

Automating Performance Testing with JMeter

Luba Ilieva, Senior QA Engineer
Musala Soft Ltd.

15.February.2007



Agenda



- General Truths about Performance
- Performance Test Design Approaches
- JMeter Architecture
- Practical Solution with JMeter
- Demo

General Truths about Performance



- Doubling an attribute, like CPU speed, doesn't necessarily increase performance 2X
- Increasing a resource that is not bottleneck will not necessarily increase performance
- With respect to cache, size matters

Test Design Approaches



- A brand new application
 - Table of comparison between environments
- Application is on production
 - Comparison analysis between versions

The Test Tool – Apache JMeter



- Apache JMeter is

100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions

- Java based

- Highly extensible

- URL: <http://jakarta.apache.org/jmeter/>

JMeter Architecture



Test Plan:

- Thread Groups
- Controllers
- Samplers
- Listeners
- Config elements
- Assertions
- Timers
- Other Elements

JMeter Outlook



The screenshot displays the Apache JMeter GUI for a test plan named "Stress Test". The interface is divided into several sections:

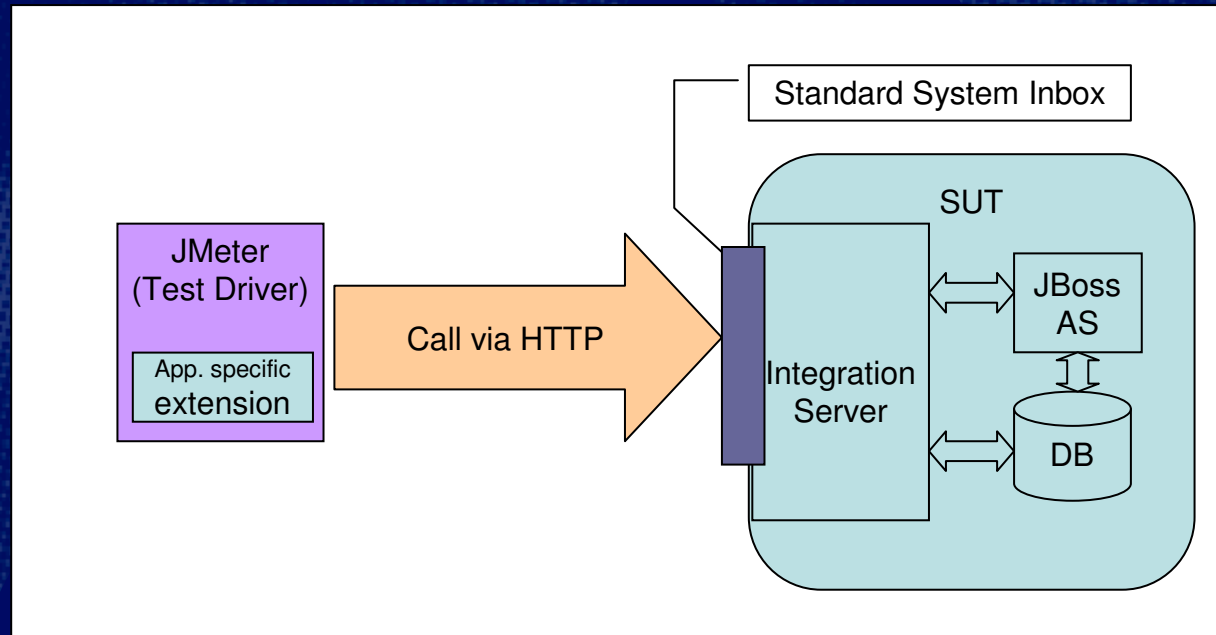
- Test Plan:** Shows the name "Stress Test" and a comment "Example Performance test".
- User Defined Variables:** A table listing variables and their values.
- Options:** Includes checkboxes for "Run each Thread Group separately" and "Functional Test Mode".
- Classpath:** A section for adding directories or jars to the classpath.

