
ALGORITHMES DE TRI PARALLELES

Daniel Etiemble
LRI, Université Paris Sud
de@lri.fr

Références

- C. Germain, « Introduction au calcul parallèle », polycopié maîtrise informatique, Orsay
- <http://www.eli.sdsu.edu/courses/spring96/cs662/notes>
- http://bitscap.bits-pilani.ac.in/param/public_html

Algorithmes séquentiels

- Tris simples
 - Tri-bulle
 - Tri-insertion
 - Etc
- Tris plus efficaces
 - Tri rapide (Quicksort)
 - Choisir un pivot
 - Liste de tous les éléments inférieurs au pivot
 - Liste de tous les éléments supérieurs au pivot
 - Appels récursifs à Quicksort sur les listes créées
- Algorithmes utiles
 - Fusion de deux listes triées.

Algorithmes parallèles

- Hypothèses
 - La liste de N éléments est distribuée sur p processeurs, avec N/p éléments par processeurs
- Tri pair-impair
- Tri par paquets
- Tri par échantillons
- Autres tris (non traités)
 - Tri bitonique
 - Hyperquicksort

Tri pair-impair

- Algorithme en supposant un élément par processeur
 - Chaque processeur P_J contient x_J pour $J=1,2, \dots, N$
 - Pour $k=1$ à $N/2$
 - Tous les processeurs impairs comparent leur valeur avec la valeur du processeur de droite. Si non ordonné, alors échange
 - Tous les processeurs pairs comparent leur valeur avec la valeur du processeur de droite. Si non ordonné, alors échange

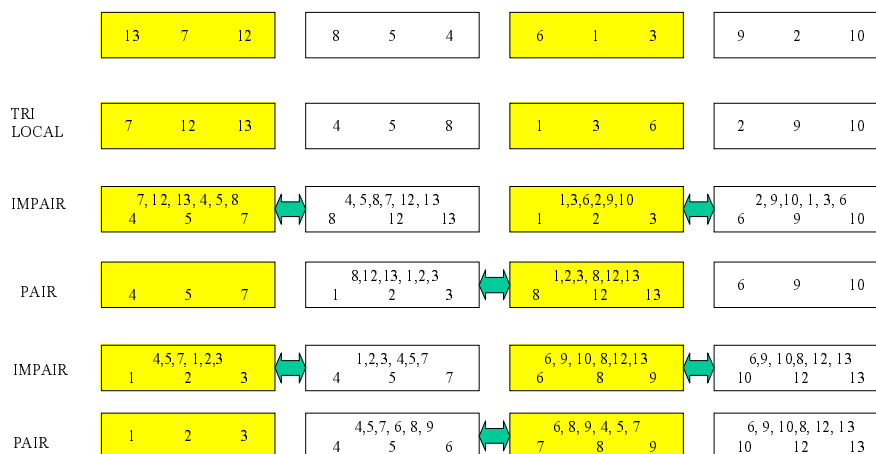
Exemple

	P1	P2	P3	P4	P5	P6	P7	P8
Initial	34	87	65	15	71	45	32	10
Impair	34	87	15	64	45	71	10	32
Pair	34	15	87	45	64	10	71	32
Impair	15	34	45	87	10	64	32	71
Pair	15	34	45	10	87	32	64	71
Impair	15	34	10	45	32	87	64	71
Pair	15	10	34	32	45	64	87	71
Impair	10	15	32	34	45	64	71	87
Pair	10	15	32	34	45	64	71	87

Tri pair impair

- Algorithme avec N/p éléments par processeurs
 - Chaque processeur trie ses N/p éléments (Quicksort)
 - Pour k à $p/2$
 - Pour tous les processeurs impairs et les processeurs de droite
 - Le processeur impair envoie sa liste triée au processeur de droite
 - Le processeur de droite envoie sa liste triée au processeur de gauche
 - Chaque processeur fait une fusion triée de sa liste avec la liste de son voisin
 - Le processeur de gauche conserve la moitié inférieure de la liste fusionnée et le processeur de droite conserve la moitié supérieure de la liste
 - Pour tous les processeurs pairs et les processeurs de droite
 - Le processeur pair envoie sa liste triée au processeur de droite
 - Le processeur de droite envoie sa liste triée au processeur de gauche
 - Chaque processeur fait une fusion triée de sa liste avec la liste de son voisin
 - Le processeur de gauche conserve la moitié inférieure de la liste fusionnée et le processeur de droite conserve la moitié supérieure de la liste

Tri pair-impair : exemple avec 4 processeurs



Tri parallèle par paquets (bucket sort)

- Suppose que les n éléments à trier sont répartis uniformément sur un intervalle (a, b) . Le nombre d'éléments par paquet est approximativement n/m (nombre de paquets)
- Soit n le nombre d'éléments et p le nombre de processeurs
 - Chaque processeur répartit son bloc de n/p éléments en p sous blocs (chaque processeur connaît l'intervalle (a, b) et donc l'intervalle pour chaque paquet)
 - Chaque processeur envoie les sous blocs au processeur adéquat. Après cette phase, chaque processeur a seulement les éléments pour le paquet qui lui est assigné
 - Chaque processeur trie son paquet interne en utilisant un algorithme de tri séquentiel optimal.

Tri parallèle par échantillon (sample sort)

- Amélioration de l'algorithme par paquet
 - Un échantillon de taille s est choisi à partir d'une séquence de n éléments et la taille d'un paquet est choisie en triant l'échantillon et en choisissant $m-1$ éléments du résultat. Ces éléments (splitters) divisent l'échantillon en m paquets de taille égale.
- Algorithme
 - Partitionnement des données d'entrée et tri local
 - Chaque processeur a n/p éléments à trier. Soit A_i la séquence pour le processeur P_i . Chaque processeur trie ses n/p éléments locaux.
 - Choix des séparateurs
 - Cette phase choisit les $p-1$ séparateurs S
 - Chaque processeur P_i sélectionne $p-1$ éléments également espacés de la séquence locale triée A_i . Ces $p-1$ sont envoyés au processeur 0 qui choisit $p-1$ séparateurs en triant ces $p(p-1)$ éléments et diffuse $p-1$ séparateurs aux autres processeurs.
 - Fin du tri
 - Chaque processeur P_i utilise les séparateurs pour partitionner les séquences locales A_i en p sous séquences $A_{i,j}$ telles que pour $0 \leq j < p-1$, tous les éléments dans $A_{i,j}$ sont plus petits que S_j et pour $j=p-1$, $A_{i,j}$ contient les éléments restants. Puis chaque processeur P_i envoie la sous séquence $A_{i,j}$ au processeur j .
 - Finalement, chaque processeur trie les sous séquences reçues en utilisant le tri fusion.