

ALGORITHMIQUE DISTRIBUÉE

MIF12

ÉLECTION DE LEADER

Isabelle GUERIN LASSOUS

perso.ens-lyon.fr/isabelle.guerin-lassous/index-M1if12.htm

isabelle.guerin-lassous@univ-lyon1.fr

Introduction

- Dans le cours précédent
 - Certains nœuds pouvaient avoir un rôle particulier
 - Source
 - Racine
 - ...
- Comment choisir ce ou ces nœuds spécifiques ?
 - Objectif de l'élection de leader
- L'élection de leader doit être
 - Sûre
 - Le leader doit être unique et un nœud a été élu
 - Rapide
 - Le choix doit se faire en un temps fini
- L'élection de leader permet de rompre la symétrie dans un système distribué
 - Symmetry breaking

Applications de l'élection de leader

- **Orchestration** dans un système distribué
 - Contrôle centralisé
 - Allocation de ressources en exclusion mutuelle
 - Consensus
 - Débloquent une situation de blocage
- **Faire face à des défaillances**
 - Restaurer un leader
 - Restaurer un jeton
- **Récupérer de l'information sur le graphe**
 - Détection de la terminaison d'un algorithme
 - Déterminer le nombre de nœuds dans le réseau

Résultat d'impossibilité

- Il n'existe pas d'algorithme déterministe d'élection de leader dans les systèmes distribués anonymes et uniformes
- Systèmes anonymes
 - Nœuds n'ont pas d'ID
 - Impossible de différencier les nœuds
 - Système est donc uniforme
 - Même degré, même code, même état initial
- Preuve sur un anneau

Preuve

Preuve

Contourner ce résultat d'impossibilité

- En utilisant des **identifiants uniques** sur les nœuds
 - Le système n'est plus anonyme et uniforme
- En utilisant des **algorithmes probabilistes**
 - Tirages aléatoires sur chaque nœud
 - Tomber indéfiniment sur des configurations symétriques est quasi impossible

Élection de leader dans un anneau unidirectionnel ou directionnel

Élection de leader dans un anneau unidirectionnel

Algorithme de Chang-Roberts

- **Hypothèses**

- Chaque nœud a un **identifiant unique** et **sait** que les identifiants sont uniques
- Chaque nœud **connaît son voisin**
- Le **nombre de nœuds** dans le système est **inconnu** de chaque nœud

- **Idées**

- Au départ, **chaque nœud est candidat** à l'élection
 - Et donc **propage sa candidature** via son ID
- Un nœud qui **reçoit un ID supérieur** au sien **ne sera pas élu**
 - Et retransmet l'ID reçue à son voisin
- Le nœud, avec l'ID maximal, **reçoit son propre ID** et est élu

- **Variables locales**

- **Etat** : non candidat / candidat / élu / perdu *# initialisé à non candidat*
- **Leader** : ID du leader ; initialisé à NULL
- **Succ** : successeur du nœud dans l'anneau

Élection de leader dans un anneau unidirectionnel

Algorithme de Chang-Roberts

1. Nœud i se porte candidat
 - $Etat := candidat$
 - $Leader := ID$
 - Envoi du message $Elec(Leader)$ à Succ
2. Si le nœud i reçoit le message $Elec(j)$
 - Si $ID > j$
 - Si $Etat \neq candidat$, alors le nœud i se porte candidat
 - Sinon si $ID < j$
 - $Etat := perdu$
 - $Leader := j$
 - Envoi du message $Elec(Leader)$ à Succ
 - Sinon si $ID = j$
 - $Etat := élu$
 - Envoi du message $Lead(ID)$ à Succ
- Algorithme à compléter avec la diffusion du message $Lead$ à tous les nœuds de l'anneau
 - Permet aux nœuds :
 - de connaître le leader
 - de savoir que l'algorithme est terminé

Algorithme de Chang-Roberts

Exemple 1

Algorithme de Chang-Roberts

Exemple 2

Élection de leader dans un anneau unidirectionnel

Algorithme de Chang-Roberts

- **Complexité**

- En nombre de messages
 - Au mieux $2n$ messages
 - Si le 1^{er} nœud à se déclarer candidat est le nœud de plus grand ID, on peut avoir seulement n messages *Elec* et n messages *Lead*
 - Dans le pire cas
 - $1+2+\dots+n$ messages *Elec* et n messages *Lead*
 - $O(n^2)$

Élection de leader dans un anneau unidirectionnel anonyme

Algorithme d'Itai-Rodeh

- **Hypothèses**

- Nœuds n'ont plus un identifiant unique
- Nœuds connaissent le nombre total de nœuds n
- Chaque nœud connaît son voisin
- Communications FIFO

- Algorithme probabiliste

- **Variables locales**

- **Etat** : actif / inactif
- **Phase** # *fonctionnement de l'algorithme en phase*
- **Succ** : successeur du nœud dans l'anneau

Élection de leader dans un anneau unidirectionnel anonyme

Algorithme d'Itai-Rodeh

- Pour chaque nœud
 1. Etat := actif
 2. Phase := 1
 3. Tirer un identifiant id aléatoirement dans $[1 ; k]$ ($k > n$)
 4. Envoyer (id,Phase,1,vrai) à Succ *# (ID, Num phase, Nb sauts traversés, Leader unique)*
 5. Tant que pas de leader
 - À la réception de (#id,#phase,#saut,unique)
 - Si Etat = inactif *# le nœud n'avait pas démarré l'algo, il ne se porte pas leader car un autre nœud l'a fait pour lui*
 - Envoyer (#id,#phase,#saut+1,unique) à Succ
 - Sinon *# le nœud est actif*
 - Si #saut = n *# nœud initiateur du message reçu*
 - Si unique = vrai
 - Nœud élu
 - Informer les nœuds de l'anneau
 - Sinon
 - Phase := Phase + 1
 - Tirer un id aléatoirement dans $[1 ; k]$ ($k > n$)
 - Envoyer (id,Phase,1,vrai) à Succ
 - Sinon *# saut <> n*
 - Si (id,Phase)=(#id,#phase) *# un même identifiant a été trouvé lors de la même phase*
 - Envoyer (id,phase,#saut+1,faux) à Succ
 - Si (#id,#phase) > (id,Phase) *# un nœud a tiré un ID plus grand que moi*
 - Envoyer (#id,#phase,#saut+1,unique) à Succ

Algorithme d'Itai-Rodeh

Exemple 1

Algorithme d'Itai-Rodeh

Exemple 2