Parallel and distributed Systems
4th assignment

Erika Koro

October 13, 2022

Objective

The objective of this assignment is to calculate the K-Nearest neighbours for all points of a dataset. Specifically, in the first serial source code is constructed the vantage point tree which sequentially is used in the KNN algortihm. In the KNNSearch the construction of the VP Tree per point is used in order to sort the array with the points, according to their median distance from the pivot and chooses the first-k elements as neighbours. In order to parallelize this algorithm, MPI specification for C is used to optimize performance and give a distributed solution. The code for this assignment can be found in here.

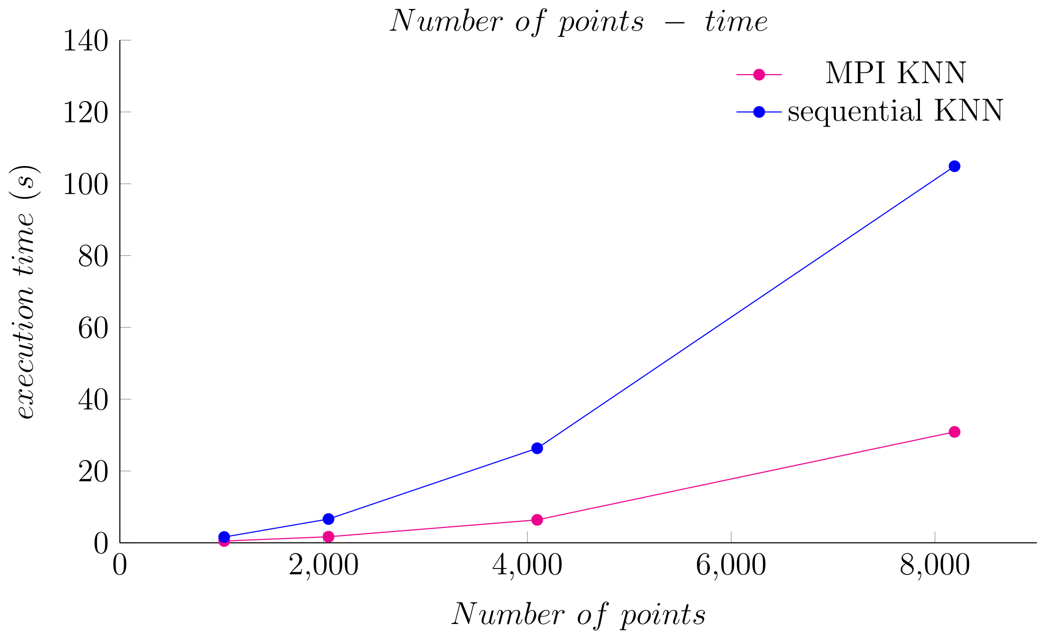
Experiments

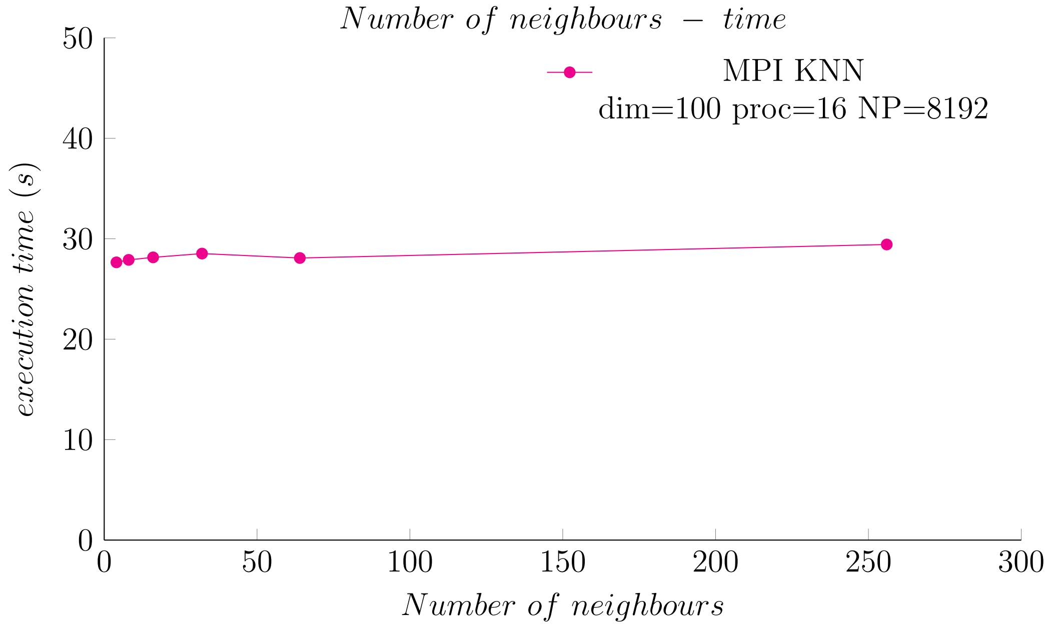
The experiments were executed in an AMD Ryzen 7 5800H, 3200.000 MHz. and the results are based on the number of points and dimension of the dataset and the number of searching neighbours is a power-of-two value with k = $2^{[1:8]}$.

