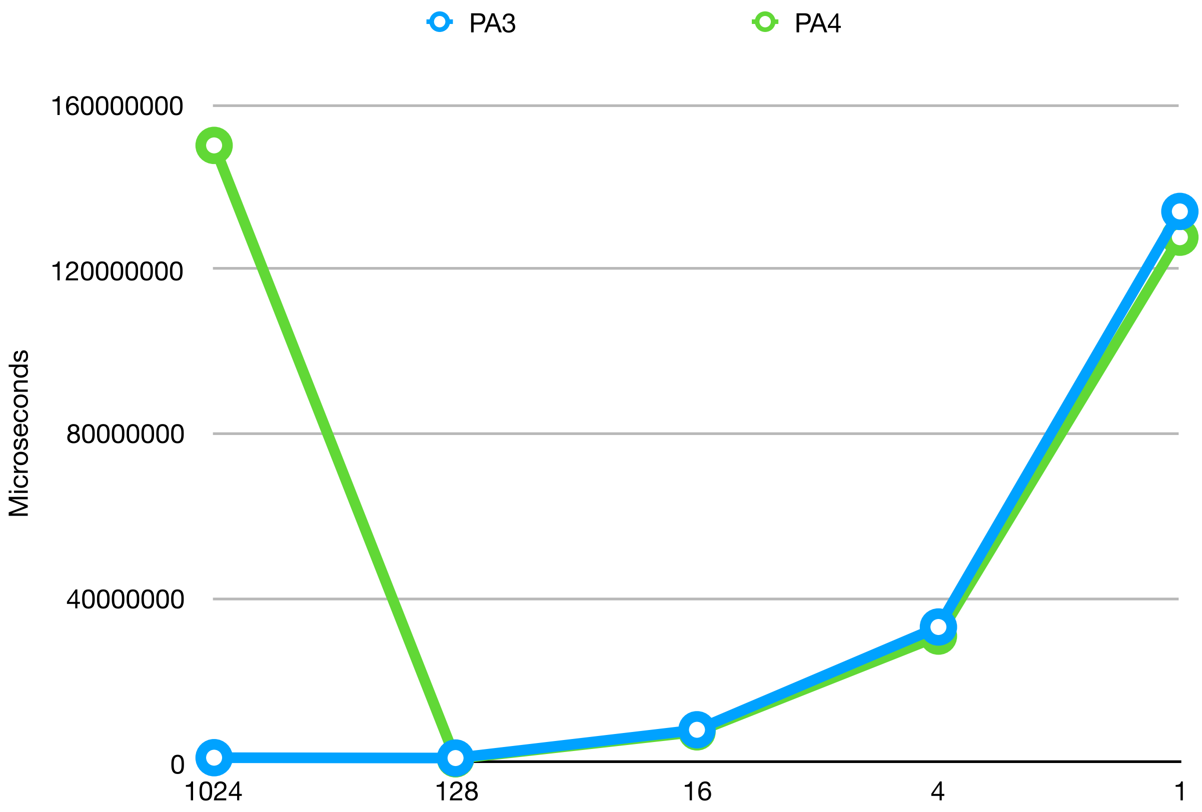
1. **Present a brief performance evaluation of your code. If there is a difference in performance from PA3, attempt to explain it. If the performance appears to have decreased, can it be justified as a necessary trade-off?**

My PA3 seems to be faster on average, with my PA4 having a gigantic spike at the end. I think this might be due to other processes on my computer rather than PA4 itself.

I don't know what is meant by "necessary trade-off", from my understanding PA3 doesn't loose anything in comparison to PA4, so if its faster then it is just better.

PA3 is likely faster or similar because is doesn't have as many printout statements as PA4, and it doesn't have as much data to go through (doesn't have separate stat threads).

