

Parallélisme

TP4

Ce travail est individuel et vous avez trois semaines pour le réaliser, soit jusqu'au 19 décembre 2018.

Analyse de performance

Dans ce TP, vous allez devoir mesurer et analyser les performances du code que vous avez écrit lors du TP2.

Speedup

Le *speedup* est défini comme le gain de temps d'exécution d'une application parallèle par rapport à celui d'une application séquentielle. Il est donc calculé avec le rapport entre le temps requis par un seul coeur de calcul pour exécuter un travail donné et le temps requis par p coeurs de calcul pour exécuter ce même travail :

$$S = \frac{T_{seq}}{T_{par}} \quad (1)$$

Idéalement on souhaite que S soit égal à p quel que soit le nombre de processeurs utilisés.

Modèle de performance

Un modèle de performance donne une expression du temps d'exécution, et par conséquent du *speedup*, d'une application. Dans le cas de l'équation de la chaleur en deux dimensions sur un domaine carré découpé en bandes, le modèle de performance d'une itération de l'algorithme peut s'écrire :

$$T_{par} = \alpha \frac{N^2}{p} + \beta N \quad (2)$$

où N est la taille d'un côté du domaine, p le nombre de processus, α une constante représentant le temps de calcul et β une constante représentant le temps de communication.

Il est possible de réécrire l'expression 2 sous la forme :

$$\frac{T_{par}}{N} = \alpha \frac{N}{p} + \beta \quad (3)$$

qui prends alors la forme de l'équation d'une droite où α est la pente et β est l'ordonnée à l'origine.

En effectuant plusieurs mesures de temps d'exécution en variant les valeurs de N et de p on peut trouver les valeurs des constantes α et β en effectuant une regression linéaire.

Loi de Amdahl

La loi de Amdahl donne le gain en temps d'exécution d'une tâche que l'on peut attendre d'une application dont on améliore une partie, sachant quelle proportion du travail total de l'application profite de l'amélioration.

Dans le cadre du parallélisme, on prends en compte quelle partie de l'application peut être parallélisée.

Supposons que l'on a une quantité W de travail à accomplir et que l'on souhaite observer comment le *speedup* varie selon une portion γ de code qui s'exécute séquentiellement ($0 \leq \gamma \leq 1$).

En notant R le nombre d'opérations qu'un processeur peut effectuer par seconde (p. ex. en *flops*), on a que :

$$T_{seq} = \frac{W}{R} \quad (4)$$

Dans une application parallèle, on peut imaginer qu'une quantité de travail γW est exécutée en séquentiel. On a donc que :

$$T_{par} = \frac{\gamma W}{R} + \frac{(1 - \gamma)W}{pR} \quad (5)$$

En substituant T_{seq} et T_{par} dans l'équation 1 on obtient :

$$S = \frac{T_{seq}}{T_{par}} = \frac{1}{\gamma + \frac{1-\gamma}{p}} \leq \frac{1}{\gamma} \quad (6)$$

La loi de Amdahl permet de modéliser le *speedup* d'une application si celle-ci se divise en une portion γ non parallélisable et une portion $1 - \gamma$ parallélisable et que l'application ne souffre pas d'autre source de ralentissement (p. ex. communications).

Travail à accomplir

Vous devez effectuer une étude de performance de votre application (équation de la chaleur sur domaine carré parallélisé avec MPI). Mesurez le temps d'exécution de l'algorithme. Mesurez le temps d'exécution de la boucle principale, dont la performance est modélisée par l'équation 2 et le temps d'écriture sur le disque. Pour le temps d'écriture, écrivez le résultat sur la partition `~/scratch/` (voir wiki baobab).

Effectuez vos mesures de temps d'exécution sur le cluster *Baobab* de l'université avec plusieurs nombres de coeurs et plusieurs valeurs de N . Par exemple 1, 2, 10, 20, 40, 60, 80, 100 et 120 coeurs et $N = 2000, 3000, 4000$. Choisissez un nombre d'itérations qui produit une exécution séquentielle de l'ordre de 10 à 20 minutes.

Une fois vos mesures effectuées, appliquez une regression linéaire sur les mesure mises sous la forme de l'équation 3 afin de déterminer la valeur des paramètre α et β .

Estimez ensuite la part non parallélisable γ de votre application en comparant le temps d'exécution de la boucle principale en séquentiel et le temps d'écriture sur le disque.

Tracez les graphiques contenant vos mesures ainsi que la droite issue de la régression. Tracez une courbe de *speedup* pour chacune des valeurs de N , une fois en prenant en compte uniquement le temps d'exécution de la boucle principale et une autre fois en prenant en compte le temps de calcul de la boucle principale plus le temps d'écriture sur le disque. A l'aide de l'équation 6, faites apparaître sur ce dernier graphique le *speedup* théorique d'une application dont la seule limite de parallélisation serait sa portion non parallélisable γ , pour des valeurs de $\gamma = 0.1$ et $\gamma = 0.05$ et pour la valeur de γ déterminée précédemment.

Rédigez un rapport où vous discutez de ces différents résultats et du comportement de votre implémentation et tentez d'expliquer les valeurs trouvées ainsi que les graphiques.

Rendu

Vous devez rendre un rapport structuré de la manière suivante :

1. Une introduction présentant l'algorithme, ce que l'on cherche, pourquoi et comment l'obtenir
2. Une présentation de vos résultats (graphiques de temps d'exécution, de *speedup*, de regression, valeurs des paramètres estimés, ...)
3. Une discussion traitant de vos résultats ainsi que de l'implémentation utilisée

Le contenu ainsi que la qualité de la structure du rapport sont évalués. Veillez à rédiger de façon claire, concise et structurée. Produisez des graphiques clairs et complets (titres et unités sur les axes) et présentez vos résultats de façon lisibles (p.ex. dans des tableaux, pas de capture d'écran ou de données présentées de façon non structurées).

Vous devez rendre un rapport au format PDF ainsi que le code que vous avez utilisé pour effectuer les mesures dans une archive au format **zip** portant votre nom et prénom sur moodle dans :

Parallélisme → TP4