

Réseaux d'interconnexion—2

1 Réseaux reconfigurables

Un réseau « butterfly » de dimension r , noté $BUT(r)$, est un réseau composé de $(r+1)2^r$ nœuds organisés en 2^r lignes de $r+1$ niveaux. Un nœud est désigné par une paire (w, i) où w est un entier codé en représentation binaire sur r bits qui numérote la ligne du nœud, et où i numérote le niveau du nœud ($0 \leq i \leq r$). Deux nœuds (w, i) et (w', i') sont reliés par un arc si et seulement si $i' = i + 1$ et l'une des deux conditions suivantes est remplie :

- $w = w'$
- w et w' ne diffèrent que par le i -ème bit.

La figure 1, représente $BUT(3)$.

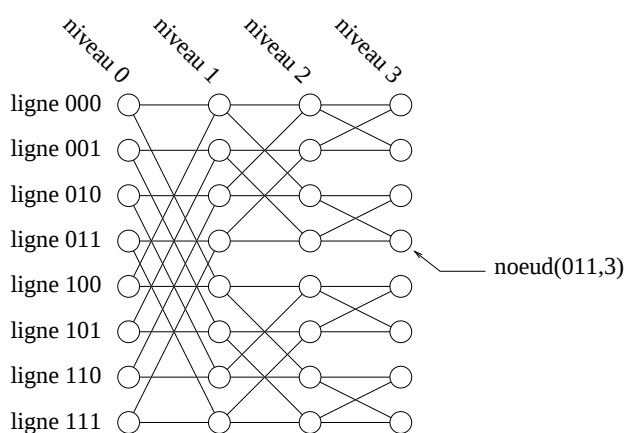


FIG. 1 – $BUT(3)$, le réseau butterfly de dimension 3.

▷ **Question 1** Quelle est l'architecture du réseau obtenu lorsque l'on regroupe les nœuds d'une même ligne en un seul nœud (et que l'on enlève les arcs redondants) ?

▷ **Question 2** Le réseau butterfly a une structure récursive. Donner deux moyens d'obtenir deux réseaux butterfly de dimension $r - 1$ à partir d'un réseau butterfly de dimension r .

▷ **Question 3** Montrer qu'il existe un unique chemin de longueur r entre un nœud $(w, 0)$ du niveau 0 et un nœud (w', r) du niveau r . Quel est le diamètre de $BUT(r)$?

Un réseau de Benes de dimension r est composé de deux réseaux butterfly mis « dos à dos ». Les nœuds des niveaux r de chaque butterfly sont fusionnés : chaque niveau comprend alors $2r + 1$ nœuds sur un même niveau. La figure 2 montre un réseau de Benes de dimension 3. On supposera ici que les nœuds du réseau sont uniquement des commutateurs 2×2 , pouvant être configurés en croix ou en lignes parallèles pour transmettre les données.

▷ **Question 4** Montrer (par récurrence sur la dimension r du réseau) que le réseau de Benes peut être configuré pour réaliser une permutation arbitraire : étant donné une permutation quelconque π des 2^{r+1} premiers entiers (de 0 à $2^{r+1} - 1$), il existe une configuration des commutateurs qui permet de connecter simultanément l'entrée i du réseau à sa sortie $\pi(i)$. Par exemple, figure 4 est représentée une configuration des commutateurs d'un réseau de Benes de dimension 1 réalisant la permutation $\pi = (0, 1, 2, 3) \rightarrow (3, 1, 2, 0)$.

2 Réduction sur un Nanneau

On dispose de p fichiers F_i distribués sur un anneau unidirectionnel de p processeurs : P_i possède le fichier F_i , pour $1 \leq i \leq p$ (on pourra par exemple assimiler ces fichiers à des matrices). On dispose d'une loi associative et a priori non commutative sur ces fichiers, notée $\odot$ (ce pourra être l'addition ou la multiplication de matrices). On souhaite calculer la réduction $F_1 \odot F_2 \odot \dots \odot F_p$, et le résultat pourra se

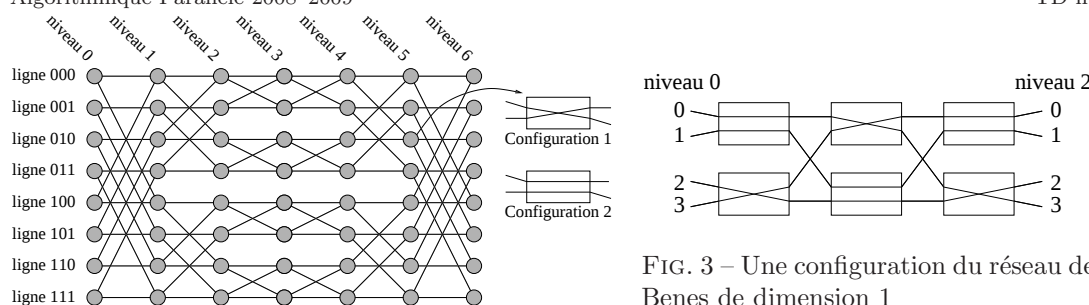


FIG. 2 – Le réseau de Benes de dimension 3

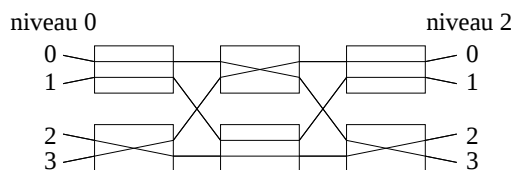


FIG. 4 – Configuration du réseau de Benes de dimension 1.

trouver sur n'importe quel processeur de l'anneau. Le coût de communication d'un fichier sur un lien de l'anneau est c . Le coût de calcul d'une opération $\odot$ est w .

▷ **Question 5** Dans un premier temps, on suppose que $w \ll c$ (c'est le cas par exemple pour l'addition de matrices). Donner un algorithme réalisant l'opération de réduction sur l'anneau, et calculer sa complexité.

▷ **Question 6** On s'intéresse maintenant à une opération $\odot$ telle que $c \ll w$ (c'est le cas par exemple pour la multiplication de matrices). Proposez un algorithme de réduction adapté à ce cas, et donner sa complexité.

3 Retours aux Nanneaux

3.1 Résolution d'un système triangulaire

On s'intéresse à la résolution du système linéaire $Ax = b$, où A est une matrice triangulaire inférieure d'ordre n et b un vecteur à n composantes. On suppose disposer d'un anneau orienté de p processeurs $P_0, P_1, \dots, P_{p-1}$. On prendra n divisible par p .

▷ **Question 7** On veut distribuer les colonnes de A aux processeurs. Quelle stratégie adopter ? Donner le schéma d'un algorithme de résolution parallèle adapté à la distribution choisie.

▷ **Question 8** On veut maintenant distribuer les lignes de A aux processeurs. Quelle stratégie adopter ? Donner le schéma d'un algorithme de résolution parallèle adapté à la distribution choisie.

À l'étape i , le processeur responsable de $x[i]$ calcule sa valeur définitive, puis la diffuse aux autres processeurs, qui mettent à jour leurs données. Le lecteur attentif aura remarqué la parfaite similitude entre cet algorithme et celui de la décomposition LU. Toutes les variantes proposées pour cette dernière peuvent être utilisées : recouvrement calcul-communication, pipeline de la diffusion, jusqu'à la technique du *look-ahead* !