PA5: High Concurrency without many Threads

Brandon DeAlmeida

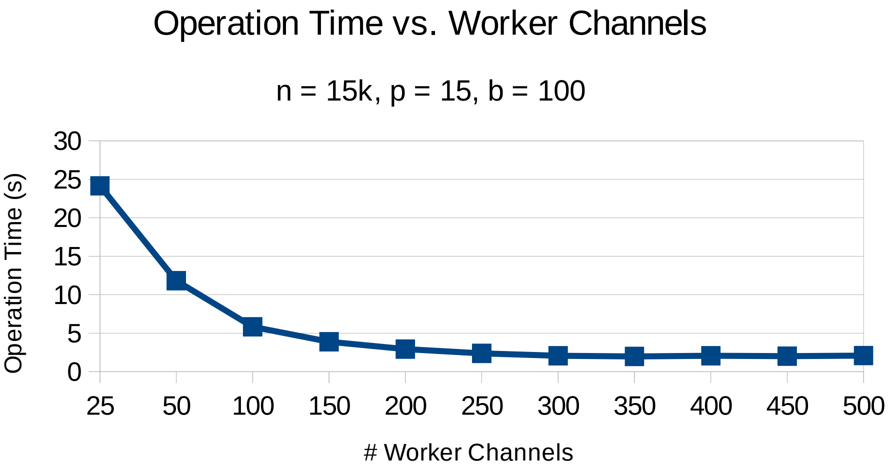
CSCE 313-508

Instructor: Tanzir Ahmed

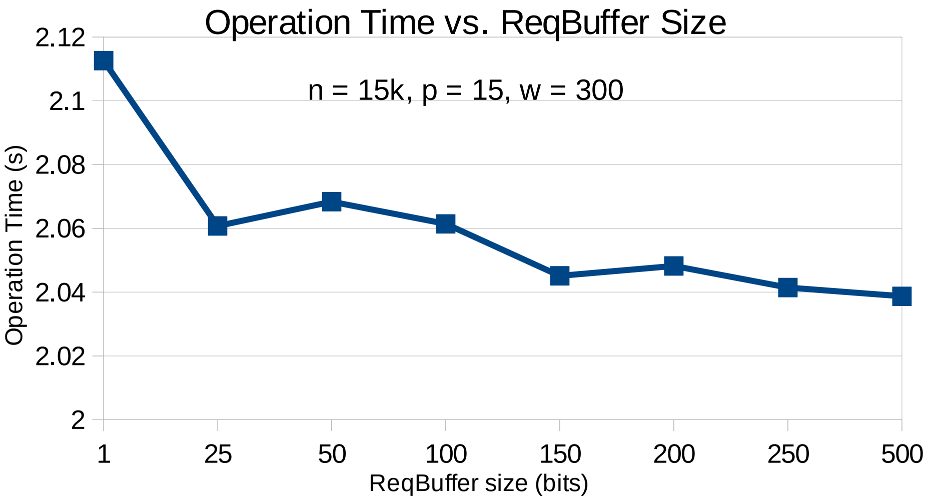
TA: Feras Khemakhem

Data:

Data Requests:

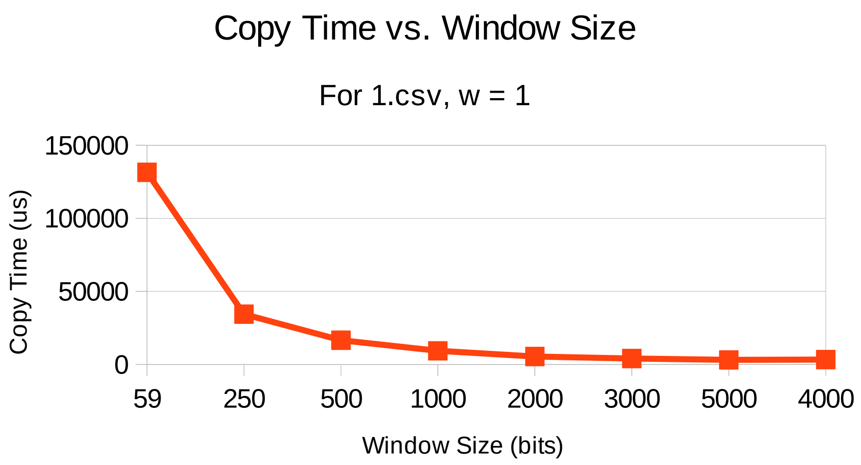


As before, each data point above was the average run-time of that test case 5 times. This implementation sees about the same time as my PA4 with only taking an average of ~2.59% off each test case, however there is a decrease in time across the board. This means that we see about the same logarithmic curve as before where increasing the number of worker channels benefits the run time until about 350 worker channels.

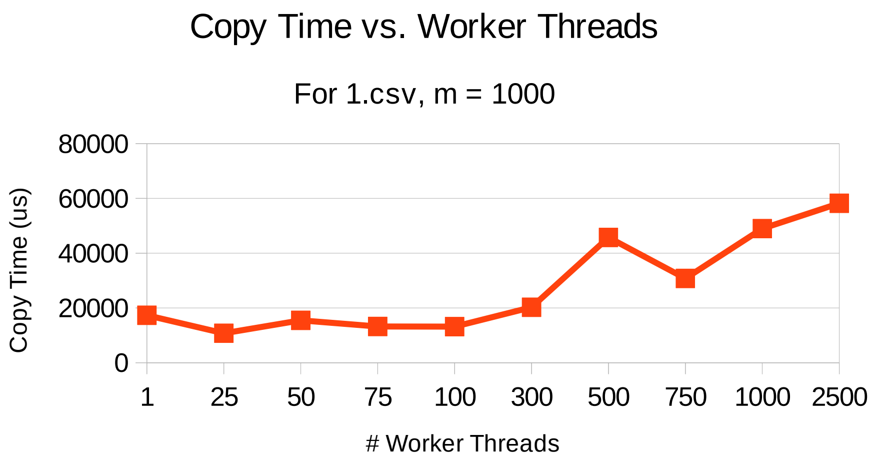


A relatively linear curve was found while increasing the request buffer size, the only large deviation can be seen after jumping up from a very little size (<25 requests). This shows that the number of worker threads tend to be the larger factor in the operation time of copying these data points. A likely reason for this trend is that since the bounded request buffer is constantly being emptied and filled by both the worker thread and patient threads (assuming that w <= p*n) it likely doesn’t generally stay full for long enough to slow down performance.

File Requests:



While the general shape of this curve carried over from PA4 to PA5, I actually saw about a 2.97% increase in the amount of time it took to copy a file using a single worker thread. Here we can see a good increase in performance by increasing the window size until about 3000bits.



As before, here we see a relatively flat trend for the worker channels until about 100 when the run time starts to increase.