Assessment Submission Form

Student Name(s)	**Kamil Smuga, Tanveer Khan**
Student Number(s)	**12258018, 12257826**
Assessment Title	**Quality Assurance Report**
Course	**Performance of Distributed Systems (COMP40550)**
Lecturers	**Prof. John Murphy and Prof. Liam Murphy**
Head Tutor	**Dr. Christina Thorpe**
Date Submitted	**08/08/2013**
OFFICE USE ONLY **Date Received**	

A SIGNED COPY OF THIS FORM MUST ACCOMPANY ALL SUBMISSIONS FOR ASSESSMENT.

STUDENTS SHOULD KEEP A COPY OF ALL WORK SUBMITTED.

Procedures for Submission and Late Submission
Ensure that you have checked the School's procedures for the submission of assessments.
Note: There are penalties for the late submission of assessments. For further information please see the University's ***Policy on Late Submission of Coursework***, (http://www.ucd.ie/registrar/)

Plagiarism: the unacknowledged inclusion of another person's writings or ideas or works, in any formally presented work (including essays, examinations, projects, laboratory reports or presentations). The penalties associated with plagiarism designed to impose sanctions that reflect the seriousness of University's commitment to academic integrity. Ensure that you have read the University's ***Briefing for Students on Academic Integrity and Plagiarism*** and the UCD ***Plagiarism Statement, Plagiarism Policy and Procedures***, (http://www.ucd.ie/registrar/)

Declaration of Authorship
I/We declare that all material in this assessment is my/our own work except where there is clear acknowledgement and appropriate reference to the work of others. **For joint assignments:** we declare that we are satisfied with the division of effort in our assignment.

Signed... Kamil Smuga ..... Date... 08/08/2013 .....

Signed... ..... Date ... 08/08/2013 .....

Signed..... Date .....

Signed..... Date .....

Executive Summary

Analyzing the performance of a soon to be released application called AdaptiveCells/J has provided some useful insight into the application after functional, load, stress testing which led to performance tuning of the system along with model simulation to improve the system.

Functional testing concluded that the application worked reliably with only config: 3, 5, 6, 8, and 10 as the rest of the configuration resulted in failures. These configurations were then used extensively for the rest of the testing.

Load testing was targeted to characterize application performance trends during average load. To simulate and monitor realistic user load use of tools like JMeter and Cacti was quiet advantageous. Load testing results provided performance characteristic for the system under average load. Testing parameters were tweaked to not exceed good usability and user experience threshold. 1 sec response time has been set as a top value for load tests. Peak simulation scenarios for certain configs have exceeded this threshold sporadically, however, it was really tiny and overall performance (testing all configs) is below 1 sec on peak. In summary, performance for 50 concurrent users is astonishingly good – system is responding instantaneously – below 50 ms. Response times for 100 and 200 concurrent users are between 500-1000 ms, which provides reasonable performance and user experience. 200-user scenario started generating marginal erroneous responses in 0.5-1.4% ranges.

Performance testing was focused on overloading the system in order to find its limits, bottlenecks and performance characteristics. Performance tests indicated CPU as a bottleneck. Every other parameter – network, memory – was utilized only in a small part. Tests with 500 users for Config 6 and 8 may indicate system responded in an acceptable time – up to 1 sec. However, it depends how much system should be sensitive for errors. Response error level was in 5-8% range, which is much higher than results from load testing. Results from 1000 and 2000 concurrent users testing exceeded 1 sec mark response time and returned errors in double-digit percentage values.

A number of system optimizations have been proposed. Rerunning the same tests for all configs has proved improvements. Changes included Jboss configuration tweaks, garbage collection parallelism and JVM heap ratio balance.

JMT modeling was used to simulate the live system, which provides facility tweak the hardware capacity in simulation mode. With the Closed Model simulating the single Jboss server we were able to validate that the model. Also, we were able to demonstrate that model behaves similar to the live scenario by increasing the number of jobs for 2 different page accesses (Config3, Config5) and comparing the throughput numbers from the load testing results.
Noticed that by adding one more server and changing routing strategy to the closed model throughput can be increased. The routing strategy that suits for this model is

Shortest R time and Join the Shortest Q increased the throughput. To improve performance we could add another instance of the application and webserver and load balance them so that requests go to Shortest Q or Shortest R time.

1. Functional testing

Approach

Functional testing phase was focused on application behavior correctness. Test cases were designed to find run-time errors and memory leaks. Tests were manual.
Tools listed below have been used for investigation:

- Splunk - log mining tool for search, analysis and visualization

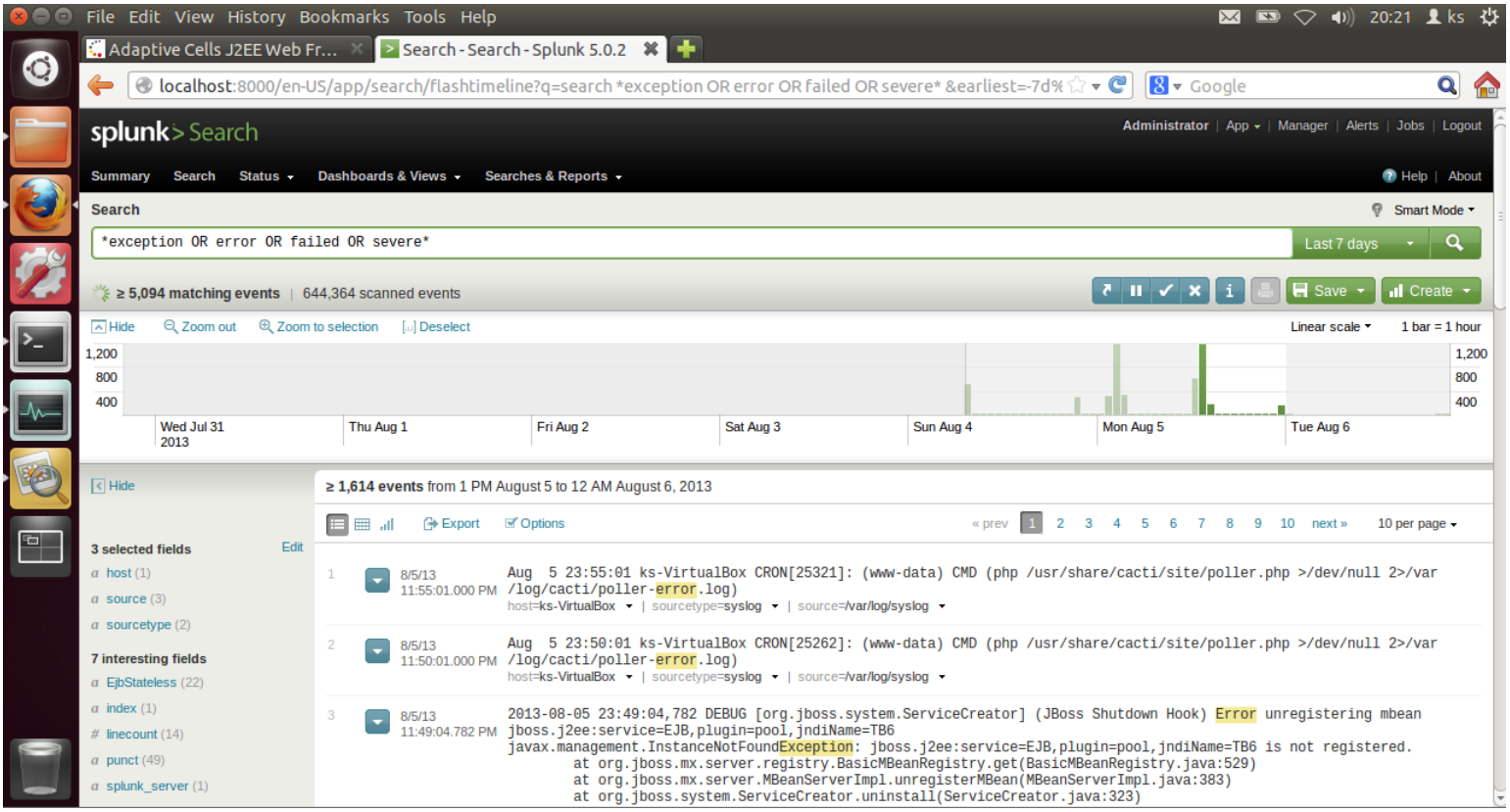


Figure 1: Splunk query results screenshot.

- Visualvm – JVM monitoring and profiling tool

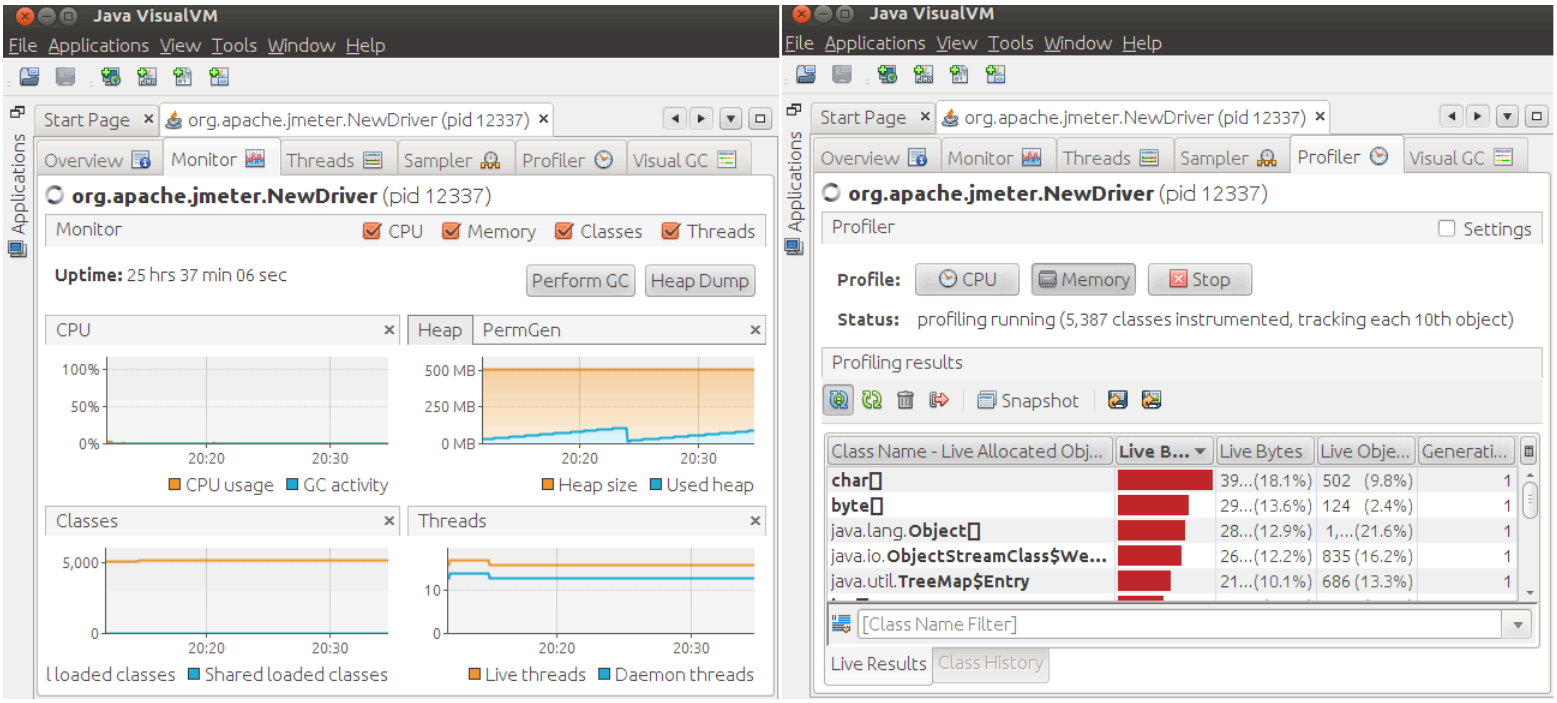


Figure 2: Visualvm monitoring and profiling views.

Tests cases were executed against every AdaptiveCells/J configuration available (config1...10).
Jboss server and boot logs (`JBOSS/server/default/log/server.log, boot.log` and `access.log`) were indexed by Splunk during tests.
For each config, 2 types of test templates were used:

a) run-time exceptions search:
   - start Jboss and let it warm up
   - start and attach Visualvm
   - run config(1...10) once
   - repeat every 2 minutes for 3-4 times
   - run config(1…10) multiple times
   - detach Visualvm
   - stop Jboss
   - analyse logs using Splunk query that is searching for following keywords in logs:
     `"exception OR error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )"`

b) memory leak recognition:
   - start Jboss and let it warm up
   - start and attach Visualvm
   - start Visualvm memory profiler tracking every object allocation and stack traces
   - switch off profiler results automatic refreshing
   - kick off garbage collection
   - clean all profiler results

*** at this point object instances are garbage collected and only new allocations will be displayed in profiler results ***

- execute config(1…10) several times
- kick off garbage collection again
- results contains objects that were not cleaned up properly
- create heap dump
- focus on objects that survived most garbage collections – generations metric in profiler
- for suspicious objects find and investigate stack traces in profiler and find references to instances in heap dump

Results

Before starting any testing executed 'run-time exceptions search' test case to find which exceptions already exists before application deployment. Tested: clean Jboss, with Visualvm attached, with memory profiler running and with G9.ear deployed. Identified numerous exceptions thrown by server before any config execution. Highlights:

- `javax.management.InstanceNotFoundException: jboss.j2ee:service=EJB,plugin=pool,jndiName=TB6 is not registered.` thrown by `org.jboss.mx.server.registry.BasicMBeanRegistry.get(BasicMBeanRegistry.java:529)` – affects all TB (1 to 6) instances
- numerous `java.sql.SQLException: Table already exist`, `java.sql.SQLException: Index already exist`, `java.sql.SQLException: Violation of unique constraint SYS_PK_48: duplicate value(s) for column(s) $$`
- `java.lang.ClassNotFoundException: org.jboss.mx.server.MBeanServerBuilderImpl Caused by: javax.management.JMRuntimeException: Failed to load MBeanServerBuilder`
- `java.lang.IllegalArgumentException: Property is not readable: propertyReplace for org.jboss.beans.metadata.plugins.AbstractPropertyMetaData`

All of those types of exceptions were investigated for every test case run and excluded from further analysis if no new instance of a given type found.

Config 1

No application specific run-time exceptions found.
Found memory leak. Identified suspicious object that was the biggest one on the heap and lived for 3 generations: `byte[]`

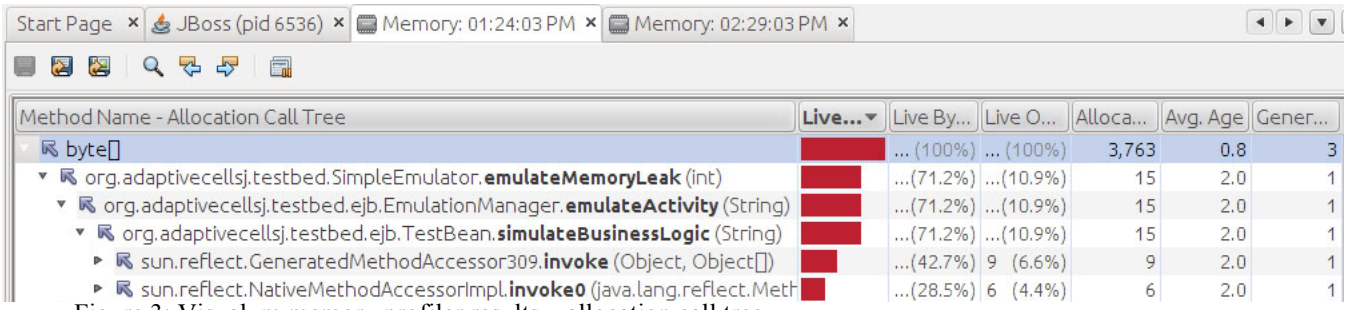


Figure 3: Visualvm memory profiler results – allocation call tree.

