SCIENCES SORBONNE UNIVERSITÉ

TME 6 - Élection

Exercice(s)

Exercice 1 – L'algorithme de Chang & Roberts (1979)

L'algorithme de Chang & Roberts réalise une élection sur un anneau unidirectionnel où les communications sont asynchrones. Le principe est le suivant :

- l'élection peut être initiée par plusieurs processus ;
- un processus initiateur envoie un jeton portant son identité à son successeur ;
- lorsqu'un processus reçoit un jeton :
  - s'il n'est pas encore initiateur, il ne peut plus le devenir : il passe dans l'état battu et transmet le jeton ;
  - s'il est initiateur, il ne transmet le jeton que si celui-ci porte une identité supérieure à la sienne.

Question 1

Quel est le processus qui gagne l'élection ? Comment le sait-il ?

On veut implémenter une version de cet algorithme qui inclut une annonce de résultat : lorsque l'algorithme se termine, tous les processus connaissent l'identité du processus élu. D'autre part, pour conserver un certain indéterminisme, chaque processus décide aléatoirement s'il est initiateur ou non en exécutant le code suivant :

```
srand(getpid());
initiateur = rand()%2;
```

Ce choix laisse un risque (faible) de n'avoir aucun processus initiateur : dans ce cas, on interrompt l'exécution et on recommence...

Question 2

Implémentez l'algorithme sous MPI, en affichant

- l'identité des processus initiateurs,
- les opérations de destruction de jeton,
- le résultat de l'élection sur tous les sites.

Question 3

Exécutez plusieurs fois le programme pour vérifier que les résultats sont corrects quand le nombre et l'identité des initiateurs varient.

Exercice 2 – Élection sur un arbre

On veut implémenter un algorithme d'élection sur une topologie en arbre. Comme pour le calcul de minimum, il s'agit de faire remonter l'information depuis les feuilles. Comme celles-ci ne font pas forcément partie des initiateurs de l'élection, il faut les avertir qu'une élection est en cours : c'est le but des messages `reveil`.

Le principe général de l'algorithme est le suivant :

— Pour avertir tous les processus qu'une élection est en cours, chaque site initiateur envoie un message `reveil` à l'ensemble de ses voisins.
— Un site qui reçoit pour la première fois un message `reveil` renvoie un message `reveil` à l'ensemble de ses voisins, sauf celui qui vient de lui en envoyer un.
— Le principe de l'élection est ensuite celui du calcul de minimum sur un arbre. Un deuxième type de messages (`ident`) circule dans l'application : ce message porte la valeur de l'identité minimum connue du site émetteur. Un site ayant reçu un tel message de tous ses voisins sauf un envoie un message `ident` au voisin qui n'a pas émis.
— Un site ayant reçu un message `ident` de tous ses voisins connaît l'identité minimum dans l'arbre. Le processus ayant cette identité minimum est élu. Le site renvoie alors un message `ident` à tous ses voisins, sauf celui à qui il en a déjà envoyé un (comme on ne cherche pas à savoir qui a pris la décision, on n'a pas besoin d'un troisième type de message).

Dès qu'une feuille sait qu'une élection est en cours, elle peut envoyer son identité. Si elle est initiateur, il est inutile qu'elle envoie un message `reveil` avant son message `ident`. On veut donc réaliser une version de l'algorithme où les feuilles n'envoient qu'un seul message.

Question 1
Quelles sont les changements induits par cette modification dans le traitement de la réception des messages ?

Question 2
Implémentez cet algorithme sous MPI, en utilisant le même tirage aléatoire que dans l'exercice 1 pour décider si un site est initiateur ou non.

Question 3
Testez votre implémentation sur l'arbre utilisé comme exemple pour le calcul de minimum. Vous pouvez à nouveau récupérer les éléments de programme sur la page web de l'UE.