

Base de données distribuées



Babacar Diop

Dpt. d'Informatique

UFR des Sciences Appliquées et de Technologies

Université Gaston Berger de Saint-Louis

2018/2019

Architecture

Architecture des BDD

Concepts

- Architectures développées en fonction de 3 paramètres
 - **Distribution** - Indique la distribution physique des données sur les différents sites
 - **Autonomie** - Indique la répartition du contrôle du système de base de données et le degré de fonctionnement indépendant de chaque SGBD constitutif
 - **Hétérogénéité** - Il s'agit de l'uniformité des modèles de données, des composants système et des bases de données

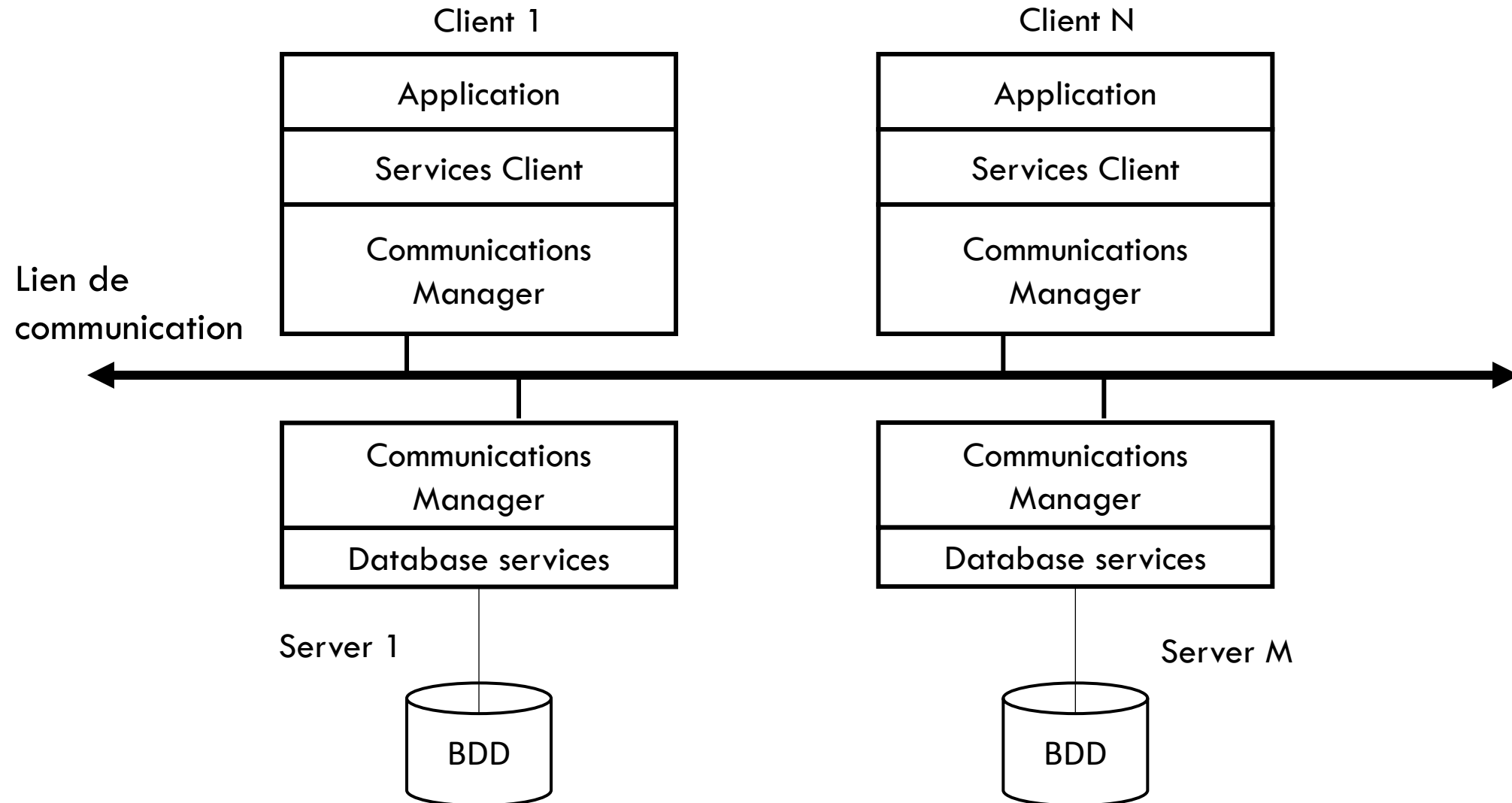
Architecture des BDD

Modèles

- Les modèles peuvent être :
 - Client - Server
 - Peer - to - Peer
 - Multi - SGBD Architecture