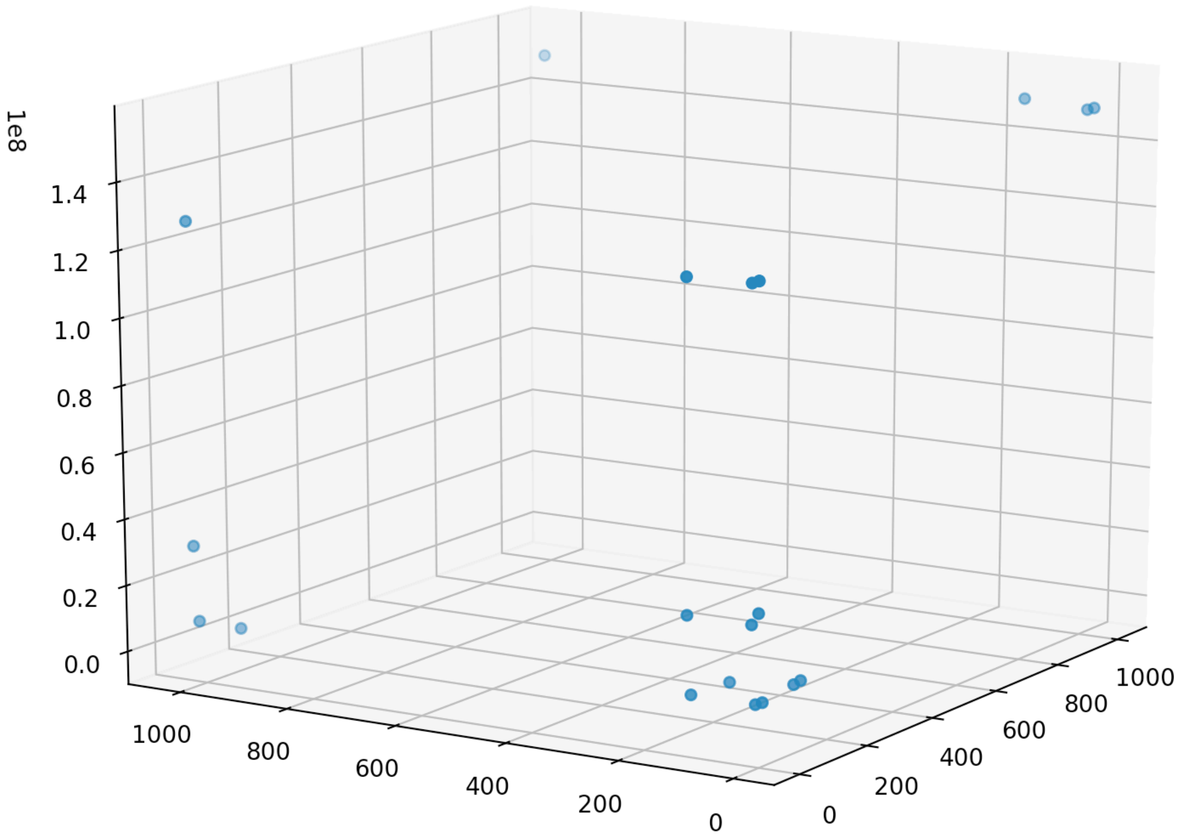


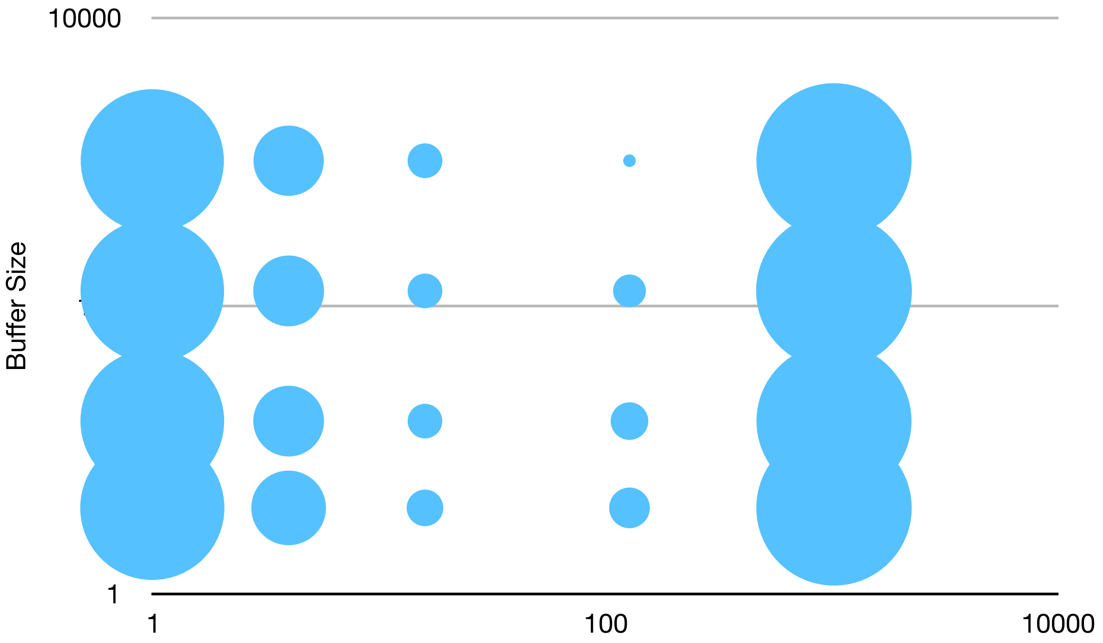
2. Make two graphs for the performance of your client program with varying numbers of worker threads and varying size of request buffer (i.e. different values of “w” and “b”) for n = 10K at the minimum. Discuss how performance changes (or fails to change) with each of them, and offer explanations for both.

Here is a 3D plot of both at the same time. Height is time in microseconds, left is the size of the buffer, and right is the number of worker threads.

Increasing the number of worker threads up to the number of cores heavily improves performance, and then after that amount it starts to decrease performance.

Increasing the buffer seems to help, but only up to a point. A really small buffer seems to be able to hurt performance but a large one doesn’t seem to make an impact. This is probably because at small values its the bottleneck, but once it’s not the bottleneck it doesn’t make a difference.





Here is a chart showing the same data, on a logarithmic scale, with the circle size being the time. There is a optimal spot around 100 threads and 1000 buffer size.