Distinction assignment

Matthijs de Vries and Elsa Carlsson

October 21, 2020

1 Different values of forkAfter & evaluation to sequential solution:

in our tests we have used various values for `forkAfter`: 0, 1, 2, 3, 4, 5, 7, 10, 15, 20 We did not see the reason to test all values, so we started small with each value to see what happens with the use of different values, as we understood the correlation between the result and `forkAfter` we decided to just try a few different larger values of `forkAfter`: 7, 10, 15, 20

from the tests it seemed clear that the sequential solution was faster then most of the parallel searches depending on the `forkAfter` value. "-" marks the sequential times. With a higher `forkAfter` value the parallel search seemed to be faster which would be more apparent on a better computer as these tests were performed on a laptop with 4CPUs x 2 cores. these were the results performed on the small map and a custom 80x40 map called vast. these results are produced from an average over ten runs at each stage.

`forkAfter`	small	vast
0	7.3 ms	267.3 ms
5	7.1 ms	246.6 ms
10	2.3 ms	74.9 ms
20	4.1 ms	44.4 ms

We understand that for some larger `forkAfter` values the result should be faster than the sequential solver however with a lower `forkAfter` value it is going to take longer due to the overhead when joining together all the subtasks making it slower than the sequential solve.

2 Custom maps

We made a custom map that is a lot larger than the medium map that we got from the lab material. This map is 40x40 in size with the goal set in the bottom left corner of the map. We also made a 80x40 map by just adding a 40x40 grid of cells the player could move to above the large map.

Varying the amount of workers in the pool:

We tried some different set of workers in the pool. this was done on the vast map and with {1, 2, 3, 4, 5, 7} workers computed from an average of 10 runs. The `forkAfter` value was set to 10.

workers	result
1	188.3 ms
2	190.5 ms
3	149.4 ms
4	144.9 ms
5	130.7 ms
7	120,2 ms
8	114.5 ms
9	85.0 ms
10	87.0 ms
15	80.7 ms
20	68.2 ms

Once again computed on a laptop with 4 CPUs and 2 cores and in this setting it seems not to differ that much. However one can still see the effect that with the more amount of workers the computation time goes down slightly.