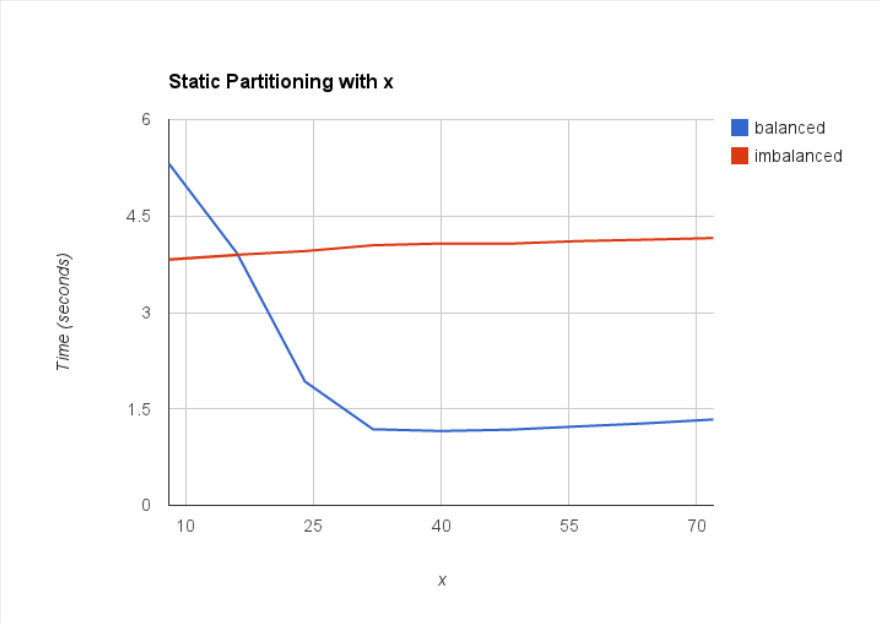
Sharon Kuo
Joseph Woo

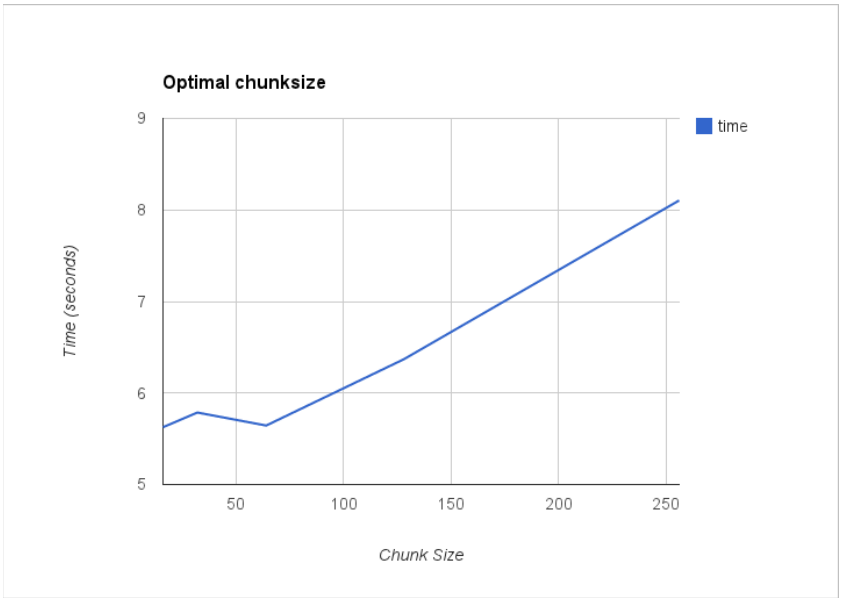
Program 3

In this program, we explored the effect changing x had on the time. Given all other parameters the same, we varied x to find the optimal value for x.



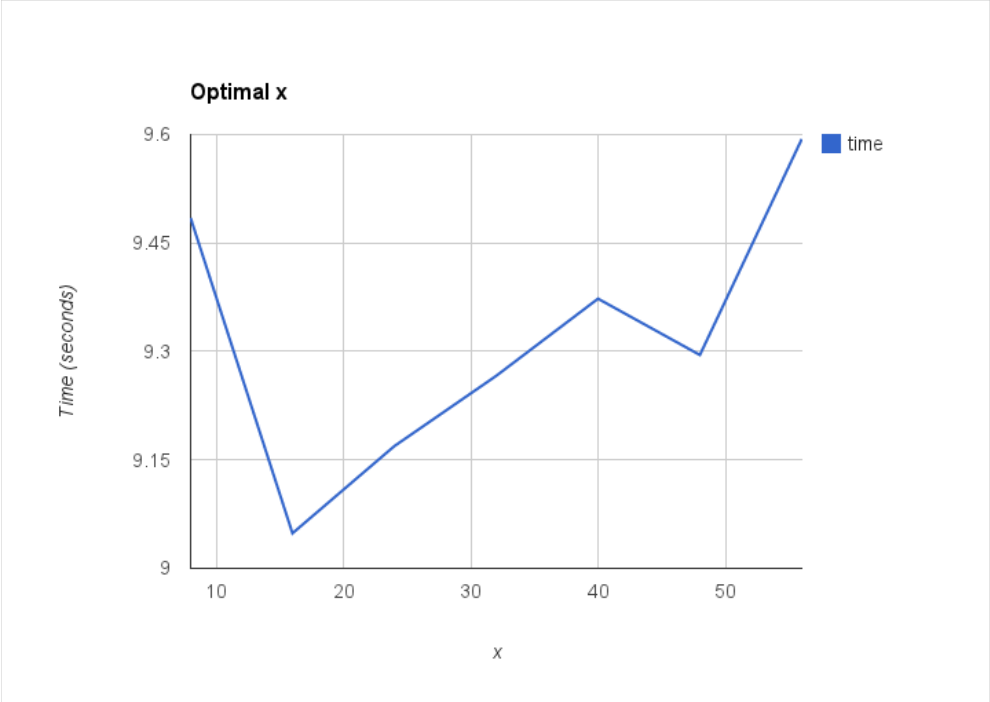
As shown above, the imbalanced workload doesn't seem to be affected much by x, but the balanced workload drops significantly when x first becomes larger, then slowly rises again. For this part, we decided that x was optimal at 32.

Next, we explored what was the best chunk size for x at 32.



We found out that the chunk size at first drops down, but then increases significantly. We think

this is because at first, the chunk size would be too small and have a lot of overhead, but as it gets bigger, it starts to get too big for the number of cores, and so a problem may be partitioned into only 2 or 3 cores, while 5 or 6 lay idle. We determined that the optimal chunk size was 64.



As we can see, it drops down significantly at first, and then after a bit, starts rising back up. We determined that the optimal x for chunk size 64 was 16.

Then, we tested to see what are overheads were. Using x of 16 and chunk size 64, we got 0.923815000 seconds. Using static partitioning, we got 0.907580000 seconds. So, it turns out we didn't have horrifically bad overhead, but it is still quite a bit.