

Examen du cours Mobilité

M2 Pro, Université Paris Diderot, Paris 7

Michel Habib

March 20, 2011

1 Question de cours

Quelles sont les similarités et les différences entre les réseaux ad-hoc (réseaux spécifiques des radio mobiles) et les réseaux d'overlay (réseaux virtuels des systèmes pair-à-pair). En conséquence peut-on utiliser les mêmes algorithmes de routage pour ces deux types de réseaux?

2 Machines PRAM

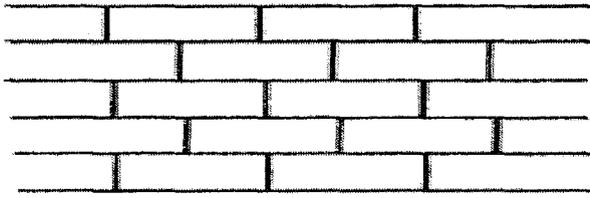
Pour chacun des exercices, on donnera une version séquentielle de l'algorithme et l'on précisera pour l'algorithme parallèle, le nombre de processeurs utilisés ainsi que la complexité en temps de votre algorithme et le travail. On précisera aussi le modèle de machine PRAM qui permet de faire tourner votre algorithme (CRCR, CREW ou encore EREW).

On considère une liste A de n éléments entiers : a_1, \dots, a_n ,

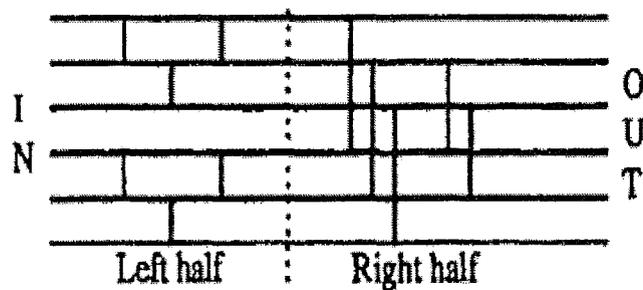
1. Ecrire un algorithme qui calcule en temps $O(\log n)$ sur une machine PRAM, pour chaque $i, 1 \leq i \leq n$ l'élément minimum de l'ensemble $\{a_i, \dots, a_n\}$.
2. Ecrire un algorithme qui extrait deux sous-listes de A , l'un contenant les entiers pairs, l'autre les impairs.

3 Réseau de tri

On considère un réseau décrit dans la Figure ci-dessous avec les conventions habituelles : les barres verticales représentent les comparateurs à deux entrées. A la sortie, le fil haut (resp. bas) recevant le max (resp. min).



1. Est-ce un réseau de tri ?
2. Même question pour le réseau suivant



3. S'il en existe un qui ne marche pas, comment l'améliorer ?

4 Analyse d'un système pair-à-pair très simple

Nous allons construire un mini-système pair-à-pair autour d'une mini-table de hachage distribuée. Les utilisateurs sont répartis sur un anneau dont les sommets sont numérotés de 0 à $2^n = K$.

Afin d'associer une place dans l'anneau quand un utilisateur c se connecte, on utilise une fonction de hachage h qui associe à l'identifiant (par exemple son adresse IP) de c un entier $h(c) \in [0, K]$. On confond pour faire simple c avec son identifiant. On supposera l'existence d'un serveur central qui gère cette fonction de hachage et qui fournit au nouvel utilisateur sa place dans l'anneau ainsi que la place de son prédécesseur dans l'anneau ainsi que son adresse IP.

1. Donner un exemple d'une telle fonction de hachage.
2. Préciser les protocoles d'insertion et de départ des utilisateurs.

3. On se propose de gérer les fichiers à partager sur le réseau avec la même fonction de hachage. Ainsi à chaque nom de fichier f on associe $h(f) \in [0, K]$, un sommet de l'anneau. Chaque utilisateur c connaît son précesseur $pred(c)$ et il va gérer les fichiers de l'intervalle $[h(pred(c)) + 1, h(c)]$.

Préciser l'algorithme qui permet à un utilisateur c du réseau de trouver un fichier f , connaissant $h(f)$. Soit M le nombre d'utilisateurs présents à un moment donné. Montrer que cette procédure utilise au plus $O(M)$ opérations de communications. Montrer sur un exemple que cette borne peut-être atteinte.

4. Si l'on suppose l'anneau doublement chaîné, est-ce que cela améliore la procédure ?
5. Proposer des protocoles d'affectation des fichiers aux utilisateurs qui assurent la redondance des fichiers.
6. * Si l'on ajoute en chaque noeud i du réseau un lien vers un autre utilisateur de numéro $p(i) \in [0, K]$. Préciser le nouvel algorithme de routage. Comment choisir la probabilité de ces nouveaux liens pour qu'en moyenne le routage soit en $O(\log M)$.
7. Ce système est-il sûr ? Comment lui permettre de résister à des attaques ?
8. Comment éviter le serveur central de hachage. Peut-on distribuer cette opération, si oui préciser comment le faire ?
9. Quelles autres améliorations de ce mini-système proposeriez-vous ?

5 Autostabilité

Pour synchroniser des horloges de taille bornée par M , on a le choix entre les deux procédures suivantes:

1. A chaque étape, chaque processeur choisit avec une certaine probabilité l'horloge $h(k)$ d'un de ses voisins, et fixe son horloge à $h(k) + 1$ modulo M .
2. A chaque étape, chaque processeur choisit avec une certaine probabilité l'horloge $h(k)$ soit son horloge soit celle d'un de ses voisins, et fixe son horloge à $h(k) + 1$ modulo M .

A votre avis quelle est la meilleure procédure pour obtenir un algorithme autostabilisant de synchronisation d'horloges sur un réseau anonyme ?