

Bringing Value to the Organization with Performance Testing

Michael Lawler
NueVista Group

Today's Agenda

- Explore the benefits of a properly performed performance test
- Understand the basic elements of the performance test
- Learn how to design, execute, and measure an effective performance test

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.

Benefits of Performance Testing

Benefits of Performance Testing

Performance in Perspective

- Just 8 hours of downtime in a year equates to 99.9% uptime. At that performance:
 - Two planes at O'Hare airport would have unsafe landings each day
 - 12 newborns would be given to the wrong parents each day
 - 291 pacemaker operations would be performed incorrectly
 - 315 entries in Webster's Dictionary would be misspelled

Benefits of 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

Benefits of Performance Testing

Justification for Testing

- Downtime
- Loss of Revenue
- Loss of Productivity
- What Else???

Benefits of Performance Testing

Justification for Testing

- 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
- What Else???

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
- What Else???

Benefits of Performance Testing

Bringing Value – Test Management Perspective

- Ensuring clarity on true application usage
- Collection and validation of performance requirements
- 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
- What Else??

Benefits of Performance Testing

Bringing Value – Test Engineer Perspective

- Providing meaningful insight on performance testing to the project team
- Enabling effective requirements elicitation
- Ensuring test design is based upon on true application usage
- Creation of a reusable and easily maintained test suite
- Providing meaningful test results analysis
- What Else???

Performance Testing Basics

Performance Testing Basics

Testing Categories and Types

Role	Responsibilities
Human Interface	Usability Testing, Standards Compliance Testing
Unit	Unit Testing
Infrastructure Driven	Environment Configuration Testing, Performance Testing , Recovery Testing, Security Testing, Coexistence Testing
Business Driven	Functional Testing, Interface Testing, End to End Testing, Data Integrity Testing
Readiness	Implementation Testing, Beta Testing, User Acceptance Testing
Regression	Regression Testing, Sanity/Smoke Testing

Performance Test Basics

A Variety of Different 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 Basics

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 Basics

Roles and Responsibilities

- Automation Architect
- Automation Developer
- Test Manager
- Application SME
- Hardware / Network SME
- Database Administrator

Performance Test Basics

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 Test Basics

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 Test Basics

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.

Performance Test Basics

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 Test Basics

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:

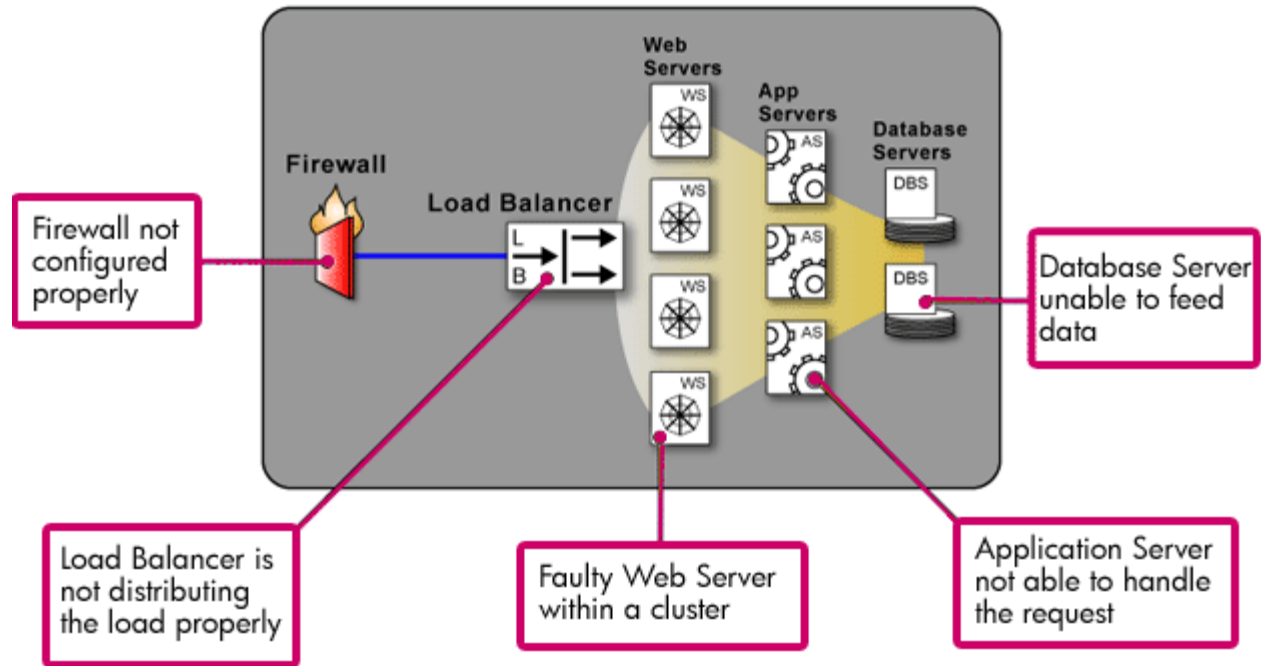
- $120 \text{ min} / 5 \text{ min} = 24$ iterations for each user
- $5000 \text{ users} \times 24 \text{ iterations} = 120,000$ transactions
- $120,000 \text{ transactions} / 120 \text{ minutes} = 1000$ transactions per minute