Architecture des BDD

Client-serveur



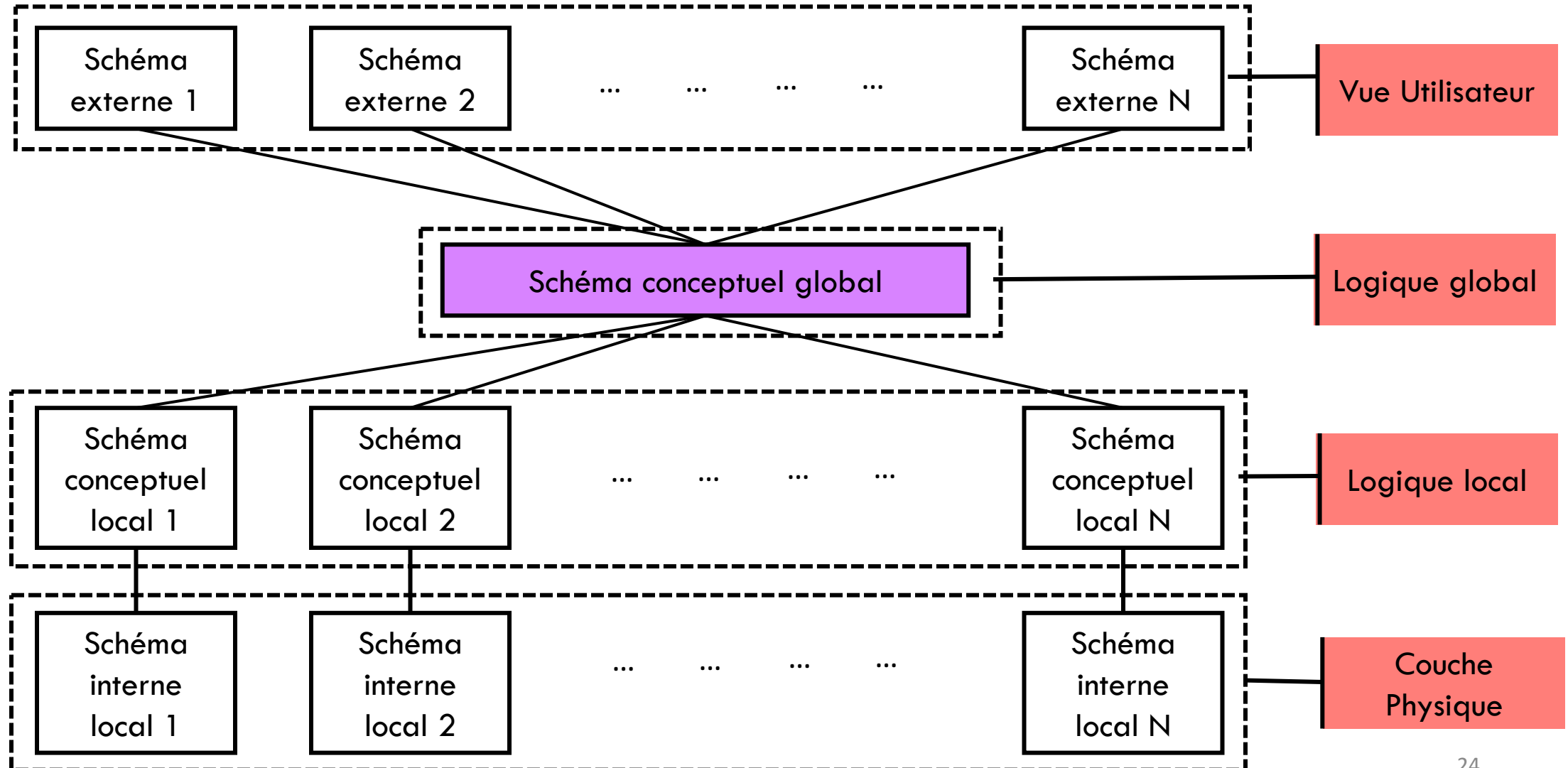
Architecture des BDD

Peer-to-peer

- Dans ces systèmes, chaque pair agit à la fois comme client et serveur
- Les pairs se partagent leurs ressources et coordonnent leurs activités
- Architecture généralement à quatre niveaux de schémas
 - Schéma conceptuel global représentant la vue logique globale des données
 - Schéma conceptuel local décrivant l'organisation des données logiques sur chaque site
 - Schéma interne local décrivant l'organisation des données physiques sur chaque site
 - Schéma externe décrivant la vue utilisateur des données

