Performance Test Report
For
OpenCRM

Submitted By:
Softsmith Infotech.

About The Application:

OpenCRM is a Open Source CRM software ideal for small and medium businesses for managing leads, accounts and other marketing related activities

Performance Test Summary:

Scenario:

To Test the application for 10 concurrent users. Here all the 10 users will login to the application and add a business lead and logout. The test was conducted to test the influence of users on the application’s performance (mainly response time).

Following all the Activates done to execute the above performance test scenario:

1) Neo Load was used to generate the test script and execute the performance test.
2) A test script was created to Login as sales man → Open the Leads page → Open the add leads Form → Save a lead → Logout.
3) All the dynamic server parameters and session values were correlated
4) In order to enable multiple users access the system, 10 users were created and there user names and password were parameterized using a data sheet.
5) All the Data required to create a lead were passed through another data sheet. This is mimic a real user work load (different sales man creating different leads)
6) The Script was verified by running the test with one user.
7) Then a workload profile was created to test the system with multiple users. The following will describe the work load.

    - The test was run with 10 users
    - The test was run for a duration of 30 minutes
    - The users were allowed to login 1 at a time. The interval between each user was 1 minute. (This is to find the influence of users on the application’s performance)
    - All the users were set to connect to the application with a network speed of 256kbps.
    - Monitors were added to analyze the health of the server during the load test.

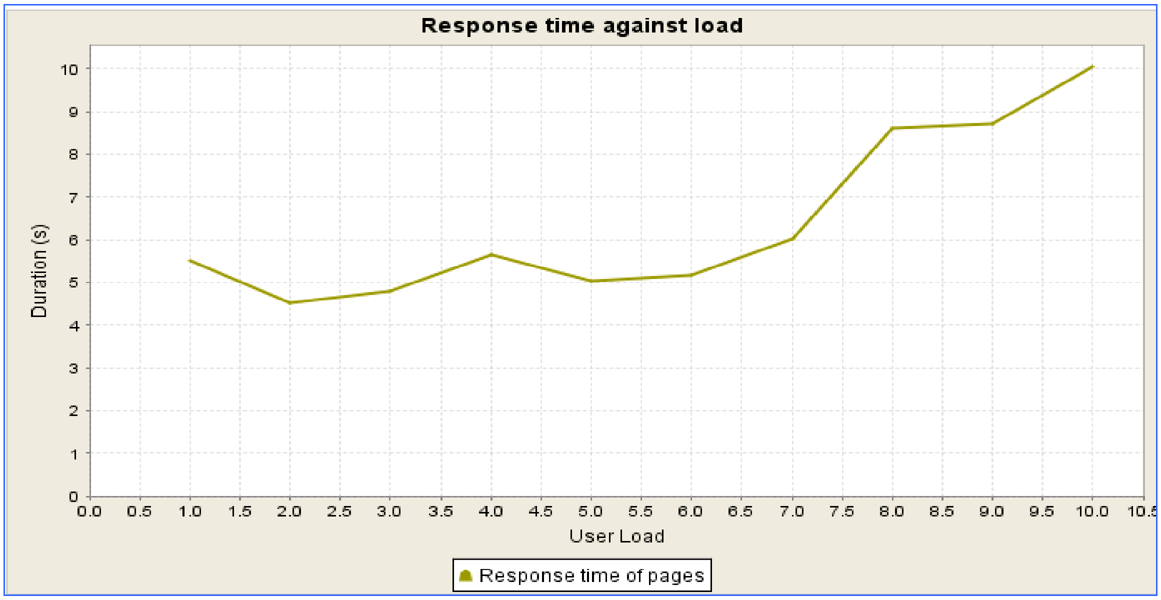
The consecutive sections will describe the performance test results in detail.