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 Test Basics

Monitoring Application Components



Design and Execute

Design and Execute

Gather & Document Requirements

- 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

Design and Execute

Gather & Document Requirements

- 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

Design and Execute

Gather & Document Requirements

Resources	Type	OS
Web Server	Microsoft IIS	<input checked="" type="checkbox"/>
	Apache	<input type="checkbox"/>
	Netscape	<input type="checkbox"/>
Application Server	Broadvision	<input checked="" type="checkbox"/>
	SilverStream	<input type="checkbox"/>
	Microsoft ASP	<input type="checkbox"/>
	Allaire ColdFusion	<input type="checkbox"/>
	IBM WebSphere	<input type="checkbox"/>
	ATG Dynamo	<input type="checkbox"/>
	Ariba Buyer	<input type="checkbox"/>
	BEA WebLogic	<input type="checkbox"/>
Database	Microsoft SQL Server	<input checked="" type="checkbox"/>
	Oracle	<input type="checkbox"/>
Middleware	BEA Tuxedo	<input type="checkbox"/>
Firewall	CheckPoint Firewall-1	<input type="checkbox"/>
SNMP Devices	SNMP Monitors	<input checked="" type="checkbox"/>
	Cisco Works	<input type="checkbox"/>
Streaming Media	Real Server	<input checked="" type="checkbox"/>
	Windows Media Server	<input type="checkbox"/>

Design and Execute

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

Design and Execute

Vendor Evaluation and Selection

- Review potential vendors and conduct an informal analysis of their capabilities
- Publish requirements to selected vendors
- Confirm their interest and ability to proceed with a tool evaluation
- Select vendors for the evaluation process
- Complete the evaluation
- Validate selected vendor pricing and implementation costs
- Finalize vendor and proceed to implementation planning

Design and Execute

Vendor Evaluation and Selection

The Vendor Landscape – At This Hour

Vendor	Product	Typical Use
HP	LoadRunner, Performance Center	Many protocols
MicroFocus – Borland, Compuware	SilkTest, QALoad	Many protocols
Original Software	TestDrive	GUI, Web, AJAX, JAVA, IBM Green Screen
IBM Rational	Performance Tester	Web, J2EE, Siebel, Citrix, SIP and SAP
Itko\Lisa	LISATest, Virtualize	Middleware, Virtual Test Environment
RadView Software	WebLOAD Professional	Web 2.0 AJAX, JSON data types, SOAP, and XML
Gomez	Reality Load XF	Many protocols
Oracle Empirix	Load Testing	Web applications

Design and Execute Implementation Planning

- Define implementation requirements, both personnel and technical
- Choose a pilot application
- Identify available resources
- Develop a work plan for implementation which includes a process development effort
- Review the approach with all stakeholders

Design and Execute

Plan and Acquire the Test Environment

Components Include:

- Server configuration
- PCs to install load testing software
- Configuration of Monitors
- Software Licenses
- Test Data

Design and Execute

Plan and Acquire the Test Environment

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

Design and Execute

Construct 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

Design and Execute

Construct a 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

Design and Execute

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.

