Performance Testing: Roles, Activities, and QA Inclusion

Michael Lawler
NueVista Group
Today’s Agenda

• Outline the components of a performance test and considerations
• Discuss various roles, tasks, and activities
• Review several design approaches specific to performance testing
Performance Testing Definition

• A category of testing in which the system's performance requirements are measured and/or tested against specified service objectives.
• Performance/stress testing is performed by an independent test team, not by developers.
Performance Testing
Primary Goals of Performance Testing

- Utilize technology to provide repeatable, consistent testing of the System Under Test (SUT)
- Increase quality through greater scope of testing performed
- Conduct testing that is very difficult, no, impossible to do with just humans
- Ensure application infrastructure can handle desired volume
Performance Testing
Functional vs. Performance Test Automation

• Functional testing is designed to insure that the application is working as designed from a screen to screen and control to control standpoint.

• Performance testing is intended to insure that the software AND hardware can handle the number of users or load of data that will make it stable for day to day use.

• Testing these look for 2 different results
  – Functional Defects, buttons or controls don’t work
  – Performance Defects, when 2000 users hit this page, the server crashed
Performance Testing
Performance in the News

• RBS boss blames software upgrade for account problems – 06/25/2011
• Google wallet anniversary falls flat as new software problems emerge – 05/30/2012
• Apple iOS5 Upgrade Causing Problems – 10/13/2011
• Software Problems at PokerStars – 05/24/2011
Justification Survey
Performance Testing

Justification for Testing

- Downtime
- Loss of Revenue
- Loss of Productivity
- Confirm that application infrastructure can handle anticipated volumes
- Confirm that application infrastructure is sized correctly
- Confirm performance to Service Level Agreements
- Identify and resolve bottlenecks before going live
Challenges Survey
Benefits of Performance Testing

Common Challenges

- Lack of clarity on true application usage
- Insufficient test environment
- Insufficient test data
- Extrapolation as a testing effort
- Lack of expertise in test design
- Lack of expertise in test results analysis
Goals Survey
Performance Testing
Test Management Goals

- Ensuring clarity on true application usage
- Collection and validation of performance requirements
- Ensuring test design is based upon true application usage
- Selection and acquisition of test scenarios or test cases
- Enabling identification and preparation of test data
- Validation of test design
- Validation of test results analysis
- Providing meaningful insight on performance testing to the project team
## Performance Testing
### The Vendor Landscape

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Typical Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP</td>
<td>LoadRunner, Performance Center</td>
<td>Many protocols</td>
</tr>
<tr>
<td>MicroFocus – Borland</td>
<td>SilkPerformer, SilkPerformer Cloudburst</td>
<td>Many protocols</td>
</tr>
<tr>
<td>Original Software</td>
<td>TestDrive</td>
<td>GUI, Web, AJAX, JAVA, IBM Green Screen</td>
</tr>
<tr>
<td>SOASTA</td>
<td>CloudTest</td>
<td>Web and Mobile</td>
</tr>
<tr>
<td>IBM Rational</td>
<td>Performance Tester</td>
<td>Web, J2EE, Siebel, Citrix, SIP and SAP</td>
</tr>
<tr>
<td>Itko\Lisa</td>
<td>LISATest, Virtualize</td>
<td>Middleware, Virtual Test Environment</td>
</tr>
<tr>
<td>RadView Software</td>
<td>WebLOAD Professional</td>
<td>Web 2.0 AJAX, JSON data types, SOAP, and XML</td>
</tr>
<tr>
<td>Compuware – dynaTrace, Gomez</td>
<td>Application Performance Management</td>
<td>Many protocols</td>
</tr>
<tr>
<td>Oracle Empirix</td>
<td>Load Testing</td>
<td>Web applications</td>
</tr>
</tbody>
</table>
Components of Performance Testing
## Performance Testing

