E-Commerce and Reliable Systems

Design for Stability
Presentation Objectives

- Describe successful E-Commerce environments
- Describe system and application behavior
- Describe system and application components
- Discuss holistic design approach
- Discuss true stability
  - Difficult to achieve in the E-Commerce marketplace
  - Worth the effort
E-Commerce Landscape

• Participants in the Internet marketplace are interested in the electronic buying and selling of goods and services
• Participants always want their marketplace objectives to be met
• The bad parts of the Internet marketplace is are the ones customers remember
• Lack of understanding of the marketplace component technologies blocks successful participant experience
• The Internet marketplace is only as good as it’s weakest link
• The Internet marketplace is a collection of services, each with a different level of stability
• The responsibility of creating a stable Internet marketplace lays on those who are empowered to effect change: you make the difference
Customer Landscape

• **Customer connection**
  Ability of customer to connect to Internet Service Provider (ISP) to access the internet

• **Internet peering points**
  Ability to traverse the Internet mesh from ISP to other ISP’s

• **Firewall and proxy support**
  Ability to open sockets and route traffic through a secure intermediary

• **Internet latency**
  Ability to send and receive data from a customer, through ISP routers, to the server, and back in a timely fashion

• **Browser versions and supported functions**
  Ability of customer browser to execute the HTML, DHTML, Java applets or Javascript code returned from the server
Behavior Definitions

• **Availability**
  The ability to always provide a response to a request.

• **Supportability**
  The ability to administrate, monitor and disable services

• **Performance**
  The ability to respond to a request in a timely manner

• **Capacity**
  The ability to provide responses to all simultaneous requests made to a site

• **Usability**
  The ability for customers to easily navigate to the desired target

• **Maintainability**
  The ability to install and upgrade services

• **Extensibility**
  The ability to add features as the services evolve

• **Feature Set**
  The ability to provide the services customers request
System Redundancy

• Multi-site design
  – Install identical systems in multiple and geographically disperse locations
  – Distribute customers to all geographic sites

• Multi-system design
  – Customer accesses multiple systems of the same type
  – Customer is sent to a backup system when primary system fails

• Data Replication
  – One read-write master to many replicas
  – Synchronization time across replicas
  – Real time updates versus batch cycle updates
Customer Load Balancing

INTERNET

LOAD BALANCE TECHNOLOGY

CUSTOMERS

SITE "A"

SITE "B"

FT-A

A-1

A-1

A-3

B-1

B-2

B-3

FT-B

WEB SERVER CLUSTER

WEB SERVER CLUSTER

WEB SERVER CLUSTER

WEB SERVER CLUSTER

WEB SERVER CLUSTER

WEB SERVER CLUSTER

WEB SERVER CLUSTER

FAILURE ROUTE AWAY

FAILURE ROUTE AWAY
Transient TCP Connections

- Connections are established, data is requested, data is delivered, connection is immediately terminated.
- A small number of sockets can service a large number of requests if requests are short enough to support arrival rates.
- Socket setup overhead may overload system.
- Socket address limitations may cause unavailability.

**Sequential & Non-Overlapping**

**Socket & Data Requests**

1. Transient TCP Connections
2. A small number of sockets can service a large number of requests if requests are short enough to support arrival rates.
3. Socket setup overhead may overload system.
4. Socket address limitations may cause unavailability.
Persistent TCP Connections

- Connections are established, data is requested, data is delivered, connection remains for subsequent requests until thread connection is terminated
- The number of requests that can be serviced equals the number of threaded sockets available
- Thread limitations may cause unavailability
Application Threads of Execution

- **Race condition**
  One thread changes shared resources and causes another thread to fail

- **Deadly embrace**
  Two threads block and wait for each other to complete

- **Exhaustion**
  All threads are waiting to complete and no new threads are available

- **Starvation**
  Attempts are made to start a new thread when no new threads are available

- **Pooling**
  A larger number of thread requestors successfully share a smaller number of actual threads
### Application Thread Model Example

