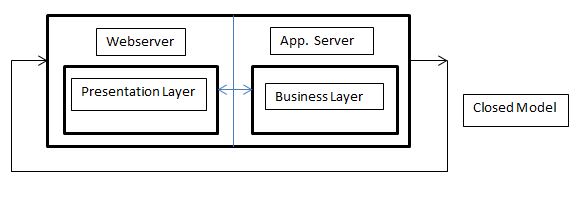
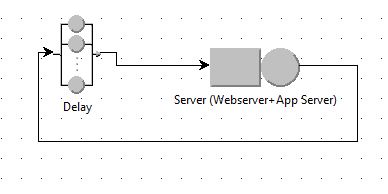
**Modelling**

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.



To model that we used Java Modelling 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. 

**Config 3:**

**Table 1.0**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

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

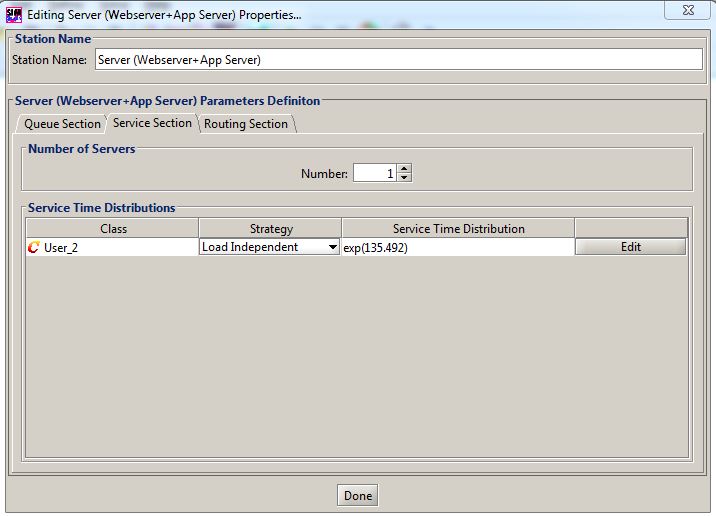
Queue Policy: Non-pre-emptive Scheduling, FCPS

Service Section:-

Service Time Distributions:

