



# GESTION DE TRANSACTIONS DISTRIBUEES

# INTRODUCTION

- Les objectifs du traitement de transactions distribuées sont les mêmes que ceux des systèmes centralisés, avec plus de complexité du fait que le SGBDD doit assurer l'atomicité des transactions globales et de toutes les sous-transactions composantes.

# Plan du chapitre

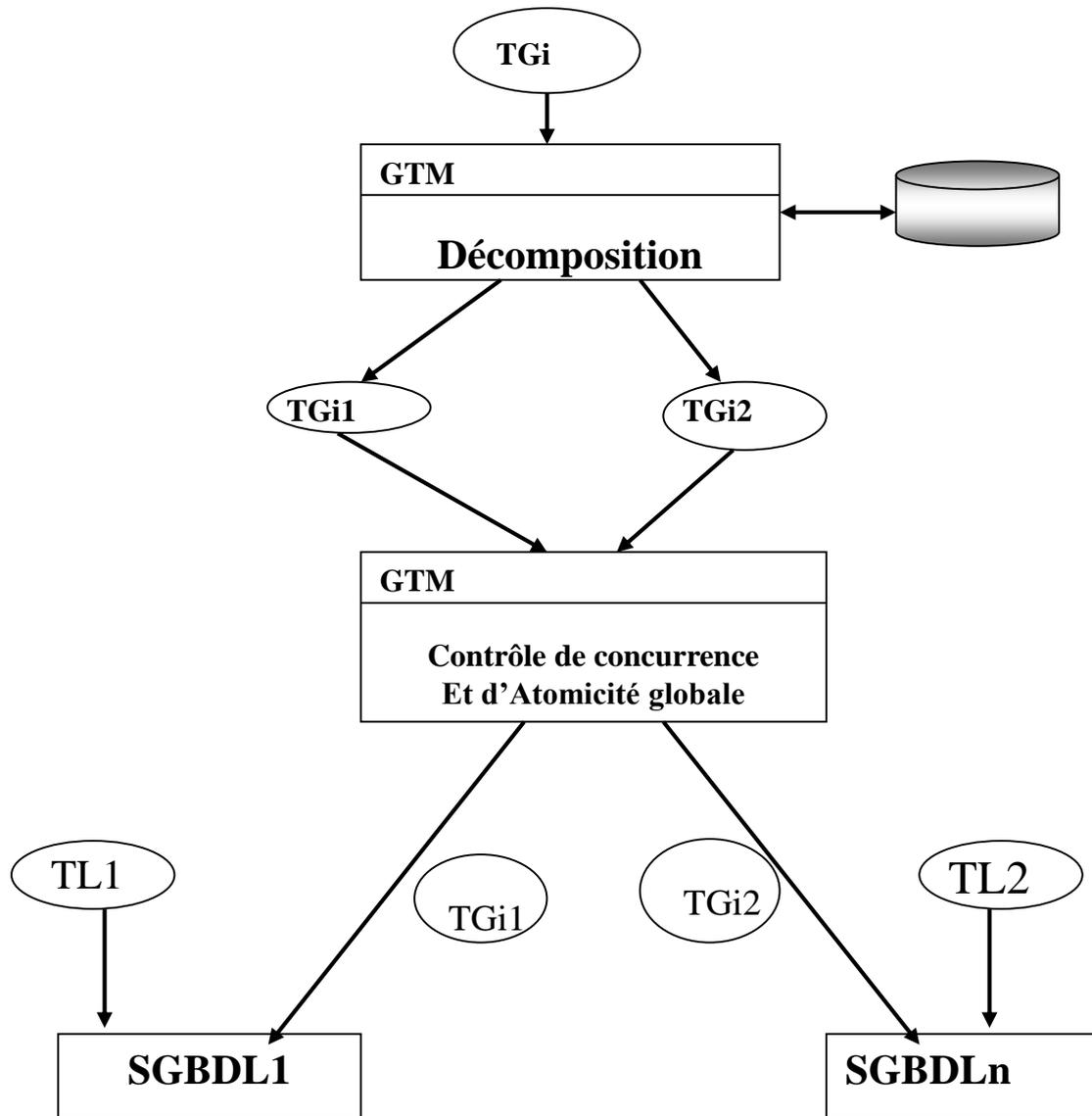
- Généralités
- Protocoles d'estampillage
- Protocoles de verrouillage
- Récupération de base de données distribuées
  - Protocoles de Validation



# Généralités

# Types de Transactions

- **Transaction locale (TL)** : initiée par les applications locales et soumises directement au SGBD local.
- **Transaction globale (TG)** : initiée par les applications globales et soumises via l'interface du Système multibase de données (SMBD). Le SMBD a une composante qui assure la gestion de transactions globales appelé « Gestionnaire de Transactions Global » soit le **(GTM)**.



