ALGORITHMIQUE DISTRIBUÉE
MIF12

ÉLECTION DE LEADER
ENSEMBLE INDÉPENDANT MAXIMAL

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
  - Débloquer 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 <> 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 1er 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+...+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 forcément 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
  - Si le nœud est actif
    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)*
  - Tant que pas de leader
    - À la réception de (#id,#phase,#saut,unique)
      1. 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
      2. 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

Élection de leader dans un anneau bidirectionnel Algorithme d'Hirschberg-Sinclair

- Hypothèses
  - Chaque nœud a un voisin à droite et à voisin à gauche auxquels il peut envoyer des messages
  - Chaque nœud peut recevoir des messages de ses 2 voisins
  - Nœuds ont un identifiant unique
- Idées
  - À la vague 0, tous les nœuds sont en compétition
  - Tous les nœuds qui gagnent à la vague *r* ont le droit de participer à la vague *r+1*
  - À la vague r, un nœud est gagnant s'il a le plus grand identifiant parmi les 2^r voisins à sa droite et les 2^r voisins à sa gauche
  - Deux nœuds qui restent en compétition après la vague *r* sont à une distance > 2^r
  - O(n.log(n)) messages échangés

Élection de leader dans un anneau bidirectionnel Algorithme d'Hirschberg-Sinclair

1. Chaque nœud i envoie *Election(i,0,1)* à ses voisins de gauche et de droite
2. À la réception de *Election(id,r,d)* reçu de son voisin de gauche (resp. droite) par le nœud i
   - Si (id > i) & ($d < 2^r$) alors
     - envoi de *Election(id,r,d+1)* à son voisin de droite (resp. gauche)
   - Si (id > i) & ($d \geq 2^r$) alors
     - envoi *de Reply(id,r)* à son voisin de gauche (resp. droite)
   - Si id = i, alors
     - envoi *Elected(i)* à son voisin de gauche
     - *Elu(i) := vrai*

Élection de leader dans un anneau bidirectionnel Algorithme d'Hirschberg-Sinclair

3. À la réception de *Reply(id,r)* reçu de son voisin de gauche (resp. droite) par le nœud i
   - Si (id <> i) alors envoi de *Reply(id,r)* à son voisin de droite (resp. gauche)
   - Sinon
     - Si déjà reçu *Reply(id,r)* de son voisin de droite (resp. gauche) alors
       - Envoi de Election(i,r+1,1) à ses voisins de gauche et droite

4. À la réception d'un message *Elected(id)* reçu de son voisin de droite par le nœud i
   - *Leader(i) := id*
   - *Done(i) := vrai*
   - Si (id <> i)
     - *Elu(i) := faux*
     - Envoi de *Elected(id)* à son voisin de gauche

Algorithme d'Hirschberg-Sinclair
Exemple

Algorithme d'Hirschberg-Sinclair
Exemple

Élection de leader dans un anneau unidirectionnel
Algorithme de Dolev-Klawe-Robeh

- Hypothèses
  - Chaque nœud a un voisin à gauche
  - Nœuds ont un identifiant unique et savent que les identifiants sont uniques
- Idées
  - À la vague 0, tous les nœuds sont en compétition
  - Si j > max(i,k), alors i reste dans la compétition, au nom de j
  - Si j < max(i,k), alors i n'est plus dans la compétition et relaye seulement les messages vers son voisin
  - Si i reste dans la compétition alors h et j ne peuvent pas être dans la compétition
  - À la phase 2, au plus la moitié des nœuds resteront dans la compétition
  - 0(n.log(n)) messages échangés

Algorithme de Dolev-Klawe-Robeh
Éléments de preuve

$$h \leftarrow i \leftarrow j \leftarrow k \leftarrow l$$

Élection de leader dans une topologie quelconque

Élection dans une topologie quelconque Algo 1

- Hypothèse
  - Système connexe
  - Chaque nœud connaît le nombre de nœuds dans le système
  - Chaque nœud connaît son identifiant unique compris entre [1 ; n]

- Idée
  - Le leader est celui qui a l'identité la plus grande (ou la plus petite)
  - Il est déterminé par chaque nœud sans échange de message

Élection dans une topologie quelconque Algo 2

- Hypothèse
  - Système connexe
  - Chaque nœud connaît le nombre de nœuds dans le système
  - Chaque nœud connaît son identifiant unique

- Idée
  - Diffusion de son identité dans tout le système
  - Lorsque chaque nœud a reçu toutes les identités du système, le leader est celui qui a l'identité la plus grande (ou la plus petite)

- O(nm) messages

Élection dans une topologie quelconque
Algo 2 : exemple

Election dans une topologie quelconque Algo 3

- Hypothèse
  - Système connexe
  - Chaque nœud connaît le diamètre du graphe sous-jacent
  - Chaque nœud connaît son identifiant unique

- Idée
  - Algorithme faiblement synchrone
  - Au départ *Leader := ID du nœud* et envoi du Leader à tous ses voisins
  - À chaque ronde
    - Chaque nœud reçoit un message de tous ses voisins
      - *Leader := max sur les leaders reçus et le sien*
      - Envoi du Leader à tous ses voisins
  - Au bout de D rondes, chaque nœud connaît le leader