Design and Execute

Sample Distribution Matrix

	8:00am	9:00am	10:00am	11:00am	12:00pm	1:00pm	2:00pm
Login	20	10	5	30	50	20	10
Search	40	30	10	50	100	40	30
Search Top 100	15	5	2	10	20	15	5
Search by Artist	20	20	6	30	50	20	20
Search by Title	5	5	2	10	30	5	5
Add to Cart	20	10	4	20	30	20	10
Checkout	20	5	4	10	20	20	5

Design and Execute

Sample Transaction Matrix

		Home Page	Login	Search	Add to Cart	Checkout
Casual User		30				
1 st Time User		10	10	40	10	10
	1 st Time Buyer	5	5	20	5	5
Repeat User		40	40	60	25	15

Design and Execute 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.

Design and Execute

Benefits of Performance Test Scenarios

- Identifies the load test objectives
- Identifies the number of scripts that are needed
- Identifies the number of load tests that are needed
- Document Scenarios

Design and Execute

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

Design and Execute

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.

Design and Execute

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

Design and Execute

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

Design and Execute

Sample Performance Test Plan

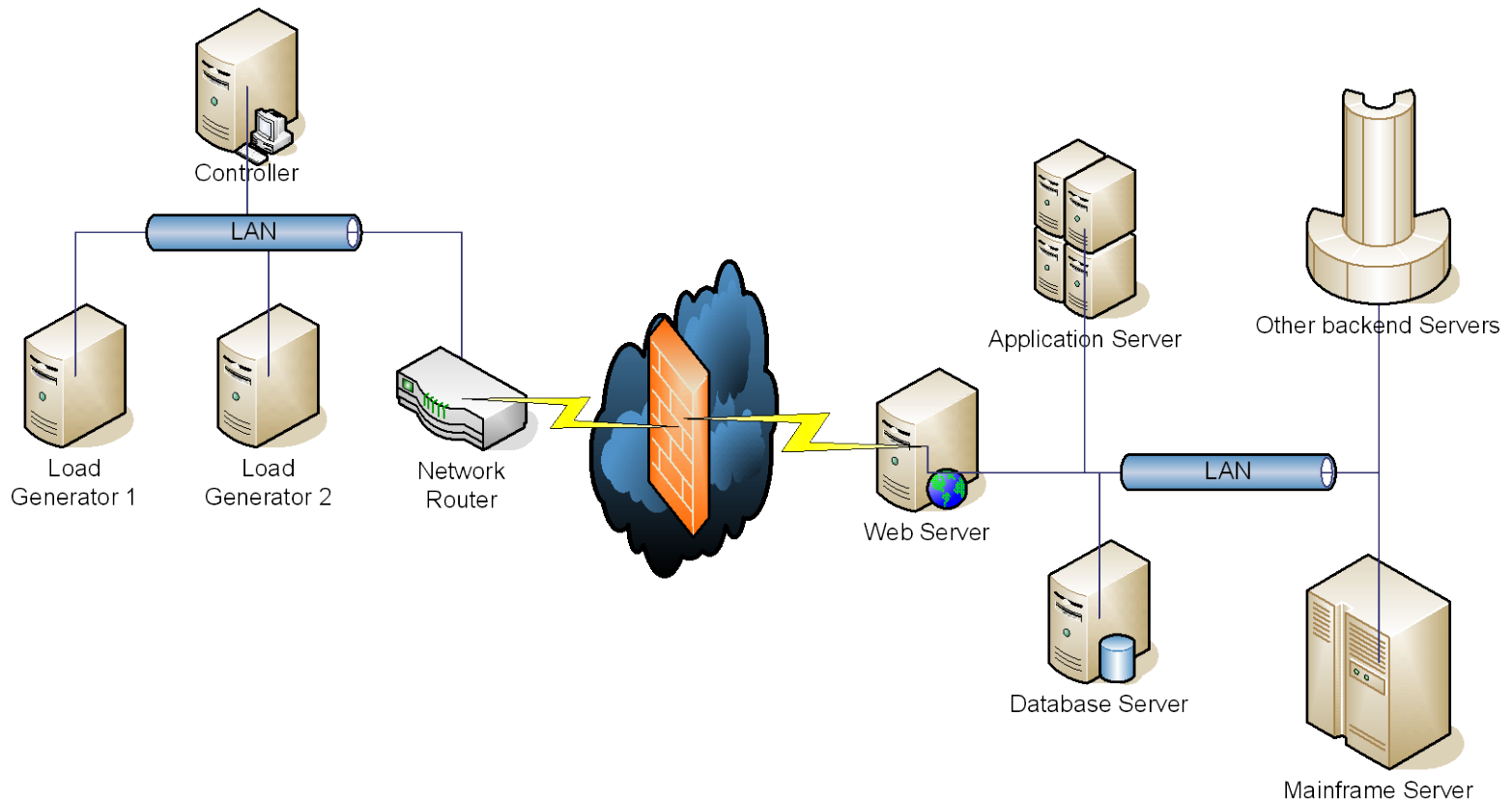
PERFORMANCE TEST SUITE EXECUTION PLAN (April 26, 2010)					Test Resource(s)	Days	StartDate
Task Name							
Performance Test Suite Execution							
	Run Performance Scripts					5 days	
	Determine Tests to Run						
	Validate Test Environment Setup						
	Validate hardware configuration						
	Validate application configuration						
	Validate test data						
	OPTIONAL Make to have appropriate TestID's						
	Notify all participating staff (testers, developers, NetAdmin, DBA)						
	Execute Scripts						
	OPTIONAL Determine Code Coverage of Tests						
	OPTIONAL Create Record of Tests						
	Results Management					3-5 days	
	Analyze Results for Defects in Performance Test Suite and SUT						
	OPTIONAL Determine Test Re-runs/Contingencies						
	OPTIONAL Make necessary script or environment changes						
	OPTIONAL Re-run tests and record results						
	Generate Bug Reports for the SUT						
	Generate PTS Modifications Needed Reports						
	Create Summary Reports of Results						
	Analyze Reports to Identify Trends, Problems, and Opportunities						
	Develop Performance Future Objectives						
	Identify Process Improvements Needed						
	Create Project Lessons Documentation						

