

TestCase description:

for Testcase sequence digrams and complete report, they are decribed in the “Lab3-Design-Report.pdf” file.

The code can be can using two different scripts:

1.run_LoadBalance_Consistency_Caches.sh --- This script is used to demonstrate the implementation of loadBalance,Consistency and Querying for state of a device through Cache.

2.run-part_FT.sh --- This script is used to demonstrate our application is **fault tolerant**. Even if one of the Gateway replicas crashes, the other one will take control.

We have decentralized our code into various Java Packages where each Package corresponds either to a component (the gateway, a sensor or a device or Back end database). In order to avoid the complexity and exceptions due to the supporting code being in different packages, **In addition to the source code we have also provided the executable jar files for each component above described. We have Submitted Source code to verify code and Jar files to execute.**

IP Address Recognition and Allocation:

We are needed to provide only the IP Address of the Gateway in the Configuration file (**configips.csv**). The Default IP Address in configips.csv is local host. This IP Address is needed for all the other components. Each Component can figure out the value of its IP Address when initiated and will register at the Gateway. Gateway stores the IP Address and can access the other components when required.

Jar files & command line arguments in various cases:

For LoadBalance, Consistency, Caching (run_LoadBalance_Consistency_Caches.sh):

The following Jar files will take the command line argument as Path to the Configuration file . **Please place all the Jar files, configips.csv in the same Directory.**

```
gnome-terminal -x sh -c "java -jar GatewayServer.jar configips.csv; bash"&
sleep 3
gnome-terminal -x sh -c "java -jar GatewayServerReplica.jar configips.csv; bash"&
sleep 3
gnome-terminal -x sh -c "java -jar motionSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar tempeSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar DoorSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar HeaterSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar bulbSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar BackendDatabase.jar configips.csv; bash"&
```

For Fault Tolerance (run-part_FT.sh): The following Jar files will take the command line arguments as Path to the Configuration file and “lab3_test”. “lab3_test” is used to distinguish between the above mentioned LoadBalance mode and Fault tolerance mode. **Please place all the Jar files, configips.csv in the same Directory.**

```
gnome-terminal -x sh -c "java -jar GatewayServer.jar configips.csv lab3_test; bash"&
sleep 3
gnome-terminal -x sh -c "java -jar GatewayServerReplica.jar configips.csv lab3_test;
bash"&
sleep 3
gnome-terminal -x sh -c "java -jar motionSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar tempeSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar HeaterSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar bulbSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar DoorSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar BackendDatabase.jar configips.csv; bash"&
```

Executable Script files:

run_LoadBalance_Consistency_Caches.sh : This Script files corresponds to the automation of the “*For LoadBalance, Consistency, Caching*” case discussed above. This script is executable and has all permissions. User Just need to run this script and Terminals Pop up.

(i) Load Balance will occur automatically and The nodes which are assigned to Gateway Replica will show which they are assigned . The Gateway and Gateway Replica terminal also displays the registered nodes under them. [See Appendix 1 screenshot of obtained results.](#)

(ii) For consistency Testing User need to manually enter and report .For example take temperature sensor and it is assigned to Gateway. Even if we report to Gateway the value will also be updated in Gateway replica. User can see that value is updated in both replicas. [User Input Needed. The few starting output in terminal will convey the operation performed by that process. See Appendix 2 screenshots for obtained results.](#)

(iii) For Cache testing User need to follow the instructions on the screen in **Gateway terminal or Gateway Replica terminal** . On entering one of the Integer values of **Device/Sensor Ids** as mentioned in the screen Querying takes place and Initially user will get a cache miss as this is the first entry and we read from database. If user query using the same **Device/Sensor Ids** then a cache hit occurs and we get data from cache. [User Input Needed. The few starting output in terminal will convey the operation performed by that process. See Appendix 3 screenshots for obtained results.](#)