Suspected method: `org.adaptivecellsj.testbed.SimulateEmulator.emulateMemoryLeak(int)`.
Therefore, checked for `SimpleEmulator` object references in heap dump. Found 6 instances:

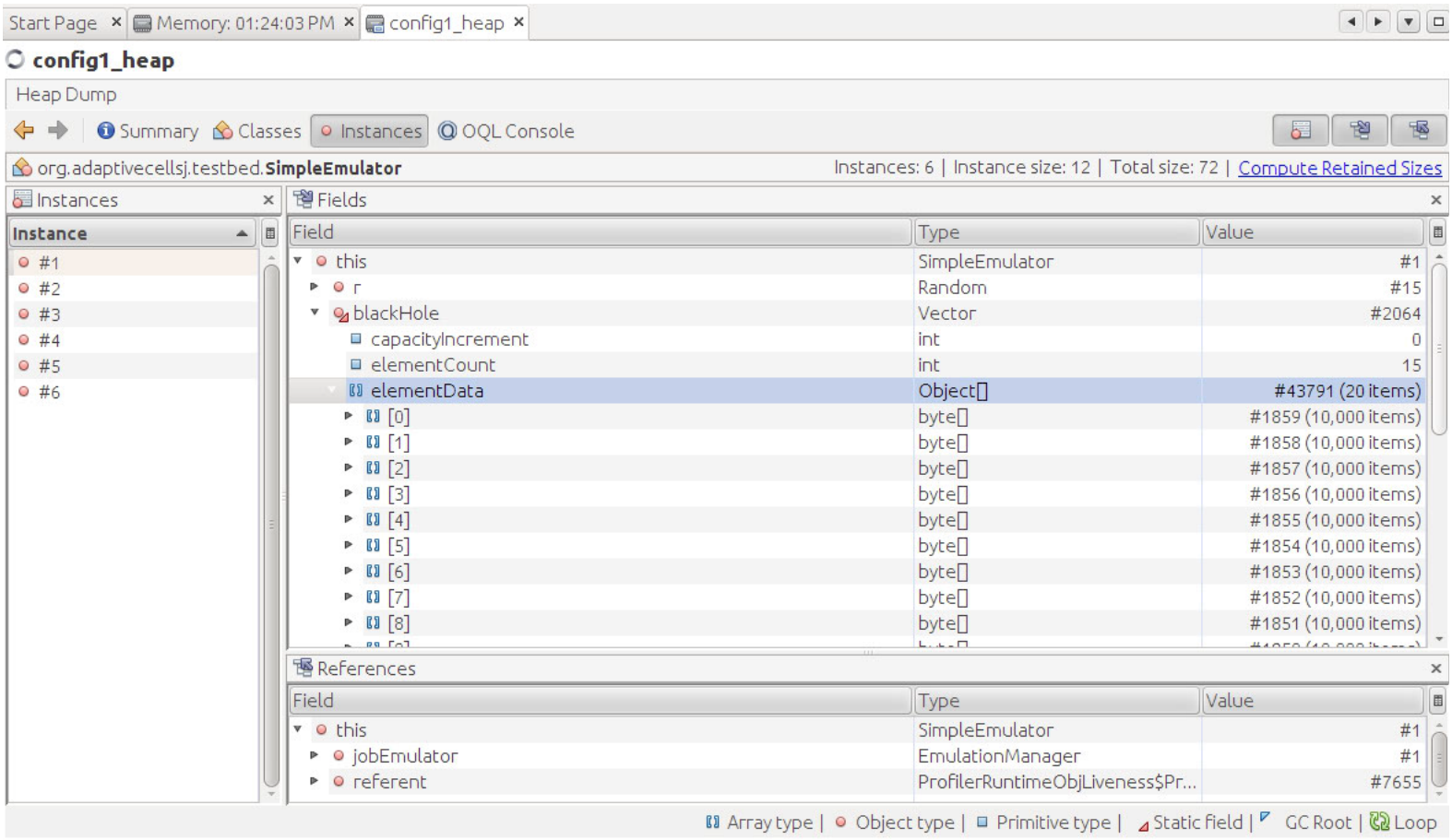


Figure 4: Heap dump view in Visualvm

All of `SimpleEmulator` instances contains a reference variable to object `Vector blackHole` that is encapsulating references to several `byte[]` objects that were not garbage collected properly. Those are the leaked objects.

Exactly the same instances of `SimpleEmulator` and `byte[]` objects leaked in Config 2 and Config 7.

Config 4

Found run-time exception thrown directly in the browser:

```
The Adaptive Cells EJBs have raised the exception:
null; nested exception is: java.lang.RuntimeException
      You might have configured the cells to raise exceptions.
This can be done adding an environment entry (namely
configXexception) to the deployment descriptors of the cell EJBs.
```

Found the same exception in Jboss server.log:

```
TransactionRolledBackException in method:
org.adaptivecellsj.testbed.ejb.TestBeanIF.simulateBusinessLogic(j
ava.lang.String) throws
java.rmi.RemoteException,java.lang.Exception,
caused by RuntimeException thrown from:
org.adaptivecellsj.testbed.SimpleEmulator.emulateException(Simple
Emulator.java:69)
```

Exception occurrences in logs are aligned with Config 4 execution times from browser. Exactly the same exception occurred for Config 9.

No memory leaks found.

Summary for all findings:

	Run-time error	Memory leak
Config 1		x
Config 2		x
Config 3		
Config 4	x	
Config 5		
Config 6		
Config 7		x
Config 8		
Config 9	x	
Config 10		

Table 1: Functional testing findings.

Only Configs: 3, 5, 6, 8, 10 will be included in further testing.

Raw data and detailed results from testing can be found:

- pre config execution results: http://bit.ly/11Ku8kD

- full results: http://bit.ly/191s1uS
- raw data: http://bit.ly/11LGol1

2. Load testing

Approach

Load testing was targeted to characterize application performance trends during average load.
To simulate and monitor realistic user load following tools were used:

- Jmeter - Java application designed to load test functional behavior and measure performance

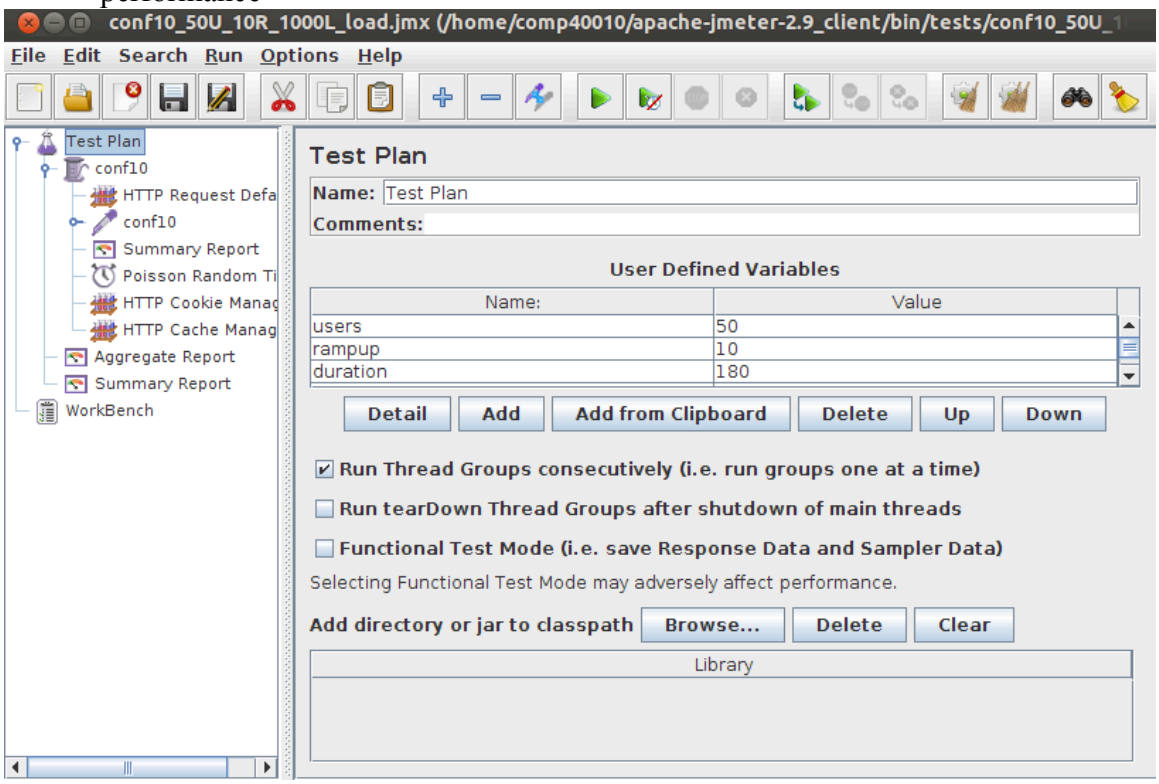


Figure 5: Jmeter test plan.

- Cacti – GUI for RRDtool that logs operating system performance data (CPU, Memory, Processes, Load average, Network in/out) in time series

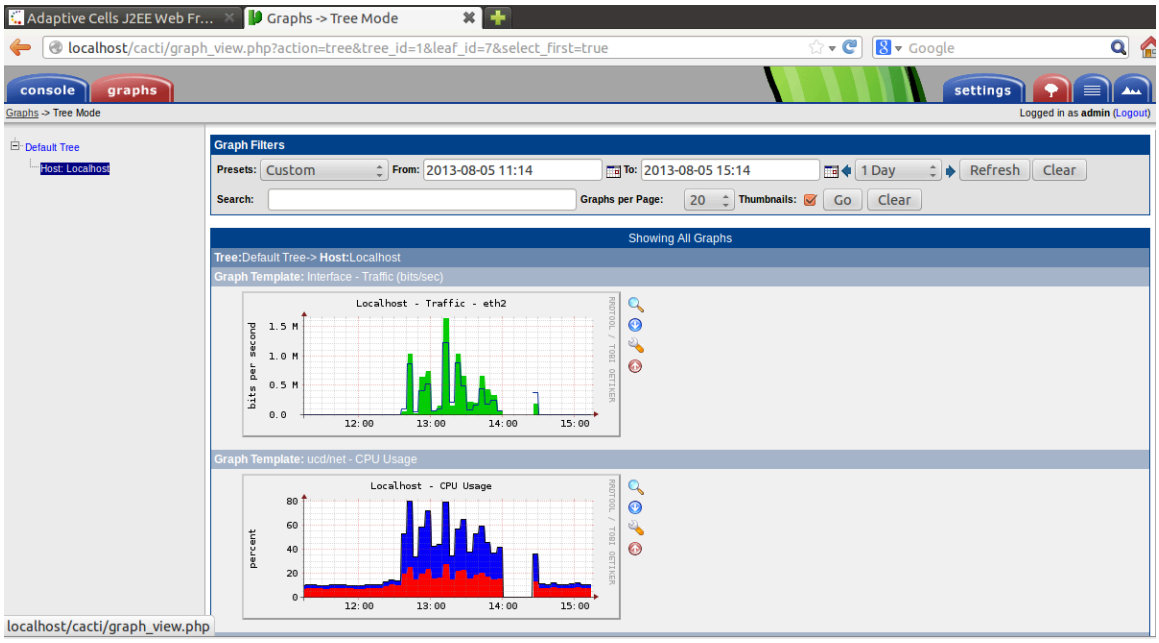


Figure 6: Cacti graphs.

To simulate realistic user load used distributed testing mode in Jmeter. Traffic was spread between 3 client hosts accessing 1 Jboss server.
Visualization of physical and network architecture:

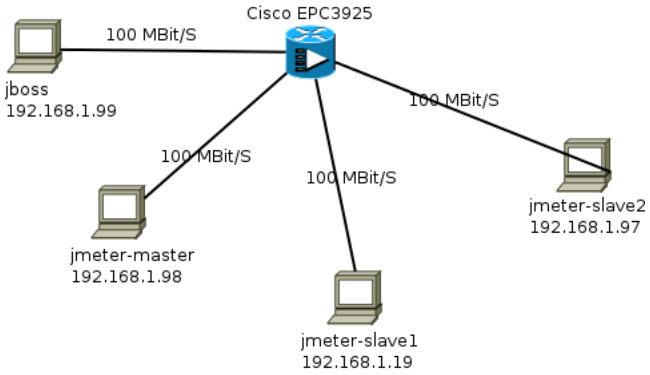


Figure 7: Physical and network architecture of testing infrastructure.

Detailed specification of nodes in table below:

Node	Description
Application server – 192.168.1.99 (jboss)	Jboss 5.0.1 GA 2x Intel Core i7 2.9 GHz, 2GB RAM, 100Mbit/s LAN, Ubuntu 12.04.2 LTC x86
Jmeter server – 192.168.1.19 (jmeter-slave1)	Jmeter 2.9 1x Intel Core i5 2.67 GHz, 1GB RAM, 100Mbit/s LAN, Ubuntu 12.04.2 LTC x86
Jmeter server – 192.168.1.97 (jmeter-slave2)	Jmeter 2.9

	1x Intel Core i5 2.67 GHz, 1GB RAM, 100Mbit/s LAN, Ubuntu 12.04.2 LTC x86
Jmeter client and server – 192.168.1.98 (jmeter-master)	Jmeter 2.9 2x Intel Core i7 2.9 GHz, 2.5GB RAM, 100Mbit/s LAN, Ubuntu 12.04.2 LTC x86
Routing and switching	Cisco EPC3925 Cisco Residential Gateway

Table 2: Testing infrastructure specification

Jmeter distributed architecture used for testing:

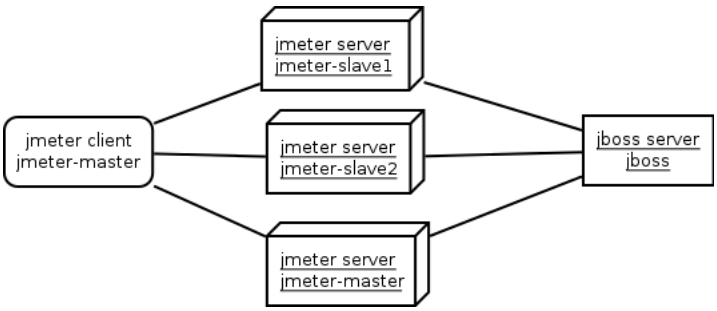


Figure 8: Jmeter testing architecture.

Every configuration was tested using Test Plan design showed on Figure 5. Explanation of test parameters that was used:

- Users – number of threads (concurrent users accessing the system) – 50, 100 and 200
- Ramp-up – period of time to start all specified threads – 10 seconds for average load simulation and 2 seconds for peak load simulation
- Poisson Random Timer – timer that simulates user pauses when accessing the system – this happens due to typing or reading. Timer pauses for minimum 500 milliseconds + random value in 200 milliseconds range. Random value added to respect various speed of typing, reading etc.
- HTTP Cookie and Cache Manager – test elements to simulate real browser behavior – cookie usage and caching. Max number of elements in cache set to 1000.

For traffic visualization video refer to http://bit.ly/1bbGgOv. Used test plan executing all of the configs.

Test plan generations have been automated: http://bit.ly/16z8JNb. Automation script takes a Jmeter test plan as a reference and populates it with different values of number of threads.
Furthermore, test execution, application monitoring, test results gathering and graph generation also have been automated: http://bit.ly/13FNQiU.
Automation script executes following steps for each test plan in specified folder:

- generate a unique named output folder
- start Jboss server
- get Jboss pid and attach Visualvm to this process

- warmup Jboss
- start distributed Jmeter testing
- gather and generate graphs from testing results
- stop Jboss
- gather garbage collection logs for Jboss

Results

Following results documentation includes detailed explanation how results were interpreted for Config 3 with 50 concurrent users. The rest of the configs analysis will include only highlights that differentiate certain results from the others. For all configs analysis procedure looked pretty much the same as for Config 3.

Config 3

Number of concurrent users: 50

During load test of application using 50 concurrent users the median response time was 13 ms. For peak simulation response time increased to 16 ms. A majority of response times were below 100 ms as showed on 'Response times over time' and 'Response time distribution' graphs:

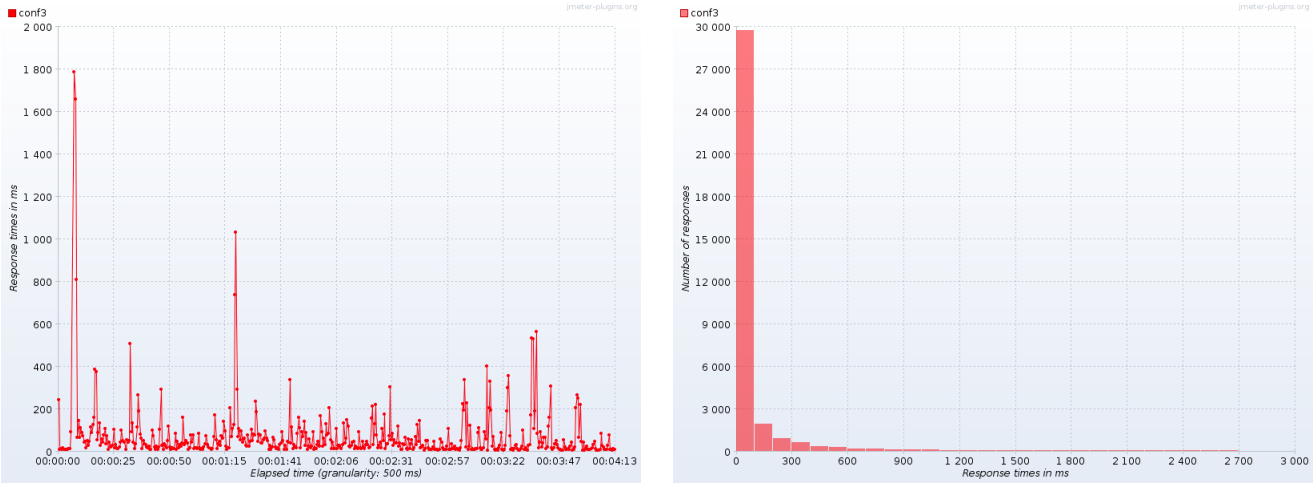


Figure 9: 'Response times over time' and 'Response time distribution' graphs for Config 3 load testing with 50 concurrent users

As far as CPU is concerned, Jboss was using around 40% for load simulation and spiked couple of times up to 60-80% during peak simulation.

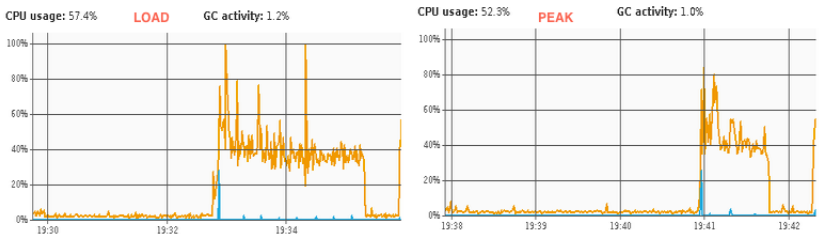


Figure 10: Jboss CPU usage for Config 3 load testing with 50 concurrent users

Operating system overall CPU didn't exceed 30% - with around 10% being used by operating system itself.