#### PRODUCT "A" SECURE/443

<table>
<thead>
<tr>
<th>ZONE A</th>
<th>WEB</th>
<th>AT</th>
<th>EP</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST A</td>
<td>3204</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZONE B</th>
<th>WEB</th>
<th>AT</th>
<th>EP</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST B-1</td>
<td>128</td>
<td>4</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>HOST B-2</td>
<td>128</td>
<td>4</td>
<td>512</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZONE C</th>
<th>WEB</th>
<th>AT</th>
<th>EP</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST C</td>
<td>3208</td>
<td>8</td>
<td>1024</td>
<td></td>
</tr>
</tbody>
</table>

#### PRODUCT "B" SECURE/443

<table>
<thead>
<tr>
<th>ZONE A</th>
<th>WEB</th>
<th>AT</th>
<th>EP</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST A</td>
<td>3204</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZONE B</th>
<th>WEB</th>
<th>AT</th>
<th>EP</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST B-1</td>
<td>32</td>
<td>2</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>HOST B-2</td>
<td>32</td>
<td>2</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZONE C</th>
<th>WEB</th>
<th>AT</th>
<th>EP</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST C</td>
<td>32</td>
<td>2</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>

#### HOST C

<table>
<thead>
<tr>
<th>ZONE C / BACKEND</th>
<th>AUTH</th>
<th>CACHE</th>
<th>WATCHDOG</th>
<th>STATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST C</td>
<td>64</td>
<td>4</td>
<td>256</td>
<td>2</td>
</tr>
</tbody>
</table>

#### IN FLIGHT TRANSACTION ESTIMATES AND TOTAL TPS FOR SERVER TYPES

<table>
<thead>
<tr>
<th>KEY:</th>
<th>TAT</th>
<th>FLT</th>
<th>TPS</th>
<th>FAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB SERVER</td>
<td>1024</td>
<td>10</td>
<td>102</td>
<td>100%</td>
</tr>
<tr>
<td>GW</td>
<td>1024</td>
<td>10</td>
<td>102</td>
<td>100%</td>
</tr>
<tr>
<td>TX</td>
<td>1024</td>
<td>10</td>
<td>102</td>
<td>100%</td>
</tr>
<tr>
<td>AUTH</td>
<td>256</td>
<td>3</td>
<td>85</td>
<td>100%</td>
</tr>
<tr>
<td>TX-1</td>
<td>512</td>
<td>7</td>
<td>73</td>
<td>100%</td>
</tr>
<tr>
<td>TX-2</td>
<td>256</td>
<td>4</td>
<td>64</td>
<td>100%</td>
</tr>
<tr>
<td>TX-3</td>
<td>256</td>
<td>4</td>
<td>64</td>
<td>100%</td>
</tr>
<tr>
<td>TX-4</td>
<td>128</td>
<td>4</td>
<td>32</td>
<td>100%</td>
</tr>
</tbody>
</table>

#### PERCENTAGE DISTRIBUTION FOR BACK END SERVERS

<table>
<thead>
<tr>
<th>TT/FL</th>
<th>TT/FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>25%</td>
</tr>
<tr>
<td>TX-1</td>
<td>25%</td>
</tr>
<tr>
<td>TX-2</td>
<td>25%</td>
</tr>
<tr>
<td>TX-3</td>
<td>25%</td>
</tr>
<tr>
<td>TX-4</td>
<td>25%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TT/FL</th>
<th>TT/FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZONE C</td>
<td>125%</td>
</tr>
</tbody>
</table>

#### ALLOCATIONS FOR BACK END SERVERS

<table>
<thead>
<tr>
<th>TT/FL</th>
<th>TT/FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>28</td>
</tr>
</tbody>
</table>

---

21 February, 2001

Jacob C. Lindeman - Copyright (c) Fidelity Investments
Instrumentation

- **Dynamic debug levels**
  Run-time setting of server software to log different levels of detail about what is happening inside the program

- **Concurrency levels**
  The number of active connections at the server

- **Operating system statistics**
  Low-level system internals such as CPU activity, RAM use, disk activity, number of active threads

- **Response times**
  The time taken to respond to various types within a fixed time period

- **Request type profiling and histograms**
  Charts and analysis of quantities and arrival rates of various request types
Prototypical Distributed Computing Model

KEY:
- WEB PRODUCT "A"
- LOAD DISTRIBUTION
- TRANSPORT PRODUCT "B"
- INTERNET CONNECTION
- DATA SOURCE PRODUCT "C"

