

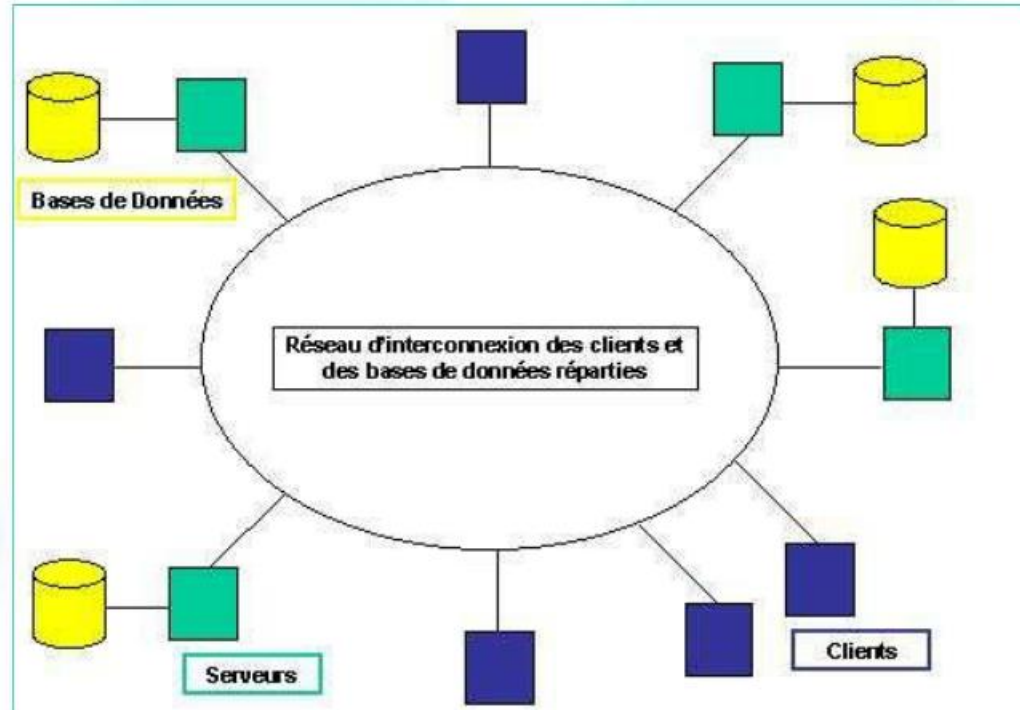


Administration de bases de données

Bases de données réparties



Bases de données réparties





Quelques définitions

- DEFINITION : UNE BASE DE DONNEES REPARTIE EST UN ENSEMBLE DE DONNEES STOCKEES SUR PLUSIEURS SITES CONNECTES PAR UN RESEAU
- SITE : MACHINE AVEC UNE BDD LOCALE
- RESEAU : MOYEN DE COMMUNICATION ENTRE SITES
- CRITERES DE CHOIX : COUT INSTALLATION, COMMUNICATION, SURETE, DISPONIBILITE
- **Pour pouvoir gérer une base de données répartie, il faut pouvoir disposer d'un SGBD Réparti**



Intérêts de la répartition des données

- Plus grande fiabilité (pannes des matériels)
- De meilleures performances
- Faciliter la croissance du système



Inconvénients de la répartition des données

- Coûts (trafic réseau, hardware et software)
- Synchronisation pour accéder aux données
- Sécurité des données
- L'interblocage distribué

Architecture

- Conception de la base de données répartie
 - Méthode top-down design
 - Définition d'un schéma global : Global Conceptual Schema
 - Distribution vers des schémas locaux
 - Fragmentation et allocation des fragments
 - Méthode bottom-up design
 - Intégration de schémas locaux dans un schéma global



Les techniques de répartition

- La fragmentation
- La réplication

La Fragmentation

- **PRINCIPE**

- Répartir une relation (table) sur plusieurs sites. Sur chaque site est stockée une partie de la relation.

