Tolérance aux pannes dans les systèmes distribués

Alain BUI

Professeur

Département de Mathématiques et Informatique

alain.bui@univ-reims.fr

Introduction

- Nombre croissant de composants dans un système => Probabilité plus grande que des composants tombent en panne pendant l'exécution de l'algorithme

- Causes diverses
  - Erreurs de conception, Accidents, Malveillances etc.

- Objectif: éviter de relancer un algorithme après chaque panne => concevoir des algorithmes capables de fonctionner malgré des pannes.

- Système Distribué: panne => système est affecté partiellement et improbable qu'il le soit en totalité
- Idée de solution: les sites corrects prennent en charge les tâches des sites défaillants
- Conséquence: perte de performances mais pas de fonctionnement erroné

Définition

- erreurs => défaillances => fautes
- Un composant est défaillant s'il ne répond plus à sa spécification (composant en SD = lien ou site)
- Une faute ou panne désigne une défaillance temporaire ou définitive d'un ou plusieurs composants du système

Spécifications

- Spécifications pour les sites
  - Si un site n'a pas atteint un état final, il finira par exécuter une autre étape de l'algorithme

- Spécifications pour les liens de communications
  - Un site j reçoit un message d'un site i au plus une fois et seulement si i a précédemment envoyé le message à j
  - Si i a envoyé un message à j et j exécute infiniment des étapes de l'algorithme alors j finira par recevoir le message de i.

Algorithmes Robustes

- Robuste : Garantir la correction du comportement global du système vis à vis des spécifications de l'algorithme

  - Spécifications définies en terme d'invariants qui doivent être constamment vérifiés
  - Aucun dysfonctionnement n'est toléré pour le système
  - Algos robustes masquent les fautes
  - Approche dite *pessimiste*

Algorithmes Robustes : exemple comportemental

- Exclusion Mutuelle
  - Propriétés de sûreté et de vivacité **toujours** vérifiées
  - Par exemple, on ne se retrouvera jamais avec une configuration où 2 sites sont en même temps en SC

- Élection
  - Un et un seul site sera élu
  - Par exemple, à aucun moment il existe une configuration où simultanément plusieurs sites décident qu'ils sont élus.

Algorithmes auto-stabilisants

- **Finir** par garantir la correction du comportement global du système vis à vis des spécifications de l'algorithme

  - Système tolère certaines périodes de dysfonctionnement
  - Algorithmes ne masquent pas les fautes
  - Approche dite *optimiste*

Algorithmes Auto-stabilisants: exemple comportemental

- Exclusion mutuelle
  - Propriété de sûreté non vérifié pendant un intervalle de temps
  - Deux sites peuvent se retrouver en SC
  - MAIS au bout d'un moment le système retrouve un comportement correct (l'algorithme retrouve de lui même un état valide)

Classification des fautes

- Des critères

  - Origine de la faute
    - Type de composant : ex. lignes ou sites
  - Cause de la faute: bénignes ou malignes
    - Défaillances temporaires ou définitives : si le composant fonctionne il fonctionne correctement

    ex. ligne transmet le msg ou non / site traite le msg ou non

    - Défaillances byzantines : comportement arbitraire du composant défaillant. Si le composant fonctionne, il ne fonctionne par correctement à l'insu des autres composants.

    ex. site répond « blanc » à certains sites et « noir » à d'autres.

Classification (suite)

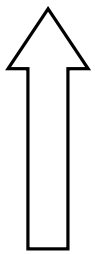
- Durée de la faute
  - Définitive
  - Temporaire
- Détectabilité de la faute
  - Détectable localement. Réparation par le site lui-même.
  - Non détectable localement. Réparation nécessite échange de messages.

- Différentes classification selon ces critères, en voici une …

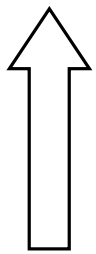
Une hiérarchie

- Sites
  - Site mort-né : site n'exécute aucune instruction de son algo.
  - Site en panne franche : site fonctionne correctement jusqu'à l'apparition de la panne et cesse totalement de fonctionner.
  - Site byzantin : comportement arbitraire.

$$\text{Mort-né} \subset \text{Panne franche} \subset \text{Byzantin}$$



Résultat d'impossibilité



Résultat de faisabilité

Typologie

- Panne franche
  - Composant fonctionne correctement puis panne et cesse immédiatement de fonctionner = panne permanente