Generic Performance test counters:

Average pages/s:	0.5
Average hits/s:	4.4
Average Request response time:	1.45s
Average Page response time:	9.77s
Average throughput:	0.30 Mb/s
Average Errors/s:	0.1

User Influence Graph:

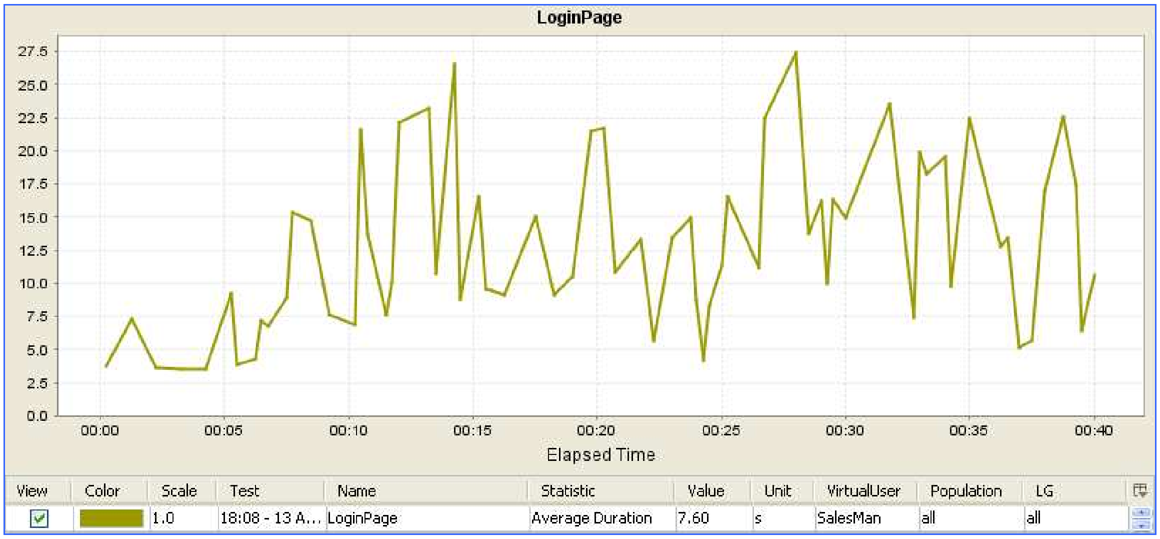
The following graph will show the response time against the number of users. The below graph is for complete scenarios. This graph gives the average page response time against the user influence for the entire scenario.



Observations:
The above graph shows that response time steeply increases as the numbers of users increase. It’s a common phenomenon that the response time will go up as the number users increase, but the increase should be very minimal. Here we see that its increasing steeply and it’s a sure performance issue.

User Influence graph for the Login Transaction:

The below graph shows how the login page performed as the number users increased.



View	Color	Scale	Test	Name	Statistic	Value	Unit	VirtualUser	Population	LG
☑	▬	1.0	18:08 - 13 A...	LoginPage	Average Duration	7.60	s	SalesMan	all	all