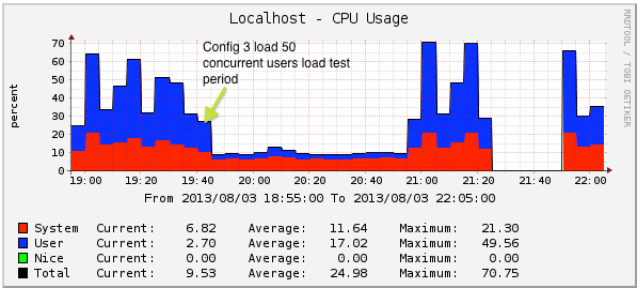


Figure 11: Operating system CPU monitoring for Config 3 load testing with 50 concurrent users

Jboss heap was used in a range from 120MB spiking up to 250 MB during both tests. Graphs show proper garbage collection process where heap comes back to the previous memory usage level:

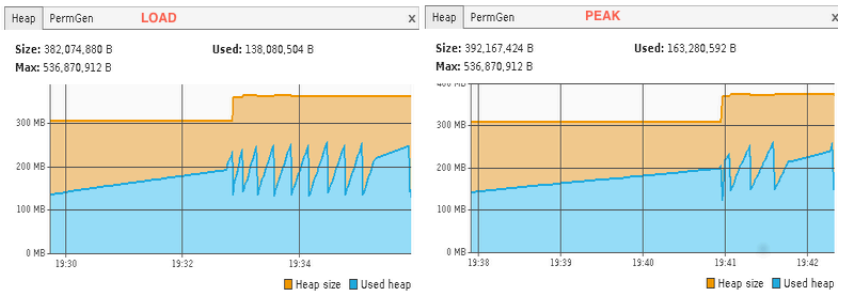


Figure 12: Jboss heap memory usage for Config 3 load testing with 50 concurrent users

Operating system memory had 600 MB free during the test:

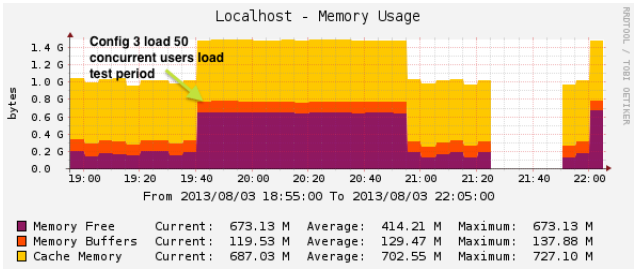


Figure 13: Operating system memory usage for Config 3 load testing with 50 concurrent users

Average throughput for load and peak simulations was 135 and 82 transactions per second respectively. Following graphs represents Throughput distribution over time showed along with hits per second graph during load simulation.

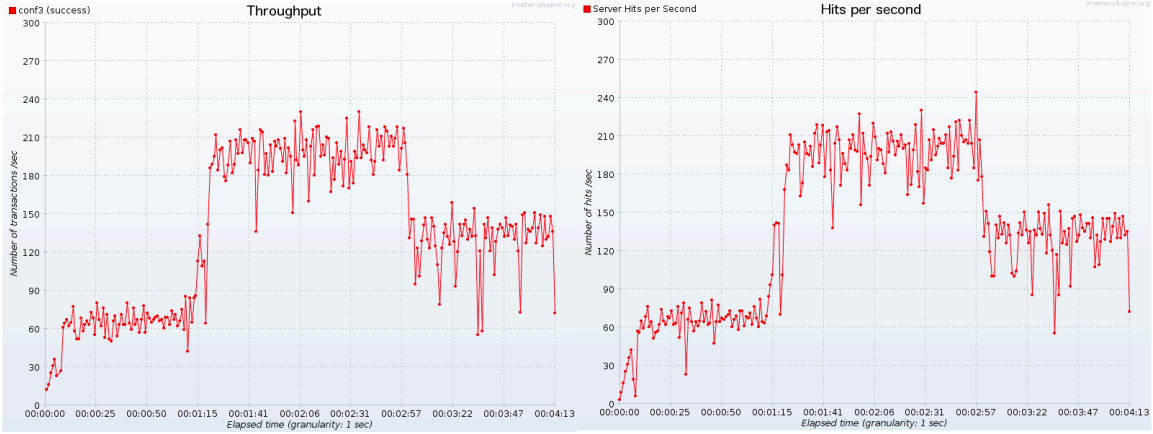


Figure 14: Throughput with hits per second visualization for Config 3 load testing with 50 concurrent users

Network traffic didn't exceed 0.2 Mbit/s for inbound and 0.1 Mbit/s for outbound traffic.

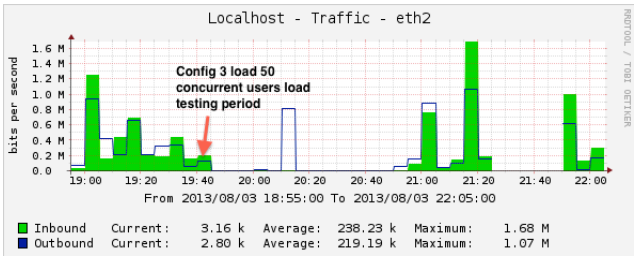


Figure 14: Network traffic for Config 3 load testing with 50 concurrent users

0 erroneous responses logged during both tests.

Number of concurrent users: 100

During load test of application using 100 concurrent users the median response time was 325 ms. For peak simulation, response time increased to 627 ms. Around 35% requests finished below 100 ms.
Jboss CPU was running in 60-80% range spiking few times up to 100% for a tiny piece of 1 sec during garbage collection. OS CPU was running on 50% with around 15% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 200 MB free – this was due to heavy system load before running the test.
Average throughput for load and peak simulations was 157 and 46 transactions per second respectively.
Network traffic didn't exceed 0.1 Mbit/s for inbound and outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 200

During load test of application using 200 concurrent users the median response time was 443 ms. For peak simulation, response time increased to 1.2 sec.
Jboss CPU was running in 60-80% range spiking few times up to 100% for a tiny piece of 1 sec during garbage collection. OS CPU was running on 48% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 200 MB free – this was due to heavy system load before running the test.
Average throughput for load and peak simulations was 148 and 40 transactions per second respectively.
Network traffic didn't exceed 0.48 Mbit/s for inbound and 0.45 Mbit/s for outbound traffic.
0.005% erroneous responses logged for load simulation and 0.026% for peak simulation. Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark:

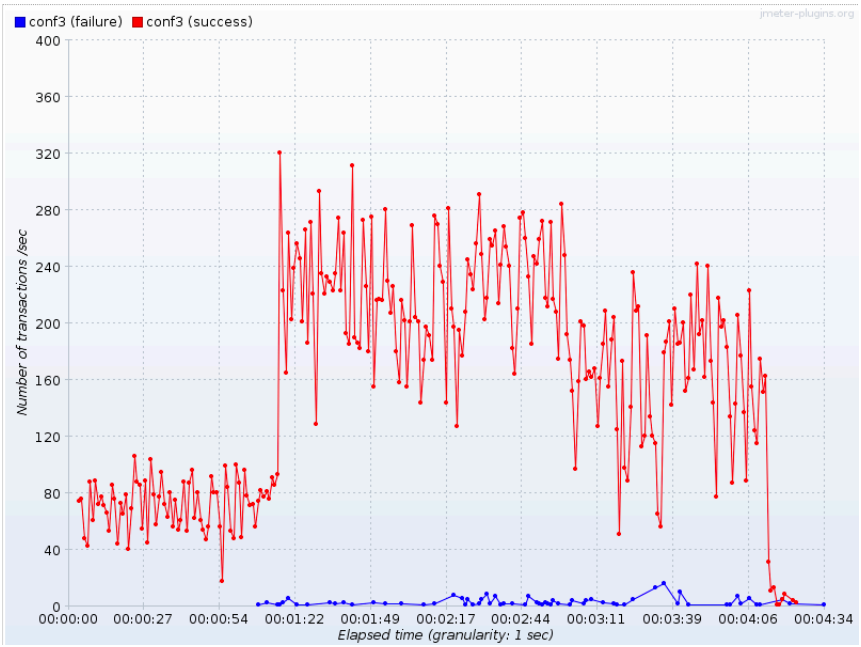


Figure 15: Erroneous responses during Config 3 load tests with 200 concurrent users

Detailed results for Config 3: http://bit.ly/1300Y42
Raw data results for Config 3: http://bit.ly/17x6SFC

Config 5

Number of concurrent users: 50

During load test of application using 50 concurrent users the median response time was 92 ms. For peak simulation, response time increased to 283 ms.
Jboss CPU was running in 40-80% (median 60%) range spiking couple times up to 100% for a tiny piece of 1 sec during garbage collection. OS CPU was running on 50% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 600 MB free.
Average throughput for load and peak simulations was 106 and 50 transactions per second respectively.
Network traffic didn't exceed 0.28 Mbit/s for inbound and 0.19 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 100

During load test of application using 100 concurrent users the median response time was 799 ms. For peak simulation, response time increased to 1.2 secs.
Jboss CPU was running in 60-80% range spiking few times up to 100%. OS CPU was running on 45% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 380 MB free.
Average throughput for load and peak simulations was 106 and 45 transactions per second respectively.
Network traffic didn't exceed 0.31 Mbit/s for inbound and 0.21 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 200

During load test of application using 200 concurrent users the median response time was 663 ms. For peak simulation, response time increased to 1.3 sec.
Jboss CPU was running in 60-80% range spiking couple of times up to 100% for a tiny piece of 1 sec during garbage collection. OS CPU was running on 51% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 190 MB free.
Average throughput for load and peak simulations was 109 and 45 transactions per second respectively.
Network traffic didn't exceed 0.29 Mbit/s for inbound and 0.18 Mbit/s for outbound traffic.
0.009% erroneous responses logged for load simulation and 0.011% for peak simulation. Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Detailed results for Config 5: http://bit.ly/1cvx4Yp
Raw data results for Config 5: http://bit.ly/15anktQ

Config 6

Number of concurrent users: 50

During load test of application using 50 concurrent users the median response time was 6 ms. For peak simulation, response time increased to 10 ms.
Jboss CPU was running in 40% range spiking couple times up to 60% and 80%. OS CPU was running on 41% with around 15% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 220 MB free.
Average throughput for load and peak simulations was 132 and 69 transactions per second respectively.
Network traffic didn't exceed 0.38 Mbit/s for inbound and 0.19 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 100

During load test of application using 100 concurrent users the median response time was 146 ms. For peak simulation, response time increased to 453 ms.
Jboss CPU was running in 60-80% range spiking a couple of times up to 100%. OS CPU was running on 48% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 210 MB free.
Average throughput for load and peak simulations was 182 and 64 transactions per second respectively.
Network traffic didn't exceed 0.57 Mbit/s for inbound and 0.29 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 200

During load test of application using 200 concurrent users the median response time was 367 ms. For peak simulation, response time increased to 541 ms.
Jboss CPU was running in 60% range spiking couple of times up to 100%. OS CPU was running on 51% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 190 MB free.

Average throughput for load and peak simulations was 169 and 49 transactions per second respectively.
Network traffic didn't exceed 0.29 Mbit/s for inbound and 0.18 Mbit/s for outbound traffic.
0.002% erroneous responses logged for load simulation and 0.021% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Detailed results for Config 6: http://bit.ly/14k0GmA
Raw data results for Config 6: http://bit.ly/193mzYA

Config 8

Number of concurrent users: 50

During load test of application using 50 concurrent users the median response time was 30 ms. For peak simulation, response time increased to 45 ms.
Jboss CPU was running in 40-60%. OS CPU was running on 20% with around 10% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 400 MB free.
Average throughput for load and peak simulations was 117 and 56 transactions per second respectively.
Network traffic didn't exceed 0.08 Mbit/s for inbound and 0.06 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 100

During load test of application using 100 concurrent users the median response time was 418 ms. For peak simulation, response time increased to 905 ms.
Jboss CPU was running in 60% range spiking a couple of times up to 100%. OS CPU was running on 48% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 210 MB free.
Average throughput for load and peak simulations was 128 and 52 transactions per second respectively.
Network traffic didn't exceed 0.45 Mbit/s for inbound and 0.21 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 200

During load test of application using 200 concurrent users the median response time was 664 ms. For peak simulation, response time increased to 689 ms.
Jboss CPU was running in 60% range spiking 1 time up to 100%. OS CPU was running on 45% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 210 MB free.
Average throughput for load and peak simulations was 114 and 56 transactions per second respectively.
Network traffic didn't exceed 0.35 Mbit/s for inbound and 0.21 Mbit/s for outbound traffic.
0.5% erroneous responses logged for load simulation and 1% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Detailed results for Config 8: http://bit.ly/1cvLaJq
Raw data results for Config 8: http://bit.ly/19NSeQD

Config 10

Number of concurrent users: 50

During load test of application using 50 concurrent users the median response time was 94 ms. For peak simulation, response time increased to 111 ms.
Jboss CPU was running in 40-60%. OS CPU was running on 45% with around 17% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 220 MB free.
Average throughput for load and peak simulations was 107 and 54 transactions per second respectively.
Network traffic didn't exceed 0.38 Mbit/s for inbound and 0.21 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 100

During load test of application using 100 concurrent users the median response time was 817 ms. For peak simulation, response time increased to 979 ms.
Jboss CPU was running in 60-80% range spiking a couple of times up to 100%. OS CPU was running on 46% with around 18% being consumed by operating system itself.

Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 210 MB free.
Average throughput for load and peak simulations was 113 and 45 transactions per second respectively.
Network traffic didn't exceed 0.38 Mbit/s for inbound and 0.20 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 200

During load test of application using 200 concurrent users the median response time was 830 ms. For peak simulation, response time increased to 1.1 secs.
Jboss CPU was running in 60% range spiking 1 time up to 100%. OS CPU was running on 45% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 190 MB free.
Average throughput for load and peak simulations was 101 and 47 transactions per second respectively.
Network traffic didn't exceed 0.31 Mbit/s for inbound and 0.18 Mbit/s for outbound traffic.
0.9% erroneous responses logged for load simulation and 1.4% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Detailed results for Config 10: http://bit.ly/1ccp6Es
Raw data results for Config 10: http://bit.ly/1bcvuYo

All configs

This test plan is testing all above configs to provide an overall application performance overview.

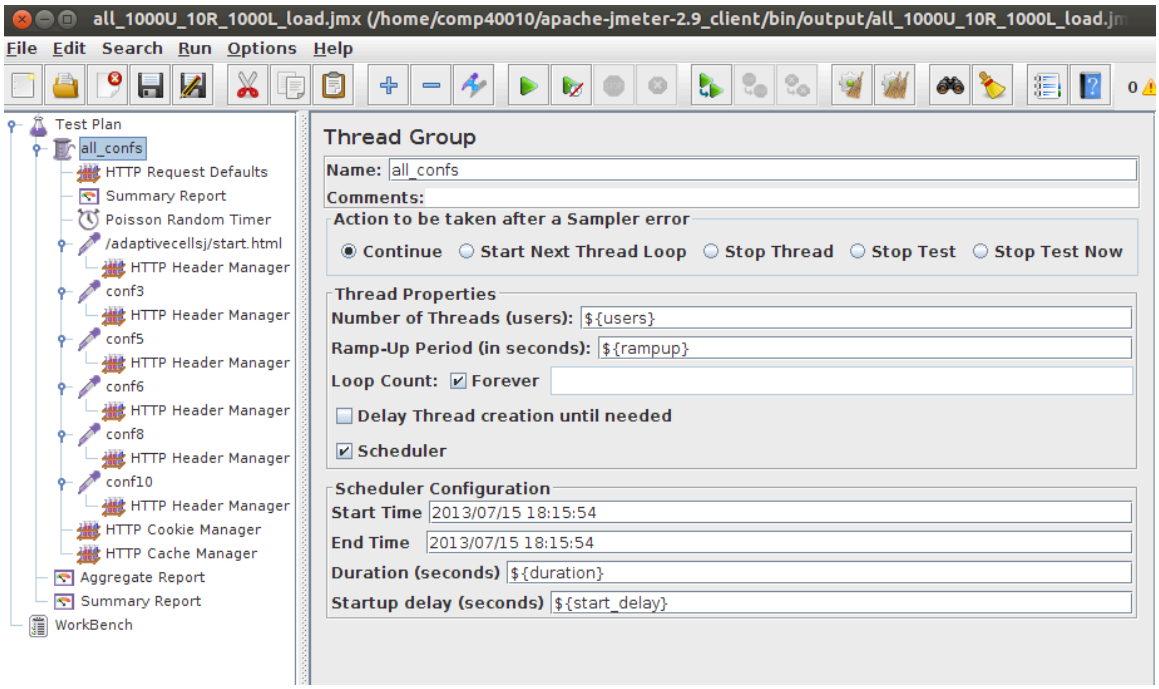


Figure 16: All configs test plan.

Number of concurrent users: 50

During load test of application using 50 concurrent users the median response time was 13 ms. For peak simulation, response time increased to 32 ms.
Jboss CPU was running in 40-60%. OS CPU was running on 45% with around 17% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 220 MB free.
Average throughput for load and peak simulations was 130 and 67 transactions per second respectively.
Network traffic didn't exceed 0.38 Mbit/s for inbound and 0.21 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 100

During load test of application using 100 concurrent users the median response time was 448 ms. For peak simulation, response time increased to 730 ms.
Jboss CPU was running in 60-80% range spiking a couple of times up to 100%. OS CPU was running on 55% with around 20% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had 190 MB free.

Average throughput for load and peak simulations was 133 and 61 transactions per second respectively.
Network traffic didn't exceed 0.45 Mbit/s for inbound and 0.25 Mbit/s for outbound traffic.
0 erroneous responses logged during both tests.

