Performance Testing and tuning of Web applications

Ajay Walgude

April 2009

“Only conducting Performance testing at the conclusion of system or functional testing is like ordering a diagnostic blood test after the patient is dead.”

Scott Barber
Agenda

- Why Performance testing?
- What and why are problems in web applications?
- Typical problem areas
- Performance engineering, the framework
- Performance testing of Web 2.0
- Performance Tuning and techniques
- Performance Tuning future
- Conclusion

Why Performance testing?

<table>
<thead>
<tr>
<th>Adequacy of developed software performance</th>
<th>Infrastructure Adequacy</th>
<th>Improving efficiency Of Performance tuning</th>
<th>Release Readiness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Criteria:</td>
<td></td>
<td></td>
<td>Release Criteria:</td>
</tr>
<tr>
<td>Functionality:</td>
<td></td>
<td></td>
<td>Quality of Product</td>
</tr>
<tr>
<td>User Acceptance:</td>
<td></td>
<td></td>
<td>Test/Integration</td>
</tr>
<tr>
<td>Business Optimization:</td>
<td></td>
<td></td>
<td>BOP</td>
</tr>
<tr>
<td>Reusability:</td>
<td></td>
<td></td>
<td>OPC/Quality of Service</td>
</tr>
<tr>
<td>Maintainability:</td>
<td></td>
<td></td>
<td>Fasilitation</td>
</tr>
<tr>
<td>Communication:</td>
<td></td>
<td></td>
<td>Configuration Management</td>
</tr>
<tr>
<td>Management:</td>
<td></td>
<td></td>
<td>Issue Management</td>
</tr>
<tr>
<td>Feedback:</td>
<td></td>
<td></td>
<td>Tuning &amp; Security</td>
</tr>
</tbody>
</table>

Lights ON
What are Performance problem areas in a Web system?

- End-user Desktop problems
- ISP performance and peering
- Load balancer problems
- Application server issues
- Internet bottlenecks
- Network vs. Server issues
- Faulty web server within a cluster
- Firewall Performance
- Server HW, device, OS, DB issues
- Content issues

Why do we have bottlenecks?

- **People**
  - Reactive than proactive, Not Thinking About “Load” or “Scalability” or “Performance” During Des & Dev
  - Limited Expertise in System Design / Development

- **Process**
  - Poor Architecture and System Design
  - Changing requirements
  - Changing project plans

- **Technology**
  - Inadequate Hardware
  - Inadequate Network Capacity
  - Poor Implementation / Tuning of Databases and components
Typical performance problems contributors

<table>
<thead>
<tr>
<th>Application Server Issues</th>
<th>Network Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threading issues</td>
<td>Packet loss</td>
</tr>
<tr>
<td>Memory Leaks</td>
<td>Excessive network usage by “chatty” applications</td>
</tr>
<tr>
<td>High Verbose Log File</td>
<td>Hardware related issues (Routers, etc)</td>
</tr>
<tr>
<td>Connection pool management</td>
<td>High Latency</td>
</tr>
</tbody>
</table>

- **Low CPU usage across servers but overall poor performance**

- **High amount of Disk I/O on Servers**

- **Web Server issues**
  - Ineffective Caching
  - Connection Pool management
  - Keep-Alive Connections Flushed
  - Content compression configuration

- **Application Server Issues**
  - Threading issues
  - Memory Leaks
  - High Verbose Log File
  - Connection pool management

- **Database Server Issues**
  - Slow Query response
  - Query and stored procedure cache related issues
  - Deadlocks
  - Temp DB issues

- **Web Server Issues**
  - Ineffective Caching
  - Connection Pool management
  - Keep-Alive Connections Flushed
  - Content compression configuration

- **High Processor Queue length**

How do we find and fix the problems? The framework

Performance engineering is the process by which software is reviewed, tested and tuned with the intent of realizing the required performance. This process aims to optimize the most important application performance trait, user experience.
All work described was performed by Capgemini or a Capgemini affiliate.

Using right methodology…

And right instrumentation…

Performance Testing
- HP: Load Runner and performance Center
- IBM: Rational performance tester
- Borland: Silk Performer
- Radview: Webload
- Network sniffers

Performance Monitoring
- HP: Sitescope
- IBM: Tivoli performance viewer
- Willy: Introscope
- BMC: Patrol
- SNMP Monitors

Performance Profiling
- HP: Diagnostic profiler for Java & .NET
  - JProbe, JProfiler
  - ANTS Profiler
  - SQL Profiler
  - STATSPACK

Performance Tuning
- Vendor Specific tool sets
  - TOAD for Oracle
  - SQL Comparator for SQL Server
  - ...
  - ...

