

UE INF4003 : TP 3

Election sur un graphe

Frédéric Raimbault

Rappel : ce TP doit être réalisé avant la séance, il sera testé et noté en début de séance. Au début de la séance, un travail complémentaire vous sera remis et vous serez évalué en temps limité sur ce nouveau travail. Les TP sont testés sous linux, sur les PC des salles de TP et sont exécutés sur le cluster *pedago*.

1 Principe

On reprend ici le principe de l'élection sur un graphe connexe quelconque :

- l'objectif est d'élire le processus de poids maximal (fonction dépendante de l'application),
- participer à une élection consiste pour un processus à lancer une exploration du réseau,
- plusieurs explorations peuvent être lancées simultanément,
- chaque exploration est identifiée par son initiateur et son poids,
- à terme, seule l'exploration initiée par le processus élu se termine correctement,
- tout processus de poids p_i visité par une exploration de poids p_j :
 - la stoppe s'il a déjà été visité par une exploration de poids p_k , avec $p_k > p_j$,
 - lance une exploration de poids p_i s'il ne l'a pas déjà fait et que $p_i > p_j$,
 - fait progresser l'exploration de poids p_j dans tous les autres cas.

Remarque : en cas d'égalité de poids c'est le poids provenant de la racine de rang le plus élevé qui est considéré comme le plus grand.

L'exploration consiste en un parcours séquentiel en profondeur. Les messages circulant sont de trois types :

- Les messages de type *explorer* véhiculent le quadruplet composé des éléments suivants.
 - *rac*, l'identité du site qui a lancé le parcours,
 - *poids*, le poids du site qui a lancé l'exploration,
 - *atteints*, l'ensemble des identités des sites visités par ce message,
 - *avisiter*, l'ensemble des identités des voisins immédiats des sites de *atteints* qui n'ont pas encore été atteints par cette exploration

Ces messages s'arrêtent quand *avisiter* est vide.

- Les messages de type *rebrousser* véhiculent le quadruplet (*rac*, *poids*, *atteints*, *avisiter*) et permettent de faire poursuivre l'exploration par les processus qui possèdent un voisin appartenant à *avisiter*.
- Les messages de type *conclure* informent de la terminaison.

2 Travail à réaliser

Votre travail consiste à mettre en œuvre l'algorithme d'élection sur un graphe par parcours séquentiel dont les principes ont été rappelés dans le paragraphe précédent.

Dans l'algorithme présenté en cours, le poids d'un processus se limite à son identité. Vous généraliserez cette fonction de poids par la méthode abstraite `Weight.getValue()`. Votre programme d'élection prendra en paramètre une instance de classe implémentant cette interface.

Pour embarquer un quadruplet dans le contenu d'un message, respectivement pour extraire un quadruplet du contenu d'un message, vous utiliserez la méthode `Quadruplet.toMsgContent(root,weight,done_list,todo_list)`, respectivement la méthode `Quadruplet.fromMsgContent(root,weight,done_list,todo_list)`.

L'interface `Weight` et la classe `Quadruplet` sont fournies. Vous utiliserez la bibliothèque `csp_V1` pour la mise en œuvre des processus et des messages.

3 Test

Vous testerez votre algorithme d'élection en utilisant comme fonction de poids la classe fournie `Identity`.