

Partie III. Systèmes de BD répartis

I. Introduction aux BD réparties BDR

1. Motivations
2. Définitions
3. Objectifs des BDR
4. Fonctions d'un SGBD Réparti
5. Architectures de références
6. Dictionnaires de données

II. Conception d'une BDR

1. Conception descendante ou ascendante
2. Stratégies de fragmentation

III. Principe d'évaluation de requêtes réparties

1. Position du problème
2. Phases dévaluation d'une req. «centralisée»
3. Phases dévaluation d'une req. répartie
4. Algorithmes de jointure : jointure directe et semi-jointure
5. Processus de localisation
6. Optimisation d'une requête répartie

Références bibliographiques

1. **SGBD Avancés - BD objets, déductives, réparties**
G. Gardarin et P. Valduriez; Ed. Eyrolles
2. **Le Client-Serveur,**
G et O. Gardarin; Ed. Eyrolles
(Chapitre. V. BD réparties et répliquées)

I. Introduction aux BD réparties BDR

1. Motivations

Développement de systèmes répartis

Solution naturelle pour gérer les données d'une grande entr.

Partage de données :

- **Interconnexion des calculateurs**
- **Systèmes spécifiques pour gérer les données réparties**
haut degré d'indépendance par rapport à l'env. réparti
la technologie des BD relationnelles offre ce type
d'indépendance
==> Les BDR s'appuient sur le *modèle relationnel*

☞ **La construction d'un SGBD réparti est fondée sur la combinaison de deux technologies : *SGBD et systèmes répartis***

2. Définitions : BDR et SGBDR

1. BDR : «ensemble de BD gérées par des sites différents et apparaissent à l'utilisateur comme une base unique»

- **Chaque BD réside sur un site différent**
- **La répartition des données est transparente à l'Ui**
- **La gestion d'une BDR nécessite sur chaque site Si l'installation des composants systèmes : système de comm. , un SGBD local, gestionnaire des données réparties (SC, SGBD GDR).**

2. Exemple de BDR

3. Schéma global :

une BDR possède un schéma appelé *schéma global*.

Il permet de définir l'ensemble des types de données de la base.

Exemple de schéma global relationnel

4. SGBD réparti

Déf1. :

« Ensemble de logiciels systèmes constitués des 3 sous-systèmes (SC, SGBD, GDR) résidant sur chaque site de la BD répartie «

Déf2.

« Système qui gère des collections de BD logiquement reliées, distribuées sur un réseau, en fournissant un mécanisme d'accès qui rend transparent la répartition des données aux Ui «.

***Client d'un SGBDR :* Application qui accède aux info. distribuées par les interfaces du SGBDR**

***Serveur de SGBDR :* SGBD gérant une BD locale intégrée dans une BDR**

☞ Un site peut être à la fois *client et serveur* du SGBDR: Il exécute des applications et gère une partie de la base.

3. Objectifs des BDR

1. Indépendance à la localisation

- Ui peut ignorer la localisation des données
- La transp. à la localisation fournit l'indép. physique
- L'info concernant la localis. est maintenue dans le dictionnaire de données et consultée par le SGBD réparti pour déterminer la localisation des relations appartenant à la req.

2. Indépendance à la fragmentation

(a) **Fragmentation** : une relation est divisée en sous-relations appelées *Fragments (efficacité)*

- Les fragments sont stockés à des sites différents
- La fragmentation augmente les perf. d'une BDR, car les accès locaux sont favorisés

Exemple : Soit une relation EMP d'une BDR entre Paris et Prague. Un placement optimal de la rel. EMP serait : EMPF localisée à Paris et EMPP localisée à Prague

afin que les données se trouvent là où elles sont les plus souvent accédées

(b) **Indépendance à la fragmentation**

- cache à l'Ui le fait que les données sont fragmentées
- l'Ui ne manipule que des relations logiques non réparties

requête (rel.) ----> SGBDR-----> requêtes (fragments)
Dict. de données

3. Indépendance à la duplication des données

Un fragment est dupliqué lorsqu'il est matérialisé par 2 ou plus copies, chacune résidant sur un site différent.

