

Java Web App Performance Testing and Tuning

By Charlie Arehart
CTO/Founder **SystemManage**
Our Practice Makes You Perfect™



Session Overview

- Introduction to Load Testing Web Applications
- Creating a Real-World Load Test
- Identifying Common Performance Bottlenecks and Tuning Opportunities
- J2EE Performance Tuning Hints and Tips
- Load Testing Best Practices
- Questions & Answers

2

Audience

- Who should be interested in Load Testing?
 - Do you know you *need* to test?
 - Do you recognize you *ought* to test?
 - Do you think you need to *bother* testing?
 - Have you never even considered testing?
- All web application developers should be
 - You may have reasons to have dismissed it
 - Many common challenges have been overcome

3

Why Should I Care?

- Even if your application doesn't expect high load
 - How it will perform under light load?
 - How will some programming or configuration change effect performance under *current* load?
 - How many users can be sustained, if load does increase?

4

Just the Facts

- Most developers are familiar with the "concept" of load testing, but very few actually do it!
- Most common reasons for not testing are time, complexity & cost (can be very expensive)
- Sites can run into performance problems with only a few users
- Performance problems are usually discovered too late to address the issues effectively

5

What does it mean for your site to perform or scale well?

- Delivers page views to users in 3-8 seconds or less (varying industry standards)
- Performs consistently throughout a visitor's session
- Response times scale linearly as user load increases, as opposed to exponentially

6

What is a Load Test?

- Simulates multiple users to measure, define, validate and maintain optimal application performance, scalability and reliability
- Serves as a compass for understanding the limits of any Web-enabled application and, subsequently, for managing its performance and growth

7

Virtual User Simulation

- You don't need to gather users together to do a test!
- Load testing tools can create "virtual users"



8

Load Testing over the LAN vs. the Internet

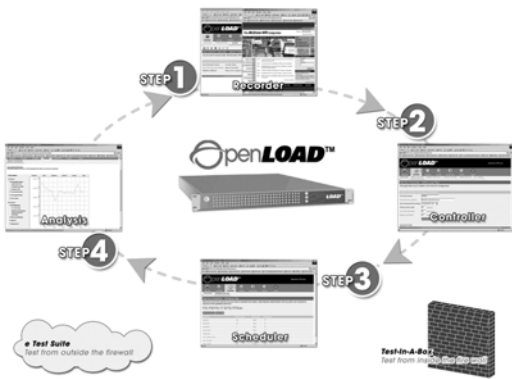
LAN Testing:

- Eliminates network variability due to the Internet
- Saturates system rather than the network

Internet Testing:

- Uncovers hidden delays due to Internet "weather"
- Exercises your pipe, routers, firewalls and load balancers in addition to the system under test
- Ensures Service Providers are meeting SLAs

10



Creating a Real-World Load Test

Customer Case Study:

- Set up a User Scenario
- Configure a Load Test Controller
- Schedule a Load Test Run
- Analyze Test Results

