M2-BIG DATA
High Performance Computing
Profiling

Lab 3 - OpenMP

Réalisé par :
Gaby Maroun

Encadré par :
Dr. Etancelin JM

11 mars 2021

Exercice 1. Matrix product

Part 1.

1. **Validate your program with A = Id (you should verify that C = B) with several values for n and process numbers.**
   For this validation, I created a check function that parse through C and B and return OK if they're equal.

   for n =2,

```
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ ./lab3mpi 2 p
A=[
1.0 0.0
0.0 1.0
B=[
0.0 2.0
1.0 3.0
C=[
0.0 2.0
1.0 3.0
```

   for n =3,

```
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ ./lab3mpi 3 p
A=[
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
B=[
0.0 3.0 6.0
1.0 4.0 7.0
2.0 5.0 8.0
C=[
0.0 3.0 6.0
1.0 4.0 7.0
2.0 5.0 8.0
```

   for n =5,

```
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ mpicc -o lab3mpi lab3mpi.c
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ ./lab3mpi 5
A=[
1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 1.0
B=[
0.0 5.0 10.0 15.0 20.0
1.0 6.0 11.0 16.0 21.0
2.0 7.0 12.0 17.0 22.0
3.0 8.0 13.0 18.0 23.0
4.0 9.0 14.0 19.0 24.0
C=[
0.0 5.0 10.0 15.0 20.0
1.0 6.0 11.0 16.0 21.0
2.0 7.0 12.0 17.0 22.0
3.0 8.0 13.0 18.0 23.0
4.0 9.0 14.0 19.0 24.0
```

The check function,

```
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ ./lab3mpi 3 c
OK
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ ./lab3mpi 5 c
OK
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ ./lab3mpi 2 c
OK
```

The problem I encountered is the rank gathering in the C by the end, the final rank would not work as planned. I worked for hours and days trying to fix the problem but I couldn't find that small breakthrough.

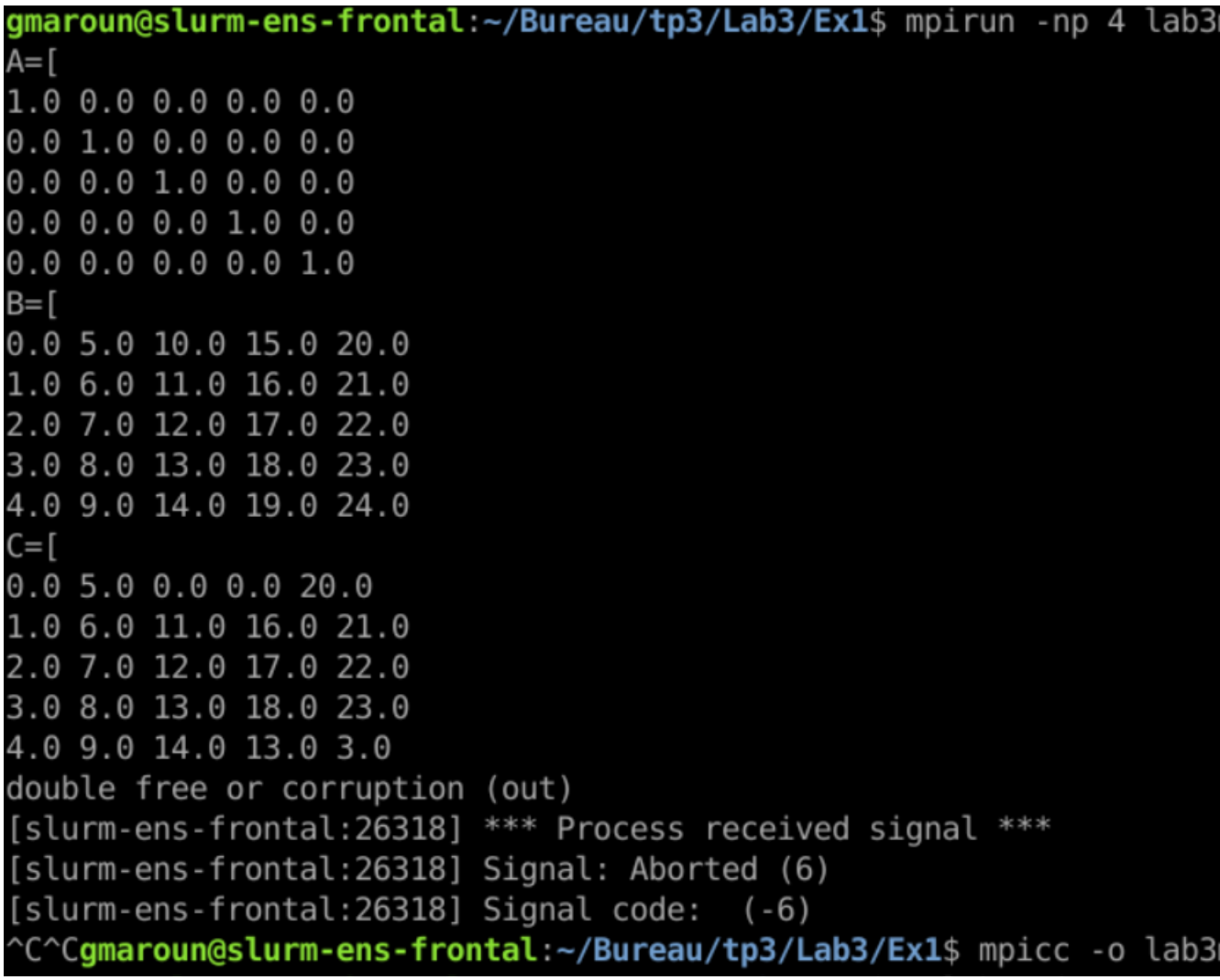
Here is what the working program return,

```
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$ mpirun -np 4 lab3mpi 5 p
A=[
1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 1.0
B=[
0.0 5.0 10.0 15.0 20.0
1.0 6.0 11.0 16.0 21.0
2.0 7.0 12.0 17.0 22.0
3.0 8.0 13.0 18.0 23.0
4.0 9.0 14.0 19.0 24.0
C=[
0.0 0.0 10.0 15.0 0.0
1.0 6.0 0.0 16.0 0.0
2.0 7.0 12.0 0.0 0.0
3.0 8.0 13.0 18.0 0.0
0.0 0.0 0.0 0.0 0.0
gmaroun@slurm-ens-frontal:~/Bureau/tp3/Lab3/Ex1$
```

