

Réseaux d'interconnexion – 2

1 REM

Un Réseau Échange-Mélange (REM) avec $p = 2^r$ processeurs est défini comme suit :

(i) numéroté les processeurs de 0 à $p - 1$ et les écrire en binaire sur r bits :

$$q = b_{r-1}b_{r-2} \dots b_2b_1b_0 \text{ avec } b_i \in \{0, 1\} \text{ pour } 0 \leq i \leq r - 1$$

(ii) pour $q = b_{r-1}b_{r-2} \dots b_2b_1b_0$, définir :

$$\begin{aligned} rot(q) &= b_0b_{r-1}b_{r-2} \dots b_2b_1 \\ exch(q) &= b_{r-1}b_{r-2} \dots b_2b_1(1 - b_0) \end{aligned}$$

(iii) pour tout q , $0 \leq q \leq p - 1$, relier q à $rot(q)$ et à $exch(q)$ par des arcs orientés.

▷ **Question 1** Dessiner un REM à 2^4 processeurs (en regroupant les nœuds par paires, puis par motif).

▷ **Question 2** Proposer un algorithme de routage d'un processeur à un autre dans un REM. Quel est le diamètre d'un REM à 2^r processeurs ?

2 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 $BUT(3)$.

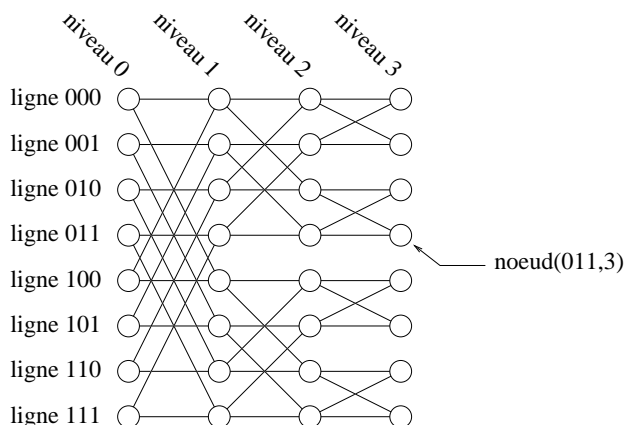


FIG. 1 – $BUT(3)$, le réseau butterfly de dimension 3.

▷ **Question 3** 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) ?

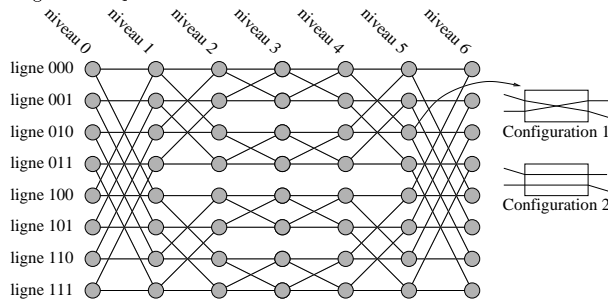


FIG. 2 – Le réseau de Benes de dimension 3

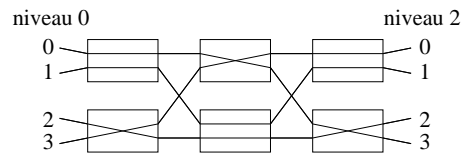


FIG. 3 – Une configuration du réseau de Benes de dimension 1

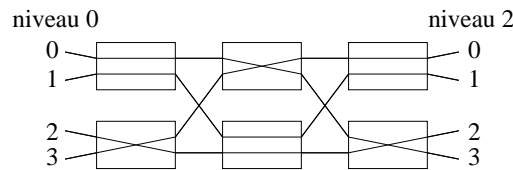


FIG. 4 – Configuration du réseau de Benes de dimension 1.

▷ **Question 4** 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 5** 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 6** 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 $\pi = (0, 1, 2, 3) \rightarrow (3, 1, 2, 0)$.

3 Réduction sur un anneau

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 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 7** 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.

