École Doctorale 2008/2009

Systèmes de Bases de Données Avancés

5. Parallélisme et distribution

École nationale Supérieure d'Informatique

Plan

Concepts préliminaires

Traitement parallèle des requêtes

Transactions et recouvrement

Replication

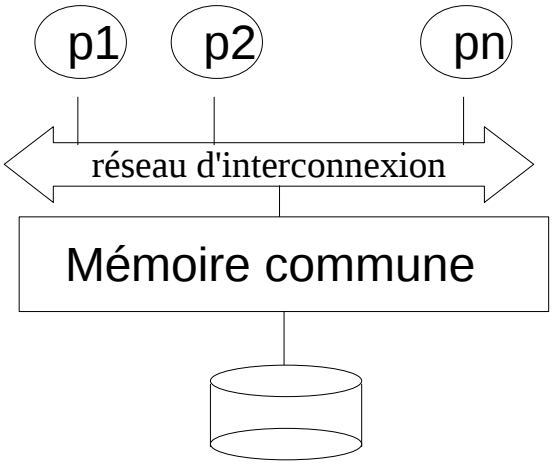
5. Parallélisme et distribution

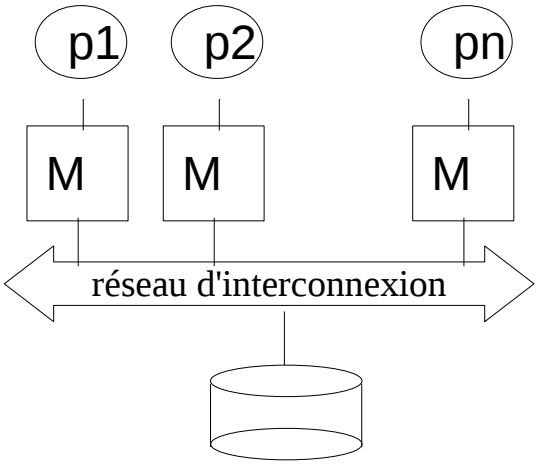
5.1. Concepts préliminaires

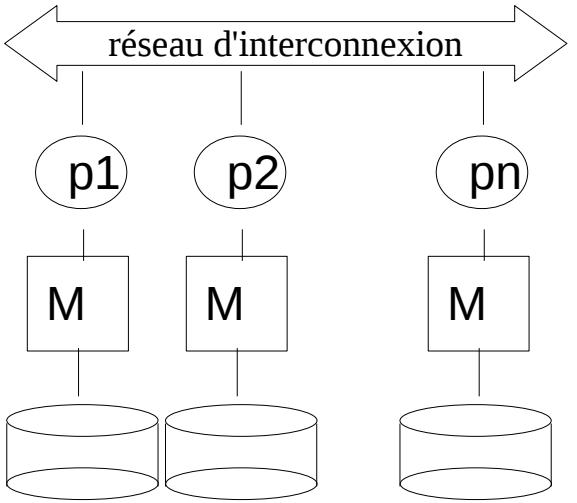
BD Parallèle vs Répartie

- Système de BD Parallèle
  - SGBD implémenté sur une machine parallèle
    - Mémoire partagée, Disque partagé, Mémoire distribuée, Hybride, Numa, ...
  - Objectif principal : Les performances
- Système de BD Réparti (distribué)
  - Un ensemble de SGBD reliés entre eux
    - Homogènes/Hétérogènes – Degré d'autonomie
  - Objectif principal : La disponibilité

Architectures parallèles







Mémoire partagée

++
Facilité d'utilisation

--
Extensibilité réduite

Disque partagé

++
Equilibrage de charge

--
Cohérence du cache

Sans partage

++
Extensibilité

--
Equilibrage de charge

Critères de performances

- Temps de réponse
  - Temps d'exécution moyen d'une tâche
- Débit
  - Nombre de validations en 1s
- Le « speed up »
  - Accélération induite par une extension en ressources
- Le « scale up »
  - Adaptation à une monté de charge

