

Assignment

What is Ping-Echo and Redundancy?

Ping-Echo technique is used to monitor resource availability. Consider two machines “m1” and “m2”. Assume on “m1” some critical service “s1” is running. Service “s1” is so critical that it has to be available 99.999% of time, making “m1” a critical machine. So to check availability of “m1” and “s1” we can set up a monitoring machine “m2” which pings “m1” periodically and “m1” sends a signal (or echo) in response to ping. This echo tells “m2” that “m1” is active and service “s1” is also active (active = up and running fine). Since m1 is critical and we want high availability so we can add another machine “m2” which is running service “s2” in it. Services “s1” and “s2” are performing same tasks. Now user suppose “m1” or “s1” is down then the redundant machine “m2” takes the charge and serves the user, this increases availability.

Aim: You have to replicate ping-echo and active redundancy technique using concept of either VMs or Docker.

Instructions:

1. Create 3 VM's or Docker instances and install Linux OS in them (Let names of VM's or docker instances are **VM1**, **VM2** and **VM3**).
2. On VM1 create a service (named **S1**) by writing a program which generates some random numbers between 1 to 40 and add two numbers where numbers are entered using monitor interface implemented in VM3.
3. On VM2 create a service (named **S2**) by writing a program which generates some random number between 70 to 100 and add two numbers where numbers are entered using monitor interface implemented in VM3.
4. In VM3 design a monitor interface which contains **4 text fields** namely **Number1**, **Number2**, **Result** and **RandomNumber** and a **button** named **Calculate**. We can enter digits in **Number1** and **Number2** and on pressing **Calculate** button we get computed result along with random number generated by computing VM which is printed in text field **Result** and **RandomNumber** respectively.

[Note* you must design monitor interface in such a way that it sends parameters to VM1 if it is up and service S1 is up. If VM1 is down or service S1 is down then result must be computed by VM2 (assuming VM2 and its service S2 is up)]

This monitor interface also contains four bulbs namely **BV1**, **BVS1**, **BV2** and **BVS2** where colour of these bulbs denotes following state related to VM1 and VM2.

Bulbs	Colour	VM States
BV1	Red	VM1 is down
BV1	Green	VM1 is up
BVS1	Black	VM1 is up but service S1 is down
BVS1	Yellow	VM1 is up and service S1 is also up
BV2	Red	VM2 is down
BV2	Green	VM2 is up
BVS2	Black	VM2 is up but service S2 is down
BVS2	Yellow	VM2 is up and service S2 is also up

What to upload:

You have to upload your code for Services “S1”, “S2” and monitor interface.

You have to upload a one page report describing how you engineered this problem.

[Total Marks=10]

Marks Distribution is as follows:

2 Marks for Implementing VM’s and Services

3 Marks for Implementing Ping-Echo and Redundancy

5 Marks for Demo