# Types d'ordonnancements

- Niveau local et global des transactions distribuées → deux niveaux d'ordonnement : Local et Global
- **Ordonnement Local**
  - L'exécution concurrente de sous-transactions globales et de transactions locales dans chaque SGBDL est modélisée par un ordonnement local. La détermination de l'ordre local (ordre total) d'exécution des actions est le point central des Pbs de gestion de transactions car l'autonomie des SGBDL rend difficile la communication de cet ordre au SMBD.
- **Ordonnement Global**
  - L'intégration des exécutions des différents SGBDLs composant la base de données distribuées est modélisée par un ordonnement global. L'ordre global est l'union des ordres relatifs aux ordonnements locaux.

# Notion de Conflits

Le GTM doit gérer deux types de conflits :

- **Conflits classiques** (ou directs) entre transactions globales ; c'est-à-dire l'accès à une même donnée par des actions de deux transactions différentes.
- **Conflits indirects** causés par l'entrelacement des transactions globales avec les transactions locales de chaque site et que le GTM ne peut détecter vu que les TLs s'exécutent en dehors de son contrôle.

# SERIALISABILITE

- **Sérialisabilité globale**
  - Une exécution globale est sérialisable si elle produit les mêmes résultats et possède les mêmes effets sur la BD qu'une **exécution globale série** des mêmes transactions.
- **Une exécution globale série**
  - Une exécution globale dans laquelle pour toute paire de transactions (TG1 avant TG2), les actions de TG1 sont exécutées avant les opérations de TG2.
- **Un ordonnancement global est dit globalement sérialisable** si et seulement si, il existe un ordre total défini sur les transactions globales validées qui est consistant avec l'ordre de sérialisation des transactions validées sur chaque SGBDL.

# Problèmes liés à l'autonomie

- **La plupart des problèmes de gestion de la concurrence dans les bases de données distribuées proviennent du fait du maintien de l'autonomie des SGBDLs.**
- Certains chercheurs ont proposés des approches sacrifiant cette autonomie en soumettant les TLs au contrôle du SMBD au même titre que les TGs.
- D'autres ont proposés de relâcher le contrôle global et de garder au niveau local un ordonnancement à deux phases strict.
- **Un ordonnancement global est localement sérialisable si l'ordonnancement local de chaque SGBDL est sérialisable.**



# Protocoles d'estampillage

# GESTION DES ACCES CONCURRENTS

## 1. Les Techniques d'estampillage

- L'ordre de sérialisation est obtenu par association d'une estampille unique à chaque transaction
- La gestion des estampilles peut être centralisée ou distribuée
  - La gestion centralisée implique un compteur ou horloge locale sur un site
  - Gestion décentralisée : 2 paramètres sont nécessaires

<estampille, identifiant du site>

- Dans ce dernier cas, il y a le problème de synchronisation

# Exemple

- *Soit un site  $S1$  avec une horloge  $H1$*
- *$S1$  reçoit un message estampillé  $\langle t, s2 \rangle$  avec  $t$  valeur ( $H1$ )*
- *Alors  $S1$  « avance »  $H1$  à  $t+1$*
  
- **Algorithme d'estampillage**
  - Soit  $i$  estampille de transaction et  $j$  estampille de granule
  - $T_i$  accède à un granule estampillé  $j$
  - Si  $j \leq i$
  - Alors  $T_i$  peut s'exécuter sinon **ABORT**( $T_i$ ) et reprise ultérieure avec  $e > j$
  - Finsi



# Protocoles de verrouillage

# Protocole de verrouillage

- **Plusieurs versions :**
  - **V2P centralisé, V2P à copie primaire, V2P distribué, Verrouillage à la majorité**
- **V2P centralisé**
- Un seul site conserve et entretient les informations de verrouillage: pour une transaction globale amorcée au site S1, les actions suivantes sont exécutées:
  1. Le coordinateur de transaction du site S1 scinde la transaction en un certain nombre de sous-transactions à l'aide du catalogue système global
    - **Mise-à-jour:** le coordinateur demande des verrous exclusifs sur toutes les copies, effectue la mise-à-jour puis libère les verrous
    - **Lecture :** le coordinateur élit une des copies pour la lecture, généralement celle de son propre site.

# Protocole de verrouillage

## V2P centralisé

2. les gestionnaires de transactions locaux impliqués dans la transaction globale demandent, puis libèrent les verrous exigés par le gestionnaire de verrouillage centralisé suivant les règles du verrouillage 2phases.
3. le gestionnaire de verrouillage centralisé vérifie la compatibilité d'une demande de verrou sur une donnée avec les verrous actuellement appliqués. Si c'est le cas, le gestionnaire de verrouillage retourne un message au site à l'origine de la demande, en confirmant le verrou, sinon il place la requête dans une file d'attente jusqu'à ce que le verrou puisse être accordé.

