TP parallélisme

Compte-Rendu 10/12/2008

Mise en oeuvre des BLAS

Encadrant :

François PELLEGRINI

Binome { Seifeddine HABASSI
Mohamed Amine EL AFRIT

Table des matières

Chapitre 1

Travail à réaliser

1.1 Question 1

Il existe trois grands types d'opération correspendant à trois types de BLAS :
- BLAS 1 : Les routines du BLAS 1 sont responsables des opérations vecteur/vecteur.
- BLAS 2 : Les routines du BLAS 2 sont responsables des opérations matrice/vecteur.
- BLAS 3 : Les routines du BLAS 3 sont responsables des opérations matrice/matrice.
- La fonction zdotu est utilisée pour faire le produit scalaire de deux vecteurs x et y.

$$ZDOTU \leftarrow (\text{transpose de } x) * y = \sum_{i=0}^{n} x(i)*y(i)$$

- La fonction saxpy est utilisée pour appliquer une opération de type

```
y <-- alpha*x + y
```

où : alpha est un réél
x et y sont deux vecteurs de même taille.

Le résultat sera enrégistré dans le même y.

1.2 Questions 2-3

La fonction ddot1 est codée dans le fichier exo1a.c, alors que la fonction affiche est codée dans le fichier affiche.c.

1.3 Questions 4-5

Pour calculer la complexitée de notre fonction ddot1 et celle de ddot de blas,on a créé un main dans le fichier exo1a.c où on utilise les fonctions gettimeofdate et difftime(Cette fonction est definie dans le fichier timeperf.c).
En effet, on prend trois dates intercalées par les deux calculs :

```
t0--> lancement de ddot1
t1--> fin de ddot1 et debut de cblas_ddot
t2--> fin de cblas_ddot
```

Et donc pour comparer les deux durées, il suffit d'afficher (t1-t0) et (t2-t1).
En lancant le programme, on trouve, comme attendu, que les performances de notre ddot1 sont moins bonnes que celles de ATLAS. Et les résultats sont les mêmes.

1.4 Question 6

Le main, dans le fichier exo1b.c, a pour rôle de :
Faire varier la taille des vecteurs. Et dans chaque tour de boucle :

- Initilise les deux vecteurs x et y
- Lance notre ddot1, calcule les performances, puis écrit les MFLOPS dans le fichier ddot1.data et un récapitulatif sur l'écran.
- Lance notre ddot du blas, calcule les performances, puis écrit les MFLOPS dans le fichier ddot.data et un récapitulatif sur l'écran.
- Au cas ou la différence entre les résultats des deux ddot est plus grande que "0.2" il l'indique.

Les fichiers résultant de ce calcul sont enrégistrés dans le répertoire "calcul_results".

1.5 Question 7

Pour obtenir le graphe voulu, il suffit de lancer la commande :

```
gnuplot plot_ddot.gp
```

Un fichier plotperformance.png est alors créé.

Voilà un exemple du graphe obtenu :

On remarque que les performances de la routine ddot de ATLAS sont toujours meilleures à celle de la notre. Par contre, on peut remaquer que la taille du tableau atteint une certaine grandeure, les performances de la routine BLAS diminue. En effet, ceci est du à une mauvaise gestion de la mémoire cache par les BLAS. Donc, quand la taille du vecteur est grande, la machine est obligèe d'utiliser une mémoire moins rapide.

REMARQUE :
On remarque que pour des vecteurs de grande taille, les résultats donnés par les deux fonctions ne sont plus égaux. Pire, plus le tableau est grand plus la différence entre les résultats est grande. Ceci est du à la propagation des "erreurs" dans les opération sur les flottants.

1.6 Questions 9-10

Pour la routine dgemm, on a créé une fonction "produit_blas". Cette fonction prend en paramétre la taille des matrices, les deux matrices à multiplier et la matrice pour stocker le résultat.
Les trois algorithmes demandés sont définis dans le fichier exo2a.c.