Number of concurrent users: 200

During load test of application using 200 concurrent users the median response time was 554 ms. For peak simulation, response time increased to 970 ms.
Jboss CPU was running in 60% range spiking 1 time up to 100%. OS CPU was running on 45% with around 18% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 112 and 56 transactions per second respectively.
Network traffic didn't exceed 0.38 Mbit/s for inbound and 0.21 Mbit/s for outbound traffic.
0.7% erroneous responses logged for load simulation and 1.3% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Detailed results for all configs: http://bit.ly/17xMx2X
Raw data results for all configs: http://bit.ly/11MC4lF

Summary

Load testing results provided performance characteristic for the system under average load. Testing parameters were tweaked to not exceed good usability and user experience threshold. Based on article by Jakob Nielsen http://www.nngroup.com/articles/response-times-3-important-limits/ (Chapter 5, Usability Engineering, Jakob Nielsen, 1993, http://www.nngroup.com/books/usability-engineering/) the threshold is set by following rules:
- 0.1 ms - user gets instant response from the system
- 1 sec - user notices the delay, however, the flow stays uninterrupted
- 10 secs - is about the limit for keeping the user's attention focused on the dialogue.
1 sec response time has been set as a top value for load tests. Peak simulation scenarios for certain configs have exceeded this threshold sporadically, however, it was really tiny and overall performance (testing all configs) is below 1 sec on peak.
In summary, performance for 50 concurrent users is astonishingly good – system is responding instantaneously – below 50 ms.

Response times for 100 and 200 concurrent users are between 500-1000 ms, which provides reasonable performance and user experience. 200-user scenario started generating marginal erroneous responses in 0.5-1.4% ranges.

3. Performance testing

Approach

Performance testing was focused on overloading the system in order to find its limits, bottlenecks and performance characteristics.
All rules, test design, testing architecture and automation tools are the same as in load testing, however, number of concurrent users has been increased.

Results

Config 3

Number of concurrent users: 500

During load test of application using 500 concurrent users the median response time was 663 ms. For peak simulation, response time increased to 1.1 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 65% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.

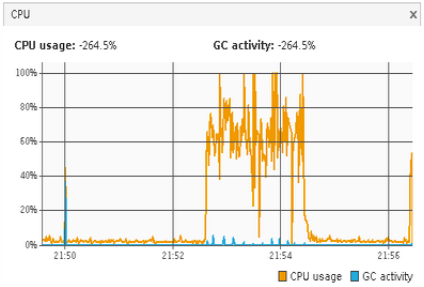


Figure 17: CPU usage during Config 3 performance test using 500 concurrent users

Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.

Average throughput for load and peak simulations was 127 and 76 transactions per second respectively.
Network traffic didn't exceed 1 Mbit/s for inbound and 0.6 Mbit/s for outbound traffic.
3% erroneous responses logged for load simulation and 10% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Number of concurrent users: 1000

During load test of application using 1000 concurrent users the median response time was 752 ms. For peak simulation, response time increased to 1.1 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 65% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 127 and 76 transactions per second respectively.
Network traffic didn't exceed 1.2 Mbit/s for inbound and 0.9 Mbit/s for outbound traffic.
8% erroneous responses logged for load simulation and 10% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

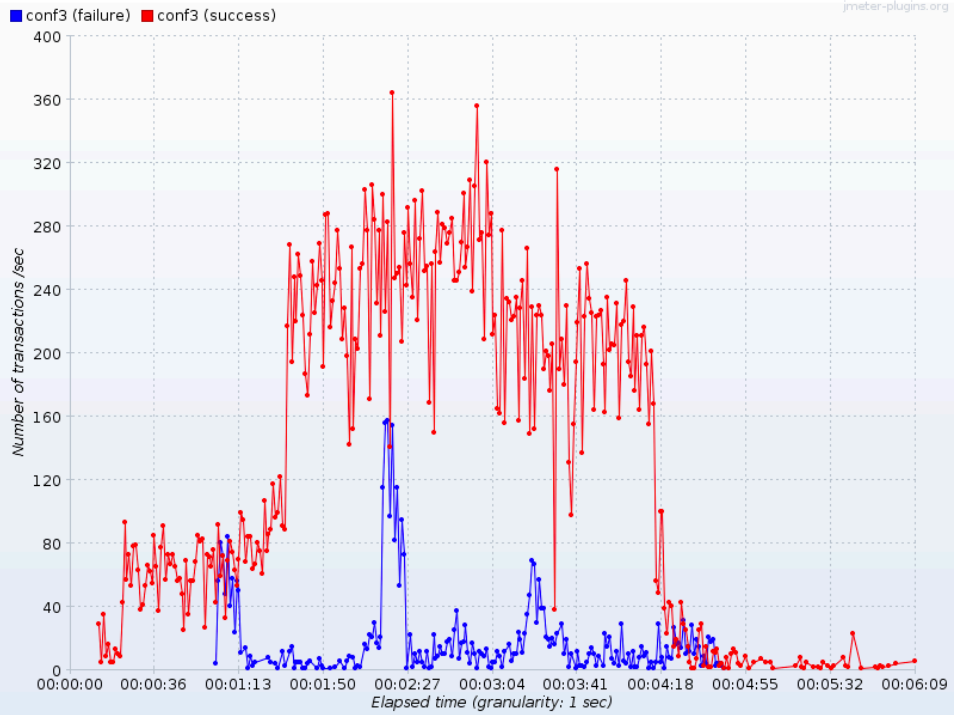


Figure 18: Failure level during Config 3 performance test using 500 concurrent users.

Number of concurrent users: 2000

During load test of application using 2000 concurrent users the median response time was 2 secs. For peak simulation, response time increased to 7 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 70% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 140 and 94 transactions per second respectively.
Network traffic didn't exceed 1.6 Mbit/s for inbound and 1 Mbit/s for outbound traffic.
17% erroneous responses logged for load simulation and 26% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Config 5

Number of concurrent users: 500

During load test of application using 500 concurrent users the median response time was 1.1 secs. For peak simulation, response time increased to 1.7 secs.
Jboss CPU was running in 60% range spiking often up to 100%. OS CPU was running on 70% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.

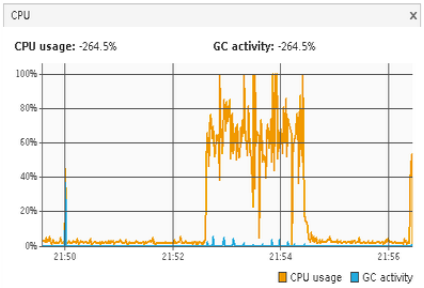


Figure 17: CPU usage during Config 3 performance test using 500 concurrent users

Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 98 and 40 transactions per second respectively.
Network traffic didn't exceed 0.6 Mbit/s for inbound and 0.4 Mbit/s for outbound traffic.
7% erroneous responses logged for load simulation and 11% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Number of concurrent users: 1000

During load test of application using 1000 concurrent users the median response time was 2 secs. For peak simulation, response time increased to 4.3 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 75% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 92 and 76 transactions per second respectively.
Network traffic didn't exceed 0.9 Mbit/s for inbound and 0.4 Mbit/s for outbound traffic.
17% erroneous responses logged for load simulation and 20% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

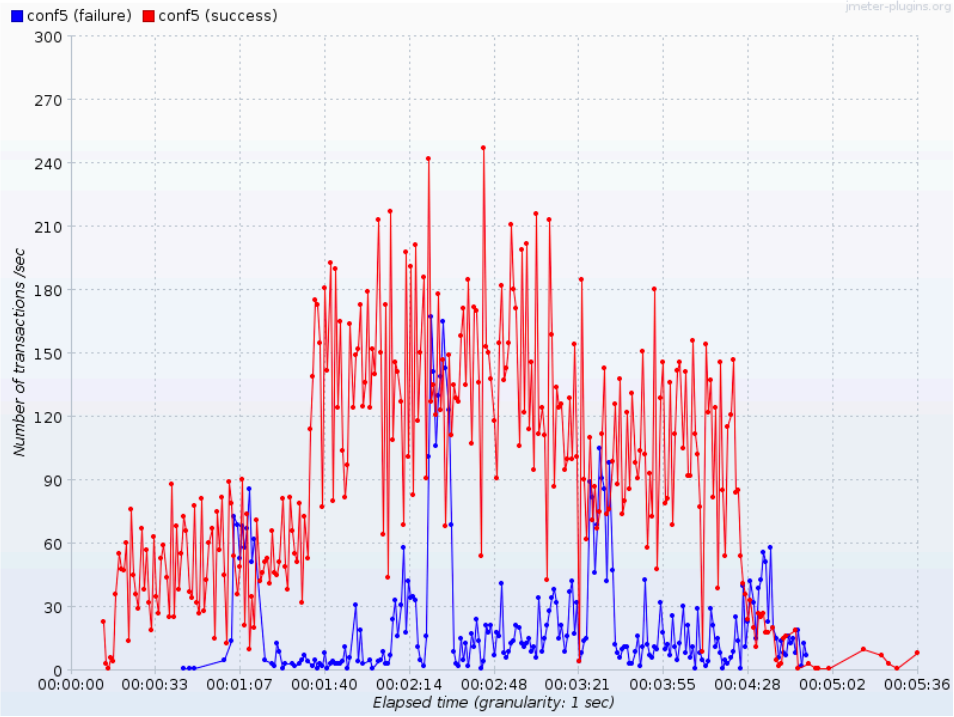


Figure 18: Failure level during Config 3 performance test using 500 concurrent users.

Number of concurrent users: 2000

During load test of application using 2000 concurrent users the median response time was 6 secs. For peak simulation, response time increased to 15 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 80% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.

Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 99 and 60 transactions per second respectively.
Network traffic didn't exceed 1.2 Mbit/s for inbound and 0.5 Mbit/s for outbound traffic.
27% erroneous responses logged for load simulation and 20% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Config 6

Number of concurrent users: 500

During load test of application using 500 concurrent users the median response time was 484 ms. For peak simulation, response time increased to 909 ms.
Jboss CPU was running in 60-80% range. OS CPU was running on 40% with around 15% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 141 and 62 transactions per second respectively.
Network traffic didn't exceed 0.3 Mbit/s for inbound and 0.2 Mbit/s for outbound traffic.
2% erroneous responses logged for load simulation and 7% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Number of concurrent users: 1000

During load test of application using 1000 concurrent users the median response time was 673 ms. For peak simulation, response time increased to 1.8 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 45% with around 15% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 128 and 59 transactions per second respectively.
Network traffic didn't exceed 0.9 Mbit/s for inbound and 0.6 Mbit/s for outbound traffic.
8% erroneous responses logged for load simulation and 21% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Number of concurrent users: 2000

During load test of application using 2000 concurrent users the median response time was 1.6 secs. For peak simulation, response time increased to 4 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 50% with around 20% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 148 and 82 transactions per second respectively.
Network traffic didn't exceed 1.2 Mbit/s for inbound and 0.7 Mbit/s for outbound traffic.
15% erroneous responses logged for load simulation and 8% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Config 8

Number of concurrent users: 500

During load test of application using 500 concurrent users the median response time was 634 ms. For peak simulation, response time increased to 1.2 sec.
Jboss CPU was running in 40-60% range. OS CPU was running on 40% with around 15% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 114 and 52 transactions per second respectively.
Network traffic didn't exceed 0.2 Mbit/s for inbound and 0.1 Mbit/s for outbound traffic.
5% erroneous responses logged for load simulation and 8% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Number of concurrent users: 1000

During load test of application using 1000 concurrent users the median response time was 1.1 secs. For peak simulation, response time increased to 2.8 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 60% with around 15% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.

Average throughput for load and peak simulations was 128 and 48 transactions per second respectively.
Network traffic didn't exceed 0.6 Mbit/s for inbound and 0.4 Mbit/s for outbound traffic.
13% erroneous responses logged for load simulation and 13.5% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Number of concurrent users: 2000

During load test of application using 2000 concurrent users the median response time was 3 secs. For peak simulation, response time increased to 15 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 50% with around 20% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 112 and 71 transactions per second respectively.
Network traffic didn't exceed 0.6 Mbit/s for inbound and 0.4 Mbit/s for outbound traffic.
20% erroneous responses logged for load simulation and 35% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Config 10

Number of concurrent users: 500

During load test of application using 500 concurrent users the median response time was 907 ms. For peak simulation, response time increased to 1.9 secs.
Jboss CPU was running in 40-60% range. OS CPU was running on 50% with around 15% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 89 and 42 transactions per second respectively.
Network traffic didn't exceed 0.2 Mbit/s for inbound and 0.1 Mbit/s for outbound traffic.
7% erroneous responses logged for load simulation and 11% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Number of concurrent users: 1000

During load test of application using 1000 concurrent users the median response time was 1.3 secs. For peak simulation, response time increased to 2.5 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 50% with around 15% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 94 and 60 transactions per second respectively.
Network traffic didn't exceed 0.4 Mbit/s for inbound and 0.2 Mbit/s for outbound traffic.
12% erroneous responses logged for load simulation and 18% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Number of concurrent users: 2000

During load test of application using 2000 concurrent users the median response time was 7.6 secs. For peak simulation, response time increased to 63 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 65% with around 20% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 104 and 50 transactions per second respectively.
Network traffic didn't exceed 0.6 Mbit/s for inbound and 0.4 Mbit/s for outbound traffic.
28% erroneous responses logged for load simulation and 54% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

All configs

Number of concurrent users: 500

During load test of application using 500 concurrent users the median response time was 575 ms. For peak simulation, response time increased to 1.9 secs.
Jboss CPU was running in 40-60% range. OS CPU was running on 40% with around 15% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 121 and 41 transactions per second respectively.
Network traffic didn't exceed 0.2 Mbit/s for inbound and 0.1 Mbit/s for outbound traffic.
3% erroneous responses logged for load simulation and 12% for peak simulation.

Erroneous responses started occurring when throughput exceeded around 100 transactions per second mark.

Number of concurrent users: 1000

During load test of application using 1000 concurrent users the median response time was 1.4 secs. For peak simulation, response time increased to 2.6 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 70% with around 20% being consumed by operating system itself. CPU levels are hitting warning levels.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 114 and 57 transactions per second respectively.
Network traffic didn't exceed 0.8 Mbit/s for inbound and 0.5 Mbit/s for outbound traffic.
13% erroneous responses logged for load simulation and 19% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Number of concurrent users: 2000

During load test of application using 2000 concurrent users the median response time was 1.3 secs. For peak simulation, response time increased to 63 secs.
Jboss CPU was running in 60% range spiking few times up to 100% for couple of seconds. OS CPU was running on 75% with around 20% being consumed by operating system itself.
Jboss heap memory usage stayed between 120-260 MB. Operating system memory had only 220 MB free.
Average throughput for load and peak simulations was 85 and 74 transactions per second respectively.
Network traffic didn't exceed 1.4 Mbit/s for inbound and 0.6 Mbit/s for outbound traffic.
11% erroneous responses logged for load simulation and 21% for peak simulation.
Erroneous responses started occurring when throughput exceeded around 60 transactions per second mark.

Garbage collection log analysis

After looking at the summary of the GC logs in below table it is clear that the throughput is almost constant. Although the pauses increase with the number of users accessing the system the throughput has been consistent at 98-99 % at an average for all the configurations, which proves that the application is working reliably under load. There are very few pauses on average of 32 and pause time of 4.10s at average for most of the configurations that can be reduced with optimization

Configuration	Users	Footprint	Freed Memory	Freed Mem/Min	Acc pauses	Throughput	Number of full gc pauses	Full GC Performance	Number of gc pauses	GC Performance
Config3										
	50	351.1 M	2517.8 M	465.786 M/m	3.64s	98.88%	3	13.2 M/s	25	1256.7 M/s
	100	349.5 M	3506.5 M	524.443 M/min	4.47s	98.89%	3	11.5 M/s	33	1312.9 M/s
	200	350.9 M	3391.2 M	494.429 M/min	3.61s	99.12%	3	16 M/s	32	1524.1 M/s
	500	321.7 M	3609.6 M	129.421 M/min	3.65s	99.78%	3	12.9M/s	34	1560.4 M/s
	1000	360.9 M	3712.9 M	446.814 M/min	3.64s	99.27%	3	15.2M/s	34	1679 M/s
	2000	341.8 M	3661.4 M	434.825 M/min	4.21s	99.17%	3	12.9M/s	34	1458.8 M/s
Config5										
	50	367.7 M	3284.8 M	486.136 M/m	4.72s	98.84%	3	10.8 M/s	32	1274.6 M/s
	100	349.4 M	3424.8 M	509.596 M/m	4.54s	98.87%	3	10.6 M/s	32	1340.7 M/s
	200	348.4 M	3543.5 M	509.376 M /m	3.66s	99.12%	3	15.3 M/s	33	1556.8 M/s
	500	355.2 M	3390.8 M	477.428 M/m	3.85s	99.10%	3	14 M/s	32	1436.1 M/s
	1000	354.1 M	3366.2 M	425.138 M/m	4.6s	99.03%	3	12.8 M/s	32	1163.1 M/s
	2000	355.2 M	3540.4 M	403.852 M/m	5.13s	99.02%	3	11 M/s	33	1104.1 M/s
Config6										
	50	348.5 M	3304.7 M	451.649 M/m	3.43s	99.22%	3	16.2 M/s	31	1547.2 M/s
	100	353.6 M	3549.8 M	540.092 M/m	3.87s	99.02%	3	13.3 M/s	33	1570.6 M/s
	200	353.4 M	3669.8 M	528.276 M/m	4.46s	98.93%	3	11.5 M/s	34	1393.2 M/s
	500	364.1 M	3769.9 M	134.342 M/m	4.15s	99.75%	3	14.1 M/s	37	1551.2 M/s
	1000	356.4 M	3678.8 M	438.763 M/m	4.6s	99.09%	3	11.3 M/s	34	1343.3 M/s
	2000	348.6 M	3934.6 M	464.343 M/m	4.71s	99.07%	3	11.3 M/s	36	1382.3 M/s
Config8										
	50	350.1 M	3296.2 M	469.986 M/m	4.36s	98.96%	3	10.8 M/s	31	1407.4 M/s
	100	344.3 M	3406.3 M	530.515 M/m	3.78s	99.02%	3	14.8 M/s	32	1452.4 M/s
	200	371.5 M	3459.9 M	493.05 M/m	3.76s	99.11%	3	16.4 M/s	34	1454.8 M/s
	500	355.9 M	3530.9 M	478.576 M/m	4.48s	98.99%	3	12 M/s	33	1308 M/s
	1000	359.6 M	3553.6 M	428.797 M/m	3.6s	99.28%	3	15.5 M/s	33	1651.4 M/s
	2000	368.2 M	3642.5 M	432.306 M/m	4.36s	99.14%	3	11.4 M/s	35	1449.3 M/s
Config 10										
	50	350.7 M	3287.9 M	479.381 M/m	3.36s	99.18%	3	16.4 M/s	31	1605.7 M/s

