Open Source and Commercial Performance Testing Tools

Palla Vinod Kumar
Accenture Delivery Center for Technology in India
Introduction

and

Setting the Context
Agenda

• Performance Testing
• Performance Test Tools
• Comparison Factors
• Accuracy
• Cost
• Features
• Test Tools Comparison
• Q & A
Performance Testing

• What?
  – Performance Testing is subjecting the SUT to either user load or data volume load and measuring the performance metrics associated with the application and the system

• Why?
  – To validate that the application can process targeted average and peak transaction volumes
  – To verify that response times and system resource utilization fall within defined target ranges
  – To validate that service level agreements (SLAs) are met in a production-like configuration.
Performance Testing Tools

• Need for Automated tools
• Types of tools
  – Load Generation Tools
    • Controller
    • Script or test generators
    • Load generators or agents
  – Monitoring tools
  – Analysis tools
  – Reporting tools
  – Profilers
Performance Testing Tools

• Commercial tools
  – LoadRunner
  – Silk Performer
  – Rational Performance Tester
  – ...

• Open source tools
  – OpenSTA
  – JMeter
  – Grinder
  – WebLoad Basic Edition
  – ...
Comparison Factors

• Accuracy
• Cost
• Features
  – Scenarios
  – Scripting
  – Monitoring
  – Analysis
Tools - Accuracy

• Accuracy has been a subject of debate and research among performance testers
• Every tool claims that it is accurate
• Each tool can provide different results for a similar test
• It should not be taken for granted that commercial tools are always accurate
• Always cross check the test results with manual testing results while the test is in progress, to validate the test results.
• Same applies for error messages.
Tools - Cost

• Commercial tools includes cost for
  – Controllers
  – Monitors
  – Tuning modules
  – Virtual users

• Open Source tools are free for commercial use also
Tools – Features - Scenarios

• How accurate can the test scenario be built to match real time usage of the application
  – User concurrency
• Goal-based scenarios
  – Transactions per second
  – Response time thresholds
• Control over a running scenario
  – Dynamically adding/stopping virtual users
    • OpenSTA, an open source tool, allows adding virtual users at runtime
  – Ability to view what a virtual user is experiencing in runtime
    • LoadRunner and Silk Performer
Tools – Features - Scenarios

• LoadRunner and Silk Performer can emulate complex scenarios
• Open source tools cannot handle complex scenarios out-of-the-box
Tools – Features - Scripting

• Record and Replay
  – Supported by most commercial and open source tools

• Manual scripting
  – Supported by most commercial and open source tools
  – Very complex and cumbersome, and is not recommended
  – Very useful for testing APIs
Tools – Features - Scripting

• LoadRunner
  – C API
  – Java API
  – JavaScript
  – VB Script
• Silk Performer
  – .Net framework
  – Java framework
  – VB framework
  – COM interface
  – DLL interface
• Grinder
  – Java API
Tools – Features - Scripting

• Commercial tools support many protocols apart from the HTTP/HTTPS protocol
• LoadRunner and Silk Performer support multi protocol scripts
• Most open source tools support only HTTP protocol and therefore can only be used for web applications
• JMeter supports Web – HTTP/HTTPS, SOAP, database via JDBC, LDAP, JMS, Mail - POP3 protocols
Tools – Features - Scripting

• LoadRunner and Silk Performer support high level context based scripting for HTTP/HTTPS protocol
• Context based recording results in a script which is short, clear and easy to maintain
• Open source tools are mainly based on low level scripting
• If the name of an image is changed, then a script recorded earlier will fail for that request if recorded in low level
Tools – Features - Scripting

• Automatic Correlation
  – Most tools (open source and commercial) support automatic correlation of user session variables
  – LoadRunner and Rational Performance Tester support automatic correlation of dynamic application data
  – This feature is very important for scripting complex applications such as enterprise class applications
  – LoadRunner allows defining new correlation rules
Tools – Features - Scripting