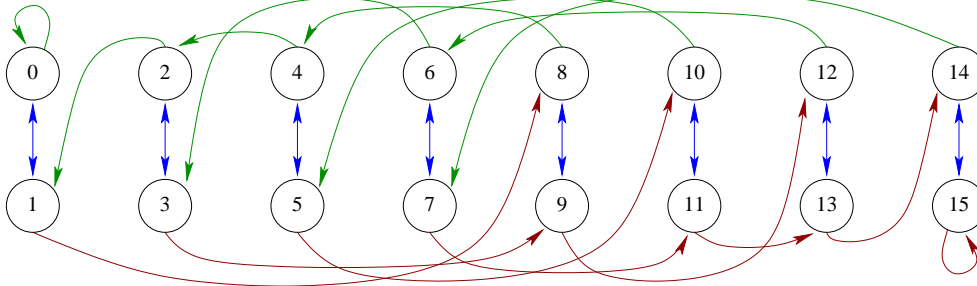
▷ **Question 8** On s’intéresse maintenant à une opération \odot telle que $c \ll w$ (c’est le cas par exemple pour la multiplication de matrices). Proposez un algorithme de réduction adapté à ce cas, et donner sa complexité.

4 Réponses aux exercices

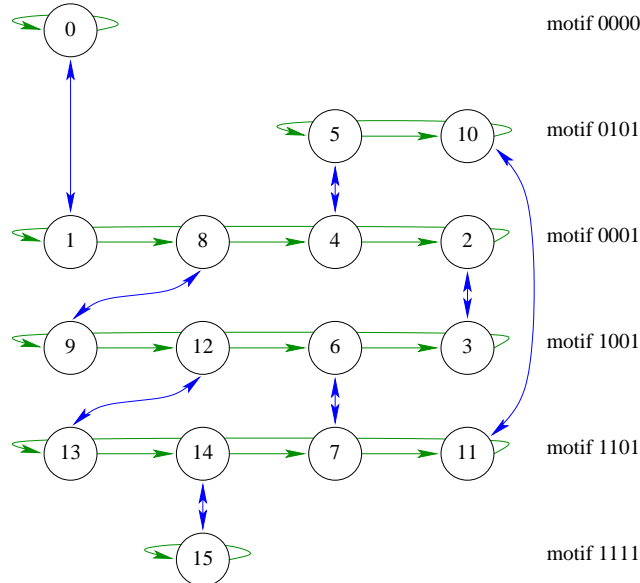
▷ Question 1, page 1

Voilà à quoi peut ressembler un REM avec 16 processeurs :

- **En regroupant les nœuds par paires.** Les liens verticaux correspondent aux opérations *rot*.



- **En regroupant les nœuds par motif.** Les liens horizontaux (qui forment des anneaux unidirectionnels) correspondent aux opérations *exch*.



▷ Question 2, page 1

Supposons qu'on veuille passer du nœud a au nœud b , on commence par calculer $c = a \text{ XOR } b$, puis on fait, pour i allant de 0 à $r - 1$:

- si $c_i = 0$, faire ROT ;
- si $c_i = 1$, faire EXCH puis ROT.

Autre vision : si on remplace, dans c les 0 par des "ROT-" et les 1 par des "ROT-EXCH-", puis qu'on retourne la séquence par une opération "miroir", la séquence obtenue représente les opérations à faire pour router de a à b . Exemple : pour router de 0 à 11 (=1101 en binaire) dans l'exemple de plateforme ci-dessus, on suit la séquence EXCH-ROT-ROT-EXCH-ROT-EXCH-ROT-EXCH-ROT, qui nous fait passer par les nœuds 0-1-8-4-5-10-11-13.

Notons que cet algorithme n'est pas optimal.

▷ Question 3, page 1

On obtient un hypercube de dimension r . En effet, un nœud w est connecté au nœud w' si et seulement si leur représentation binaires diffèrent d'un seul bit. En conséquence, on pourra simuler chaque pas d'un algorithme s'exécutant sur un hypercube de dimension r en au plus r pas sur le réseau butterfly.

