TDTP8 : Produit matriciel et routines BLAS

Version du 5 décembre 2017

Récupérer le code du produit matriciel séquentiel $C+ = A \times B$ (les 3 matrices A, B et C sont toutes considérées comme étant carrées, d'ordre N). L'exécution pour différentes valeurs de N pourra se faire avec le script `run.sh`.

1. (**Produit matriciel naïf**) Sur votre machine locale, mesurer la performance du produit matriciel "naïf" (3 boucles imbriquées), par exemple pour $N = 512$, et comparer la à la performance crête de votre machine.

   Déterminer les options d'optimisation à la compilation qui permettent d'améliorer effectivement vos performances.

   Une fois ces options d'optimisation déterminées, mesurer les performances de votre produit matriciel lorsque N augmente à l'aide du script `run.sh`. Qu'observez-vous ?

2. (**Produit matriciel récursif par bloc**) Ecrire un pseudo-code C récursif implémentant un produit matriciel par bloc pour des matrices allouées dynamiquement en mémoire sous la forme de tableaux 1D. On supposera que N est une puissance de 2. La fonction récursive à implémenter aura la signature suivante :

```
mm (int crow, int ccol, /* Upper-left corner of C block */
    int arow, int acol, /* Upper-left corner of A block */
    int brow, int bcol, /* Upper-left corner of B block */
    int n,              /* Blocks are n x n */
    int stride,         /* Stride for whole matrix */
    float *A, float *B, float *C);
```

   Implémenter cet algorithme et déterminer les valeurs de `seuil` qui permettent d'obtenir de meilleures performances que le produit matriciel naïf vu ci-dessus.

3. (**Produit matriciel parallèle avec OpenMP**) Comment paralléliser le produit matriciel naïf avec OpenMP ?

   Comment les performances varient en fonction du nombre de threads utilisés ?

   Mêmes questions pour le produit matriciel récursif par bloc.

4. (**BLAS**) A l'aide de la librairie OpenBLAS, utiliser la routine BLAS adaptée pour effectuer ce produit matriciel.

   Quel nombre de threads utilisez-vous ?

   Comment modifier le nombre de threads utilisés par la bibliothèque OpenBLAS ?

   Comment les performances varient en fonction du nombre de threads utilisés ?

   Documentation sur les BLAS :
   - BLAS reference card (Fortran) : fichier `blas-qref.pdf`
   - Interface C des BLAS :
     - voir fichier : `cinterface.pdf`
     - fichier en-tête : `cblas.h` (ainsi que `common.h` pour les OpenBLAS)
   - OpenBLAS : `https://github.com/xianyi/OpenBLAS/wiki`
   - GotoBLAS2 FAQ (OpenBLAS est un *fork* de GotoBLAS2) : `http://www.tacc.utexas.edu/tacc-projects/gotoblas2/faq`

