Autostabilisation II — le retour

Slide 1

Élection d'un chef – leader election

Jérôme DURAND-LOSE
Laboratoire I3S, CNRS UPREES-A 6070,
06903 SOPHIA ANTIPOLIS Cedex,
FRANCE.

jdurand@unice.fr http ://www.i3s.unice.fr/~jdurand/

Plan de l'exposé

Slide 2

Problématique

Réponse de S. Dolev, A. Israeli and S. Moran
Uniform Dynamics Self-Stabilizing Leader Election,
IEEE Transactions on parallel and distributed systems 8(4), 1997.

Réponse de J. Beauquier, J. Durand-Lose, M. Gradinariu et C. Johnen
Token based self-stabilizing uniform algorithms,
RR LRI 1250, 2000, soumis.

Cadre

Slide 3

Réseau de processeurs

Problème panne passagère (déconnexion, corruption de données/messages, redémarrage d'une machine...)

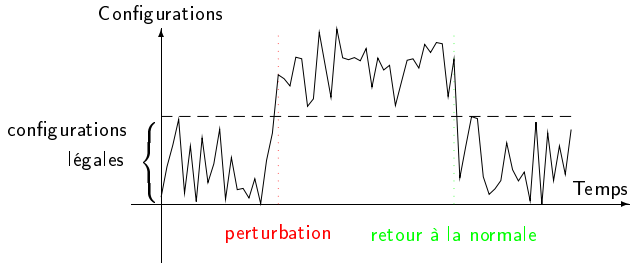
but retour à la « normale »

Condition sans intervention lourde (humaine, redémarrage du réseau...)

⇝ *autostabilisation*

Autostabilisation

Slide 4



Autostabilisation – plus formellement

Slide 5

$\mathcal{P}$	un paradigme
$\mathcal{L}$	une ensemble de configurations légales
correction	toute exécution commençant dans une configuration légale vérifie $\mathcal{P}$
convergence	toute exécution atteint une configuration légale en temps fini

Paradigmes couverts

Slide 6

Élection d'un chef

Exclusion mutuelle

Synchronisation

PIF (propagation avec retour)

Avoir un arbre recouvrant

Nommer les processeurs

Nombreux résultats (impossibilité ou algorithme)

Slide 7

Selon le paradigme	
Selon le *type de réseau*	anneau | graphe complet | graphe quelconque orienté | non
Selon la *similarité des nœuds*	uniforme, anonyme | semi-uniforme | avec identités
Selon le *mode de mise à jour*	séquentielle | synchrone | distribuée
Selon *l'« adversaire » considéré*	
Selon *l' atomicité des opérations*	
Selon la *connaissance du réseau*	
Selon le *processeur*	déterministe, aléatoire,
Selon *modèle de communication*	mémoire partagée, files files à délai borné, file ne conservant pas l'ordre

Folklore important en fonction des cas considérés

Cas considéré

Slide 8

Graphe connexe non orienté

Processeurs identiques, sans identité

Mise à jour quelconque

Adversaire quelconque

Aucune connaissance sur le réseau

Opération atomiques contiennent au plus une entrée ou une sortie

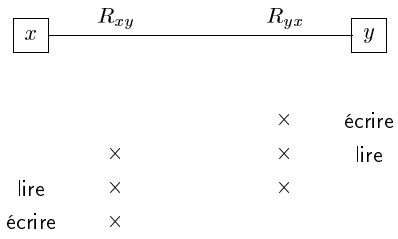
Algorithmes *aléatoires*

Communication par registres

cas relativement général

Communication par registres

Slide 9



Slide 10

S. Dolev, A. Israeli and S. Moran

Uniform Dynamics Self-Stabilizing Leader Election,

IEEE Transactions on parallel and distributed systems 8(4), 1997.