All work described was performed by Capgemini or a Capgemini affiliate.
And appropriate testing type

- Baseline Testing
- Network Testing
- Load Testing
- Stress Testing
- Spike Testing
- Volume Testing
- Endurance Testing

**Network Testing**
- Execution of performance tests where load is injected at strategic network points

**Load Testing**
- Testing the service level when put under real world volumes of data

**Stress Testing**
- Testing the maximum load of transaction/users that can be handled until breakdown point

**Spike Testing**
- Handling large amount of data

**Endurance Testing**
- To find problems that grow over a period of time, during continuous repetitive transactions

**Spike Testing**
- Testing by suddenly increasing the number of concurrent users logged on to determine the server behavior

**Baseline Testing**
- Compares the performance of a new or unknown server to a known reference standard, such as existing software or measurements

All work described was performed by Capgemini or a Capgemini affiliate.
Case Study

- **Data collection**
  - Captured call profiles that helped in “drilling down” to the slow methods
  - Identified improper memory usage trends
  - Collected information on exceptions, slow SQLs, etc.

- **Outcome**
  - Successfully able to track the query causing dead lock.
  - Sybase monitors were indicating only dead locks but not able to point exact query

Following Query was encountering deadlock (Observed from Exceptions tab):

```
select childassoc0_.type_qname as col_0_0_, childassoc0_.qname as col_1_0_, childassoc0_.is_primary as col_2_0_,
childassoc0_.assoc_index as col_3_0_, nodeimpl2_.id as col_4_0_, nodeimpl2_.protocol as col_5_0_, nodeimpl2_.identifier as col_6_0_,
nodeimpl2_.uuid as col_7_0_ from alf_child_assoc childassoc0_ inner join alf_node nodeimpl1_ on
childassoc0_.parent_node_id=nodeimpl1_.id inner join alf_node nodeimpl2_ on childassoc0_.child_node_id=nodeimpl2_.id where
childassoc0_.parent_node_id=? order by childassoc0_.assoc_index, childassoc0_.id
```

SQL state [40001]; error code [1205]; Your server command (family id #0, process id #164) encountered a deadlock situation. Please re-run your command.

Methods Tab

Queries Tab

Web now and then (Web 2.0)

Web 1.0 + Consumer Generated Content + Rich Media + XML & Mash-Ups

**Web 2.0 is,**

Loosely defined term that refer to a second generation of web-based applications and the technologies that support them

The term can also refer to the transformation of websites into computing platforms that support thin client computing and the enablement of greater sharing of information.
Challenges with web 2.0 performance testing

- Multiple web page http\s calls
- Web rich user experience similar to fat clients
- Application responsiveness effected by client activity
- End user experience is greatly effected by number of users
- Fragmented framework market provides many frameworks to test and is shifting fast

Challenges with existing performance testing solutions

- Not able to record and replay multiple HTTP calls per page
- Web applications have hundreds of HTTP calls tools must give these calls a context
  - Link them to page elements
  - Provide logical sequence
  - Create automatic transactions
- Does not isolate slow calls that effect the user experience vs. slower cache calls
- No support for specific frameworks – using the same terminology for reports and providing a higher level abstraction
- Web 2.0 approach refreshes only what is updated on a web page, instead of refreshing the entire page
Web 2.0 Solutions – HP LoadRunner / Performance Center

<table>
<thead>
<tr>
<th>Technology</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA / Web Services</td>
<td>Web Services, Web / Http</td>
</tr>
<tr>
<td>FLASH</td>
<td>FLEX, AMF, Web / Http</td>
</tr>
<tr>
<td>AJAX</td>
<td>AJAX, Web / Http</td>
</tr>
</tbody>
</table>

- Built-in record and replay capabilities for Rich Internet Applications
- Performance testing for Flex & Web services
- Client-side breakdown
- Ajax framework solutions based on Click and Script technology that reduces the scripting process to a few mouse clicks; this includes generic Ajax support and leading framework support (such as Microsoft, Google, etc.)

Performance Tuning

- It is a process of identifying and eliminating bottlenecks
- Performance tuning of Web-based systems is a multi-variable and iterative process

- It is a collaborative effort between cross functional teams
- Performance tester is an enabler since he provides the necessary information in an efficient manner
Performance Tuning Process

- Define Benchmark
- Establish Baseline
- Collect Data
- Analyze results
- Identify Bottleneck
- Configure and Tune application / components
- Test and Measure
- Keep monitoring…

Performance Tuning Techniques: There are no silver bullets

- There are multiple parameters to tune
- Each component has tuning possibilities but not every component needs tuning
- Tuning methodology for Application server is a bit different compared to the application or database tuning
- There may be no single good configuration for all env.
- Faster response time is essential but there’s more to it
  - Data Integrity
Future of performance tuning

- Real time Performance optimization?
  - Fix issues as they occur
- Automatic/Adaptive performance parameter tuning?
  - Monitor performance
  - Adjust runtime parameters

Conclusion

Performance not be taken as a final step in SDLC

Know your system well
- Collect history data
- Read logs
- Stress test/ soak test application at Peak load
- Keep monitoring

End-user experience is not all
- Checking further into the system for critical performance bottlenecks may avoid server crashes only few days after production

There are no silver bullets for performance tuning, it’s a collaborative effort
References and acknowledgements

- www.sqe.com/ControllImages/sqe/Image/Webinars
- “Load Testing 2.0 for Web 2.0”, an HP white paper
- www.nasscom.in/upload/59145/Day%202/SE-4B/Perf%20%20Security%20Management(Sai%20Chintala)

Q & A