21 February, 2001 Jacob C. Lindeman - Copyright (c) Fidelity Investments
Resource Exhaustion

- Too many active processes to allocate CPU time
- No free TCP sockets
- No free threads of execution
- No free memory or swap space
- No free disk space
- No free file descriptors
- No free network bandwidth
- Too many scheduled processes
Prototypical Tiered Computing Model
Performance & Capacity

• Little’s Law
  \[ n = \text{number of customers in the system (in flight request)} \]
  \[ t = \text{mean time customers are in the system (response time)} \]
  \[ r = \text{customer arrival rate (new requests)} \]

• Max number of queued in-flight requests
  \[ n = t \times r \]

• Max number of new requests per second
  \[ r = n / t \]

• Moral
  Given a finite set of resources, the slower the response time, the smaller the queue of customers that can be supported

• Choice
  Use fewer resources versus provide more resources
Typical User and System Histograms

Arrival Rate

S/min Average Arrival Rate

TPS

S/min Average TPS

Transaction Load

Response Time

p/read/s

text

s/1 mn

Free swap

pswch/s

21 February, 2001 Jacob C. Lindeman - Copyright (c) Fidelity Investments
TPS Requirements Model

- Goal: Maintain 4:1 Headroom @ 0-8 Sec Delay

<table>
<thead>
<tr>
<th>Server Type</th>
<th>Number of Servers</th>
<th>Current Total TPS</th>
<th>Total TPS Req'd @ 4:1</th>
<th>Per Server TPS Req'd @ 4:1</th>
<th>Current TPS Capacity Per Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>3</td>
<td>32</td>
<td>125</td>
<td>41.8</td>
<td>420 (10:1)</td>
</tr>
<tr>
<td>Broker</td>
<td>3</td>
<td>32</td>
<td>125</td>
<td>41.8</td>
<td>240 (4.8:1)</td>
</tr>
<tr>
<td>Data</td>
<td>1</td>
<td>80</td>
<td>322</td>
<td>322</td>
<td>1353 (4.1:1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avg Data Resp Time (sec)</th>
<th>Concurrent In-Flight Requests</th>
<th>Concurrent Data Requests</th>
<th>Concurrent Data Connections (min - max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>125</td>
<td>288</td>
<td>125 – 288</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>576</td>
<td>250 – 576</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>1152</td>
<td>500 – 1152</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>2304</td>
<td>1000 – 2304</td>
</tr>
<tr>
<td>10</td>
<td>1250</td>
<td>2880</td>
<td>1250 – 2880</td>
</tr>
<tr>
<td>15</td>
<td>1875</td>
<td>4320</td>
<td>1875 - 4320</td>
</tr>
</tbody>
</table>
Middle numbers inside the bubbles represent total capacity in WEB equivalent TPS for all bubbles ahead of MIDDLE and MIDDLE VPS for all bubbles behind MIDDLE. Bottom numbers inside the bubbles represent the headroom for a load of 180 TPS and 60,000 users. Links connecting components show the TPS (black) or VPS (blue) for traffic flowing between components based on a load of 180 WEB TPS and 60,000 users. Underscored numbers are estimated.
Conclusions

• The entire system must be available when customers want to do business on E-Commerce sites
• The success of E-Commerce sites is impacted by Performance, Capacity, Availability, Security, Supportability, Usability, Maintainability, Extensibility, Feature Set & Cost
• Do not wait until the end of the process to consider the whole problem space: do it now!
• Do not sacrifice quality and stability for speed of delivery: the customer will pay the price, and will not return to your site!
• There is no better qualified person than you to work through and solve E-Commerce stability solutions
Reading

• Little laws for utility processes and waiting times in queues
  By Serfozo, R.F.
  Queueing Systems Theory and Applications (Sept. 1994)

• RFC 1180 - A TCP/IP Tutorial
  http://info.internet.isi.edu/in-notes/rfc/files/rfc1180.txt

• Sun Performance and Tuning: SPARC and Solaris
  By Adrian Cockcroft
  http://sunsite.queensu.ca/sunmicro/sunpress/books/Cockcroft/
  Cockcroft.html