AOS_P2 Project Report

Team Member

Shen Zhang sxz162330
Qi Wang qxw170003

Testing Methodology

Use the fact that all the dc machines from school are actually the same and they share the same local time. So we could print out the time when each process enters and leaves the critical section and use the Tester.java to check whether they overlap.

Performance Analysis

	#nodes	inter-request delay	cs-execution time	totalMsg / #request satisfied	Time from making request to leaving CS	#request satisfied / total Execution Time
	n	**d**	**c**	**Avg Message Complexity**	**Avg ResponseTime (ms)**	**Avg Throughput**
n as variable	3	10	10	7.90	1058.67	2.91
	5	10	10	16	1852.12	2.88
	10	10	10	72.8	5587.30	1.78
d as variable	5	10	10	16	1852.12	2.88
	5	20	10	16.1	2259.34	2.15
	5	30	10	16.3	2693.88	1.94
c as variable	5	10	10	16	1852.12	2.88
	5	10	20	16.04	2935.77	1.72
	5	10	30	16	3125.59	1.59

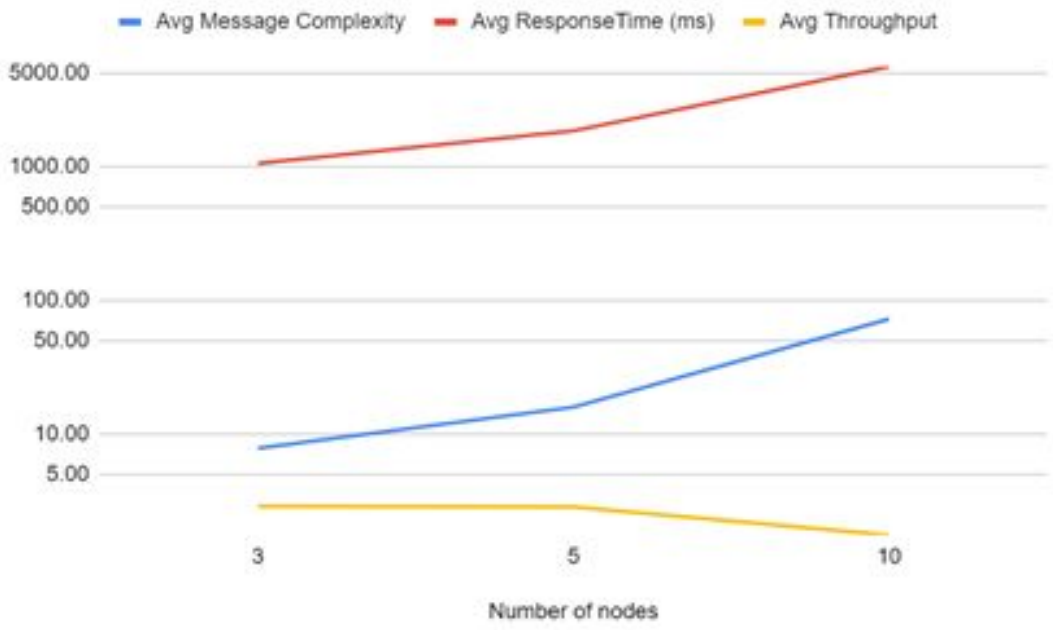

Fig.1 Performance according to number of nodes (log scale)