Architecture des BDD

Peer-to-peer



Architecture des BDD

Multi-SGBD

- Il s'agit d'un système de base de données intégré constitué d'un ensemble de deux ou plusieurs systèmes de base de données autonomes

Architecture des BDD

Multi-SGBD

- Trois niveaux de schémas globaux
 - **Niveau de vue multi-SGBD** - décrit plusieurs **vues** d'utilisateur comprenant des sous-ensembles de la base de données distribuée intégrée
 - **Niveau conceptuel multi-SGBD** - représente une multi-base de données intégrée comprenant des **définitions globales logiques** de structure de plusieurs bases de données
 - **Niveau interne multi-SGBD** - décrit la **répartition** des données sur différents sites et le **mappage** de plusieurs bases de données vers des données locales

Architecture des BDD

Multi-SGBD

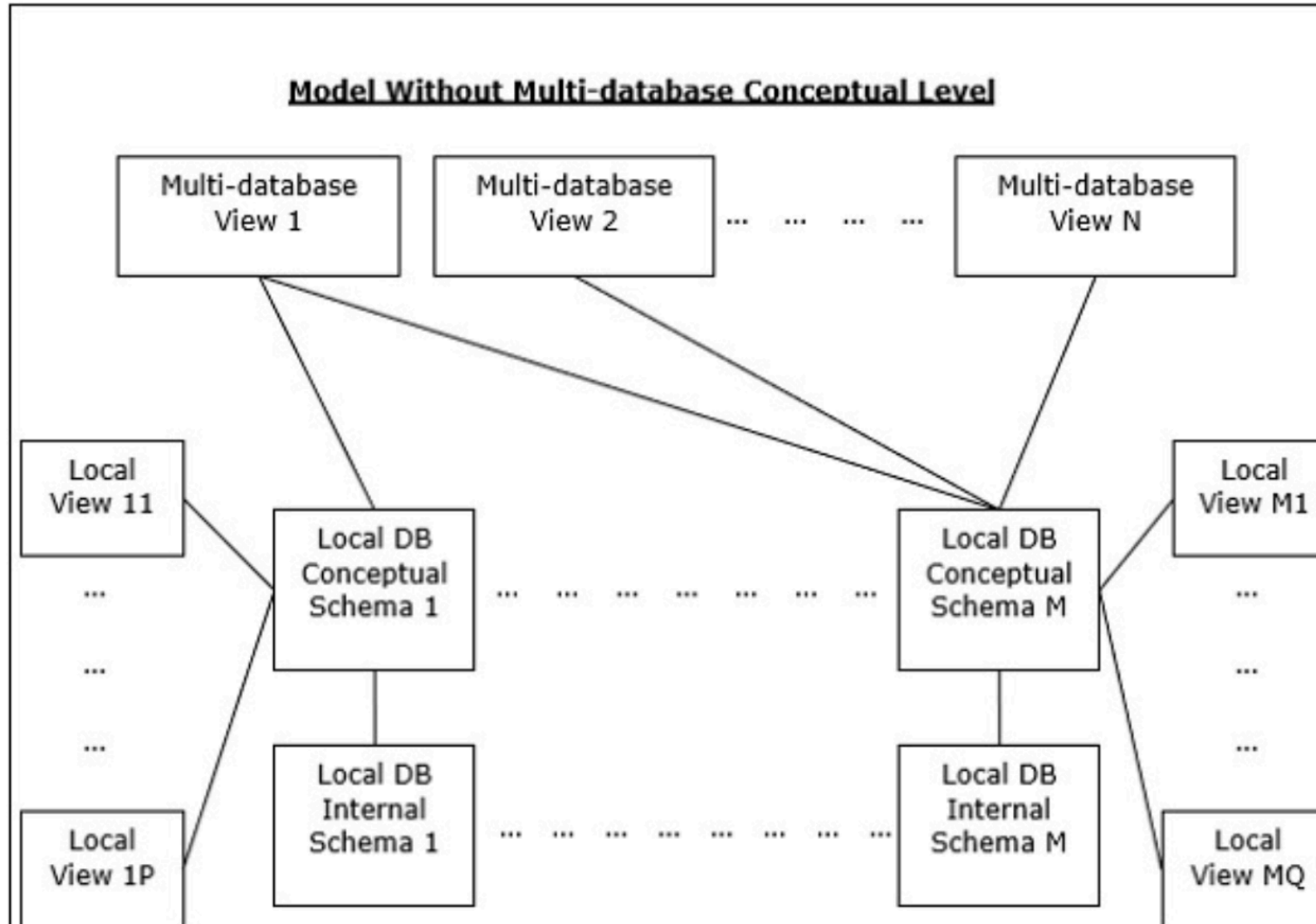
- Trois niveaux de schémas locaux
 - **Niveau de la base de données locale** - décrit la vue publique des données locales
 - **Niveau conceptuel de la base de données locale** - décrit l'organisation des données locales sur chaque site
 - **Base de données locale Niveau interne** - décrit l'organisation des données physiques sur chaque site