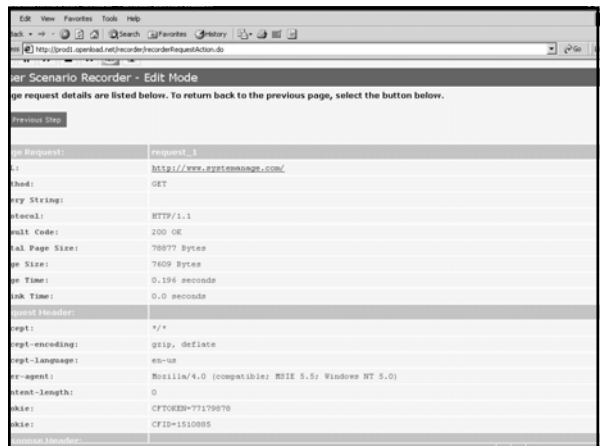
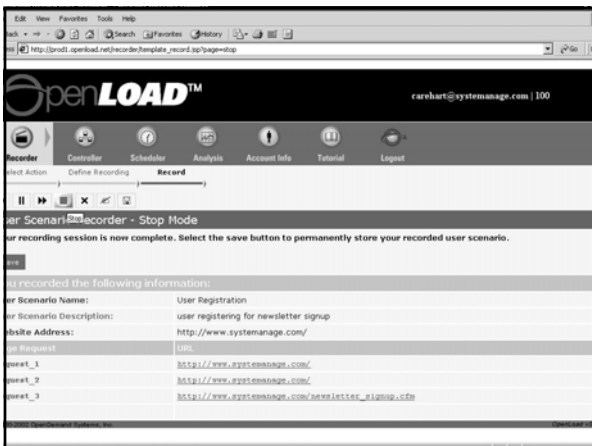
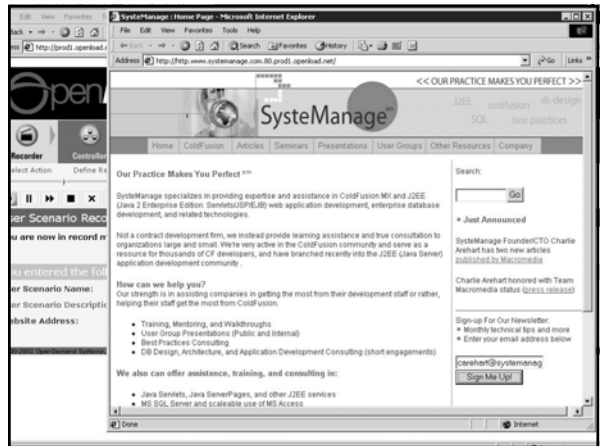
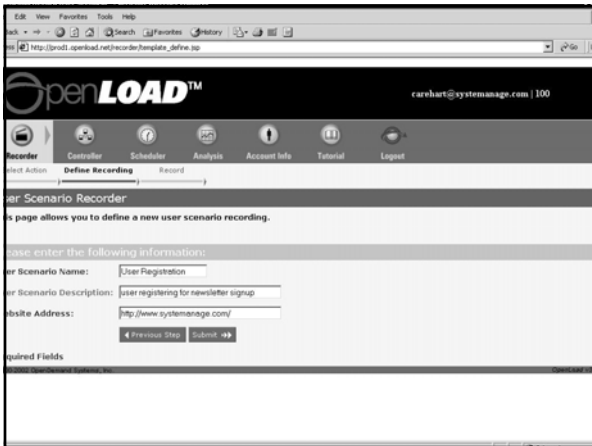
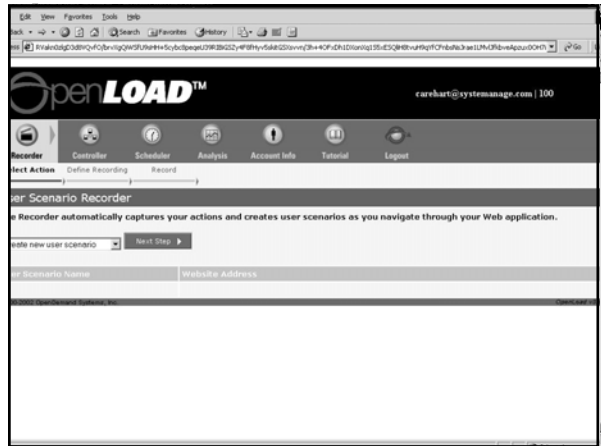
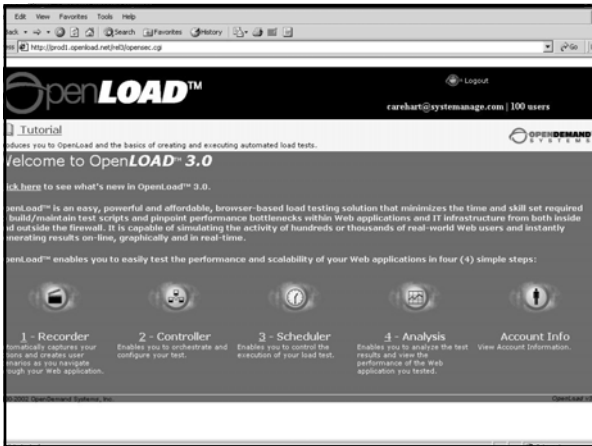


11

Step 1: Recorder

- Setup a "User Scenario"
 - This describes how a user might browse through your site
 - Literally, like turning on a VCR, you'll record your steps through the site
 - Record/stop/pause/resume
 - No scripting required!