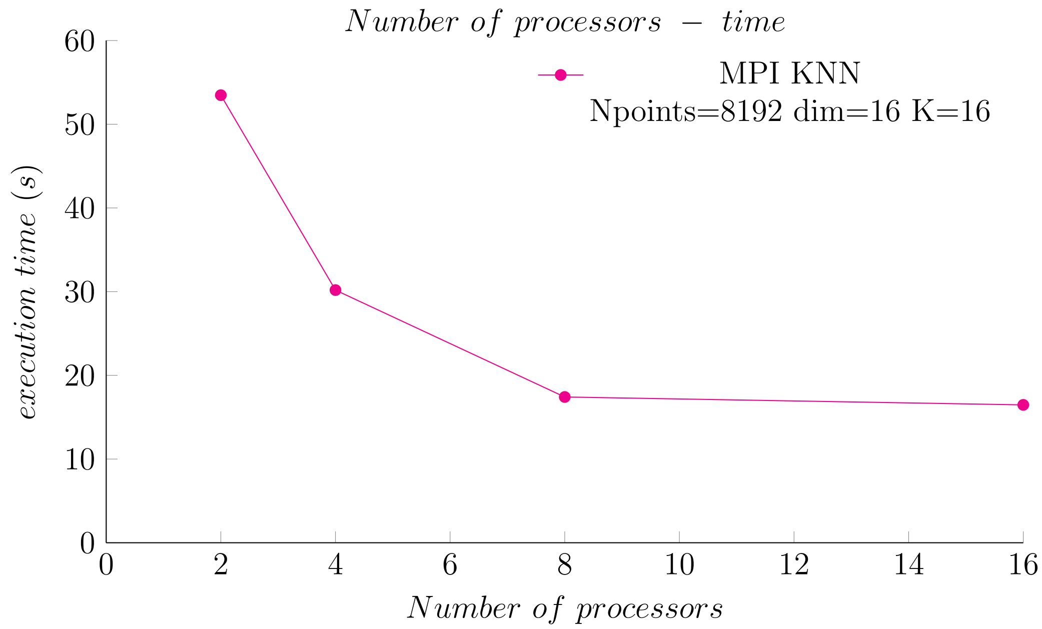
Observations

First of all, by keeping stable the dimension and the number of processes (optimal=16) and increasing the number of points causes averagely 5 times slower KNN algortihm than MPI. Secondly, keeping the number of points, the dimension and the processes stable and increasing the number of k-nearest neighbours keeps MPI's execution time constant.Meanwhile, increasing the number of k increases KNN sequential's execution time. Also, by increasing the number of dimension caused 2-times slower performance in MPI. At last, as it is expected by raising the number of processes from 2 to 16 is observed the optimized performance, a number that can be predicted by the number of available cores.

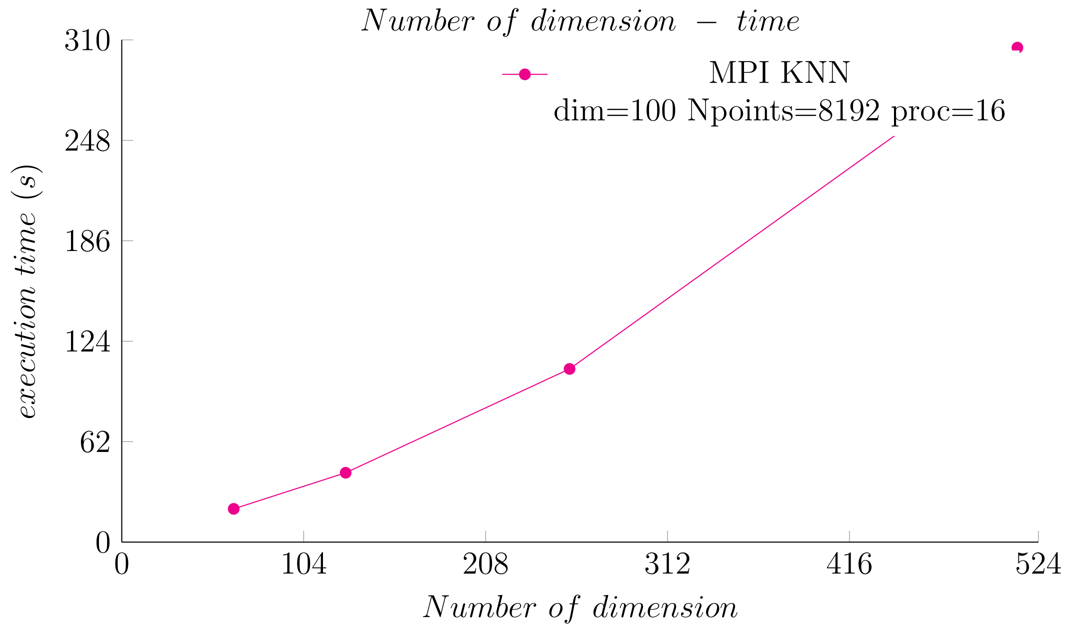
Plots


Number of points − time
MPI KNN
sequential KNN
execution time (s)
0
20
40
60
80
100
120
140
0
2,000
4,000
6,000
8,000
Number of points


Number of neighbours − time
MPI KNN
dim=100 proc=16 NP=8192
execution time (s)
0
10
20
30
40
50
0
50
100
150
200
250
300
Number of neighbours



Number of processors − time
MPI KNN
Npoints=8192 dim=16 K=16
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
execution time (s)
Number of processors



Number of dimension − time
MPI KNN
dim=100 Npoints=8192 proc=16
310
248
186
124
62
0
0
104
208
312
416
524
execution time (s)
Number of dimension

Conclusion

In conclusion, needs to be marked the contribution of Message Passing Interface in solving problems efficiently regarding the memory and the execution time. Using all the available cores, optimizes the performance for any number of points and changing the number of neighbours(k) does not affect the performance.