

# Autostabilisation : de l'exclusion mutuelle sur un anneau à l'élection d'un chef sur un graphe quelconque

Jérôme DURAND-LOSE



Laboratoire d'Informatique Fondamentale d'Orléans,  
Université d'Orléans, ORLÉANS, FRANCE

JIRC – 30 juin 2005 – Blois



## Introduction

## Modèle

## Exclusion mutuelle

Jeton

Passage

## Élection d'un chef

Architecture

Correction

Convergence

## Conclusion

# Plan

## Introduction

## Modèle

## Exclusion mutuelle

Jeton

Passage

## Élection d'un chef

Architecture

Correction

Convergence

## Conclusion

# Cadre : Algorithmique distribuée

## Réseau de processeurs / machines

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

**but** retour à la « normale »

**Contrainte** sans intervention lourde  
(humaine, redémarrage du réseau...)

# Cadre : Algorithmique distribuée

## Réseau de processeurs / machines

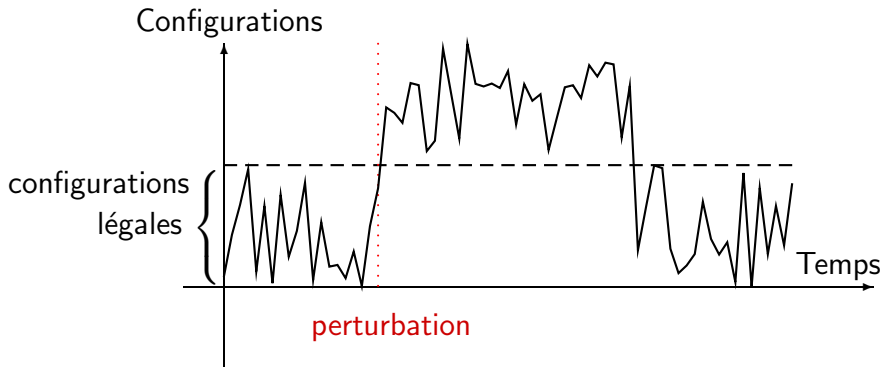
**Problème** panne passagère  
(redémarrage d'une machine, corruption de données/messages,  
déconnexion...)

**but** retour à la « normale »

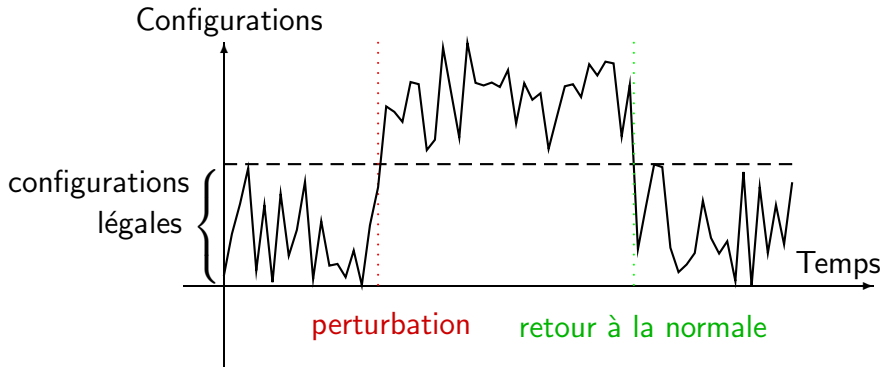
**Contrainte** sans intervention lourde  
(humaine, redémarrage du réseau...)

↪ *autostabilisation*

# Autostabilisation



# Autostabilisation



# Autostabilisation – plus formellement

$\mathcal{P}$  un paradigme

$\mathcal{L}$  une ensemble de configurations légales

*correction* toute exécution entrant dans une configuration légale vérifie  $\mathcal{P}$  à partir de cet instant

*convergence* toute exécution atteint une configuration légale en temps fini



# Paradigmes couverts

Élection d'un chef

Exclusion mutuelle

Synchronisation

PIF (propagation avec retour)

Avoir un arbre recouvrant

Nommer les processeurs

Engendrer un « sens de la direction »

...

# Paradigmes couverts

Élection d'un chef ←

Exclusion mutuelle ←

Synchronisation

PIF (propagation avec retour)

Avoir un arbre recouvrant

Nommer les processeurs

Engendrer un « sens de la direction »

...

# Origines de ces thématique et travaux

- ▶ de l'*autostabilisation*



Edsger W. Dijkstra.

Self-stabilizing systems in spite of distributed control.

*Journal of the ACM*, 17(11):643–644, 1974.

- ▶ exclusion mutuelle pour les anneaux



Joffroy Beauquier and Sylvie Delaët.