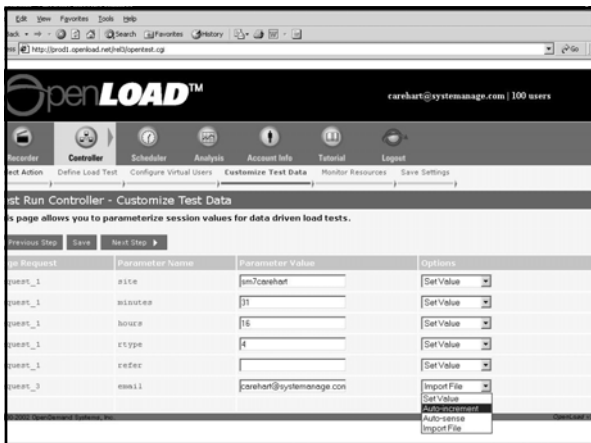
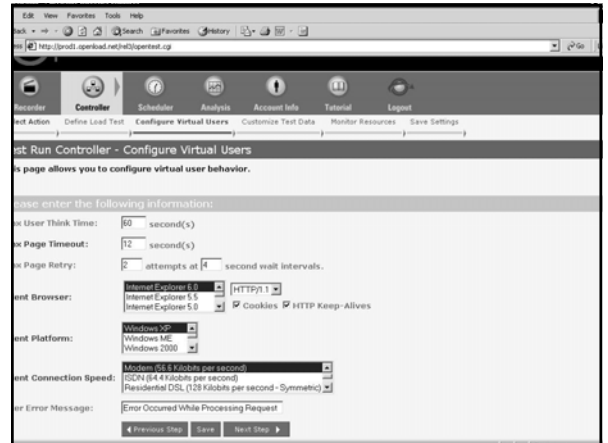
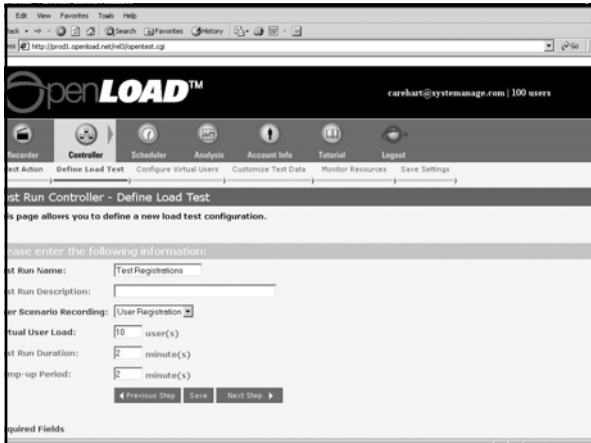
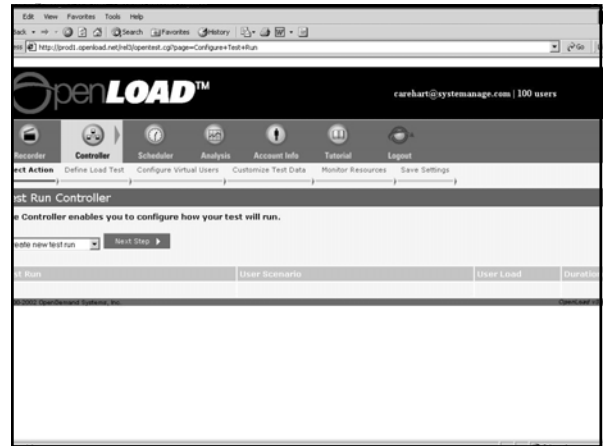
12



Step 2: Controller

- Configure a Load Test “Controller”
 - This describes the characteristics of the virtual users you want to simulate
 - Number of users/duration of test
 - Browser type/connection speed to simulate
 - Get varying form input data from a file
 - Set server-side monitoring
 - And more

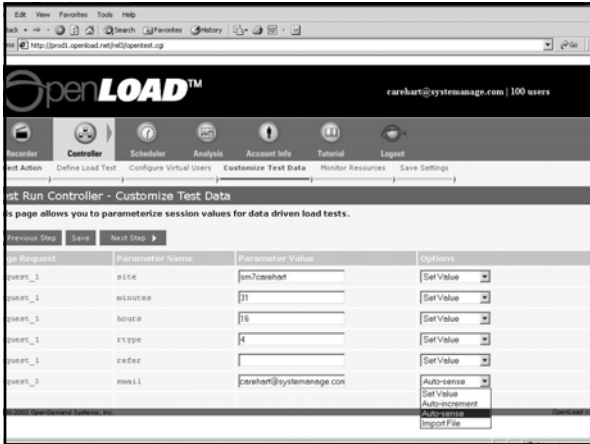
19



Auto-increment

- **Sequentially increments the parameter value by one for each new virtual user**
 - **Allows each virtual user to pass a different parameter value, such as a unique login ID**
- **Default behavior is to append a unique virtual user ID to the parameter value**
 - **Can instead specify where auto-increment takes place by inserting a '?' at any point in the parameter value**

