
OS202 - Programming Parallel Computers

Travail Dirigée

12 septembre 2024

Guilherme Nunes Trofino
2022-2024

Table des matières

1	Introduction	2
1.1	Information Matier	2
1.2	Caracteristiques Ordinateur	2
2	Cours Amphi	3
2.1	Interblocage	3
2.2	Alice Problem's	3
3	Mandelbrot	5
4	Produit Matrice-Vector	6
4.1	Vecteur Colonne	6
4.2	Vecteur Ligne	6

1. Introduction

Repository Hello! My name is Guilherme Nunes Trofino and this is my LaTeX notebook of OS202 - Programming Parallel Computers that can be found in my GitHub repository : https://github.com/tr0fin0/classes_ensta.

Disclaimer This notebook is made so it may help others in this subject and is not intend to be used to cheat on tests so use it by your on risk.

Suggestions If you may find something on this document that does not seam correct please reach me by e-mail : guitrofino@gmail.com.

1.1. Information Matier

Référence Dans cette matière le but sera de comprendre . Ce travail est sur <https://github.com/> avec l'objectif d'étudier et démontrer l'augmentation de performance quand on utilise la programmation parallèle.

1.2. Caracteristiques Ordinateur

CPU On utilisé le commande `lscpu` pour avoir des informations sur le processeur de mon ordinateur en retournant le suivant :

```

1  Architecture:          x86_64
2      CPU op-mode(s):    32-bit, 64-bit
3      Address sizes:      39 bits physical, 48 bits virtual
4      Byte Order:         Little Endian
5      CPU(s):             20
6      On-line CPU(s) list: 0-19
7      Vendor ID:          GenuineIntel
8      Model name:         12th Gen Intel(R) Core(TM) i7-12700H
9      CPU family:         6
10     Model:              154
11     Thread(s) per core: 2
12     Core(s) per socket: 14
13     Socket(s):          1
14     Stepping:           3
15     CPU max MHz:        4700.0000
16     CPU min MHz:        400.0000

```

On peut voir qui mon ordinateur a, théoriquement, 20 CPU's disponibles avec les mémoires suivants :

```

1  Caches (sum of all):
2      L1d:    544 KiB    (14 instances)
3      L1i:    704 KiB    (14 instances)
4      L2:     11.5 MiB    ( 8 instances)
5      L3:     24 MiB     ( 1 instance)

```

Ces données seront utilisés pour l'analyse des performances.

2. Cours Amphi

Résolution.

2.1. Interblocage

Résolution. On considère le l'exercice sur l'interblocage donné dans le cours et décrivez deux scénarios...

1. Blocking :

```

1  if (rank==0)
2  {
3      MPI_Recv(rcvbuf, count, MPI_DOUBLE, 1, 101, commGlob, &status);
4      MPI_Send(sndbuf, count, MPI_DOUBLE, 1, 102, commGlob);
5  }
6  else if (rank==1)
7  {
8      MPI_Recv(rcvbuf, count, MPI_DOUBLE, 0, 102, commGlob, &status);
9      MPI_Send(sndbuf, count, MPI_DOUBLE, 0, 101, commGlob);
10 }

```

2. Non-Blocking :

```

1  MPI_Request req;
2  if (rank == 0)
3  {
4      double vecteur[5] = { 1., 3., 5., 7., 22. };
5      MPI_Isend(vecteurs, 5, MPI_DOUBLE, 1, 101, commGlob, &req);
6      // Some compute with other data can be executed here!
7      MPI_Wait(req, MPI_STATUS_IGNORE);
8  }
9  else if (rank==1)
10 {
11     MPI_Status status; double vecteurs[5];
12     MPI_Irecv(vecteurs, 5, MPI_DOUBLE, 0, 101, commGlob, &req);
13     int flag = 0;
14
15     do {
16         // Do computation while message is not received on another
17         data MPI_Test(&req, &flag, &status);
18     } while(flag);
19 }

```

La probabilité d'une interblocage est grand car une erreur dans le rank d'une variable pour causer une loop.

2.2. Alice Problem's

Résolution. On considère les equations suivants :

$$S(n) = \lim_{n \rightarrow \infty} \frac{n}{1 + (n-1)f} \rightarrow \frac{1}{f} \quad (2.1)$$

Où :

1. n : number of computing units ;
2. f : fraction of t_s which can't be parallelized ;
3. t_s : time to run code sequential ;

Comme le code 90% du code peut-être parallèle $f = 1 - 0.9 = 10\%$ et comme $n \gg 1$ on a que $S(n)$ sera au maximum $S(n) = 10$.

3. Mandelbrot

Résolution. On considère que Mandelbrot est un ensemble fractal :

$$z_{n+1} = \begin{cases} z_0 & = 0 \\ z_{n+1} & = z_n^2 + c \end{cases} \quad \text{où } c : \text{valeurs complexe donnée} \quad (3.1)$$

On peut montrer que si il existe N tel que $|z_N| > 2$, alors la suite diverge

$$c = (x_{\min} + p_i \frac{x_{\max} - x_{\min}}{W}) + i(y_{\min} + p_j \frac{y_{\max} - y_{\min}}{H}) \quad (3.2)$$

Image de W par H pixels telle qu'à chaque pixel (p_i, p_j)

Il semble qu'on étudie ses fractals car un français les a découverts.

```

1  MPI_Request req;
2  if (rank == 0)
3  {
4      double vecteur[5] = { 1., 3., 5., 7., 22. };
5      MPI_Isend(vecteurs, 5, MPI_DOUBLE, 1, 101, commGlob, &req);
6      // Some compute with other data can be executed here!
7      MPI_Wait(req, MPI_STATUS_IGNORE);
8  }
9  else if (rank==1)
10 {
11     MPI_Status status; double vecteurs[5];
12     MPI_Irecv(vecteurs, 5, MPI_DOUBLE, 0, 101, commGlob, &req);
13     int flag = 0;
14
15     do {
16         // Do computation while message is not received on another data
17         MPI_Test(&req, &flag, &status);
18     } while(flag);
19 }
```

`mpirun -np 4 ./helloWorldMPI.exe`

4. Produit Matrice-Vector

4.1. Vecteur Colonne

Résolution.

4.2. Vecteur Ligne

Résolution.
