Project 2
Instructor: Prof. Neeraj Mittal
Author: Malay Virendra Bhavsar (MXB230055)

Project Overview

This individual project focuses on developing a distributed mutual exclusion service by implementing the algorithm proposed by Roucairol and Carvalho. The goal is to provide two essential functions, cs-enter() and cs-leave(), to facilitate proper synchronization of processes that need to access shared resources in a concurrent system. The project is implemented using Java (Version 22.0.2), with a strong emphasis on socket programming and distributed communication techniques to handle inter-process communication. The system will be executed exclusively on the dcXX.utdallas.edu machines, ensuring that all nodes in the network can participate in the mutual exclusion process in a controlled environment.

Evaluation Summary

For the purpose of this experiment, the number of nodes in the distributed system, denoted as *n*, will be fixed at 10. The inter-request delay, *d*, which determines the time between consecutive requests made by the nodes, will be set to a constant value of 20. To study the system's behavior under varying loads, the critical section contention factor, *c*, will be varied between 500 and 750, with increments of 50 (i.e., 500, 550, 600, 650, 700, 750).

At each value of *c*, several key performance metrics will be measured to evaluate the system's efficiency and responsiveness. These metrics include the mean message complexity (which indicates the average number of messages exchanged for a critical section request), response time (the time taken from the initiation of a request to its completion), and system throughput (the rate at which requests are successfully satisfied per unit of time). Each experimental configuration will involve at least 5 independent runs to ensure reliability, and each node will generate a minimum of 500 requests for critical sections during the experiment. This ensures that the system is subjected to sufficient load for meaningful analysis, allowing for a comprehensive understanding of how the system performs under different conditions of contention and request rates.

Evaluation Results

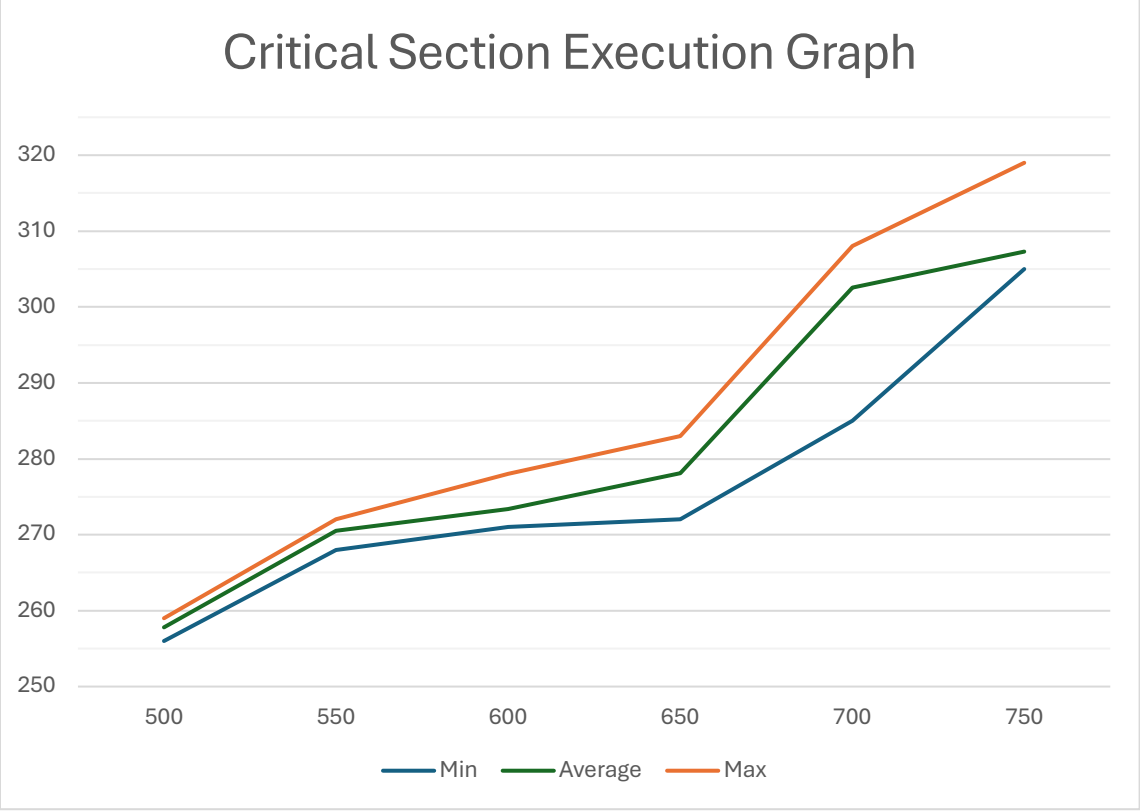
1. **Critical Section Execution Average Time**

The graph titled *Critical Section Execution Average Time* illustrates the relationship between the number of critical sections and the average execution time for each critical section. The x-axis represents the number of critical sections, while the y-axis shows the average time taken for their execution.

