# Examen – Algorithmes et architectures parallèles

Yves Robert

Lundi 7 Janvier 2002, 9h-12h

Les sept exercices sont indépendants.

## 1 Réseaux d'interconnexion

**Amusette** Quelle différence voyez-vous entre un tore 3D de taille $4 \times 4 \times 4$ et un hypercube de dimension 6 ?

**Produit scalaire** On veut calculer le produit scalaire de deux vecteurs de taille $n$ distribués sur un anneau orienté de $p$ processeurs. A la fin tous les processeurs doivent connaître le résultat. Ecrire le pseudo-programme correspondant.

**Plongement** Peut-on plonger un arbre binaire complet à $2^n - 1$ sommets dans une grille 2D de taille $r \times r$ à déterminer ?

## 2 Algorithmique : étiquetage en composantes

On considère une image binaire de taille $n \times n$ stockée sur une grille 2D de processeurs. On suppose d'abord que cette grille est de taille $n \times n$, donc chaque processeur stocke un seul pixel 0 (point du fond) ou 1 (point de l'image).

L'étiquetage en composantes consiste à étiqueter tous les pixels d'une composante connexe avec le même numéro (et des numéros distincts pour des composantes distinctes). On utilisera le voisinage des 4 voisins cardinaux pour définir l'adjacence de deux pixels.

1. Proposer un algorithme simple pour réaliser cet étiquetage et donner sa complexité au pire
2. Donner un exemple d'image où votre algorithme atteint cette complexité
3. Comment procéder pour une grille de taille $p \times p$, où $p < n$ ?

## 3 Pipeline

On dispose de deux additionneurs pipeline (identiques) à 4 étages. Le temps de cycle de chaque étage est d'une unité de temps. On dispose aussi d'un nombre illimité d'éléments de stockage d'une unité de temps.

1. Soit $(a_i)_{i \geq 1}$ une suite de nombres en entrée. On veut réaliser le calcul de la suite $(b_i)_{i \geq 4}$ définie par $b_i = a_i + a_{i-1} + a_{i-2} + a_{i-3}$. Donner le schéma fonctionnel du pipeline permettant de réaliser ce calcul.
2. On dispose d'un troisième additionneur identique. Combiner celui-ci avec le pipeline précédent pour réaliser le calcul de la suite récurrente $(x_i)_{i \geq 1}$ définie par $x_i = a_i + x_{i-1}$ (on pose $x_0 = 0$).

# 4 Ordonnancement

On considère le modèle avec coût de communication du cours. Pour le graphe de tâches de la Figure 1, les poids des tâches sont indiqués entre parenthèses, et ceux des arêtes le long de celles-ci.

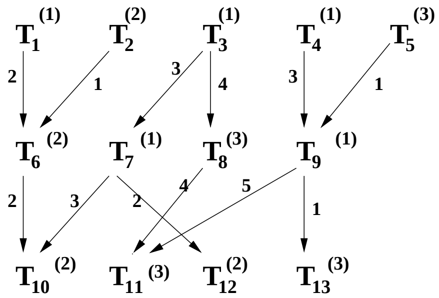


Fig. 1 – Graphe de tâches à ordonnancer.

1. On suppose disposer d'un nombre illimité de processeurs. Ordonnancer le graphe avec l'heuristique du chemin critique modifié.
2. On suppose disposer d'un nombre illimité de processeurs. Ordonnancer le graphe avec l'heuristique de Sarkar.
3. On suppose disposer de seulement $p = 3$ processeurs. Quel est l'ordonnancement optimal ?

# 5 Analyse de dépendances

On considère le nid de boucles suivant :

$$
\begin{array}{l}
\textsf{for } i = 1 \textsf{ to } N \textsf{ do} \\
\textsf{for } j = 1 \textsf{ to } N \textsf{ do} \\
\quad S_1 : a(i+1, j-2) = b(i, j-6) + 1 \\
\quad S_2 : b(i, j-2) = c(i+2, j+5) - 1 \\
\quad S_3 : a(i, j) = b(i+1, j-1) \\
\quad S_4 : c(i, j-1) = a(i, j-1) + 1
\end{array}
$$

1. Donner le graphe de dépendance réduit, en identifiant les flow, les anti et les output dépendances ; pour chaque dépendance, donner à la fois le niveau de dépendance et le vecteur de distance.
2. Déterminer le meilleur ordonnancement linéaire $\pi = (a, b)$ pour ce nid de boucles.

# 6 Kennedy-Allen

1. Construire un nid de boucles parfaitement imbriquées dont le graphe de dépendances réduit (avec les niveaux de dépendance) est exactement le graphe représenté à la Figure 2.
2. Exécuter l'algorithme d'Allen et Kennedy sur ce nid de boucles.

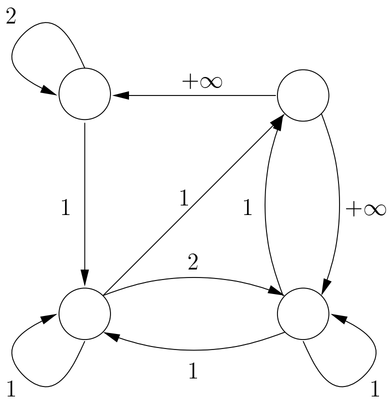


FIG. 2 – Exemple de graphe réduit

# 7 Recherche d'ordonnancement

On considère le nid de boucles suivant :

for i := 1 to N do
for j = 1 to N do
$S_1 : a(i+1, j+1) = a(i+1, j) + b(i, j+2)$
$S_2 : b(i+1, j) = a(i+1, j-1) + b(i, j-1)$
$S_3 : a(i, j+2) = b(i+1, j+1) - 1$

1. Donner le graphe de dépendance réduit associé à ce nid de boucles en précisant le type des dépendances (dépendances en sortie, dépendances de flot ou anti-dépendances), leur expression (vecteur de dépendance) et leur niveau.
2. Appliquer la méthode de Kennedy et Allen pour restructurer le nid de boucles. Quel est la latence du nid obtenu ?
3. Pour le nid de boucles initial, trouver l'ordonnancement linéaire de latence minimale. Donner un équivalent (en fonction de $N$) de la longueur du plus long chemin de dépendance. Exhiber un tel chemin.
4. Le graphe de dépendance réduit est ici fortement connexe. Cependant, cette propriété peut être levée si l'on supprime l'une des anti-dépendances.
   (a) Expliquer comment on peut, en introduisant un nouveau tableau $c$, transformer le nid de boucles en un nid de boucles équivalent pour lequel le graphe de dépendance réduit a deux composantes fortement connexes, l'une étant réduite à l'instruction 3.
   Remarque : ne pas oublier l'initialisation du tableau $c$ et la réécriture terminale dans le tableau initial.
   (b) Montrer alors que l'on peut scinder ce nouveau nid en deux nids, le premier itérant sur les instructions 1 et 2, le second étant complètement parallèle et itérant sur l'instruction 3.
   (c) Donner, pour le nid itérant sur les instructions 1 et 2, l'expression de l'ordonnancement linéaire de latence minimale. Donner un équivalent (en fonction de $N$) de la longueur du plus long chemin de dépendance. Exhiber un tel chemin.