- **disponibilité des données (panne, requêtes complexes)**
- **amélioration des perf. en augmentant la localité des références (cas où le coût de comm. est facteur dominant)**
- **les avantages de la duplication doivent être comparés avec le coût de la maintenance des copies (en cas de maj)**
- **l'indépendance à la duplication rend la duplication invisible à l'Ui (il ne voit que des rel non dupliées et non fragmentées)**
- **l'info. relative à la duplication est maintenue dans le dictionnaire de données**
- **exploitée par le SGBDR lors de l'évaluation des req. en req. optimisées sur des fragments, avec le souci de maintenir la cohérence des copies**

4. Indépendance aux SGBD

- **permet de cacher le fait que les SGBD locaux peuvent être différents (Modèles et langages)**
- **c'est l'objectif particulier pour les BDR hétérogènes et aussi les BD fédérées :**
 - traducteur du modèle de données et du langage de chaque SGBD**
 - ==> Choix d'un modèle pivot et son langage associé.**

5. Autonomie des sites

- permet à chaque site opérationnel de contrôler et de manipuler ses données locales indépendamment des autres sites.
- pour être autonome, chaque site doit disposer localement de toute l'info. qui se trouve dans le dictionnaire de données
==> l'administration de la BDR peut être décentralisée
:administrateur local

6. Extensibilité d'une BDR

est sa capacité d'augmentation incrémentale, par de nouveaux sites dans le réseau, avec un minimum d'impact sur

- les BD locales,
- les prog. d'appl. existants

==> permet d'augmenter les capacités de traitement et de stockage.

7. Performances (minimiser le TR et maximiser le débit)

- Fragmentation
- Duplication (Panne, traitement //)

4. Fonctions d'un SGBD réparti (en dehors de celles des SGBD Locaux et du GC)

- (1) Gestion d'un dictionnaire de données global**
- (2) Définition des données réparties**
- (3) Contrôle sémantique des données réparties**
- (4) Evaluation des requêtes réparties**
- (5) Gestion des transactions réparties**

5. Architectures de références

1. Architectures des schémas d'une BDR

- ce schéma décrit une organisation est idéale
- décrite en termes de niveaux : du global au local
- info. relatives aux schémas et aux transf. de schémas sont maintenues dans le dictionnaires de données.
- les SGBDRs n'implémentent pas tous les niveaux de schémas

Schéma global SG : " Description globale et unifiée de toutes les données appartenant à la BDR indépendamment de l'env. réparti" (i.e. indépendamment de toute base locale)

==> Indép. physique des données

Quant à l'indép. logique, elle est assurée par le S. Ext.G

Requête répartie : "Requête exprimée conformément aux données décrites par le schéma global d'une BDR"

Schéma de placement SP :

- indique la manière dont les relations sont placées sur les différents sites.
- Il contient toutes les info. relatives à la localisation, fragmentation, et duplication.
- en particulier les règles de transfo. des données du niveau SG en données localisées, fragmentées et dupliquées.

==> le SP assure donc l'indépendance à la localisa. à la frag. , et à la duplication.

Schéma local : " Schéma décrivant les données d'une BD locale gérées par le SGBD local "

Requête locale : " Requête exprimée conformément aux données décrites par le schéma (local) d'une BD locale"

- le schéma local interne et le schéma local logique sont identiques à ceux d'une BD centralisées.

- schéma local externe : représente les fragments tels qu'ils sont décrits dans le schéma de placement *en objets externes*, considérés comme *des vues* de la base locale.

==> le schéma local externe assure l'indépendance des SGBD (obligatoire en cas d'env. hétérogène)

Caractéristiques d'une BDR hétérogènes : intégration des BD locales gérées par des SGBD différents.

Deux niveaux d'hétérogénéités :

- (i) les BD ont le même modèle, mais sont gérées par des SGBD différents
- (ii) les BD ont un modèle différent et sont gérées par des SGBD différents.

2. Architecture fonctionnelle d'un SGBD réparti

Voir Schéma

- le *contrôle sémantique des données* transforme la req. exprimée sur des vues, en une req. exprimée sur des rel. concept. globales, puis ajoute les contrôles d'autorisations et les contrôles d'intégrité sémantique.

- l'*évaluation de requêtes réparties* transforme une req. exprimée sur des relations en un *Plan d'exécution réparti*.

Plan d'exécution réparti : " Programme décrivant la stratégie d'exécution optimisée d'une req. répartie. Il peut être vu comme une séquence optimisée de req. locales, synchronisées par des échanges inter-sites de données intermédiaires "

- ***La gestion de transactions (réparties)*** coordonne l'exécution d'une req. répartie, constituant une trans., en communiquant avec le gerant de données des sites sur lesquels les req. locales s'exécutent comme une sous-transaction.

Cette fonction assure le contrôle de concurrence réparti et la validation de transactions (Voir cours sur les transactions)

- ***Evaluation d'une req. locale :***

req.(exprimée/SP) ---Traduction--> req(exprimée/SEL)

SP : Schéma de Placement; SEL : Schema Externe Local

Exemple : dans le cas d'une BD hétérogène, cette traduction nécessite la conversion de la req.(exprimée dans le modèle et le langage pivot) en une req. (exprimée dans le modèle de la base locale).

- la req. est ensuite transformée en une req. optimisée exprimée suivant le schéma interne local.

