



CONFIDENTIAL



Load Testing with Apache JMeter and Badboy – A Tutorial

COPYRIGHT NOTICE

All rights reserved. These materials are confidential and proprietary to CampusEAI Consortium. No part of these materials should be reproduced, published in any form by any means (electronic or mechanical, including photocopy or any information storage or retrieval system), nor should the materials be disclosed to third parties without the express written authorization of CampusEAI Consortium.



1. Table of Contents

- 1. **Table of Contents** 2
- 2. **Version Control** 3
- 3. **Introduction**..... 4
- 4. **Overview of Apache JMeter** 5
 - 4.1. *Introduction of Apache JMeter*..... 5
 - 4.2. *Features of Apache JMeter*..... 5
- 5. **Overview of Badboy** 6
 - 5.1. *Introduction of Badboy*..... 6
 - 5.2. *Badboy Features*..... 6
- 6. **Load Testing with Apache JMeter and Badboy** 7
 - 6.1. *Overview*..... 7
 - 6.2. *Assumptions*..... 7
 - 6.3. *Steps*..... 7
 - 6.3.1. *Install Weather Portlet on Jupiter*..... 7
 - 6.3.2. *Use Badboy to capture user actions and export to JMeter*..... 8
 - 6.3.3. *Run JMeter to replay the test cases*..... 12
- 7. **Conclusion**..... 23



2. Version Control

| Version No. | Author | Published Date | Sections Changed | Description of changes |
|-------------|-------------|----------------|------------------|------------------------|
| Draft | Zhan Xu | 05/03/2005 | All | Start of Doc |
| 1.0 | Siraj Shaik | 5/6/2005 | All | Reviewed |
| | | | | |
| | | | | |
| | | | | |
| | | | | |



3. Introduction

This document will guide readers through a simple procedure of executing a load test plan, using Apache JMeter and Badboy. Apache JMeter is a tool for load testing. Badboy is a capture/replay testing tool. Combining them together, readers can construct and launch a test plan in a short time.

Readers should refer to the user manuals of Apache JMeter and Bodboy to get a better understanding of them. The parameters shown in the document should be modified in order to simulate more realistic situations.



4. Overview of Apache JMeter

4.1. Introduction of Apache JMeter

Apache JMeter is a 100% pure Java desktop application designed to load test client/server software (such as a web application). It may be used to test performance both on static and dynamic resources such as static files, Java Servlets, CGI scripts, Java objects, databases, FTP servers, and more. JMeter can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

Additionally, JMeter can help you regression test your application by letting you create test scripts with assertions to validate that your application is returning the results you expect. For maximum flexibility, JMeter lets you create these assertions using regular expressions.

4.2. Features of Apache JMeter

Apache JMeter features include:

- Can load and performance test HTTP and FTP servers as well as arbitrary database queries (via JDBC)
- Complete portability and 100% Java purity.
- Full Swing and lightweight component support (precompiled JAR uses packages javax.swing.*).
- Full multithreading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
- Careful GUI design allows faster operation and more precise timings.
- Caching and offline analysis/replaying of test results.
- Highly Extensible:
 - Pluggable Samplers allow unlimited testing capabilities.
 - Several load statistics may be chosen with pluggable timers .
 - Data analysis and visualization plugins allow great extensibility as well as personalization.
 - Functions (which include JavaScript) can be used to provide dynamic input to a test
 - Scriptable Samplers (BeanShell is supported in version 1.9.2 and above)

For more information about Apache JMeter, readers are encouraged to visit JMeter website

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



5. Overview of Badboy

5.1. Introduction of Badboy

Badboy is an application which tries to make developing Web applications easier by combining a capture/replay style unit testing tool with live diagnostics on the activities your browser is performing while you navigate your site.

5.2. Badboy Features

Badboy embeds Microsoft's Internet Explorer and monitors and controls its actions to provide you with a superior, turbo charged browser for testing and development work.

This allows you to

- capture and see the CGI parameters, pages and framesets being requested
- to modify them and replay them at any time without manual intervention
- to save them as scripts and share them with your colleagues.
- to capture performance statistics as you browse
- to regression test whole areas of complex web sites at the click of a button

Badboy includes powerful functionality to allow you to create scripts that navigate complex websites repeatedly without user intervention, even for highly dynamic web sites. It does all this with simple capture-reply and drag and drop script creation that anyone can use. Once created, scripts can be shared to increase productivity for your whole organization.

For more information, readers are encouraged to visit Badboy's website

<http://www.badboy.com.au/>



6. Load Testing with Apache JMeter and Badboy

6.1. Overview

The aim of this document is to guide the readers through a simple example so as to get a first-hand experience on using Apache JMeter and Badboy for load testing. It is not a full documentation on how to use these tools. Readers are always encouraged to visit the official website of Apache JMeter and Badboy to get the latest version and try the new features.

In this example, we will use CampusEAI Weather Portlet as our target. CampusEAI members can download it from the depository.

6.2. Assumptions

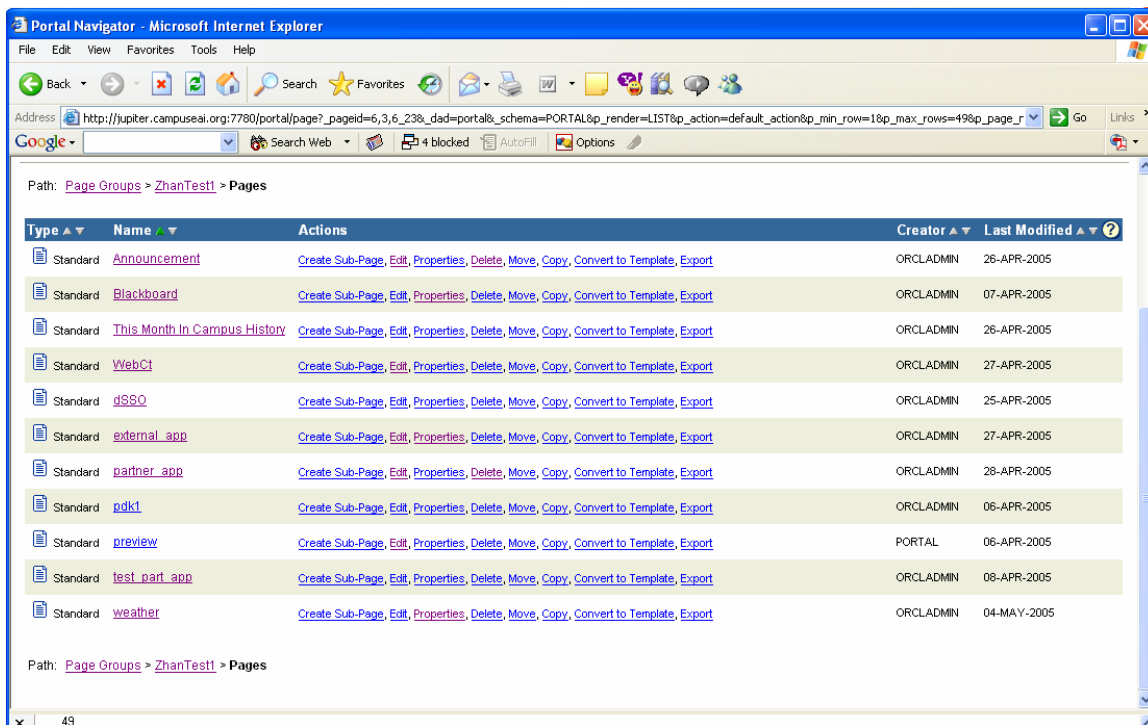
| | |
|--------------------------|-----------------------------------|
| Operating System: | Windows XP |
| Apache JMeter directory: | C:\xz\jakarta-jmeter-2.0.1 |
| Badboy directory: | C:\xz\Badboy |
| Portal server: | http://jupiter.campuseai.org:7780 |