# Protocole de verrouillage V2P centralisé

- **Avantages**

- mise en place relativement simple,
- Un seul gestionnaire de verrouillage traite toutes les informations

- **Les coûts de communication sont relativement faibles : si n sites  $2n + 3$  messages**

- 1 demande de verrouillage
- 1 message d'accord de verrouillage
- n message de mise-à-jour
- n confirmations
- 1 demande de libération de verrou

- **Inconvénients**

- la centralisation peut impliquer des problèmes d'embouteillage et de réduction de fiabilité;
- panne du site central entraîne des pannes systèmes majeures

# Protocole de verrouillage V2P à copie primaire

- Amélioration du V2P centralisé en répartissant les gestionnaires de verrouillage sur un certain nombre de sites:
- chaque gestionnaire de verrouillage est responsable de la gestion des verrous sur un sous-ensemble de données. Pour chaque donnée dupliquée une copie est choisie comme **copie primaire**, les autres devenant **copies esclaves**.
- Lorsqu'une donnée doit être mise-à-jour, le coordinateur de transaction doit déterminer où se trouve la copie primaire afin d'envoyer la demande de verrouillage au gestionnaire de verrouillage adéquat. Seule la copie primaire est verrouillée exclusivement, dès qu'elle est mise-à-jour, celle-ci est propagée sur les copies esclaves

# Protocole de verrouillage V2P distribué

- Répartition des gestionnaires de verrouillage sur chacun des sites. En l'absence de toute réplication , ce protocole est équivalent au V2P à copie primaire:
- **Avantage:** prise en charge des verrous de manière décentralisée
- **Inconvénients:** gestion plus complexes des verrous et coûts de communication plus élevés:  $5n$  messages
  - $n$  messages de verrouillages
  - $n$  messages d'accord de verrouillage
  - $n$  messages de mise-à-jour
  - $n$  confirmations
  - $n$  demandes de libérations de verrous

# Protocole de verrouillage

## Verrouillage à la majorité

- Extension du V2P distribué, en évitant le verrouillage de toutes les copies d'une donnée dupliquée avant une mise-à-jour.
- Quand une transaction souhaite lire ou écrire une donnée dupliquée sur  $n$  sites, elle doit expédier une demande de verrou à plus de la moitié des  $n$  sites où la donnée est entreposée. La transaction ne peut poursuivre tant qu'elle n'a pas obtenu le verrou sur une majorité de copies.
- Nombre de messages
- $\lceil (n+1)/2 \rceil$  messages de demandes de verrouillages
- $\lceil (n+1)/2 \rceil$  messages de demandes de déverrouillages

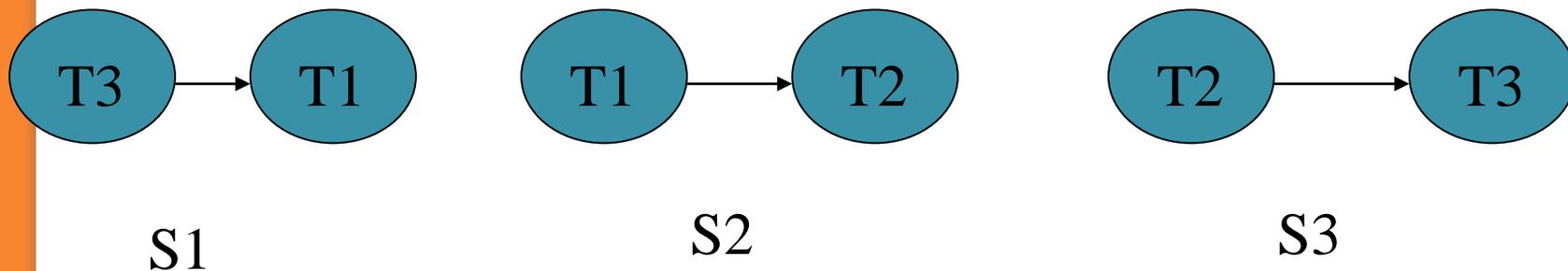
# Interblocage

Soit trois transactions T1, T2, T3

- T1 démarre au site S1 et crée un agent au site S2
- T2 est amorcée au site S2 et crée un agent au site S3
- T3 est amorcée au site S3 et crée un agent au site S1

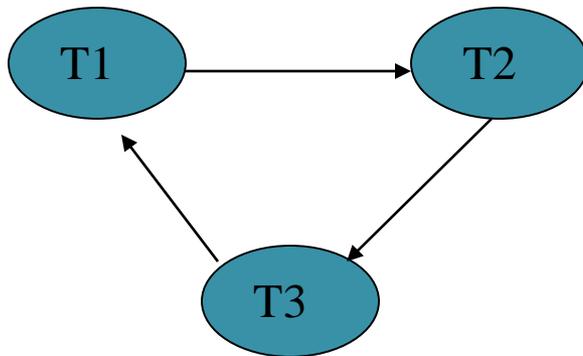
Temps	S1	S2	S3
T1	S-lock (T1,x1)	S-lock (T2,y2)	X-lock (T3,z3)
T2	X-lock(T1,y1)	X-lock (T2,y2)	
T3	X-lock(T3,x1)	X-lock(T1,y2)	X-lock (T2,z3)

- T1 → T2 → T3 → T1



Graphes d'attente locaux

# Interblocage



Graphe d'attente  
global

Dans un contexte distribué, il ne suffit pas de construire le graphe d'attente propre à chaque site, mais il faut également construire un graphe d'attente global = union des graphes d'attente locaux.

