users of highly available services and to guide research
Previous Work

- Instantaneous availability (MTTF/MTTF+MTTR)
- Server availability from end-user perspective
- TCP connection problems e.g. ACK loss, delayed response, high server processing time
- Analyze server partial availability based on status codes

MTTF : Mean time to failure / MTTR: Mean time to repair
Project Objectives

➢ Develop server monitoring tool that
  - Measures availability based on MTTF, MTTR
  - Analyze status codes returned by server
  - Monitors TCP server connection establishment
  - Translates measurements to user perspective data
  - Logs statistical availability data
Project Challenges

- Able to support multiple Internet services e.g. HTTP, DNS, SMTP etc
- Handle multiple IP address per Internet domain
- Accurately measure availability metrics
- Provide end-user perspective data
- Monitor packets during TCP connection phase
Server’s connections demand (uluru.ee.unsw.edu.au)

- Connection requests peak during the day and late night
Server Monitoring

- Monitoring of WebCT website (WebSitePulse.com)
- Provide availability data and response times etc
- Not sufficient

<table>
<thead>
<tr>
<th>Location</th>
<th>Last check</th>
<th>Status</th>
<th>Resp. time (ms.)</th>
<th>Ping Loss</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brisbane, Australia</td>
<td>05/15/2005 16:58 GMT 9:00</td>
<td>OK</td>
<td>324.9980</td>
<td>0.00%</td>
<td>19.51</td>
<td>19.98</td>
<td>20.74</td>
</tr>
<tr>
<td>San Francisco, CA</td>
<td>05/15/2005 16:57 GMT 9:00</td>
<td>OK</td>
<td>421.4650</td>
<td>0.00%</td>
<td>160.68</td>
<td>161.07</td>
<td>161.33</td>
</tr>
</tbody>
</table>
Connection Problems

- DNS failure
- Redirection errors
- Resource non existent
- Internal server errors
- TCP connection failure
- Caching
Preliminary Data

Data obtained from several university websites of different geographical region

<table>
<thead>
<tr>
<th>Location</th>
<th>Success</th>
<th>Client</th>
<th>Redirection</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNSW</td>
<td>24049</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Australia</td>
<td>23590</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>North America</td>
<td>25528</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Asia</td>
<td>24064</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
DNS Mapping

- DNS failure cause inability to resolve hostname
- High DNS response time

Approach
- Monitor gethostbyname() function
- Measure response times, retries
- Availability uncertain for that interval
TCP Connection

- Connection involves 3 way handshake
- TCP connection errors
  - SYN-ACK packet loss
  - High server processing time
- Utilize packet capture library (pcap.h) to
  - Monitor 3 way handshake, check packet losses
  - Measure response times, number of attempts
  - Analyze incoming/outgoing packets e.g. ICMP from router
Measuring Servers Availability

- Analyze returned packets from server
- Determine the right probing resolution
- Monitor multiple IP address per server domain
- Adaptive probing rate
  - When failure detected, increase probing resolution to measure MTTR
  - Probing resolution back to normal after repair
  - Use MTTF/MTTF+MTTR to compute availability
Availability – User’s perspective

- Server failures during non-peak times less severe than peak
- Availability statistics such as uptime insufficient
- Approach
  - Weightage for different times of day
  - Translate availability time to end-user impact
  - E.g. server outage minutes, degree of service
Availability – Scenarios

Good Availability

Ideal Availability

Poor Availability
Thesis A Progress Report

Research/analysis
- Literature review
- SMTP, POP, DNS using Telnet
- Packet capture library in C
- Server demand patterns

Programming implementation
- DNS/HTTP client
- Data logger
Web Client Implementation

- SAMPLE OUTPUT – www.google.com.au

Query summary
Time: Tue May 24 17:50:03 2005
IP address: 66.102.7.147 Status code: 200 Response time: 0.193251848 seconds
IP address: 66.102.7.99 Status code: 200 Response time: 0.168308973 seconds
IP address: 66.102.7.104 Status code: 200 Response time: 0.169835091 seconds

- Response time in nanoseconds
- Incorporate data logging feature
Thesis B Plan

INTERFACE

SERVERS

CONTROLLER

LOG FILE

Request

Response

HTTP

DNS

SMTP

POP

Send Data

Log Analysis
# Thesis B Plan

<table>
<thead>
<tr>
<th>Task</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weightage for server connections demand patterns</td>
<td>July 28&lt;sup&gt;th&lt;/sup&gt;</td>
<td>3&lt;sup&gt;rd&lt;/sup&gt; Aug</td>
<td>7 Days</td>
</tr>
<tr>
<td>TCP connection monitoring module</td>
<td>4&lt;sup&gt;th&lt;/sup&gt; Aug</td>
<td>17&lt;sup&gt;th&lt;/sup&gt; Aug</td>
<td>14 Days</td>
</tr>
<tr>
<td>DNS mapping monitoring module</td>
<td>18&lt;sup&gt;th&lt;/sup&gt; Aug</td>
<td>22&lt;sup&gt;nd&lt;/sup&gt; Aug</td>
<td>5 Days</td>
</tr>
<tr>
<td>SMTP/POP/FTP/HTTP monitoring modules</td>
<td>23&lt;sup&gt;rd&lt;/sup&gt; Aug</td>
<td>12&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>21 Days</td>
</tr>
<tr>
<td>Translating server downtime to user perspective statistics</td>
<td>13&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>19&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>7 Days</td>
</tr>
<tr>
<td>User interface module</td>
<td>20&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>26&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>7 Days</td>
</tr>
<tr>
<td>Statistical log analysis/display</td>
<td>27&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>30&lt;sup&gt;th&lt;/sup&gt; Sept</td>
<td>4 Days</td>
</tr>
<tr>
<td>Modules integration, testing and performance analysis</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; Oct</td>
<td>8&lt;sup&gt;th&lt;/sup&gt; Oct</td>
<td>8 Days</td>
</tr>
<tr>
<td>Additional implementation (downtime cause determination, more servers)</td>
<td>9&lt;sup&gt;th&lt;/sup&gt; Oct</td>
<td>18&lt;sup&gt;th&lt;/sup&gt; Oct</td>
<td>10 Days</td>
</tr>
<tr>
<td>Thesis consolidation and submission</td>
<td>19&lt;sup&gt;th&lt;/sup&gt; Oct</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; Nov</td>
<td>14 Days</td>
</tr>
</tbody>
</table>
Questions?