6.3. Steps

6.3.1. Install Weather Portlet on Jupiter

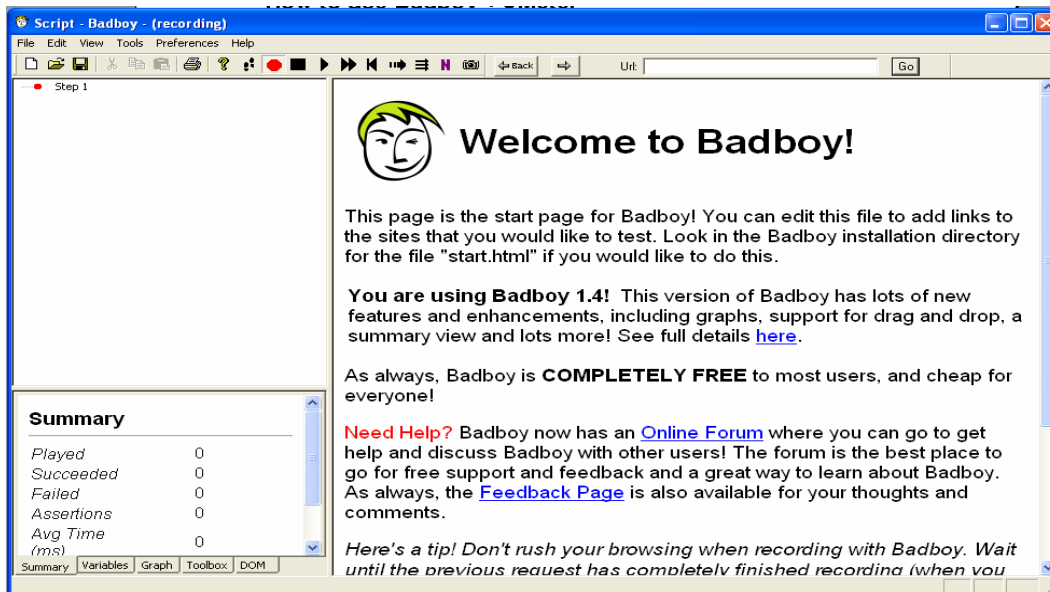
Readers can visit <http://www.campuseai.org> to download the latest version of Weather Portlet from the Portlets Repository.

Follow the installation documents of the Weather Portlet. Create a page group 'ZhanTest1'. Then create a page 'weather' under this page group and add Weather Portlet into this page. Also, make this page accessible for authenticated users.



6.3.2. Use Badboy to capture user actions and export to JMeter

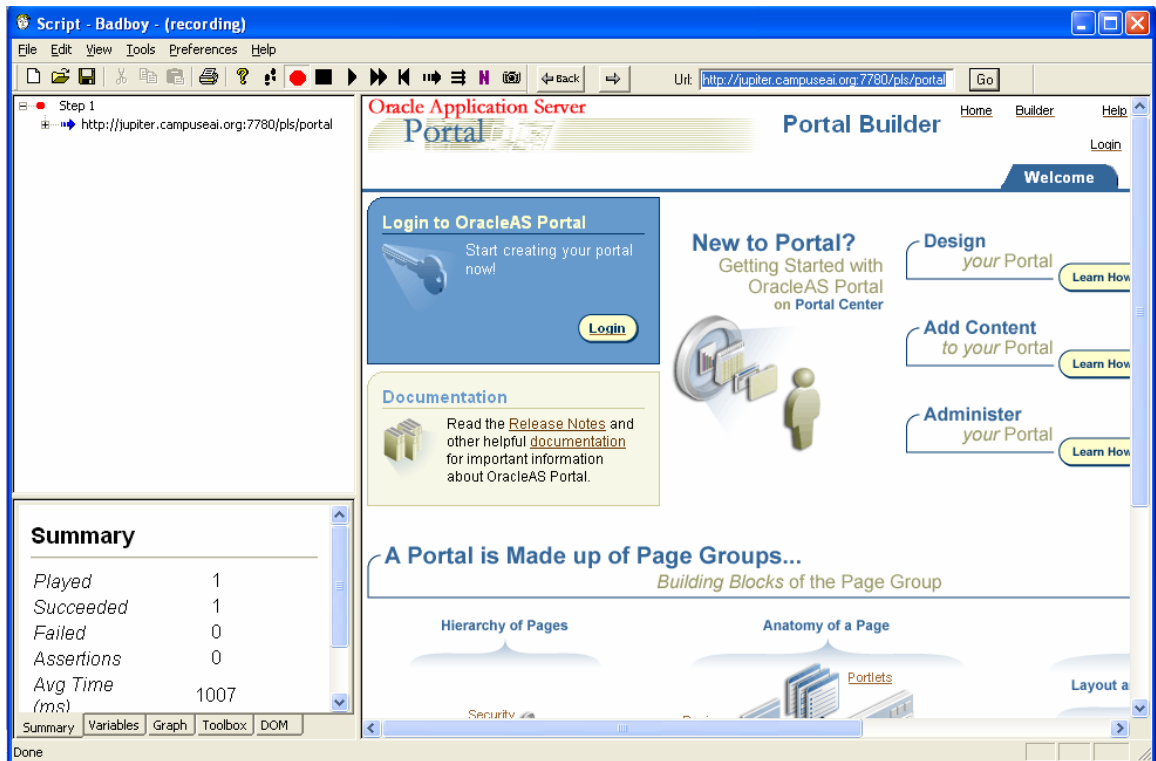
Start Badboy





Make sure that the Record Button  is set to 'Record On'.

Type in the URL of the portal page as if you are going to login and visit the weather portlet. The following figure is a screen shot. Note that on the left pane, Badboy records the URL you just visited.



