A black and white logo

AI-generated content may be incorrect.THE OPEN UNIVERSITY OF SRI LANKA

FACULTY OF ENGINEERING TECHNOLOGY

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

BACHELOR OF SOFTWARE ENGINEERING (HONOURS)

EEX5362 –PERFORMANCE MODELLING

**Deliverable 01**

Name: P. Komathy,

SNumber: S92065503,

Regi Number: 321425503.

System Chosen: **Cloud Computing Resource Allocation System**

**1. System Overview:**

The selected system for performance modeling is a Cloud Computing Resource Allocation System. This system represents how incoming user requests (jobs) are handled by a pool of servers (virtual machines/containers) in a cloud data center from arrival through processing to completion.

Key components and resources include:

* User requests / jobs arriving with diverse sizes and priorities.
* Front-end load balancer that routes requests to available servers.
* Compute servers (VMs / containers) that process jobs.
* Shared storage / network I/O that may introduce delays.
* Auto-scaling controller that adds/removes servers based on utilization thresholds.
* Monitoring and scheduling policies that influence allocation and placement decisions.

because workloads are dynamic and resources are finite, the system exhibits queueing, resource contention (CPU, memory, I/O), and variable latency.

**2. High-Level Problem Definition:**

Cloud service providers must dynamically allocate computing resources to handle incoming user requests efficiently.

However, unpredictable workloads and limited capacity often cause bottlenecks that increase response time, reduce throughput, and violate service-level agreements (SLAs).

**Problem Statement:**

To model and analyze the performance of a cloud computing resource allocation system to identify bottlenecks, measure response time and utilization, and evaluate optimization strategies such as auto-scaling and improved scheduling.

**3. Dataset Description:**

Simulation and analysis use synthetic or real traces with the following attributes:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Request ID | Unique identifier for each request/job |
| Arrival Time | Time the request enters the system |
| Request Type / Priority | Classifies the request (compute-intensive, I/O-intensive) |
| Service Time | Duration the request occupies a server |
| Queue Wait Time | Time waiting before being assigned to a server |
| Assigned Server | Server or VM that processed the request |
| Completion Time | Time the request is completed |
| CPU Utilization | Average CPU load during request processing |
| Memory Usage | Memory consumed by the request |
| Response Time | Completion Time − Arrival Time |

This attributes enables quantitative measurement of system performance metrics such as **average response time, throughput, queue length, and server utilization.**

**4. Performance Objectives:**

The main performance objectives of the Cloud Computing Resource Allocation System are to evaluate and improve the efficiency of cloud resource management.

The study focuses on:

* **Minimizing response time:** Reducing the time between task submission and completion to improve user experience.
* **Maximizing throughput:** Increasing the total number of tasks processed within a given time period.
* **Identifying bottlenecks:** Detecting performance-limiting components such as overloaded servers or inefficient scheduling.
* **Optimizing resource allocation:** Ensuring balanced and efficient use of CPU, memory, and storage resources across virtual machines.
* **Improving scalability:** Assessing system performance under varying workloads and resource availability.
* **Enhance Scalability and Elasticity :** Evaluate how system performance changes as the number of incoming tasks or users increases and resources scale dynamically.

**5. Tools and Techniques:**

Recommended modeling and analysis stack:

* **Python (SimPy) -** Discrete-Event Simulation of arrivals, queues, and server service processes.
* **NumPy -** Random variate generation and statistics.
* **Pandas -** Data capture and post-processing.
* **Matplotlib -** Visualization (latency CDF, utilization over time, queue length time-series).
* **Tabulate :** Table formatting for console output
* **Queueing Theory** (M/M/c, M/G/c approximations) **-** Analytical baselines for validation.
* **Optional:** Cloud billing models to compute cost implications of scaling.

1. **Expected Outcomes:**

By modeling the cloud resource allocation system the project aims to:

* Quantify response-time vs. load relationships and identify capacity limits (maximum sustainable arrival rate for target SLA).
* Evaluate scaling policies to trade off cost and SLA compliance.
* Identify bottlenecks (CPU-bound vs. I/O-bound) and suggest targeted mitigations (caching, vertical scaling, request shedding).
* Provide recommendations for right-sizing baseline capacity and auto-scaling parameters to minimize cost while meeting performance objectives.
* Deliver visualizations and tables (latency CDFs, utilization plots, throughput vs. load) suitable for executive decision-making.