École Nationale Supérieur des Techniques Avancées

OS202

Travaux Dirigés n°2

GALVÃO Mateus

Palaiseau

2024

Table des matières

1 Exercice 1

1.1 Item 1

Un scénario sans interblocage est une situation dans laquelle deux processus ou plus ne peuvent pas progresser parce que chacun attend la libération de ressources détenues par l'autre. Cela crée une impasse, où aucun des processus ne peut avancer, entraînant une condition d'attente infinie.

1.2 Item 2

Un scénario avec interblocage se produit lorsque deux processus attendent mutuellement la finalisation l'un de l'autre en raison de l'utilisation de ressources partagées entre eux.

1.3 Item 3

La probabilité de blocage peut être considérée comme relativement élevée en raison de l'utilisation fréquente de ressources partagées entre plusieurs processus dans les principaux scénarios de programmation parallèle, en particulier lorsqu'il est nécessaire de limiter les dépenses de ressources.

2 Exercice 2

2.1 Item 1

La loi d'Amdahl est la suivante :

$$S = \frac{1}{(F_s + \frac{F_p}{n})}$$

où :
— S est l'accélération, c'est-à-dire le facteur d'accélération parallèle.
— F_s est la fraction séquentielle du programme (partie qui ne peut pas être parallélisée).
— F_p est la fraction parallélisable du programme (partie qui peut être parallélisée).
— n est le nombre de processeurs.

Dans le cas d'Alice, elle mentionne que la partie parallèle représente 90% du temps d'exécution en séquentiel. Par conséquent, $F_s = 0.1$ et $F_p = 0.9$. Donc :

$$S = \frac{1}{(0.1 + \frac{0.9}{n})}$$

Si n devient très grand, la fraction $\frac{0.9}{n}$ tendra vers zéro, et l'accélération maximale approchera idéalement $\frac{1}{0.1} = 10$.

Ainsi, l'accélération maximale que peut obtenir Alice avec son code est d'environ 10 fois, en supposant que le nombre de processeurs est suffisamment grand.

2.2 Item 2

Pour déterminer un nombre raisonnable de nœuds de calcul n sans gaspiller de ressources CPU, il est important de considérer les coûts associés à la parallélisation. La loi d'Amdahl montre que même avec une proportion importante du programme parallélisée, il existe une limite à l'accélération en raison de la partie séquentielle du code.

Dans le cas d'Alice, la partie parallélisée représente 90% du temps d'exécution, ce qui est relativement élevé. Cependant, même avec une accélération maximale de 10 fois, cela signifie que la partie séquentielle du code limitera toujours l'efficacité de la parallélisation.

Une bonne limite se situe donc autour de 60 à 70%, car au-delà, comme nous l'avons dit, divers facteurs commencent à influencer le résultat final. En utilizant 65%, donc :

$$0.65 = \frac{1}{(0.1 + \frac{0.9}{n})} \rightarrow n \approx 15$$

2.3 Item 3

La loi de Gustafson pour l'accélération est la suivante :

$$S = n + (1 - n)f$$

, où $f = 1 - p$

Si Alice obtient une accélération maximale de quatre en doublant le nombre de nœuds de calcul, donc :

$$4 = n + (1 - n) \cdot 0.1n \approx 5$$

Donc, l'accélération maximale avec la même quantité de noeuds sera :

$$S' = 5 + (1 - 5) \cdot 0.1 \cdot 2S' = 4.2$$

Ainsi, en doublant la quantité de données à traiter avec une complexité d'algorithme parallèle linéaire, Alice peut espérer une accélération maximale d'environ 4.2 en utilisant la loi de Gustafson.

3 Exercice 3

3.1 Item 1

Temps de calcul et speedup pour différents nombres de processus du code Mandelbrot par ligne :
— 1 tâche (sans parallèlisation) : 2.473s. Speedup = 1.
— 2 tâches : 1.376. Speedup = 1.797.
— 3 tâches : 1.056. Speedup = 2.342.
— 4 tâches : 0.935. Speedup = 2.645.

On observe ainsi que l'accélération augmente avec le nombre de processus, car davantage d'informations peuvent être traitées en moins de temps.

3.2 Item 2

Temps de calcul pour différents nombres de processus du code Mandelbrot avec la stratégie maître–esclave :
— 2 tâches : 2.734. Speedup = 0.904.
— 3 tâches : 1.479. Speedup = 1.672.
— 4 tâches : 1.247. Speedup = 1.983.

On se rend donc compte que plus il y a de processus, plus l'accélération est importante grâce à cette stratégie. Cependant, elle s'est avérée moins efficace que la méthode précédente, puisqu'un des processus (maître) était réservé à la gestion des autres (esclaves).