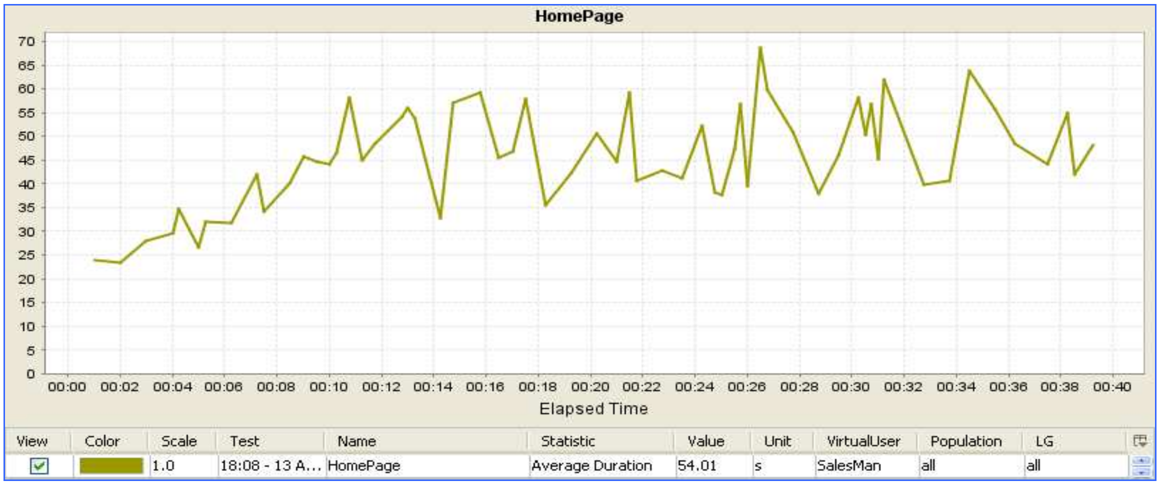
Response Time Table

Minimum Response time in Seconds	3.5
Average Response time in seconds	12.9
Maximum response time in seconds.	28

Observations:
As already stated above we have gradually increased the number of users. As you can see that the response time has been steadily increasing as the number of user's increases. **The Average response time for Login page is 12.9 Seconds.** The huge difference in the minimum, average and maximum values indicate that the application is not responding quickly enough as the number of users increases.

User Influence graph for the Home Page Transaction:

The below graph shows how the home page performed as the number users increased.



View	Color	Scale	Test	Name	Statistic	Value	Unit	VirtualUser	Population	LG
☑	▬	1.0	18:08 - 13 A...	HomePage	Average Duration	54.01	s	SalesMan	all	all

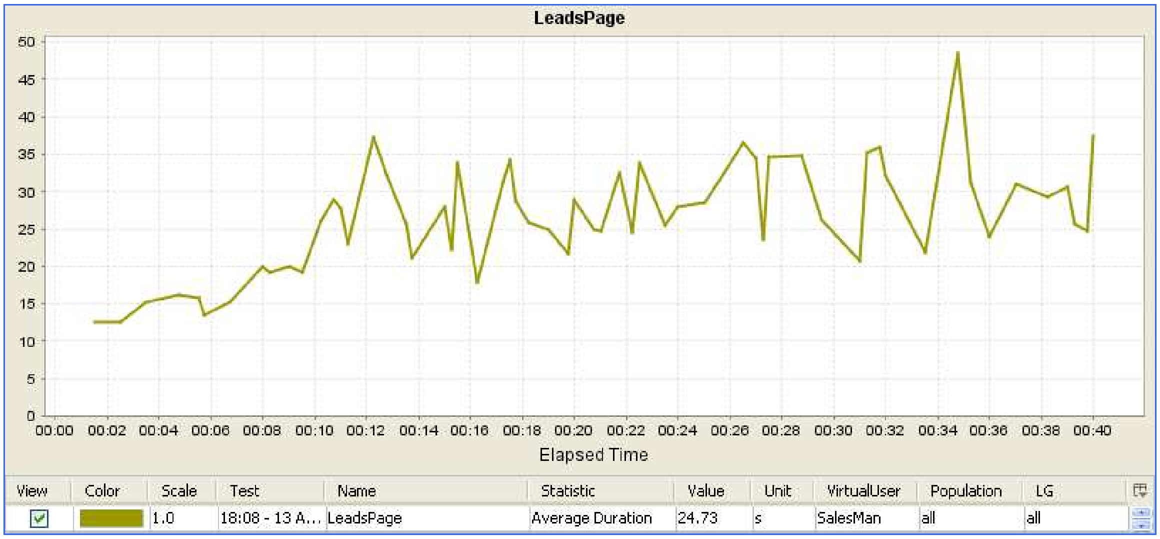
Response Time Table

Minimum Response time in Seconds	23.4
Average Response time in seconds	46.5
Maximum response time in seconds.	69.1

As you can see that the response time has been steadily increasing as the number of user's increases. **The Average response time for Home page is 46.5 Seconds.** In the above graph it's evident that the response time for 1 user itself is 23.4 seconds which is really alarming. More over an average response time of 46.5 seconds is not acceptable.