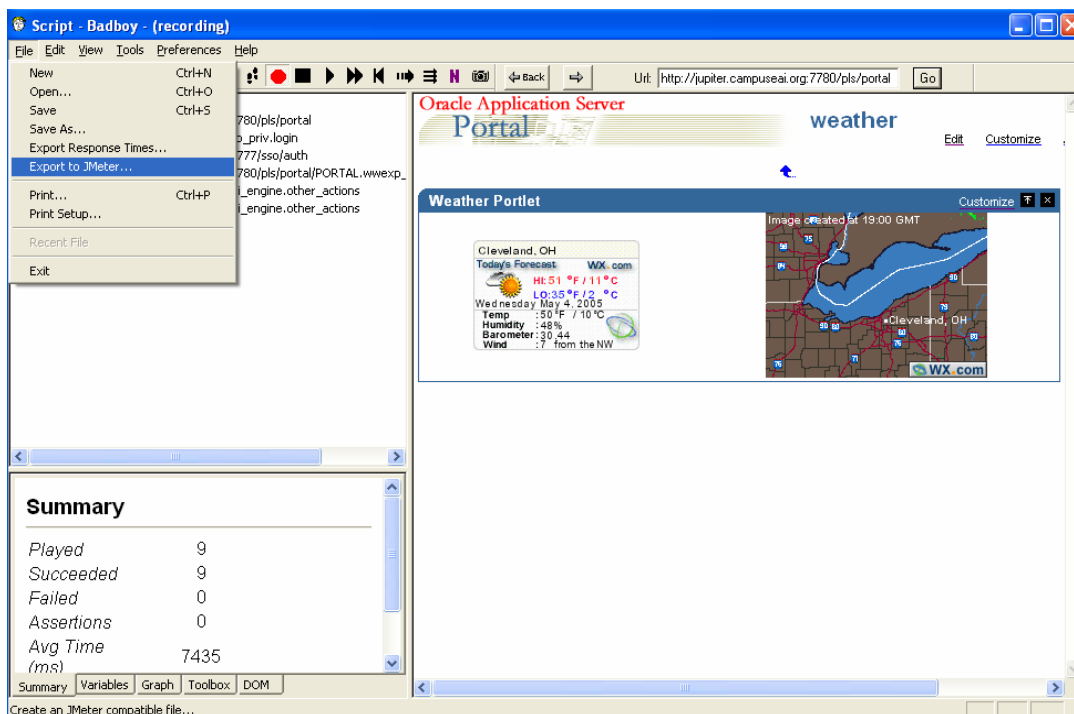
Continue to login to the portal. Navigate inside the portal and visit the page that contains the Weather Portlet. In this example, the navigation path is Navigator → ZhanTest1 → Pages → weather.

The following figure is a screen shot of Badboy during the navigation. Note that Badboy records every link we just visited.

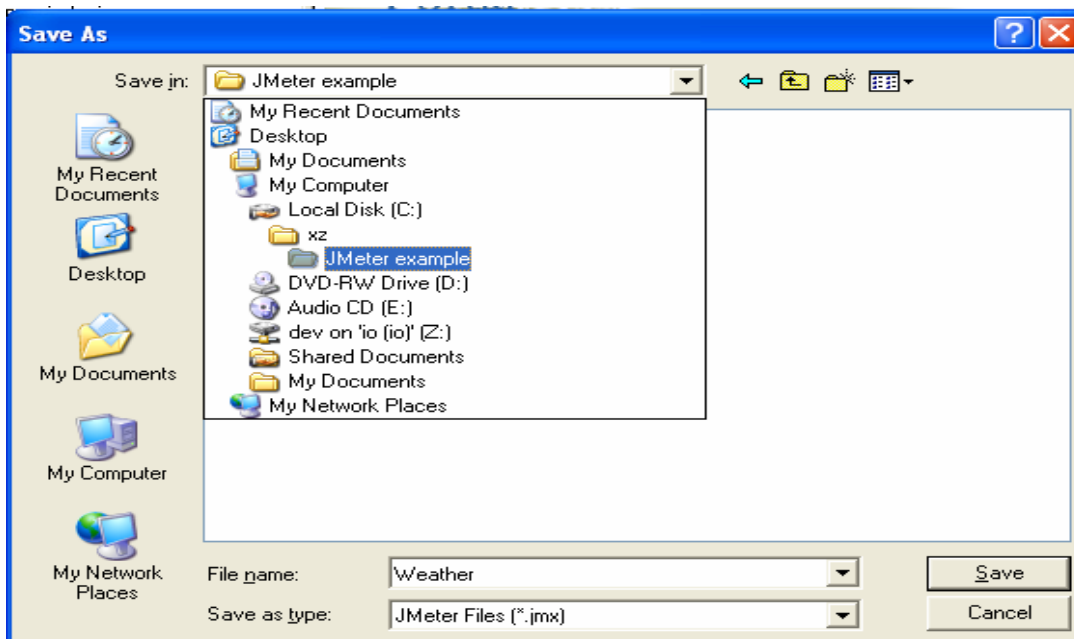


Continue the navigation until all the test links have been visited.

Click File → Export to JMeter... and save the jmx file.



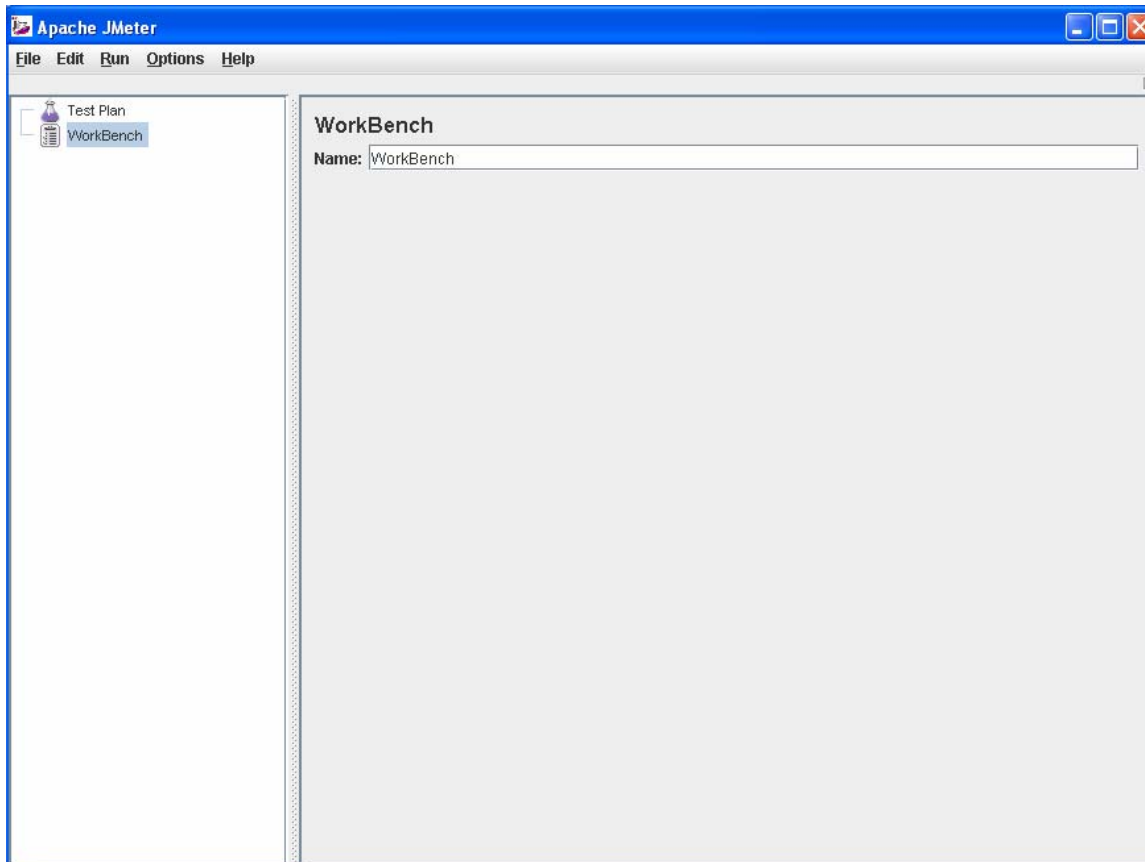
In this example, we save the jmx file as “C:\xz\JMeter example\Weather.jmx”.



