Polytech Sorbonne

Compte rendu TP3 - HPC

Ensemble de Mandelbrot - Master/Worker Algorithm

Romaric Kanyamibwa

MAIN4

Année 2017 - 2018

Table des matières

1 Introduction

Comme dans le TP précédent l'objectif de ce TP est de mettre en œuvre le calcul parallèle des ensembles de Mandelbrot en utilisant le modelé master/worker ensuite nous allons comparer le temps d'exécution celui de la version séquentielle et de la version parallele normale. Enfin l'implentantation du master/worker nous aidera à decouvrir une nouvelle façon de paralléliser un programme séquentielle.

2 Ensemble de Mandelbrot

On définit comme ensemble de Mandelbrot les points du plan complexe $\mathbb{C}$ vérifiant la suite récurrente de nombres complexes suivante:

$$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases}$$

quand z_n converge.

3 Programme Parallèle avec Master-Slave

Pour comparer la performances de versions parallèle et de la version séquentielle nous allons dessiner l'ensemble sur le plan complexe.

Nous allons commencer avec la version non-parallélise qui se trouve dans le fichier *mandel.c* et ensuite nous allons implémenter le modelé parallèle master/slave sur le fichier $mandelbrot_master_worker.c$.

Dans lees images ci-dessous nous avons travaillé avec les points z définit sur le domaine D=$\{z = x + iy$, $(x, y) \in$ [-1.5,-0.1]x[-1.3,0.1]$\}$
Dans ces images les Incréments sont définit par $x_{inc} = \frac{x_{max} - x_{min}}{w-1}$ et $y_{inc} = \frac{y_{max} - y_{min}}{h-1}$ seront : x_{inc} =0.000250313 , y_{inc} =0.000250313
Les images sont de dimensions 800x800

Figure 1 – *Prof=10*

Figure 2 – *Prof=15*

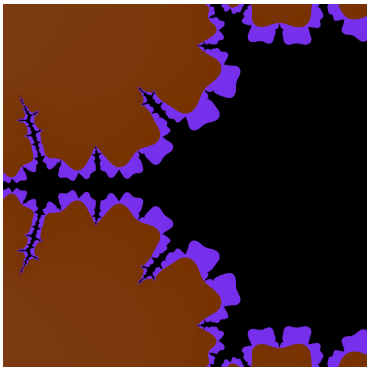

Figure 3 – *Prof=20*

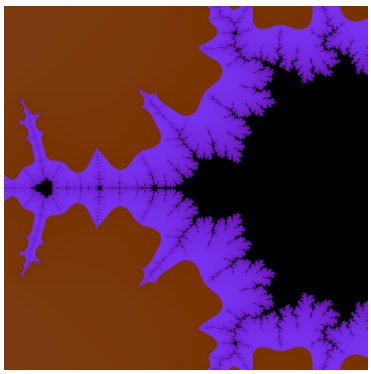

Figure 4 – *Prof=50*

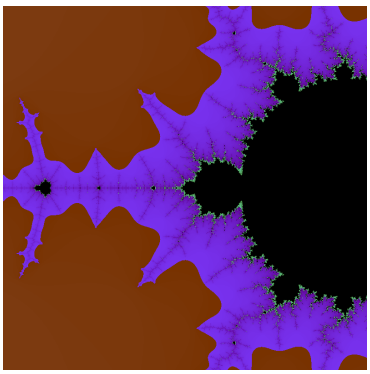

Figure 5 – *Prof=100*

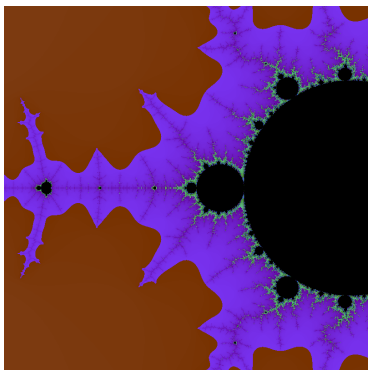

Figure 6 – *Prof=1000*

Après avoir vu comment le profondeur du calcul influe sur le graphe nous allons maintenant voir comment l'efficacité et le speed-up varier selon le nombre de processeur qu'on utilise.Nous allons fixer un profondeur de 10 000 et on exécute tous notre calcul sur les données par defaut.