Multi-SGBD avec niveau conceptuel



Architecture des BDD

Multi-SGBD sans niveau conceptuel



Architecture des BDD

D'autres architectures ...

- Les alternatives de conception de la distribution pour les tables d'un SGBDR sont les suivantes:
 - Non répliqué et non fragmenté
 - Entièrement répliqué
 - Partiellement répliqué
 - Fragmenté
 - Mixte

Architecture des BDD

Non répliqué et non fragmenté

- Différentes tables sont placées sur différents sites de manière à ce qu'elles se trouvent à proximité du site où elles sont le plus utilisées
- Convient mieux aux SGBD où le pourcentage de requêtes nécessaires pour joindre des informations dans des tables placées sur différents sites est faible
- Si une stratégie de distribution appropriée est adoptée, cette alternative de conception permet de réduire les coûts de communication lors du traitement des données

Architecture des BDD

Entièrement répliqué

- Sur chaque site, une copie de toutes les tables de la base de données est stockée
- Les requêtes deviennent très rapides et ne nécessitent qu'un coût de communication négligeable
- Au contraire, la redondance massive des données nécessite des coûts énormes lors des opérations de mise à jour
- Par conséquent, cela convient aux systèmes où un grand nombre de requêtes doit être traité avec un nombre de mises à jour faible

Architecture des BDD

Partiellement répliqué

- Des copies de tables ou des portions de tables sont stockées sur différents sites
- La distribution des tables est faite en fonction de la fréquence d'accès
- Cela tient compte du fait que la fréquence d'accès aux tables varie considérablement d'un site à l'autre
- Le nombre de copies des tables (ou parties) dépend de la fréquence d'exécution des requêtes d'accès et du site qui génère les requêtes d'accès

Architecture des BDD

Fragmenté

- Une table est divisée en deux ou plusieurs pièces appelées fragments ou partitions, et chaque fragment peut être stocké sur un site différent
- Cela tient compte du fait qu'il arrive rarement que toutes les données stockées dans une table soient requises sur un site donné
- De plus, la fragmentation augmente le parallélisme et permet une meilleure reprise après panne

Architecture des BDD

Fragmenté

- Les trois techniques de fragmentation sont
 - Fragmentation verticale
 - Fragmentation horizontale
 - Fragmentation hybride

Architecture des BDD

Distribution mixte

- Combinaison de fragmentation et de répliquions partielles
- Les tables sont initialement fragmentées sous n'importe quelle forme (horizontale ou verticale), puis ces fragments sont partiellement répliqués sur les différents sites en fonction de la fréquence d'accès aux fragments

Architecture des BDD

Réplication

- La réplication de données consiste à stocker des copies distinctes de la base de données sur deux sites ou plus. C'est une technique populaire pour pallier à la tolérance aux pannes des bases de données

Architecture des BDD

Avantages de la réplication

- Fiabilité en cas de défaillance d'un site
- Réduction de la charge du réseau
- Temps de réponse plus rapide
- Transactions simplifiées avec moins de jointures de tables situées sur différents sites