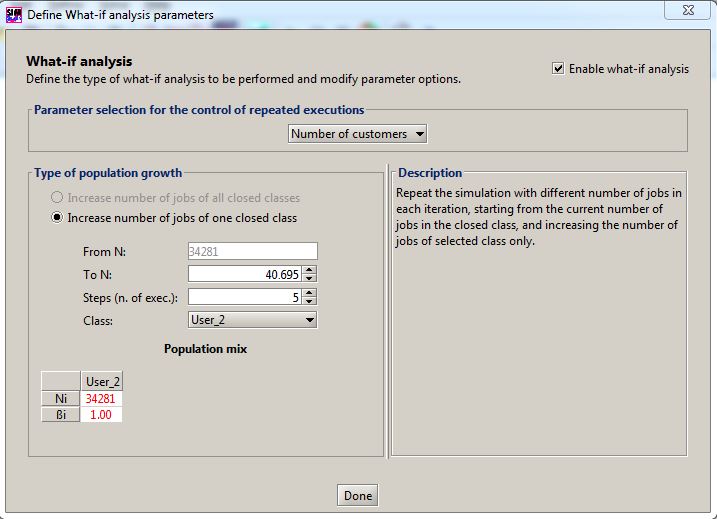
Strategy: Load Independent

Service Time Distributions: Exponential, λ **=** 135.4915972

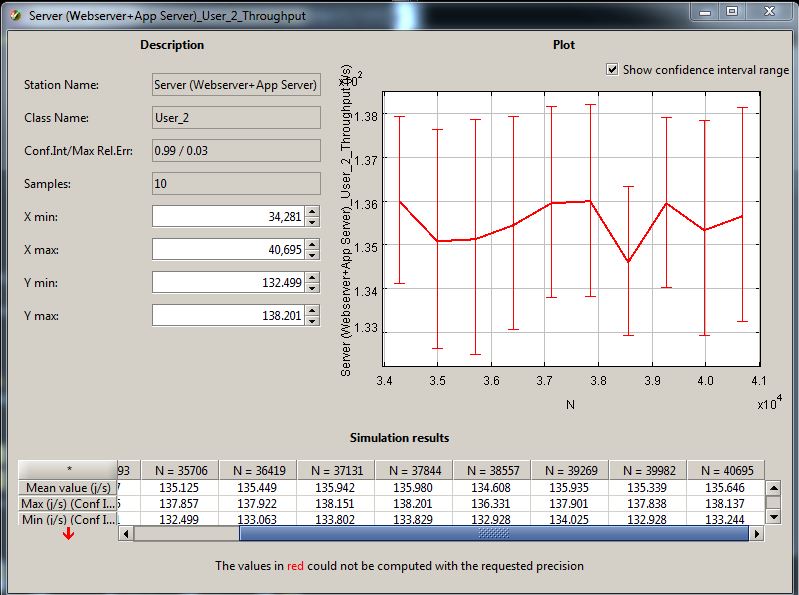


**What if Analysis?**

**Number of Customers:** 34281 to 40694



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



**Config 5:**

**Table 1.1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

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:

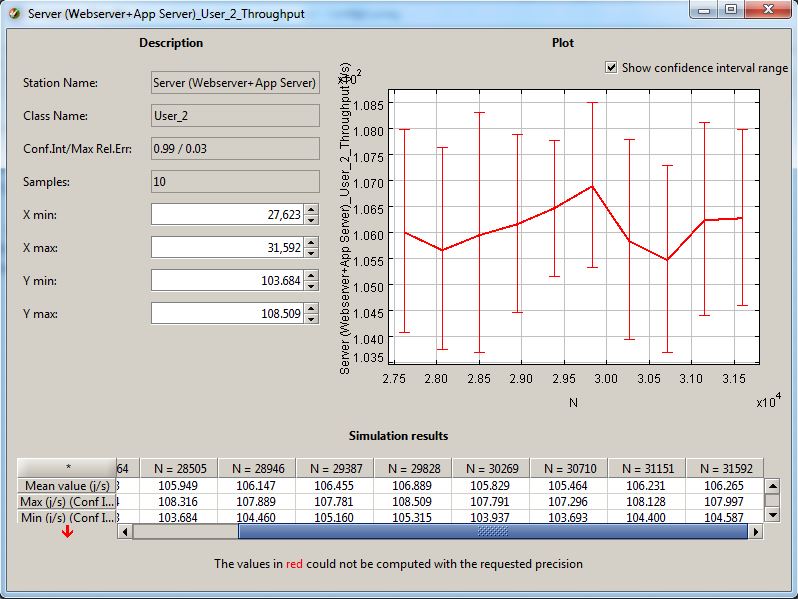
Strategy: Load Independent

Service Time Distributions: Exponential, λ **=** 106.2647868

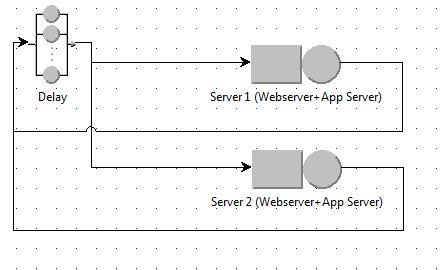
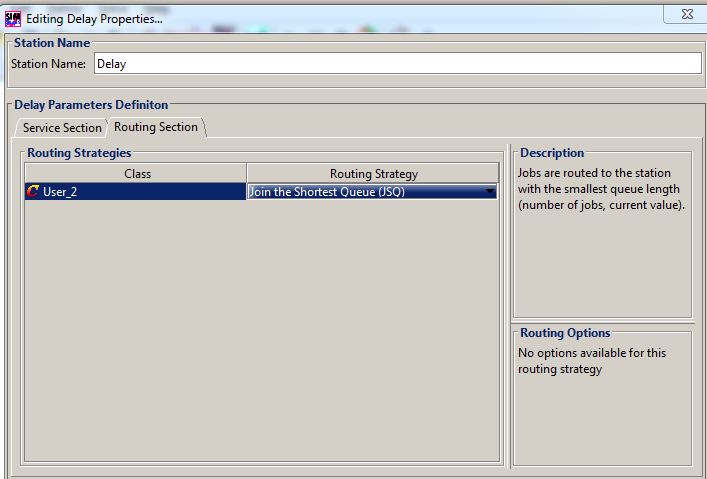
**What if Analysis?**

