Performance Testing
Why and How?

Sudheer Parpatakam
Agenda

• Why Performance Testing?
• Performance Testing Workflow
• HP LoadRunner Solutions
Why Load Test Your Application?

- Prevents costly failures of mission-critical applications.
- Assures performance and functionality under real-world conditions.
- Locates potential problems before your customers do.
-Reduces development time.
-Reduces infrastructure costs.
Types of Performance Testing

- **Component Testing**: Find the behavior and performance of each tier.
- **Load Testing**: Determine whether the system handles anticipated real-world load.
- **Stress Testing**: Find system’s breaking point; measure whether system’s environment is properly configured for unexpected, high-transaction volume.
- **Volume Testing**: Check stability of system under extended periods of load.
Examples of Performance Test Objectives

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

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

**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?
Performance Testing Expert Workflow

- Establish Goals
- Gather System Usage Information
- Analyze System Under Test
Defining Goals

Why start with conceptual goals?

Conceptual goals should outline all of your load test objectives.

Examples of conceptual goals:

- A high-priority example is the responsiveness of a “Search” function. Are we able to get search results within a reasonable time?
- A second example is the system administrator’s concern that the “Update” transaction functions during heaving usage.

In the initial stages, jotting down goals that can’t be measured will allow later filtering to create more focused goals.
Quantifying Load Testing Goals

**Examples of Conceptual Goals**
- Search should be fast enough
- Confirm that Update still works under heavy traffic

**Examples of Quantitative Goals**
- Search transaction time of 8 seconds or less during peak hours for 5000 concurrent users
- Attain 200 concurrent users for the Update transaction during peak time - 12 noon
GATHERING INFORMATION

Gathering System Usage Information

Why gather system usage information?

Allows you to:
• Decide which business processes to test.
• Isolate peak loads and peak load times.
• Document user actions and valid input data for each business process.

Where can we get information?

• Using the site, first-hand
• Consulting with administrators
• Consulting with executives
• Researching competitor’s sites
GATHERING DATA

Obtaining Customer Locations

- Use web site's domain report log
- Consult database of customer addresses
- Create estimated model (if application is not yet live)
Analyzing the System Under Test

Why 

- Allows you to setup monitors.
- Allows you to effectively coordinate with backend experts.
- Provides system information that helps isolate performance problems.

Where 

- System Administrators
- Backend experts
- Application experts
- Database Administrators

gather system usage information?
Identifying Business Processes to Test

Mission-Critical Business Process
• Business Processes that are crucial in day-to-day operations.

Heavy Throughput
• Heavy throughput business processes may not be mission-critical but are very popular.

Dynamic Content
• Dynamic content consists of sever requests that are customized for each user.
Business Processes to Test: Mission-Critical Transactions

- Mission-critical

Examples:
E-commerce transactions
Downloading orders

Get info from:
Application Experts
Management Team
Business Processes to Test: Heavy Throughput

- Mission-critical
- Heavy throughput

Examples:
- Home page viewing
- Super Bowl ads
- Accessing a login screen
Business Processes to Test: Dynamic Content

- Mission-critical
- Heavy throughput
- Dynamic content

Examples:
- Dynamic page generation
- Streaming media - hear streaming audio view streaming video

Thank you for your order. Your order number is 81-9341.

Please keep this confirmation number for future reference.

For your reference, we have sent an e-mail confirmation of this order to you at PerfTester@hp.com.

Dynamically generated.
# Business Process Profile

<table>
<thead>
<tr>
<th>Business Process Name</th>
<th>Typical Day</th>
<th>Peak day</th>
<th>Dynamic content</th>
<th>Mission critical</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign in</td>
<td>70/hr</td>
<td>210/hr</td>
<td>Light</td>
<td>High</td>
<td>?</td>
</tr>
<tr>
<td>Create new account</td>
<td>10/hr</td>
<td>15/hr</td>
<td>Moderate</td>
<td>Low</td>
<td>?</td>
</tr>
<tr>
<td>Search for flights</td>
<td>130/hr</td>
<td>180/hr</td>
<td>Moderate</td>
<td>Moderate</td>
<td>?</td>
</tr>
<tr>
<td>View flight booking</td>
<td>20/hr</td>
<td>30/hr</td>
<td>Moderate</td>
<td>High</td>
<td>?</td>
</tr>
<tr>
<td>Purchase ticket</td>
<td>40/hr</td>
<td>90/hr</td>
<td>Heavy</td>
<td>High</td>
<td>?</td>
</tr>
</tbody>
</table>
Documenting User Steps and Input Data
Determining Valid Test Data to Use

**Sign-in Business Process**

<table>
<thead>
<tr>
<th>User Actions</th>
<th>Test Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type user name and password</td>
<td>username: pattyr</td>
</tr>
<tr>
<td></td>
<td>password: test</td>
</tr>
<tr>
<td></td>
<td>username: paulb</td>
</tr>
<tr>
<td></td>
<td>password: rain</td>
</tr>
<tr>
<td></td>
<td>username: ericc</td>
</tr>
<tr>
<td></td>
<td>password: brazil</td>
</tr>
<tr>
<td></td>
<td>username: joee</td>
</tr>
<tr>
<td></td>
<td>password: jojo</td>
</tr>
<tr>
<td></td>
<td>username: franzo</td>
</tr>
<tr>
<td></td>
<td>password: fober</td>
</tr>
</tbody>
</table>

18
Valid Test Data Sources

Valid input data comes from three sources:

Master Data

• Also known as Application Data
• Data is resident in the application’s database
  – Examples: ID numbers and passwords

User-Generated Data

• Originates with the user
  – Examples: new unique ID or email address

External Data

• Data is unknown before the application is run
  – Examples: confirmation and purchase order numbers
Defining Concurrency

Concurrency
A set of users acting upon an application in a similar manner at the same time.