3. Principaux problèmes posés par la gestion de données en env. réparti : Sources de complication

S1 : Répartition et duplication

S2 : Grand nombre de ressources (sites multiples, réseau de comm)

S3 : Etat partiel du système

6. Dictionnaires de données

- L' info contenue dans le dico concerne les 3 niveaux de SG
- utilisé pour traduire , analyser, et évaluer des requêtes
- géré comme une BDR : fragmenté, dupliqué
- placement d'un dico :
 - degré d'autonomie souhaité pour chaque site
 - performances à obtenir
- Organisation du dico : central., duplication, et répartition

Tableau 1: Organisation du Dic. de données

Dico	S1	S2	S3
Centra- lisé	BUVEUR VIN, ABU		
Dupliqué	BUVEUR VIN ABU	BUVEUR VIN ABU	BUVEUR VIN ABU
Réparti	BUVEUR	VIN	ABU

A1 : Approche centralisée

- Site unique (un seul point d'accès) : Administ. et maj sont simples et efficaces
- accès au site central par des req. initiées sur d'autres sites
- si le site central tombe en panne ==> BDR inutilisable

A2: Approche dupliquée

- gère une copie du dico. sur chaque site
- analyse et évaluation des req. très efficaces
- pb de cohérence en cas de maj du dico.

A3 : Approche réparti

- les 2 approches A1 et A2 sont incompatibles avec l'objectif d'autonomie de site
- cette approche A3 consiste à fragmenter le dico en dico locaux, chacun contenant uniquement l'info pertinente aux données du site où il est stocké
- manipulation des données locales efficaces
- manipulation des données distantes est plus complexes :
 - détection des données non locales
 - détermination du site de leur définition.

II. Conception d'une BDR

1. Conception descendante ou ascendante

(i) Conception descendante

- un SCG est d'abord élaboré
- les objets de ce schéma sont distribués sur les sites permettant ainsi de définir des schémas locaux SL

Déf. BDR homogène : «BDR obtenue en divisant une BD en un ensemble de BD locales, chacune étant gérée par le même SGBD»

Avantages :

- facilite de croissance incrémentale de l'entreprise :
l'ajout d'un nouveau site
- augmente les perfs. : exploiter les capacités de trait. //

(ii) Conception ascendante

- permet l'intégration de BD locales existantes dans une BD fédérée
- consiste à intégrer des SL existants --> un ou pls SG
- nécessite une réconciliation sémantique des schémas

Déf. BDR hétérogène: « BDR obtenue en intégrant dans une BD unique un ensemble de bases de données locales gérées par des SGBD différents»

2. Stratégies de fragmentation

A. Def. fragment : « sous-relation obtenue par sélection de tuples et d'attributs à partir d'une relation globale, localisée sur un site unique ».

==> deux façons possibles de diviser une rel. : *horizontalement (sélection de tuples) et verticalement (sélection d'attributs)*.

On obtient donc 3 types de fragmentation :

Frag. horizontale, Frag. verticale, et Frag. mixte

B. Règles de la fragmentation:

Objectif : préserver la cohérence sémantique de la BDR

R1 : décomposition sans perte (similaire à la)

R2 : reconstruction (d'une rel. globale)

R3 : non duplication (cas d'une Frag. horizontale)

C. fragmentation horizontale

« Fonction qui partitionne une relation en sous -relations, chacune étant défini par une opération de sélection appliquée à la relation »

- adapté à la régionalisation/départementalisation d'une entreprise : $Region = 'A'$

- limite le nombre de fragments accédés (req. portant sur des attributs utilisés dans la définition des fragments)

Ex. $VIN1 = \sigma_{Mil < 1984} VIN$

$VIN2 = \sigma_{Mil \geq 1984} VIN$

Etudier la vérification des règles R1, R2, et R3

D. fragmentation verticale

«Fonction qui partitionne une relation en sous-relations, chacune étant défini par une opération de projection appliquée à la relation»

- duplication dans chaque fragment le ou les attributs clés afin de reconstruire la relation à partir de ses fragments verticaux
- la reconstruction est obtenue par la jointure des fragments selon les attributs communs
- la duplication d'attributs communs \implies R3 est violée pour les attributs clés

Exemple $VIN1 = \Pi_{NV, CRU, Mil} VIN$
 $VIN2 = \Pi_{NV, DEGRE, PRIX} VIN$

$VIN = \text{Join nat. } (VIN1, VIN2, Q = \{\text{attr. communs}\} = NV)$
 On ne conserve qu'un seul attribut

Avantage

- Limitation du nombre de fragments accédés : requêtes de projection portant sur les attributs utilisés dans la définition des fragments

E. Fragmentation mixte : frag. horiz. et la frag. vertic.

« Fonction qui partitionne une relation en sous-relations de sous-tuples, les sous-tuples étant définis par fragmentation verticale et les sous-relations par frag. horizontale»

Exemple $VIN1 = \Pi_{NV, DEGRE, PRIX} VIN$
 $VIN2 = \sigma_{MIL < 1984} (\Pi_{NV, CRU, Mil} VIN)$
 $VIN3 = \sigma_{MIL \Rightarrow 1984} (\Pi_{NV, CRU, Mil} VIN)$
 $VIN = VIN1 \bowtie (VIN2 \cup VIN3)$

F. Problèmes à étudier (hors programme)

- ☞ **Choix de la fragmentation** : Horizontale vs verticale dépendant des besoins d'accès des applications.
- ☞ **Allocation des fragments** : Allocation des fragments à des sites.

