

Programmation parallèle

UFR SAT, CFPP - MaDSI 1

Travaux pratiques 1

Objectifs:

1. Découvrir les capacités multi-core
 2. Comprendre les MPI
 3. Exploiter les différents blocs de processeurs de votre PC pour accélérer l'exécution de code
 4. Tester différents programmes de calcul intensifs et de pouvoir en observer le temps
-

Dans ce TP, vous allez utiliser votre PC en tant que super-calculateur à un seul noeud en ordonnant aux différents cœurs de votre processeur d'effectuer vos opérations de calcul.

Vous allez ensuite convertir la logique de ce code en version MPI, l'exécuter sur un cœur du processeur, puis mettre progressivement en ligne les cœurs restants. Vous constaterez une amélioration progressive de la vitesse de traitement à mesure que vous activez des cœurs successivement. Commençons par la simple équation π comme suit:

$$\pi = \int_0^1 \frac{4}{(1+x)} dx$$

Etape 1 - Installer Linux sur votre PC

Vous pouvez visiter le site Web Ubuntu: <http://www.ubuntu.com/download/desktop/install-ubuntu-desktop> pour obtenir un tutoriel/des instructions et télécharger une version gratuite du système d'exploitation. je

Etape 2 - Accéder aux détails techniques de votre PC

Accéder aux détails techniques de votre processeur via le Panneau de configuration de votre ordinateur.

Sur Windows 7

- Cliquer sur Systèmes et sécurité,
- cliquer sur Système,

- ensuite cliquer sur Gestionnaire de périphériques,
- cliquer sur Processeur.

Sur Windows 10 :

- Cliquez sur l'icône Windows en bas à gauche de la barre des tâches,
- cliquez sur l'icône Explorateur de fichiers, puis sur l'icône Ce PC
- cliquez sur l'onglet Ordinateur en haut à gauche de la fenêtre affichée,
- cliquez sur Propriétés du système. Notez les informations affichées dans la fenêtre.
- Ensuite, cliquez sur Gestionnaire de périphériques, puis sur Processeurs.

Vous devriez maintenant voir le nombre total de processeurs / unités de traitement dans les cœurs de votre PC.

Etape 3 - Écrire/exécuter le code π de la série

Ainsi, vous apprendrez à écrire et à exécuter un simple code C pour calculer π . Ce code effectuera une intégration numérique (avec 300 000 itérations) sur une représentation de la fonction x de π . Voici le code :

```
#include <math.h> // math library

#include <stdio.h> // Standard Input/Output library


int main(void) {

long num_rects = 300000;           //10000000000;

long i;

double x,height,width,area;

double sum;


width = 1.0/(double)num_rects;    // taille d'un segment
sum = 0;


for(i = 0; i < num_rects; i++) {

    x = (i+0.5) * width;           // x: distance au centre du ième segment

    height = 4/(1.0 + x*x);

    sum += height;                 // Somme des segments individuels

}                                  // Approximation de la valeur de Pi avec l'équation finale

area = width * sum;
```

```

printf("n");
printf(" Calculated Pi = %.16fn", area);
printf(" M_PI = %.16fn", M_PI);
printf("Relative error = %.16fn", fabs(area - M_PI));
return 0;
}

```

Le code a été écrit à l'aide de l'éditeur **vim** dans l'environnement terminal Linux comme suit:

- Commencez par changer le répertoire dans le dossier Desktop en entrant la commande

cd Desktop

- Créez maintenant un fichier nommé Pi.c en entrant la commande `vim Pi.c`
- Appuyez sur **i** pour lancer la modification.
- Tapez ou copiez le code comme indiqué précédemment.
- Enregistrez le fichier en entrant **Esc : wq**, qui vous ramène à l'invite de commande (\$)

Etape 4 - Compiler et exécuter

Compilez le code à l'aide de la commande :

mpicc Pi.c -o Pi

Cette procédure génère un nom de fichier exécutable Pi situé dans le dossier Desktop.

Enfin, exécutez le code à l'aide de la commande

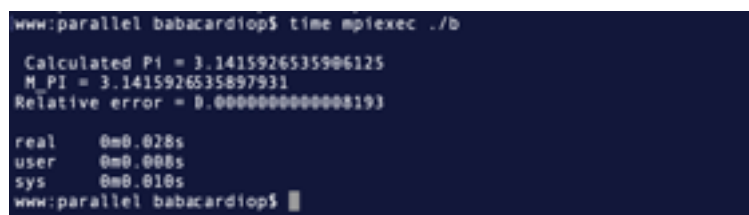
time mpiexec Pi

ou

time mpiexec ./Pi

Le code générera la sortie décrite dans l'exécution suivante. Vous pouvez voir quatre sorties similaires, chacune émanant d'un cœur du processeur à quatre cœurs:

N'hésitez pas à modifier le code en modifiant la valeur de `num_rects` (actuellement égale à 300 000) et observez la valeur modifiée du calcul de Pi. Le temps de calcul du code devrait changer en conséquence. Notez que ce calcul a été exécuté sur un seul cœur du processeur.



```

www:parallel babacardiop$ time mpiexec ./b
    Calculated Pi = 3.1415926535906125
      M_PI = 3.1415926535897931
Relative error = 0.0000000000008193

real    0m0.028s
user    0m0.008s
sys     0m0.010s
www:parallel babacardiop$

```

La durée d'exécution de (0m0,028s) est fonction du processeur utilisé dans cet exemple, qui est un iCore7. Votre processeur peut avoir une vitesse plus lente, et par conséquent le temps de calcul peut être plus élevé.