Quelques remarques

- Problèmes liés au parallélisme
  - le déséquilibre de charge
  - l'interférence
  - l'initialisation des tâches
- Facteurs favorisant le parallélisme
  - l'influence du modèle relationnel
  - l'apport des MMDB

5.2. Requêtes parallèles

Evaluation parallèle

- Requête => GFD
  - GFD : Graphe de Flot de Données
- Les sommets du GFD sont des opérateurs de base
  - Scan, Join, Sort, Split, …
- Les arcs du GFD représentent des « flots » de données entre opérateurs
  - Matérialisée
  - Pipeline

Types de parallélisme

- Inter Requêtes

  (Augmente le débit)

  - Plusieurs requêtes en parallèle
- Intra Requête

  (Diminue le temps de réponse)

  - Inter Opérateurs
    - Indépendant
    - Pipeline
  - Intra Opérateurs

Fragmentation de données

- Fragmentation Horizontale
  - Généralisée (par restrictions)
  - Partitionnement « Round Robbin »
  - Partitionnement Par hachage
  - Partitionnement Par intervalle
- Fragmentation Verticale
  - Par sous-schémas
- Fragmentation Hybride
- Duplication des fragments

Exemple

Relations de base

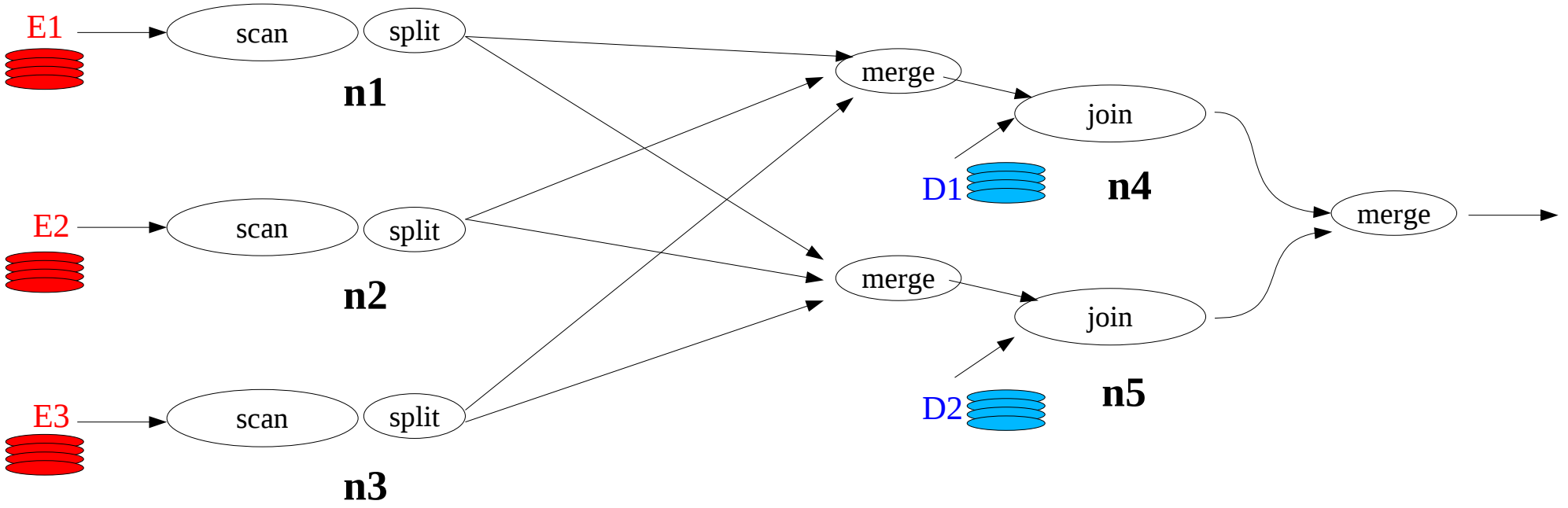
E(nss, nom, fonct, dept, ...)
D(numDept, nomDept, ...)

