Ordonnancement pour les plates-formes hétérogènes

1 Équilibrage un-dimensionnel

On a B tâches atomiques identiques à exécuter sur une plate-forme comportant p processeurs hétérogènes. L'exécution d'une tâche sur le processeurs P_i prend un temps t_i. On considère que l'exécution d'une tâche sur un processeur ne nécessite pas de communication préalable.

▷ **Question 1** *On cherche un ordonnancement des B tâches sur les p processeurs qui minimise le temps d'exécution total. Donner un algorithme calculant le nombre de tâches c_i à allouer au processeur P_i, et montrer son optimalité.*

▷ **Question 2** *On suppose maintenant qu'on ne connaît pas initialement le nombre total de tâches B à effectuer. Donner un algorithme incrémental qui calcule un ordonnancement optimal pour tout ensemble de tâches $[1, m]$ ($1 \leq m \leq B$).*

On reprend l'application de balayage d'image étudiée sur un anneau. On cherche à paralléliser un calcul comme celui représenté sur la figure 1, où le domaine de calcul est un rectangle, et les dépendances sont données par le vecteur :

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

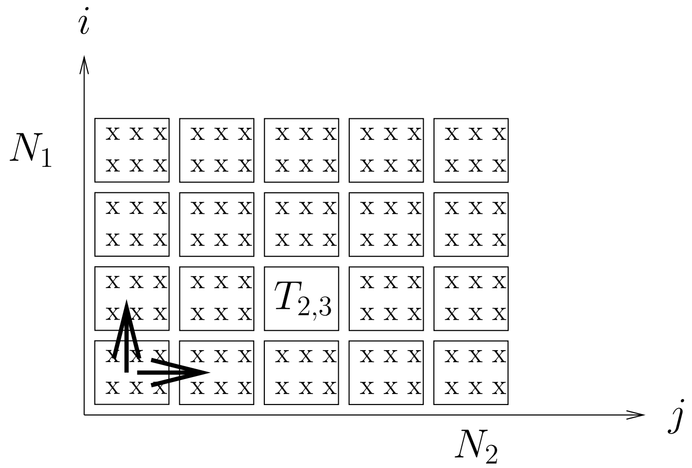


Fig. 1 – Domaine de calcul partitionné en tuiles

On cherche à allouer des colonnes de tuiles à p processeurs hétérogènes : le processeur P_i traite une tuile en temps t_i.

▷ **Question 3** *Comment se comporte une stratégie dynamique gloutonne (qui alloue la prochaine colonne au prochain processeur libre) ? Traiter un exemple avec trois processeurs tels que $t_1 = 3$, $t_2 = 5$, $t_3 = 8$.*

▷ **Question 4** *Décrire comment adapter l'allocation cyclique utilisée pour l'anneau homogène en s'inspirant de l'équilibrage de charge trouvé précédemment.*

2 Équilibrage à deux dimensions

On souhaite maintenant étudier une distribution 2D des données, par exemple pour effectuer un produit de matrices, comme étudié dans un TP MPI : la charge d'une machine correspond à l'aire de la zone qui lui est attribuée.

On adapte le produit de matrices sur une grille homogène au cas hétérogène comme sur la figure 2(a) : la grille est composée de $p \times q$ processeurs, et un rectangle de taille $h_i \times v_j$ est assigné au processeur $P_{i,j}$.

