

## Chapitre 5

### Etat global

## Etat global d'un système distribué

Sauvegarde d'une configuration du système à un moment donné :

- un état global du système
- = une configuration du système
- = ensemble des états des processus
- = un *snapshot*

Utilisation

- Retour en arrière en cas de faute
- Simulation centralisée pour prévision
- Détection d'une situation particulière (terminaison, interblocage...)
- ...

## Etat global d'un système distribué

Mécanisme :

un algorithme distribué s'exécute sur le système localement  
= un algorithme sur chaque processus

→ On instrumente cet algorithme  
= on ajoute du code localement dans les algorithmes des processus

le code ajouté a pour fonctionnalité de :

sur demande, sauvegarder l'état global du système

 Le code ajouté ne doit pas modifier le fonctionnement de l'algo original !

## Etat global : définition

un état global du système =

pour tout processus, son état  
état d'un processus  
= ensemble des valeurs de variables à ce moment

pour tout canal, son état  
état d'un canal  
= messages en cours d'envoi dans ce canal,  
avec leur ordre d'arrivée dans le canal

**Problème : un moment donné ?? → pas de temps global  
comment faire pour que  
les processus sauvegardent tous « en même temps » ??**

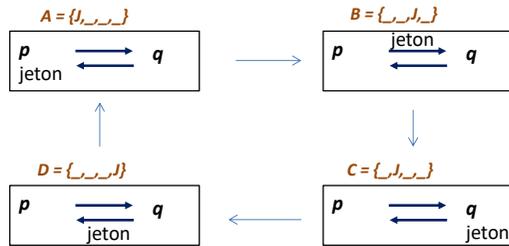
# Etat global : exemple

2 processus p et q, un jeton qui va de l'un à l'autre



État global = { Sp, Sq, Cpq, Cqp }

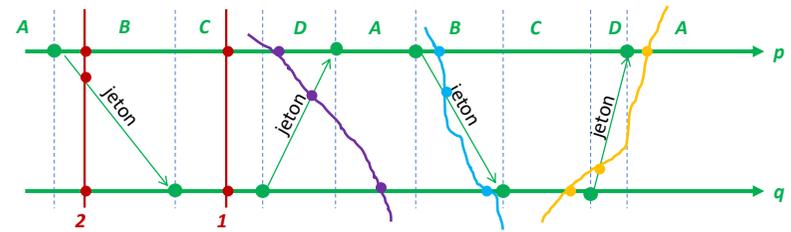
Exécution du système :



État de p = Sp  
 État de q = Sq  
 État du canal p→q = Cpq  
 État du canal q→p = Cqp

# Etat global : exemple

L'exemple a 4 états globaux, A, B, C, D



coupe cohérente : {1, 2} = B  
 coupe incohérente : {1, 2} = D  
 coupe ?? : {1, 2} = C

Comment les processus doivent ils sauvegarder ?

Chaque processus sauvegarde état et canaux localement

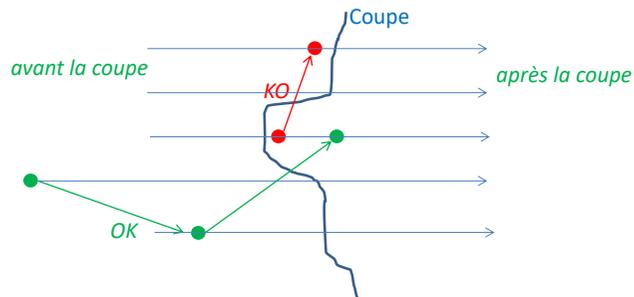
coupe idéale : 1 : {1, 2} = C  
 2 : {1, 2} = B

**Coupe** = quand sauvegarder localement chaque processus pour que globalement, Ces informations locales forment un état global du système

# Coupe cohérente

Coupe cohérente =

pour tout événement e antérieur à la coupe, s'il y a des événements qui précèdent causalement e alors ils doivent aussi être antérieur à la coupe