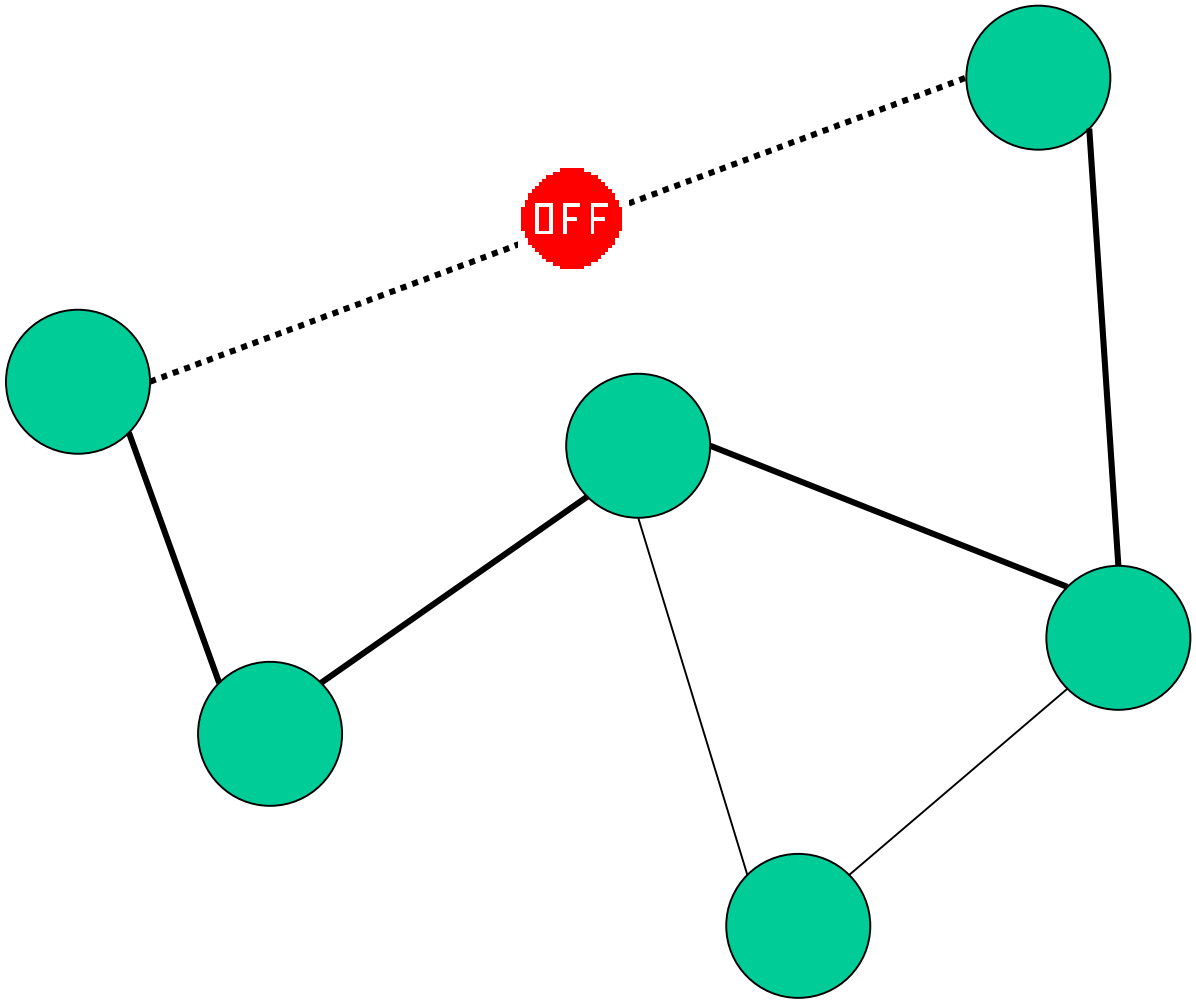
    - Panne franche de site
    - Coupure d’une ligne => changement de topologie du réseau

- Panne transitoire
  - Comportement erroné des composants pendant une certaine période. Comportement correct ensuite.
    - On peut distinguer si la panne n'apparaît qu'une fois ou plus ou moins périodiquement
    - Corruption mémoire
    - Annulation d'une transaction
    - Perte de messages sur une ligne
- Panne byzantine
  - Toute panne engendrant une comportement s'écartant des spécifications

Robustes vs Auto-stabilisants

	Fautes transitoires	Fautes définitives	Masquant
AS	OUI	NON	NON
Robuste	NON	OUI	OUI

- 2 approches complémentaires
- Choix dépend du problème à résoudre


OFF