- O(Dm) messages

Election dans une topologie quelconque
Algo 3 : exemple

Ensemble indépendant maximal

Ensemble indépendant maximal

- Ensemble indépendant dans un graphe
  - Sous-ensemble de sommets du graphe tel que 2 sommets quelconques de ce sous-ensemble ne sont pas voisins dans le graphe
- Intérêt
  - Par ex., faire travailler en même temps des nœuds sur des données / ressources disjointes

- Ensemble indépendant M est maximal
  - S'il n'existe pas d'ensemble indépendant M', différent de M, et tel que M soit inclus dans M'
- Ensemble indépendant maximum
  - Ensemble indépendant de plus grande cardinalité
  - Problème NP-hard

Ensemble indépendant maximal
Exemples

Ensemble indépendant maximal
Algorithme simple mais lent

- Hypothèse
  - Chaque nœud a un identifiant unique
- Au départ, état de chaque nœud est indécis
- Pour chaque nœud, répéter
  1. Si tous les voisins de plus grand ID ont décidé de ne pas rejoindre l'EIM
     - État := appartient # le nœud rejoint l'EIM
  2. Si un voisin est dans l'EIM
     - État := n'appartient pas
  3. Envoyer son état
  4. Recevoir les états de chacun des voisins non dans l'EIM
- Algorithme faiblement synchronisé
- Algorithme qui peut être très lent

Algorithme simple mais lent
Exemple

Ensemble indépendant maximal
Utilisation du coloriage

- Initialisation
  - Chaque nœud connaît tous ses voisins
  - Il existe un k-coloriage du graphe
- Idées
  - Les nœuds qui ont la même couleur appartiennent à un ensemble indépendant, pas forcément maximal
  - Ajout de nœuds à un ensemble indépendant initial tant que cela est possible, en itérant sur les couleurs
  - Chaque nœud va maintenir un tableau indiquant si lui ou ses voisins font partie de l'ensemble indépendant maximal en construction
    - Éléments du tableau à faux au départ

Ensemble indépendant maximal
Utilisation du coloriage

- Algorithme
  - Au départ, aucun nœud n'appartient à l'ensemble indépendant
  - Pour i de 1 à k
    - Pour chaque nœud u
      - Si Couleur[u]=i
        - Si aucun voisin ne fait partie de l'ensemble indépendant
          - Sélection[u] := True
      - Envoi de Sélection[u] à tous ses voisins
      - Attendre un message de chacun de ses voisins v et mettre à jour Sélection[v]

- Algorithme faiblement synchronisé
- Taille de l'ensemble indépendant lié à l'ordre des couleurs
- Terminaison
  - Fin de la 1ère boucle Pour
- k rondes

Utilisation du coloriage
Exemple

Ensemble indépendant maximal Algorithme rapide

- Algorithme faiblement synchrone et probabiliste
- Lors d'une ronde
  1. Chaque nœud u choisit de rejoindre l'ensemble indépendant (EI) avec une probabilité 1/d(u) où d(u) correspond au degré de u
  2. Échange des choix entre voisins
  3. Si le nœud u a choisi de rejoindre l'EI
     - Si aucun voisin de u de degré plus élevé n'a rejoint l'EI alors u reste dans l'EI
     - Sinon u se retire de l'EI
  4. Échange des choix entre voisins
  5. Les nœuds qui viennent de rejoindre l'ensemble indépendant ainsi que leurs voisins ne participent plus aux rondes suivantes

- Si 2 nœuds voisins ont choisi de rejoindre l'ensemble indépendant et ont le même degré, utilisation des identifiants pour les départager
  - => identifiants uniques
- Correction de l'algorithme ?
- L'algorithme de termine en moyenne avec O(log n) rondes (n nombre de nœuds)

Ensemble indépendant maximal
Algorithme rapide : exemple

Ensemble indépendant maximal
Autre algorithme rapide

- Algorithme faiblement synchrone et probabiliste
- Lors d'une ronde
  1. Chaque nœud u choisit une valeur r(u) aléatoirement dans [0; 1]
  2. Échange des valeurs entre voisins
  3. Si r(u) < r(v) pour tous voisins v de u alors u rejoint l'ensemble indépendant et informe ses voisins
  4. Les nœuds qui viennent de rejoindre l'ensemble indépendant ainsi que leurs voisins ne participent plus aux rondes suivantes

- Correction de l'algorithme ?
- L'algorithme de termine après au plus $3\log_{4/3} m + 1$ rondes en moyenne (m nombre d'arêtes)

Ensemble indépendant maximal
Autre algorithme rapide : exemple

Ce qu'il faut retenir

- Intérêt et applications de l'élection de leader
- Résultat d'impossibilité dans les systèmes distribués anonymes
- Algorithmes d'élection de leader sur un anneau unidirectionnel et bidirectionnel et leur complexité en nombre de messages
- Algorithmes d'élection de leader sur des topologies quelconques et leur complexité en nombre de messages
- Ensembles indépendants maximal et maximum : définitions
- Algorithmes pour construire un ensemble indépendant maximal