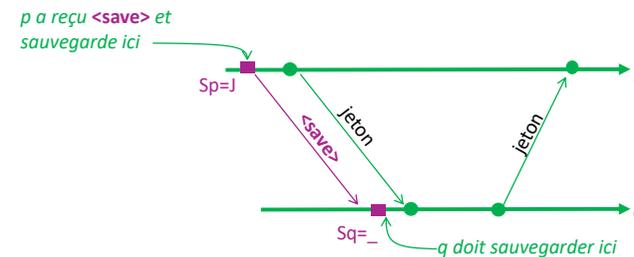
# Quand un processus doit il s'enregistrer ?

Les processus doivent s'enregistrer de telle façon que les dates de sauvegarde forment une coupe cohérente

→ sinon, l'état global formé peut ne pas représenter un état du système !!

On suppose :

- les canaux sont FIFO
- les processus s'enregistrent sur réception d'un message <save>



Comment faire ?  
 - p sauvegarde et envoie <save> à q  
 - q reçoit <save> et sauvegarde aussitôt !

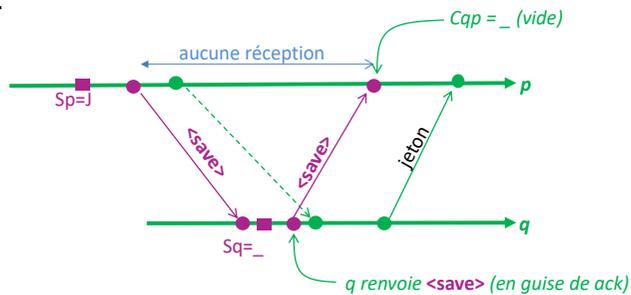
# Enregistrement des canaux

Un processus enregistre l'état de ses canaux entrants

→ p enregistre Cqp

→ Cqp va contenir les messages reçus par p depuis que q s'est enregistré

Cas 1 :



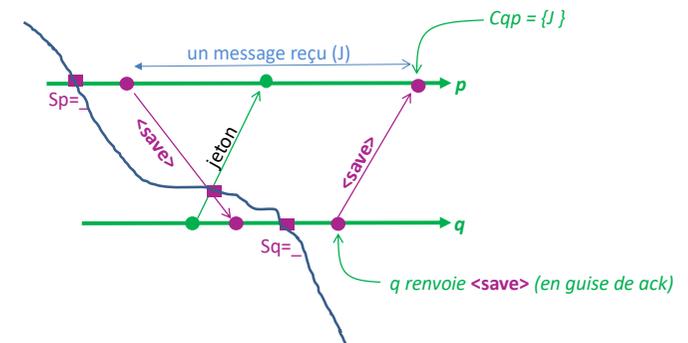
# Enregistrement des canaux

Un processus enregistre l'état de ses canaux entrants

→ p enregistre Cqp

→ Cqp va contenir les messages reçus par p depuis q s'est enregistré

Cas 2 :



# Sauvegarde d'un état global (snapshot)

## Algorithme de Shandy Lamport

Hypothèse : topologie connexe

pas de faute, tout asynchrone

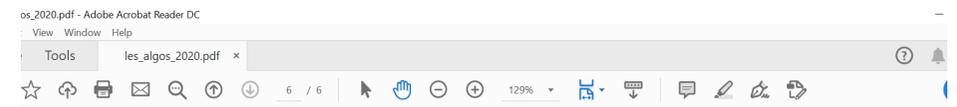
canaux FIFO

Instrumentation d'un algorithme existant.

→ Voir algorithme

### Principe :

- 1) La coupe est provoquée par des messages <save> (demande de sauvegarde)
- 2) A réception d'un message <save> par p, Sp est enregistré
- 3) p enregistre ses canaux Cqp (messages arrivant sur p) entre le moment où Sp est enregistré et le moment où p reçoit un message <save> de q par lequel p est informé de la sauvegarde de Sq.