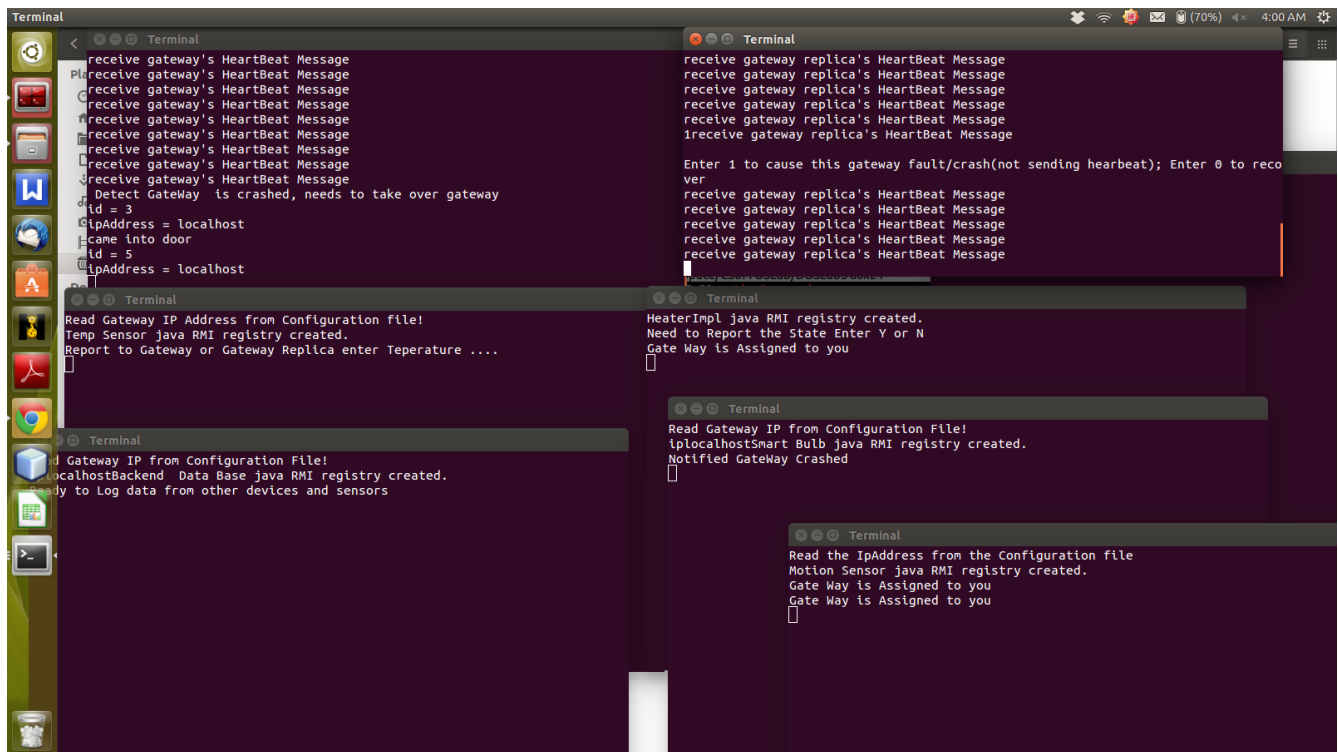
run-part_FT.sh : This Script files corresponds to the automation of the “*For Fault Tolerance*” case discussed above. This script is executable and has all permissions. User Just need to run this script and Terminals Pop up. [See Appendix 4 screenshots for obtained results.](#)

(i) As terminals pop up the Gateway and Gateway Replica will be sending Heart beats to each other.

(ii) **Enter 1** in either Gateway or Gateway Replica **to crash** one of them. After that user will not see any

heartbeats.

(iii) **Enter 0** in the previously crashed Gateway/Gateway Replica **to recover** the Gateway/Gateway Replica. User will see heart beats being exchange again.



The screenshot shows a Linux desktop environment with a dock on the left containing icons for various applications. Five terminal windows are open, displaying the following text:

- Terminal 1 (top left):**

```
receive gateway's HeartBeat Message
receive gateway's HeartBeat Message
receive gateway's HeartBeat Message
receive gateway's HeartBeat Message
receive gateway's HeartBeat Message
receive gateway's HeartBeat Message
receive gateway's HeartBeat Message
Detect GateWay is crashed, needs to take over gateway
id = 3
ipAddress = localhost
came into door
id = 5
ipAddress = localhost
```
- Terminal 2 (top right):**

```
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
Enter 1 to cause this gateway fault/crash(not sending hearbeat); Enter 0 to recover
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
receive gateway replica's HeartBeat Message
```
- Terminal 3 (middle left):**

```
Read Gateway IP Address from Configuration file!
Temp Sensor java RMI registry created.
Report to Gateway or Gateway Replica enter Teperature ...

```
- Terminal 4 (middle right):**

```
HeaterImpl java RMI registry created.
Need to Report the State Enter Y or N
Gate Way is Assigned to you

```
- Terminal 5 (bottom right):**

```
Read the IPAddress from the Configuration file
Motion Sensor java RMI registry created.
Gate Way is Assigned to you
Gate Way is Assigned to you

```