TP n° 1

Programmation MPI

L'adresse email pour l'envoi des comptes rendus à utiliser est `pablo.de-oliveira-castro@cea.fr`. Le compte rendu de TP est à fournir au plus tard le 22 octobre. Dans le compte rendu, inclure les codes ainsi que les traces d'exécution.

Dans ce TP vous mettrez en pratique les bases de programmation MPI. Le sujet est divisé en trois parties. Dans la partie I vous vous échaufferez sur quelques exercices préliminaires. Dans la partie II vous vous interesserez à la multiplication matricielle distribuée. Finalement dans la partie III vous chercherez à estimer la valeur de π de manière parallèle.

I) Préliminaires

Exercice 1 [**Hello World MPI**]

Écrivez un programme MPI qui affiche pour chaque processus le message : "`Hello from processus : <rang du processus courant>`". Le processus de rang 0 devra afficher également le nombre total de processus crées. Observer que l'ordre des messages affichés varie d'une exécution à l'autre.

Pour compiler un programme MPI, il faut rajouter à votre PATH le chemin vers le compilateur MPI `mpicc`. Puis pour générer le binaire :

```
mpicc -o hello_world hello_world.c
```

Pour lancer un programme MPI, il faut exécuter la commande suivante :

```
mpirun -n nb_processus -machinefile fichier_des_machines_a_utiliser hello_world
```

Le fichier `fichier_des_machines_a_utiliser` comportant un nom ou l'adresse IP d'une machine par ligne.

Exercice 2 [**Communication en anneau**]
On va écrire un programme où les processus se passent un jeton de manière circulaire. Pour l'exemple, le jeton sera un entier que chaque processus incrémentera avant de le passer à son voisin.

Le jeton sera crée sur le processus de rang 0 qui l'initialisera à 1. Le processus de rang n passera le jeton au processus de rang $n+1$, sauf si celui ci n'existe pas, auquel cas le jeton retournera au processeur de rang 0 qui l'affichera et mettra terme au programme (cf. figure 1).

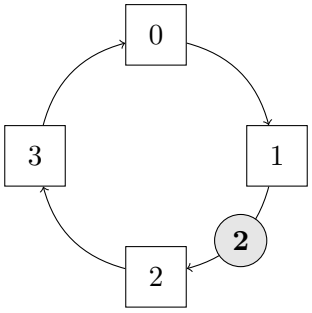

FIG. 1 – Exemple de communication en anneau.

II) Multiplication de matrices

Vous allez maintenant écrire un algorithme de multiplications de matrices carrés. En entrée l'algorithme prendra deux matrices A et B de taille N et de type `double` et calculera la matrice $C = A * B$.

Vous allez paralléliser l'algorithme en exploitant la décomposition selon les lignes de la matrice A :

$$C = A * B = \begin{vmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{vmatrix} * B = \begin{vmatrix} L_1 * B \\ L_2 * B \\ \vdots \\ L_n * B \end{vmatrix}$$

Dans votre implémentation le processus de rang 0 n'effectuera pas de calcul, il sera chargé de distribuer les données aux autres processus que l'on appelera *travailleurs*.

Exercice 3 [**Broadcast de A et de B**]

Commencez par une première version où les matrices A et B sont envoyées en broadcast à tous les *travailleurs*. Chaque *travailleur* fera la multiplication sur la partie des lignes de A lui correspondant.

Les résultats seront rassemblés sur le processus de rang 0 qui affichera le résultat de la multiplication.

Exercice 4 [**Distribution des lignes de A parmi les processus**]

Modifiez le programme précédent de façon à ce que seules les lignes utilisés par un *travailleur* lui soient envoyées. Ceci présente l'avantage de réduire les communications et donc d'éviter les congestions du réseau.

III) Calcul de π

Le nombre π peut être défini comme l'intégrale de 0 à 1 de $f(x) = \frac{4}{1+x^2}$. Une manière simple d'approximer une intégrale est de discrétiser l'ensemble d'étude de la fonction (c.à.d. découper

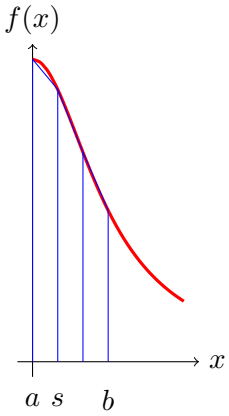

FIG. 2 – Approximation de π ($a = 0$ et $b = 1$)

l'intervalle $[a; b]$ en n intervalles, dont on approxime l'intégrale comme la surface du trapèze rectangle formé par les extrêmes) et considérer l'approximation suivante (cf. figure 2) :

$$\text{Supposons } s = \frac{b-a}{n}, \text{ alors } \int_a^b f(x)\,dx \approx \sum_{i=1}^{n} s \times \frac{f(a+i\times s) + f(a+(i+1)\times s)}{2}$$

On peut alors répartir le calcul de π en répartissant l'intervalle entre les processus.

Exercice 5 Proposez un programme MPI qui parallélise une approximation de la valeur de π.