### Algorithme 8 Algorithme de Shandy Lamport - Snapshot

Modification d'un algorithme A pour qu'il effectue une sauvegarde de son état global.

On suppose que l'algorithme A pour le processus p est de la forme :

- 1: <init>
- 2: Pour toujours
- 3: Réceptionner (M) de q
- 4: ...
- 5: Fin Pour toujours

Variables supplémentaires pour A :

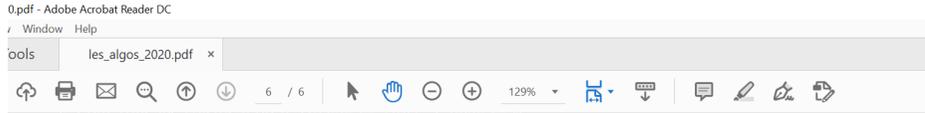
- 1: save : tableau de booléens indicés sur p et ses voisins.
- 2: Cqp : tableau indicé sur les voisins de p, Cqp[q] contient le contenu du canal de q vers p, une fois la sauvegarde faite.

En plus de l'initialisation <init> de A :

- 1: save[p] ← faux
- 2: Pour tous les voisins q de p, save[q] ← faux; Cqp[q] ← ∅

Sur demande de sauvegarde :

1. enregistrer Sp



Sur demande de sauvegarde :

- 1: enregistrer  $S_p$
- 2:  $save[p] \leftarrow vrai$
- 3: Envoyer  $\langle SAVE \rangle$  à tous les voisins de  $p$

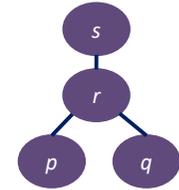
Dans la boucle "Pour Toujours" de l'algorithme  $\mathcal{A}$  :

- 1: Si  $M = SAVE$  alors
- 2: enregistrer  $C_{qp}[q]$ ;  $save[q] \leftarrow vrai$
- 3: Si non  $save[p]$  alors
- 4: enregistrer  $S_p$
- 5:  $save[p] \leftarrow vrai$
- 6: Envoyer  $\langle SAVE \rangle$  à tous les voisins de  $p$
- 7: Fin Si
- 8: Si  $save[p]$  et pour tout  $q$ ,  $save[q]$  alors enregistrement fini FinSi
- 9: Fin Si

Dans la boucle "Pour Toujours" de l'algorithme  $\mathcal{A}$ , ajout pour tous les autres types de messages :

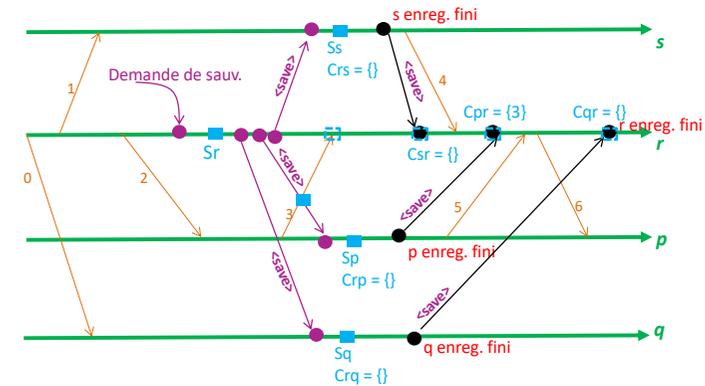
- 1: Si  $M \neq SAVE$  et  $save[p]$  et non  $save[q]$  alors
- 2:  $C_{qp}[q] \leftarrow M + C_{qp}[q]$
- 3: Fin Si

# Algorithme de snapshot : exemple



Algorithme à instrumenter

La demande de sauvegarde arrive par le processus  $r$



# Algorithme de snapshot : correction ?

**Lemme A :** En temps fini, l'algorithme termine :

*suite à une demande,*

*il y a un nombre fini de messages échangés qui conduisent à la*

*sauvegarde locale des processus et des canaux*

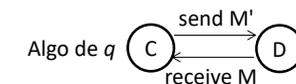
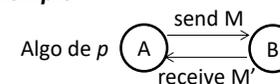
Preuve au tableau