	100	343.9 M	3394.4 M	510.699 M/m	3.88s	99.03%	3	12.7 M/s	32	1580.1 M/s
	200	350.4 M	3548.3 M	515.224 M/m	3.65s	99.12%	3	13.8 M/s	33	1677.7 M/s
	500	359.3 M	3392.2 M	476.631 M/m	3.96s	99.07%	3	12.3 M/s	32	1511.4 M/s
	1000	361.4 M	3716.6 M	441.5 M/m	4.48s	99.11%	3	10.3 M/s	34	1550.9 M/s
	2000	353.6 M	3516.7 M	123.3 M/m	4.32s	99.75%	3	12.4 M/s	33	1338.9 M/s

We have attached below some screenshots from GC Viewer for different loads for only Config3. Note: Only Config3 screenshots have been added. Please refer to Config3 in above table for below screenshots

Scenario: Load test for 50 Users:

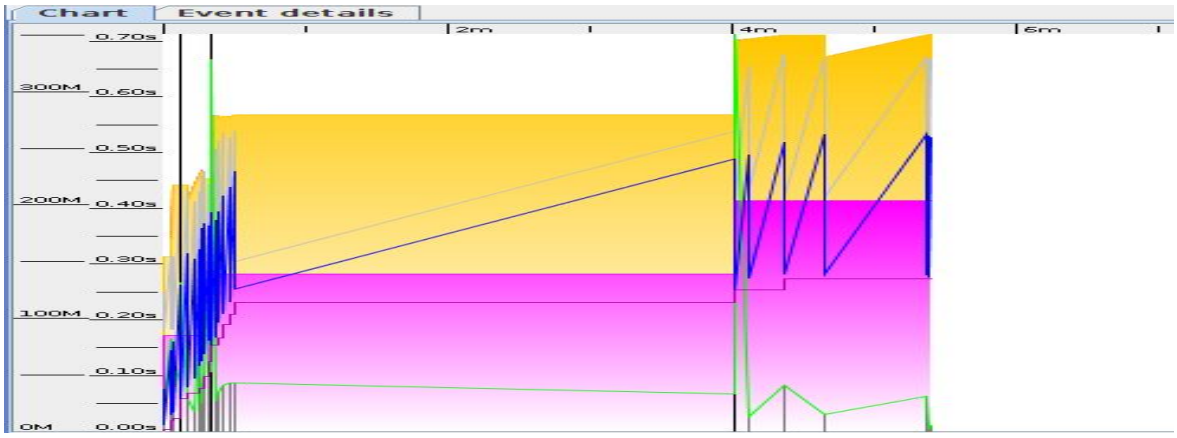


Figure 1 : GC Graph

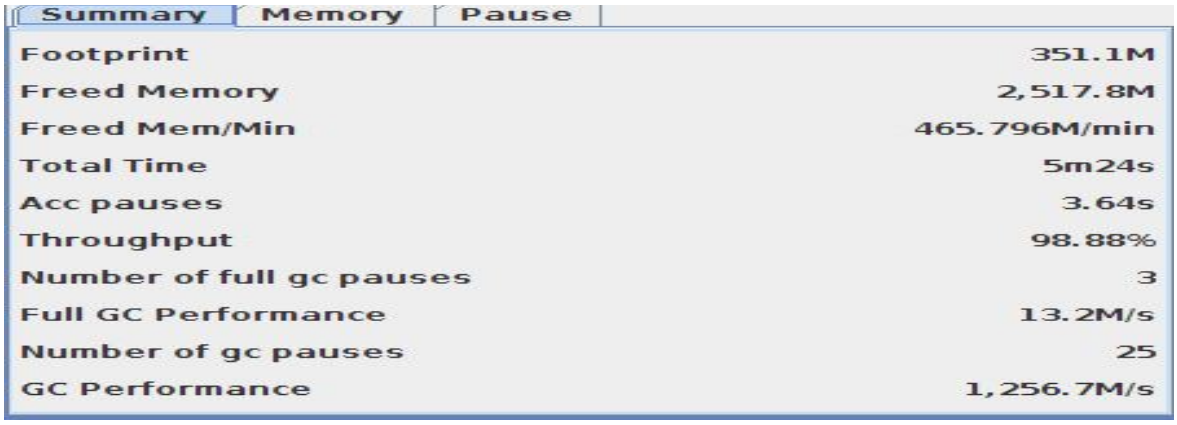


Figure 2: Summary

Scenario: Ran Load test for 100 Users

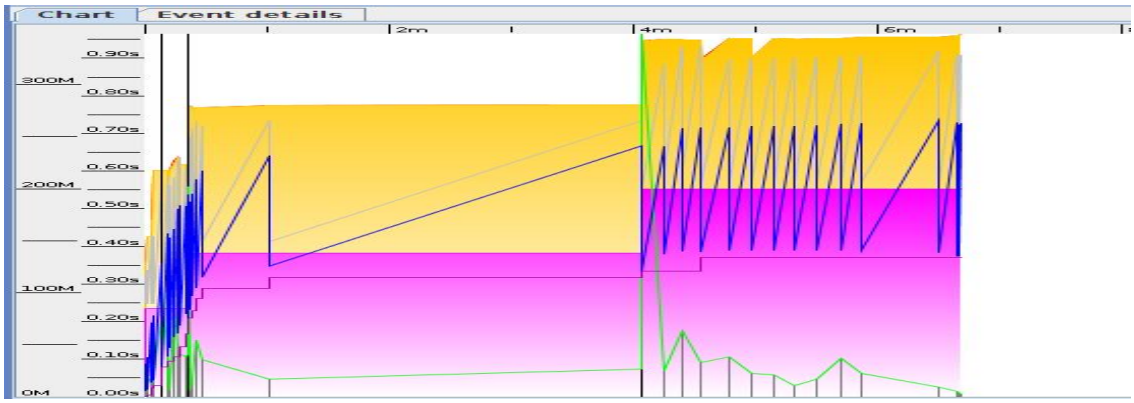


Figure 3 : GC Graph

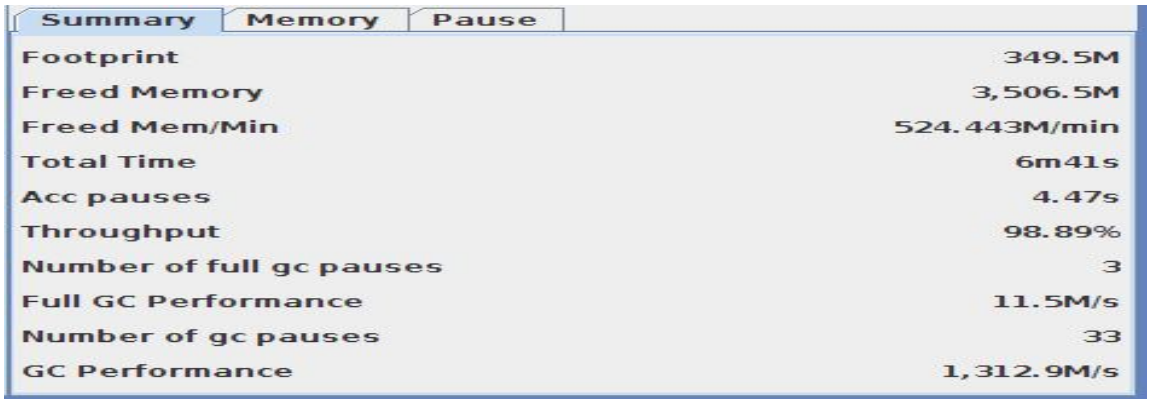


Figure 4 : Summary

Scenario: Ran Load test for 200 Users

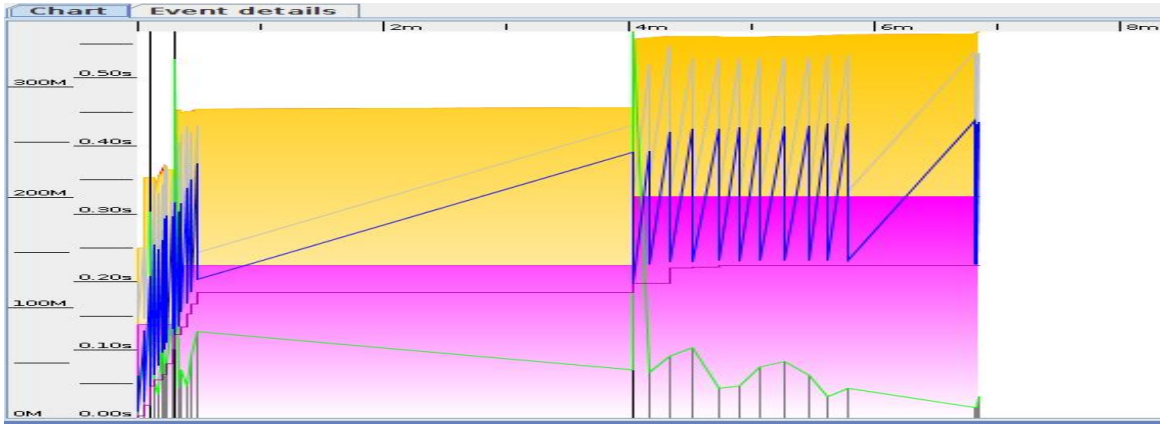


Figure 5 : GC Graph

Summary	Memory	Pause
Footprint		350.9M
Freed Memory		3,391.2M
Freed Mem/Min		494.429M/min
Total Time		6m51s
Acc pauses		3.61s
Throughput		99.12%
Number of full gc pauses		3
Full GC Performance		16M/s
Number of gc pauses		32
GC Performance		1,524.1M/s

Figure 6 : Summary

Scenario: Ran Load test for 500 Users

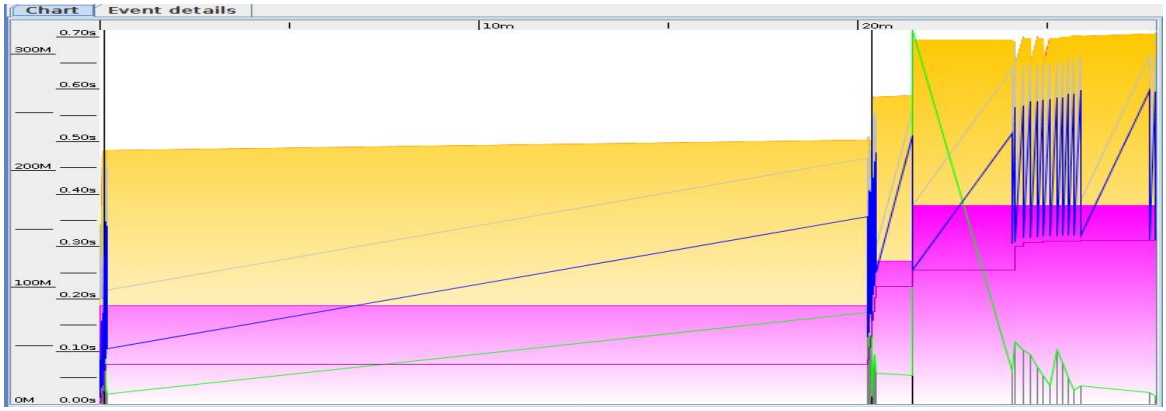


Figure 7 : GC Graph

Summary	Memory	Pause
Footprint		321.7M
Freed Memory		3,609.6M
Freed Mem/Min		129.421M/min
Total Time		27m53s
Acc pauses		3.65s
Throughput		99.78%
Number of full gc pauses		3
Full GC Performance		12.9M/s
Number of gc pauses		34
GC Performance		1,560.4M/s

Figure 8 : Summary

Scenario: Ran Load test for 1000 Users

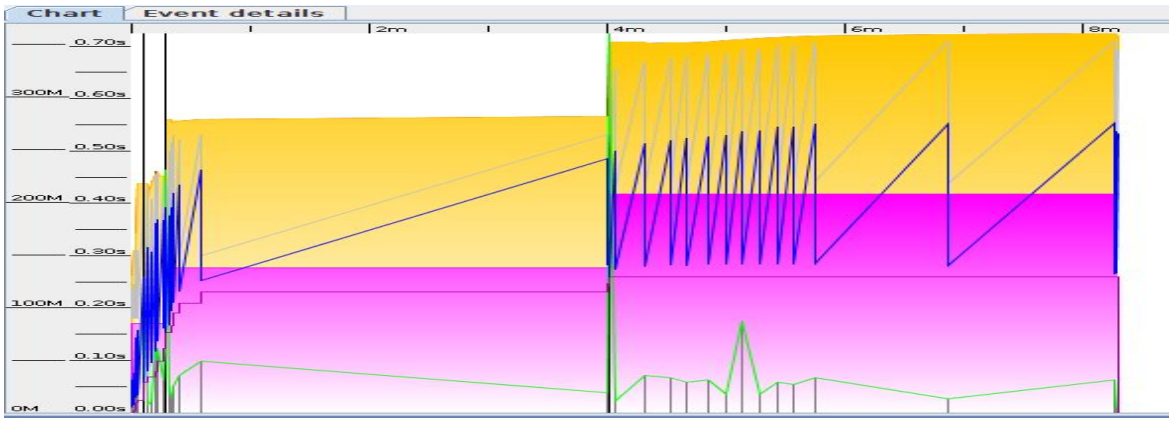


Figure 9 : GC Graph

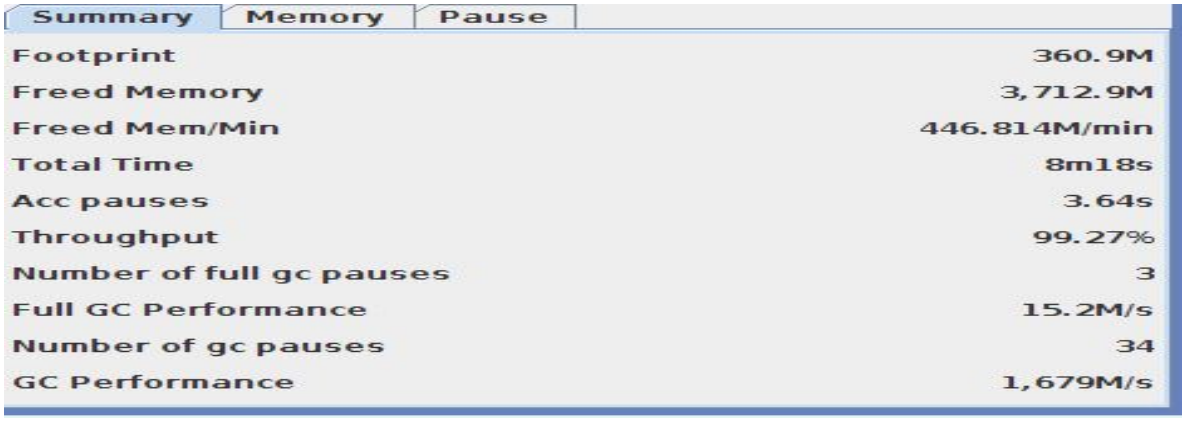


Figure 10 : Summary

Scenario: Ran Load test for 2000 Users

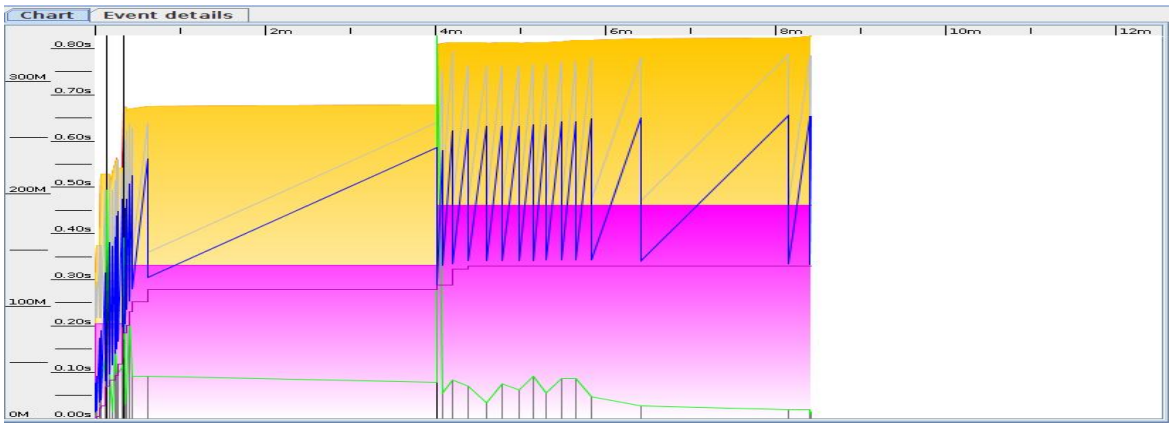


Figure 11 : GC Graph

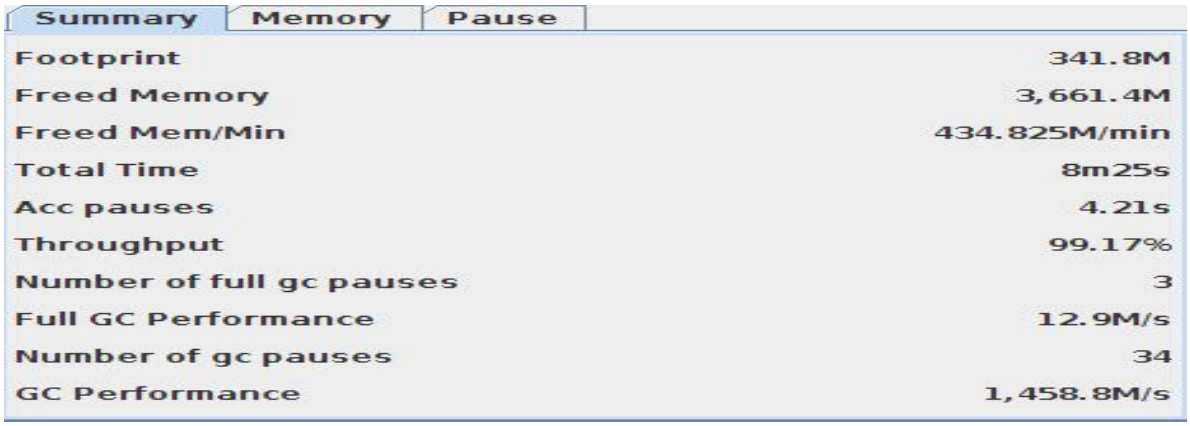


Figure 12: Summary

Summary

Performance tests indicated CPU as a bottleneck. Usage on 60-70% on operating system level occurred to have high impact on generating errors. Every other parameter – network, memory – was utilized only in a small part.
Tests with 500 users for Config 6 and 8 may indicate system responded in an acceptable time – up to 1 sec. However, it depends how much system should be sensitive for errors. Response error level was in 5-8% range, which is much higher than results from load testing.
Results from 1000 and 2000 concurrent users testing exceeded 1 sec mark response time and returned errors in double-digit percentage values.