**Testing Categories and Types**

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human Interface</strong></td>
<td>Usability Testing, Standards Compliance Testing</td>
</tr>
<tr>
<td><strong>Unit</strong></td>
<td>Unit Testing</td>
</tr>
<tr>
<td><strong>Business Driven</strong></td>
<td>Functional Testing, Interface Testing, End to End Testing, Data Integrity Testing</td>
</tr>
<tr>
<td><strong>Readiness</strong></td>
<td>Implementation Testing, Beta Testing, User Acceptance Testing</td>
</tr>
<tr>
<td><strong>Regression</strong></td>
<td>Regression Testing, Sanity/Smoke Testing</td>
</tr>
</tbody>
</table>
Performance Testing
A Typical Corporate Environment

Business to Business

Business Partner
Customer

Firewall

Internet

Web Server

PC Client

Web Server

Intranet

Web Server

Data Server

App. Server

Firewall
Performance Testing
A Variety of Components

Performance testing can be performed on both hardware and software, below are several places where most business will conduct this type of testing

– Software Applications
– Networks
– Databases
– Mainframes
– Firewalls
– Routers
Performance Testing
Test Objectives

**Application response time**
- How long does it take to complete a task?

**Configuration sizing**
- Which configuration provides the best performance level?

**Acceptance**
- Is the system stable enough to go into production

**Regression**
- Does a new version of the software adversely affect response time?

**Reliability**
- How stable is the system under a heavy work load?

**Capacity planning**
- At what point does performance degradation occur?

**Bottleneck identification**
- What is the cause of the performance degrading?

**Product Evaluation**
- What is the best server for 100 users?
Performance Testing
Test Analysis – Common Measurements

• Processor
  – CPU Utilization
• Memory
  – % Committed Bytes
  – Pages / Sec
• Hard Drive
  – Average Disk Queue Length
  – Disk Transfer Rate
• Database
  – Number of Concurrent Connections
  – Number of Record Locks
Performance Testing
Test Tool Components

Most Performance or Load Testing Tools will be broken down into these 3 main pieces. They may be named differently, but they will perform the same action

– Controller
– Load Generator(s)
– Virtual Users
Performance Testing
Test Tool Components
Performance Testing
Test Tool Components

Most communication on this site is HTTP since this is a web application.
Performance Testing
Monitoring Application Components
Performance Testing Roles, Tasks, and Activities
Performance Testing
Roles and Responsibilities

• Automation Architect
• Automation Developer
• Test Manager
• Application SME
• Hardware / Network SME
• Database Administrator
Performance Testing
Requirements Gathering

• Use a Project Questionnaire and Performance Testing Informational Survey to gather information such as:
  – Project scope
  – Workload
  – Test data
  – Monitoring Tools
  – Environment

• Use Techniques such as Joint Requirements Planning (JRP) and Reverse Engineering to gather requirements
Performance Testing
Requirements Gathering

• Operating Systems – Windows, Linux, etc.
• Application Languages - .Net, Java, J2EE, SAP, Oracle, etc.
• Communication Protocols – http, COM/DCOM, MQSeries, etc.
• Browsers – Internet Explorer, Firefox, etc.
• Platform
  – Mainframe
  – Object Oriented
  – Service Oriented Architecture
## Performance Testing

### Requirements Gathering – Another View

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Server</td>
<td>Microsoft IIS, Apache, Netscape</td>
<td>Windows 2003</td>
</tr>
<tr>
<td>Application Server</td>
<td>Broadvision, SilverStream, Microsoft ASP, Allaire ColdFusion, IBM WebSphere, ATG Dynamo, Ariba Buyer, BEA WebLogic</td>
<td>Windows 2003</td>
</tr>
<tr>
<td>Database</td>
<td>Microsoft SQL Server, Oracle</td>
<td>Windows 2003</td>
</tr>
<tr>
<td>Middleware</td>
<td>BEA Tuxedo</td>
<td>Windows 2003</td>
</tr>
<tr>
<td>Firewall</td>
<td>CheckPoint Firewall-1</td>
<td>Windows 2003</td>
</tr>
<tr>
<td>SNMP Devices</td>
<td>SNMP Monitors, Cisco Works</td>
<td>Windows 2003</td>
</tr>
<tr>
<td>Streaming Media</td>
<td>Real Server, Windows Media Server</td>
<td>Windows 2003</td>
</tr>
</tbody>
</table>
Performance Testing
Analyze System Under Test (SUT)

• Map System Architecture Data Flow
• Identify key components within the system architecture
• Verify performance testing tool compatibility with the SUT
• Review System Architecture with Project Team
Performance Testing
Plan and Acquire the Test Environment

Confirm the infrastructure
• Options include production-like test environment or Disaster Recovery site

Confirm the source and volume of test data
• Performance testing can use large amounts of data
  – Login IDs for all virtual users
  – Data for all types of transactions
• Determine how test data will be obtained and used
Performance Testing
Transaction Concurrency

How many transactions will need to be run per minute if a load test has to be run for two hours with 5000 users, assuming an average transaction length of five minutes?

