

# Détecteurs de pannes non fiables pour les systèmes asynchrones distribués

David Baelde

sous la direction de

Franck Petit et Vincent Villain

ENS Lyon & LaRIA

# Un modèle réaliste ...

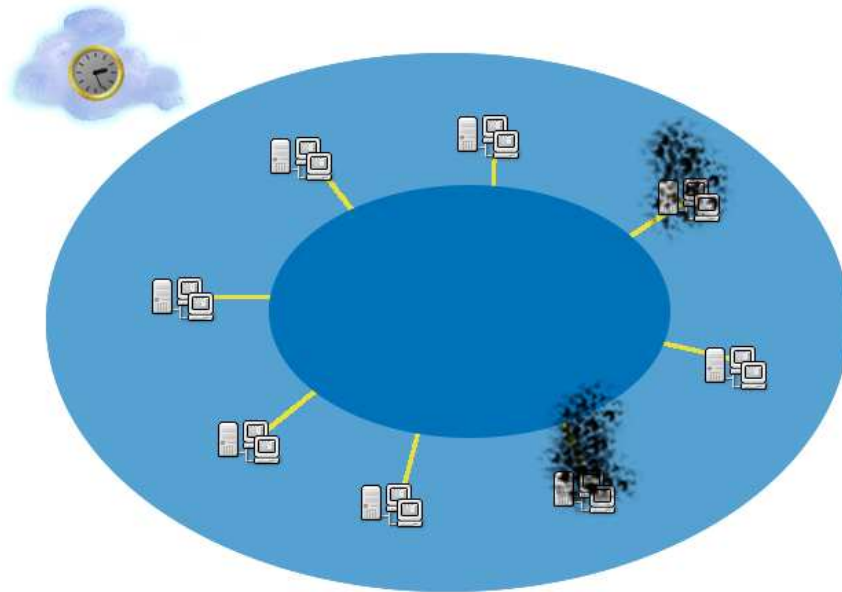
$$\Pi = \{p_1, p_2, \dots, p_n\}$$

Système **asynchrone**.

Possibilité de **panne des processus**.

Pertes d'information ?

Pour la théorie, on a une horloge globale.



# ...dans lequel on ne peut rien faire.

## Définition (Consensus)

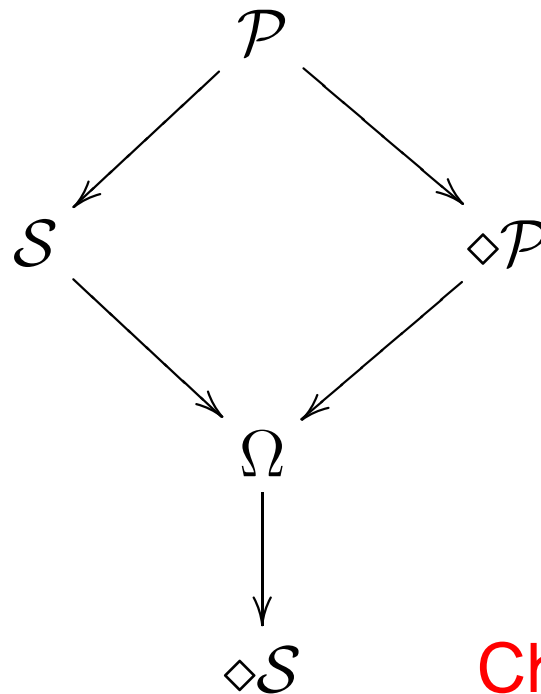
- Chaque processus correct propose une valeur.
- Tous les processus corrects en choisissent une,
- et c'est la même pour tous,
- et elle a été proposée.

## Théorème (FLP85)

Le Consensus n'est pas soluble  
dès qu'il y a la possibilité d'une seule panne.

# Que faire alors ?

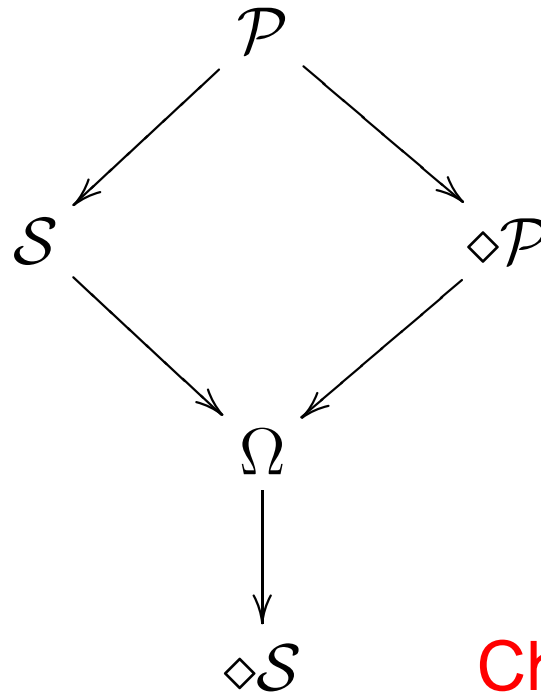
- Détecteurs de pannes non fiables : oracles distribués.



