

TD 1 - Fondements de la conception des ordinateurs

Exercice 1.1 (Familiarisation avec la loi d'Amdahl)

Un *mode vectoriel* est un mode dans lequel on effectue des opérations de haut niveau portant sur des vecteurs au même titre que l'on manipule habituellement des opérations sur de simples scalaires. On gagne ainsi en concision et en vitesse d'exécution.

Pour cet exercice, on considérera l'amélioration d'une machine liée à l'ajout d'un mode vectoriel. Quand un calcul s'exécutera en mode vectoriel, il sera 20 fois plus rapide qu'en mode d'exécution normal. Le pourcentage du nombre d'opérations effectuées en mode vectoriel est appelé *pourcentage de vectorisation*.

Question 1.1.1

Tracez un graphe donnant l'accélération liée au calcul réalisé en mode vectoriel en fonction du pourcentage de vectorisation. Quel est le pourcentage de vectorisation nécessaire pour obtenir une accélération de 2 ?

Question 1.1.2

L'accélération est maximale lorsque la totalité des instructions est effectuée en mode vectoriel. Quel est le pourcentage de vectorisation nécessaire pour obtenir la moitié de l'accélération maximale atteignable à partir du mode vectoriel ?

Question 1.1.3

Le pourcentage de vectorisation mesuré pour des programmes est de 70%. Le groupe de conception matériel dit qu'il peut doubler la vitesse du mode vectoriel en intégrant un dispositif spécial. Vous vous demandez si l'équipe compilation pourra augmenter l'utilisation du mode vectoriel pour accroître les performances d'une autre façon. Quel pourcentage de vectorisation (relative à une utilisation normale) faudrait-il pour obtenir le même gain de performance ?

Exercice 1.2 (Loi de Gustafson)

On supposera qu'une machine parallèle à N processeurs permet de diviser le temps d'exécution d'une fraction de code parallélisé par N . On appellera T_s le temps d'exécution de la partie séquentielle d'un programme, T_p celui de la partie parallèle et T_i le temps total sur i processeurs.

Question 1.2.1

Sachant que la fraction séquentielle d'un programme à paralléliser est s , exprimez le speed-up pour une machine parallèle à N processeurs. Quel est le speed-up maximum envisageable ?

Question 1.2.2

Que devient ce speed-up si, pour une proportion de code séquentiel s fixée, on augmente grandement le nombre de processeurs ?

Question 1.2.3

La loi de Gustafson donne le gain que l'on peut avoir avec N processeurs en exécutant un programme parallélisé dans le cas où il est possible d'augmenter

la quantité de données traitées (contrairement à la loi d'Amdahl, qui s'applique à une quantité de données fixe). Elle traduit le fait que l'on peut traiter plus de données en augmentant le nombre de processeurs.

En théorie, chaque processeur peut traiter autant de données. Ce qui veut dire que lorsque l'on cherche à comparer le temps mis par 1 processeur pour exécuter le même calcul que sur N processeurs, on a : $T_1 = T_s + (N * T_p)$.

Que devient ce temps total si l'application s'exécute sur N processeurs ? Déduisez-en le speed-up obtenu sur N processeurs.