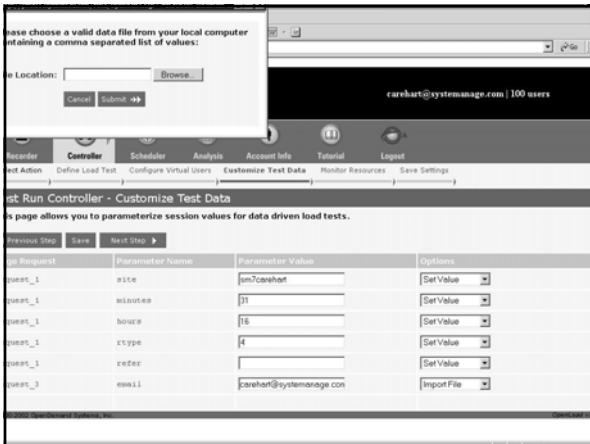
24



Auto-sense

- Automatically retrieves value associated with selected parameter name from each Web server response
- Allows for easy recognition of session dependent parameter values hidden within pages, such as session IDs

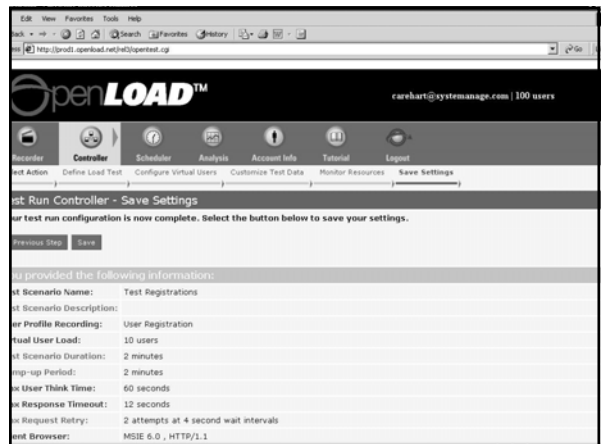
26



Import File

- Uploads file containing list of comma separated values and associates them with selected parameter name
- Allows for greater customization of test data input

28



Step 3: Scheduler

- Schedule a Load Test
 - Can run a test immediately
 - Schedule one for a future time
 - And more



31

OpenLOAD™ carhart@systemmanage.com | 100 users

Recorder Controller Scheduler Analysis Account Info Tutorial Logout

Next Action Define Load Test Configure Virtual Users Customize Test Data Monitor Resources Save Settings

Test Run Controller

Test run "Test Registrations" was successfully saved.

The Controller enables you to configure how your test will run.

No Scheduler [Next Step >]

Run	User Scenario	User Load	Duration
Test Registrations	User Registration	10	120s

OpenLOAD™ carhart@systemmanage.com | 100 users

Recorder Controller Scheduler Analysis Account Info Tutorial Logout

Next Action Schedule Load Test

Test Run Scheduler - Execute Test

Select the test run you wish to execute from below. Selecting the submit button will cause the test run to execute at the next available time slot.

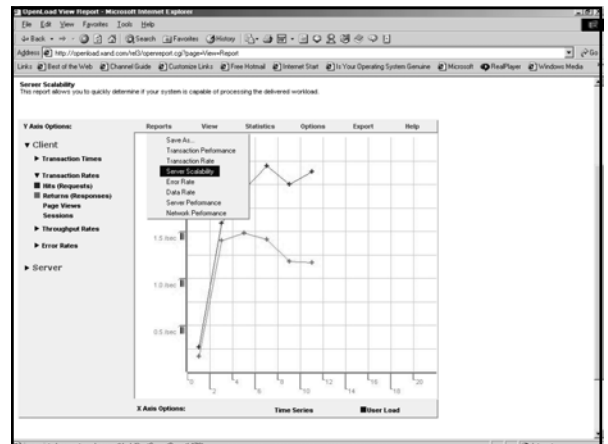
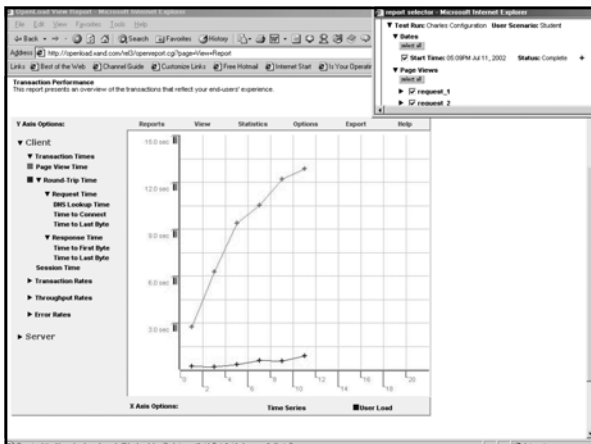
Previous Step [Submit >>]