Name:	Value
ConcurrentThreadsNumber	50
MsgCountPerMinute	20
HTTPResponseOutputFolder	C:\temp\Results\
ServerName	musalashare5.emea.hpqcorp.net
ServerPort	5000
ServicePath	/invoke
TestDataPath	D:\Automated Testing\JMeter\TestData

Annotations on the screenshot:

- Selected Element:** Points to the "Stress Test" element in the tree view.
- Active Element:** Points to the "Thread Group" element in the tree view.
- Deactivated Element:** Points to the "Constant Throughput Timer" element in the tree view.
- Run Test Indicator:** Points to the green square icon in the top right corner of the test plan area.
- Properties Area:** Points to the "User Defined Variables" table.

Practical Case - Background Info

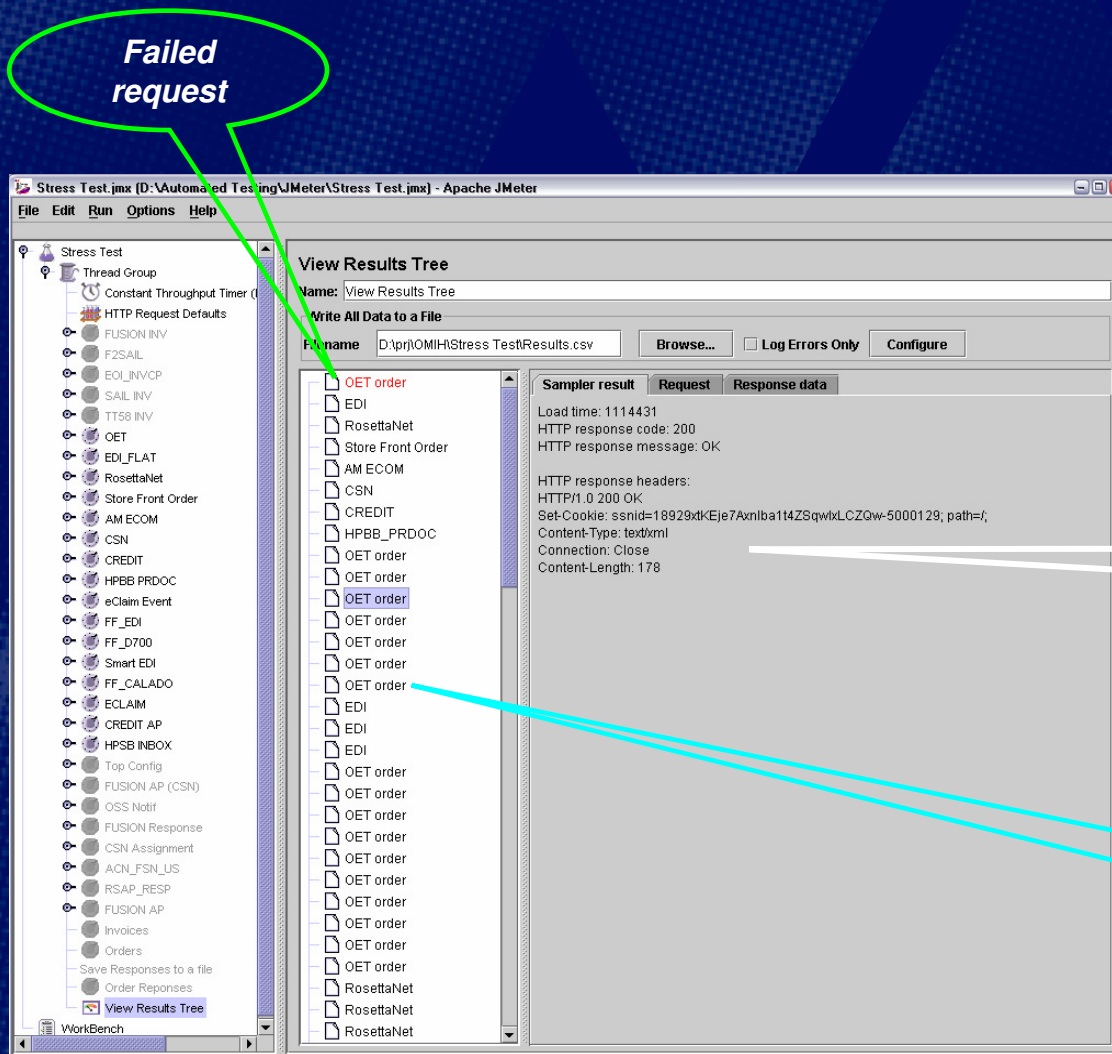


- **Technology:** webMethods IS, Java, Oracle 9i
- **Goals:** Identify bottlenecks in webMethods components
- **Approach:** Simulated production peak hours load

