

TD #1: OpenMP – Cartographie et Réduction de Vecteurs de caractères

Jonathan Rouzaud-Cornabas (jonathan.rouzaud-cornabas@insa-lyon.fr)

1 Consignes

Nous vous rappelons que vous bénéficiez de Intel Parallel Studio XE (simple demande sur le site d'Intel).

Nous vous fournissons une machine virtuelle avec clang/llvm 3.8 et aftermath pour visualiser le code parallèle.

Il nous paraît important de visualiser les performances de votre code avec Intel Advisor/VTune pour les micro optimizations. Pour l'optimisation du code parallèle, nous vous recommandons Aftermath (un guide d'utilisation est fourni sur moodle).

Pour chaque sous partie (vecteur-vecteur, matrice-vecteur, matrice-matrice), il est nécessaire en plus d'avoir un code fonctionnel de fournir les performances en passage à l'échelle fort et faible. Vous pouvez utiliser les timer haute précision (sous partie chrono de la STL) pour calculer les performances.

1.0.1 Passage à l'échelle fort

En passage à l'échelle fort, on garde la taille du problème (taille des vecteurs et des matrices) et on augmente le nombre de processeurs. Le minimum est de tester sur 1, 2 et 4 coeurs dans votre cas. Si possible, aller plus loin.

1.0.2 Passage à l'échelle faible

En passage à l'échelle faible, on augmente la taille du problème avec l'augmentation du nombre de coeurs. En pratique, vous multipliez la taille du problème par 2 pour 2 coeurs et 4 pour 4 coeurs.

2 Exercice

Pour information, le fonctionnement de l'algorithme sous jacent est assez proche de MapReduce Word-Count. Le but est de compter le nombre d'occurences de chaque lettre présent dans une matrice.

Question 2.1 Ecrivez le code séquentiel permettant d'initialiser une matrice avec un ensemble aléatoire de lettre (minuscule a-z).

Question 2.2 Ecrivez le code permettant d'afficher le résultat.

Question 2.3 Ecrivez le code séquentielle permettant de faire le calcul.

2.1 Parallel for

Question 2.4 Ecrivez le code parallèle permettant de faire le calcul sur le problème complet.

Question 2.5 Proposer un code de découpage du problème en sous problème.

Question 2.6 Proposer une adaptation de votre code parallèle pour calculer un sous problème.

Question 2.7 Proposer un code permettant de sommer le résultat des sous-problème.

2.2 Task

Question 2.8 Récrivez votre code de calcul du problème complet avec tâches et plus des boucles for.

Question 2.9 Réécrivez votre code de découpage, calcul de sous problèmes et de leurs sommes avec des tâches (pensez aux dépendances).

2.3 Du caractères au mot

Question 2.10 Proposer un code parallèle pour reproduire la procédure suivante:

- Construire tous les mots possibles à gauche, à droite, en haut et en diagonale (x4) de tailles N (par défaut, 6)
- Refaire la même chose mais au lieu de faire le calcul des lettres, faites le pour les mots.