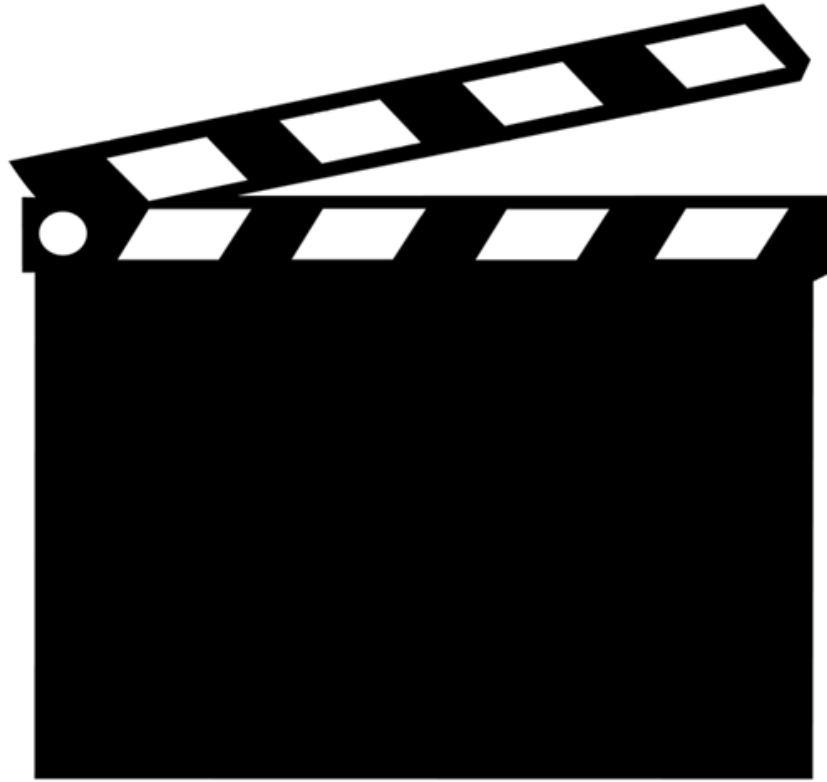


C018SA-W6-S1



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

Bases de données distribuées

- Les données d'une application sont de plus en plus souvent distribuées entre plusieurs SGBD/serveurs
- Dans cette séquence, on explique
 1. pourquoi on distribue des données, et
 2. comment on réalise une telle distribution.

Systeme distribue

Un **systeme distribue** est une application qui coordonne les actions de plusieurs ordinateurs pour realiser une tache particuliere

Le **calcul distribue** a une composante importante de gestion de donnees :

- Les donnees de l'application
- Les echanges de donnees entre les ordinateurs (communications)
- Les etats du systeme et des sessions des utilisateurs
- Les profils des utilisateurs...

Base de données distribuées

Une **base de données distribuées** : une grande quantité de données résidant sur plusieurs machines

Un **système de gestion de données distribuées** : un logiciel qui permet d'avoir un point d'entrée unique sur une base de données distribuées

Contradiction apparente

- Les calculs et données sont décentralisés
- Un **point d'entrée unique** pour accéder à toutes les données de manière transparente

Base de données distribuées (suite)

Fragmentation

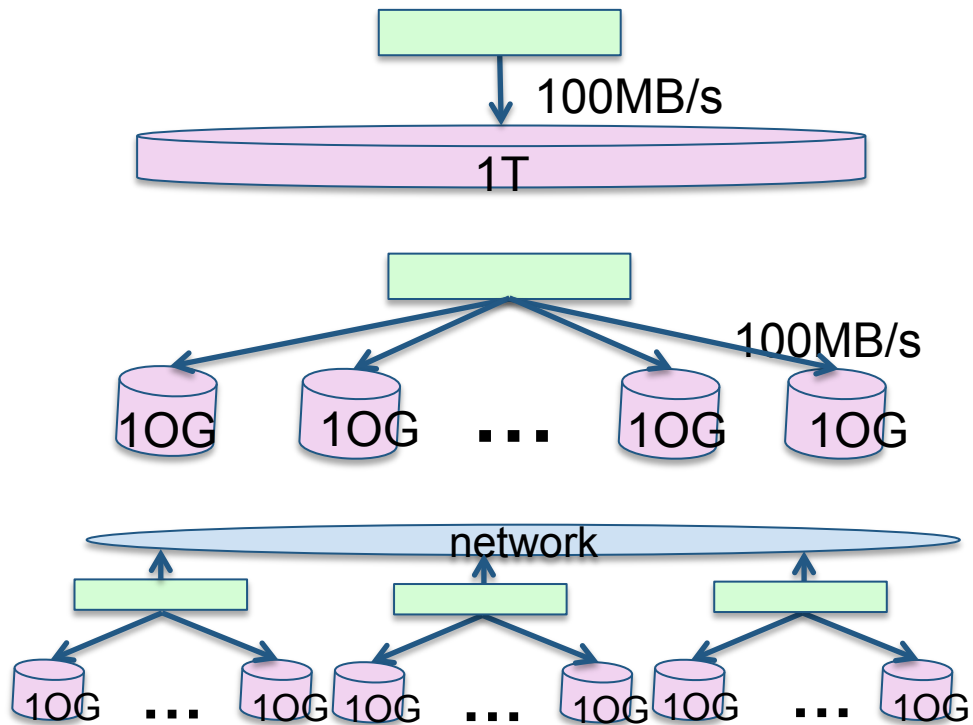
- On part d'une grosse base de données
- On la distribue en fragments pour améliorer les performances
- On obtient une base de données distribuées