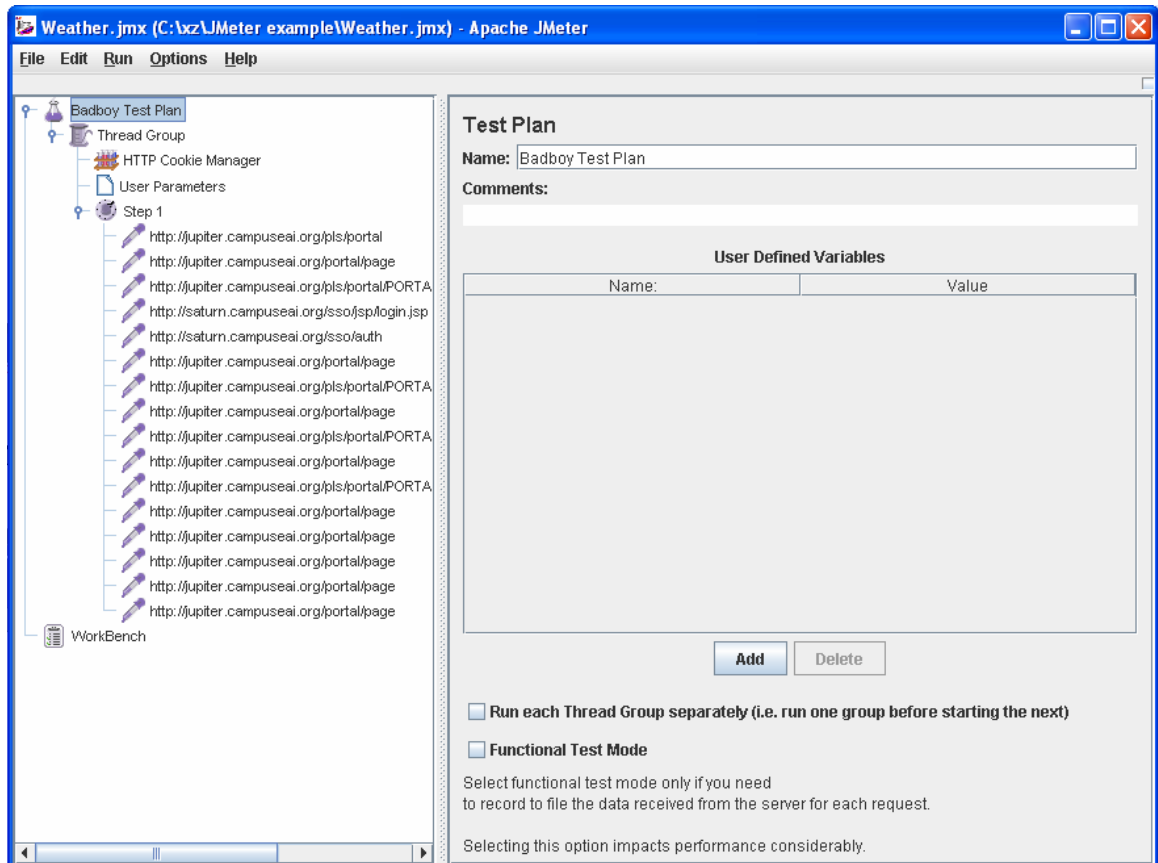


6.3.3. Run JMeter to replay the test cases

Start JMeter.



Click File → Open, and select the 'Weather.jmx' we just saved within Badboy.



At the same time, open the OEM (Oracle Enterprise Manager). Generally, the link is like

<http://your.portalserver.address:1810>

Then login as `ias_admin`. Select your portal instance → `OC4J_Portal`, you will see a list of statistics such as 'Open JDBC Connections', 'Active Sessions' etc, as shown in the following figure.



The screenshot shows the Oracle Enterprise Manager 10g console for the application server 'OC4J: OC4J_Portal'. The status is 'Up' with a start time of 'Apr 26, 2005 11:12:38 AM' and 1 virtual machine. Performance metrics include 0.00% CPU usage, 203.21 MB memory usage, and 38.38 MB heap usage. Under 'JDBC Usage', there are 8 open connections and 1,178 total connections. Under 'Response - Servlets and JSPs', there are 3 active sessions and 2 active requests. Under 'Response - EJBs', there are 0 active EJB methods.

| General | |
|------------------|--------------------------|
| Status | Up |
| Start Time | Apr 26, 2005 11:12:38 AM |
| Virtual Machines | 1 |

| Status | |
|-------------------|--------|
| CPU Usage (%) | 0.00 |
| Memory Usage (MB) | 203.21 |
| Heap Usage (MB) | 38.38 |

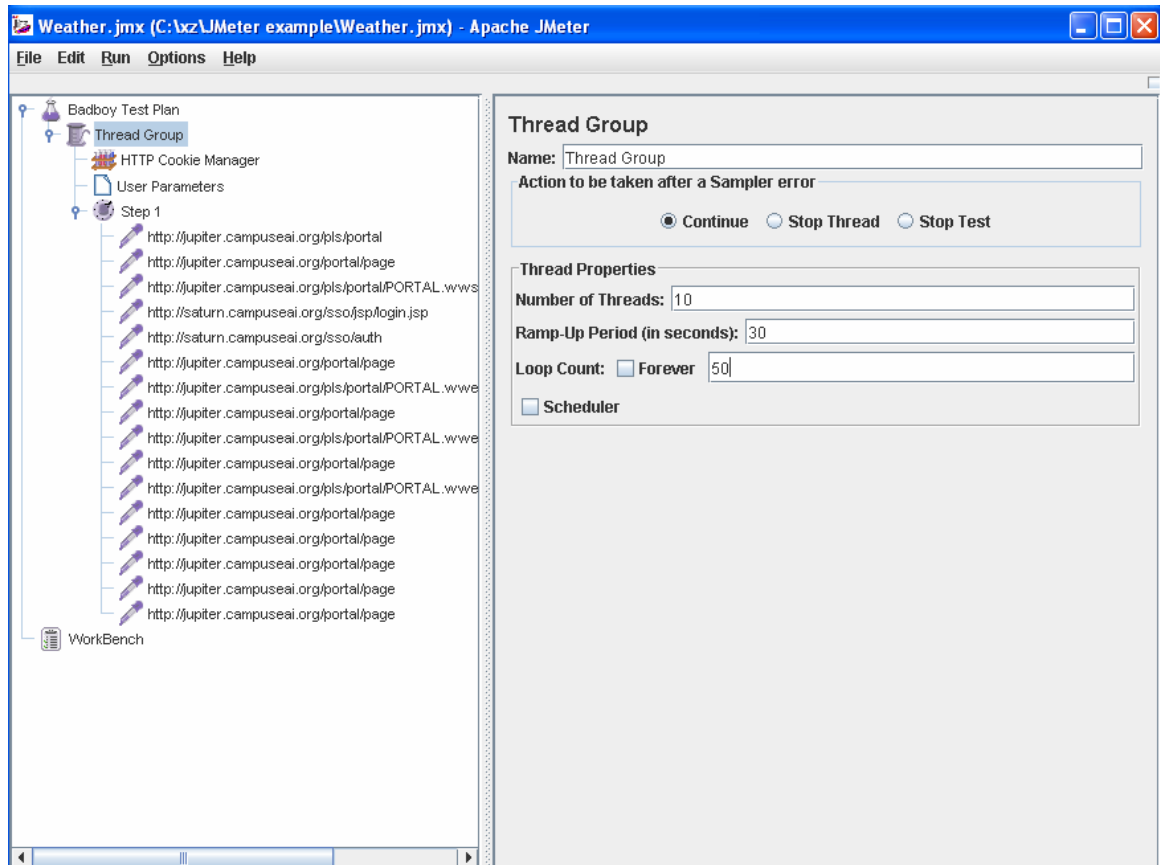
| JDBC Usage | |
|------------------------|-------|
| Open JDBC Connections | 8 |
| Total JDBC Connections | 1,178 |
| Active Transactions | 1 |
| Transaction Commits | 1 |
| Transaction Rollbacks | 0 |

| Response - Servlets and JSPs | |
|-----------------------------------|------|
| Active Sessions | 3 |
| Active Requests | 2 |
| Request Processing Time (seconds) | 0.04 |
| Requests per Second | 0.09 |

| Response - EJBs | |
|------------------------------------|------|
| Active EJB Methods | 0 |
| Method Execution Time (seconds) | 0.00 |
| Method Execution Rate (per second) | 0.00 |

Now let's go back to JMeter and add some parameters to make it more realistic.

