

Discrete event simulation

Distributed Computing assignment



February 10, 2025

University of Genova

Submitted by: Abdul Jalil Rohi (5684362)

Abstraction

This project simulates and compares the behavior of different queuing systems. The main goal is to analyze how different scheduling and load balancing techniques affect the performance of the system, especially in terms of the average time a job spends in the system (W).

The project simulates various queuing systems, including:

* **M/M/1 queue:** A single server queue with memoryless arrival and completion delays, following an exponential distribution.
* **M/M/N queue:** A multiple server queue with memoryless arrival and completion delays, following an exponential distribution.
* **M/M/N with Supermarket Model:** A load balancing approach where a job is assigned to the least loaded queue among a randomly sampled subset of queues.
* **M/M/N with Round-robin Scheduling:** A preemptive scheduling algorithm where each job is assigned a time slot for execution, and if it doesn't finish within that time, it's preempted and put back in the queue.

**The Assignment summarizes the following:**

* **Goal:** Simulate and analyze scheduling and load balancing in a queuing system.
* **Scheduling Algorithms:** FIFO, Round Robin.
* **Load Balancing:** Supermarket Model.
* **Job Size Distribution:** Exponential, Weibull.
* **Metrics:** Average waiting time (w), Queue length distribution.
* **Data Logging:** λ, μ, max\_t, n, d, w, queue\_size.
* **Reporting:** Can be adapted to focus on **average waiting time** and **queue length distribution**.

**Introduction**

The aim of this project is to simulate and analyze the performance of different scheduling algorithms and load balancing techniques in a queuing system. Specifically, the project focuses on evaluating the effectiveness of the First-In, First-Out (FIFO) and Round Robin (RR) scheduling algorithms, along with the supermarket model for load balancing. The key performance metrics analyzed are the average waiting time (w) and the queue length distribution.

These metrics are crucial for evaluating the efficiency and effectiveness of scheduling strategies because they provide insights into how well the system manages job queues and resource allocation. The average waiting time reflects the overall responsiveness of the system, indicating how long jobs typically wait before being processed. The queue length distribution provides a more detailed view of the system's behavior, showing how jobs are distributed across queues and how often queues of different lengths occur. By analyzing these metrics, we can assess the impact of different scheduling strategies on the system's performance and identify the most suitable approach for various scenarios.

**Supermarket Model Implementation**

The supermarket model is implemented using the ***supermarket\_decision*** method in the Queues class. This method samples a subset of queues (the number of queues sampled is determined by the parameter d) and selects the queue with the shortest length. This selection process aims to distribute incoming jobs more evenly across the queues, thereby balancing the load and potentially reducing waiting times.

**Round Robin Scheduling Implementation**

Round Robin scheduling is implemented by assigning a fixed time slice, called a quantum, to each job. If a job doesn't complete within its quantum, it's interrupted and returned to the queue, allowing other jobs to execute. This preemptive approach ensures fairness and prevents starvation, especially in scenarios with varying job sizes. The quantum parameter controls the length of the time slice, influencing the balance between fairness and overall throughput.

**Simulation Configurations**

To evaluate the performance of the scheduling algorithms and load balancing techniques, simulations were run with various configurations:

* **Varying Arrival Rates (lambd)**: The arrival rate, which represents the frequency of new jobs entering the system, was varied to observe the system's behavior under different load conditions. The values used for lambd were 0.5, 0.7, 0.9, 0.95, and 0.99.
* **Different Sample Sizes (d)**: The number of queues sampled in the supermarket model (d) was varied to assess its impact on load balancing and waiting times. The values used for d were 1, 2, 5, and 10.
* **Different Time Quantum Values**: The time quantum for Round Robin scheduling was adjusted to explore its influence on fairness and overall system efficiency. The values used for the time quantum were 0.1, 0.5, 1, 2, and 5.
* **Different Job Size Distributions (Weibull Shapes)**: The Weibull distribution with varying shapes was used to simulate different job size distributions, allowing for the evaluation of the scheduling algorithms under various workload characteristics. The shapes used were 0.5, 1, 1.5, 3, and 3.75.

**Data Collection and Organization**

The simulations were run for a fixed duration (max\_t), and data on the average waiting time (w) and queue length distribution (***queue\_size***) was collected. The ***w*** value represents the average time a job spends in the system, from arrival to completion. The queue\_size data captures the lengths of the queues at different points in time, providing insights into the distribution of jobs across the queues. This data was organized into separate CSV files for each configuration, with each file containing columns for lambd, mu, max\_t, n, d, w, and queue\_size.

 **Queue Length Distribution**: This metric shows the probability distribution of the number of entities (e.g., customers, jobs) in the queue at any given time. It helps in understanding how often the queue reaches certain lengths, which can be crucial for capacity planning and resource allocation.

 **Average Time Spent in the System**: This metric calculates the average time an entity spends in the entire system, from arrival to departure. It includes both the waiting time in the queue and the service time. This metric is useful for evaluating the efficiency of the system and identifying potential bottlenecks.

A blue rectangular bars with white text

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

A group of graphs with different colored lines

AI-generated content may be incorrect.