



UE High Performance Scientific Computing

Examen de Algorithmes pour le Calcul à Hautes Performances

26 Janvier 2022

— Tous les documents sont autorisés.

1 Parallélisme mémoire partagée (6pts)

1.1 Question 1 (3pts)

On considère le pseudo-code ci-dessous :

```
do i=2,...,n
  x(i) = red(x(i))
  y(i) = blue(x(i), y(i-1))
end do
```

Dans ce code nous avons supposé

- que les deux tableaux `x` et `y` sont de taille `n` ;
 - la notation `output = function(input1, input2, ...)` pour les appels a fonction.
- Le coût des fonctions, est de 2 secondes pour `red` et 1 seconde pour `blue`.

Dessinez le graphe de dépendances du pseudo-code ; identifiez le parcours critique et calculez le speedup idéal atteint par une exécution parallèle avec un nombre infini de processeurs pour $n \rightarrow \infty$

1.2 Question 2 (3pts)

En pratique, les algorithmes de factorisation de type “block” sont presque toujours préférés à ceux de type “point” pour une exécution séquentielle autant que parallèle :

- Quels sont les facteurs qui limitent l’efficacité d’une factorisation de type “point” lors de son exécution sur un seul coeur ?

- Dans le cas d’une exécution séquentielle, quelles sont les propriétés des factorisations de type “block” qui permettent à ces algorithmes de mieux exploiter les caractéristiques des processeurs les plus couramment utilisés aujourd’hui ?
- Pourquoi les algorithmes de factorisation de type “block” sont-ils plus adaptés que les algorithmes de type “point” pour une parallélisation sur architecture à mémoire partagée ?

2 Parallélisme mémoire distribuée (11pts)

2.1 Iso-efficacité (8pts)

Considérez le code séquentiel suivant

```
for(i=0; i<m; i++)
    for(j=0; j<n; j++)
        x[i] = x[i] + x[i]*j;
```

où x est un vecteur de taille m et toutes les opérations sont sur des réels, virgule flottante en précision double. Sur un ordinateur parallèle à mémoire distribuée avec p processeurs, ce code peut être parallélisé à l’aide de communications collectives comme ceci

```
MPI_Scatter(x, m/p, MPI_DOUBLE,
            xloc, m/p, MPI_DOUBLE,
            0, MPI_COMM_WORLD);
for(i=0; i<m/p; i++)
    for(j=0; j<n; j++)
        xloc[i] = xloc[i] + xloc[i]*j;
MPI_Gather(xloc, m/p, MPI_DOUBLE,
           x, m/p, MPI_DOUBLE,
           0, MPI_COMM_WORLD);
```

Ici nous avons fait l’hypothèse que m est un multiple de p . En utilisant le modèle de Hockney avec paramètres γ (temps pour un calcul scalaire en précision double), α (latence d’accès au réseau) et β (temps de transfert d’un scalaire réel en précision double) :

1. (2pts) Écrivez le modèle $T(m, n, 1)$ pour le temps d’exécution du code séquentiel ;
2. (3pts) Écrivez le modèle $T(m, n, p)$ pour le temps d’exécution du code parallèle ;
3. (3pts) Étudiez l’iso-efficacité du code parallèle en faisant l’hypothèse simplificatrice $\alpha = 0$ et répondez aux questions suivantes :
 - (a) cas m variable et n fixe : si le nombre de processeurs augmente de p à p' , est-il possible de garder une efficacité constante en augmentant m ? si oui, de combien faut-il l’augmenter ?
 - (b) cas m fixe et n variable : si le nombre de processeurs augmente de p à p' , est-il possible de garder une efficacité constante en augmentant n ? si oui, de combien faut-il l’augmenter ?

Fournissez une explication pour la différence de comportement dans les deux cas.

2.2 Communications collectives (3pts)

La communication collective broadcast peut être aussi réalisée en utilisant une scatter suivie d'une allgather (par exemple, la version butterfly vue en cours). Illustrez le fonctionnement d'une telle implémentation, calculez son coût et comparez-le avec celui de la broadcast classique avec arbre binomial : sous quelles conditions cette implémentation est plus efficace que la classique ?

3 Méthodes de factorisation (3pts)

Trois méthodes de factorisation de matrices denses ont été étudiées en cours, notamment, Cholesky LU et QR . Expliquez les cas d'utilisation de ces trois méthodes. Pour résoudre un système linéaire avec matrice carrée et non-symétrique il est possible d'utiliser les factorisations LU et QR ; discutez les avantages et désavantages de l'une par rapport à l'autre du point de vue numérique (robustesse, stabilité) et des performances (coût opérationnel, efficacité, scalabilité).