CHAPTER 6

Clustering Requirements and Types

Clustering is one of the best high availability solution in the IT industry to achieve proper load balancing, failover and session replications or you can it’s a perfect business continuity solution nowadays. It’s totally transparent to the end users accessing the applications and cluster act as a single service instance for the end user.

Oracle Weblogic server is the best clustering supported application server in the market today having the good high availability feature solutions. Below testing has been done on the same and I will describe the different algorithms support with the weblogic for proper load balancing.

Clustering Requirements

For better solution, Clustering should support the following requirements -

First, Applications must continue to serve the requests of clients without interrupting the new and current connections and for that there shouldn’t be and bottlenecks for the further scaling. If needed we can add more application servers to the cluster at run time to met the increased demand of the users so in that way you should be able to do the dynamic changes at the run time.

Next, designed architecture should design to maintain high availability. If any server goes down or fail in the cluster the requests should not impacted and continue to processing the requests in the cluster. Application should deploy homogeneously on all the servers in a particular cluster. If any one of the server went down then the replica of same should be failover to the other server in the cluster. Applications should not be connected with the cluster features and developers should not have any dependency on the cluster features like load balancing, failover and replication so that applications can design independently and can be deployed as platform independent anywhere.

And the final requirement is that, the clustered services should be managed in way as the non implemented clusters. A cluster should appear as a single point to the system administrator and administrator should able to handle a cluster a single server[31].
BEA WebLogic Clustering

The two great features of the weblogic server make it to achieve the high availability and scalability and these capabilities are the load balancing and failover. Failover means suppose if a connection is get connected with a particular session on a particular server and suddenly that server goes down then the complete session of that connection will failover to the other server in the cluster.

So to meet the condition where newly created objects should take over the failed objects below mentioned should need to be met -

- There should be a copy of each session should be maintained
- To know the exact behavior of a object if it is failed, the information of each object should available to other failover object and also to program which handle the failover like location the status of all of the objects.
- All the information about the currently running jobs should make available to all the objects and programs manages the failover. So in that way an object which take over the failed objects know about where it was during failure, what was that doing and what need to process next.

Weblogic use J2EE based standard technologies and some internal inbuilt technologies like JNDI (java naming and directory interface), multicast, unicast, IP socket etc to maintain and share the information about the availability of the objects in the cluster. All these features and technologies allow weblogic to determine that the particular functionality has been interrupted and the stage where it got failed and the backup failover server where this needs to be replicate.

All the current running jobs information are stored in a logical object to which we called state and weblogic maintains the details of each and every state using the inbuild technology called replica aware stubs and session replication. In case of failure of a particular session stub and session replication allow that job to failover to the other live objects from the same place where it was failed.
Another great feature of weblogic server clustering is load balancing which allow
the equally distribution of the jobs or as per defined cluster algorithms among the
servers within a cluster.

To achieve load balancing:

- There should be more than one copy of an object that perform the job
- For each and every object the operational and location status should be available

Weblogic allows all of the objects are clustered and deployed across all of the
clustered managed servers so that in case of failure of an object other one
alternative object will perform the rest of the job.

**Type of Clustering in WebLogic Server**

We have lots of resources in the field of information technology and weblogic
supports clustering of various kind of resources. Each and every resource having
a different kind of behaviors like the invocation, related to control and functioning
with in an application. So the clustering benefits of load balancing and failover
varies for different kind of resources and weblogic provides support for different
kind of resources[31][32].

Below are the objects clustered in weblogic server

- Servlets
- JSPs
- EJBs
- Remote Method Invocation objects
- Java Messaging Service destinations
- Java Database Connectivity connections

**How It Works**
For EJB’s and RMI objects load balancing and failover is achieved using the replica aware stubs, which may locate the instance of the object within the cluster. Sometimes we called it smart stubs. These replica aware stubs are created and handled for RMI and EJB’s as a result of outcome of the compilation process of objects. So all the RMI and EJBs are homogeneously deployed across all of the servers in the cluster.

Weblogic supports different kind of cluster algorithms for load balancing the EJB’s and RMI’s like round robin, weight base and ransom. Whichever the method you will select will be maintained within the replica aware stub used for the objects clustered.

Server Communication

For proper load balancing, servers within a cluster communicate with each other using the multicast, unicast or IP sockets.

Servers within the cluster communicate with each other using any one of the above communication technology. Using the IP multicast utility servers used the simple mechanism of broadcast in which each and every server communicate with each other using the heartbeat mechanism where they send messages to each other to let each know the availability of the other system. A multicast I.P defined in a given selected defined range from 224.0.0.0 to 239.255.255.255 and port can be used anyone which is not in used. Using multicast which is one too many communication each and every server communicate with each other in the cluster.

- JNDI updates-Cluster-wide –
  
  Every server in the cluster communicate with each other server to inform about his availability using the multicast address and port, each server then listen for the updates from other server and update its local JNDI tree to update the running deployment of the clustered objects.

- Heartbeats in Cluster–
  
  every server in a cluster use multicast to send heartbeat to other servers to inform then its availability, till all the servers listen each other be in synch in cluster and in case of any server goes down cluster remove that server from the cluster.

- IP sockets are used for the peer to peer communication between the servers and used when two servers in the cluster communicate with each other.