

Anneaux de processeurs et P-RAM

<http://graal.ens-lyon.fr/~lmarchal/algopar.html>

1 Anneaux de processeurs

Pour triangulariser une matrice A d'ordre n de façon numériquement stable, on peut utiliser les rotations de Givens. L'opération de base $\text{ROT}(i, j, k)$ consiste à combiner les deux lignes i et j , qui doivent toutes deux commencer par $k - 1$ zéros, pour annuler l'élément en position (j, k) :

$$\begin{pmatrix} 0 & \dots & 0 & \mathbf{a}'_{i,k} & a'_{i,k+1} & \dots & a'_{i,n} \\ 0 & \dots & 0 & \mathbf{0} & a'_{j,k+1} & \dots & a'_{j,n} \end{pmatrix} \leftarrow \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} 0 & \dots & 0 & \mathbf{a}_{i,k} & a_{i,k+1} & \dots & a_{i,n} \\ 0 & \dots & 0 & \mathbf{a}_{j,k} & a_{j,k+1} & \dots & a_{j,n} \end{pmatrix}$$

L'algorithme séquentiel peut s'écrire :

```
GIVENS(A)
1: Pour  $k = 1 \dots n - 1$  :
2:   Pour  $i = n \dots k + 1$  (de -1 en -1) :
      ROT( $i - 1, i, k$ )
```

On considère qu'une rotation $\text{ROT}(i, j, k)$ s'exécute en temps unité, indépendamment de k .

▷ **Question 1** Mettre en œuvre cet algorithme sur un anneau de n processeurs.

▷ **Question 2** Comment pourrait-on adapter cet algorithme sur un anneau bidirectionnel comportant seulement $\lfloor \frac{n}{2} \rfloor$ processeurs.

2 P-RAM

2.1 Sélection dans une liste

▷ **Question 3** Soit L une liste contenant n objets coloriés soit en bleu, soit en rouge. Concevoir un algorithme EREW efficace qui sépare les éléments bleus des éléments rouges (c'est-à-dire qui construit une nouvelle liste ne contenant que les éléments bleus).

2.2 Procédure mystère

On définit les deux opérateurs suivants pour un tableau $A = [a_0, a_1, \dots, a_{n-1}]$ de n entiers :

- $\text{PRESCAN}(A)$ renvoie le tableau $[0, a_0, a_0 + a_1, a_0 + a_1 + a_2, \dots, a_0 + a_1 + \dots + a_{n-2}]$
- $\text{SCAN}(A)$ renvoie le tableau $[a_0, a_0 + a_1, a_0 + a_1 + a_2, \dots, a_0 + a_1 + \dots + a_{n-1}]$

Nous avons vu en cours comment réaliser ces deux opérateurs en temps $O(\log n)$ sur une P-RAM EREW.

▷ **Question 4** Étant donné un tableau de booléens $Flags$, que fait la procédure suivante ?

```
SPLIT(A, Flags)
1:  $Idown \leftarrow \text{PRESCAN}(\text{not}(Flags))$ 
2:  $Iup \leftarrow n - \text{REVERSE}(\text{SCAN}(\text{REVERSE}(Flags)))$ 
3: Pour  $i = 1$  à  $n$  en parallèle :
4:   Si  $Flags(i)$  Alors
5:      $Index[i] \leftarrow Iup[i]$ 
6:   Sinon
7:      $Index[i] \leftarrow Idown[i]$ 
8:  $Result \leftarrow \text{PERMUTE}(A, Index)$ 
9: Renvoyer  $Result$ 
```

Les noms des différentes fonctions sont relativement intuitifs ; en particulier, REVERSE renverse le tableau, et $\text{PERMUTE}(A, Index)$ réordonne le tableau A selon la permutation $Index$. L'horrible expression $\text{REVERSE}(\text{SCAN}(\text{REVERSE}(Flags)))$ effectue simplement un SCAN à partir de la fin du tableau $Flags$, dont les éléments sont considérés comme des entiers.

Voici un exemple d'utilisation :

$$\begin{array}{rcl}
 A & = & [5 \quad 7 \quad 3 \quad 1 \quad 4 \quad 2 \quad 7 \quad 2] \\
 Flags & = & [1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0] \\
 Idown & = & [0 \quad 0 \quad 0 \quad 0 \quad \boxed{0} \quad \boxed{1} \quad 2 \quad \boxed{2}] \\
 Iup & = & [\boxed{3} \quad \boxed{4} \quad \boxed{5} \quad \boxed{6} \quad 7 \quad 7 \quad \boxed{7} \quad 8] \\
 Index & = & [3 \quad 4 \quad 5 \quad 6 \quad 0 \quad 1 \quad 7 \quad 2] \\
 Result & = & [4 \quad 2 \quad 2 \quad 5 \quad 7 \quad 3 \quad 1 \quad 7]
 \end{array}$$

Quel est le coût de la fonction SPLIT ?

On considère la procédure MYSTÈRE suivante :

MYSTÈRE(A , $Number_Of_Bits$)
 1: **Pour** $i = 0$ à $Number_Of_bits - 1$:
 2: $bit(i) \leftarrow$ tableau indiquant si le i -ème bit des éléments de A est à 1
 3: $A \leftarrow$ SPLIT($A, bit(i)$)

- ▷ **Question 5** Faire tourner la procédure sur $A = [5, 7, 3, 1, 4, 2, 7, 2]$ avec $Number_Of_Bits = 3$.
- ▷ **Question 6** Que fait la procédure MYSTÈRE ?
- ▷ **Question 7** Avec des entrées de taille $O(\log n)$ bits, quelle est la complexité avec n processeurs ? Et avec seulement p processeurs ? Quelles sont les valeurs de p les plus intéressantes ?