IN3030 - Obligatorisk Innlevering 1

Aleksander Rosseland Kvåle

14. februar 2023

The algorithms were run using an Intel Core i5-1038N67 CPU @ 2GHZ running on MAC OS. The CPU has a total of 4 cores and 8 threads(hyperthreading), and thus I ran the parallel algorithm using 8 threads in the program. We can see from the attached tables and graphs below that A2 is a significantly faster algorithm than A1. This is expected as they are both sequential algorithms, but A2 skips a vital step of actually sorting the entire array. In A2 we simply sort the top K elements at all times, disregarding the rest of the array. This is very easy to spot once we get into the higher values of N.
However, I did struggle to get the parallel algorithm to run any faster than the sequential on anything but the max amount of n. As can be seen in the table we have a speedup of about 3-3.5 on both K values of 20 and 100. This is close to half of the performance boost we expect when running on 8 cores, but we must also expect a bit of loss in time when executing the parallel algorithm, as this has more setup than the sequential one. We must also take into account the fact that the parallel algorithm has more steps than the sequential. After I've had each and every thread sort their individual part of the array, I then have to combine them all into the final array of K numbers. In this case I chose to run a new loop over the last 7x100 elements and compare them to the array of the first thread(not the main thread). As can be seen in the last part of the findklargest method in the class Para.

For lower values of n the parallel algorithm actually performs much worse than the sequential. As stated above this is directly related to the fact that the parallel version has more steps in total. Even if we divide some workload over all threads, there are still sequential parts to this algorithm as well that must be executed in order to get the correct results.

Table 1 k = 20

n	A1 time	A2 time	A2 parallell	Sppeedup
1000	0,22	0,15	0,89	0,17
10000	0,83	0,33	1,19	0,28
100000	8,27	0,59	1,83	0,32
1000000	83,84	1,81	4,59	0,39
10000000	991,41	6,83	11,83	0,58
100000000	11118,88	57,88	16,2	3,57

Table 2 k = 100

n	A1 time	A2 time	A2 paralle	Speedup
1000	0,22	0,46	1,43	0,32
10000	0,94	0,52	1,6	0,33
100000	8,05	0,85	2,8	0,30
1000000	84,41	1,69	4,74	0,36
10000000	983,85	7,18	12,13	0,59
100000000	11038,1	52,93	15,87	3,34