On the left pane, click 'Badboy Test Plan' → Thread Group. The parameters for the threads are shown on the right pane. This is shown in the following figure.



Thread group elements are the beginning points of any test plan. All elements of a test plan must be under a thread group. As the name implies, the thread group element controls the number of threads JMeter will use to execute your test. The controls for a thread group allow you to:

- Set the number of threads
- Set the ramp-up period
- Set the number of times to execute the test

Each thread will execute the test plan in its entirety and completely independently of other test threads. Multiple threads are used to simulate concurrent connections to your server application.

The ramp-up period tells JMeter how long to take to "ramp-up" to the full number of threads chosen. If 10 threads are used, and the ramp-up period is 100 seconds, then JMeter will take 100 seconds to get all 10 threads up and running. Each thread will start 10



(100/10) seconds after the previous thread was begun. If there are 30 threads and a ramp-up period of 120 seconds, then each successive thread will be delayed by 4 seconds.

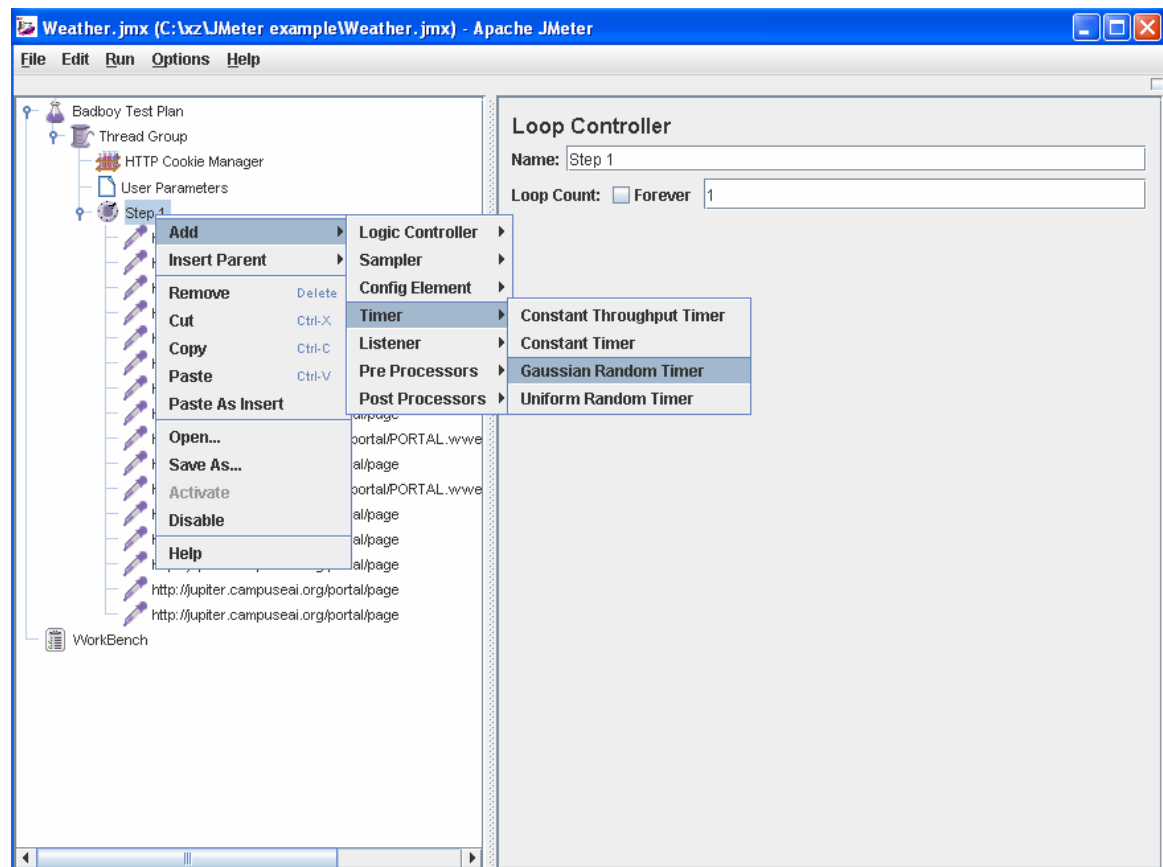
By default, a thread group is configured to loop indefinitely through its elements. Alternatively, you can set the number of times the thread group will loop before ending. If the number is set at one, then JMeter will execute the test only once before stopping.

In our example, we will set

| | |
|--------------------|------------|
| Number of threads: | 10 |
| Ramp-Up Period: | 30 seconds |
| Loop Count: | 50 |

These parameters means that every 3 seconds, a user will start to visit portal and the 'Weather Portlet'. Totally, there will be 500 (10 * 50) users.

Now let's add some randomization into user's action. Right click Step 1 → Add → Timer → Gaussian Random Timer, as shown in the following figure.

