

Proj3 prob2 report

Antoine DESRUET 50221600

Environment

Os	Pop!_OS 22.04 LTS x86
CPU	Intel i7-8665U (8) @ 1.900GHz
Memory	16Gb
GCC version	14.0.0
GNU Make version	4.3

Build

In the prob2 folder:

GNU Make

```
make
```

CLI

```
gcc -o a.out ./prob2.c -fopenmp
```

Static

Tables

Execution time

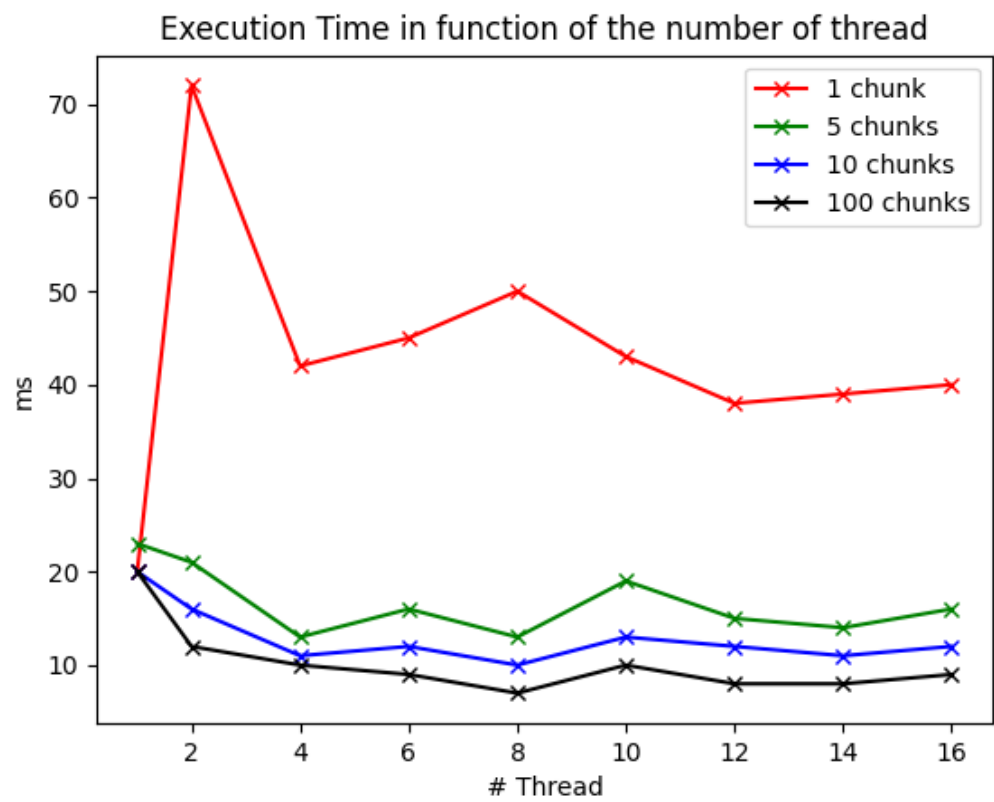
Number of thread	Execution time 1 chunks	Execution time 5 chunks	Execution time 10 chunks	Execution time 100 chunks
1	20	23	20	20
2	72	21	16	12
4	42	13	11	10
6	45	16	12	9
8	50	13	10	7
10	43	19	13	10
12	38	15	12	8
14	39	14	11	8
16	40	16	12	9