Performance optimization

Approach

Following official Red Hat tuning guide for Jboss (https://access.redhat.com/site/documentation//en-US/JBoss_Enterprise_Application_Platform/5/html/Performance_Tuning_Guide/index.html) picked a number of optimizations related to tested system:

- set –Xms and –Xmx values to the same value – 512 MB - to avoid sizing decision from the virtual machine
- changed frequency of distributed garbage collection running for RMI system that keeps track of access requests to remote objects running on the server. By default it runs every minute. Set parameter: -Dsun.rmi.dgc.client.gcInterval=36000000

- turned on parallel garbage collection to make use of 2 CPU cores. Parameter: -XX:+UseParallelGC
- changed heap ratio for suggested one – the young and tenured generation set to 1:2. Young generation heap size has been also set to 1/3 of all heap – 170 MB. Parameters: -XX:NewSize=170m -XX:NewRatio=2
- enlarged Jboss thread pool size defined in $JBOSS_DIR/conf/jboss-service.xml from 10 to 50
- removed unnecessary lo4j logging – disabled Console logging and bumped up logging level up to ERROR
- changed ThreadLocalPool and StrictMaxPool for stateless session beans from 15 and 30 up to 100
- disabled Hot deployment Jboss feature

To apply all the changes created automation script: http://bit.ly/19OrQ96
All changes have been applied and 'All configs' performance test has been rerun to compare the results.

Results

All configs: Number of concurrent users: 2000

Median response time after tuning – 1.7 secs during load, 3.7 secs during peak.
Compared to previous results – decrease during load by 24%, improvement during peak by 1700%.
Erroneous responses: load 9%, peak 3%. Improvement during load by 31%, improvement during peak by 700%.

All configs: Number of concurrent users: 1000

Median response time after tuning – 1.5 secs during load, 1.6 secs during peak.
Compared to previous results – decrease during load by 10%, improvement during peak by 16%.
Erroneous responses: load 13%, peak 5%. Improvement during load by 20%, improvement during peak by 633%.

All configs: Number of concurrent users: 500

Median response time after tuning – 459 ms during load, 1.4 secs during peak. Compared to previous results – improvement during load by 50%, improvement during peak by 35%.
Erroneous responses: load 3%, peak 5%. The same value for load, improvement during peak by 240%.

Raw data results: http://bit.ly/1etB1ti

GC Log Analysis after Performance Tuning

After comparing the GC logs of server before and after tuning it shows after optimising the server the throughput has improved. Also the full GC pauses have reduced from 3 to 1 pause for each test. The duration for each pause has reduced as well.

Configuration	Users	Footprint	Freed Memory	Freed Mem/Min	Acc pauses	Throughput	Number of full gc pauses	Full GC Performance	Number of gc pauses	GC Performance
Before Tuning	50	343.8 M	3161.4 M	439.575 M/m	3.27s	99.24%	3	17.1 M/s	30	1537.8 M/s
	100	306.8 M	3409.4 M	490.551 M/m	5.5s	98.68%	3	27.3 M/s	36	986 M/s
	200	349.8 M	3405.5 M	442.835 M/m	4.37s	99.05%	3	14.2 M/s	32	1183.6 M/s
	500	350.1 M	3534.5 M	450.246 M/m	5.44s	98.85%	3	9357.7 K/s	33	1143.8 M/s
	1000	355.4 M	3281.2 M	118.493 M/m	4.37s	99.74%	3	12.1 M/s	31	1248.7 M/s
	2000	359.1 M	3497.5 M	352.041 M/m	5.09s	99.15%	3	9768 K/s	33	1226.1 M/s
After tuning	50	490.7 m	3163.3 M	442.55 M/m	2.73s	99.36%	1	782.5 K/s	31	1303.5 M/s
	100	490.7 M	3269.4 M	503.06 M/m	2.88s	99.26%	1	733.6 K/s	32	1275.7 M/s
	200	490.7 M	3407.8 M	480.597 M/m	3.37s	99.21%	1	647.2 K/s	33	1137.7 M/s
	500	490.7 M	3416.6 M	414.337 M/m	3.21s	99.35%	1	480.6K/s	33	1251 M/s
	1000	490.7 M	2061.9 M	2321.72 M/m	2.61s	95.10%	1	876 K/s	21	877.1 M/s
	2000	490.7 M	3239.2 M	375.418 M/m	3.32s	99.36%	1	493.1 /s	32	1205.5 M/s

We have attached below some screenshots from GC Viewer for different loads for all configurations.

Scenario: Performance test for 50 Users before tuning

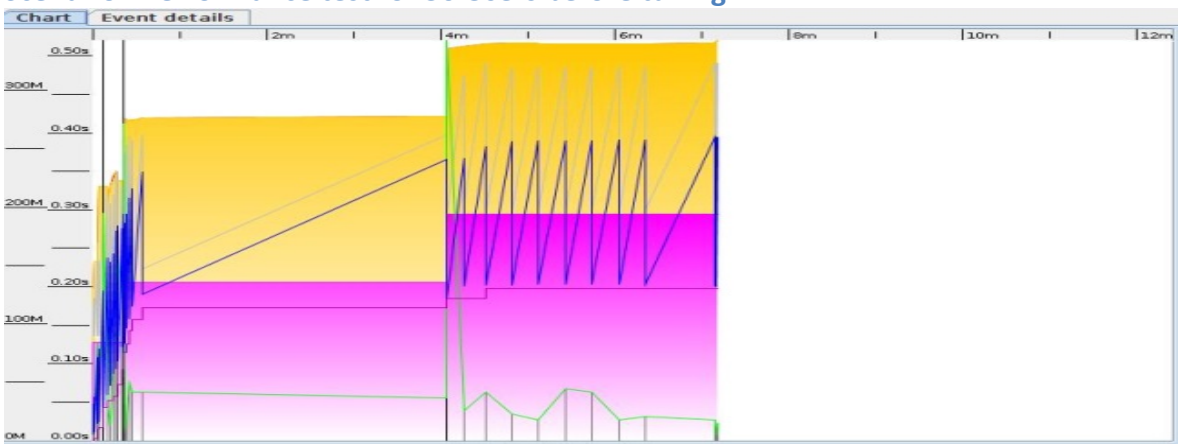
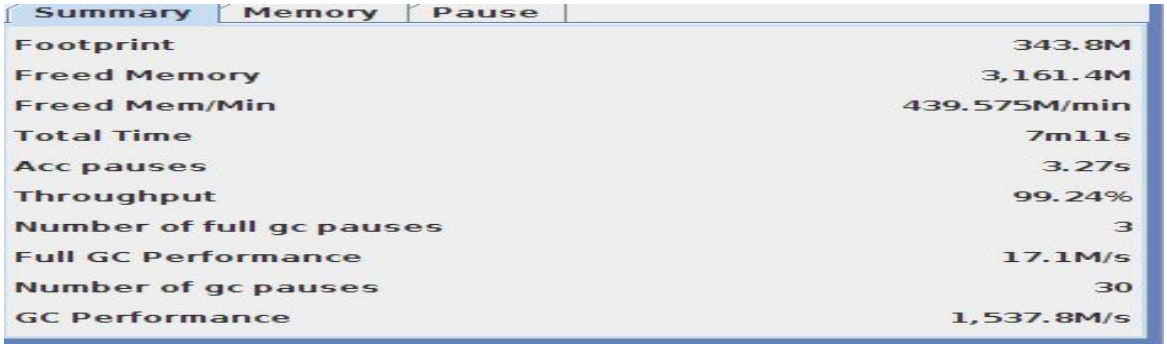


Figure 13 : GC Graph



Summary	Memory	Pause
Footprint		343.8M
Freed Memory		3,161.4M
Freed Mem/Min		439.575M/min
Total Time		7m11s
Acc pauses		3.27s
Throughput		99.24%
Number of full gc pauses		3
Full GC Performance		17.1M/s
Number of gc pauses		30
GC Performance		1,537.8M/s

Figure 14 : Summary

Scenario: Performance test for 100 Users before tuning:

Figure 15 : GC Graph

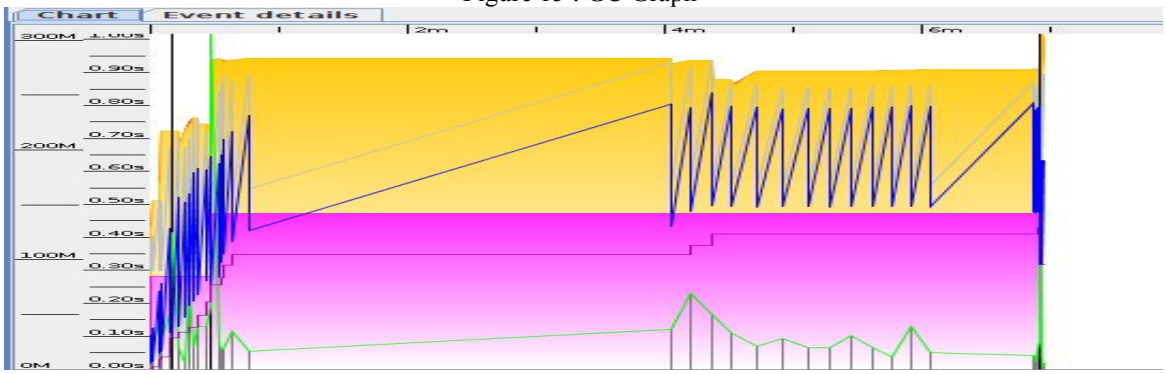
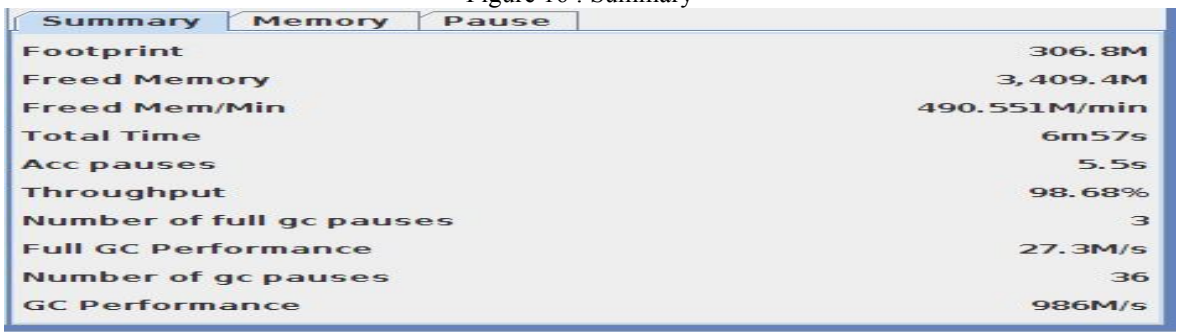


Figure 16 : Summary



Summary	Memory	Pause
Footprint		306.8M
Freed Memory		3,409.4M
Freed Mem/Min		490.551M/min
Total Time		6m57s
Acc pauses		5.5s
Throughput		98.68%
Number of full gc pauses		3
Full GC Performance		27.3M/s
Number of gc pauses		36
GC Performance		986M/s

Scenario: Performance test for 200 Users before tuning:

Figure 17 : GC Graph

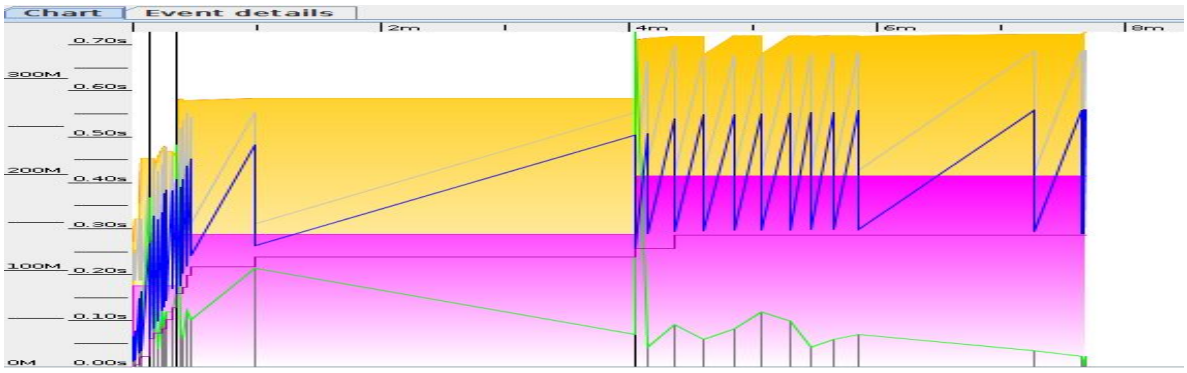


Figure 18 : Summary

Summary	Memory	Pause
Footprint		349.8M
Freed Memory		3,405.5M
Freed Mem/Min		442.835M/min
Total Time		7m41s
Acc pauses		4.37s
Throughput		99.05%
Number of full gc pauses		3
Full GC Performance		14.2M/s
Number of gc pauses		32
GC Performance		1,183.6M/s

Scenario: Performance test for 500 Users before tuning:

Figure 19 : GC Graph

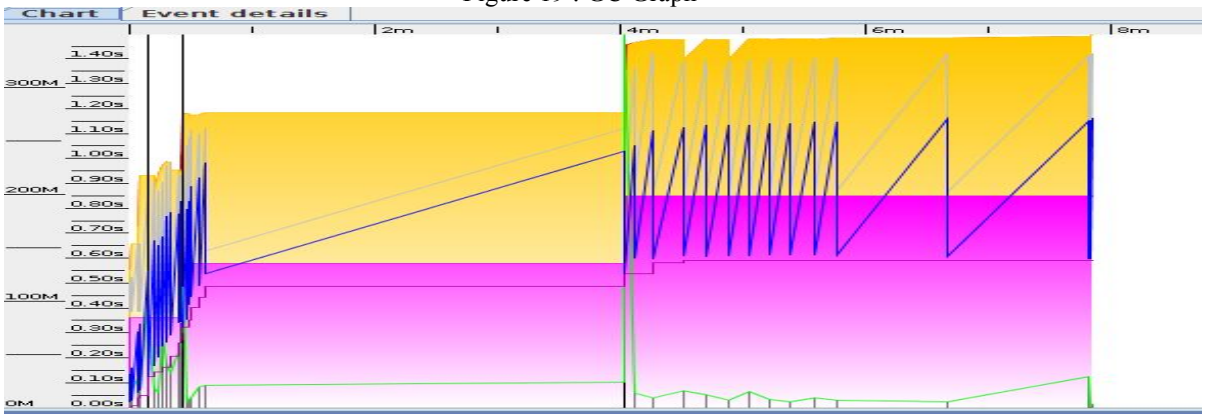


Figure 20 : Summary

Summary	Memory	Pause
Footprint		350.1M
Freed Memory		3,534.5M
Freed Mem/Min		450.246M/min
Total Time		7m51s
Acc pauses		5.44s
Throughput		98.85%
Number of full gc pauses		3
Full GC Performance		9,357.7K/s
Number of gc pauses		33
GC Performance		1,143.8M/s

Scenario: Performance test for 1000 Users before tuning:

