Examen du cours Mobilité
M2 Pro, Université Paris Diderot, Durée 3h

Michel Habib

March 21, 2017

Note: les exercices sont indépendants, pour une question l'étiquette * signale sa difficulté.

1 Questions de cours

1. Rappelez la définition d'un système autostable et expliquer pourquoi cette notion est importante dans le cadre des applications distribuées, on donnera des exemples.
2. Rappelez la définition d'un système pair-à-pair. Donner trois exemples de systèmes pair-à-pair vraiment utiles (c'est à dire, ne servant pas qu'à télécharger des fichiers illégaux). Expliquer pourquoi ces systèmes sont plus performants avec une architecture pair-à-pair qu'avec une architecture client-serveur.
3. Pourquoi est-ce difficile de réaliser une application distribuée ?
4. Pourquoi les algorithmes de parallélisation automatique de code, ne suffisent-ils pas à résoudre tous les problèmes de la programmation parallèle ?

2 Machines PRAM

Pour chacune des questions, 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 ainsi que son travail. On précisera aussi le modèle de machine PRAM qui permet de faire tourner votre algorithme (CRCR, CREW ou encore EREW).

On considère une forêt d'arborescences. Chaque noeud i d'une arborescence est associé à un processeur $P(i)$ et possède un pointeur vers son père: $pere(i)$.

1. Ecrire un algorithme P-RAM CREW pour que chaque noeud i détermine $racine(i)$ l'adresse de la racine de son arborescence. Déterminer la complexité.

2. Une question de cours au passage: comment définir un algorithme de complexité optimale sur une machine P-RAM ?
3. L'algorithme de la question 1 utilise-t-il des écritures concurrentes ?
4. Peut-il s'éxecuter sur une machine EREW ? A quel coût ?
5. * En déduire un algorithme P-RAM qui calcule les composantes connexes d'un graphe.

3 Autostabilité

On considère un anneau avec un privilège qui tourne sur cet anneau, comme dans l'algorithme autostable d'exclusion mutuelle de Dijkstra vu en cours. On supposera en outre la taille de l'anneau connue. Il est constitué de N processeurs.

1. Comment peut-on gérer l'accès au privilège lorsque le système n'est pas dans un état légal? Afin par exemple, si le privilège représente l'accès à une imprimante, et que cette tâche n'est pas urgente (c'est à dire qu'il est préférable que l'impression se fasse à coup sûr, plutôt qu'instantanément). Comment éviter de perdre des impressions?
2. Même question dans le cas où l'on privilégie la vitesse de réponse plutôt que la sureté.
3. Peut-on généraliser, ces méthodes à d'autres réseaux, si oui lesquels ?

3.1 Rotor routage

Définition 1: un parcours (resp. cycle) Eulérien d'un graphe orienté est un parcours (resp. cycle) qui utilise une fois et une seule chaque arc (par contre il peut passer plusieurs fois en un sommet).

Définition 2: on appelle rotor routage, un mécanisme de routage dans les réseaux qui dont le principe de fonctionnement est défini ci-après. On suppose disposer en chaque sommet d'une fonction de routage nommée Rotor.

On suppose que les arcs sortant d'un sommet sont numérotés de 0 à $degre^+(x)$-1. $degre^+(x)$ étant le nombre d'arcs sortant de x. Quand un paquet passe en un noeud x, le routage envoie le paquet sur l'arc sortant numéro $Rotor(x)$, et exécute l'instruction : $Rotor(x) \leftarrow Rotor(x) + 1$ modulo $degre^+(x)$.

On supposera initialement pour $\forall x$, $Rotor(x) = 0$.

1. Montrer sur un exemple de graphe orienté un parcours Eulérien (cf. définition 1). On peut choisir des petites grilles symétriques 2x2 ou 3x3, par exemple.

2. On considère un paquet unique avec un TTL (Time To Live) infini qui circule sur un réseau muni du rotor routage. En reprenant les grilles symétriques 2x2 ou 3x3, puis un arbre symétrique, exhiber les parcours possibles suivant les points de départ du paquet.
3. Montrer que sur les exemples précédents que le paquet suit un parcours Eulérien du graphe.
4. A quoi peut bien servir un tel parcours Eulérien ?
5. Dans quelle partie du cours avons nous déjà vu cette notion ?
6. Application à l'autostablité (on peut associer un privilège à chaque arc traversé) montrer que ce parcours Eulérien est autostable. Dans ce cas une panne c'est un noeud du réseau qui n'applique plus le rotor routage.
7. * Sur quels types de graphes orientés connexes le rotor routage d'un paquet construit un parcours Eulérien. (Piste: on peut essayer avec un graphe planaire connexe symétrique).
8. ** Que dire s'il y a plusieurs paquets qui circulent en même temps sur le graphe avec le rotor routage ? (Piste reprendre les exemples).