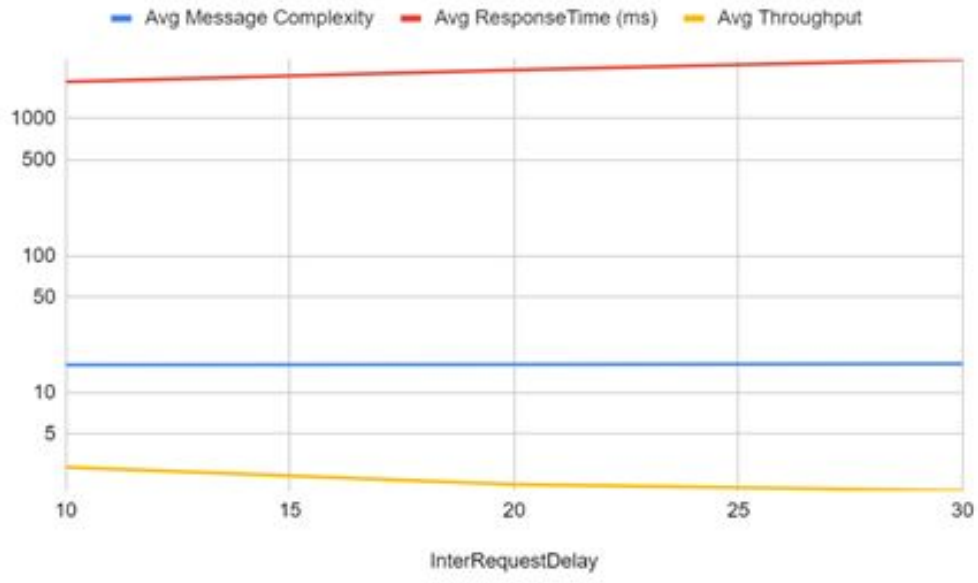

Fig. 2 Performance according to the inter-request delay (log scale)

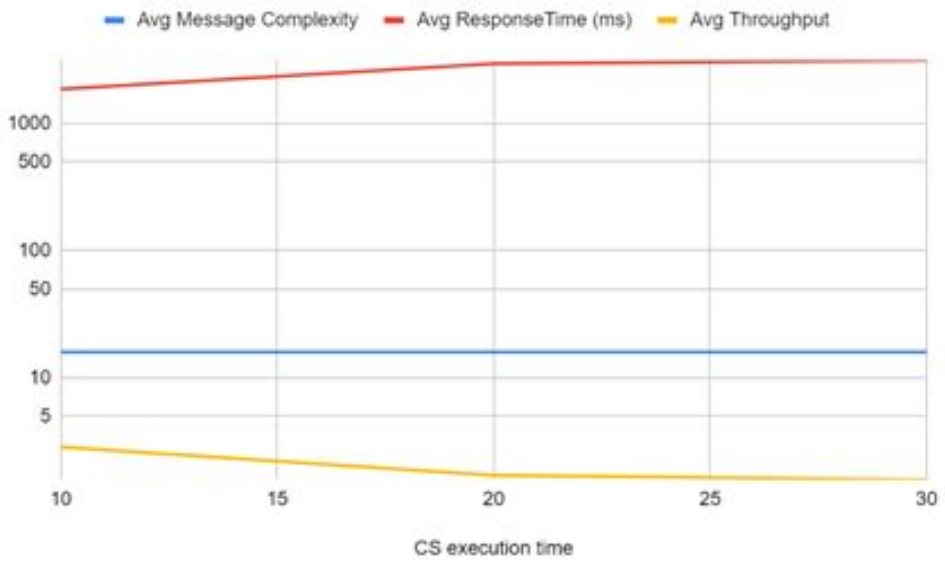

Fig. 3 Performance according to the CS execution time (log scale)

Observation

From the table and plotted charts above, we can see that:

1. With the number of nodes in the system increasing, both message complexity and response time will increase, but the throughput will decrease.

2. With the inter-request delay increase, the message complexity won't change, the response time will increase, while the throughput will decrease.

3. With the CS execution time increase, the message complexity won't change, the response time will increase, while the throughput will decrease.

Discussion

The result matches our expectation according to the Roucairol-carvalho algorithm.

Through the observation, we confirmed that the message complexity only affected by the number of nodes in the system, won't be affected by the other 2 parameters. Response time will increase along with all the 3 parameters, which makes sense. Since with more nodes, each node needs to communicate with more neighbors, thus sending out more messages, increase of inter-request delay and CS execution time will also affect the total time of execution, leading to the decrease in the throughput. Throughput, on the contrary, will decrease with the increase of all 3 parameters. Because with the same number of requests made, we need more time to

execute with more nodes and longer d&c, which means more unit time for each node to process, which leads to the decrease of throughput.