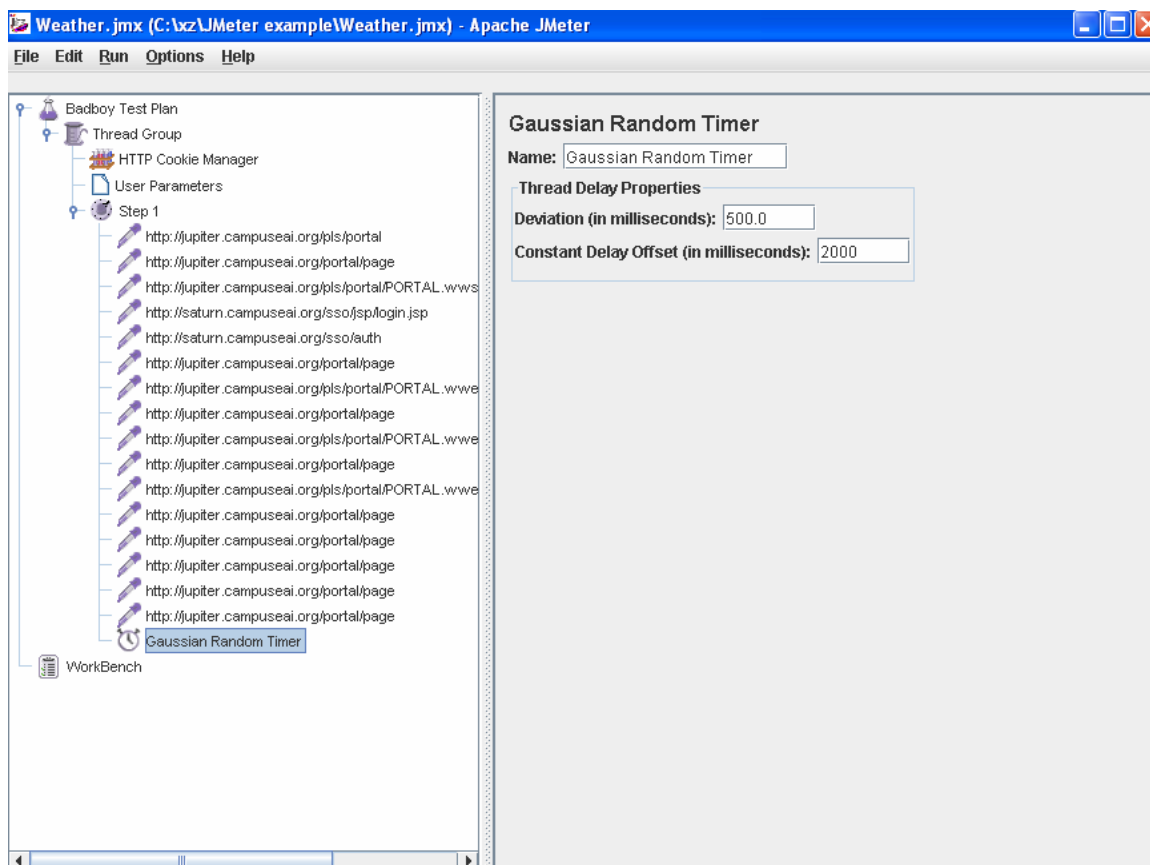




There are two parameters for Gaussian Random Timer. We can specify it as the following

Deviation: 500 ms
Constant Delay Offset: 2000 ms

User can refer to any statistics book for Gaussian distribution and get a better understanding of Gaussian Random Timer.



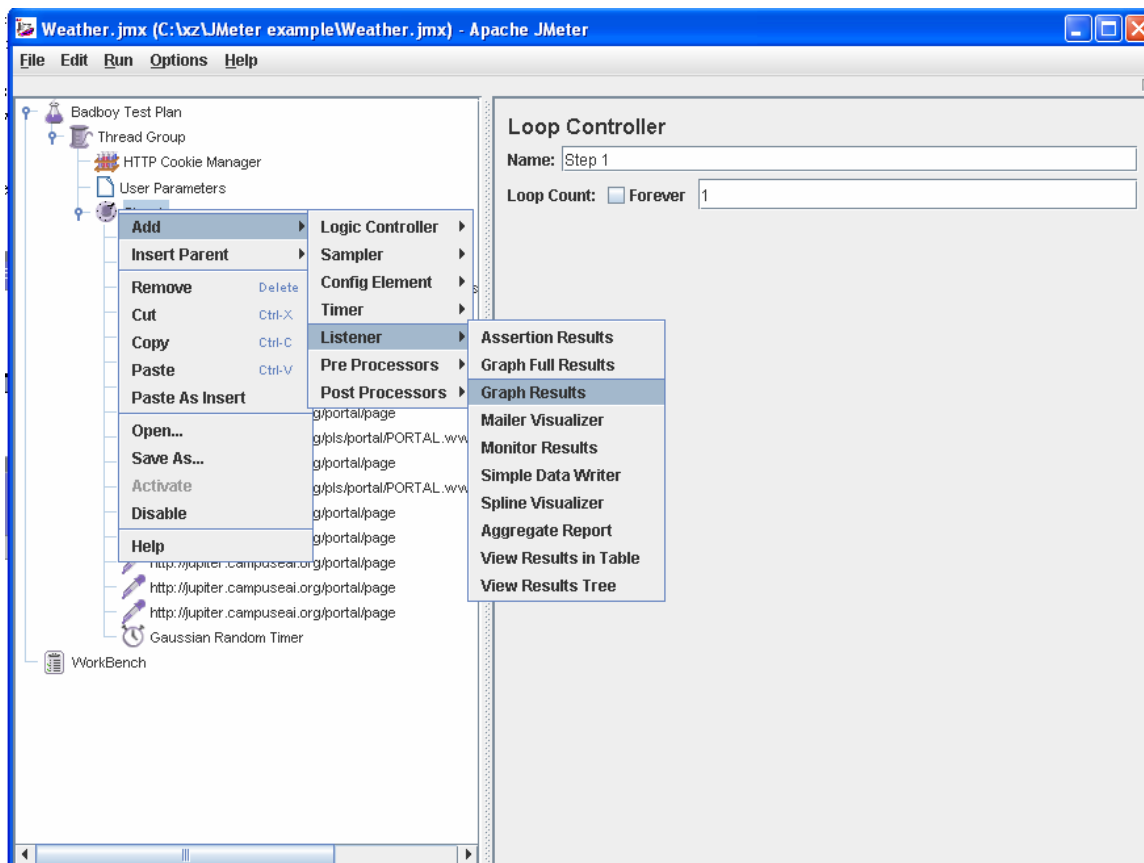
In our case, this setting of Gaussian Random Timer defines the users' behavior. Each user will 'click' the links specified on the left pane one by one. There will be an interval between every consecutive 'clicks', which is Gaussian distribution with average 2000ms and deviation 500ms.

Readers can modify parameters of thread groups and Gaussian Random Timer to cater their own test needs.

