Logical Clocks Experiments

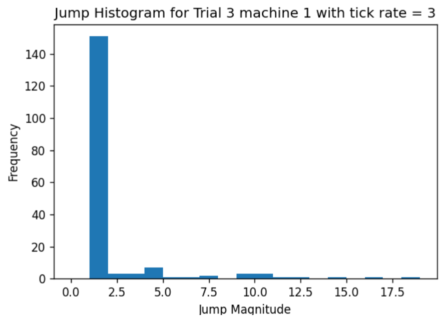
In our analysis of the performance of our scale model logical clock system, we examined the average/standard deviation and maximum jump magnitude as well as the maximum amount of drift and maximum queue length over several different trials. We calculated the drift by taking the 1/rate * logical clock time - # of seconds elapsed in system time (e.g. comparing the seconds elapsed in machine time to the seconds elapsed in real time).

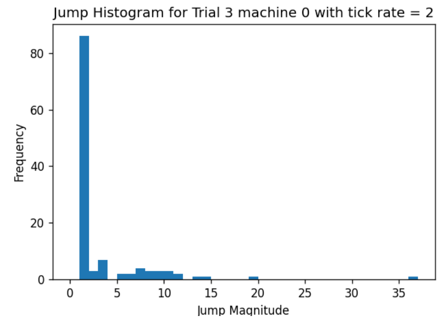
The first set of five trials ran three machines for a minute with the default tick range of (1, 6) and default probability of an internal event at 70%. The statistical results from these runs are below:

Trial	Tick Rate	Mean Jump	Stdev Jump	Max Jump	Max Drift	Max Queue Length
0	5	1.191	0.723	6	11.4	1
0	6	1.0	0.0	1	0.0	1
0	5	1.187	0.683	5	11.2	2
1	4	1.0	0.0	1	0.0	0
1	2	1.916	2.582	15	54.5	1
1	1	3.797	4.341	16	165.0	1
2	6	1.0	0.0	1	0.0	1
2	6	1.0	0.0	1	0.0	1
2	5	1.194	0.598	5	11.6	1
3	2	2.933	4.609	37	115.0	1
3	3	1.944	2.802	19	56.33	1
3	6	1.0	0.0	1	0.0	0
4	6	1.0	0.0	1	0.0	1
4	5	1.191	0.543	5	11.4	1
4	6	1.0	0.0	1	0.0	1

We can see that when the clock rates are closer together, the magnitude of jumps (both maximum and average) and their variance (standard deviation) are lower (e.g. in trials 0, 2, and 4, as opposed to trials 1 and 3). We also see that the jumps (maximum and average) are larger when the tick rate is relatively low compared to the maximum tick rate of a machine in the trial (e.g. comparing machine 2 and 0 in trial 1, which have tick rates of 1 and 4, respectively). We don't see as much of a trend with queue size, except that when one machine moves much faster than the others, it is likely to never have messages waiting on the queue. We see a similar trend as jumps with drift, where the maximum drift is larger when there is a higher variance in tick rates. This makes sense, as a machine performing 4 actions / second communicating with a machine that performs 1 action / second would push the logical clock value of the slower machine far beyond the actual number of events for that machine.

We can also examine histogram plots of jump magnitude. In each trial, we noted that the machine with the maximum rate never had any jumps over 1, which is the expected behavior. Additionally, the largest tick rate also had the We also noted that the frequency decreases exponentially over magnitude of the jump, as shown below. More plots can be found in our /plots/ directory.





We then ran a set of three trials with three machines for a minute, changing the tick range to (1, 3) and allowing the probability of an internal event to be 40%. The statistical results from these runs are below:

Trial	Tick Rate	Mean Jump	Stdev Jump	Max Jump	Max Drift	Max Queue Length
0	3	1.0	0.0	1	0.0	1
0	3	1.0	0.0	1	0.0	1
0	2	1.479	0.986	6	28.5	2
1	2	1.479	1.215	9	28.5	1
1	3	1.0	0.0	1	0.0	1
1	2	1.445	1.143	7	26.5	1
2	2	1.496	1.129	8	29.5	1
2	3	1.0	0.0	1	0.0	0
2	2	1.487	1.282	9	29.0	2

With our new set of experiments, we see that the variance in all of our variables decreased. This makes sense given that the difference between tick rates is now smaller, meaning there is less variation in the speed at which a machine is reading and sending messages. The trend in queue size is not as clear, which makes sense – while all the machines read messages off the queue more often, all of them are also sending messages more often. The logical clock values of the largest tick rates still increment normally (by 1), with the logical clock values of the smaller tick rates incrementing with slightly larger jumps.

Overall we see when our tick rate is the same then the experimental values are close to the expected values. As the tick rates between each machine diverge we also see larger jumps in the logical clock (large adjustments for each cycle) for machines with smaller tick rates. This is due to the fact that slower tick rates receive more messages than larger tick rates in between process runs and therefore need to update more frequently. Comparing machines with larger and smaller tick rates we see that machines with larger tick rates behave more normally and experience less drift, smaller queue sizes (since they process messages faster and receive fewer), and tend to increment by single jumps, rather than large logical clock jumps like we see in the smaller tick rates.