1.7 Questions 11-12

En lancant le programme prog2a.c, celui-ci fait une suite de tests avec des tailles de matrice différentes puis affiche les diffétences entre les résultats trouvés par les

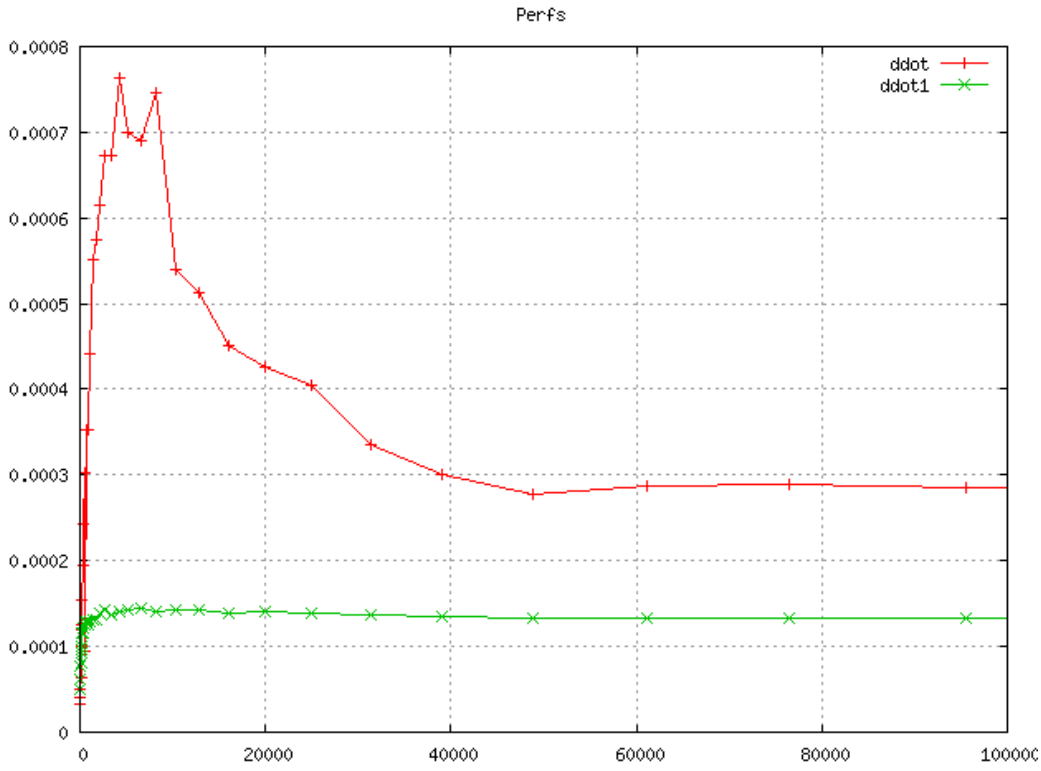

FIG. 1.1 – Performances

algorithmes et ceux de la routine BLAS.
On peut alors voir que les résultats sont toujours identiques. **REMARQUE :**
Les fichiers donnant des informations sur les performances de chaque algorithme sont hénérés dans le répertoire calcul_results.
Pour visualiser les performances de ces algorithmes en temps et en MFLOPS il suffit de lancer réspectivement les commandes :

```
gnuplot plot_exo2a_time.gp
gnuplot plot_exo2a_MFLOPS.gp
```

Les résultats sont alors stockés dans des fichiers PNG.
Ci_dessous deux courbes montrant les performances en temps et en MFLOPS obtenues. (dans les figures 1.2 et 1.3)

Pour des matrices de tailles assez petites, les performances de tous les algorithmes sont égales. Par contre, en augmentant la taille des matrices on remarque que les routines BLAS deviennent de loin plus performantes que les autres fonctions. Et on remarque aussi que pour l'algorithme kij, les performances deviennent mediocre quand la taille des matrices augmente.

