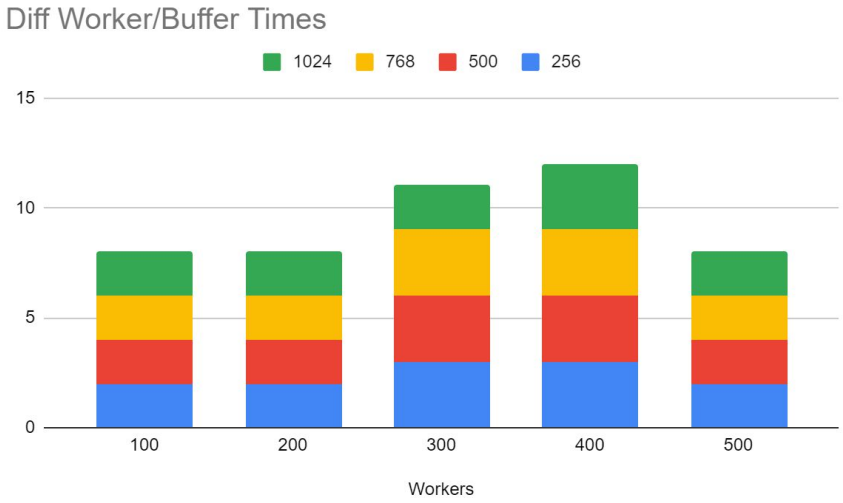
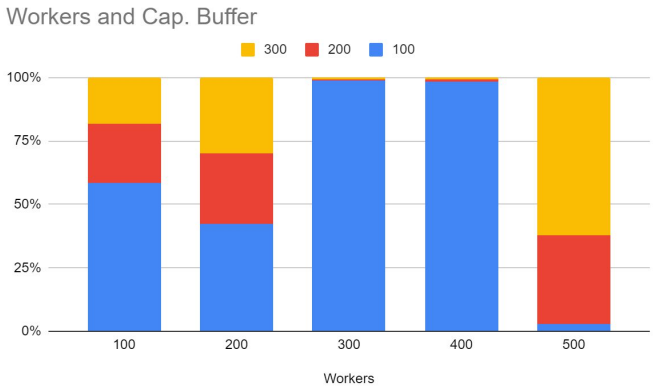
Bobby Vielma
UIN: 426007346

PA4 Report



The chart above represents the different times taken when different buffer and worker thread sizes are chosen. I chose to represent this data as a stacked bar chart since the times were generally within a few microseconds within each other and could not be differentiated visually with a regular bar chart. Some performance differences can be seen, especially within the 300 and 400 size worker threads. This could be due to having a greater request buffer size and more information being able to be processed at once. Some of this variance is also due to the conditions the system was under when the program was called. The other worker thread sizes don't see as much variance, and also did not take as long as the 300 and 400 sizes. This could be due to some sizes being more optimal for our program and not being affected much by a changing buffer size. For instance, if 100 worker threads were already working as fast as possible under a 256 size request buffer then increasing the request buffer will not speed up the program because the workers can't keep up. I wouldn't say we see any linear scaling with any parameters.



This chart is for the capacity buffer and number of workers. I used a stacked bar chart out of percentages because the times were so small that they were not visible. The threads with 500 workers took up to 41 seconds to run, making the bounds of the chart large compared to the microseconds the other threads took. I believe the reason for the amount of time it took for the 500 threads was because my laptop was being slowed down by other applications and the sheer amount of threads happening at once ended up slowing the process down instead of speeding it up. The other threads saw visible decreases in the amount of time it took them to complete the task when the capacity buffer was increased in size. I would say for the most part there is linear scaling, with a few outliers.