Synchronisation Distribuée
Algorithmes pour les environnements incertains

Laurent PHILIPPE

Master 2 Informatique
UFR des Sciences et Techniques

2022/2023

Sommaire

Sommaire

Environnements incertains

Définition

- Nombre de processus non connus, éventuellement borné
- Pertes de messages
- Pertes de connexion
- Fautes

Introduction

- Systèmes P2P
- Réseaux Ad-Hoc

Sommaire

Un algorithme pour systèmes pairs à pairs

Le protocole SIGMA (Lin et all)

- Plusieurs ressources en exclusion mutuelle
- Chaque ressource est dupliquée de façon logique $\mathtt{n}$ fois
- Chaque réplique est associée à un coordinateur (site) qui gère son accès (extension du mode centralisé)
- Quand un client veut accéder à une ressource : il doit avoir la permission de $\mathtt{m}$ ($> n/2$) coordinateurs
  $\Rightarrow$ doit être propriétaire de $\mathtt{m}$ réplicas

Le protocole SIGMA

Implantation

- Utilisation d'un système à base de DHT (Distributed Hash Table) pour accéder aux répliques
- Chaque ressource possède un nom unique appelé `rname`
- La ième réplique d'une ressource `rname` a le nom `rname-i` (utilisé pour calculer une clé avec une h-fonction)
- Chaque processus sait calculer les `n` clés correspondantes aux `n` répliques d'une ressource

Le protocole SIGMA

Structures de données de chaque client C_i

- i : identificateur
- Tableaux contenant les réponses des répliques.
  - $Rep_P[]$ mémorise son propriétaire
  - $Rep_H[]$ l'horloge logique associée à la requête
- H_i : horloge logique

Le protocole SIGMA

Structures de données de chaque réplique D_k

- C_p : client propriétaire de cette réplique
- H_p : horloge logique correspondant à la requête du propriétaire
- *Queue* : file contenant les clients en attente de réponse organisés dans l'ordre de leurs horloges logiques

Le protocole SIGMA

Client

Accès à une ressource :

- Calcule son horloge logique H_i
- Envoie une requête aux n répliques pour demander l'accès : $REQ(H_i)$
- Attend réponses (nombre à fixer)

Le protocole SIGMA

Client

Réception d'une réponse :

- Stocke la réponse dans les tableaux $Rep_H[]$ et $Rep_P[]$
- Quand assez de réponses (avec $m > n/2$), calcule le gagnant de l'accès à cette ressource : propriétaire de `m` répliques
  - S'il est gagnant, entre en Section Critique
  - S'il n'y pas de gagnant, libère les répliques dont il est propriétaire (envoi du message `YIELD` aux répliques) et se remet en attente de réponses

Le protocole SIGMA

Client

Sortie de Section critique :

- Informe toutes les répliques associées à la ressource (envoi du message `REL`)

Le protocole SIGMA

Réplique

Réception d'une requête depuis un client j :

- Si la réplique a déjà un propriétaire : requête est mise en file d'attente
- Si la réplique n'a pas de propriétaire : j devient propriétaire de cette réplique, son horloge est enregistrée
- Dans tous les cas, envoi d'une réponse à j contenant le propriétaire de la réplique

Le protocole SIGMA

Réplique

Réception d'un message `REL` depuis un client j :

- Si j était propriétaire de la réplique :
  - la réplique devient libre
  - prend le 1er client C de la file d'attente et lui envoie une réponse : C devient propriétaire de cette réplique
- Si j n'était pas propriétaire de la réplique : aucune action

Le protocole SIGMA

Réplique

Réception d'un message `YIELD` depuis un client j :

- Si j était propriétaire de la réplique :
  - le client j est mis en file d'attente
  - prend le 1er client C de la file d'attente et lui envoie une réponse : C devient propriétaire de cette réplique
- Si j n'était pas propriétaire de la réplique : aucune action

Le protocole SIGMA

Commentaires

- Si ressource beaucoup demandée : il se peut qu'aucun client ne puisse devenir propriétaire (résolu par "YIELD")
- Section critique attribuée pour un temps donné à un client (bail) : tolérance aux fautes
- Problème si coordinateur d'une réplique tombe en panne et redémarre : peut avoir perdu ses informations et donner des permissions inadéquates (probabilité très faible)

Sommaire

Election dans les environnements sans fil (Vasudevan et all)

Présentation

- Algorithmes traditionnels d'élection basés sur des hypothèses non réalistes dans les environnements sans fil (ex : échanges de messages fiables, connaissance de tous les processus, pas de modification de la topologie du réseau ...)
- Algorithme de Vasudevan propose de gérer la panne des nœuds et le partitionnement du réseau
- S'inspire de l'algorithme de détection de la terminaison de Dijsktra et Scholten
- Permet de choisir un processus en fonction d'informations (capacité, ressources, ...) et non plus uniquement à partir de son numéro

Election dans les environnements sans fil (Vasudevan et all)

Principe de l'algorithme

- 3 messages : `ELECTION`, `ACK`, `LEADER`
- Nœud `source` : noeud qui détecte la panne et lance l'élection
- Construction d'un arbre de recouvrement (Spaning Tree) ayant pour racine ce nœud `source`
- Quand il détecte la panne du coordinateur, le nœud `source` envoie un message `ELECTION` à ses voisins immmédiats
- 1er temps : présentation de l'algorithme sans mobilité, ni panne

Election dans les environnements sans fil (Vasudevan et all)

Principe de l'algorithme

Détection de la panne du coordinateur par P_i

- P_i devient le nœud `source`
- P_i envoie un message `ELECTION` à ses voisin immmédiats
- P_i attend des aquittements de tous ses voisins (`ACK`)

Election dans les environnements sans fil (Vasudevan et all)

Principe de l'algorithme

Réception d'un message `ELECTION` de P_j

- Si premier message `ELECTION` reçu par P_i :
  - P_j devient le `père` de P_i pour cette élection,
  - P_i envoie un message `ELECTION` à ses voisins immédiats (sauf à son `père`)
  - P_i attend les réponses
- Si P_i a déjà un père, il envoie un message `ACK` sans information à P_j

Election dans les environnements sans fil (Vasudevan et all)

Principe de l'algorithme

Réception d'un message `ACK` de P_j

- Comptabilisation de l'aquittement
- Si message `ACK` contient de l'information :
  Stocker cette information, enregistrer P_j comme fils
  Si dernier aquittement attendu :
  Aggréger toutes les informations reçues de ses fils
  - si (non nœud `source`) : envoyer les informations aggrégées à son père
  - si nœud `source` : déterminer le leader et envoyer un message `LEADER` sur l'arbre de recouvrement

Election dans les environnements sans fil (Vasudevan et all)

Exemple

5 processus A, B, C, D, E

- A (3) a comme voisins immédiats B et C
- B (2) a comme voisins immédiats C et E
- C (5) a comme voisins immédiats D et B
- D (10) a comme voisins immédiats C et E
- E (7) a comme voisins immédiats B et D
- A détecte la panne du coordinateur

Election dans les environnements sans fil (Vasudevan et all)

Mobilité et pannes

- Gestion de plusieurs élections en parallèle
- Partitionnement du graphe
- Panne de nœuds, redémarrage