User Influence graph for the Leads Page Transaction:

The below graph shows how the leads page performed as the number users increased.



View	Color	Scale	Test	Name	Statistic	Value	Unit	VirtualUser	Population	LG
☑	▬	1.0	18:08 - 13 A…	LeadsPage	Average Duration	24.73	s	SalesMan	all	all

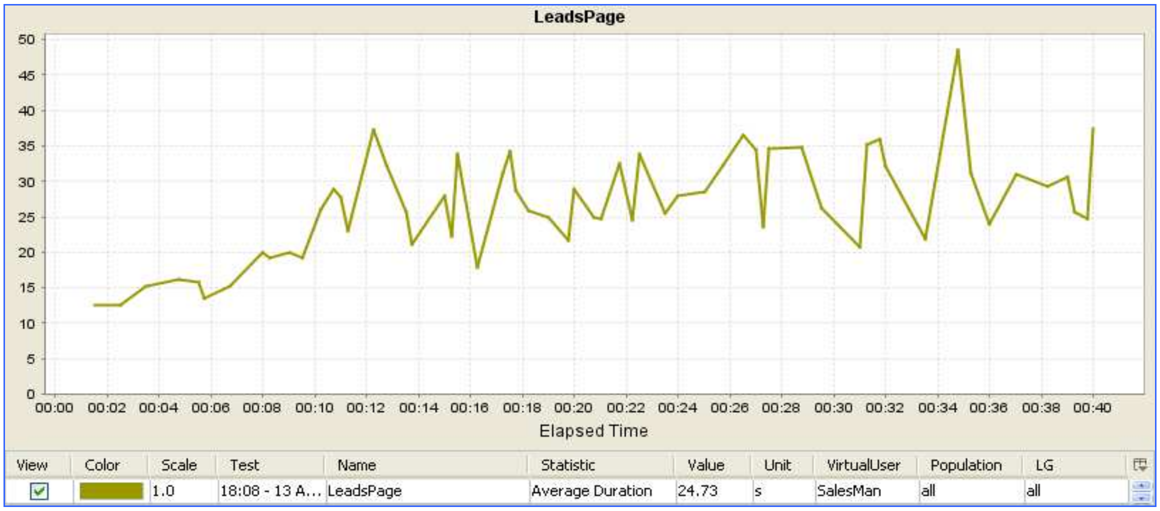
Response Time Table

Minimum Response time in Seconds	12.6
Average Response time in seconds	26.7
Maximum response time in seconds.	48.5

As you can see that the response time has been steadily increasing as the number of user's increases. **The Average response time for Leads page is 26.7 Seconds.** The huge difference in the minimum, average and maximum values indicate that the application is not responding quickly enough as the number of users increases. More over an average response time of 26.7 seconds is not acceptable.

User Influence graph for the Add leads From Transaction:

The below graph shows how the Add leads from performed as the number users increase.



View	Color	Scale	Test	Name	Statistic	Value	Unit	VirtualUser	Population	LG
☑	■	1.0	18:08 - 13 A...	LeadsPage	Average Duration	24.73	s	SalesMan	all	all