Principes

Slide 11

Processeur :

- soit racine
- soit fils d'un de ces voisins

Casser les cycles

Fusionner les arbres

chef ≡ Racine de l'unique arbre

Information détenue par un processeur

Slide 12

chaque processeur suppose être sur un arbre enraciné
avec des plus cours chemins à la racine

Id identité, mot sur {0,1}

$racine$ Id de la racine

d distance à la racine

p voisin père dans l'arbre

Attention se sont des *variables*

Elles ne reflète que ce que « croit » un processeur

Elles ne sont bien positionnées dans une configuration légale

Comportement d'un noeud

Slide 13

Cherche le voisin avec
1 - la racine la plus élevée possible
2 - la distance la plus petite

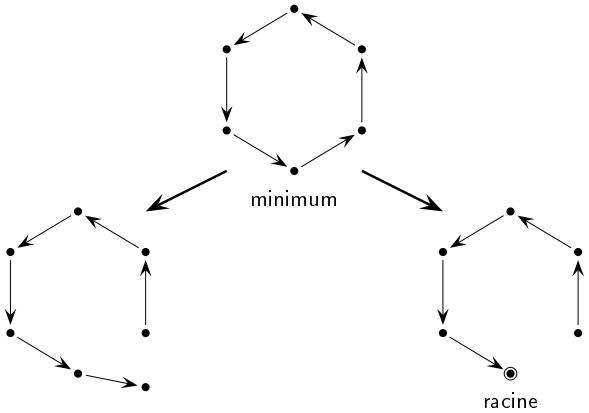
Si inférieur à l'information détenue
se met à jour

Sinon devient racine (balaye toute information)

Il ne peut apparaître qu'une racine avec Id plus élevé
ou un chemin plus court

Briser les cycles

Slide 14



Le minimum rompre le cycle en s'orientant hors du cycle ou en devenant une racine

Sans cycle

Slide 15

Progressivement les processeurs s'orientent

vers le chef avec l'Id la plus élevée et vers la distance minimale

Nombre de processeurs avec l'Id la plus élevée

1 se sera le chef

plusieurs ambiguïté...

Prendre connaissance de la présence d'autres chef puis tirer à pile ou face

¿ Oui, mais comment ?

Détection de concurrents

Slide 16

Colorier l'arbre, autre couleur ⇒ autre arbre

Chaque nœud garde en mémoire la couleur courante et la couleur précédante

Technique

Il tire une couleur au hasard parmi 8 différente des deux dernières

Déclenche le coloriage de l'arbre dans la couleur choisie

Attend un retour d'information disant que tous l'arbre est colorié

Recommance

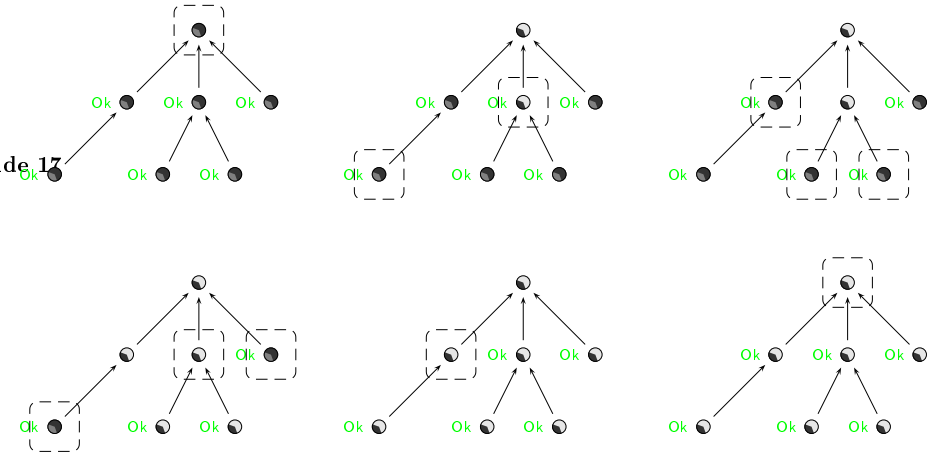
Dans un arbe, chaque nœud ne peut être que de la nouvelle couleur ou de la précédante