Three lines are plotted in the graph:

- **Min**: This line shows the minimum execution time observed for each number of critical sections.
- **Max**: This line represents the maximum execution time recorded for each number of critical sections.
- **Average**: The average execution time is shown as the middle line, providing an overall sense of how the execution time changes as the number of critical sections increases.

This graph provides a clear visual comparison of the variability in execution time, highlighting trends such as the increase in average execution time with a growing number of critical sections, and how it compares to the extremes (min and max times).



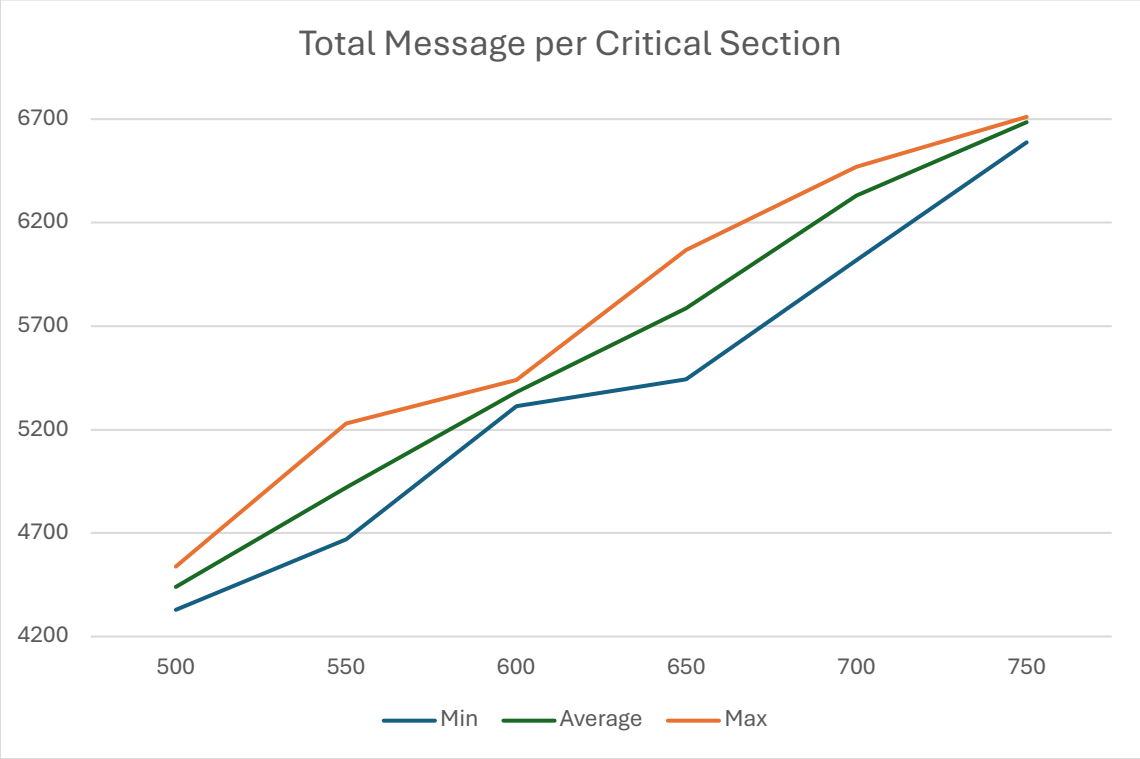
2. **Total Messages per Critical Section**

The graph titled *Total Messages per Critical Section* shows how the number of messages varies with the number of critical sections. The x-axis represents the number of critical sections, while the y-axis indicates the total number of messages associated with each critical section.

There are three lines plotted in the graph:

- **Min**: The minimum number of messages observed for each number of critical sections.
- **Max**: The maximum number of messages recorded for each number of critical sections.
- **Average**: The average number of messages, providing an overall trend of message count per critical section.

This graph helps visualize the distribution and variation of messages across different critical sections, highlighting both the variability and overall trends in message passing as the number of critical sections increases.



3. **Critical Section Throughput**

The graph titled *Critical Section Throughput* illustrates the decreasing trend in throughput as the number of critical sections increases. The x-axis represents the number of critical sections, while the y-axis shows throughput, likely measured as operations completed per unit of time.

Three lines are shown on the graph:

- **Min**: The minimum throughput recorded for each number of critical sections.
- **Max**: The maximum throughput observed for each number of critical sections.
- **Average**: The average throughput, representing the overall trend for each corresponding number of critical sections.

As the number of critical sections increases, throughput decreases, which suggests that the system experiences diminishing performance with a higher load of critical sections. The gap between the minimum, maximum, and average lines further emphasizes the variability in throughput as more critical sections are added, with the average throughput steadily declining as system resources are potentially strained.