Performance

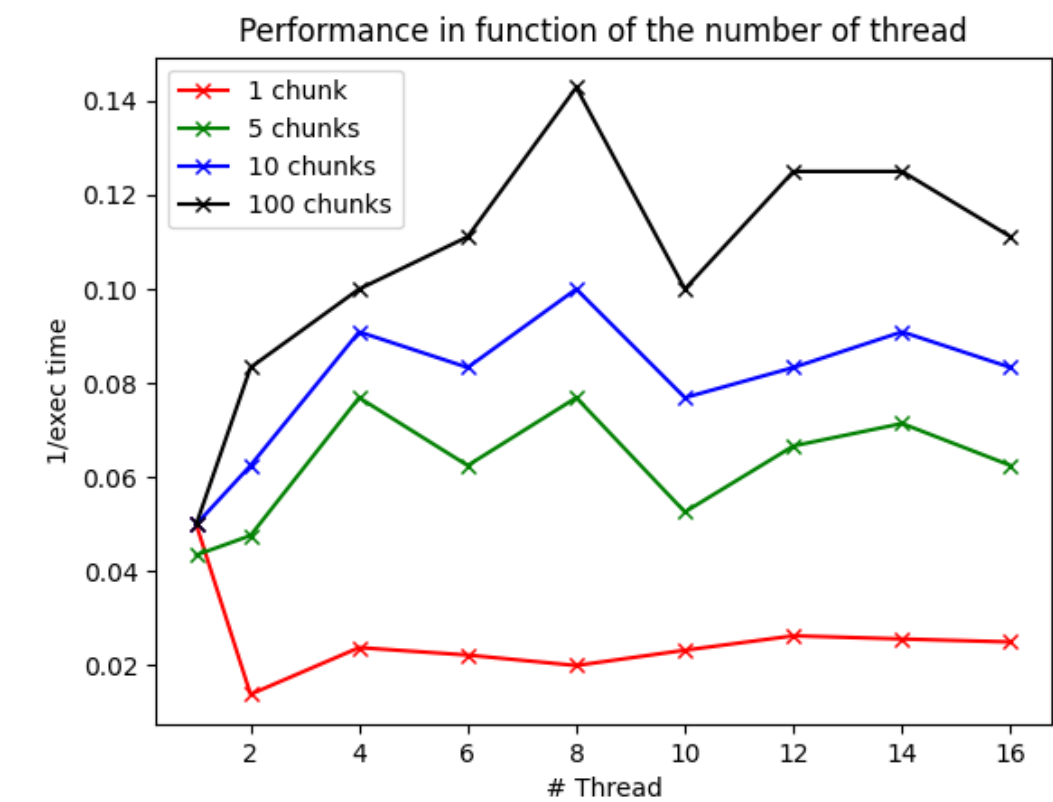
Number of thread	Performance 1 chunks	Performance 5 chunks	Performance 10 chunks	Performance 100 chunks
1	0.05	0.043478260869565216	0.05	0.05
2	0.013888888888888888	0.047619047619047616	0.0625	0.08333333333333333
4	0.023809523809523808	0.07692307692307693	0.09090909090909091	0.1
6	0.022222222222222223	0.0625	0.08333333333333333	0.11111111111111111
8	0.02	0.07692307692307693	0.1	0.14285714285714285
10	0.023255813953488372	0.05263157894736842	0.07692307692307693	0.1
12	0.02631578947368421	0.06666666666666667	0.08333333333333333	0.125
14	0.02564102564102564	0.07142857142857142	0.09090909090909091	0.125
16	0.025	0.0625	0.08333333333333333	0.11111111111111111

Graphs

Execution time



Performance



Dynamic

Tables

Execution time

Number of thread	Execution time 1 chunks	Execution time 5 chunks	Execution time 10 chunks	Execution time 100 chunks
1	181	49	35	24
2	191	41	25	14
4	183	43	22	8
6	157	39	19	8
8	180	30	19	7
10	159	37	18	7
12	137	31	18	7
14	181	37	18	9
16	142	37	16	8