Trois méthodes de détection existent

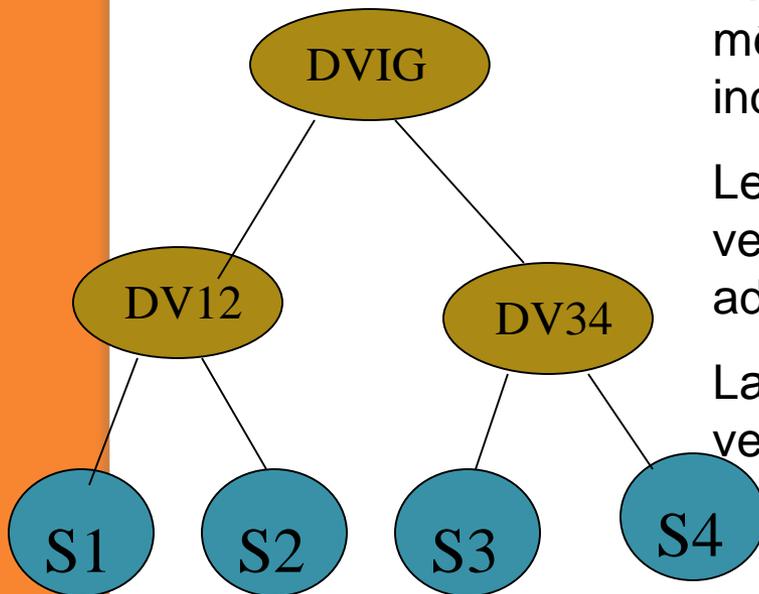
Détection **centralisée**; **hiérarchique**; **distribuée** de l'interblocage

# Détection centralisée de l'interblocage

- Un seul site est désigné comme le coordinateur de détection de verrou indéfini (CDVI)
- il a la responsabilité de construire et d'entretenir le graphe d'attente global;
- il vérifie la présence de cycles:
- il a la possibilité de briser le cycle en annulant les transactions sélectionnées.
- Le CDVI doit prévenir tous les sites concernés par l'annulation des transactions.

# Détection hiérarchique de l'interblocage

- Les sites du réseau sont structurés en une hiérarchie. Chaque site envoie son graphe d'attente local au site de détection de verrou indéfini juste au dessus de lui dans la hiérarchie



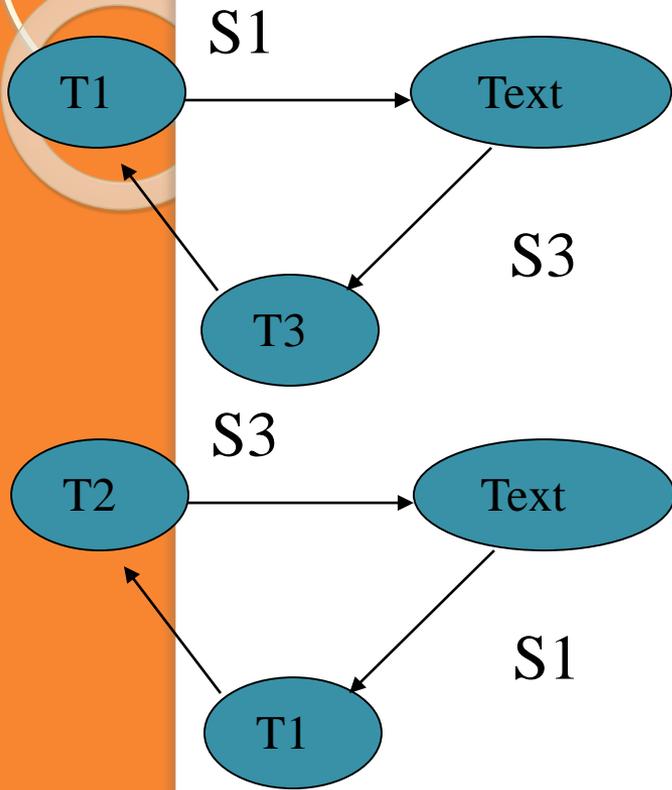
Les feuilles de niveau 1 sont les sites eux mêmes où se produit la détection de verrou indéfini

Les nœuds  $Dv_{ij}$  de niveau 2 détectent les verrous indéfinis qui impliquent des sites adjacents  $i$  et  $j$