No of Customers: 27,623 to 31,592

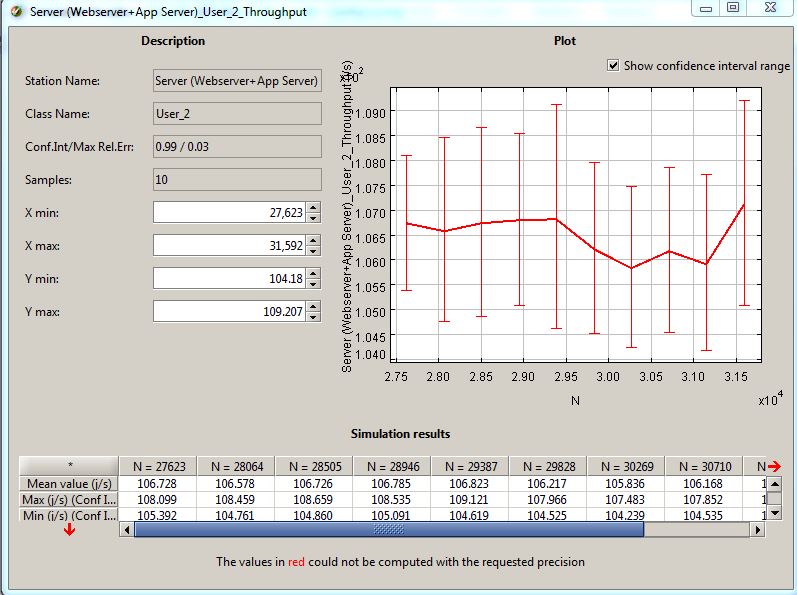
To validate the model represents the actual server increase the number of jobs from 34281 to 40694 using the above Table 1.0 and measure the throughput and compare the throughput with the actual reading in Table 1.0 for 200 users. Below figure simulates the behaviour of model/system on increase of jobs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

****

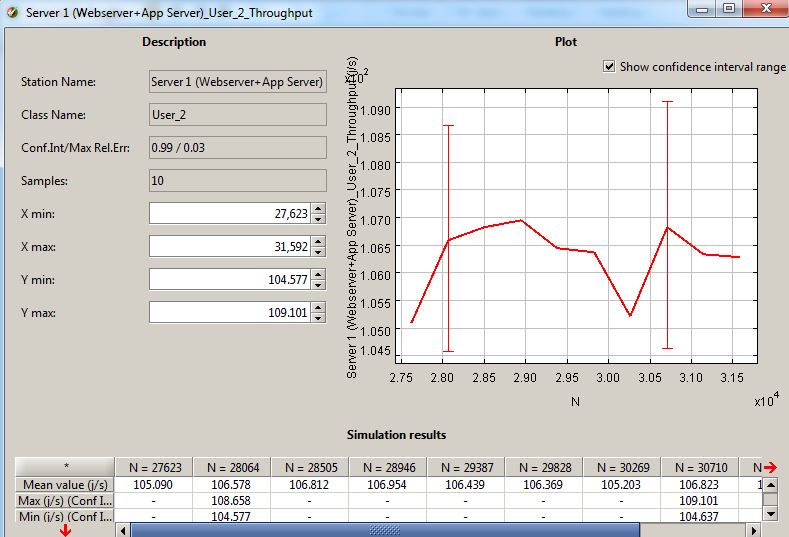
**Improving Performance by Adding 1more server:**

** **

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



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

****

**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.