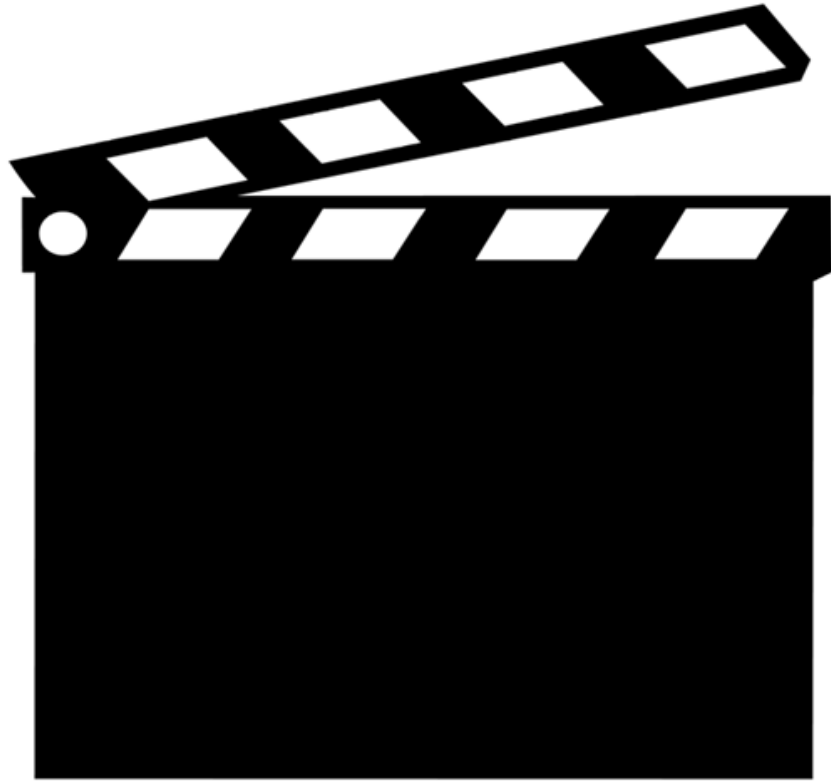


C018SA-W6-S6



SEMAINE 6 : Bases de données distribuées

1. Introduction
2. Différentes architectures
3. Fragmentation
4. Optimisation de requête
5. Réplication
- 6. Concurrence**
7. Conclusion : cinq tendances

Sans réplication – 2PL

Etendre ce qui marche dans le cas centralisé
(voir Concurrency et Transaction)

La notion de sérialisabilité s'étend facilement

Le **verrouillage à deux phases** (2PL) aussi

- Ordonnanceur local sur chaque machine
- Verrous locaux

Difficulté – Détection de deadlocks

Détection de cycles dans le graphe « WaitFor »

- Gestion centralisée ou pas de ce graphe

Timeout

L'extension d'estampillage au cas distribué
est aussi possible

Et avec réplication ?

- Méthode incorrecte mais efficace
- Méthode correcte mais coûteuse

Réplication asynchrone (incorrecte)

Les copies ne sont mises-à-jour que de temps en temps

Les copies peuvent avoir des valeurs différentes – pas de maintien de la cohérence

Bien meilleures performances

Permet beaucoup plus d'autonomie entre les sites

Et une méthode correcte

- Sur une copie, l'ordonnancement doit être équivalent à un ordonnancement sériel sur une base de données à une seule copie, et
- Toutes les copies doivent être identiques

Problème avec la réplication

Entité x dupliquée sur site_1 et site_2

Deux transactions :

- T1: $\text{read}(x)$; $x:=x+5$; $\text{write}(x)$; commit
- T2: $\text{read}(x)$; $x:=x*10$; $\text{write}(x)$; commit

Deux ordonnancements

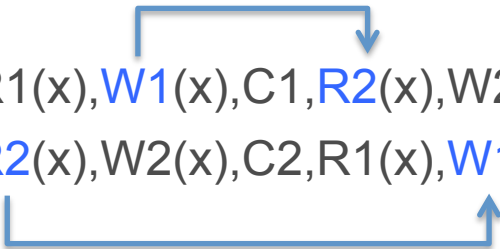
- Site_1 : $R1(x), W1(x), C1, R2(x), W2(x), C2$
- Site_2 : $R2(x), W2(x), C2, R1(x), W1(x), C1$

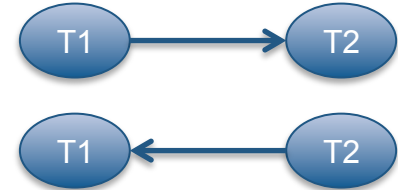
Supposons que au départ

- Avant $x@s1 = x@s2 = 1$
- Après $x@s1 = 60$ and $x@s2 = 15$
- Incohérence entre les deux copies

La sérialisabilité ne fonctionne plus

Chacun des deux ordonnancements est sérialisable (sériel même)

- Site₁ : R1(x), W1(x), C1, R2(x), W2(x), C2
 - Site₂ : R2(x), W2(x), C2, R1(x), W1(x), C1
- 



Problème : deux opérations conflictuelles
apparaissent dans des ordres différents
dans les deux sites

Sérialisabilité revisitée

1. L'ordonnancement doit être sérialisable sur chaque site
2. Deux opérations conflictuelles doivent apparaître dans le même ordre sur tous les ordonnanceurs où elles apparaissent

Question : comment garantir ça ?

Une technique ROWA

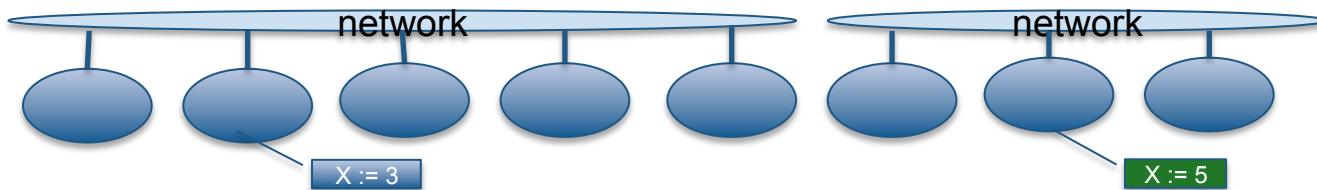
ROWA : Read-once/write-all

- Read(x) traduit en `read(x@s)` sur une copie `x@s` de `x`
- Write(x) traduit en `write(x@s)` pour toutes les copies `x@s` de `x`

Si une copie échoue, cela bloque la transaction

Alternative

- Write sur toutes les copies disponibles
- À condition d'avoir une majorité de copies disponibles



Se protéger contre les partitionnements du réseau

Pourquoi ROWA marche ?

Deux opérations conflictuelles doivent inclure au moins un *write*

Les *write* sont dans le même ordre (l'ordre défini par la majorité)

Très efficace quand il y a beaucoup plus de *read* que de *write*

Merci

Serge Abiteboul