Chandra et Toueg, 1996

# Que faire alors ?

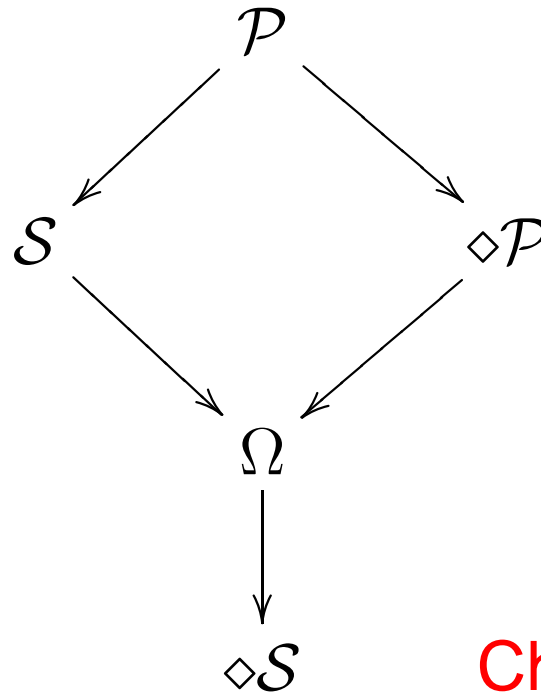
- Détecteurs de pannes non fiables : oracles distribués.
- $\Omega$  classe minimale dans  $\mathcal{E}_{\lfloor \frac{n}{2} \rfloor}$  pour le Consensus



Chandra et Toueg, 1996

# Que faire alors ?

- Détecteurs de pannes non fiables : oracles distribués.
- $\Omega$  classe minimale dans  $\mathcal{E}_{\lfloor \frac{n}{2} \rfloor}$  pour le Consensus
- L'Élection de Leader nécessite  $\mathcal{P}!!?$



Chandra et Toueg, 1996

# Objectifs

- Digérer les travaux précédents

# Objectifs

- Digérer les travaux précédents
- Vérifier les résultats de Cho et Park

# Objectifs

- Digérer les travaux précédents
- Vérifier les résultats de Cho et Park
- Comprendre, au fond

# Objectifs

- Digérer les travaux précédents
- Vérifier les résultats de Cho et Park
- Comprendre, au fond
- Étendre leur théorème

# Objectifs

- Digérer les travaux précédents
- Définir le problème
- Vérifier, regarder . . .

# Définition de Cho et Park

## Définition (Election de Leader)

- **Safety** Il n'y a jamais de désaccord sur l'identité du leader.
- **Liveness** Il finit toujours par y avoir accord de tous les processus sur un unique leader.

# Définition de l'élection de leader

## Définition (Élection de Leader)

•  $|\Pi|$  variables  $leader_p \in \{\perp\} \cup \Pi$ .

• **Safety**

$$\forall t \in \mathbb{N}, \forall p, q \in correct(F, t),$$

$$leader_p(t) = \perp \vee leader_q(t) = \perp \vee leader_p(t) = leader_q(t)$$

• **Liveness**

$$(\exists t \in \mathbb{N}, p \in correct(F, t), leader_p(t) \notin correct(F, t)) \Rightarrow$$

$$(\exists t' > t, \exists q \in correct(F, t'), \forall p \in correct(F, t'), leader_p(t') = q)$$

$\Rightarrow$  Définition de la classe  $\mathcal{L}$

# Définition alternative

## Définition (Élection de Leader booléene)

•  $|II|$  variables  $leader_p \in \{\perp, \top\}$ .