Design and Execute Test Tool Components

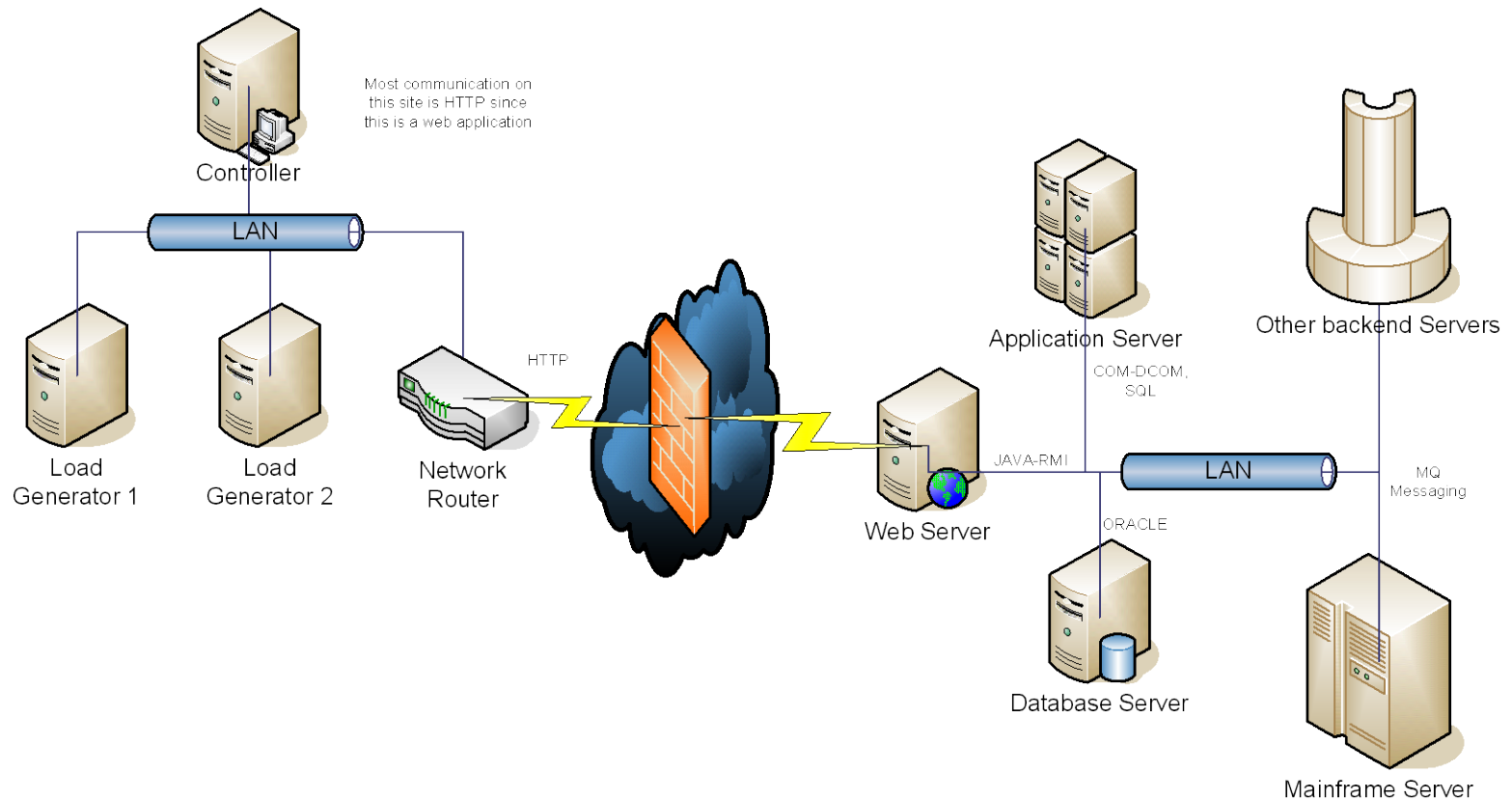
Most Performance or Load Testing Tools will be broken down into these 3 main pieces. They may be named differently, but they will perform the same action

- Controller
- Load Generator(s)
- Virtual Users

Design and Execute Test Tool Components



Design and Execute Test Tool Components



Design and Execute

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

Design and Execute

Test Preparation

- Identify critical test sets to run from the Test Execution plan
- Generate detailed test execution records
- Identify needed changes to the suite as well as to the SUT
- Develop records to be reviewed during post-implementation review for individual as well as team learning

Design and Execute

Test Execution

- Run the script with one virtual user to verify a baseline for the test
- Verify / schedule test script execution by communicating to the team members
- Run the script as per the scenarios developed