Application Level
• How many users are active on the system?

Business Process Level
• How many users are buying tickets?

Transaction Level
• How many users are buying tickets NOW?
CONCURRENCY

Application Concurrency

TDD (Task Distribution Diagram) for HP Web Tours site:

- **Business Processes**
  - Create new account
  - Search for flights
  - View flight booking
  - Purchase ticket

- **Administrative Processes**
  - Invoice processing
  - System backup

**Target Time**

**Peak time between 10am and 12Noon**

**FLIGHT RESERVATION (typical work day)**
### Business Process Concurrency

<table>
<thead>
<tr>
<th>Business Processes</th>
<th>Number of Users (between 10am and 12 noon)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new account</td>
<td>100</td>
</tr>
<tr>
<td>Search for flights</td>
<td>5000</td>
</tr>
<tr>
<td>View flight book</td>
<td>1600</td>
</tr>
<tr>
<td>Purchase ticket</td>
<td>500</td>
</tr>
</tbody>
</table>

User concurrency changes based on the time of day.
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 min / 5 min = 24 iterations for each user
- 5000 users X 24 iterations = 120,000 transactions
- 120,000 transactions / 120 minutes = 1000 transactions per minute

Apply the transactional concurrency to the application:
- The test is run during the 10-12 AM time slot
- The test should consist of 5000 users running 24 iterations
- The system must be able to handle 1000 transactions per minute
## Understanding the System Components

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Server</td>
<td>Microsoft IIS</td>
<td>Windows 2003</td>
</tr>
<tr>
<td></td>
<td>Apache</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Netscape</td>
<td></td>
</tr>
<tr>
<td>Application Server</td>
<td>Broadvision</td>
<td>Windows 2003</td>
</tr>
<tr>
<td></td>
<td>SilverStream</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Microsoft ASP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allaire ColdFusion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBM WebSphere</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATG Dynamo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ariba Buyer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BEA WebLogic</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>Microsoft SQL Server</td>
<td>Windows 2003</td>
</tr>
<tr>
<td></td>
<td>Oracle</td>
<td></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</td>
<td>Windows 2003</td>
</tr>
<tr>
<td></td>
<td>Cisco Works</td>
<td></td>
</tr>
<tr>
<td>Streaming Media</td>
<td>Real Server</td>
<td>Windows 2003</td>
</tr>
<tr>
<td></td>
<td>Windows Media Server</td>
<td></td>
</tr>
</tbody>
</table>
Mapping Business Processes to Infrastructure Components

<table>
<thead>
<tr>
<th>Business Process</th>
<th>Web Servers</th>
<th>App Servers</th>
<th>Database Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Home Page</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Monitoring Application Components

- Firewall not configured properly
- Load Balancer is not distributing the load properly
- Faulty Web Server within a cluster
- Application Server not able to handle the request
- Database Server unable to feed data
About the Test Environment

A test environment:

- Allows data to be written, read, and destroyed without affecting production users.
- Allows test system to be rebooted during test runs without affecting production users.

The test environment:

- Should mirror the production system.
- Needs business processes that are functioning correctly.
- Should include Benchmark runs.
- Must contain sufficient hardware to generate the load test.
Manual Testing is Problematic
The LoadRunner Solution
The Controller is an administrative center for creating, maintaining, and executing **Scenarios**. The Controller assigns Vusers and load generators to Scenarios, starts and stops load tests, and performs other administrative tasks.

Load generators (also known as hosts) are used to run the Vusers that generate load on the application under test.

LR Analysis uses the load test results to create **graphs** and **reports** that are used to correlate system information, identify bottlenecks, and performance issues.
VuGen

Virtual User Generator (VuGen) – records Vuser scripts that emulate the steps of real users using the application.
Supported Protocols

**PLATFORMS**
- 2000, 2003, XP

**HTTP WEB**
- CORBA
- WAP, COM
- RMI, LDAP
- Winsock
- Streaming
- FTP, IIOP
- POP3, iMode
- RealPlayer
- MS Media
- Voice XML
- SMTP, AJAX
- RDP, .NET
- Web Services

**MIDDLEWARE**
- CORBA
- COM
- EJB
- TUXEDO
- Jolt
- JDBC
- MQSeries

**ERP/CRM**
- Oracle apps., SAP
- Siebel
- PeopleSoft
- Clarity
- Baan

**LEGACY**
- 3270, 5250, VT100
Supported Performance Monitors

- **Sys. Resource Monitors**
  - Windows, UNIX, SNMP, Antara Flame Thrower, SiteScope

- **Network Monitors**
  - Network Delay

- **Firewall Monitors**
  - Check Point

- **Web Server Monitors**
  - Apache, IPlanet, MS IIS

- **Database Monitors**
  - DB2, SQL, Oracle, Sybase

- **App. Server Monitors**
  - Ariba, BroadVision, ColdFusion, WebLogic, WebSphere, Oracle9iAS, SilverStream, IPlanet (NAS), Fujitsu INTERSTAGE, MS Active Server Pages

- **ERP / CRM Server Monitors**
  - SAP Portal, SAPGUI, Siebel Web, PeopleSoft (Tuxedo)

- **App. Component Monitors**
  - COM+
Benchmarking Run

To validate that there is enough test hardware available in the test environment, benchmark the business processes against the testing hardware.

Take a business process and execute a small number of users against the application.

- Validates the functionality of the business process
- Potentially exposes unknown data dependencies

Evaluate the testing infrastructure against the footprint.

- Do I have enough hardware to generate the user load?
- Do I have enough memory?
- Do I have enough CPUs?
Summary

- Performance Testing Objectives
- Define Goals, Gather Data & Analyze System
- HP LoadRunner Solution
Questions?