**Determine how many transactions run per minute:**
- \( 120 \text{ min} / 5 \text{ min} = 24 \text{ iterations for each user} \)
- \( 5000 \text{ users} \times 24 \text{ iterations} = 120,000 \text{ transactions} \)
- \( 120,000 \text{ transactions} / 120 \text{ minutes} = 1000 \text{ transactions per minute} \)

**Apply the transactional concurrency to the application:**
- The test is run during the 10 AM-12 NOON time slot
- The test should consist of 5000 users running 24 iterations
- The system must be able to handle 1000 transactions per minute
Performance Testing
Utilize a Transaction Matrix

A Critical Tool to Successful Design
• Identify user profiles to the System Under Test (SUT)
• Identify the transactions within each user profile
• Document Transaction Matrix
Performance Testing
Benefits of a Transaction Matrix

• Tells the probable kinds of users to the system
• Tells which transactions are to be performed by each of the identified users to the system
• Helps in scripting the performance test cases, by easily making out what transaction is performed by each of the identified users to the system.
Concurrency Survey
Performance Testing
Create the Transaction Matrix

- Analyze the SUT
- Review the business requirements documentation
- Identify the different transactions that could be performed in the application during peak load
- Create a Distribution Matrix
- Identify the probable different end-users who you think will be visiting the site
- Identify what transactions will be performed by each of the users you have identified
Performance Testing
Sample Distribution Matrix

<table>
<thead>
<tr>
<th></th>
<th>8:00am</th>
<th>9:00am</th>
<th>10:00am</th>
<th>11:00am</th>
<th>12:00pm</th>
<th>1:00pm</th>
<th>2:00pm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Login</strong></td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>30</td>
<td>50</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td><strong>Search</strong></td>
<td>40</td>
<td>30</td>
<td>10</td>
<td>50</td>
<td>100</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td><strong>Search Top 100</strong></td>
<td>15</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td><strong>Search by Artist</strong></td>
<td>20</td>
<td>20</td>
<td>6</td>
<td>30</td>
<td>50</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td><strong>Search by Title</strong></td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td><strong>Add to Cart</strong></td>
<td>20</td>
<td>10</td>
<td>4</td>
<td>20</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td><strong>Checkout</strong></td>
<td>20</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>
Performance Testing
Sample Transaction Matrix

<table>
<thead>
<tr>
<th></th>
<th>Home Page</th>
<th>Login</th>
<th>Search</th>
<th>Add to Cart</th>
<th>Checkout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casual User</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st Time User</td>
<td>10</td>
<td>10</td>
<td>40</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>1st Time Buyer</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Repeat User</td>
<td>40</td>
<td>40</td>
<td>60</td>
<td>25</td>
<td>15</td>
</tr>
</tbody>
</table>
Performance Testing
Test Scenario Selection

• It is not feasible to load test every transaction \ Business Process
• Identify transactions that are:
  – Data intensive
  – High Volume
  – Mission Critical
• Concentrate on a peak time frame to find peak load
• Create a transaction matrix to identify possible scenarios and user profiles
• Typically 20% of the users will generate 80% of the load on a system so not all user types need to be represented.
Performance Testing
Creation of Performance Test Scenarios

• Analyze and document the performance requirements
• Interview the Business and Development personnel
  – Review performance test objectives
  – Review user profiles
  – Review transactions to be performed by the users
  – Review metrics
  – Identify and document the Scenarios for the test
Performance Testing
Creation of Performance Scenario Document

• Scenario document tells the probable percentage of users identified to the system
• The metrics identified for each of the users will be used while constructing a work load definition before executing the scripts
• Scenario document helps in creating a test suite execution plan enabling the smooth flow in the communication channel before executing the scripts.
Performance Testing
Sample Performance Scenarios