Probabilistic self-stabilizing mutual exclusion in uniform rings.

In *Principles of Distributed Computing (PODC '94)*, page 378, 1994.

# Résultats : impossibilités, complexités ou algorithmes

Selon le paradigme, mais aussi. . .

le <i>type de réseau</i>	anneau    graphe complet    graphe quelconque orienté    non
la <i>similarité des nœuds</i>	anonyme    avec identités uniforme    semi-uniforme
le <i>mode de mise à jour</i>	séquentielle    synchrone    distribué
l' <i>atomicité des opérations</i>	
les <i>processeurs</i>	déterministes, aléatoires,
<i>modèle de communication</i>	mémoire partagée, files files à délai borné, conservant ou non l'ordre avec / sans répétitions / pertes
la <i>connaissance du réseau</i>	
l'« <i>adversaire</i> » <i>considéré</i>	

# Plan

Introduction

Modèle

Exclusion mutuelle

  Jeton

  Passage

Élection d'un chef

  Architecture

  Correction

  Convergence

Conclusion

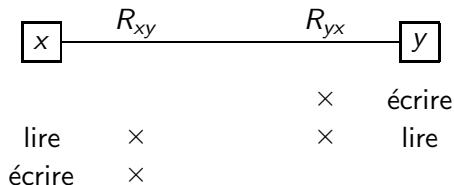
# Processeurs

Identiques et anonymes

Définition par

- ▶ automate fini probabiliste
- ▶ ensemble de règles du type  
Garde + probabilité : transition  
Garde : transitions  $\times$  probabilités

# Communication par registres



Graphe de communication non orienté quelconque

# Mise à jour

Séquentielle, distribuée ou parallèle

Granularité :

(au plus un registre de communication lu ou écrit par activation)

Différents types de démons

Selon le cas considéré, aménagement au niveau du code



# Plan

Introduction

Modèle

Exclusion mutuelle

Jeton

Passage

Élection d'un chef

Architecture

Correction

Convergence

Conclusion

## Définition – Exclusion mutuelle

Ressource partagée mais accessible à un seul client à la fois

Les propriétés suivantes doivent être vérifiées :

**Unicité** (*instantané*)

chaque instant,

un et un seul processeur est privilégié

**Équité** (*dynamique*)

dans une exécution infinie,

chaque processeur est privilégié infiniment souvent

Privlège  $\iff$  Jeton

# Exclusion mutuelle autostabilisante

Problèmes possibles :

Réponse

▶ inéquité

↪ mouvement aléatoire des jetons

▶ plusieurs jetons

↪ mouvement aléatoire + rencontre fusion

▶ pas de jetons

↪ condition topologique l'empêchant

# Plan

Introduction

Modèle

Exclusion mutuelle

Jeton

Passage

Élection d'un chef

Architecture

Correction

Convergence

Conclusion

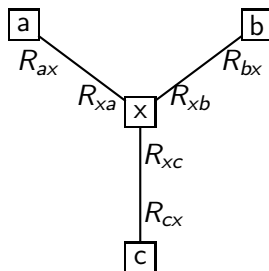
# Définition du jeton

Pour  $n$  machines

Soit  $k$  t.q.  $n \not\equiv 0 \pmod{k}$   
 $k$  est une *constante* du protocole

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

$R_{xy}$  vaut 0 s'il n'y a pas d'arc



$$x \text{ a un jeton} \iff \sum_y R_{xy} - \sum_y R_{yx} \not\equiv 1 \pmod{k}$$

«  $x$  ne donne pas ce qu'il ne reçoit  $+ 1$  »

(loi de Kirchhoff biaisée)

# Présence obligatoire d'un jeton

Aucun jeton

⇔

$$\forall x, \sum_y R_{xy} - \sum_y R_{yx} \equiv 1 \pmod{k}$$

⇔

$$\sum_x \sum_y R_{xy} - \sum_x \sum_y R_{yx} \equiv 0 \equiv n \pmod{k}$$

# Présence obligatoire d'un jeton

Aucun jeton

⇔

$$\forall x, \sum_y R_{xy} - \sum_y R_{yx} \equiv 1 \pmod{k}$$

⇔

$$\sum_x \sum_y R_{xy} - \sum_x \sum_y R_{yx} \equiv 0 \equiv n \pmod{k}$$

Or

$$n \not\equiv 0 \pmod{k}$$

↪ il doit obligatoirement y avoir *au moins un* jeton !

