**Overview:**

Management System

Server#1

**Active**

Server#2

**Active**

Server#3

**Active**

Server#4

**Standby**

N Clients

N Clients

N Clients

**Requirements:**

1. N+1 redundancy needs to be provided for the Servers. At any time there will be 1 standby server which should replace the active one if it fails.
2. Servers are initially manually provisioned to come up with some default configuration.
3. Each server connects to the Management System and receives a unique set of configuration.
4. The received configuration needs to be persisted.
5. The Standby server needs to be aware of the distinct configuration of every Active server, so that whichever one goes down, it comes up with the configuration specific to that server.
6. The clients that were connected to the previously Active server (that failed), move to the newly Active server.

**Solution Proposed:**

1. Create multiple virtual IP groups, one for each Active server.
2. Create a OCF custom resource to enable/disable the server functionality.
3. Put the constraint to run both resources together and configure a higher score for Active instance and lower score for Standby.
4. Active servers connect to the Management System and receive unique set of configuration.
5. The per server configuration that is received is then store under “Node Attributes” in the form of name-value pair (using command crm\_attribute)
6. Apart from the unique configuration received from Management system, the servers themselves will detect certain run-time information that needs to be stored and communicated with standby node. Same mechanism as above would be used.
7. Pacemaker will ensure that above configuration is synchronized with all instances including the standby one.
8. In the event of any failover, pacemaker will execute the service start script. In the script we will read the configuration specific to the node that went down (again using crm\_attribute) and become active.
9. Because of the virtual IP, the clients should be oblivious to this change (except for the active state that is not check-pointed)
10. Per node statistics will be maintained under Node Attributes with type “status”.