• Rendezvous Points
  – Most open source and commercial tools support rendezvous points
  – Commercial tools such as LoadRunner and Silk Performer allow the rendezvous points to be defined in detail which allows complex scenarios to be emulated
Tools – Features - Monitoring

• Most commercial tools support real time monitoring of server side metrics such as CPU usage, memory, …

• Of the open source tools, OpenSTA supports real time monitoring of server resources available through perfmon and SNMP agents

• Most of the open source and commercial tools support the monitoring of client side metrics such as response time, hit rate
Tools – Features - Monitoring

• Performance monitoring of database servers, web servers, application servers are available only in commercial tools and require a separate license

• This should not be a major concern as most applications/technology provide a means for monitoring their performance
  – Performance counters
  – JMX counters
  – SNMP agents
  – Application specific monitoring tools
Tools – Features - Analysis

• Primary use of analysis is to visualize the performance data so as to make observations and create reports

• Most tools provide the following data
  – Running virtual users
  – Response time
  – Hits per second
  – Throughput (bytes/sec)
  – Page time Breakdown
  – Transactions per second
Tools – Features - Analysis

• Most commercial tools allow resource usage graphs overlay on client side metrics graphs
• Among the open source tools, OpenSTA and WebLoad Basic edition has analysis and reporting capabilities on par with commercial tools
• Open source reporting tools such as JCharts can be integrated with JMeter to produce desired graphs
• Both commercial and open source tools allow raw data to be exported to Microsoft Excel so that desired charts can be produced
• Commercial tools offer the advantage to use RDBMS/Access as results repository
## Test Tools Comparison

**Protocols** - The communication protocols that can be captured, manipulated and replayed by the tool.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many supported. Protocols are charged per item. Has a multi-protocol recording feature.</td>
<td>HTTP 1.0 / 1.1 / HTTPS (SSL) only.</td>
<td>Many supported. Protocols are charged per item. Has a multi-protocol recording feature.</td>
<td>HTTP 1.0 / 1.1 / HTTPS (SSL), FTP, JMS, JDBC</td>
<td>HTTP 1.0 / 1.1 / HTTPS (SSL), Can be extended to support Java API testing</td>
</tr>
</tbody>
</table>
## Test Tools Comparison

Playback functions - Replaying of the script and script debugging facilities.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended logging supports view of parameter values and Server messages. Also view and comparison with ‘recorded’ version of web page view and client response messages. Debugging facilities in script generator, step and breakpoints.</td>
<td>Similar playback facilities, but no integrated comparison function. The debugging functions are in the controller, including set break points and single stepping.</td>
<td>Extended logging supports view of parameter values and Server messages. Also view and comparison with ‘recorded’ version of web page view and client response messages. Debugging facilities in script generator, step and breakpoints.</td>
<td>No integrated comparison functionality</td>
<td>No integrated comparison functionality</td>
</tr>
</tbody>
</table>
Scripting Language - The medium used to represent the captured protocol data and manipulate the data for play-back.

<table>
<thead>
<tr>
<th><strong>LoadRunner</strong></th>
<th><strong>OpenSTA</strong></th>
<th><strong>Silk Performer</strong></th>
<th><strong>Jmeter</strong></th>
<th><strong>Grinder</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Called TSL, it uses standard syntax for “C” and allows C function libraries to be added. Has extensive customized functions for the different protocols supported by the tool.</td>
<td>Called SCL, it uses a proprietary, “BASIC” like language that has special automation scripting facilities. Is limited in available functions, such as string manipulation and supports direct DOM addressing.</td>
<td>Called BDL. Has extensive customized functions for the different protocols supported by the tool.</td>
<td>Java</td>
<td>Jython, a mix of python and Java</td>
</tr>
</tbody>
</table>
## Extensibility - The medium used to represent the captured protocol data and manipulate the data for play-back.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional TSL or “C” function libraries, limited to functional capabilities of the tool.</td>
<td>SCL script modules can be defined in 'Include' files. Open Source therefore new tool functionality can be added using C++.</td>
<td>Additional BDL or “C” function libraries, limited to functional capabilities of the tool.</td>
<td>New functions can be made available by creating them in Java</td>
<td>New functionality can be made available through java classes</td>
</tr>
</tbody>
</table>
## Test Tools Comparison