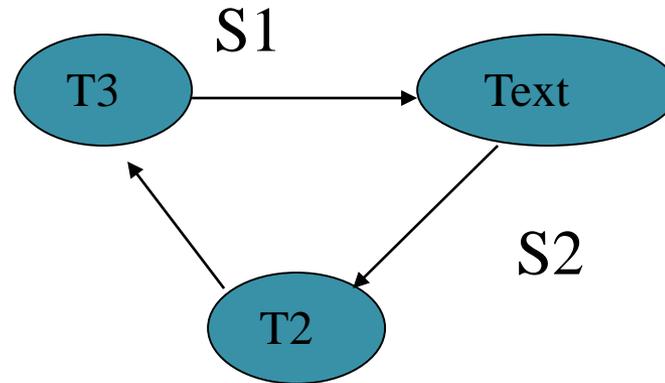
La racine de l'arborescence est le détecteur de verrou indéfini global

***L'approche hiérarchique réduit la dépendance d'un site de détection central et minimise les coûts de communication***

# Détection distribuée de verrou indéfini



- un nœud extérieur est ajouté au graphe pour faire référence à un agent d'un site distant
- quand une transaction T1 du site S1 crée un agent à un autre site S2 alors un arc est tracé de T1 vers Text.



Le graphe d'attente local du site S1 est transmis au site que T1 attend c.-à-d. S2

Le graphe d'attente local du site S2 est étendu : Text  $\rightarrow$  T3  $\rightarrow$  T1  $\rightarrow$  T2

\ Text

## Détection distribué de verrous indéfinis

- Il contient encore un verrou indéfini, et le graphe est transmis au site que T2 attend à savoir S3: le graphe d'attente local au site S3 devient: Text  $\rightarrow$  T3  $\rightarrow$  T1  $\rightarrow$  T2  $\rightarrow$  T3  $\rightarrow$  Text
- Ce graphe d'attente contient un cycle donc existence d'un verrou indéfini et un protocole de récupération adéquat est nécessaire.
- Inconvénient essentiel : communications intersites

# Récupération de bases de données distribuées

- Comme dans le cas de la récupération locale, la récupération répartie a pour principe de maintenir l'atomicité et la durabilité des transactions distribuées.
- Pour s'assurer de l'atomicité d'une transaction globale, le SGBD doit s'assurer de ce que les sous-transactions de la TG soient valides ou s'annulent toutes.
- Le SGBD doit suivre les étapes suivantes :
  - Annuler toute transaction affectée par la défaillance
  - Affubler le site défaillant d'un indicateur empêchant tout autre site de s'en servir
  - Vérifier périodiquement si le site défaillant a retrouvé ses facultés ou si le site diffuse un message indiquant qu'il a récupéré
  - Au redémarrage le site défaillant doit amorcer une procédure de récupération pour annuler toute transaction partielle active au moment de la panne
  - Après la récupération locale, le site doit mettre à jour sa copie de la BD pour lui restituer sa cohérence par rapport au reste du système.

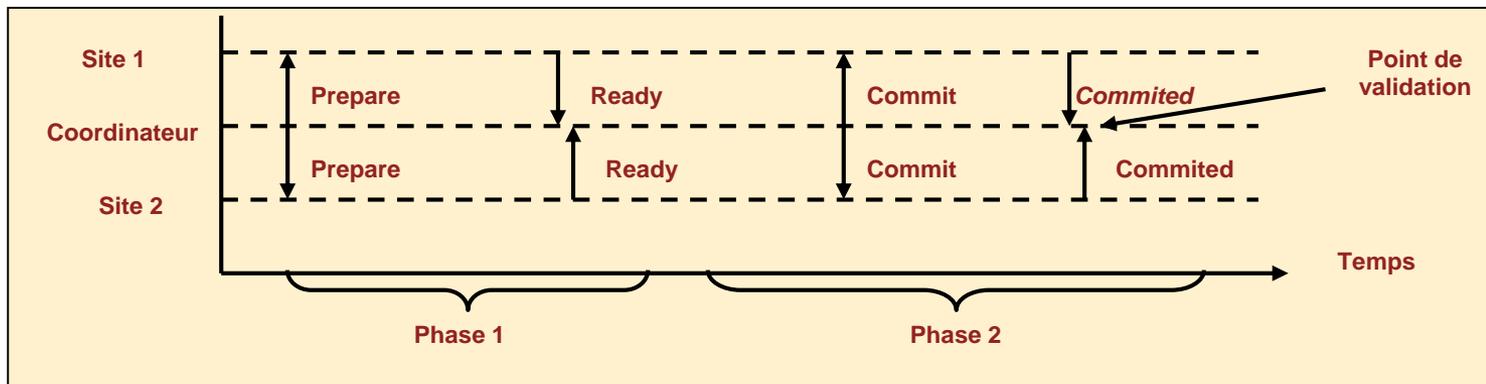
# Le Protocole de Validation à deux étapes : 2PC

## Déroulement d'une transaction dans un SGBD distribué

- L'étape la plus importante dans le gestion d'une transaction dans un contexte distribué, est celle de *la validation*. Elle consiste dans l'intégration définitive des mises à jour dans la base de données.
- Le protocole le plus utilisé, pour assurer l'atomicité d'une transaction globale, est *le protocole de validation en deux étapes*
- Ce protocole, est exécuté sur chaque site ; il est contrôlé par le site maître **Coordinateur** ou *gestionnaire de transaction*.
- Les sites où la transaction globale a des agents sont appelés les **participants** où gestionnaires de ressources.
- Nous supposons que le coordinateur connaît l'identité de tous les participants et que chaque participant connaît l'identité du coordinateur, mais pas nécessairement tous les autres participants.