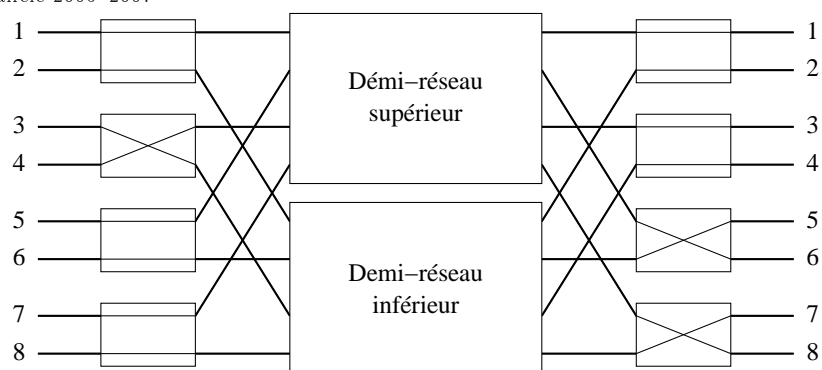


FIG. 5 – Partage du réseau de Benes de dimension r .

▷ **Question 4, page 1**

Si on supprime tous les nœuds du niveau 0, on obtient un premier réseau composé des nœuds des 2^{r-1} premières lignes, et un deuxième réseau composé des nœuds des 2^{r-1} dernières lignes.

Si on supprime tous les nœuds du niveau r , c'est un peu plus difficile à voir : le premier réseau est composé des nœuds des lignes de numéro pair, et le deuxième réseau est composé des nœuds des lignes de numéro impair. En effet, supprimer le dernier niveau équivaut à supprimer le dernier bit des numéros de ligne.

▷ **Question 5, page 1**

Au niveau i , on choisit le lien horizontal (sur la même ligne) si les représentation binaires de w et w' ne diffèrent pas sur le i -ème bit, et l'autre lien sinon.

Le plus long chemin est de longueur $2r$, par exemple du nœud $(0, 0)$ au nœud $(2^r - 1, 0)$.

▷ **Question 6, page 2**

Si $r = 1$, le réseau est constitué d'un seul commutateur, et le résultat est évident. Supposons le résultat vrai pour un réseau de Benes de dimension $r - 1$.

L'observation fondamentale pour la preuve est la suivante : les niveaux 1 à $2r - 1$ d'un réseau de Benes de dimension r contiennent deux réseaux de Benes de dimension $r - 1$, comme indiqué figure 5.

Pour réaliser la permutation π , on va acheminer la moitié des messages *via* le demi-réseau supérieur, et l'autre moitié *via* le demi-réseau inférieur. La contrainte est que tous les 2^r chemins doivent être à arêtes disjointes. Mais elle est facile à respecter. On commence par acheminer le premier message, de l'entrée $x_1 = 1$ à la sortie $y_1 = \pi(1)$, *via* le demi-réseau supérieur. La sortie y_1 partage un commutateur avec une autre sortie, soit y_2 celle-ci. Soit x_2 l'antécédent de y_2 ($\pi(x_2) = y_2$) : on achemine le message de x_2 à y_2 *via* le demi-réseau inférieur. Soit alors x_3 l'entrée qui partage le commutateur avec x_2 et $y_3 = \pi(x_3)$, on achemine ce message *via* le demi-réseau supérieur. On continue ainsi jusqu'à boucler, en voulant acheminer un message à travers le demi-réseau inférieur, en réponse à une contrainte sur une sortie, mais ce message partage une entrée avec le premier message acheminé (ce qui ne viole pas de contrainte, le premier message ayant été acheminé *via* le demi-réseau supérieur). On recommence la procédure avec un message arbitraire n'ayant pas encore été routé. À la fin, la moitié des chemins auront été acheminés *via* le premier demi-réseau, et l'autre moitié *via* le deuxième demi-réseau. L'hypothèse de récurrence permet de conclure pour l'acheminement complet *à l'intérieur* des demi-réseaux.

▷ **Question 7, page 2**

Code du processeur P_i , possédant F_i :

1. Si $i \neq 0$, recevoir A du processeur précédent (P_{i-1})
2. $A \leftarrow A + F_i$
3. Si $i \leq n - 1$, envoyer A au suivant (P_{i+1})

Complexité : $p \times (w + c)$

▷ **Question 8, page 2**

Dans le cas $c \ll w$, de façon similaire au SCAN de base par saut de pointeurs, sauf que contrairement aux PRAM, le transfert d'un fichier entre deux processeurs P_i et P_{i+d} coûte $d \times c$. Complexité : $p \times c + \log(p) \times w$.