• Load Test
  – Home Page User 20%
  – Product User 20%
  – Search User 10%
  – Repeat User 50%

Ramp of users at an interval of 1 user every 5 seconds until 350 users are logged in for 60 minutes
Performance Testing
Sample Performance Scenarios

• Scalability Test
  – Home Page User     25%
  – Product User       25%
  – Search User        15%
  – Repeat User        55%

Ramp of users at an interval of 1 user every 3 seconds with continuous ramp up for 30 minutes
Performance Testing
Test Execution

• Run the script with one virtual user to verify a baseline for the test
• Run the script with five virtual users to verify a baseline for the test with multiple users
• Verify / schedule test script execution by communicating to the team members
• Run the script as per the scenarios developed
Performance Testing

Test Analysis

• Analyze results in the Performance Test Suite and the SUT
• Generate defect report for the SUT
• Meet with the technical experts to analyze the test results
• Generate Performance Test Suite (PTS) modifications report
• Create Summary report of the results
• Analyze reports to identify trends, problems and opportunities
Performance Testing
Test Analysis

During the execution of a performance test, a failure does not always mean that a defect has been found. The result will need to be analyzed to find the root cause:

– A test tool problem may arise
– A piece of hardware may be down for maintenance during the test or unavailable
– The data source may be unavailable or have incorrect data in it
Performance Testing
Test Analysis – Sample Results Graph

Number of visitors increases

But KB per second and transaction per second falls

Because transactions start to fail
Performance Testing
Test Evaluation

• Verify that all requirements were met
• Identify and evaluate any outages
• If there are any items that need action, decide on the best way of addressing the items. Options include:
  – do not close the project
  – define a follow-on project
  – initiate a maintenance process.
• Consider how the Test Suite will be maintained
Performance Testing Design Options
Performance Testing
Test Design Options

• Testing approach should meet the “client” test objectives. Typically it is a combination of the following types of performance tests:
  – Load Testing
  – Baseline
  – Scalability / Capacity
  – Stress and Hot Spot Testing
  – Spike and Bounce Testing
  – Endurance Testing
  – Integrity Testing
Performance Testing
Test Design Options

- **Load Testing**
  
  *Attempts to model the anticipated real world usage and utilization over a short period of time, with the expected number of users and average user interaction delay times. This is looking for the typical user experience.*

- **Baseline**
  
  *Initial test from which remaining tests are measured, typically measured with one user.*

- **Scalability / Capacity**
  
  *Testing by increasing the workload in an effort to determining stress points and to measure the limits of the capacity of the system.*

- **Stress and Hot Spot Testing**
  
  *Testing over a short period of time when the site is hit with larger than expected loads, requiring extensive computations/data retrieval. Here you are looking for how the system breaks down under stress. A variation is “hot spot testing”, where you focus the stress on a specific portion of the product, looking for a weak link.*
Performance Testing
Test Design Options

- Spike and Bounce Testing
  Testing with a sudden growth in load over a very short period of time, looking to see if the system can respond to abrupt changes in the workload. A variation is to follow the spike with a bounce down to a very low load level, and then continue repeating the up and down pattern. This tests whether the system can recycle its resources properly.

- Endurance Testing
  A load or stress test that is run for an extended period of time, typically several days, with the purpose of detecting slow-to-appear defects. This measures the reliability of the system.

- Integrity Testing
  Combines functional testing with stress testing to ensure that functionality, which worked under low volumes still works.
Design Survey
Performance Testing
Test Scripting Techniques

When creating a Performance Testing script you will use a method called “Record & Playback”.

• While discouraged in Functional Test Automation, it is the recommended method of script creation in Functional Test Automation

• The script will be recorded, then modifications will be made as necessary to add user data from files or databases to simulate different users at test time
Performance Testing Exercise
Performance Testing Review

• Outline the components of a performance test and considerations
• Discuss various roles, tasks, and activities
• Review several design approaches specific to performance testing
Questions and Discussion

Michael Lawler
mikelawler@nuevista.com
www.nuevista.com
630-472-6838, x 104