Analyzing Results – JMeter Metrics

JMeter Tree View

- Node for every request
- Properties of the request



The screenshot shows the Apache JMeter interface with the 'View Results Tree' window open. The tree view on the left lists various requests, including 'OET order' (highlighted in red) and 'EDI'. A callout bubble labeled 'Failed request' points to the 'OET order' node. The 'Request Properties Area' on the right displays details for the selected 'OET order' request, such as 'Load time: 1114431', 'HTTP response code: 200', and 'HTTP response message: OK'. A callout bubble labeled 'Successful Request' points to the 'EDI' node in the tree view.

*Request
Properties Area*

*Successful
Request*

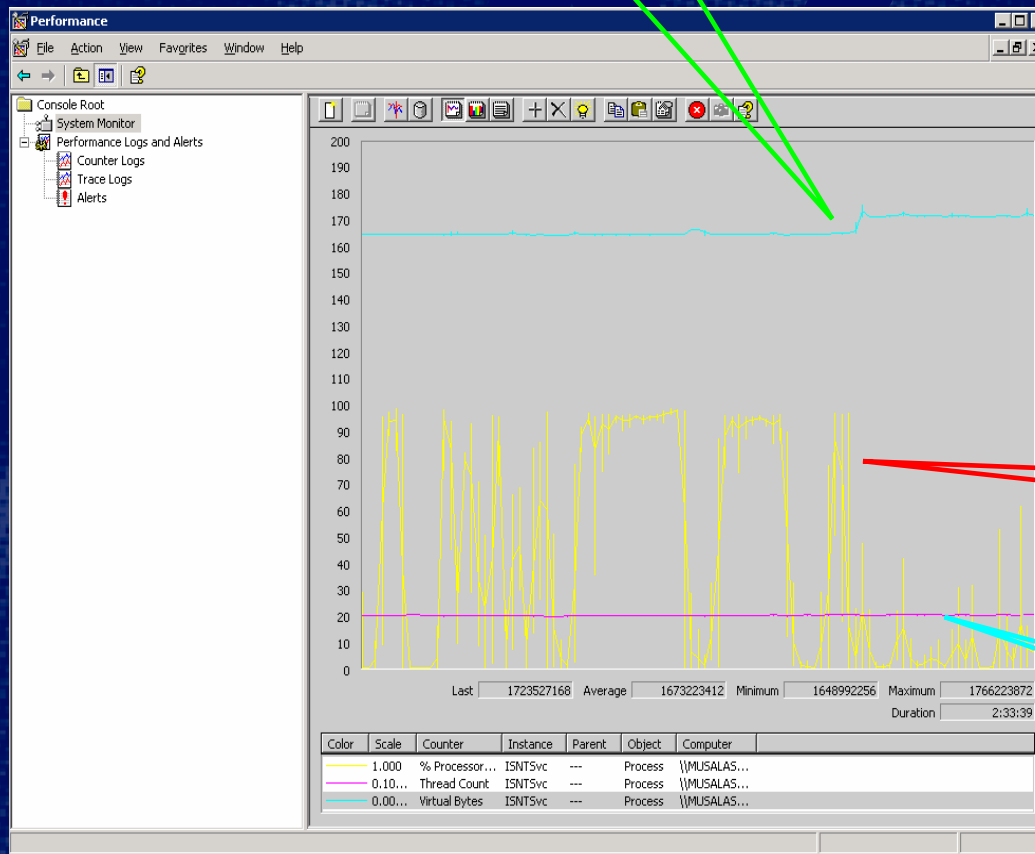
Analyzing Results – Server Utilization



Windows Performance Monitor Utility

Perfmon.exe

- CPU utilization
- VM utilization
- Thread Count



Analyzing Results – Manual Correction



- Runtime
 - Monitor PerfMon GUI and JMeter Tree View Results
- Recorded data
 - JMeter CSV file (timestamp, request time, request name, response status)
 - Excel Spreadsheet



JMeter Demonstration



Thank You!

e-mail:
luba.ilieva@musala.com

presentation:
http://www.musala.com/ppt/MusalaSoft_Perf_Testing_JMeter.pdf