

Ben Steves

CS 374

Prof Adams

9-14-21

Analysis - Message Passing

For this week's project, runtimes for the message passing program increased as the number of processes increased. On the lab computers, as the number of processes double, the time also essentially doubles, though not perfectly. For example, in the Unix lab - going from 32 to 64 to 128 to 256 processes, the runtime was 0.0259s, 0.0485s, 0.1042s, and 0.3312s respectively. On Borg, runtimes increase at a much slower rate, and runtimes for all numbers of processes are quicker than they were on the Unix lab. Differences really started to show when the program needed a larger number of processes - for 128 and 256 processes, Borg ran the program at 0.0096s and 0.0137s, both being much faster than 0.1042s, and 0.3312s for the same number of processes on the lab machines.

There are some key differences in the runtime trends from both project1 and project2. In last week's project, we were distributing multiple processes across a for loop, and more processes sped up the program as it was running in parallel with the other processes. The program in project 2 slowed with more processes - increasing the number of processes also increased the time the program took to move around the ring. More interesting, maybe, is that Borg was always faster with the message passing program, while it was not always such with the circuitSatisfiability program. The cores on Borg are more numerous, more powerful, and have

more data storage than those from the Unix lab which makes a difference in the runtime for the message passing program.