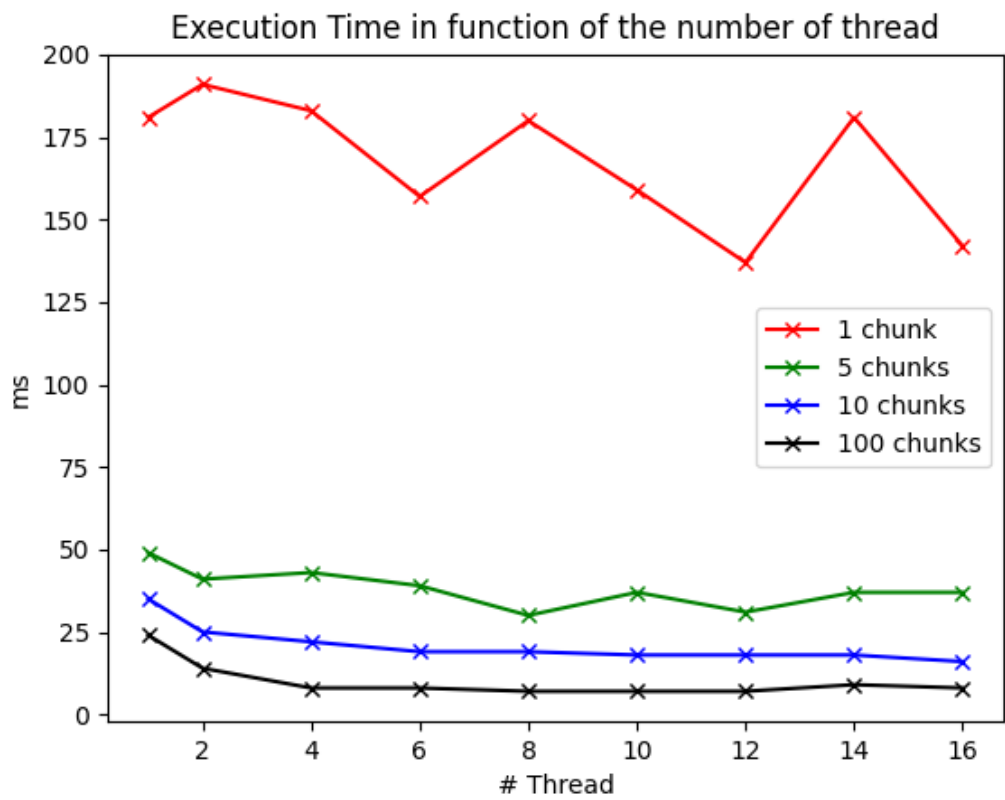
Performance

Number of thread	Performance 1 chunks	Performance 5 chunks	Performance 10 chunks	Performance 100 chunks
------------------	----------------------	----------------------	-----------------------	------------------------

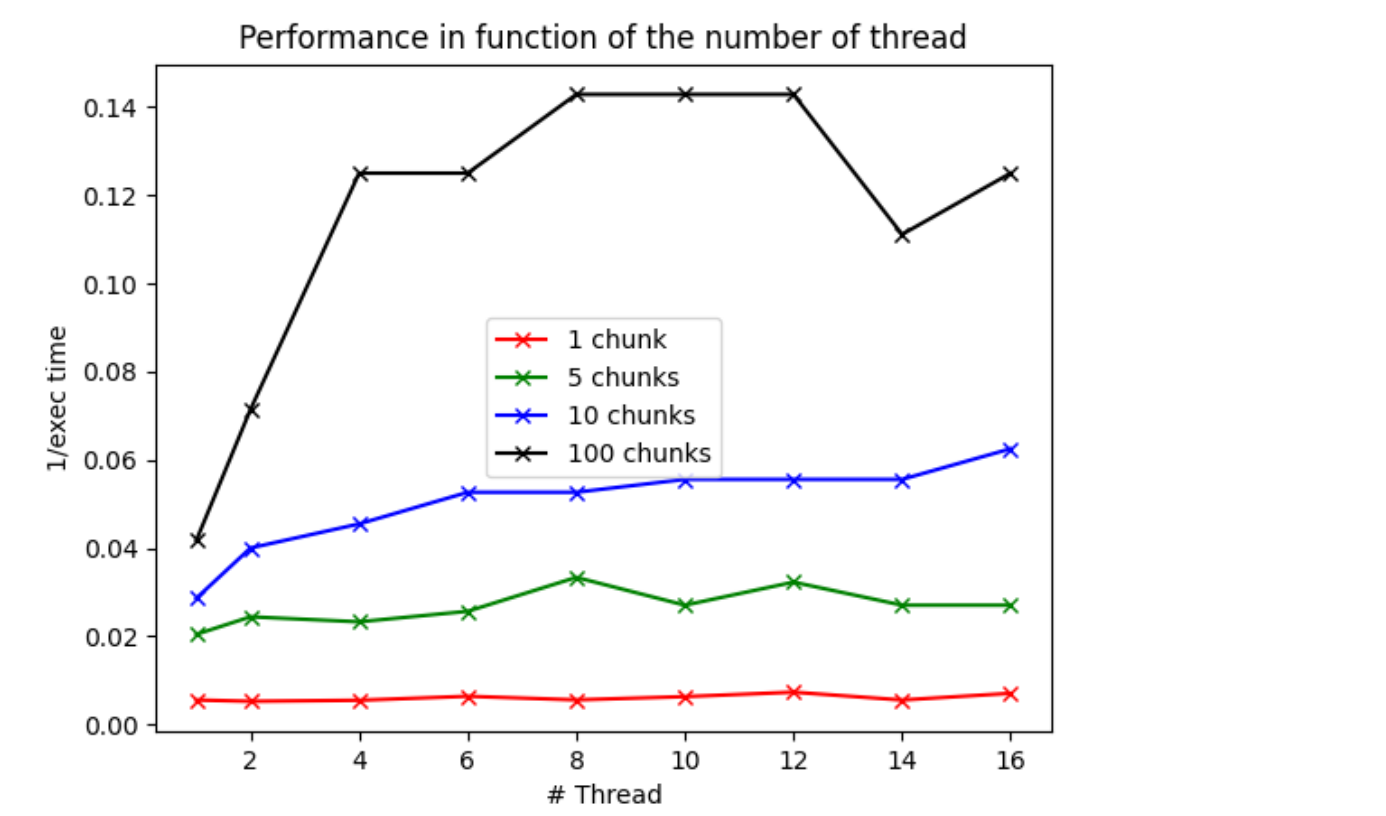
Number of thread	Performance 1 chunks	Performance 5 chunks	Performance 10 chunks	Performance 100 chunks
1	0.0055248618784530384	0.02040816326530612	0.02857142857142857	0.041666666666666664
2	0.005235602094240838	0.024390243902439025	0.04	0.07142857142857142
4	0.00546448087431694	0.023255813953488372	0.045454545454545456	0.125
6	0.006369426751592357	0.02564102564102564	0.05263157894736842	0.125
8	0.005555555555555556	0.033333333333333333	0.05263157894736842	0.14285714285714285
10	0.006289308176100629	0.02702702702702703	0.055555555555555555	0.14285714285714285
12	0.0072992700729927005	0.03225806451612903	0.055555555555555555	0.14285714285714285
14	0.0055248618784530384	0.02702702702702703	0.055555555555555555	0.11111111111111111
16	0.007042253521126761	0.02702702702702703	0.0625	0.125

