CHAPTER 7

Test Setup

To understand the concept of failover and load balancing in weblogic, I ran multiple tests to gather the data and to determine the efficiency and effectiveness of all the three available algorithms. In my LAB environment, I have created a domain with three weblogic server instances on machine A and two more clustered servers on another machine B. clustered server were named as D and DCL2.

For testing I have created a stateful session bean and deployed it on the created clustered. This bean only contained a ping method which on return simply prints a string. Client java program will create lots of threads and will invoke the deployed session bean many times in a row and will make a entry in log file for each receiving request. Since the request will divert in cluster so it will be logged in the different server logs according to the algorithm. Later I have examined the logs to know how the loads or requests were distributed[32][33].

Challenges Faced

Throughout the testing of this concept I got lots of challenges and problems and out of all on some I was able to overcome however on some not. The first and the main challenge was the setup of the domain and cluster on the weblogic about which the information is not very clear anywhere, also the documentation is not so good enough and helpful to determine the best settings like beans in the cache. This caused lots of testing to setup the domain, cluster and deployment to achieve the desired solution.

Another challenge was how to configure the logging and collect the statistics of the tests. Logging is the main method to capture and troubleshoot the real picture of processed requests. Since my concern was not all the requests goes to a single server instead of that equally distributed over the cluster so it was the main work me to find out the proper solution. I tried to used the java options like print line but it doesn’t work for me properly. After doing further investigation I found
the weblogic inbuilt mechanism to logging the requests to each and every server and configured the same.

Another challenge was, since weblogic server is the J2EE based application server, to build the java based test program in J2EE format. I found that a normal java program is different than writing the program in J2EE standard. So studied further and later I was able to create and deploy a testing java application.

Another challenge which I didn’t realized initially was I have configured lots of logic in the descriptors of the deployment. It took a lot to me to get only used one descriptors and to remove not required one to get it functioning properly.

Rest of the issues I found during the testing specially in the testing of failover capabilities. I have done complete setup according to the requirement and the standard document but wasn’t being able to get the results. Failover has not been worked first for the cache and beans level on the cache. Initially I have a very low value for the cache and found lots of exceptions on the first server only not on second server. Failover was not utilizing the objects properly, later I have killed one of the server to check if requests diverting to other server or not.
Results

During my testing I found that the most efficient algorithm for the load balancing was round robin and this was specifically to my testing environment where each hardware were of same configuration across the all managed servers participating in the clustered way.

Another tested algorithm was weight based. Its useful if you have hardware with the different configurations. Like suppose if you have machine A with 2 cpu and 2 GB RAM and machine B with 1 cpu ang 1 GB RAM then definitely you want more traffic on machine A with respect to the traffic on Machine B. So I have changed the weitage percentage on server D to 100% and 50% on server DCL2. During testing the results was as expected, 1/3 load on second server and 2/3 on first server.

Last one tested algorithm was random, test results shown that the random algo results were almost the same as the results of the round robin. This was may be due to I had only two options in the cluster or the tests was being uniform and program was able to predict[34].

Cluster Algorithm graph

Fig 20