Home exam 1 - IN3200

Candidate nr:
(Dated: March 9, 2020)

This is an abstract

I. INTRODUCTION

II. METHODS

Shellsort

In order to sort the data in the functions read_graph_from_file2 and top_n_webpages, shellsort was applied. It's therefore instructive to list the actual algorithm.

Algorithm 1 Shellsort (ascending order)

```
for gap = N/2; gap > 0; gap/ = 2 do
    for i = gap; i < N; i = i + 1 do
        tmp = arr[i];
        for j = i; j >= gap; j = j − gap do
            if arr[j − gap] < tmp then
                break;
            arr[j] = arr[j − gap];
        arr[j] = tmp;
```

III. RESULTS

Timing of serial codes

The measured time of the serial implementations of the various functions are shown in table I.

Function name	Time in seconds
read_graph_from_file1	0.174369
count_mutual_links1	1133.817
read_graph_from_file2	0.990147
count_mutual_links2	0.008971
top_n_webpages	0.068952

TABLE I. The table shows the measured time using clock() from the Ctime-library. read_graph_from_file1 and count_mutual_links1 was applied to a web-graph containing $N = 10000$ nodes and $N_{\text{links}} = 37841$ edges as found in the file test_webpages.txt. The data in this file was extracted from web-NotreDame.txt. read_graph_from_file2 and count_mutual_links2 was applied directly to the web-graph contained in web-NotreDame.txt. This file contained $N = 325729$ nodes and $N_{\text{links}} = 1479143$ edges.

Parallelized version of count_mutual_links1

Using OpenMP to parallelize count_mutual_links1, the results in table II were obtained.

Number of threads	Time in seconds
1	1038.65
2	572.79
4	374.42
8	312.68

TABLE II. The table presents the time used by the function count_mutual_links1 as a function of threads. The web-graph used contained $N = 10000$ nodes and $N_{\text{links}} = 37841$ edges as found in the file test_webpages.txt. The data in this file was extracted from web-NotreDame.txt. All times were measured using omp_get_wtime().

Parallelized version of count_mutual_links2

Using OpenMP to parallelize count_mutual_links2 and measuring the time used by the function for different number of threads yielded the results shown in table III.

Number of threads	Time in seconds
1	0.007014
2	0.004470
4	0.003221
8	0.004967

TABLE III. The table shows the time used by count_mutual_links2 when parallelized with OpenMP as a function of number of threads. The function was applied to the web-graph contained in web-NotreDame.txt containing $N = 325729$ nodes and $N_{\text{links}} = 1479143$ edges.. The timing was done using the OpenMP library function omp_get_wtime().

IV. DISCUSSION

V. CONCLUSION