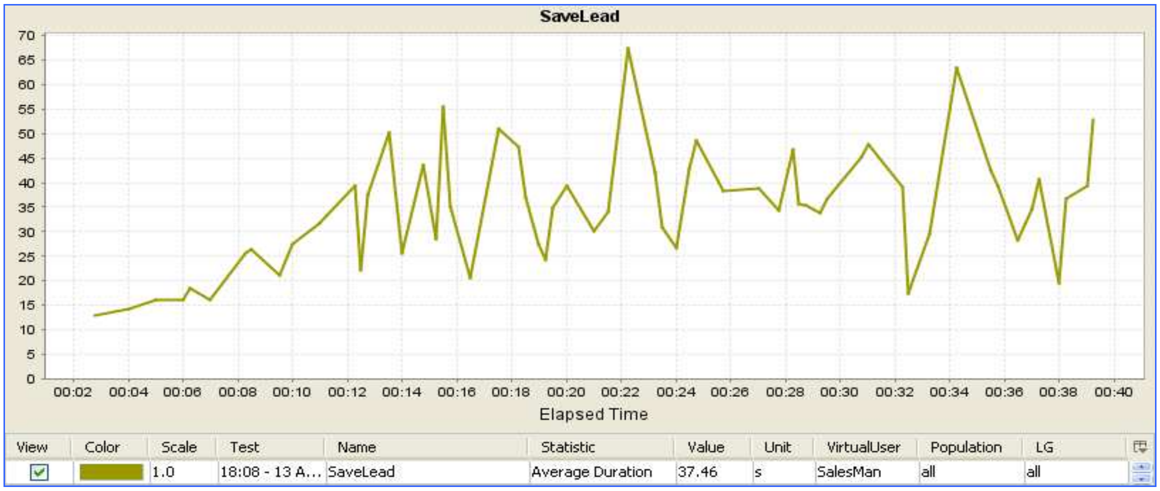
Response Time Table

Minimum Response time in Seconds	6.28
Average Response time in seconds	21.3
Maximum response time in seconds.	46.2

As you can see that the response time has been steadily increasing as the number of user's increases. **The Average response time for Add Leads form is 21.3 Seconds.** The huge difference in the minimum, average and maximum values indicate that the application is not responding quickly enough as the number of users increases.

User Influence graph for the Save Leads Transaction:

The below graph shows how the save leads transaction has performed as the number users increased.



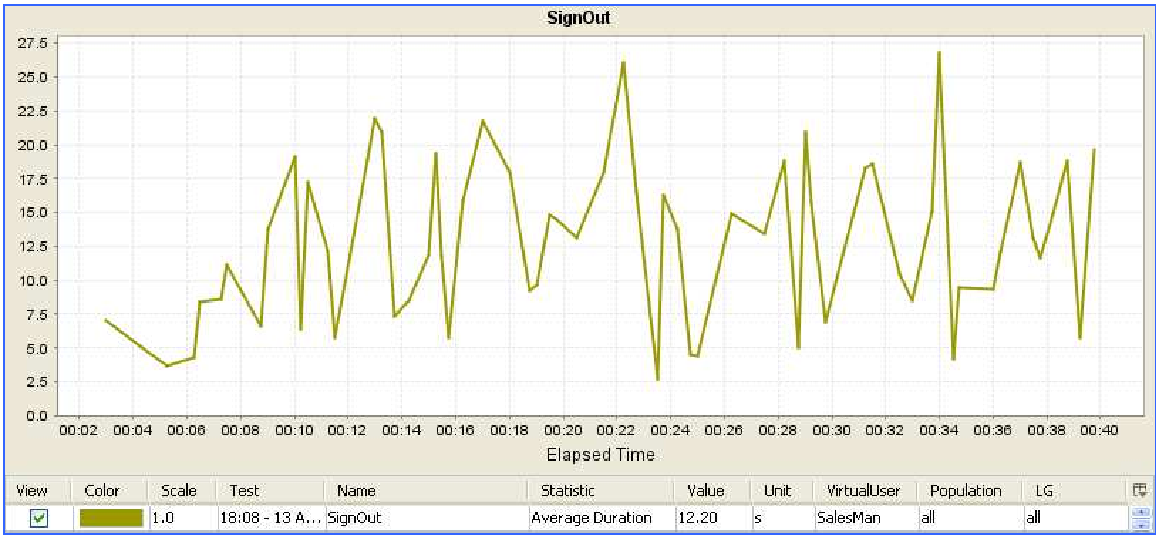
View	Color	Scale	Test	Name	Statistic	Value	Unit	VirtualUser	Population	LG
☑	■	1.0	18:08 - 13 A...	SaveLead	Average Duration	37.46	s	SalesMan	all	all

Response Time Table

Minimum Response time in Seconds	12.8
Average Response time in seconds	35.4
Maximum response time in seconds.	67.3

As you can see that the response time has been steadily increasing as the number of user's increases. **The Average response time for Leads page is 35.4 Seconds.** The huge difference in the minimum, average and maximum values indicate that the application is not responding quickly enough as the number of users increases. More over an average response time of 35.4 seconds is not acceptable.

