Travaux Dirigés et Pratiques n°4

Algorithmes parallèles et répartis

—Élèves-Ingénieurs de 5-ème année—

Tri en environnement distribué

Cette feuille porte sur l'étude des problèmes de tri d'entiers en parallèle et en distribué. Nous voyons un algorithme de tri dit "pair-impair" et étudions sa portabilité sur machine parallèle à structure d'anneau et machine parallèle virtuelle. Nous envisageons alors effectivement une implémentation d'un algorithme de tri de type "tri rapide" en environnement distribué.

▸ **Exercice 1.** (**parallélisation du tri pair-impair**) *Cet exercice est une étude des possibilités théoriques de parallélisation de l'algorithme de tri "pair-impair". Le tri "pair-impair" est une variante du tri à bulle. Il permet globalement de trier n éléments en n étapes, chaque étape consistant en $n/2$ comparaisons-échanges. L'algorithme alterne entre deux phases, appelées "phase paire" et "phase impaire".*

Soit $(a_1, a_2, \ldots, a_n)$ une suite d'entiers à trier. Durant la phase impaire, les éléments d'indice impair sont comparés à leurs voisins de droite et le cas échéant ils sont échangés pour que chaque sous-suite de deux éléments soit croissante. De la même manière, durant la phase paire, les éléments d'indice pair sont comparés et réordonnés par rapport à leurs voisins de droite. Après n étapes, la suite est triée.

1. *Donner la complexité de l'algorithme séquentiel de tri "pair-impair".*
2. *Proposer une version parallèle de cet algorithme, utilisant $\mathcal{O}(n)$ processeurs organisés en anneau par exemple.*
3. *Calculer la complexité de cet algorithme parallèle.*
4. *Proposer une version distribuée sur $p \ll n$ processeurs de l'algorithme de tri "pair-impair" de n entiers.*
5. *Donner la complexité de cet algorithme en nombre de comparaisons.*
6. *Puis en nombre d'échanges de messages.*

▸ **Exercice 2.** (**vers un tri rapide distribué**) *Cet exercice vise à proposer une première solution au problème de la distribution du "quick-sort" sur un réseau de machines.*

1. *Proposer un algorithme de tri s'appuyant sur une itération de tris rapides.*

2. *Proposer des solutions pour réaliser la fusion de tableaux triés distribués sur la machine virtuelle.*

3. *Implémenter en* `PVM` *un algorithme de tri rapide s'appuyant sur une solution de fusion proposée précédemment.*

▸ **Exercice 3.** **(tri rapide en distribué)** *L'objet de cet exercice est une implémentation effective d'un algorithme de tri en distribué. Nous avons vu précédemment un algorithme qui nécessite une ou plusieurs phases de fusion des résultats triés.*

1. *Est-il possible de complètement faire disparaître la phase de fusion des tableaux triés distribués sur la machine virtuelle ?*

2. *Quelle solution adopter pour effectivement suivre cette voie. Proposer un algorithme complet réalisant un tri rapide distribué en utilisant la technique décrite ci-dessus.*

3. *Quel est la complexité de l'algorithme en nombre de messages ?*

4. *Quel est la complexité de l'algorithme en nombre de comparaisons (en parallèle) ?*

5. *Réaliser effectivement cet algorithme sous* `PVM`.