Fragments (par intervalle)

E1 (nom < 'gzzzz')	sur le noeud n1
E2 (nom >= 'gzzzz' et < 'nnnn')	sur le noeud n2
E3 (nom >= 'nnnn')	sur le noeud n3
D1 (numDept < 200)	sur le noeud n4
D2 (numDept >= 200)	sur le noeud n5

```
Select * From E, D Where E.dept=D.numDept  and
        E.fonct = 'directeur'
```

Exemple : GFD



Scan : fonct='directeur' Split : dept < 200 --> D1
dept >=200 --> D2

Join : E.dept = D.numdept'

Select * From E, D Where E.dept=D.numDept and E.fonct = 'directeur'

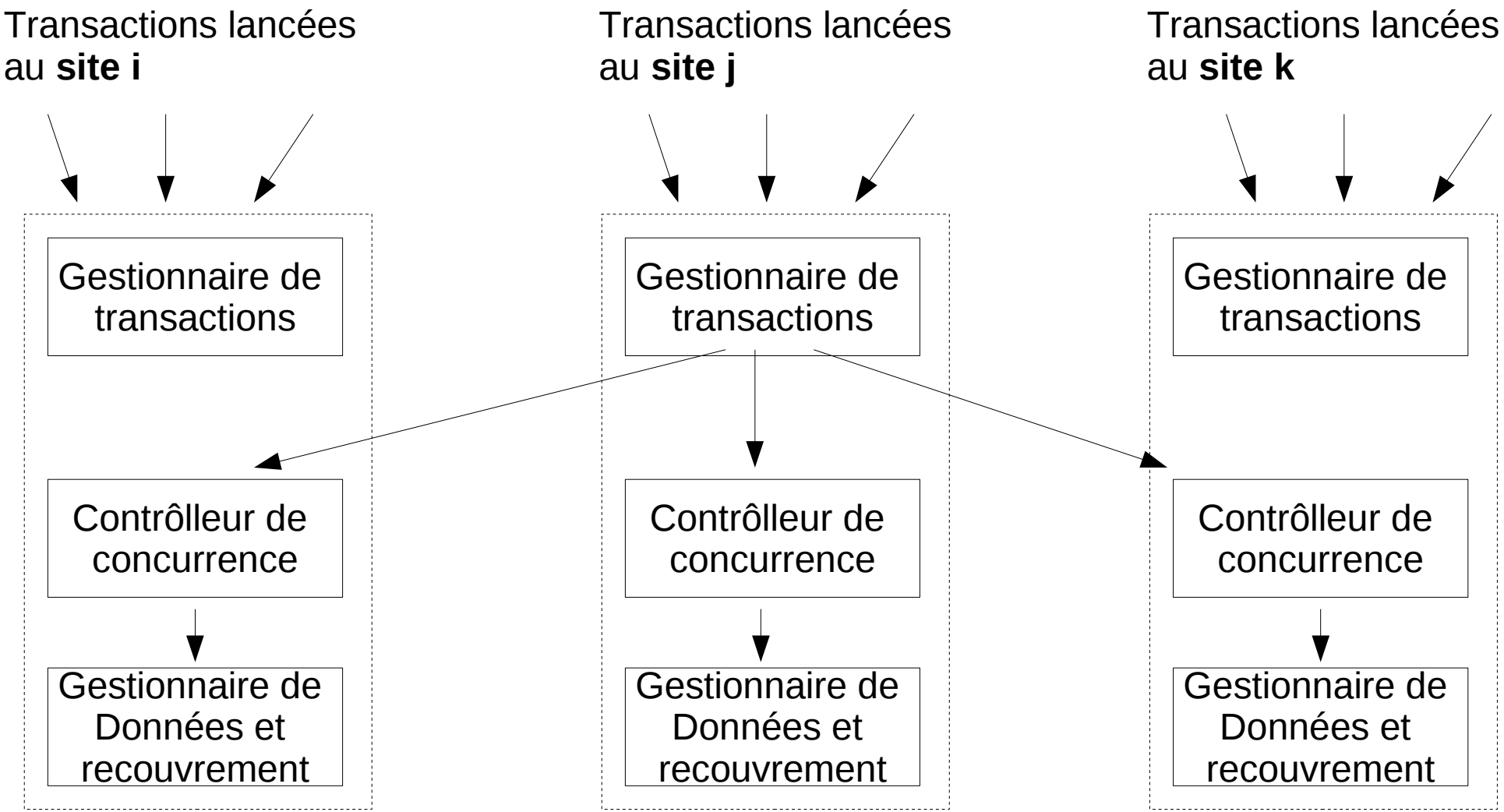
5.3. Transactions réparties

Transactions réparties

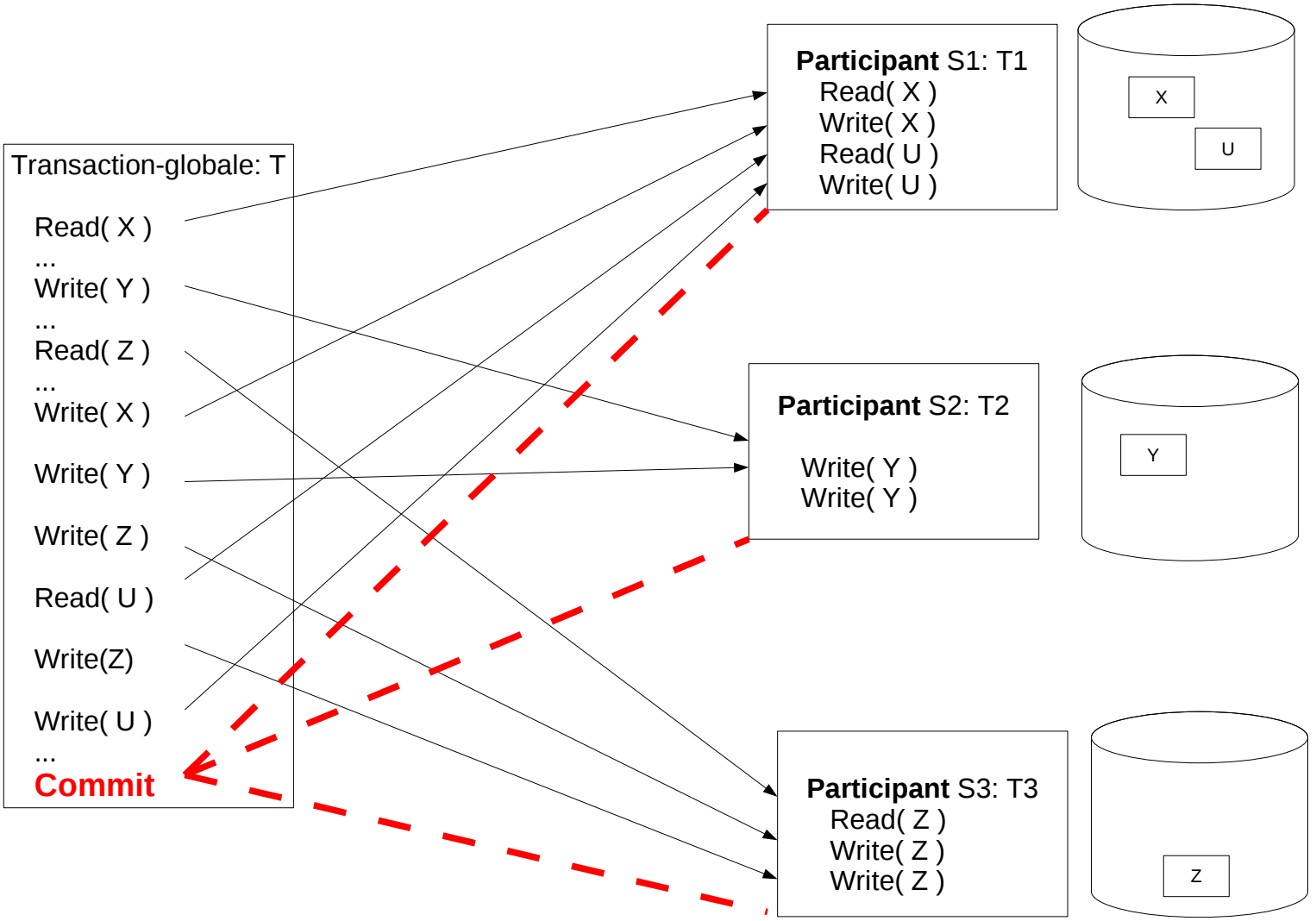
- Transaction manipulant des données réparties sur plusieurs noeuds

