Calcul distribué de la valeur de π

Introduction

π est une constance mathématique représentant le ration entre la circonférence de n'importe quel cercle par rapport à son diamètre. Dans ce TP, on calculera sa manière de manière distribuée à l'aide de la bibliothèque OpenMPI.

I. Méthode de Monte Carlo

A. Présentation

Cette méthode vise à calculer une valeur numérique avec des techniques probabilistiques. Le but de cet exercice est de traduire l'algorithme présenté en C en un algorithme exécutable dans un environnement distribué. Le but réel est donc de se familiariser un minimum avec la bibliothèque OpenMPI.

Le code est disponible à l'adresse suivante ou dans le fichier *« TP1_RE51_exo1_Vincent_MERAT.c »*. Le résultat de l'exécution de ce code est disponible à l'adresse suivante.

B. Explications

L'ajout des fonctions OpenMPI pour rendre l'algorithme distribué était assez simple. Il s'agit en effet de fonctions pour initialiser les différents nœuds ainsi que leur dire quoi calculer et ensuite pour récupérer leurs résultats.

Le fait de distribuer le calcul permet de multiplier facilement et plus rapidement le nombre d'itérations de calculs. Dans l'exemple à cette adresse, on peut voir que l'algorithme a été réalisé de sorte à lancer 600 000 000 d'itérations, or comme ces itérations sont distribuées sur 6 nœuds, le résultat est donc calculé très rapidement.

Une fois que chaque nœud (processus) a fini ses calculs, il renvoie son résultat au maître qui lui va calculer la valeur de π une fois toutes les réponses récupérées.

Bien sûr, plus le nombre d'itérations est élevé, plus la valeur calculée de π sera précise. La contrepartie sera que le temps de calcul sera plus long.

II. Formule de Ramanujan

A. Présentation

L'efficacité de la méthode précédente était discutable du fait qu'il fallait un grand nombre d'itérations pour commencer à obtenir une valeur relativement précise (dépassant les 2 chiffres après la virgule). Dans cette partie, nous allons étudier une autre méthode utilisant la formule de Ramanujan suivante :

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(n!)^4 396^{4n}}$$

Le code est disponible à l'adresse suivante ou dans le fichier *« TP1_RE51_exo2_Vincent_MERAT.c »*. Le résultat de l'exécution de ce code est disponible à l'adresse suivante.

B. Explications

L'une des principales difficultés afin d'utiliser cette méthode était de transcrire la formule précédente en C. C'est pourquoi on utilisera également la bibliothèque *« math.h »* pour calculer les puissances et on a écrit une fonction *« factorial »* pour calculer les factorielles. Enfin, afin de ne pas encombrer le *main*, on a écrit une fonction *« calcul_n »* dans lequel sera retranscrit le calcul.

Ce programme permet donc de distribuer le calcul (les factorielles demandant un nombre important de ressources) aux autres nœuds. On a choisi d'utiliser une liste que le maître rempliera avec les différents éléments à calculer. Chaque nœud aura donc pour but de récupérer un calcul dans la liste, de l'effectuer et de rendre le résultat. Une fois tous les résultats obtenus, le maître pourra facilement en déduire la valeur de π.

Cette seconde méthode est donc plus efficace que la précédente car elle demande tout de même moins de ressources pour obtenir un résultat aussi précis.