Architecture des BDD

Inconvénients de la réplication

- Augmentation des exigences de stockage en termes de coûts
- Augmentation du coût et de la complexité de la mise à jour des données
- Couplage entre applications et base de données –
 - si des mécanismes de mise à jour complexes ne sont pas utilisés, la suppression des incohérences de données nécessite une coordination complexe au niveau de l'application

Architecture des BDD

Techniques de réplication

- Certaines techniques de réplication couramment utilisées sont
 - Réplication instantané
 - Réplication en temps quasi réel
 - Pull replication

Architecture des BDD

Fragmentation

- La fragmentation consiste à diviser une table en un ensemble de tables plus petites appelées fragments.
- La fragmentation peut être de trois types:
 - horizontale,
 - verticale,
 - hybride (combinaison horizontale et verticale).

Architecture des BDD

Fragmentation

- La fragmentation horizontale peut en outre être classée en deux techniques:
 - la fragmentation horizontale primaire,
 - la fragmentation horizontale dérivée
- La fragmentation doit être effectuée de manière à ce que la table originale puisse être reconstruite à partir des fragments. Cette exigence est appelée "capacité de reconstruction"

Architecture des BDD

Avantages de la fragmentation

- Les données étant stockées à proximité du site d'utilisation, l'efficacité du système de base de données est accrue
- Les techniques d'optimisation des requêtes locales sont suffisantes pour la plupart des requêtes car les données sont disponibles localement
- Les données non pertinentes n'étant pas disponibles sur les sites, la sécurité et la confidentialité du système de base de données peuvent être préservées

Architecture des BDD

Inconvénients de la fragmentation

- Lorsque des données provenant de différents fragments sont requises, les vitesses d'accès peuvent être très élevées
- En cas de fragmentation récursive, le travail de reconstruction nécessitera des techniques coûteuses
- L'absence de copies de sauvegarde des données sur différents sites peut rendre la base de données inefficace en cas de défaillance d'un site

Architecture des BDD

Fragmentation verticale

- Dans la fragmentation verticale, les champs ou les colonnes d'une table sont regroupés en fragments
- Afin de maintenir la capacité de reconstruction, chaque fragment doit contenir le ou les champs de clé primaire de la table
- Par exemple, considérons qu'une base de données d'une université conserve les enregistrements de tous les étudiants inscrits dans une table d'étudiants ayant le schéma suivant:

Regd_No	Name	Course	Address	Semester	Fees	Marks
---------	------	--------	---------	----------	------	-------

Architecture des BDD

Fragmentation verticale

Dans ce cas, le concepteur pourrait fragmenter la base de données comme suit avec SQL:

```
CREATE TABLE STD_FEES AS  
    SELECT Regd_No, Fees  
    FROM STUDENT;
```

Regd_No	Name	Course	Address	Semester	Fees	Marks
---------	------	--------	---------	----------	------	-------

Architecture des BDD

Fragmentation horizontale

- La fragmentation horizontale regroupe les n-uplets d'une table en fonction des valeurs d'un ou de plusieurs champs
- La fragmentation horizontale devrait également confirmer la règle de la capacité de reconstruction. Chaque fragment horizontal doit avoir toutes les colonnes de la table de base d'origine

Architecture des BDD

Fragmentation horizontale

- Par exemple, dans le schéma de l'étudiant, si les détails de tous les étudiants du cours d'informatique doivent être conservés à la School of Computer Science, le concepteur fragmentera la base de données horizontalement comme suit:

```
CREATE COMP_STD AS
```

```
SELECT *
```

```
FROM STUDENT
```

```
WHERE COURSE = "Computer Science";
```

Architecture des BDD

Fragmentation hybride

- Dans la fragmentation hybride, une combinaison de techniques de fragmentation horizontale et verticale est utilisée
- C'est la technique de fragmentation la plus flexible car elle génère des fragments avec un minimum d'informations superflues
- Cependant, la reconstruction de la table originale est souvent une tâche coûteuse

Architecture des BDD

Fragmentation hybride

- La fragmentation hybride peut être réalisée de deux manières:
 - Au début, générer un ensemble de fragments horizontaux; puis générer des fragments verticaux à partir d'un ou plusieurs des fragments horizontaux
 - Dans un premier temps, générer un ensemble de fragments verticaux; puis générer des fragments horizontaux à partir d'un ou plusieurs des fragments verticaux