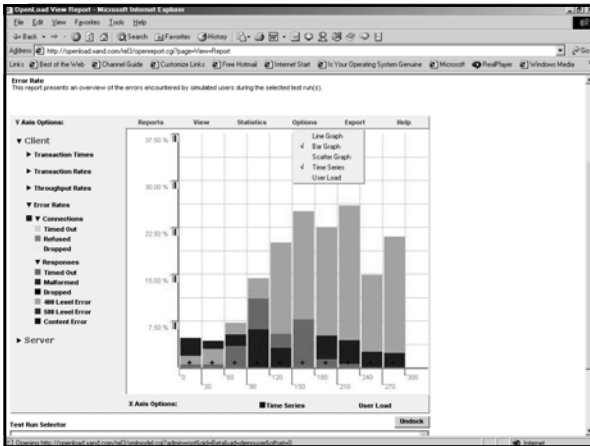
Run	User Load	Duration	Run when #
Test Registrations	10	120s	<input checked="" type="checkbox"/>

Step 4: Analysis

- Analyze Test Results
 - View the performance of your web app
 - Provides multiple graphs for review
 - Each graph offers variations of how/what data is presented
 - Can graph multiple types of data at once
 - Can even graph multiple test runs at once

34





The screenshot displays the 'Test Summary' report in OpenLoad. It contains a table with the following data:

Select Test	Reports	View	Statistics	Options	Export	Help
Website	www.review.com	Graph Mode Test Mode		Select a Test Scenario		
User Load	Expected 15 users	Actual 10 users	Expected 10 users	Actual 6 users		
Duration	Expected v Actual 300 sec	Actual 289 sec	Expected 300 sec	Actual 273 sec		
Shared Pages (time in seconds)	Pageview	% Timedout	Load Time	Pageview	% Timedout	Load Time
	request_2	68.87%	11.43 sec	login	0.00%	3.87 sec
	request_1	53.33%	10.25 sec	request_6	0.00%	3.44 sec
	request_2	53.33%	10.35 sec	request_9	0.00%	3.44 sec
	request_4	53.33%	9.89 sec	request_1	0.00%	3.41 sec
	request_5	14.29%	8.88 sec	request_8	0.00%	2.78 sec
% Pageview Times:		% of Pageviews		% of Pageviews		
<= 4 seconds		5.41%		68.42%		
<= 8 seconds		24.32%		31.59%		
<= 12 seconds		21.62%		0.00%		
> 12 seconds		48.65%		0.00%		
Error Rates:		Error Rates		Error Rates		
% Timedouts		0.00%		0.00%		
% Refused		0.00%		0.00%		
% Dropped		4.57%		0.00%		
% Response Errors		18.18%		0.00%		

Additional OpenLoad Features

- Supports cookies, sessions, HTTPS/SSL
- Supports web/Windows Authentication
- Supports Frames JavaScript, Java, Flash
- Planned Enhancements:
 - Even more intelligent analysis
 - Broader server-side monitoring
 - Much more

39

Key Benefit for Java Developers

- Easy to use
- No software installation required
- Puts testing into developers hands
 - No longer the province of "the test team"
- Allows you to perform testing throughout application life-cycle
 - Load test from development to deployment

40

Maybe High "Load" Isn't a Problem You Foresee

- Can load testing still be useful for you?
- Can certainly look to common performance tuning opportunities as sources of bottlenecks in high or even low load situations



41

Common Tuning Opps

- Determine impact of various opportunities:
 - Coding changes
 - Database design/config/coding changes
 - JDK tuning and/or version upgrade
 - App server configuration changes
 - Web server configuration changes
 - System architecture changes
 - HTML coding changes



42

Common Tuning Opps: "Classic" Coding Issues

- Certain classic tuning questions may not be that important compared to many others we will discuss
 - Using StringBuffer rather than String
 - Minimize number of synchronized methods in loops
 - NULL objects for garbage collection
 - Local variables are faster than class variables
 - Wrap session beans around entity beans