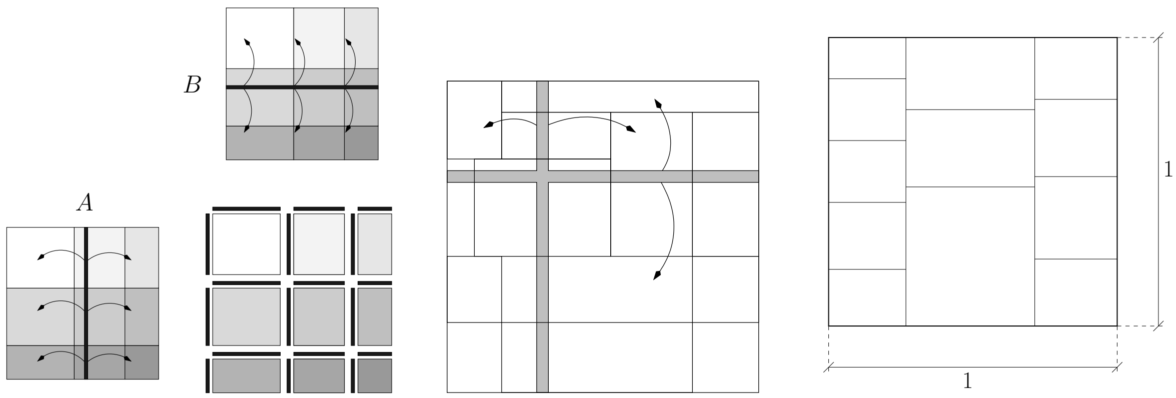
L.Marchal , C. Tedeschi , Y.Robert

▷ **Question 5** *On considère quatre processeurs sur une grille 2×2, avec les temps de cycle $t_{1,1} = 1$, $t_{1,2} = 2$, $t_{2,1} = 3$ et $t_{2,2} = 6$; comment équilibrer la charge de ces machines sur une grille 2D ? Même question avec $t_{2,2} = 5$. En déduire une condition sur les temps de cycle des processeurs pour qu'il existe un équilibrage parfait.*

▷ **Question 6** *Dans le cas général, exprimer le problème d'optimisation correspondant à l'équilibrage de charge sur une grille 2D de processeurs.*

En fait, il n'y a pas de raison de limiter la généralisation au cas de la grille. On pourrait par exemple partitionner la charge comme représenté sur la figure 2(b). Dans ce cas, il y a une solution évidente, consistant à diviser le domaine par bandes, de surfaces proportionnelles à la vitesse des processeurs… on retombe sur l'équilibrage 1D. On va voir que celui-ci n'est pas satisfaisant pour le produit de matrices à cause des communications.

▷ **Question 7** *Pour une distribution 2D donnée, quel est le coût des communications, dans le cas de communications séquentielles ? dans le cas de communications parallèles ? Pour le cas des communications séquentielles, exprimer le problème d'optimisation correspondant à un partitionnement libre.*



(a) Produit de matrices sur une grille hétérogène 3×3

(b) Partitionnement libre pour le produit de matrices

(c) Partitionnement par colonnes

FIG. 2 – Différentes solutions pour l'équilibrage 2D.

Le problème de décision associé à ce problème d'optimisation est NP-complet, et la preuve de ce résultat est assez technique. On va s'intéresser à une relaxation de ce problème, en organisant les processeurs par colonnes comme indiqué sur la figure 2(c).

La vitesse du processeur P_i est s_i, et $\sum s_i = 1$; on cherche à partitionner le carré unité en $\mathcal{C}$ colonnes de largeur $c_1, \ldots, c_{\mathcal{C}}$. Chaque colonne C_i est elle-même partitionnée en k_i lignes. Il y a bien sûr $\sum k_i = p$ rectangles au total et on cherche à minimiser la somme des demi-périmètres (cas des communications séquentielles).

On suppose que dans la solution optimale, tous les s_i de la première colonne sont inférieurs à tous les s_i de la deuxième colonne, etc… On se restreint donc à une solution où $s_1 \leq s_2 \leq \cdots \leq s_p$ et $s_1, \ldots, s_{k_1}$ sont affectés à la première colonne, $s_{k_1+1}, \ldots, s_{k_1+k_2}$ sont affectés à la deuxième colonne, etc…

Pour $q \in \{1, \ldots, p\}$, on appelle $f_{\mathcal{C}}^{\text{périmètre}}(q)$ le périmètre total du partitionnement optimal du rectangle de hauteur 1 et de largeur $(\sum_{i=1}^{q} s_i)$ en $\mathcal{C}$ colonnes et q rectangles d'aires respectives $s_1, \ldots, s_q$.

▷ **Question 8** *Donner une formule de récurrence pour calculer $f_{\mathcal{C}}^{\text{périmètre}}(q)$. Donner un algorithme calculant tous les $f_{\mathcal{C}}^{\text{périmètre}}(q)$, pour $q = 1 \ldots p$ et $\mathcal{C} = 1 \ldots q$. Comment déduire un partitionnement en colonnes optimal à partir de ces valeurs ?*

▷ **Question 9** *Montrer que l'hypothèse qu'on a utilisée (les processeurs sont alloués dans l'ordre $s_1 \leq s_2 \leq \cdots \leq s_p$) est justifiée.*