- **TECHNIQUES**

Découpage Horizontal	$\Leftrightarrow$	Sélection
Découpage Vertical	$\Leftrightarrow$	Projection
Découpage Mixte	$\Leftrightarrow$	Sélection et projection

- **PROBLEME**

- Recomposer la relation initiale



La Fragmentation Horizontale

- On considère la relation R
- Att un attribut de la relation R
- $vAtt_1, vAtt_2, \dots, vAtt_n$, la liste exhaustive des valeurs de Att
- $\forall i \in \{1, \dots, n\}$ on stocke sur le site s_i :
 $\sigma(R; Att = vAtt_i)$
- $R = \bigcup_i \sigma(R; Att = vAtt_i)$

La Fragmentation Verticale

- On considère la relation R
- $(Att_{1,1}, \dots, Att_{1,k_1}) \dots (Att_{1,1}, \dots, Att_{1,k_n})$ une liste d'ensembles d'attributs de R
- Chaque ensemble de la liste contient une clé de la relation R
- La réunion des ensembles recouvre tous les attributs de R
- Sur le site i , on stocke $\Pi(R; Att_{i,1}, \dots, Att_{i,k_i})$



La Fragmentation Mixte

- 1) On considère une relation R
- 2) On la fragmente verticalement (projections)
- 3) Chaque fragment vertical peut être lui même décomposé en fragments horizontaux (sélections)



Exemple

Création d'un link sur ma base de données

```
create database link db_toulouse connect to user_distant identified  
by password using 'XE'
```

Remarque : using → nom du service
requête locale et requête distante

```
select * from table_name;  
select * from table_name@db_toulouse;
```



Opérations sur les liens

Fermeture

- Utile si forte charge
- `ALTER SESSION CLOSE DATABASE LINK lien;`

Suppression

- `DROP DATABASE LINK lien`



Fragmentation Horizontale

```
CREATE VIEW V1  
AS SELECT Table1.cle, Table1.attr1  
FROM Table1@site1  
UNION  
SELECT Table2.cle, Table2.attr1  
FROM Table2@site2;
```



Fragmentation Verticale

```
CREATE VIEW V1  
AS SELECT Table1.cle, Table1.attr1,  
Table2.attr2  
FROM Table1@site1, Table2@site2  
WHERE Table1.cle=Table2.cle ;
```



Exemple pratique

- Relation Employé (nss, nom, loc, ...)
- Relation Taux (pays, valeur, ...)
- 12 bureaux d'environ la même taille dont
6 à Paris, 4 à Marseille et 2 à Lyon
- 80% des requêtes dans une ville portent sur les
employés de la ville
- 10% des requêtes dans une ville portent sur Taux



Exemple suite

- Créer 3 bases de données : P, M, L
- Sur chaque base : les employés de la ville et une copie de Taux.
- Si trop cher :
 - fusionner M et L
 - ou maintenir Taux à M.

Fragmentation Horizontale

- Fragments définis par sélection

Ex : Clients(NClient, Nom, Ville)

Client1 = $\sigma_{\text{Ville} = \text{Paris}}(\text{Client})$

Client2 = $\sigma_{\text{Ville} \neq \text{Paris}}(\text{Client})$

- Reconstruction par union des fragments

Ex : Client = Client1 U Client2



Fragmentation horizontale

Relation Client

NoClient	Nom	Ville
C1	Dupont	Paris
C2	Martin	Marseille
C3	Martin	Paris
C4	Talon	Lille



Fragmentation Horizontale suite

NoClient	Nom	Ville
C1	Dupont	Paris
C3	Martin	Paris

Client1

NoClient	Nom	Ville
C4	Talon	Lille
C2	Martin	Marseille

Client2



Fragmentation Horizontale Dérivée

Fragments définis par (semi) jointure

Ex : Commande(NC, NClient, Produit, Qté)

Commande1 = Commande $\bowtie$ Client1

Commande2 = Commande $\bowtie$ Client2

Reconstruction par union des fragments

Ex : Commande = Commande1 $\cup$ Commande2



Fragmentation Horizontale Dérivée - Exemple

Relation Commande

NC	NClient	Produit	Qte
Co1	C1	P1	10
Co2	C1	P2	200
Co3	C2	P3	30
Co4	C4	P4	5



Fragmentation Horizontale Dérivée - Exemple

Relation Commande1

NC	NClient	Produit	Qte
Co1	C1	P1	10
Co2	C1	P1	200

Relation Commande2

NC	NClient	Produit	Qte
Co3	C2	P3	30
Co4	C4	P4	5



Fragmentation Verticale

- Fragments définis par projection

Ex : Commande(NC, NClient, Produit,Qté)