# Plan

Introduction

Modèle

**Exclusion mutuelle**

Jeton

Passage

Élection d'un chef

Architecture

Correction

Convergence

Conclusion

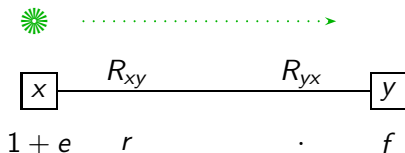


# Passage du jeton

soit

$$e = \left( 1 - \left( \sum R_{xy} - \sum R_{yx} \right) \right) \text{ mod } k$$

Passage

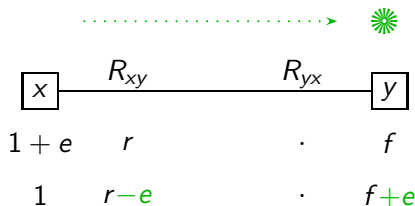


# Passage du jeton

soit

$$e = \left( 1 - \left( \sum R_{xy} - \sum R_{yx} \right) \right) \text{ mod } k$$

Passage



Si y n'a pas de jeton ( $f=1$ ), il y a passage

Si y a déjà un jeton ( $f \neq 1$ ), ils fusionnent ou s'annulent

# Plan

Introduction

Modèle

Exclusion mutuelle

  Jeton

  Passage

Élection d'un chef

  Architecture

  Correction

  Convergence

Conclusion

## Définition – Élection d'un chef

Besoin d'un processeur « distingué » pour coordonner les autres

Les propriétés suivantes doivent être vérifiées :

**Unicité** (*instantané*)

chaque instant,

un et un seul processeur est le chef

tous les processeurs sont d'accord sur le chef

**Stabilité** (*dynamique*)

on ne change jamais de chef

Chef  $\iff$  Jeton de chef 

# Élection d'un chef autostabilisante

Problèmes possibles :

Réponse

- ▶ pas de jetons chef  
    ↪ condition topologique (la même)
- ▶ plusieurs jetons chef  
    ↪ détection grâce à d'autres jetons  
    ↪ déplacement aléatoire si détection

# Plan

Introduction

Modèle

Exclusion mutuelle

  Jeton

  Passage

Élection d'un chef



  Architecture

  Correction

  Convergence

Conclusion

# Architecture

Jeton chef :  ou 

Ne bouge pas

Jeton de test :  ou 

Mouvement aléatoire  
Conserve sa couleur

Jetons implantés de la même manière

Un bit de plus dans les registres pour transmettre une couleur avec le jeton

Rencontre ? ?

# Rencontre

Si les deux jetons sont de même couleur

tirer aléatoirement une nouvelle couleur identique  
pour les deux  
relance le jeton de test

**Sinon** tirer aléatoirement la couleur du jeton de test  
relance le jeton de test  
passer le jeton de chef



# Plan

Introduction

Modèle

Exclusion mutuelle

  Jeton

  Passage

Élection d'un chef

  Architecture

  Correction

  Convergence

Conclusion

# Correction

Si les conditions suivantes sont vérifiées

- ▶ il y a un et un seul jeton de test
- ▶ il y a un et un seul jeton de chef
- ▶ ils sont de la même couleur

Elles le seront toujours et le chef ne bougera pas

# Plan

Introduction

Modèle

Exclusion mutuelle

Jeton

Passage

Élection d'un chef

Architecture

Correction

Convergence

Conclusion

# Convergence emboîtée

Les propriétés suivantes vont être vérifiées les unes après les autres

1. il y a un et un seul jeton de test
2. il y a un et un seul jeton de chef
3. ils sont de la même couleur

# Plan

Introduction

Modèle

Exclusion mutuelle

  Jeton

  Passage

Élection d'un chef

  Architecture

  Correction

  Convergence

Conclusion

# Résultats

Algorithmes autostabilisants pour deux paradigmes

Utilisation d'une condition topologique

Utilisation d'une composante aléatoire

Nombre fini d'état (de l'ordre plus petit non diviseur de  $n$ )

Jetons : brique de base pour construire d'autres algorithmes autostabilisants

Temps de convergence. . . mal connu, au cas par cas

## Références

Recherches réalisées au LaBRI, U. Bordeaux 1 et à l'I3S, U. Nice

- ▶ pour l'*exclusion mutuelle*



Jérôme Durand-Lose.

Randomized uniform self-stabilizing mutual exclusion.

*Information Processing Letters*, 74:203–207, 2000.

- ▶ pour l'*élection d'un chef*

en collaboration avec l'équipe *Parallélisme*, du LRI, U. Paris 12,  
Orsay



Joffroy Beauquier, Jérôme Durand-Lose, Maria Gradinariu, and  
Colette Johnen.

Token based self-stabilizing uniform algorithms.

*Journal of Parallel and Distributed Computing*, 62(5):899–921, 2002.