- Extensions du modèle centralisé
  - Identification globale des transactions
  - Contrôle de concurrence réparti
  - Validation atomique

Système transactionnel réparti



Branches d'une transaction



Validation atomique

- Protocole assurant la prise d'une décision commune à tous les participants à une transaction répartie
  - Soit tous valident, soit tous annulent
- Initié à la fin d'une transaction globale
  - Par le gestionnaire de transaction du site principal de la transaction
- Doit résister aux pannes
  - De sites et de communications

Principe général du 2PC

« Two phases Commit » protocol

Protocole de validation en 2 phases :

- **Première phase**
  - Le coordinateur demande un vote des participants
    - Sur la possibilité de valider la transaction
  - Les participants répondent « oui » ou « non »
- **Deuxième phase**
  - Le coordinateur « décide » et informe les participants qui appliquent la décision
    - si tous les votes sont à « oui », Décision = 'Valider'
    - sinon Décision = 'Annuler'

Principe de base du 2PC

- Tout participant à une transaction répartie, peut annuler unilatéralement sa branche sans demander l'avis des autres
  - À condition qu'il n'a pas encore voté « oui »
- Un participant à une transaction répartie ne peut valider sa branche que si tous les autres vont procéder de même
  - La validation est un consensus entre les participants

Résistances aux pannes

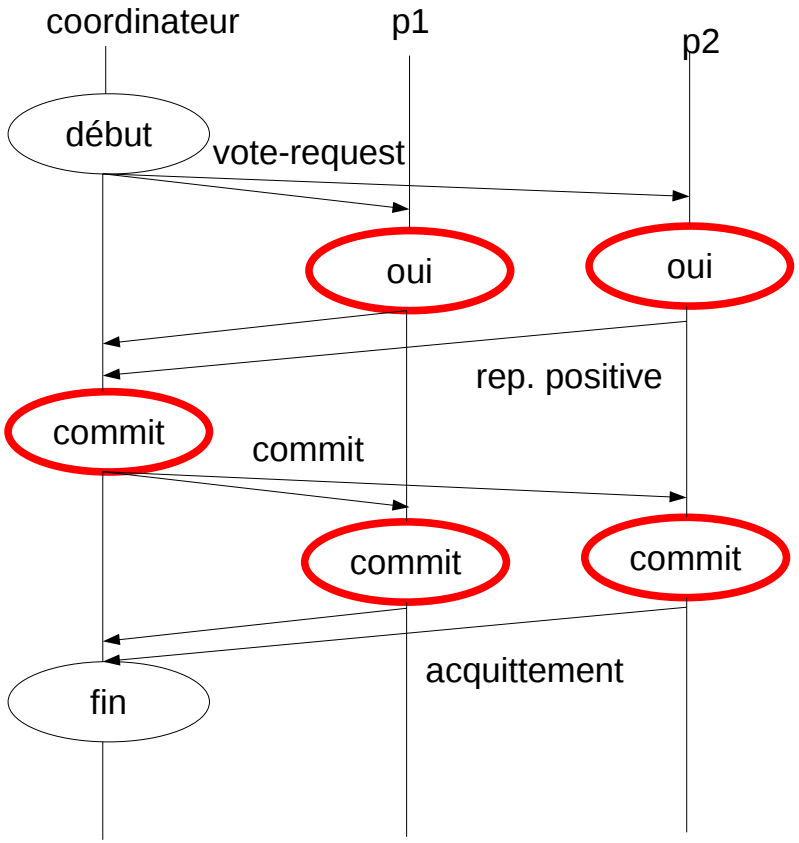
- Durant le déroulement du protocole, certaines informations doivent être écrites sur le journal