Scripting Interface - The interfaces supplied by the tool application for the purpose of script editing.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Captures in several modes, high level context based and low level HTTP view. Has both a graphical tree structure and a script view. Script view has function sensitive entry.</td>
<td>Has low-level HTTP protocol view and provides graphical tree representation of the Document Object Model (DOM) structure. Viewable captured HTML rendering and addressable server-header table. Some language sensitive, syntax color coding functionality.</td>
<td>Captures in several modes, high level context based and low level HTTP view. Has both a graphical tree structure and a script view. Script view has function sensitive entry.</td>
<td>Has low-level HTTP protocol view and provides graphical tree representation of the requests</td>
<td>Does not support record and playback. HTTP plug-in is available to create HTTP tests.</td>
</tr>
</tbody>
</table>
## Test Tools Comparison

Correlation - The task of substituting values in dynamic data to enable successful playback.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
</table>
Cookie Management - Detection, recording and playback of HTTP cookies. Both tools need additional code to manage JavaScript generated cookies.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP header cookies are managed automatically and can be manipulated manually if required.</td>
<td>HTTP header cookies are managed automatically and can be manipulated manually if required.</td>
<td>HTTP header cookies are managed automatically and can be manipulated manually if required.</td>
<td>HTTP header cookies are managed automatically and can be manipulated manually if required.</td>
<td>Cookies should be handled in the test script</td>
</tr>
</tbody>
</table>
Test Tools Comparison

Parameterization - Automatically changing dynamic data values for more accurate emulation of real users.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensive facilities for data entry including wizard interface to DB interrogation. No standard function to lock data sources and maintain uniqueness of concurrently accessed data across distributed tests.</td>
<td>Extensive facilities for data entry including wizard interface to automatically generate test data. Standard functions for sequential, random and pseudo-random data-file access.</td>
<td>Extensive facilities for data entry including wizard interface to automatically generate test data. Standard functions for sequential, random and pseudo-random data-file access. Has standard common locking facilities for maintaining uniqueness of parameters for an individual load injector or across all injectors on a distributed test.</td>
<td>Standard functions for sequential, random and pseudo-random data-file access.</td>
<td>Parameterization has to be handled in the script</td>
</tr>
</tbody>
</table>
## Test Tools Comparison

Monitoring - Resource usage information is captured during execution. It can be shown during execution and used to build performance reports.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many supported. On-line graphics during execution. Supplied with Apache, Netscape et IIS, other monitors are charged per item. Results used for reporting. New facility to allow remote users to monitor real-time results via a browser based interface. Note: To monitor through a firewall requires TCPIP access through a specific port.</td>
<td>Supports Integrated real-time, graph view of Windows NT/2000 Performance (Perfmon) et SNMP collectors. Various measurements of test progress including VU-specific, custom status and activity information. Web-relay allows monitors to run on remote machines beyond firewall. On-line graphics during execution, et monitored results used for reporting.</td>
<td>Many supported. On-line graphics during execution. Results used for reporting.</td>
<td>No inbuilt support for online monitoring. Allows the functionality to be extended for real time monitoring of client side metrics</td>
<td>Can monitor the response time and transactions per second in real time</td>
</tr>
</tbody>
</table>
Test Tools Comparison

Distributed Load Tests - The ability to distribute the generation of load across multiple load-injector machines.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports multiple load-injectors managed by a single controller.</td>
<td>Supports multiple load-injectors managed by a single controller. Uses TCP/IP if on the same network or the Web-relay feature uses HTTP to control load-injectors located within remote DMZs.</td>
<td>Supports multiple load-injectors managed by a single controller.</td>
<td>Supports multiple load-injectors managed by a single controller.</td>
<td>Supports multiple load-injectors managed by a single controller.</td>
</tr>
</tbody>
</table>
Test Tools Comparison

