TD 2 Performance

2.1 Modèle d'Amdhal

1. La fraction séquentielle d'une application représente 4% du temps d'exécution à 1 processeur. Calculer l'accélération et l'efficacité à 2^n processeurs ($2 \leq n \leq 7$). Quelle est la borne supérieure de l'accélération ?
2. La mesure des accélérations des programmes A et B founit les résultats suivants.

Procs	2	3	4	4	6	7	8
A	1,90	2,73	3,47	4,16	4,8	5,38	5,93
B	1,94	2,72	3,34	3,82	4,20	4,49	4,72

Calculer la fraction séquentielle pour chacune des exécutions. Comment peut-on décrire le comportement parallèle des deux programmes ?

2.2 Débit asymptotique et demi-performance

1. Un algorithme est composé d'un calcul scalaire et d'un calcul vectoriel sur un vecteur de taille N. Sur 1 processeur, le temps du calcul scalaire est s et le temps du calcul vectoriel est dN, où N est la taille du vecteur. On parallélise cet algorithme en répliquant le calcul scalaire et en distribuant le calcul vectoriel.

   Calculer le débit asymptotique, la demi-performance et l'accélération de l'algorithme sur une machine à p processeurs.
2. Un deuxième algorithme est composé d'une réduction et d'un calcul vectoriel. Sur 1 processeur, le temps de la réduction est aN et le temps du calcul vectoriel est dN. Mêmes questions.

2.3 Iso-efficacité

1. On considère la réduction d'un vecteur de taille n sur p processeurs (on suppose toujours n multiple de p). Définir la fonction d'iso-efficacité. Calculer les valeurs de n nécessaires pour atteindre une efficacité de 0,9 sur 2, 32, 512 et 1024 processeurs.
2. La relation d'iso-efficacité sur un système est notée $n > f(p)$; la mémoire nécessaire pour un problème de taille n est notée $M(n)$. Etablir la hiérarchie d'extensibilité pour les problèmes suivants :

$f(p)$	Cp	Cp	$Cp \log p$	$C\sqrt{p}$	$C\sqrt{p}\log p$
$M(n)$	n	n^2	n^2	n^2	n^2

2.4 Extensibilité

On considère deux modèles de performances, qui évaluent l'interaction entre taille du problème, temps d'exécution et parallélisme.

- Mémoire constante (MC) : l'encombrement mémoire par processeur est fixée. R_{MC} est le rapport entre le temps d'exécution du problème traité sur p processeurs et le temps d'exécution du problème **initial** sur 1 processeur.

- Temps de réponse constant (TC) : le temps d'exécution est fixe. R_{TC} est le rapport entre l'encombrement mémoire total du problème traité sur p processeurs et l'encombrement mémoire du problème **initial** sur 1 processeur.

Produit de matrices

Un algorithme séquentiel de paramètre caractéristique n effectue n^3 calculs sur n^2 données (exemple produit de matrices). On suppose l'algorithme parfaitement parallélisable.

1. Sur 1 processeur, on peut traiter un problème de paramètre 8 en temps 1. Quel problème peut-on traiter sur 4 processeurs en MC ? Quel est le temps d'exécution parallèle ? Quel problème peut-on traiter sur 4 processeurs en TC ? Quel est l'encombrement mémoire total ? parallèle ?
2. Calculer le paramètre caractéristique de l'algorithme parallélisé sur une machine à p processeurs (noté $n(p)$) dans le modèle MC et dans le modèle TC.
3. Calculer R_{MC} et R_{TC}.

Tree-codes

La complexité séquentielle de l'algorithme de Barnes-Hut est $O(\Delta t^{-1}\theta^{-2}N\log N)$. Lorsque cet algorithme est appliqué à la simulation de l'évolution des galaxies, il présente trois facteurs d'erreurs : la résolution spatiale contribue pour $1/\sqrt{N}$; la résolution temporelle contribue pour Δt^2 et l'approximation pour θ^2. On suppose les erreurs indépendantes, *i.e.*

$$E = a\frac{1}{\sqrt{N}} + b\Delta t^2 + c\theta^2.$$

1. On multiplie N par k. Par quel facteur faut-il multiplier les autres paramètres pour obtenir une évolution cohérente de l'erreur ? Calculer alors la nouvelle complexité séquentielle.
2. On suppose l'application complètement parallélisable. Comparer la résolution spatiale obtenue par un scaling naif (on ne tient pas compte de l'influence de Δt et θ sur l'erreur) et par un scaling cohérent, dans le modèle TC.
3. En imposant un temps de réponse constant égal à celui de la configuration 1M, 1 processeur, comparer le nombre de processeurs nécessaires pour multiplier la résolution spatiale par 16 en scaling naïf et en scaling cohérent.