TD 1 Performances

1.1 Analyse des données du Top500

1. Calculer R_{peak}/C, R_{max}/C et R_{peak}/CW pour le TOP5. Que remarque-t-on ?
2. En utilisant l'interface de statistiques, donner la part des architectures MPP et Cluster dans le TOP 5, le TOP 10, et le TOP 500.
3. Recueillir les donnéeàs pour tous les processeurs de la famille *BlueGene/P*. Pour chaque configuration, donner l'accélération relativement à la plus petite configuration, et la comparer au nombre de processeurs.
4. Vous utiliserez l'interface *List Statistics*, choix *Number of Processors*. Donner la zone où se situe la médiane du nombre de processeurs (ou de cores), pour les années 2000 à 2009 (liste de Novembre). La médiane est la valeur m telle que 50% des valeurs de l'échantillon soient inférieures ou égales à m.

1.2 Accélération

1. 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 ?

1.3 Autres modèles de performances

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 naïï (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.