Design and Execute

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

Design and Execute

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

Design and Execute

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

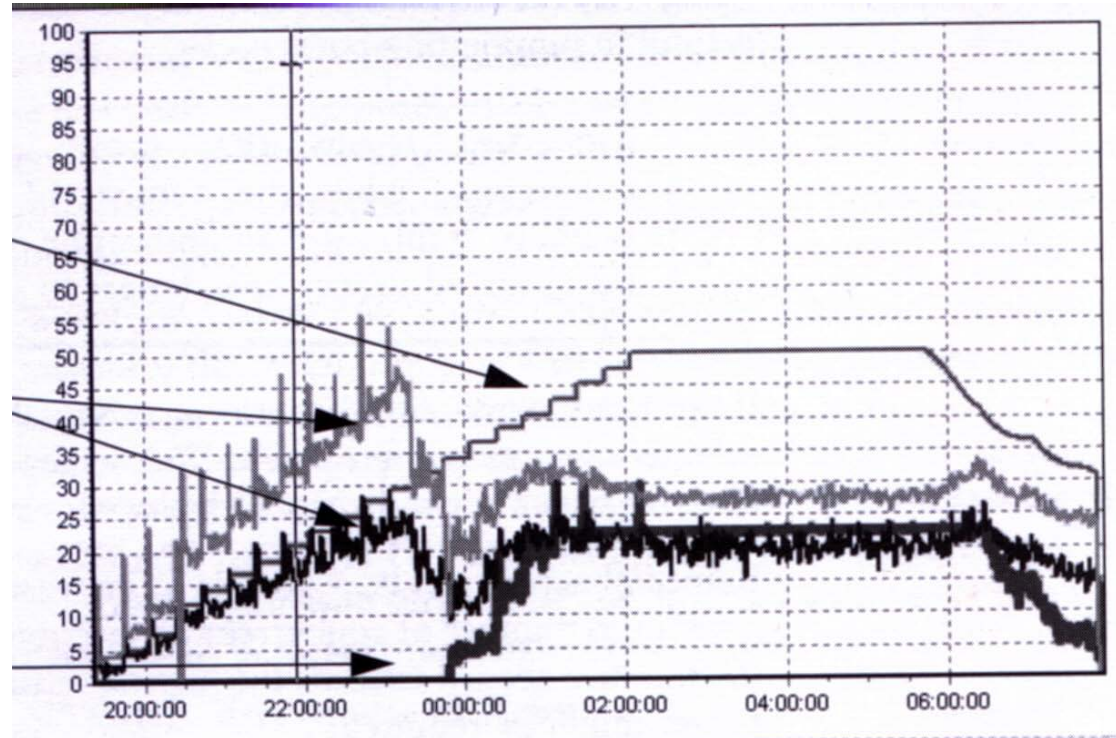
Design and Execute

Test Analysis – Sample Results Graph

Number of visitors increases

But KB per second and transaction per second falls

Because transactions start to fail



Design and Execute

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

You've Succeeded

Now What?

- Ensure that the software license quantities and types meet the needs of the automated testing initiative
- Plan a workforce model for current and future needs
- Provide for ongoing process and technical reviews
- Help the organization understand the value of test automation

You've Succeeded Now What?

- Annual review of the technology plan
 - Will architectures change?
 - Will application usage demand and patterns change?
 - Do user licenses and types of automated testing software match the technology plan?

You've Succeeded Now What?

- Workforce Management
 - Understand the unique skills that are needed to support effective use of the tool
 - Implement the organizational model that works – Center of Excellence or team oriented
 - Utilize techniques to attract and hold a skilled workforce
 - Keep the workforce capable

You've Succeeded Now What?

- Ongoing Process Improvement
 - Create the process for ongoing improvement
 - Incorporate the unique components of Automated testing into process improvement activities
 - Promote the value of process improvement
 - Demonstrate how Automated testing can be popularized through process improvement activities

Bringing Value with Performance Testing Review

- Explore the benefits of a properly performed performance test
- Understand the basic elements of the performance test
- Learn how to design, execute, and measure an effective performance test

Bringing Value with Performance Testing Benefits

- Utilize technology to provide repeatable, consistent testing of the System Under Test (SUT)
- Conduct testing that is very difficult, no, impossible to do with just humans
- Ensure application infrastructure can handle desired volume
- Minimize Downtime and Loss of Revenue

Bringing Value with Performance Testing

Basic Elements

- 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.
- Performance testing is typically performed on multiple infrastructure components
- There are several different design options for performance tests

Bringing Value with Performance Testing

Test Management Perspective

- Ensuring clarity on true application usage
- Collection and validation of performance requirements
- 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

Benefits of Performance Testing

Bringing Value – Test Engineer Perspective

- Providing meaningful insight on performance testing to the project team
- Enabling effective requirements elicitation
- Ensuring test design is based upon on true application usage
- Creation of a reusable and easily maintained test suite
- Providing meaningful test results analysis

Bringing Value with Performance Testing

Design and Execute

- Understand application infrastructure components
- Select compatible vendor software for the infrastructure and test team skills
- Ensure understanding of application usage through use of a transaction matrix
- Provide meaningful results analysis

Questions and Discussion

Michael Lawler

mikelawler@nuevista.com

www.nuevista.com

630-472-6838, x 104