43

Common Tuning Opps: DB Design/Config

- Database processing is usually large contributor to poor performance
- While some have benefit of a DBA, many do not
 - Starts with choice of right DBMS for the job
 - Proceeds to good database design
 - Proper use of indexing can have big performance impact
 - Use DBMS options to test/tune your SQL and DB design
- Use and tune connection pooling
- Consider stored procedures if supported by DBMS

44

Common Tuning Opps: DB Coding

- Biggest goal is to avoid unnecessary DB I/O
 - Avoid SELECT *
 - Consider SELECT count(*) to get table count
 - Test for your database. Not always best.
 - Use aggregate functions, inner and outer joins; unions
 - Versus coding to achieve same result in app server
 - Avoid sorting (ORDER BY) where possible
 - See Ben Forta's "Teach Yourself SQL in 10 Minutes"
 - If new to SQL or only know basic SELECTs
- Don't do in servlet/JSP what can be done better in the DBMS/SQL

45

Common Tuning Opps: JDK Tuning and/or Version Upgrade

- Tune size of JVM, JVM parms for application profile
 - Small/large footprint apps have different needs
 - See vendors usage guides for specific JVM tips
- Consider alternate JVM
 - Sun's HotSpot - java.sun.com/products/hotspot/index.html
 - BEA's JRockit - www.bea.com/products/weblogic/jrockit/index.shtml
 - acquired from Appeal Virtual Machines
- Consider upgrade to latest JDK

46

Common Tuning Opps: Web Server Config

- Web server limitations (IIS on Win2k Pro supports only 10 users by default, for instance)
- Don't use SSL pages where they're not needed
 - Ok for credit card/privacy info request processing
 - But don't put whole site or section under SSL if not needed
- Consider hardware level encryption and decryption

47

Common Tuning Opps: System Architecture

- Consider benefits of scaling hardware/software
 - Vertical scaling: adding memory, CPU power, multiple CPUs per box
 - Horizontal scaling: S/W vs H/W load clustering, round robin DNS
- Consider tiered server configuration
 - Separating database server from web/app server
 - Possibly creating separate image server
 - Possibly sharing a single file/template server for multiple clustered app servers

48

Common Tuning Opps: HTML Coding

- Beware of page “weight” (graphics: too many, large)
- Consider pre-load scripts to load images before they are needed
- Validate on the client whenever possible (JavaScript or ActionScript)

49

Performance Opportunity: Monitoring

- Each app server may offer its own monitoring features
 - Look for slow requests, excessive queue lengths & wait times
- Things you can monitor with O/S tools
 - Memory contention issues (paging)
 - Heavy disk I/O (high service times)
- Other possible monitors
 - Network latency (packet collisions & lost)
 - Database monitoring (excessive IO, locking, etc.)
- OpenLoad has added integration with such monitoring
 - More will be added

50

Improving Apparent Response Time

- Leverage browser caching where appropriate
- Consider gzipping output
 - Can set at web server level, can also program

51

More Tuning Opportunities

- These have been just a few of the more common tuning opportunities
 - Not all will make sense for all sites/apps
- There are still more
 - Many of which may not be obvious but may be valuable
- Planning to create a benchmark site
 - To be published showing impact of various changes
 - Using a standard application suite
- Still need to test/tune these things for yourself

52

Leverage Knowledge of Others

- Perform design and code walkthroughs
- Keep up on, apply industry best practices, coding stds
- Participate in user groups
- Read Java Developers Journal, Java Pro magazines
- Watch sites such as TheServerSide.com, etc.
- Join email mailing lists



53

Resources for Learning More

- JavaWorld Performance Tuning Channel
 - javaworld.com/channel_content/jw-performance-index.shtml
- Java Performance Tuning
 - javaperformancetuning.com/resources.shtml
- Use JDBC for industrial-strength performance
 - javaworld.com/javaworld/jw-01-2000/jw-01-ssj-jdbc.html

54

Test your app for FREE

- Register at <http://www.opendemand.com/java/>
- Test for FREE. Choose either:
 - up to 25 virtual users for one hour
 - or up to 5 virtual users for a currently unlimited time period
- Obtain online copy of this presentation
- Try-out performance tuning tips
 - Test again and compare results to your baseline

55

Q&A



carehart@systemmanage.com
<http://www.systemmanage.com>

info@opendemand.com
<http://www.opendemand.com/>

56