• **Safety**

$$\forall t \in \mathbb{N}, \forall p, q \in correct(F, t), \\ \neg(leader_p(t) \wedge leader_q(t))$$

• **Liveness**

$$(\exists t \in \mathbb{N}, \nexists p \in correct(F, t), leader_p(t)) \Rightarrow \\ (\exists t' > t, \exists q \in correct(F, t'), leader_q(t))$$

$\Rightarrow$  Définition de la classe  $\mathcal{L}_B$

# Résultats

- $\mathcal{L}$  et  $\mathcal{L}_B$  pas différenciés.

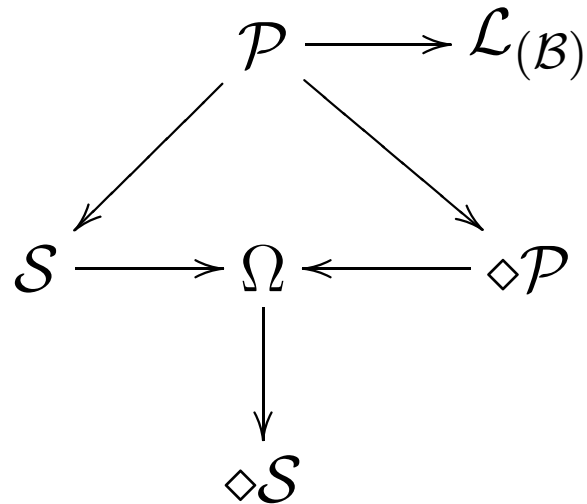
# Résultats

- $\mathcal{L}$  et  $\mathcal{L}_B$  pas différenciés.
- On a retrouvé les résultats de Cho et Park.

# Résultats

- $\mathcal{L}$  et  $\mathcal{L}_{\mathcal{B}}$  pas différenciés.
- On a retrouvé les résultats de Cho et Park.
- **Théorème (La nouvelle dimension)**

$$\neg(\mathcal{L} \succeq \diamond \mathcal{S})$$



# Stabilité

## Définition (Election stable)

C'est l'Élection de Leader, avec en plus ...

**Stabilité** Le leader ne change que s'il a crashé.

# Stabilité

## Définition (Election stable)

C'est l'Élection de Leader, avec en plus ...

**Stabilité** Le leader ne change que s'il a crashé.

- Notre preuve de  $\neg(\mathcal{L} \succeq \diamond S)$  ne tient pas.

# Stabilité

## Définition (Election stable)

C'est l'Élection de Leader, avec en plus ...

**Stabilité** Le leader ne change que s'il a crashé.

- Notre preuve de  $\neg(\mathcal{L} \succeq \diamond \mathcal{S})$  ne tient pas.
- On a clairement  $\mathcal{L} \succeq \Omega$ .
- C'est déjà mieux ...

# Encore mieux ?

## Définition (Élection désynchronisée)

- **Liveness** Si un processus correct  $r$  lance  $app\_req(i)$  à l'instant  $t$ , alors, pour tous les autres processus corrects  $p$  il existe un instant  $t_p > t$  auquel  $p$  lance  $app\_run(r, i)$ .
- **Safety** Si deux processus corrects  $p$  et  $q$  lancent  $app\_run(r, i)$ , aux instants  $t_p$  et  $t_q$ , alors  $leader_p(t_p) = leader_q(t_q)$ .

# Encore mieux ?

## Définition (Élection désynchronisée)

- **Liveness** Si un processus correct  $r$  lance  $app\_req(i)$  à l'instant  $t$ , alors, pour tous les autres processus corrects  $p$  il existe un instant  $t_p > t$  auquel  $p$  lance  $app\_run(r, i)$ .
- **Safety** Si deux processus corrects  $p$  et  $q$  lancent  $app\_run(r, i)$ , aux instants  $t_p$  et  $t_q$ , alors  $leader_p(t_p) = leader_q(t_q)$ .

## Théorème (☺)

$Consensus \succeq \mathcal{L}_S$

# Encore mieux ?

## Définition (Élection désynchronisée)

- **Liveness** Si un processus correct  $r$  lance  $app\_req(i)$  à l'instant  $t$ , alors, pour tous les autres processus corrects  $p$  il existe un instant  $t_p > t$  auquel  $p$  lance  $app\_run(r, i)$ .
- **Safety** Si deux processus corrects  $p$  et  $q$  lancent  $app\_run(r, i)$ , aux instants  $t_p$  et  $t_q$ , alors  $leader_p(t_p) = leader_q(t_q)$ .

## Théorème (☺)

$Consensus \succeq \mathcal{L}_S$

## Théorème (☹)

$\mathcal{L}_S$  soluble sans détecteur de panne !

# Fin ?

