

Cours : “Parallélisme”

Travaux dirigés

E. Goubault, S. Putot & O. Bouissou & S. Zennou & S. Krishnan

TD 4

14 février 2007

1 Parallélisation, avec petit

On considère le code (en format `petit`) suivant:

```
real X(101), Y(102)

for I=1, 100 do
  X(I) = Y(I+2)+1
  Y(I) = X(I+1)+1+X(I)
endfor
```

- Calculez les dépendances de flot, de sortie et anti, avec `petit`. Pour ce faire, se reporter à l’annexe de ce sujet. Décrivez le GDRN correspondant.
- Appliquez l’algorithme d’Allen et Kennedy sur ce code.
- Montrez qu’en introduisant un tableau auxiliaire (sans l’aide de `petit`), il est possible de transformer le code de cette boucle de sorte que la distribution de boucle produise des boucles parallèles.
- Refaire cette distribution de boucle sous `petit` (autre solution possible qu’à la question précédente - cf. annexe)

2 Parallélisations simples

On souhaite calculer:

- (1) Les entrées $B(i, j)$ ($1 \leq i \leq N$, $1 \leq j \leq N + 1$) du triangle de Pascal. En supposant que l’on initialise le tableau B à 0 sauf $B(1, 1) = 1$, cela revient à calculer:

$$B(i, j) = B(i - 1, j) + B(i - 1, j - 1)$$

pour i de 1 à N et j de 1 à i

Question a Quel est le graphe de dépendance de (1)?

Question b Comment paralléliser (1)?

3 Parallélisation de l'algorithme de Cholesky

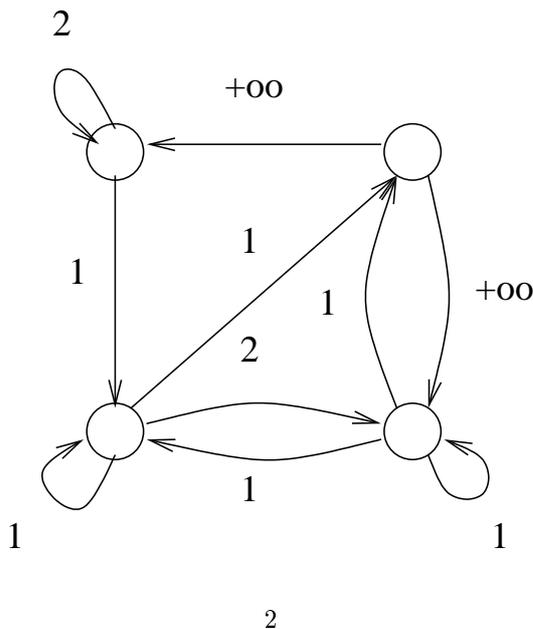
Soit A une matrice symétrique définie positive. L'algorithme de Cholesky permet de résoudre le système linéaire $Ax=b$ en décomposant A en le produit d'une matrice M avec M transposée. Ci-dessous, une portion de code implémentant cette décomposition en langage `petit`:

```
real a(100, 100)
real b(100)
integer n
for k=1, n do
  a(k,k)=sqrt(a(k,k))
  for i=k+1, n do
    a(i,k) = a(i,k)/a(k,k)
    for j=k+1, i do
      a(i,j) = a(i,j)-a(i,k)*a(j,k)
    endfor
  endfor
endfor
```

- (1) Déterminer le graphe de dépendance du code donné plus haut. Une des boucles `for` est-elle parallélisable telle quelle?
- (2) Essayez de casser les cycles de dépendance impliquant le noeud 5 en appliquant une transformation simple.
- (3) Opérer un échange de boucle entre les noeuds 9 et 10 du programme obtenu à la question précédente (c'est-à-dire les boucle `for i` et `for j` imbriquées). Qu'obtient-on en termes de dépendances?
- (4) Inverser l'ordre de parcours de la boucle `for j` dans le programme précédent (noeud 9 a priori). Qu'obtient-on en termes de dépendances?
- (5) Quelles sont les boucles parallélisables? Ecrire le code `petit` correspondant.

4 Dépendances de données

Donner un nid de boucle parfait où toutes les dépendances sont uniformes, et dont le GDRN est exactement le graphe suivant:



Exécuter l'algorithme d'Allen et Kennedy sur ce nid de boucles.

A Utilisation de `petit`

- Pour le calcul de dépendances “simples”:

lancer `petit` sur le code `ex1.t` pris sur la page web du cours (pour le premier exercice):

```
petit ex1.t &
```

(après avoir paramétré correctement son `PATH` comme expliqué en début de séance). Faites ensuite `System`, `DDalg` puis `eps` pour sélectionner un calcul de dépendances simple. Pressez `Xcape` pour revenir au menu du niveau supérieur (mais pas `Quit` cela quitte l'application). Faites ensuite, au menu de niveau supérieur `CalcDD` pour calculer les dépendances, puis `Graph` pour faire afficher le graphe. Les arcs de couleur bleue sont des dépendances anti ou de sortie, les arcs de couleur rouge sont des dépendances de flot. Les annotations entre parenthèses sont les mêmes que celles que l'on trouve dans le cours pour les `GDRV`.

- Pour avoir plus d'informations sur les dépendances:

Changer le mode de dépendances ou faisant `System`, `DDalg` puis `Omega` (puis deux fois `Xcape`). Refaire `CalcDD`. Faire ensuite, dans le menu de plus haut niveau: `Browse` puis `DD` pour parcourir textuellement les dépendances. Pour les voir apparaître dans une fenêtre séparée, faire `Togg` sous le même menu. Vous pouvez sous ce même menu parcourir l'ensemble des dépendances ainsi trouvées par `Cycl`. Vous les verrez ainsi défiler dans la fenêtre correspondante, avec les informations sur les vecteurs d'itération concernés. L'attribut `[0]` (éventuellement combiné à d'autres) indique que l'arc de dépendances sur lequel vous êtes est dans un cycle de dépendances.

- Pour faire introduire un tableau intermédiaire:

Vous devez être en mode de calcul de dépendances `eps` (voir plus haut). Ensuite, vous devez être dans le menu de parcours de dépendances `Browse`, `DD`. Une fois que vous êtes sur l'une des dépendances anti ou de sortie que vous voulez casser, vous pouvez essayer de faire en sorte que `petit` introduise un tableau intermédiaire pour ce faire, par `Expd`.