**Lemme B :** La coupe formée par les événements de sauvegarde est cohérente.

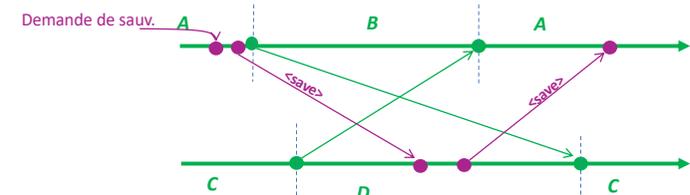
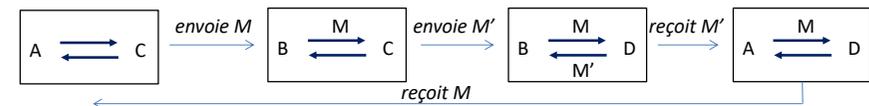
Preuve au tableau

# Propriété de l'état global enregistré

Exemple :



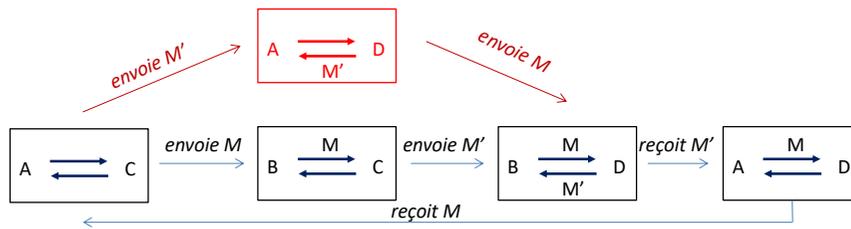
Une exécution :



Etat enregistré :  
 $S_p = A$   
 $S_q = D$   
 $C_{pq} = \{ \}$   
 $C_{qp} = \{ M' \}$

## Propriété de l'état global enregistré

L'état enregistré n'est pas atteint par la trace !



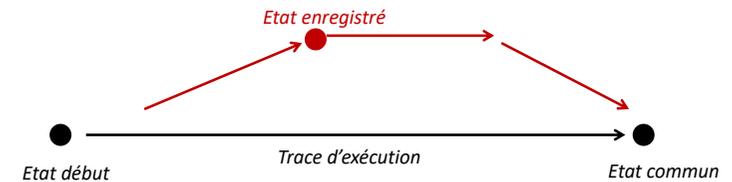
Il aurait pu l'être si les 2 envois étaient inversés (d'abord M' puis M)...

→ L'état global n'est pas forcément atteint par l'exécution, mais aurait pu l'être si des événements concurrents avaient lieu dans un autre ordre...

## Propriété de l'état global enregistré

**Admis :**

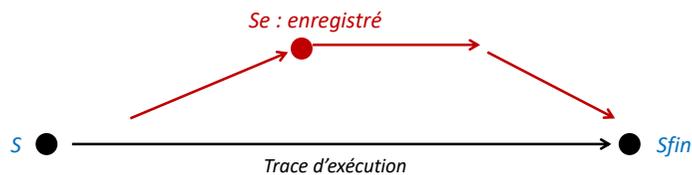
Au bout d'un temps fini,  
la trace fictive enregistrée (= celle atteinte par l'état global enregistré)  
et la trace d'exécution  
vont atteindre un même état global



## Propriété stable

Une propriété d'un état global est stable  
si quand elle devient vraie, elle le reste pour toujours

Exemple : terminaison interblocage



Si la propriété stable  $P$  est vraie en  $Se$  alors elle est encore vraie en  $Sfin$

→ En détectant  $P$  sur  $Se$ , on a la garantie que la trace va atteindre un état où  $P$  est vraie.

NB: si  $P$  n'est pas vraie sur  $Se$ , on ne peut rien dire !!