# Le Protocole de Validation à deux étapes : 2PC

- Le 2PC opère en deux phases
  - Une phase de vote
  - Une phase de décision



- Validation Réussite

# Phases de validation

## ***La préparation***

- Le coordinateur demande à tous les participants (aux sites ) concernés par la transaction, de se préparer pour la validation
  - envoi d'un message '**se\_préparer**' à tous les participants et attendre la réponse dans un délai déterminé (Ready).

## ***La validation***

- le Coordinateur ordonne aux sites participants à la transaction- la validation de leurs mises à jour (**commit**)
- il attend que tous les participants confirment la validation dans un temps d'attente déterminé (**committed**)..
- Dès que toutes les confirmations ont été reçues, écriture d'un enregistrement 'fin\_transaction dans le journal'.

***Remarque*** : le protocole nécessite la journalisation des mises-à-jour préparées et des états des transactions dans un journal local à chaque site.

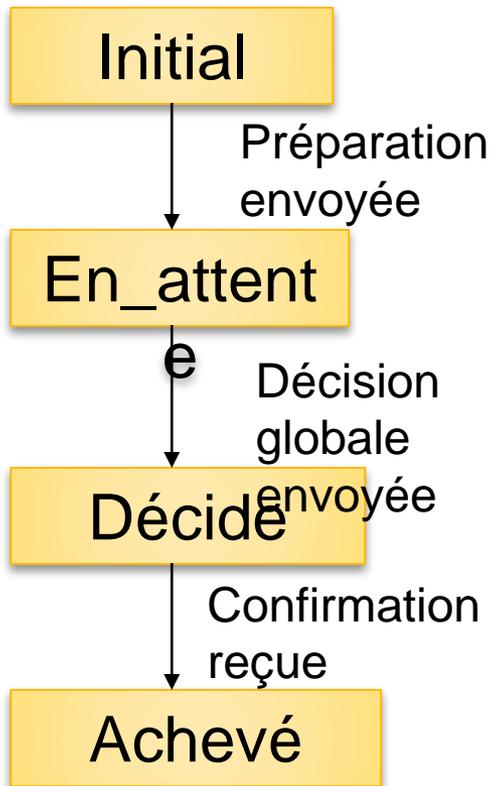
# Phases de Validation

Coordinateur	Participant
Validation Ecrire début_validation dans le journal Envoyer se_préparer à tous les participants	Se préparer Ecrire Ready dans le journal Envoyer Ready au coordinateur Attendre validation globale ou annulation globale
valider (Commit) Si tous les participants ont voté Ready Ecrire Commit dans le journal Envoyer validation globale à tous les participants : Validation globale Attendre les confirmations	Validation globale Attendre les confirmations Ecrire commit dans le journal  Valider la transaction
Confirmation (committed) Si tous les participants ont confirmé Envoyer confirmation Ecrire fin_transaction dans le journal	