Figure 21 : GC Graph

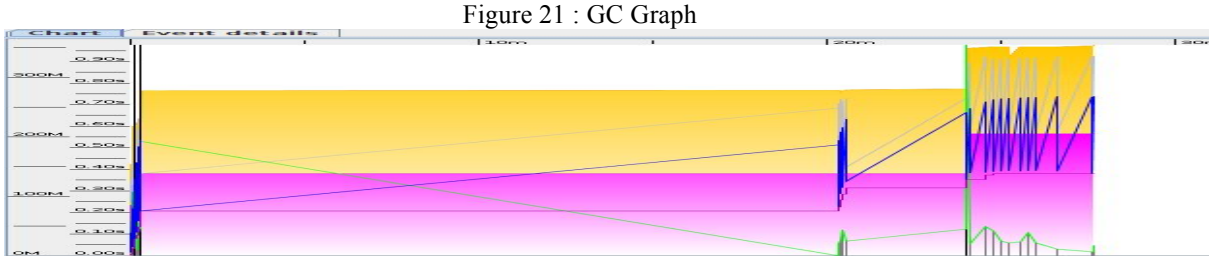


Figure 22 : Summary

Summary	Memory	Pause
Footprint		355.4M
Freed Memory		3,281.2M
Freed Mem/Min		118.493M/min
Total Time		27m41s
Acc pauses		4.37s
Throughput		99.74%
Number of full gc pauses		3
Full GC Performance		12.1M/s
Number of gc pauses		31
GC Performance		1,248.7M/s

Scenario: Performance test for 2000 Users before tuning:

Figure 23 : GC Graph

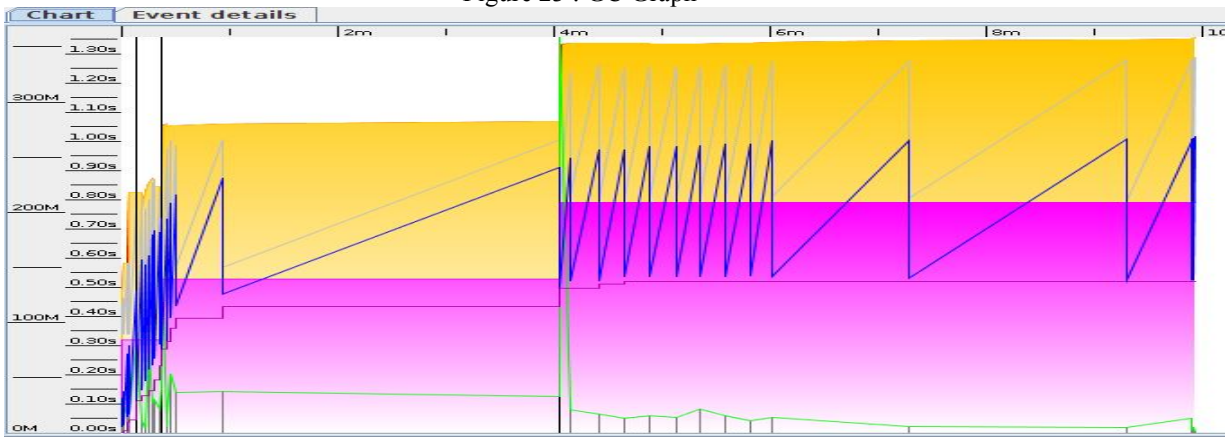


Figure 24 : Summary

Summary	Memory	Pause
Footprint		359.1M
Freed Memory		3,497.5M
Freed Mem/Min		352.041M/min
Total Time		9m56s
Acc pauses		5.09s
Throughput		99.15%
Number of full gc pauses		3
Full GC Performance		9,768K/s
Number of gc pauses		33
GC Performance		1,226.1M/s

Scenario: Performance test for 50 Users after tuning:

Figure 25 : GC Graph

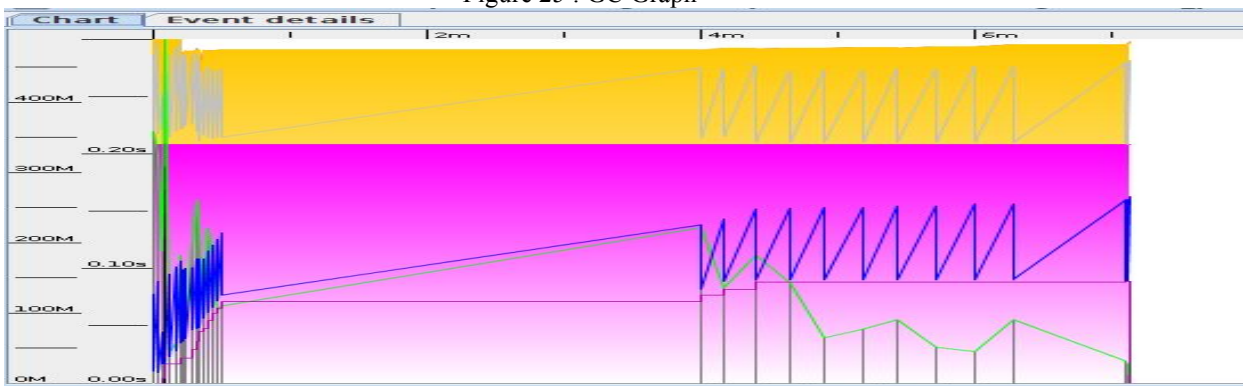


Figure 26 : Summary

Summary	Memory	Pause
Footprint		490.7M
Freed Memory		3,163.3M
Freed Mem/Min		442.55M/min
Total Time		7m8s
Acc pauses		2.73s
Throughput		99.36%
Number of full gc pauses		1
Full GC Performance		782.5K/s
Number of gc pauses		31
GC Performance		1,303.5M/s

Scenario: Performance test for 100 Users after tuning:

Figure 27 : GC Graph

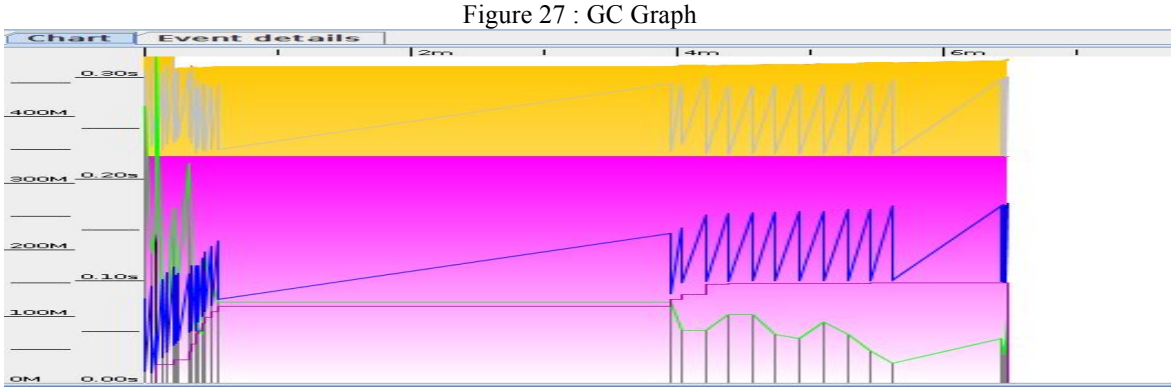


Figure 28 : Summary

Summary	Memory	Pause
Footprint		490.7M
Freed Memory		3,269.4M
Freed Mem/Min		503.06M/min
Total Time		6m29s
Acc pauses		2.88s
Throughput		99.26%
Number of full gc pauses		1
Full GC Performance		733.6K/s
Number of gc pauses		32
GC Performance		1,275.7M/s

Scenario: Performance test for 200 Users after tuning:

Figure 29 : GC Graph

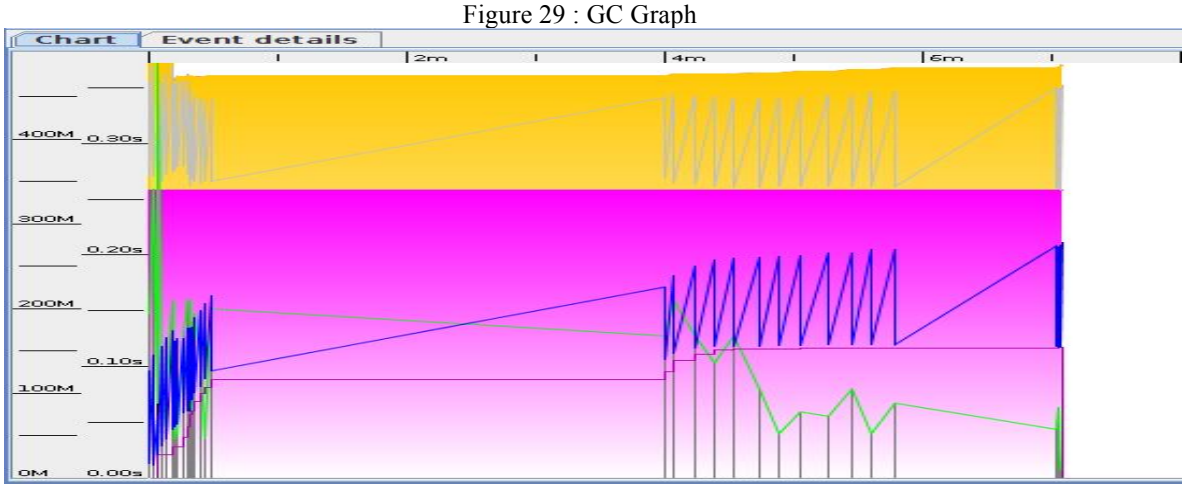


Figure 30 : Summary

Summary	Memory	Pause
Footprint		490.7M
Freed Memory		3,407.8M
Freed Mem/Min		480.597M/min
Total Time		7m5s
Acc pauses		3.37s
Throughput		99.21%
Number of full gc pauses		1
Full GC Performance		647.2K/s
Number of gc pauses		33
GC Performance		1,137.7M/s

Scenario: Performance test for 500 Users after tuning:

Figure 31 : GC Graph

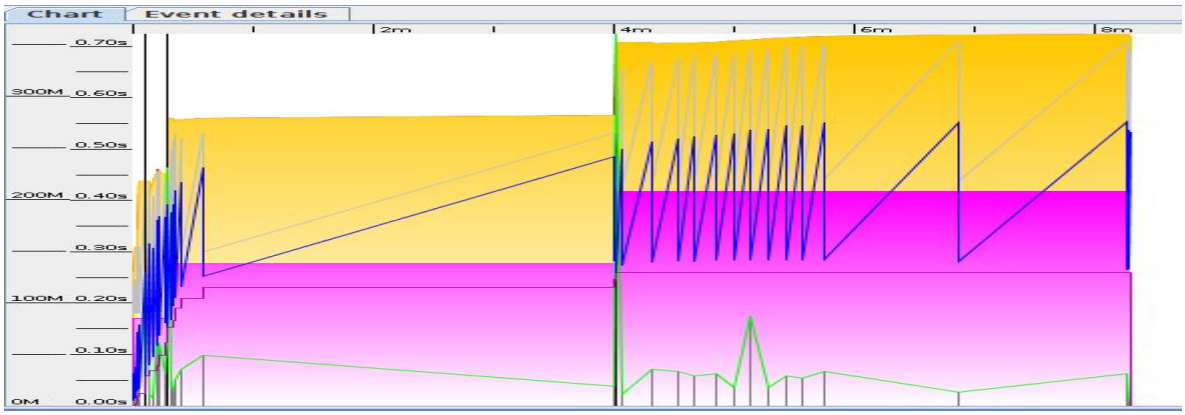


Figure 32 : Summary

Summary	Memory	Pause
Footprint		490.7M
Freed Memory		2,061.9M
Freed Mem/Min		2,321.72M/min
Total Time		53s
Acc pauses		2.61s
Throughput		95.1%
Number of full gc pauses		1
Full GC Performance		876K/s
Number of gc pauses		21
GC Performance		877.1M/s

Scenario: Performance test for 2000 Users after tuning:

Figure 33 : GC Graph

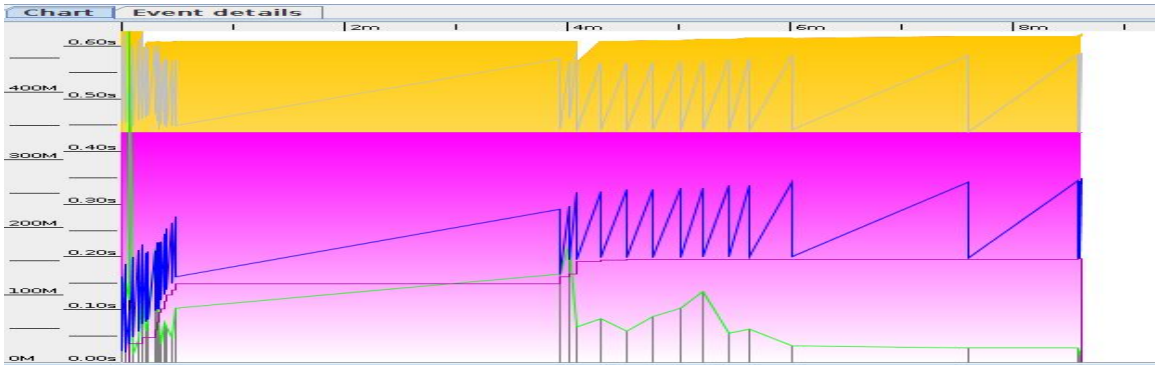


Figure 34 : Summary

Summary	Memory	Pause
Footprint		490.7M
Freed Memory		3,239.2M
Freed Mem/Min		375.418M/min
Total Time		8m37s
Acc pauses		3.32s
Throughput		99.36%
Number of full gc pauses		1
Full GC Performance		493.1K/s
Number of gc pauses		32
GC Performance		1,205.5M/s

4. Modelling

The system that we are testing is simple enough with basically a JBoss server, which acts both as Webserver and Application server. There is no backend database in this system.

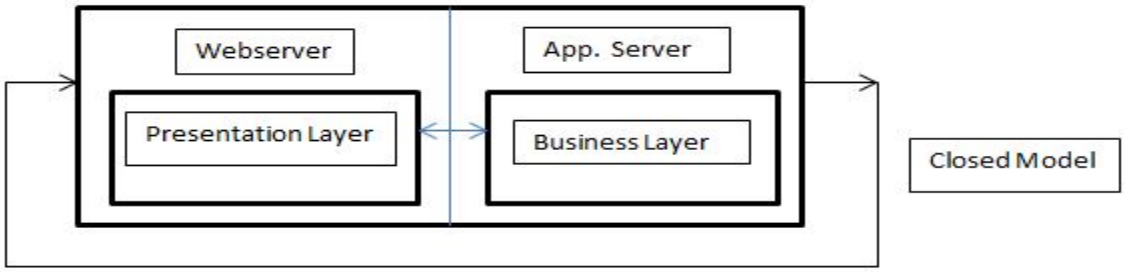


Figure 35: Closed Model Diagram

To model that we used Java Modeling Tool (JMT).We will be using the Closed Model to represent the system as we don't have the arrival rate from the load test but we know the number of customers/jobs that were run after the JMeter load tests. Since it is a closed model it doesn't have external arrivals or departures. There are fixed number of jobs that keep circulating among the queues. Arrival rate is equal to throughput. We used exponential distribution, as the exponential distribution is the continuous approximation of the geometric distribution. It is used to model the time between successive events, or the time required to service an event. The exponential distribution has the property of have no memory, i.e., knowing the time that the last event occurred is in no way helpful in predicting when the next event may occur. It is very common to assume that the interarrival times of events in a computer system follow an exponential distribution.

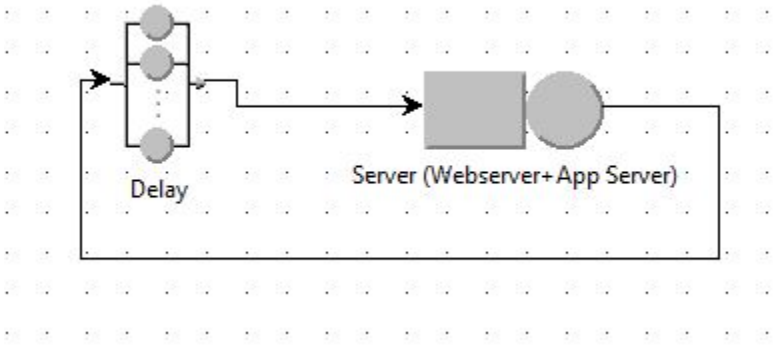


Figure 36 : Close Model JMT Screenshot

Config 3:

sample	Jobs	average	median	min	max	throughput	std. deviation
50 users	34281	64	13	3	2603	135.4915972	173.3141534
200 users	40694	1907	443	3	63165	148.1716131	5620.021347
1000 users	46692	11470	752	3	286634	126.7093084	24452.15789

Table 1.0

Using results from the Load Testing we modelled and ran the simulation for different number of users.

Set Performance Indices:

Number of Customers, Throughput and Response Time:

1. Conf. Interval : 0.99
2. Max Rel. Err : 0.01

Simulation Parameters:

1. Simulation random seed: random
2. Maximum duration: 300
3. Maximum number of samples : 1,000,000

Delay:

Service Time Distribution: Set to Constant: 0.7s

Set to constant 700ms as that is the thinking time set for JMeter load test

Server (Webserver+App Server)

Queue Section:-

Capacity: infinite

Queue Policy: Non-pre-emptive Scheduling, FCPS

Service Section:-

Service Time Distributions:

Strategy: Load Independent

Service Time Distributions: Exponential, $\lambda = 135.4915972$

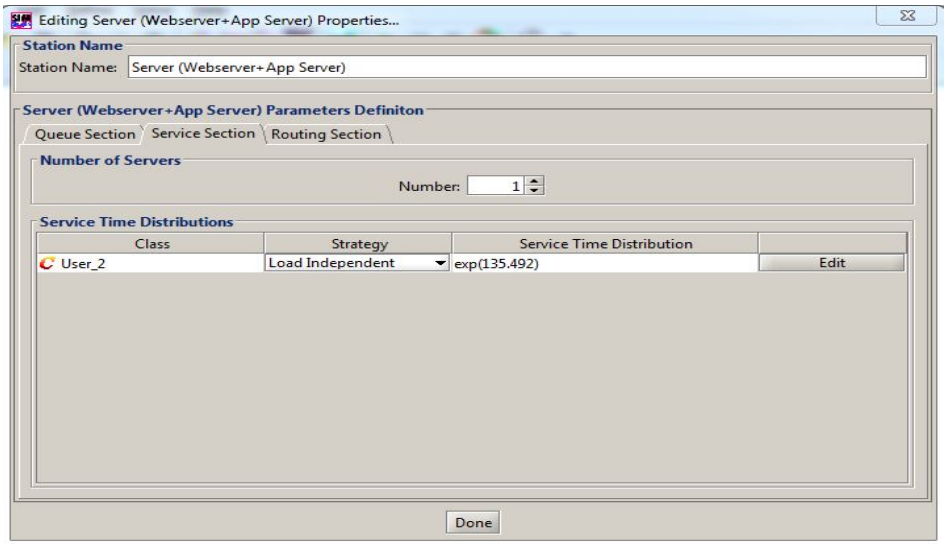


Figure 37 : Station Service Section

What if Analysis?