While if I try to increase the nloc by 1,

```
int nloc=n/(float)size+1;
```

I get this, which is so close to what we want but at some point, it out passes the size, which makes it stop



2. **Perform a strong scalability analysis for n = 1120 up to 112 process (the entire cluster).**

   The scalability analysis came as follow,

   launch file,

```
#SBATCH -N 4
#SBATCH -n 112
#SBATCH -c 1

cd /home/3/gmaroun/Bureau/tp3/Lab3/Ex1

mpirun -np 1 lab3mpi 1120
mpirun -np 10 lab3mpi 1120
mpirun -np 20 lab3mpi 1120
mpirun -np 28 lab3mpi 1120
mpirun -np 40 lab3mpi 1120
mpirun -np 56 lab3mpi 1120
mpirun -np 80 lab3mpi 1120
mpirun -np 112 lab3mpi 1120
```

And the timing by process demanded written in the scalability.dat file,

process time

```
2 64.747
4 27.673
8 11.6807
16 5.51606
28 3.23508
56 4.77716
112 4.41333
```

It is so obvious, how fast it gets when we increase the number of process. although, at some point after 16, it became unnecessary as the time didn't change much, sometimes even went slower by just 1 second.

3. **Compare with the performances of the OpenMP program.**
   To compare to the previous lab's openMP program, I performed the following threads, with 12544 threads using all the threads of all the nodes of the entire clusters

```
#!/usr/bin/env bash
#SBATCH -N 4
#SBATCH -n 112
#SBATCH -c 1

cd /home/3/gmaroun/Bureau/tp3/Lab3/Ex1/lab2

#OMP_NUM_THREADS=10 ./parallel 1000

export OMP_PROC_BIND=true

#Strong scalabillity
OMP_NUM_THREADS=1

OMP_NUM_THREADS=2 ./parallel 1120
OMP_NUM_THREADS=4 ./parallel 1120
OMP_NUM_THREADS=8 ./parallel 1120
OMP_NUM_THREADS=16 ./parallel 1120
OMP_NUM_THREADS=28 ./parallel 1120
OMP_NUM_THREADS=112 ./parallel 1120
OMP_NUM_THREADS=12544 ./parallel 1120
```

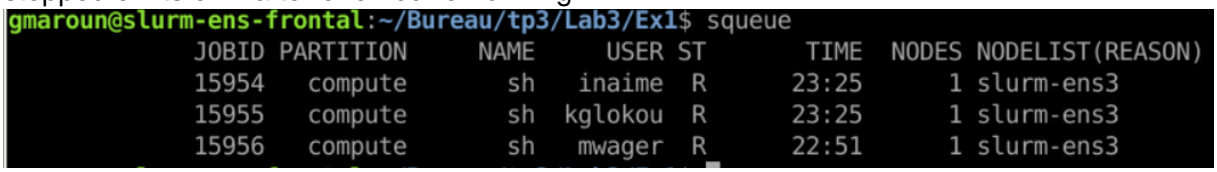
And the results for different number of threads came as follows,

time threads

```
42.44 2
21.28 4
11.23 8
5.65 16
3.09 28
3.64 112
9.12 12544
```

It seems that the two version are not far from each other, even though the openMP version has 1 second difference is its favor.

I would like to not that having to wait ours for me to try and execute on 4 nodes, was a big obstacle that didn't let me focus more on finding solutions. As you may see in the following image, many users used the bash for ours but I wasn't able to stop their works, which stopped on its own after one hour of running.



Part 2

For part 2, I tried everything I could to make it work with allgather but I needed more instructions and clarifications, which might've been possible if not for the online courses.

In the end, I would like to apologize for the delay. I would also like to thank you for your hard work and patience during this semester and I wish you all the luck in the future.

La fin.