Graphs

Execution time



Performance



Guided

Tables

Execution time

Number of thread	Execution time 1 chunks	Execution time 5 chunks	Execution time 10 chunks	Execution time 100 chunks
1	28	25	23	23
2	13	11	11	11
4	7	7	8	8
6	7	8	7	7
8	13	7	7	7
10	8	7	7	8
12	6	8	8	8
14	8	8	8	8
16	8	8	8	8

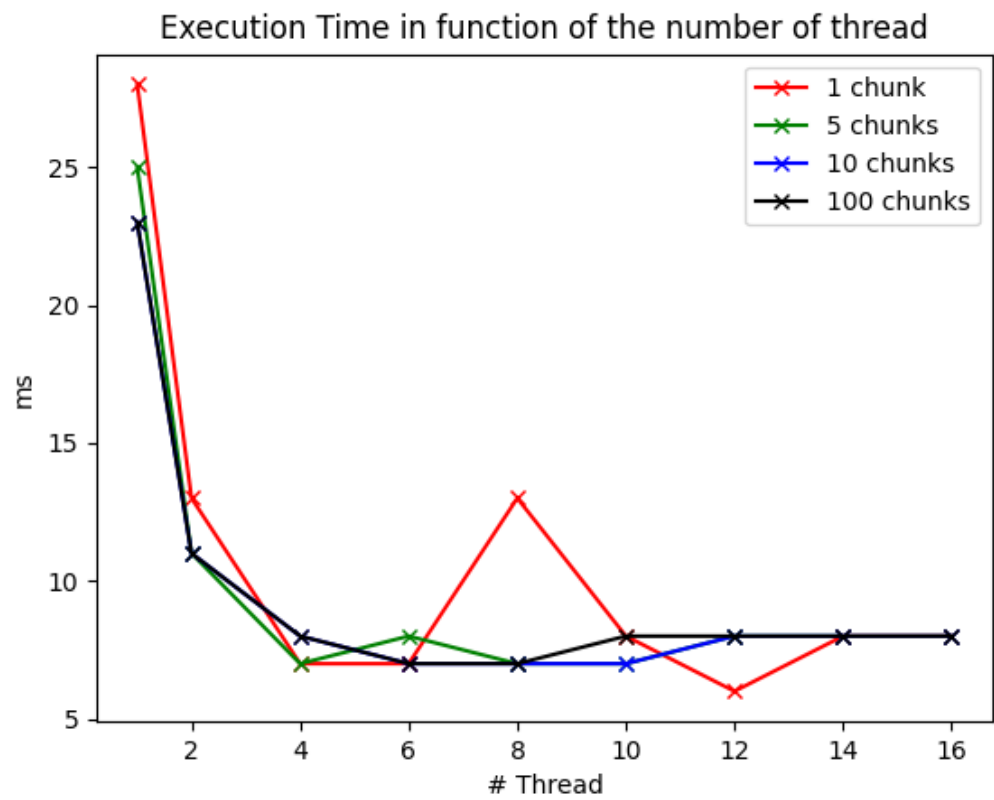
Performance

Number of thread	Performance 1 chunks	Performance 5 chunks	Performance 10 chunks	Performance 100 chunks
1	0.006	0.020	0.028	0.042
2	0.006	0.024	0.040	0.072
4	0.006	0.023	0.045	0.125
6	0.007	0.025	0.052	0.125
8	0.006	0.033	0.052	0.142
10	0.007	0.027	0.055	0.142
12	0.008	0.032	0.055	0.142
14	0.006	0.027	0.055	0.110
16	0.008	0.027	0.062	0.125

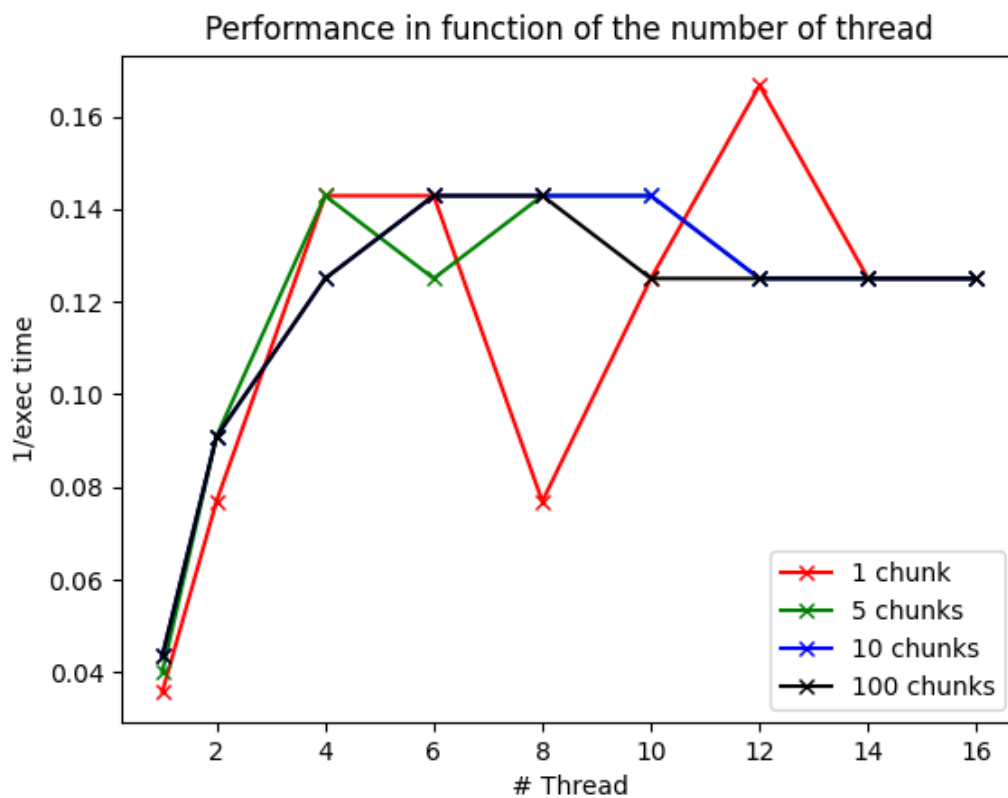
Number of thread	Performance 1 chunks	Performance 5 chunks	Performance 10 chunks	Performance 100 chunks
1	0.03571428571428571	0.04	0.043478260869565216	0.043478260869565216
2	0.07692307692307693	0.09090909090909091	0.09090909090909091	0.09090909090909091
4	0.14285714285714285	0.14285714285714285	0.125	0.125
6	0.14285714285714285	0.125	0.14285714285714285	0.14285714285714285
8	0.07692307692307693	0.14285714285714285	0.14285714285714285	0.14285714285714285
10	0.125	0.14285714285714285	0.14285714285714285	0.125
12	0.16666666666666666	0.125	0.125	0.125
14	0.125	0.125	0.125	0.125
16	0.125	0.125	0.125	0.125

Graphs

Execution time



Performance



Explanation

We can see that the more important the chunk size is, the more high the performance is.

This work for every scheduling types.

This is because each task take the same time so threads can do a lot of tasks in one time. We don't need to split thoses tasks.