User Influence graph for the Signoff Transaction:
The below graph shows how the signoff page has performed as the number users increased.



View	Color	Scale	Test	Name	Statistic	Value	Unit	VirtualUser	Population	LG
☑	■	1.0	18:08 - 13 A…	SignOut	Average Duration	12.20	s	SalesMan	all	all

Response Time Table

Minimum Response time in Seconds	2.7
Average Response time in seconds	12.7
Maximum response time in seconds.	26.7

Page Elements Response time:

Below five is the table of page elements which took maximum time for downloading or executing.

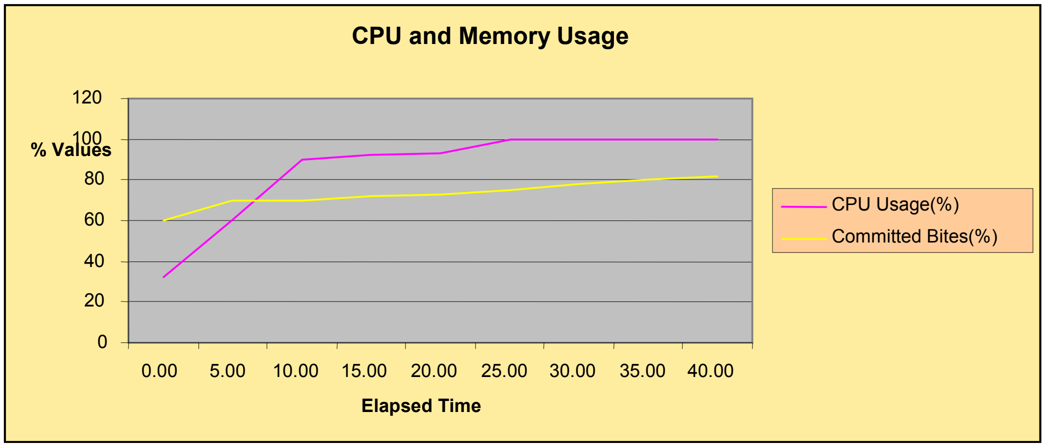
Element	Page	Average Response Time
Controls.js	Home Page	26.1
general.js	Leads page	23.6
vtlib.js	Save Lead	18.7
QuickCreate.js	Add lead Form	17.3
calc.js	Home Page	15.7

Note:
Fine tuning these elements will improve the performance of the application.

Server Resources Utilization Graphs:

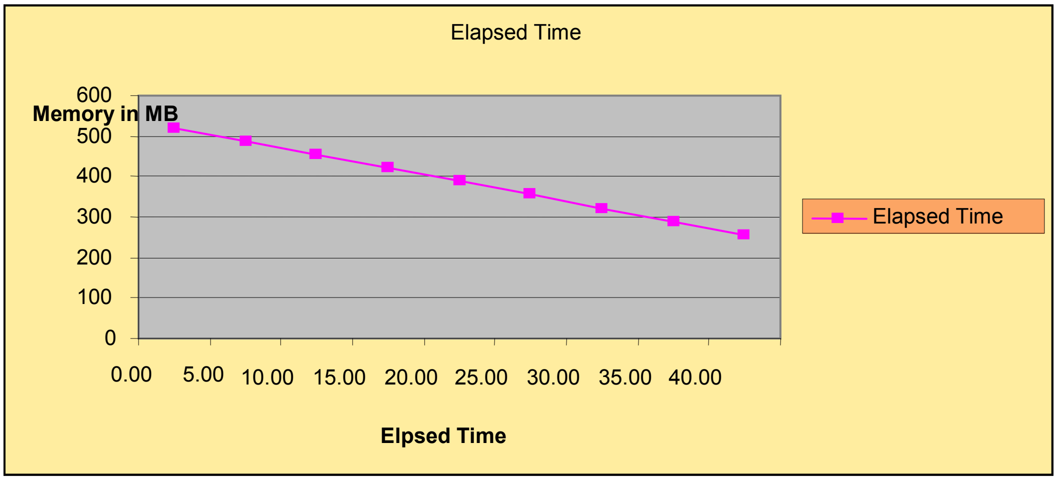
The below graph will show the resource utilization at the server side as the load test was being performed.

