CE 6378 Section 0U1

Project 2

Report

Perry Wu

Psw101020

After running the average with multiple data points on the project, I was able to find some performance metrics with regards to the general protocol implemented. Below are some tables that describes the findings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | 7 | 8 | 9 | 10 |
| Throughput # msgs processed at a time | 15 | 16 | 18 | 20 |
| Messages Complexity(# msgs) | 72 | 84 | 96 | 108 |
| Time (ms) | 3286.583333 | 3609.5 | 5302.75 | 6875 |

In this table, we see that the time increases as the number of nodes increase. This falls in line with the hypothesis that each node must send a set number of messages. This combined with the fact that each CS enter method is blocking, contributes to the overall factor of average time taken to execute.

Also, we see that the number of messages for now increases with respect to the number of nodes found in the table. This falls in line with the fact that each node must send messages to release, request, and execute critical sections.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cs wait | 10 | 20 | 30 | 40 |
| Throughput | 15 | 15 | 15 | 15 |
| Messages Complexity | 48 | 48 | 48 | 48 |
| Time | 1496.125 | 1930.5 | 1497.75 | 1798.75 |

Overall, we can see that the cs-wait time doesn’t seem to impact much on the time. And throughput and message complexity is held constant throughout since no other variable has changed. CS Wait doesn’t seem to have much impact compared to CS Execution time, which is the time needed to remain in critical section.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cs execution time | 10 | 20 | 50 | 100 |
| Throughput | 15 | 15 | 15 | 15 |
| Messages Complexity | 48 | 48 | 48 | 48 |
| Time | 1496.125 | 1945.5 | 2349 | 3893.375 |

In the above table, as we increase cs execution time, it falls in line with the expectation that the time will also increase. As we see in the plot there can be an instance of an exponential function forming as we increase the execution time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| num messages | 2 | 4 | 10 | 20 |
| throughput | 15 | 15 | 15 | 15 |
| messages Complexity | 48 | 96 | 240 | 480 |
| Time | 1496.125 | 4545.438 | 4639.6 | 4689.7 |
|  |  |  |  |  |

The last two graphs shows the relation between the number of messages that needs to be sent, i.e. number of critical sections generated versus the actual message complexity. This shows the fact that that the number of messages nodes must send dramatically increases the message complexity, and the overall time.

In conclusion. We see the two most impactful attributes of this protocol is the number of messages needed to be sent, as well as the number of nodes. These two numbers dictate the largest amount of increase in time, and message complexity.