<table>
<tr><th colspan="2">SUBMITTED BY</th></tr>
<tr><td>Name:</td><td>Hamza Ali</td></tr>
<tr><td>Class:</td><td>BSSE (2021-2025)</td></tr>
<tr><td>Subject:</td><td>Cloud Computing</td></tr>
<tr><td>Semester:</td><td>8th</td></tr>
<tr><td>Department:</td><td>Software Engineering</td></tr>
<tr><th colspan="2">SUBMITTED TO</th></tr>
<tr><td>Teacher:</td><td><b>Dr. Muhammad Imran</b></td></tr>
</table>

Load Balancing Assignment Report

Task 1: Basic Cloud Simulation Setup

Data Center Configuration

- **One host** with the following specifications:
    - 4 CPUs (Processing Elements)
    - 8GB RAM
    - 1000GB storage
    - 10Gbps bandwidth

VM Configuration

- **VM1**: 2 CPUs, 4GB RAM, 1000 MIPS
- **VM2**: 1 CPU, 2GB RAM, 1000 MIPS

Cloudlet Configuration

- **Cloudlet 1**: 10,000 million instructions (assigned to VM1)
- **Cloudlet 2**: 20,000 million instructions (assigned to VM2)
- **Cloudlet 3**: 15,000 million instructions (assigned to VM1)

Simulation Results

Time-Shared Policy Results

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0    SUCCESS        0        0    10    0.1    10.1
    1    SUCCESS        0        1    20    0.1    20.1
    2    SUCCESS        0        0    15    0.1    15.1
```

Analysis of Results

Time-Shared Policy

- All cloudlets started execution immediately (at time 0.1)
- Resources are shared between cloudlets running on the same VM
- Cloudlet 2 (ID: 2) ran in parallel with Cloudlet 1 (ID: 0) on VM 0
- Average execution time: 15.0 seconds
- Higher overall resource utilization but each task has partial resources
- Some overhead due to context switching between tasks

Space-Shared Policy Results

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0    SUCCESS        0        0    8    0.1    8.1
    1    SUCCESS        0        1    16    0.1    16.1
    2    SUCCESS        0        0    12    8.1    20.1
```

Analysis of Results

Time-Shared Policy

- All cloudlets started execution immediately (at time 0.1)
- Resources are shared between cloudlets running on the same VM
- Cloudlet 2 (ID: 2) ran in parallel with Cloudlet 1 (ID: 0) on VM 0
- Average execution time: 15.0 seconds
- Higher overall resource utilization but each task has partial resources
- Some overhead due to context switching between tasks

Space-Shared Policy

- Cloudlets on the same VM executed sequentially
- Cloudlet 2 (ID: 2) had to wait for Cloudlet 1 (ID: 0) to finish before starting
- Individual tasks executed faster (8.0s vs 10.0s, 12.0s vs 15.0s)
- Average execution time: 12.0 seconds when looking at individual times
- Lower resource utilization but each task gets 100% of allocated resources
- No execution overhead from context switching

Task 2: Load Balancing Simulation

Overview

The second task implements a load balancing algorithm that migrates VMs between hosts based on utilization thresholds. This simulates a real-world scenario where cloud providers need to balance workloads across their infrastructure to optimize resource usage and maintain performance.

Setup Configuration

- **Data Center**: Contains 2 identical hosts (H1 and H2)
- **Host Specifications**:
    - 4 CPUs (PEs) each
    - 8GB RAM each
    - 1000GB storage each
    - 10Gbps bandwidth each
- **Initial Allocation**:
    - **Host H1**: 4 VMs (overloaded at 85% utilization)
    - **Host H2**: 1 VM (underutilized at 20% utilization)
- **Load Balancing Thresholds**:
    - Overload threshold: 80% CPU utilization
    - Underload target: Below 50% CPU utilization

Migration Algorithm

The algorithm works as follows: 1. Monitor host utilization 2. Identify overloaded hosts (utilization > 80%) 3. For each overloaded host, select a VM to migrate 4. Find a suitable target host with sufficient resources 5. Migrate the selected VM 6. Verify new utilization levels

Simulation Results

```
Host 0 initial CPU utilization: 85.00%
Host 1 initial CPU utilization: 20.00%

--- Starting Load Balancing ---
Host H1 (ID: 0) current utilization: 85.00%
Host H2 (ID: 1) current utilization: 20.00%
Host H1 is overloaded (> 80.00%)
Considering VM 2 for migration:
  - VM utilization contribution: 25.00%
  - H1 utilization after migration would be: 60.00%
  - H2 utilization after migration would be: 45.00%
Migrating VM 2 from Host H1 to Host H2
Migration complete
New utilization - H1: 60.00%, H2: 45.00%
```

Cloudlet Execution After Load Balancing

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0    SUCCESS        0        0    10    0.1    10.1
    1    SUCCESS        0        1    12    0.1    12.1
    2    SUCCESS        0        2    14    0.1    14.1
    3    SUCCESS        0        3    16    0.1    16.1
    4    SUCCESS        0        4    18    0.1    18.1
    5    SUCCESS        0        0    20    0.1    20.1
    6    SUCCESS        0        1    22    0.1    22.1
```

Load Balancing Results

```
========== LOAD BALANCING RESULTS ==========
Host ID | Before Migration | After Migration | Change
--------|------------------|-----------------|--------
0       | 85.00%           | 60.00%          | -25.00%
1       | 20.00%           | 45.00%          | +25.00%
```

Analysis of Results

The load balancing algorithm successfully: 1. Identified Host H1 as overloaded (85% utilization > 80% threshold) 2. Selected VM 2 for migration, which contributed 25% to H1's utilization 3. Migrated VM 2 from H1 to H2 4. Balanced the load between hosts: - H1 utilization decreased from 85% to 60% - H2 utilization increased from 20% to 45% - Both hosts now operate within optimal range

This demonstrates how dynamic VM migration can be used to: - Prevent performance degradation in overloaded hosts - Better utilize resources across the data center - Balance workloads without disrupting service - Improve overall system efficiency

Conclusion

This assignment successfully demonstrated two key aspects of cloud resource management:

1. **Allocation Policies**: We compared Time-Shared and Space-Shared policies, showing how each has distinct advantages depending on the workload characteristics. Time-Shared policies maximize parallelism and resource utilization, while Space-Shared policies optimize individual task performance and predictability.

2. **Load Balancing**: We implemented a threshold-based VM migration algorithm that effectively balanced workloads across hosts, demonstrating how cloud providers can dynamically adjust resource allocation to optimize performance and utilization.

The simulations provide valuable insights into cloud resource management strategies and their impact on system performance. These techniques are fundamental to building efficient, scalable, and responsive cloud computing environments.