CPU and Memory Utilization



Observations:
As you can see that the CPU and Memory are fully used all the time. This could be a problem for increasing response time. Increasing the hardware configuration of the server could fix this issue.

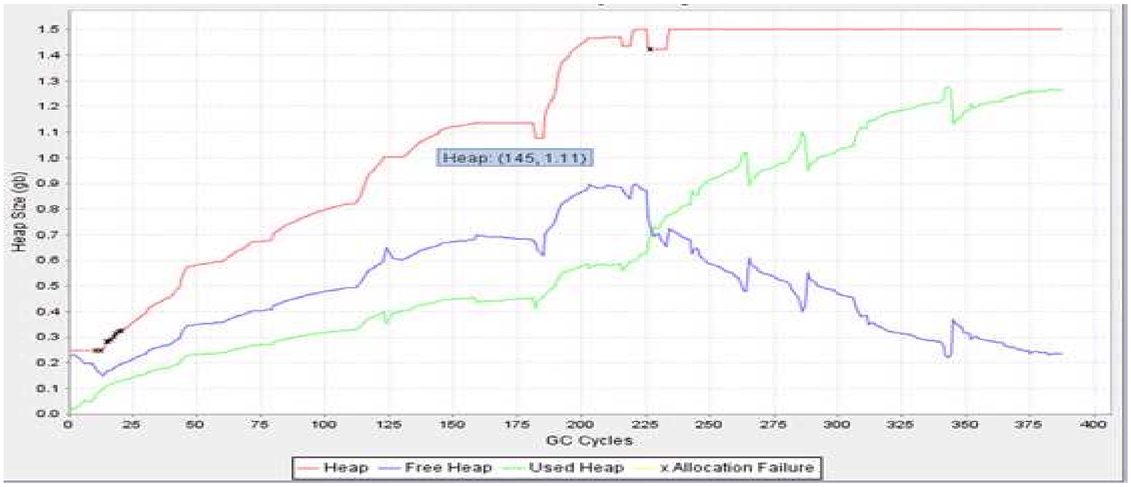
Available Memory:



Observations:

The available memory has gradually come down and not increased after the load least and is an indication of memory loss. More about this can be found my using a probing client.

Tomcat JVM Heap Size:



Observation

As we run the test for longer duration the free heap is decreasing. This is because of low heap size allocated at the server side.

Overall Observations and suggestions:

1) The average response time increases as the number of users increase. This means that the application’s performance degrades as the number of users increase. The system cannot handle more users
2) The CPU and Memory are full at the server side during the load test. This is an indication that the existing
3) The JVM heap size at the server is very less. Increasing the JVM heap size at the server side will improve the performance of the application.
4) Fine tuning the Page elements specified in “**Page Elements Response time”** can be helpful in improving the performance of the application.
5) Try setting the connector attribute setting – “compression to on”. This might improve the performance.
6) Tune the Connector (web server) thread pool settings to more closely match the web request load you have. This is difficult to do, but is very important to get right for best performance. The important Connector attribute for this is maxThreads. Setting this too low means that you may not have enough threads to handle all of the requests, in which case requests have to sit idle for some time without being handled until another request thread is freed up