IP Spoofing - The ability to emulate the behavior of different IP addresses accessing a system. Particularly useful with Load balancing systems.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports IP spoofing including automated router updates for IP forwarding.</td>
<td>No inbuilt features to enable IP spoofing.</td>
<td>Supports IP spoofing</td>
<td>No inbuilt features to enable IP spoofing.</td>
<td>No inbuilt features to enable IP spoofing.</td>
</tr>
</tbody>
</table>
**Test Tools Comparison**

WAN/LAN Emulation - The ability to emulate the behavior of different network infrastructures during a test.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>New feature added to version 7.6. Allows the emulation of latency, packet loss, link faults et dynamic routing effects over the LAN used in a test. Requires a special license.</td>
<td>No inbuilt features to enable WAN/LAN emulation.</td>
<td>No inbuilt features to enable WAN/LAN emulation.</td>
<td>No inbuilt features to enable WAN/LAN emulation.</td>
<td>No inbuilt features to enable WAN/LAN emulation.</td>
</tr>
</tbody>
</table>
Test Tools Comparison

Caching - The ability to emulate the caching of pages as performed by a web browser.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can control browser cache emulation during playback and control setting for each individual vuser.</td>
<td>No specific faculties, although can be emulated with simple script code.</td>
<td>Can control browser cache emulation during playback and control setting for each individual vuser.</td>
<td>No specific faculties, although can be emulated with script code.</td>
<td>No specific faculties, although can be emulated with script code.</td>
</tr>
</tbody>
</table>
## Test Tools Comparison

User connection speed emulation - The ability to emulate the different network speeds that can be used by real users.

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can emulate different network speeds during playback</td>
<td>No inbuilt features to emulate user-connection speed emulation.</td>
<td>Can emulate different network speeds during playback</td>
<td>No inbuilt features to emulate user-connection speed emulation.</td>
<td>No inbuilt features to emulate user-connection speed emulation.</td>
</tr>
</tbody>
</table>
### Test Tools Comparison

Reporting and Analysis - The facilities to examine and investigate the results of a test including timers and monitored resources and display the results in graphical format

<table>
<thead>
<tr>
<th>LoadRunner</th>
<th>OpenSTA</th>
<th>Silk Performer</th>
<th>Jmeter</th>
<th>Grinder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sophisticated et large range of charts and graphs with overlay facilities. -Automatic report generation into MS-Word. Analyzer is a separate application that can be distributed to users.</td>
<td>Simple charts and graphs sufficient for analyzing key load related statistics and resource usage monitors. Resource usage monitors supports graph overlays. Can be exported to Microsoft Excel. No license restrictions on OpenSTA distribution thus stats can be viewed by any user with access. -Free tools and excel macros available through public user-forums.</td>
<td>Sophisticated et large range of charts and graphs with overlay facilities. -Automatic report generation HTML format.</td>
<td>Simple charts and graphs for analyzing client side metrics. Can be extended to use open source reporting tools.</td>
<td>Test results are provided in log files.</td>
</tr>
</tbody>
</table>
Open Source Vs Commercial Tools

• Do not write off open source tools
• Evaluate the requirements of an application before deciding on a test tool
References

• OpenSTA, the free performance testing tool, versus the big-guns... By Andy O'Brien and Antony Marcano of "etest associates":http://www.etest-associates.com

• Stress, Load, Volume, Performance, Benchmark And Base Line Testing Tool Evaluation And Comparison by Cordell Vail

• http://www.softwareqatest.com/qatweb1.html
• http://www.docstoc.com/docs/13159/Website-Load-Tester-Tools-Overview
• http://www.opensourcetesting.org/performance.php