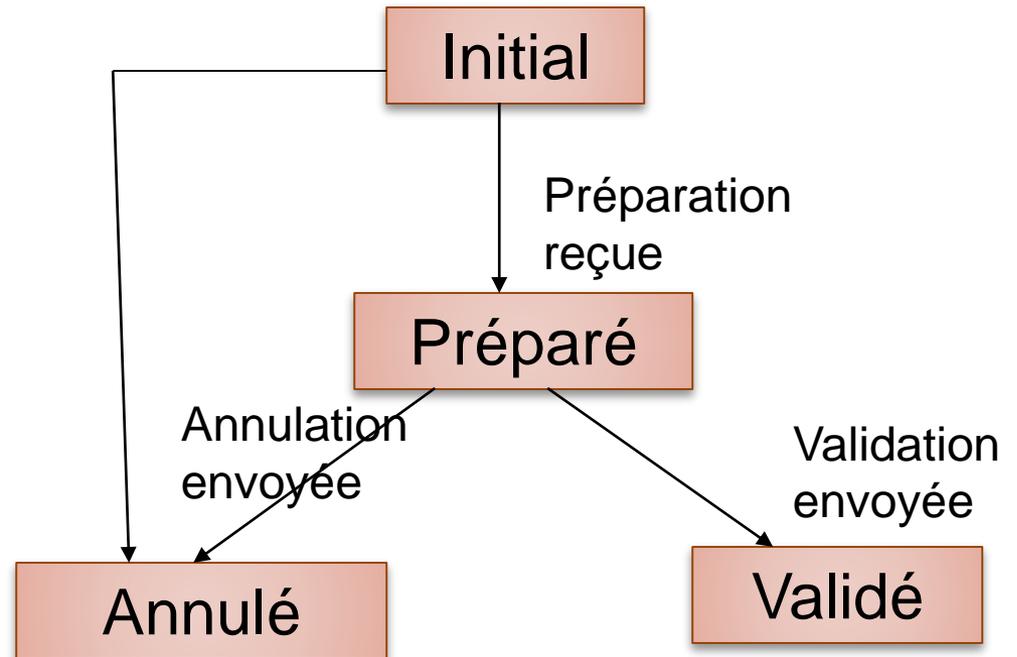
# Cas D'annulation

Coordinateur	Participant
<p>Validation</p> <p>Ecrire <i>début_validation</i> dans le journal</p> <p>Envoyer <i>se_préparer</i> à tous les participants</p>	<p>Se préparer</p> <p>Attendre les réponses</p> <p>Ecrire annulation dans le journal</p> <p>Envoyer annulation au coordinateur</p> <p>Annuler la transaction</p>
<p>Annulation</p> <p>Si un participant a voté Abort (annuler)</p> <p>Ecrire Annulation dans le journal</p> <p>Envoyer Annulation globale à tous les participants</p> <p>Attendre les confirmations</p>	

# Diagramme d'états-transitions du 2PC



Le coordinateur



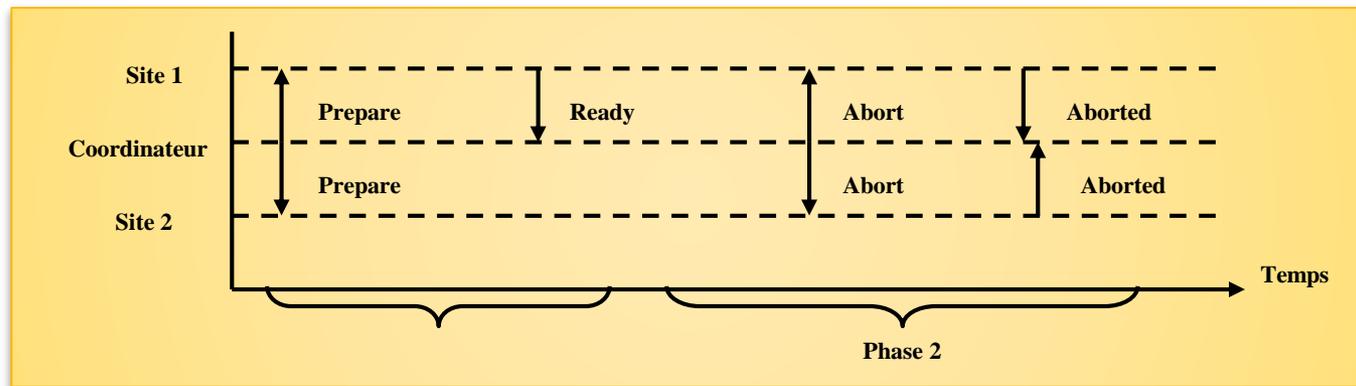
Le participant

# Gestion des Pannes

- Les cas de pannes se présentent sous trois formes
- **Panne du coordinateur** lui-même
- dans ce cas, les sites participants qui se sont déclarés prêts à valider sont bloqués, d'où le blocage des ressources jusqu'à la reprise de coordinateur.

## **Panne d'un des sites ou rupture de liaison**

- ces deux cas sont traités de manière similaire, si les accusés *Ready* des sites ne parviennent pas au coordinateur dans un délai précis ; celui ci considère que la réponse est négative et leur envoie une annulation (*Abort*).