Number of Customers: 34281 to 40694

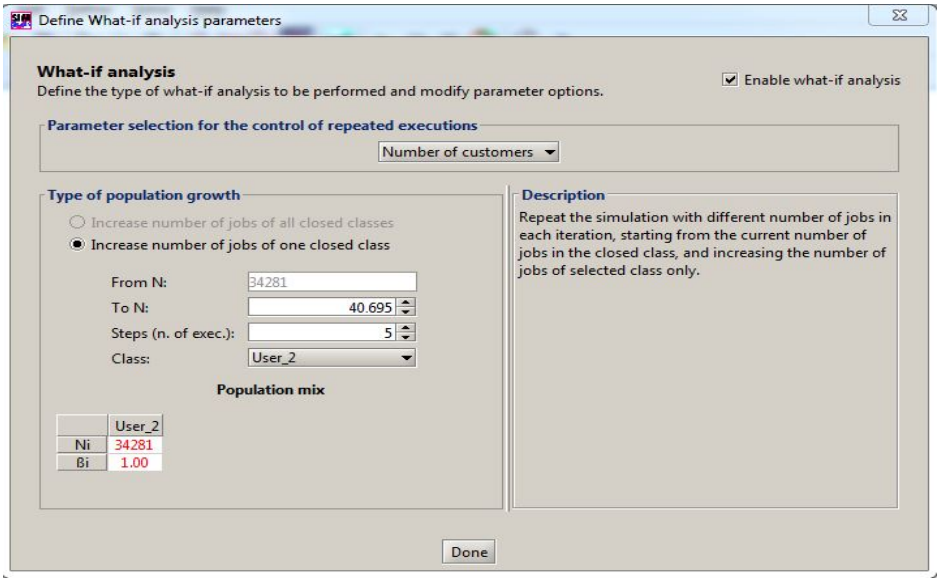


Figure 38 : What if Analysis

To validate the model represents the actual server increase the number of jobs from 34281 to 40694 using the above Table 1.0 and measure the throughput and compare the throughput with the actual reading in Table 1.0 for 200 users. Below figure simulates the behaviour of model/system on increase of jobs to prove the throughput reduces to some extent with increase in number of jobs. Although it doesn't match exactly to live system but simulates the behaviour of live system

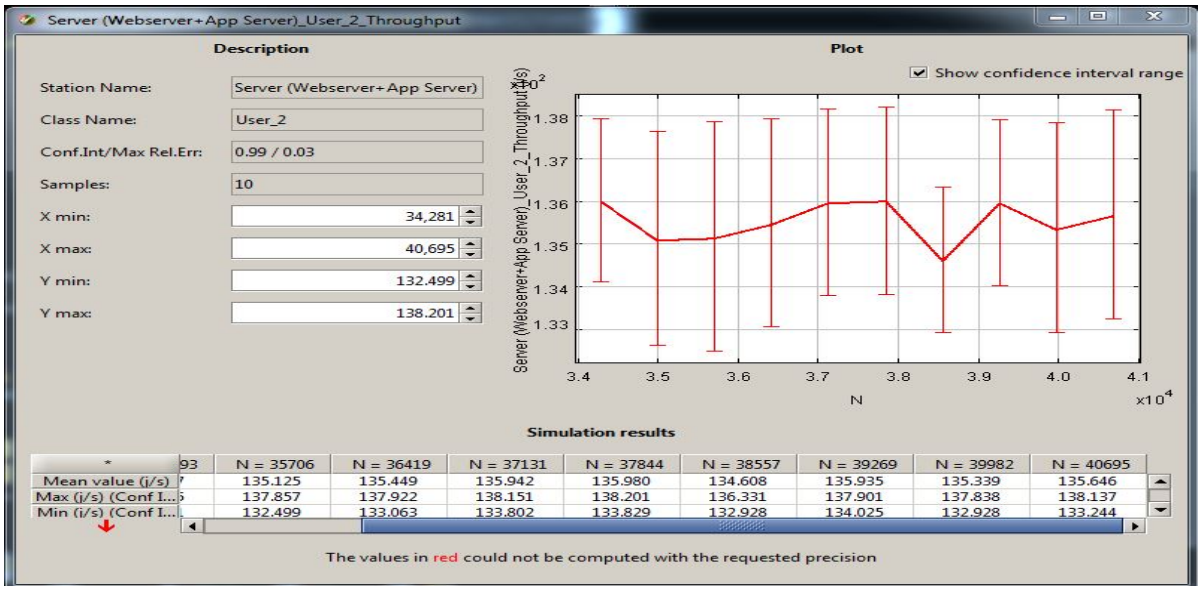


Figure 39 : Simulation Result

Config 5:

sample	Jobs	average	median	min	max	throughput	std. deviation
50 users	27623	250	92	4	4575	106.2647868	381.015596
200 users	31592	2684	663	5	128760	108.9649844	7382.742058
1000 users	31176	17329	2054	5	170611	92.75700378	26522.73391

Table 1.1

Using results from the Load Testing in Table 1.1 we also ran the simulation for accessing Config5 page for different number of users. Most of the JSIM configuration are same as Config3 except

Classes Characteristics:

Population = 27623

Service Section of station Server (Webserver + App Server):-

Service Time Distributions:

Strategy: Load Independent

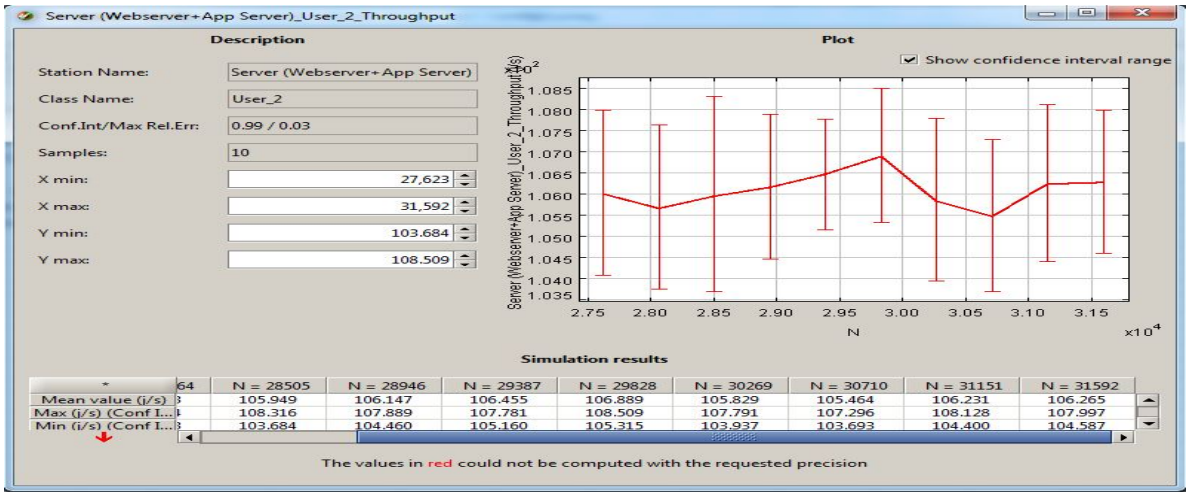
Service Time Distributions: Exponential, $\lambda = 106.2647868$

What if Analysis?

No of Customers: 27,623 to 31,592

To validate the model represents the actual server increase the number of jobs from 34281 to 40694 using the above Table 1.0 and measure the throughput and compare the throughput with the actual reading in Table 1.0 for 200 users. Below figure simulates the behaviour of model/system on increase of jobs to prove the throughput reduces to some

extent with increase in number of jobs. Although it doesn't match exactly to live system but simulates the behaviour of live system



*	64	N = 28505	N = 28946	N = 29387	N = 29828	N = 30269	N = 30710	N = 31151	N = 31592
Mean value (j/s)		105.949	106.147	106.455	106.889	105.829	105.464	106.231	106.265
Max (j/s) (Conf I...		108.316	107.889	107.781	108.509	107.791	107.296	108.128	107.997
Min (j/s) (Conf I...		103.684	104.460	105.160	105.315	103.937	103.693	104.400	104.587

Figure 40 : Simulation Result

Improving Performance by Adding 1 more server:

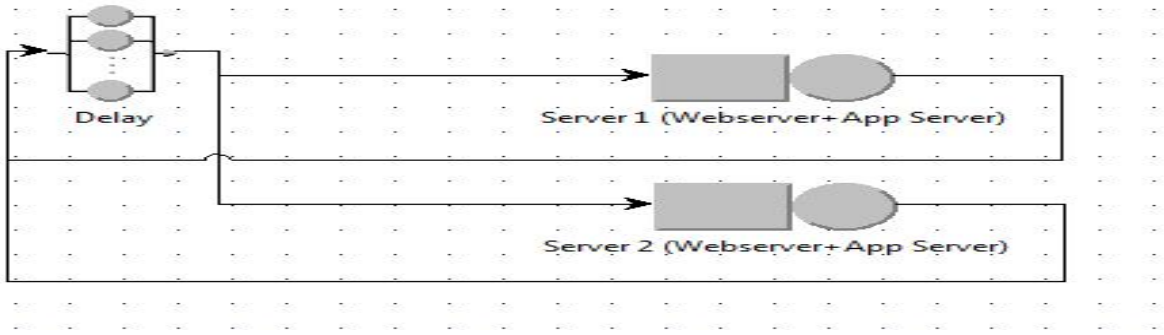


Figure 41 : Closed Model with 2 Servers

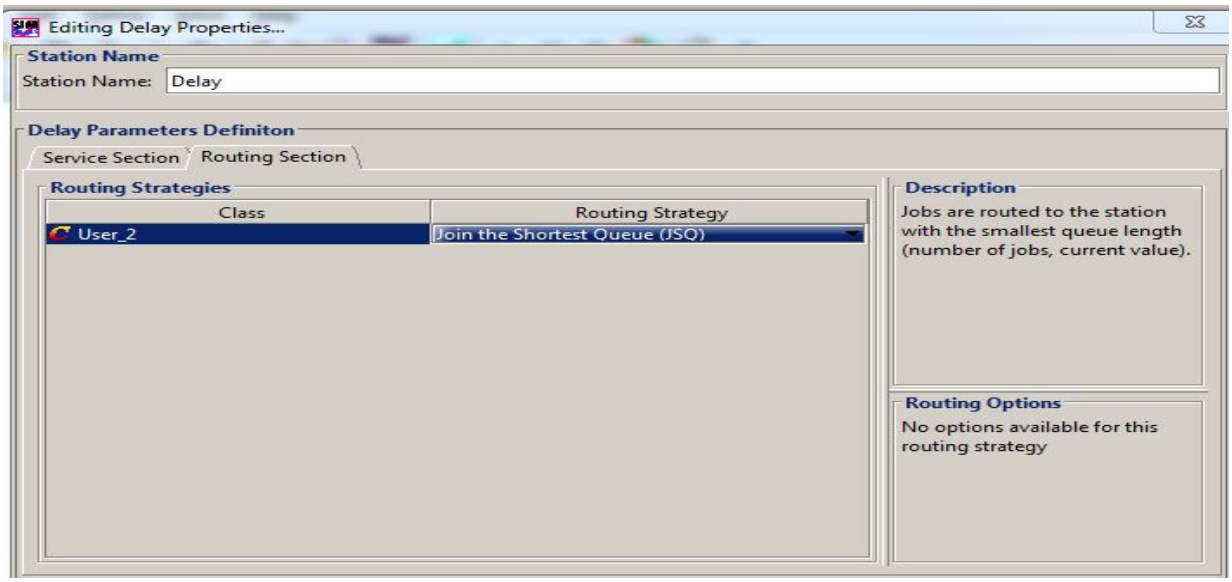


Figure 42 : Routing Section

Added one more server to the model with similar configuration. Noticed that the throughput has increased with the addition of new server.

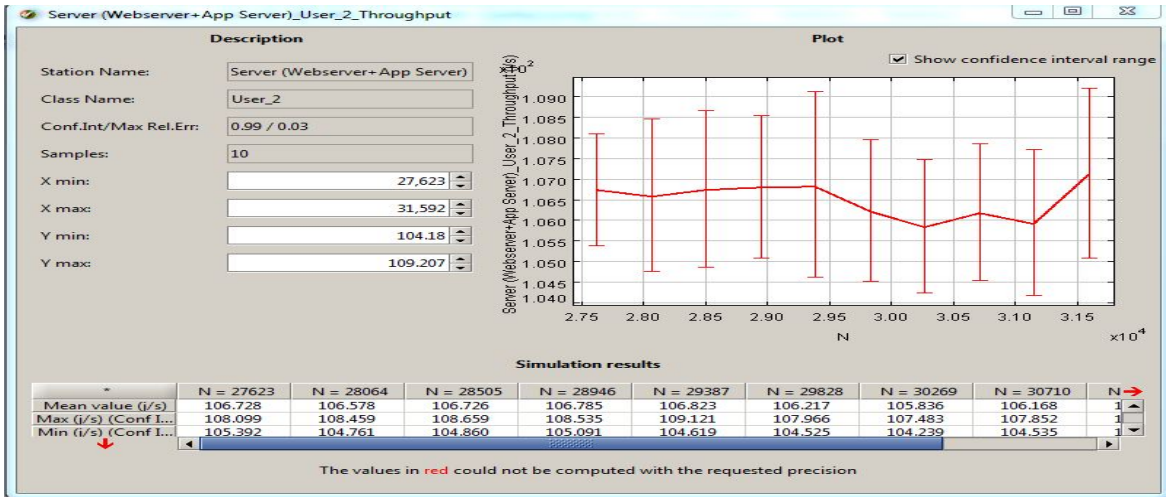
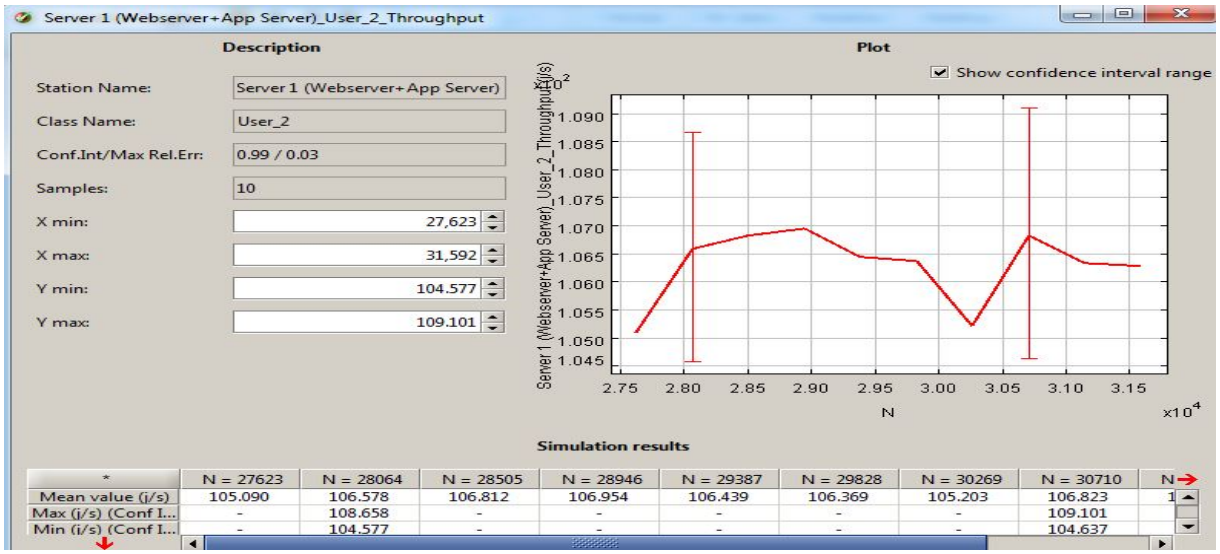


Figure 43 : Simulation Result

Changing the routing section to Shortest R time also increased the throughput



*	N = 27623	N = 28064	N = 28505	N = 28946	N = 29387	N = 29828	N = 30269	N = 30710	N
Mean value (j/s)	105.090	106.578	106.812	106.954	106.439	106.369	105.203	106.823	1
Max (j/s) (Conf I...	-	108.658	-	-	-	-	-	109.101	
Min (j/s) (Conf I...	-	104.577	-	-	-	-	-	104.637	

Figure 44: Simulation Result

Conclusion and Recommendation:

With the Closed Model simulating the single Jboss server we were able to validate that the model. Also, we tried different What If Analysis? To demonstrate that model behaves similar to the live scenario by increasing the number of jobs for 2 different page accesses (Config3, Config5) and comparing the throughput numbers from the load testing results. Noticed that by adding one more servers and changing routing strategy to the closed model throughput can be increased. The routing strategy that suits for this model is Shortest R time and Join the Shortest Q increased the throughput. To improve performance we could add another instance of the application and webserver and load balance them so that requests go to Shortest Q or Shortest R time.