

4. Simulate Load Balancing in Cloud using Round Robin Scheduling using Java

Objective:

To simulate **load balancing in cloud computing** using the **Round Robin algorithm**, where incoming tasks are evenly distributed across multiple virtual machines (VMs).

Concept Covered:

- Load Balancing
- Round Robin Task Assignment
- VM Resource Utilization

Algorithm / Steps:

1. Start the program.
2. Create a class Task with:
 - o Task ID
 - o Task execution time
3. Create a class VirtualMachine with:
 - o VM ID
 - o A list of assigned tasks
4. In the main method:
 - o Create a list of VMs (e.g., 3 VMs).
 - o Create a list of tasks (e.g., 10 tasks with varying execution times).
 - o Assign each task to a VM using **Round Robin** (cyclic allocation).
 - o Print assigned tasks per VM.
5. End the program.

CODE:

```
import java.util.*;  
  
class Task {  
    int id;  
    int executionTime;  
  
    public Task(int id, int executionTime) {  
        this.id = id;  
        this.executionTime = executionTime;  
    }  
}  
  
class VirtualMachine {  
    int id;  
    List<Task> assignedTasks = new ArrayList<>();  
  
    public VirtualMachine(int id) {  
        this.id = id;  
    }  
  
    public void assignTask(Task task) {  
        assignedTasks.add(task);  
    }  
  
    public void showTasks() {  
        System.out.println("VM " + id + " Tasks:");  
        for (Task task : assignedTasks) {  
            System.out.println(" -> Task ID: " + task.id + ", Time: " + task.executionTime + "ms");  
        }  
        System.out.println();  
    }  
}  
  
public class LoadBalancerSimulation {  
    public static void main(String[] args) {  
        // Step 1: Create Virtual Machines  
        int numberOfWorkstations = 3;  
        List<VirtualMachine> vms = new ArrayList<>();  
        for (int i = 1; i <= numberOfWorkstations; i++) {  
            vms.add(new VirtualMachine(i));  
        }  
  
        // Step 2: Create Tasks  
        int numberOfTasks = 10;  
        List<Task> tasks = new ArrayList<>();  
        Random rand = new Random();  
        for (int i = 1; i <= numberOfTasks; i++) {  
            tasks.add(new Task(i, rand.nextInt(1000) + 500)); // 500ms to 1500ms  
        }  
  
        // Step 3: Round Robin Load Balancing  
        System.out.println("--- Task Assignment using Round Robin ---\n");  
        int vmIndex = 0;  
        for (Task task : tasks) {  
            VirtualMachine vm = vms.get(vmIndex);  
            vm.assignTask(task);  
        }  
    }  
}
```

```

        System.out.println("Assigned Task " + task.id + " to VM " + vm.id);
        vmIndex = (vmIndex + 1) % numberOfVMs;
    }

    // Step 4: Show final task distribution
    System.out.println("\n--- Final Task Distribution ---\n");
    for (VirtualMachine vm : vms) {
        vm.showTasks();
    }
}
}

```

OUTPUT:

D:\Softwares\Java\Programs>javac LoadBalancerSimulation.java

D:\Softwares\Java\Programs>java LoadBalancerSimulation

--- Task Assignment using Round Robin ---

Assigned Task 1 to VM 1
Assigned Task 2 to VM 2
Assigned Task 3 to VM 3
Assigned Task 4 to VM 1
Assigned Task 5 to VM 2
Assigned Task 6 to VM 3
Assigned Task 7 to VM 1
Assigned Task 8 to VM 2
Assigned Task 9 to VM 3
Assigned Task 10 to VM 1

--- Final Task Distribution ---

VM 1 Tasks:

- > Task ID: 1, Time: 880ms
- > Task ID: 4, Time: 1014ms
- > Task ID: 7, Time: 1060ms
- > Task ID: 10, Time: 562ms

VM 2 Tasks:

- > Task ID: 2, Time: 629ms
- > Task ID: 5, Time: 751ms
- > Task ID: 8, Time: 1315ms

VM 3 Tasks:

- > Task ID: 3, Time: 678ms
- > Task ID: 6, Time: 1011ms
- > Task ID: 9, Time: 925ms

***** END *****