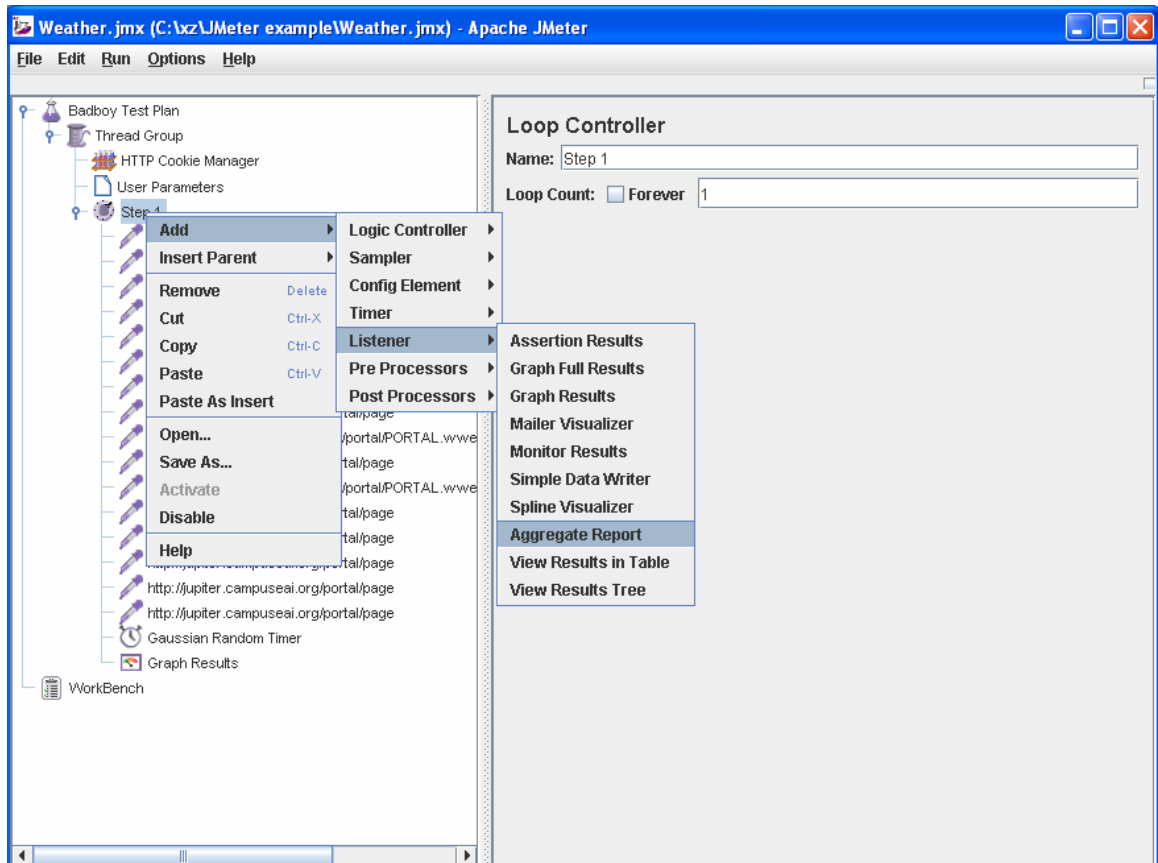


After defining the thread groups and Gaussian Random Timer, we need to add some listeners so that we can get the statistics of our test. Two useful listeners are Graph Results and Aggregate Report.

To add Graph Results, right click 'Step 1' → Add → Listener → Graph Results, as shown in the following figure



To add Aggregate Report, right click 'Step 1' → Add → Listener → Aggregate Report, as shown in the following figure.



Now let's start the test.

Click Run → Start



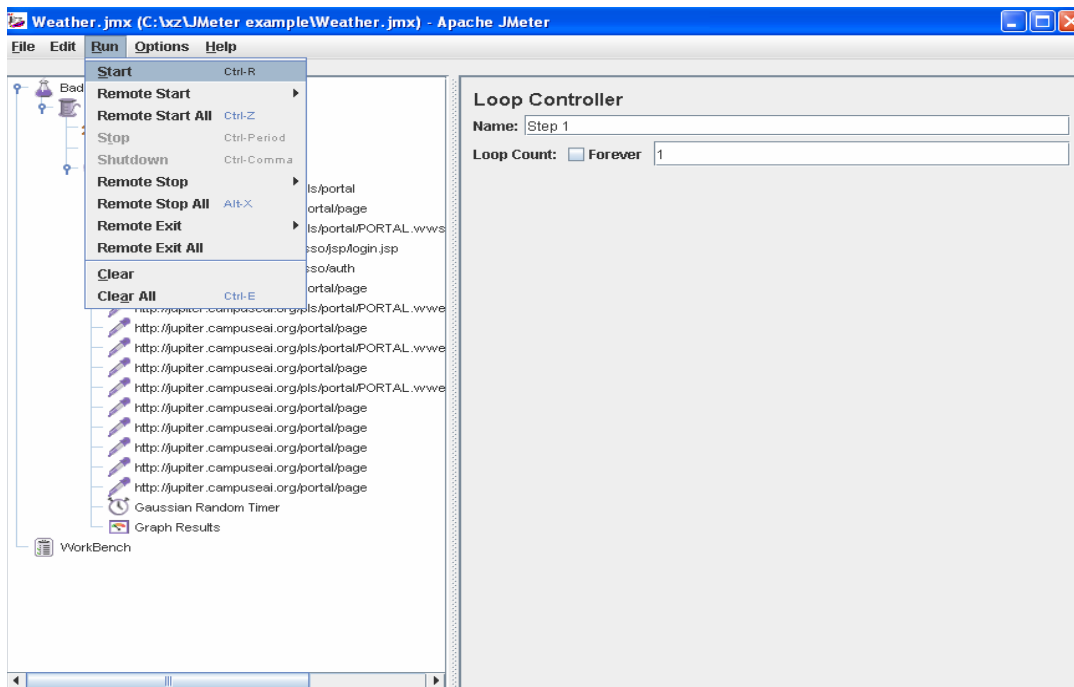
Load Testing with Apache JMeter and Badboy - A Tutorial