- Panne d'un site

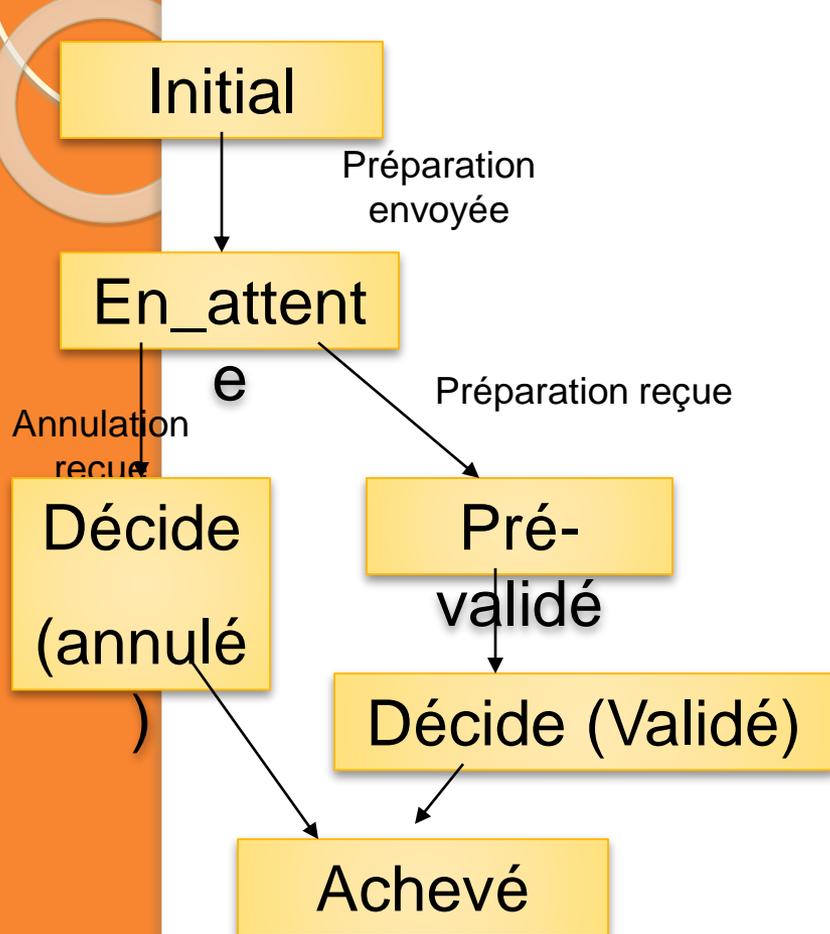
# Gestion des Pannes

- Quand le site *coordinateur* est en panne, les participants prêts à valider sont bloqués, on dit alors que le protocole à *deux étapes* est *bloquant*.
- Afin de résoudre ce problème, le protocole à deux étapes est augmenté d'une nouvelle étape de préparation, et appelé *protocole de validation en trois étapes*.
- On rajoute entre l'état « prêt » et l'état « valider » un état appelé « prêt à valider ». Si le coordinateur est en panne, et qu'aucun participant n'a reçu le message « prêt à valider », alors la sous transaction est abandonnée afin de libérer les ressources.

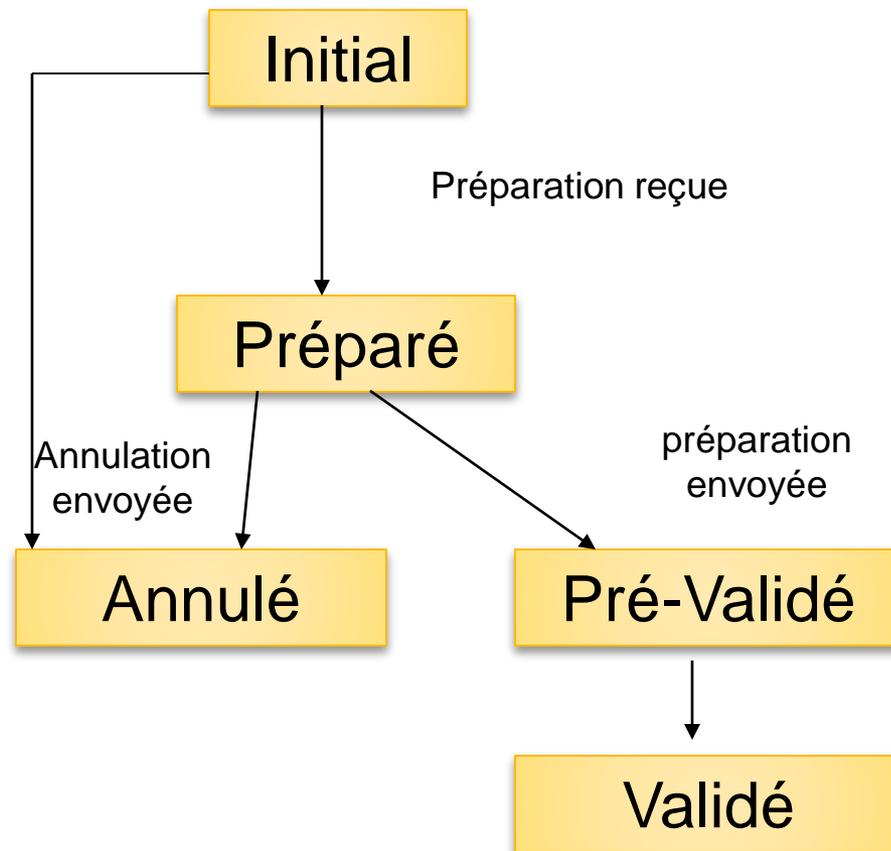
# Protocole de Validation en trois étapes: 3PC

- Tous les processus opérationnels sont informés de la décision globale de valider grâce au message prêt\_valider avant que le premier processus ne valide.
- Si le coordinateur défaille, les sites opérationnels peuvent communiquer entre eux et déterminer s'il faut valider ou annuler la transaction sans attendre que le coordinateur se rétablisse. (Utilisation d'un protocole d'élection pour élire un nouveau coordinateur)
- Si aucun des sites opérationnels n'a reçu de message prêt\_valider alors ils avortent la transaction.

# Diagramme d'états-transitions du 3PC



Le coordinateur



Le participant