À la mise à jour de la couleur d'un nœud, si la couleur d'un voisin n'est pas une des deux couleurs... il n'est pas dans le même arbre

il fait remonter l'information au chef

Propagation de la couleur

Slide 17



Entre deux chefs

Slide 18

Quand un chef décelle la présence d'un autre chef, il tire au hasard 0 ou 1 et le rajoute devant son Id

La nouvelle Id se répandra en même temps que la prochaine couleur

Si l'autre ne l'a pas détecté ou à fait un mauvais tirage, tous les noeuds de l'autre arbre vont Progressivement se joindreau premier

Le hasard assure que :

- deux chefs voisins vont se détecter
- un prendra le pas sur l'autre

Critique

Slide 19

Aucune connaîssance nécessaire sur le graphe

Aléatoire

Convergence rapide :

$O(\Delta.D.\log(n))$ rounds

max degré

diamètre

nombre de sommets

Nombre d'états **non borné** car Id non borné (espérance de $O(\log(n))$)

Si le nombre d'états est connu, convergence en $O(\Delta.D)$ et nombre d'Id fini

Slide 20

J. Beauquier, J. Durand-Lose, M. Gradinariu et C. Johnen

Token based self-stabilizing uniform algorithms,

RR LRI 1250, 2000, soumis.

Exclusion mutuelle autostabilisante

Slide 21

Jeton $\Longleftrightarrow$ privilège

Problèmes possibles :

- plusieurs jetons
  $\rightsquigarrow$ marches aléatoires
- pas de jetons
  $\rightsquigarrow$ condition topologique

Condition topologique

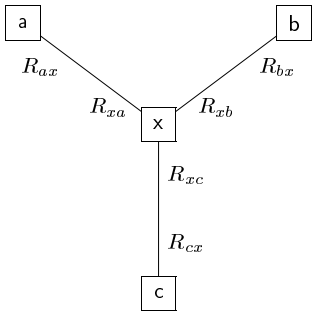
Slide 22

k t.q. $\neg(k \setminus n)$

$R_{.,.} \in \{0, 1, ..(k-1)\}$

Jeton $\Leftrightarrow$

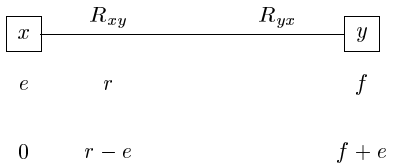
$\sum R_{x.} - \sum R_{.x} \neq 1 \bmod k$



Passage du jeton

Slide 23

$$e \;=\; (\,1 - (\,\sum R_{x.} - \sum R_{.x}\,)\,) \mod k$$



Élection

Slide 24

2 type de jetons

teste	**chef**
sert à tester la présence d'un autre chef	indique le chef
bouge en permanence	ne bouge que si un autre chef est détecté

Chaque jeton garde une couleur enmémoire

Algorithme

Slide 25

Quand un chef reçoit :

un jeton de teste de la **même couleur**

- il tire une nouvelle couleur pour lui et le jeton
- il relance le jeton de teste

un jeton de teste d'une **couleur différente**

- il se déplace un peu avec le jeton de teste
- il s'arrête et relâche le jeton de teste

S'il rencontre un autre chef, ils fusionnent

Ce qui fait converger

Slide 26

L'aléatoire assure que :

- le nombre de jeton de teste diminue jusqu'à un
- une fois qu'il n'y en a qu'un, un chef se déplacera ssi il y a un autre chef
- s'il y a deux chef, il finiront par fusionner

Critique

Différents mode de déplacement des jetons de teste sont possibles

Nombre d'états finis

Slide 27

Il faut connaître un nombre entier qui ne divise pas le nombre de sommets

Temps de convergence mal connu

Si l'on ajoute un niveau de jeton en dessous :

- graphes orientés
- tout type d'adversaire