<Debut2PC, T, Liste_participants>	Par le coordinateur
<Vote_OUI, T, Liste_participants>	Par un participant
<Commit, T>	Par tous
<Abort, T>	Par tous
<Fin2PC, T>	Par le coordinateur

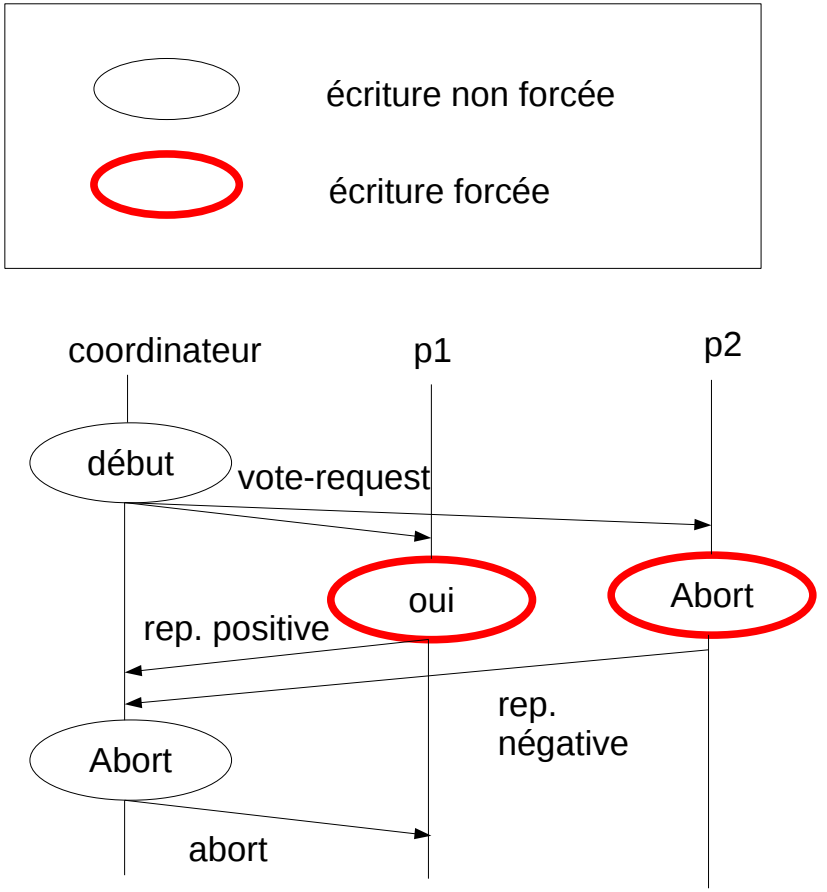
- Dans certains cas de pannes, le 2PC est Bloquant

Presumed-Abort 2PC

version standardisée du 2PC



a) Cas de la validation



b) Cas de l'annulation

5.4. Replication

Propagation des mises à jour

- Replication synchrone
  - Propagation des m-a-j avant le « commit »
  - Consistante
  - Coûteuse
- Replication asynchrone
  - Propagation des m-a-j retardée après le « commit »
  - Produit de l'inconsistance
  - Efficace

Concurrence et replication

- Protocole ROWA (synchrone)
  - « Read Once, Write All »
- Primary Copy (asynchrone)
  - Pour chaque donnée, il y a un site « maître » qui accepte les m-a-j
  - Les autres copies (secondaires) sont utilisées pour les lectures seulement
- Par « quorum » (synchrone)
  - Read R copies, Write W copies
  - R+W > N et 2W > N (N le nombre de copies)