NLINES=1			
Nombre de Processeurs	t [en sec]	Speed-up	Efficacité
1	5.78661	-	-
2	6.09112	0,95	47.5%
4	2.32685	2,4868	62.17%
8	1.89736	3.049	38.12%

Dans le modèle master/worker chaque worker calcule un bloc de taille *nlines* et dès que un worker finit son calcul , le master lui donne un autre calcul à faire.Donc pour améliorer d'avantages notre programme nous allons faire varier la taille du bloc calculer par un worker.

NLINES=2			
Nombre de Processeurs	t [en sec]	Speed-up	Efficacité
1	5.78661	-	-
2	6.19286	0,9344	46.72%
4	2.29905	2,5169	62.92%
8	1.91743	3.017	37.72%

NLINES=4			
Nombre de Processeurs	t [en sec]	Speed-up	Efficacité
1	5.78661	-	-
2	6.15758	0,9397	46.98%
4	2.55053	2,2687	56.71%
8	1.89716	3.05	38.12%

NLINES=16			
Nombre de Processeurs	t [en sec]	Speed-up	Efficacité
1	5.78661	-	-
2	6.18471	0,9356	46.78%
4	2.31788	2,4965	62.41%
8	1.87148	3.091	38.64%

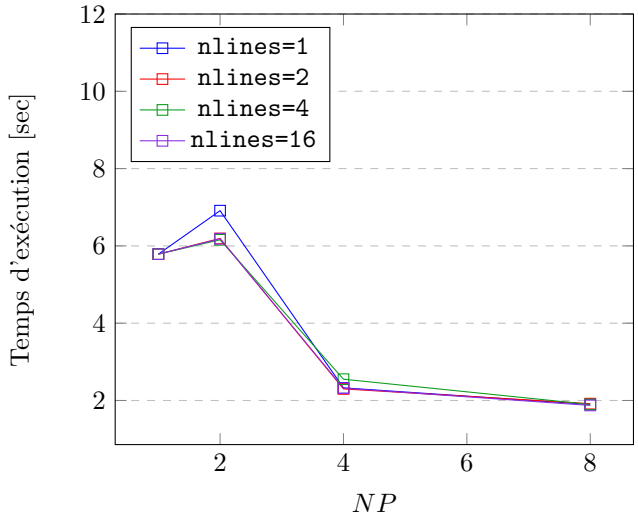

Figure 7 – *Temps d'exécution du programme pour de nlines différents*

4 Conclusion

En faisant varier la taille du bloque calculer par chaque processus et en changeant le nombre de processeurs on remarque qu'on arrive à un speed up de 3.Le speed-up optimale est atteint avec 8 processus , au dessus

de 8 le speed-up ne change pas de façon significatif. Malgre l'optimalité du speed up à 8 processeurs on voit que pour 8 processeurs l'efficacité est la plus faible.Les ressources qu'on dédie au calcul ne change pas de façon significative le temps de calcul. En effet on voit que l'efficacité est optimale quand on utilise 4 processeurs même si le speed-up n'est que de 2,3 .

Dans ce TP aussi on remarque que la taille du bloc n'as pas une très grande influence sur le temps du calcul, parce que au final on voir pour différents *nlines* le temps d'exécution, le speed-up et l'efficacité reste quasiment les mêmes.

Avec cette implémentation du modèle master-worker on arrive à avoir un programme 3 fois plus rapide quelque chose que on avait pas avec la version parallèle simple.Néanmoins pour avoir cette accélération il faut dédier des ressources disproportionnelles.