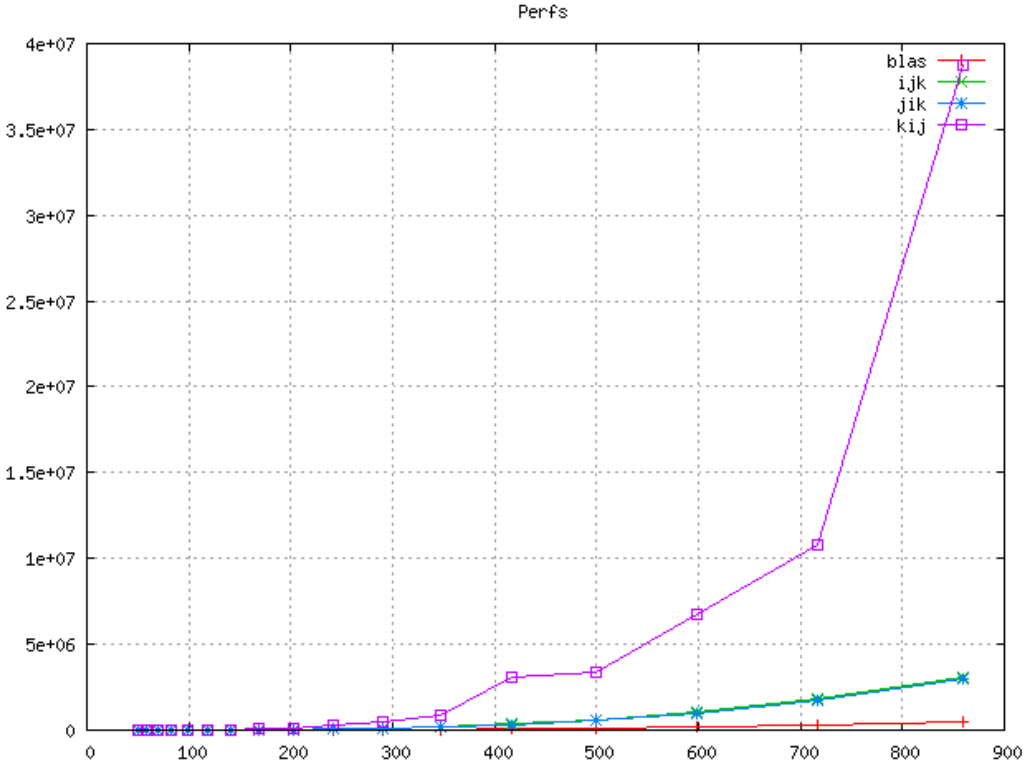

FIG. 1.2 – Performances en temps

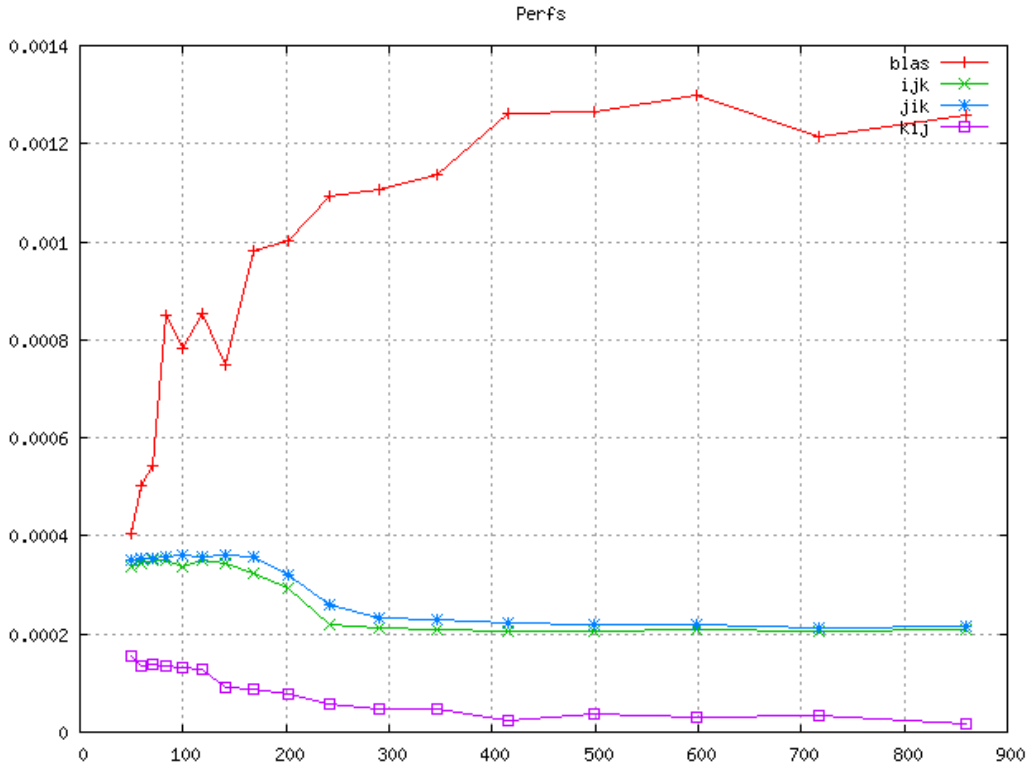

FIG. 1.3 – Performances en MFLOPS