

Université Gaston Berger de Saint-Louis <hr/> UFR DES SCIENCES APPLIQUEES ET DE TECHNOLOGIE <hr/> Section Informatique	Master Développement et de Systèmes d'Information	Année Universitaire 2017-2018 B. DIOP
---	--	--

Fiche TD¹ N°2 - Traitement de problèmes parallèles (Suite)

Exercice 1

En informatique distribuée, l'élection de leader est le processus de désignation d'un processus unique en tant qu'organisateur d'une tâche répartie entre plusieurs ordinateurs (nœuds). Avant que la tâche ne commence, tous les nœuds du réseau ne savent pas quel nœud servira de « leader » (ou de coordinateur) de la tâche ou ne pourra pas communiquer avec le coordinateur actuel. Cependant, après l'exécution d'un algorithme d'élection de leader, chaque nœud du réseau reconnaît un nœud particulier et unique en tant que « leader ».

Modèle asynchrone : algorithme simple

Solution dans un anneau unidirectionnel

```

En ne recevant aucun message :
    Envoyer monID dans le sens des aiguilles d'une montre
À la réception de "m":
    Case of :
        m.id < monID: envoie m dans le sens des aiguilles d'une montre
        m.id > monID: rejeter m
        m.id == monID:
            leader = monID
            envoie <terminate, monID > dans le sens des aiguilles d'une
            montre
            Fin
En recevant <terminate, id>:
    leader = id;
    Envoyer <fin, id> dans le sens des aiguilles d'une montre
    Fin

```

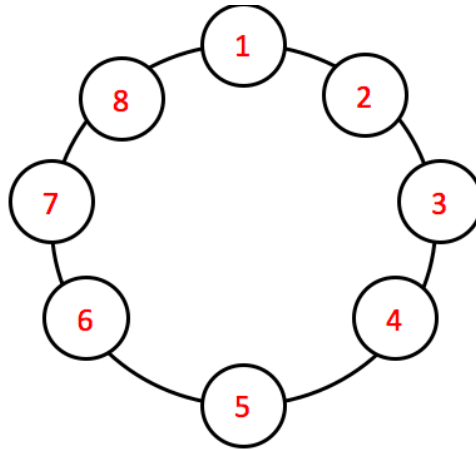


Figure 1. Exemple d'élection de leader dans un ring

Réponses aux questions :

1. Oui c'est un problème parallélisable avec dépendance de données entre processus
2. Complexité
 - a. Complexité au pire des cas $O(n * n)$.
 - b. Complexité d'échange de messages $O(n)$.
3. Les trois cas possibles concernent les trois cas de complexité :
 - a. Au meilleur
 - b. Au moyen
 - c. Pire des cas

Pour chacun des cas la complexité peut être calculé en fonction de n , le nombre de nœuds dans l'anneau.

Solution dans un anneau bidirectionnel

Pour les anneaux bidirectionnels asynchrones : réaliser un $O(n \log(n))$ en termes de complexité du message

Chaque nœud fonctionne en phases :

- Dans chaque phase, les nœuds encore actifs envoient leur ID dans les deux directions
- A la phase k , les jetons parcourent une distance de 2^k et retournent à leurs points d'origine
- Un jeton peut ne pas revenir en arrière s'il rencontre un nœud avec une valeur de ID inférieure
- Un nœud ne passe à la phase suivante que s'il reçoit ses jetons du tour précédent

En ne recevant aucun message

Si endormi alors

endormi = faux

phase = 0

envoie [monID, out, 1] à gauche et à droite

Après avoir reçu [id, out, h] de la gauche:

Cas où :

id < monID et h > 1: envoie [id, out, h-1] à droite

id < monID et h == 1: envoyer [id, in, -] à gauche

id = mon-id; leader = mon-id

En recevant [monID, in, -] de gauche à droite:

phase = phase + 1

envoie [monID, out, 2^{phase}] à gauche et à droite

En recevant [not-monID, in, -] de gauche ou de droite:

envoyer [not-monID, in, -] à droite ou à gauche

Correction de l'algorithme

- Les messages du nœud avec l'identifiant inférieur ne sont jamais ignorés
- Les messages provenant de nœuds avec un identifiant supérieur finissent par atteindre le nœud avec l'ID le plus bas et est ignoré
- Par conséquent, le bon leader est élu (sécurité)
- Finalement le nœud avec l'ID le plus bas atteint phase $\log(n)$ et envoie son ID sur l'ensemble de l'anneau

Complexité dans la communication

- En phase 0, chaque processeur envoie un message :
 - Maximum de $4n$ messages
- En phase $k + 1$:
- Seuls les processeurs qui envoient des jetons sont ceux qui ont "gagné" dans la phase précédente
- Il y a au plus un gagnant pour $2^{(k+1)}$ processeurs
- Gagnants après la phase k : $n/2^{(k+1)}$
- Les jetons parcourent une distance durant la phase $k + 1$ de : $2^{(k+1)}$
- Nombre total de messages dans la phase $k + 1$:
 - $4 * 2^{(k+1)} * \frac{n}{2^{(k+1)}} < 16n$
- Nombre total de phases : $1 + \log n$
- Nombre de messages : $O(n \log n)$

Modèle synchrone : 2 algorithmes

- L'élection de leader avec moins de messages $O(n \log n)$ est possible
 - Peut transmettre des informations en n'envoyant pas de message :
 - "Si vous n'entendez pas de moi, alors supposez que ..."
- Supposez que:
 - Les Uids sont des entiers positifs
 - Peut être manipulé à l'aide d'opérations arithmétiques arbitraires
- Deux algorithmes: **TimeSlice**, **VariableSpeeds**
- TimeSlice:
 - n est connu de tous les processeurs (non-uniformes)
 - La communication unidirectionnelle est suffisante
 - $O(n)$ messages

- Rappelons qu'un tour dans les réseaux synchrones est :
 - Délivrer tous les messages, demander à chaque processeur d'effectuer un pas de calcul
- Définir la notion de phase
 - Chaque phase se compose de "n" tours
 - En phase $k \geq 0$
 - Si personne n'est encore élu
 - Processeur avec ID k :
 - Se déclare leader
 - Envoie un jeton avec son ID autour
- Complexité des messages : $O(n)$
- Complexité temporelle : $n * (\text{valeur minimale de l'ID})$