CommandeA = $\pi_{\text{NC, NClient}}(\text{Commande})$

CommandeB = $\pi_{\text{NC, Produit, Qté}}(\text{Commande})$

- Reconstruction par jointure

Ex : Commande = CommandeA * CommandeB

- Utile si forte affinité des attributs

Fragmentation Verticale – Suite

Exemple

NC	NClient
Co1	C1
Co2	C2
Co3	C3
Co4	C4

Relation CommandeA

NC	Produit	Qte
Co1	P1	10
Co2	P2	200
Co3	P3	30
Co4	P4	5

Relation CommandeB



La réplication

- **PRINCIPE**
 - Une relation stockée sur un site est recopiée sur un ou plusieurs sites
- **AVANTAGES**
 - Disponibilité des données
 - Augmentation du parallélisme en lecture
 - Diminution du coût imposé par les transmissions
- **INCONVENIENTS**
 - Cohérence des différentes copies
 - Propagation des mises a jour



La solution Oracle en informatique répartie

- **La couche Oracle Net**

- Le fichier Listener.ora (fichier de configuration pour traiter les connexions distantes)
- Le fichier tsnane.ora (fichier contenant des alias de bases de données)
- L'outil assistant Net Configuration pour mettre à jour les fichiers ci-dessus
- Le programme listener pour détecter les requêtes de connexion

- **Le lien de base de données**

- create [public] database link nomLien connect to user identified by passwd using 'nomService ' où nomService est un alias de bases de données défini sur le serveur de connexion



La solution Oracle en informatique répartie

- **La réplication**

- create materialized view image REFRESH FAST ... AS SELECT FROM T@Lien;

- **La Défragmentation**

- Création de vue avec une jointure sur plusieurs tables $T_1@L_1$, $T_n@L_n$



Validation d'une transaction

- Quoi
 - Une transaction T répartie $\Rightarrow n$ transactions T_i locales
 - Il faut coordonner les sites S_i .
- Comment
 - Il faut un site coordinateur
 - On utilise la validation à deux phases



Propriétés d'un protocole de validation

C'est un protocole de consensus sur l'exécution d'une transaction entre les participants qui émettent un vote

- Oui pour une validation (« commit »)
- Non pour un abandon (« rollback »)

1. La décision générale de validation est prise si tous les participants ont votés oui

2. Un participant ne peut plus revenir sur sa décision

- (voir <http://cedric.cnam.fr/vertigo/Cours/Valeur-C/Exposes/concurrence.ppt>)



Le protocole de validation : Phase 1

- Le coordonateur **C** écrit <T préparée> dans son journal
- **C** envoie « préparation » à tous les sites S_i
- Chaque site **S_i** prend une décision pour T_i :
 - s'il valide T_i , alors :
 - **S_i** écrit <T prête > dans son journal
 - **S_i** envoie « T prête » au coordonateur **C**
 - s'il ne valide pas T_i , alors :
 - **S_i** écrit <non T> dans son journal
 - **S_i** envoie « abandon T » au coordonnateur **C**
- (voir <http://cedric.cnam.fr/vertigo/Cours/Valeur-C/Exposes/concurrence.ppt>)



Le protocole de validation : Phase 2

- A partir des messages reçus des sites S_i , **C** prend une décision :
 - S'il a reçu « T prête » de tous les sites S_i :
 - il ajoute <T validée> à son journal
 - il envoie « valider T » à tous les sites S_i
 - S'il a reçu au moins un message « abandon T » d'un site S_i :
 - il ajoute <T abandonnée> à son journal
 - il envoie <abandon T> à tous les sites S_i
- Chaque site **S_i** :
 - inscrit dans son journal le message envoyé par **C** (<T validée> ou <T abandonnée>)
 - envoie un accusé de réception au coordinateur **C**
- Quand le coordinateur **C** a reçu tous les accusés des sites **S_i** , il ajoute : <T finie> à son journal
- (voir <http://cedric.cnam.fr/vertigo/Cours/Valeur-C/Exposes/concurrence.ppt>)

Traitement de reprise après une panne

- Après panne du coordinateur :
 - Si reprise dans la phase préparatoire avant diffusion du statut de la transaction => abandonner la transaction.
 - Si reprise après décision prise => ne rien faire.
- Après panne chez un exécutant :
 - Si reprise dans la phase préparatoire => abandonner la transaction.
 - Si reprise en état de décision connue => ne rien faire.
- (voir <http://cedric.cnam.fr/vertigo/Cours/Valeur-C/Exposes/concurrence.ppt>)