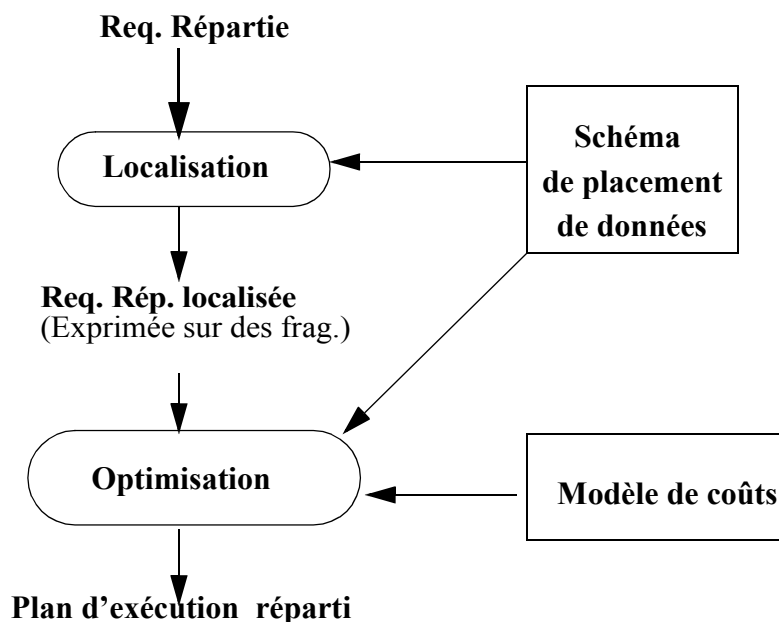
III. Principe d'évaluation de requêtes réparties

1. Sources de la complexité du problème :

- possibilités de fragmentation et de duplication
- communication et parallélisme
- choix des sites d'exécution

2. Système centralisé : Décomposition, Optimi., exécution

3. Système repartit : Décomp., Localisation, Optimi., exéc.



Phases d'évaluation d'une req. répartie

4. Algorithmes de jointure (relations résidant à des sites différents S_1, S_2, \dots, S_n)

(i) Jointure directe JD: $R_1(S_1) \bowtie R_2(S_2)$

Choix du site où la JD sera exécutée

(ii) Jointure à base de Semi-Jointure SJ : $\Pi . SJ. JD$

Objectif :

La semi-jointure consiste à réduire la taille de la relation opérande afin de réduire les coûts de communication

Défi. d'une Semi-Jointure (R, S, Q) : {des tuples de R participant à la jointure de R et S selon la qualification Q}

5. Localisation

Transformation d'une req., en fonction du SP, en une req. répartie équivalente exprimée sur des fragments .

Le processus de localisation d'une requête répartie procède en 2 étapes :

E1 : Génération d'une req. canonique équivalente.

Une req. canonique, exprimée sur des fragments, est obtenue simplement en remplaçant chaque relation de la req. répartie par la requête de reconstruction correspondante. La reconstruction d'une relation à partir de ses fragments peut être obtenue par une opération relationnelle spécifiant la règle de reconstruction.

E2 : Simplification

- (i) Restructuration de l'arbre algébrique afin de rapprocher les opérateurs unaires des fragments**
- (ii) Application des règles de simplification (i.e. opérations inutiles).**

6. Optimisation de requêtes réparties

(i) Stratégie simple (sans optimisation) :

- Exécuter d'abord les opérateurs unaires sur les fragments comme des req. locales
- Transférer les résultats vers un site unique pour assembler les résultats

(ii) Stratégie avec optimisation (Minimisation d'une fonction de coûts)

Le processus d'optimisation procède en 2 étapes :

E1 : Optimisation globale

E11 : Choix des meilleurs sites = Meilleure localisation des copies de fragments :

- (i) Sélection des copies de fragments à accéder,**
- (ii) Sélection des sites d'exécution**

E12 : Ordonnancement des requêtes locales = Ordonnancement des sites de manière à minimiser une fonction de coûts

E2 : Optimisation locale (optimisation sur chaque site)

E21 : Optimisation Physique

- (i) Choix de l'algorithme de jointure/sélection le plus approprié**
- (ii) Ordonnancement des jointures**

E22 : Parallélisation (si serveur parallèle) : parallélisme Intra opération et Inter opération (pipeline et indépendant).