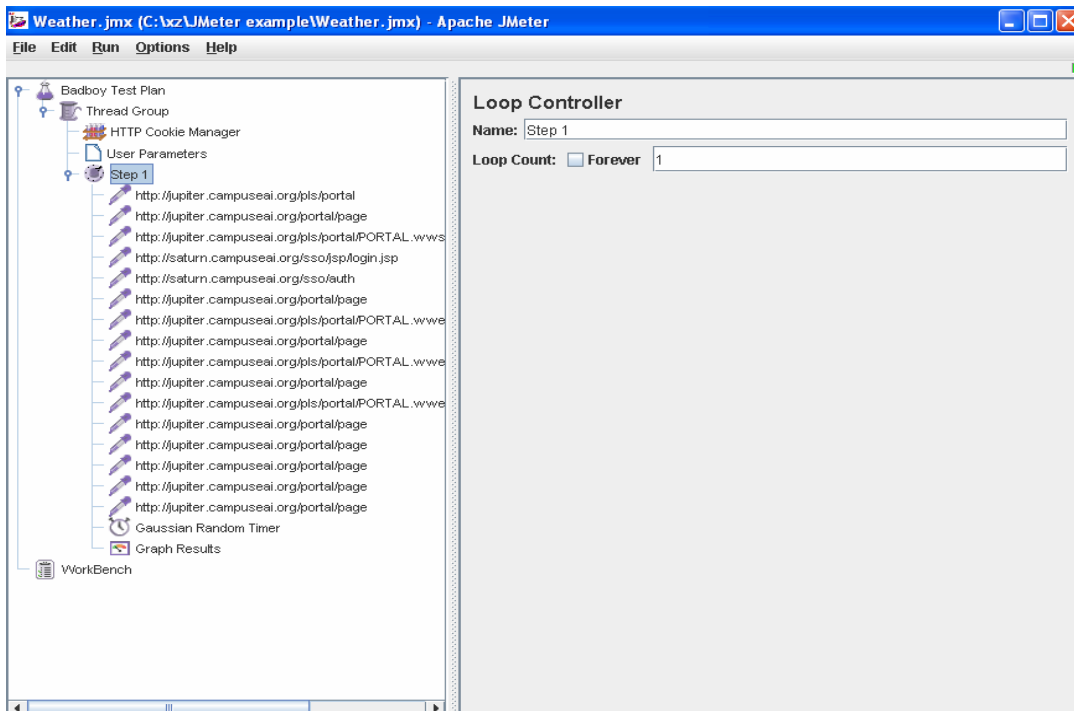
LOAD TESTING TUTORIAL

MAY 2005

SOFTWARE DEVELOPMENT PRACTICE



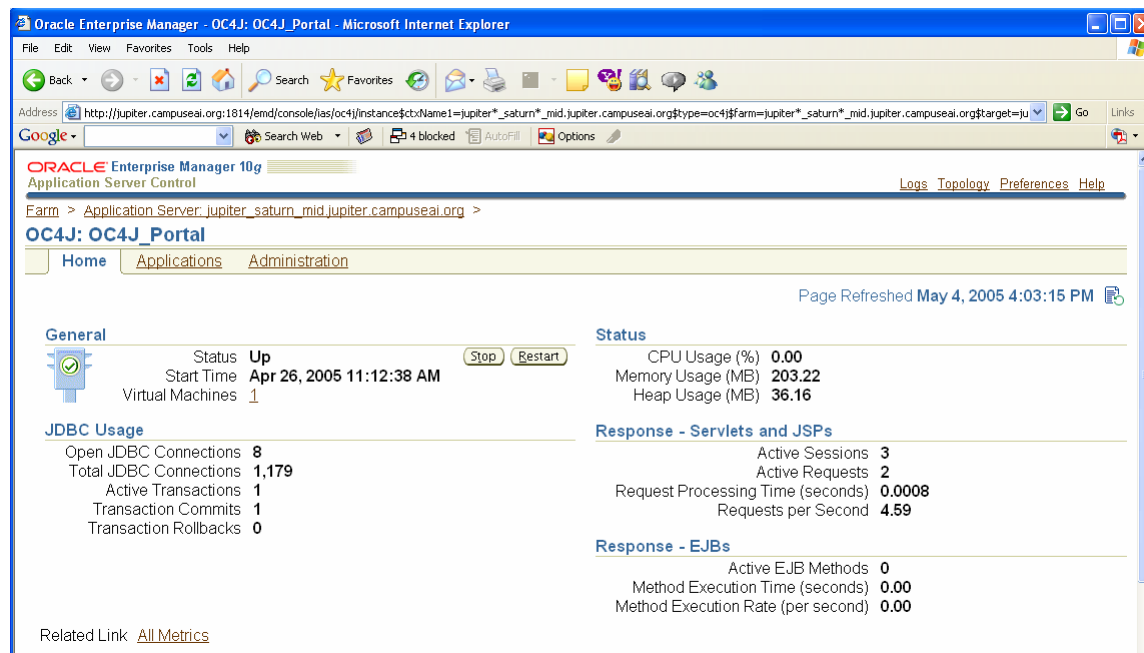
Note that a green bar is shown on the right-top of JMeter. This means that the tests are running.





Wait for the green bar to become grey and finish the tests.

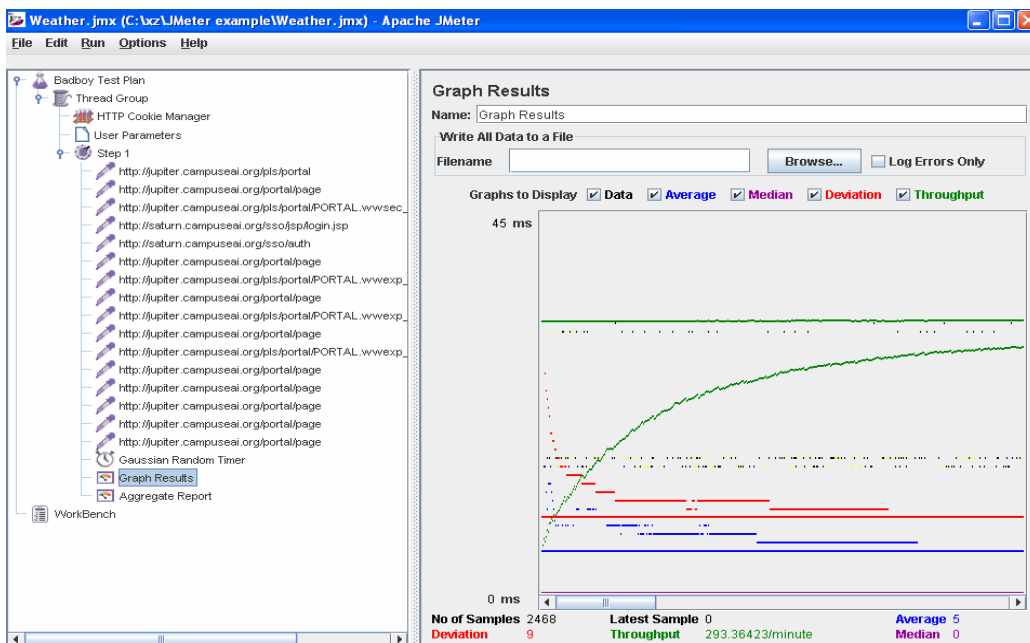
While waiting, we can go back to the OEM and see the statistics for OC4J_Portal. Note that in our case, the statistics of 'Requests per Second' is changed from 0.09 to 4.59. This change is because of the running of JMeter and our test plan for Weather Portlet.



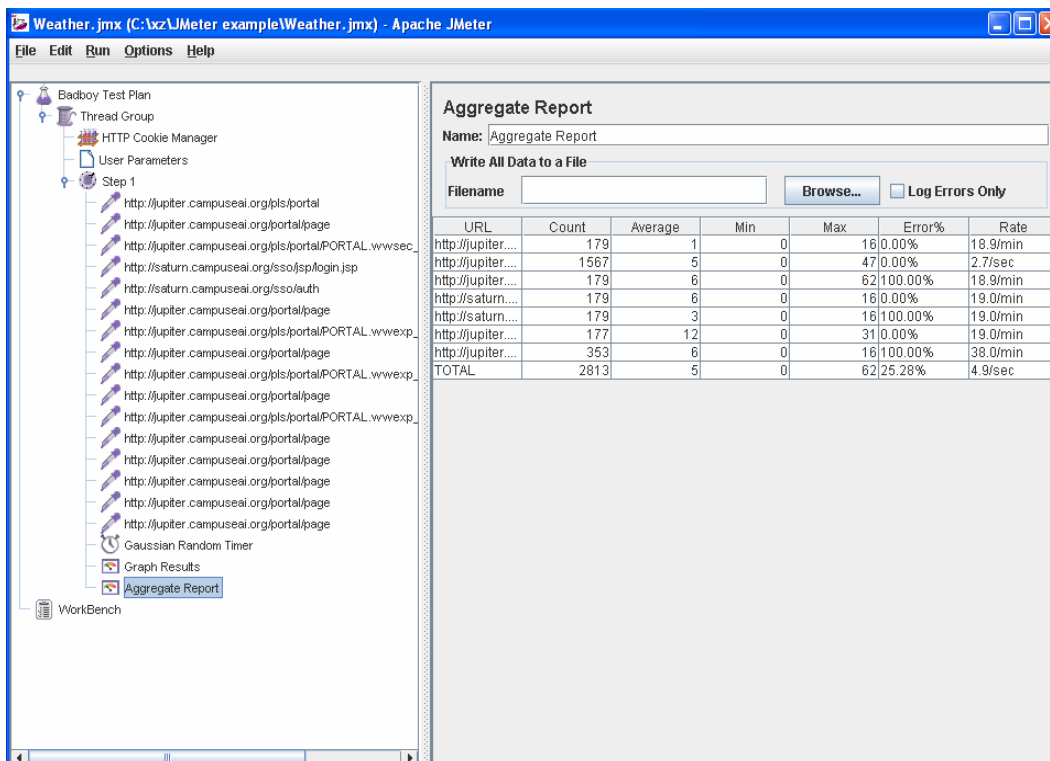
We can click 'Graph Results' and see the graphical statistics, as shown in the following figure.



Load Testing with Apache JMeter and Badboy - A Tutorial



We can also click 'Aggregate Report' and see the aggregate statistics, as shown in the following figure.





7. Conclusion

This document guides readers through a simple procedure of creating and executing a load test plan, using Apache JMeter and Badboy.

From the 'Graph Results', a time-line based statistics such as Throughput and Average is shown, while the test plan is being executed.

From the 'Aggregate Report', we can see the statistics such as Count, Error and Rate for each link that is covered by the test plan. Final statistics for all the links is also available for review.

The statistics is a starting point for further improvement of our code. For example, we can find out which link takes the longest time and start our code optimization.