Intégration

- On part de bases de données existantes (autonomes)
- On les intègre pour obtenir un point d'entrée unique
- On obtient une base de données distribuées

Réplication de données

Pourquoi fragmenter : performance



Un ordinateur et un disque

- 1 téraoctet
- Scan séquentiel
- 166 minutes (> 2.5 heures)

100 disques en parallèle

- moins de 2mn

100 ordinateurs distribués

- Chacun son propre CPU
- Chacun son disque
- **Ça passe à l'échelle**

Pourquoi intégrer : conserver l'autonomie

Autonomie des différents systèmes

Permet

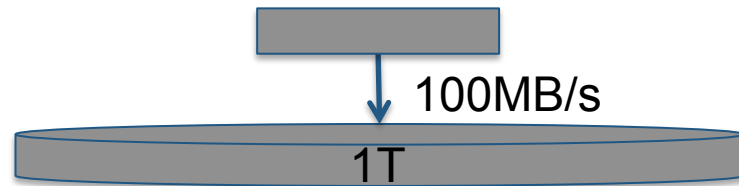
- L'intégration de données provenant d'organismes indépendants (par ex, entreprises indépendantes)
- L'intégration de données provenant de différentes branches d'un même organisme mais en laissant plus de liberté à chaque branche

Tendance dans l'industrie : des structures de moins en moins hiérarchiques, de plus en plus d'autonomes

Pourquoi répliquer : sûreté

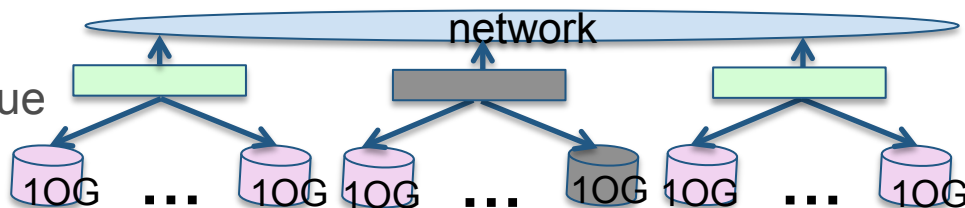
Mono-serveur

- Panne du processeur ou du disque
- Arrêt du service
- Perte des données



Multiserveurs

- Panne d'un processeur ou d'un disque
- Le service continue
- Grâce à la réplication,
pas de perte des données



Enormément utilisé

- Réplication par 2 : banque
- Réplication par 3 : airbus

Concept essentiel : transparence

de localisation

- l'utilisateur n'a pas savoir où sont les données

de réseau/système

- il n'a pas à connaître les aspects techniques du réseau

de fragmentation

- il n'a pas à connaître comment les données sont fragmentées

de réplication

- il n'a pas à savoir comment/si elles sont répliquées

Chaque entité a un nom unique

Pas seulement des avantages

Complexité

- Rajoute des couches de logiciel
- Communication
- Distribution du contrôle
- Conception de la base de données
- Gestion des contraintes d'intégrité

Absence de standards

Moins d'expérience

Incohérences

- Que fait-on quand deux systèmes donnent des informations contradictoires
- Exemple : réservation de place
- Exemple : collision d'avion

Merci

Serge Abiteboul

