

Bases de données Réparties, Fédérées et Réplication



Bernard ESPINASSE
Professeur à Aix-Marseille Université (AMU)
Ecole Polytechnique Universitaire de Marseille



Avril 2013

1. Répartition & Fédération
2. Bases de données Réparties
3. Bases de données Fédérées
4. Accès aux données réparties
5. Réplication

Plan

1. Introduction : Répartition & Fédération

2. Bases de données Réparties

- Problèmes majeurs
- Fragmentation horizontale
- Evaluation et optimisation des requêtes réparties

3. Bases de données Fédérées

- Procédure d'intégration : Traitement de l'hétérogénéité sémantique, traduction des schémas, intégration des schémas
- Cas de conflits dans l'intégration
- Architecture de référence

4. Accès aux données réparties

- RDA, DRDA, SQL-CLI, ODBC, JDBC
- Vues réparties et SGBD répartis

5. Réplication

- Objectifs de la réplication
- Modes de réplication

Bibliographie du cours

Livres :

- Gardarin G., « Bases de données objet et relationnel », Ed. Eyrolles, 1999 (ISBN : 2-212-09060-9).
- Gardarin G. & Gardarin, « Le Client-Serveur », Ed. Eyrolles, ISBN 2-212-08876-0, 1996
- Miranda & Ruols, « Client-Serveur : Concepts, moteurs, SQL et architectures parallèles », Ed. Eyrolles, ISBN 2-212-08816-7, 1994
- ...

Présentations de :

- René J. Chevance
- Georges Gardarin
- Didier Donsez
- ...

1. Introduction

- La répartition des données
- BD Réparties
- Fédération de BD existantes : BD Fédérées

Introduction : définitions

• BD Répartie (*Distributed DB*)

un schéma global et une collection de BDs logiquement reliées et réparties entre plusieurs sites

• Base de données fédérée - a priori hétérogène (*Federated DB*)

Plusieurs BD hétérogènes capables d'interopérer via une vue commune (**modèle commun**)

• Multibase

Plusieurs BD (hétérogènes ou non) capables d'interopérer sans une vue commune (**absence de modèle commun**)

• SGBD réparti ou SGBD distribué (*Distributed DBMS*)

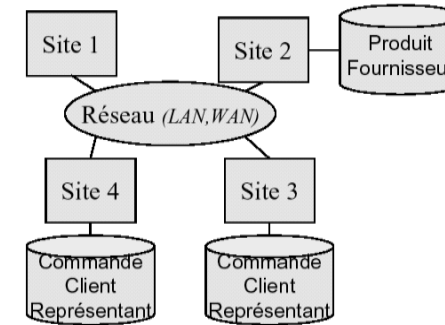
Système gérant une collection de BD logiquement reliées, réparties sur différents sites en fournissant un moyen d'accès rendant la distribution transparente à l'utilisateur

Exemple de BD répartie

Schéma global :

- Produit (NoProd, DesProd, Prix, NoFour)
- Fournisseur (NoFour, NomFour, VilleFour)
- Client (NoCli, NomCli, VilleCli)
- Représentants (NoRep, NomRep, VilleRep)
- Commande (NoProd, NoCli, Date, Qte, NoRep)

Schéma des données réparties :



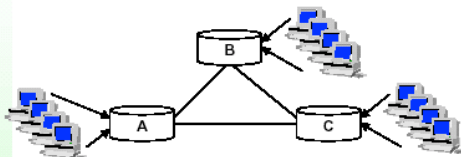
La répartition des données

Centralisé



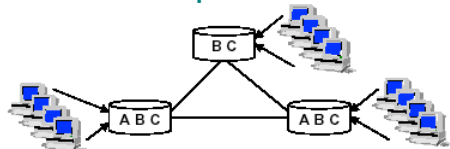
- + : Égalité des accès
Facilité de gestion
- : Contention sur la BD

Répartition



- + : Rapidité d'accès aux données locales
Autonomie locale de chaque site,
Égalité des accès, facilité de gestion
Accès possible aux autres sites
- : Gestion globale de la BD

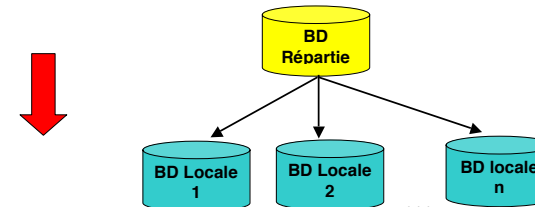
Duplication



- + : Disponibilité des données
Rapidité d'accès aux données locales
- : Coordination de mises à jour

BD Réparties

Approche descendante : **décomposition** en BD locales



Intérêts :

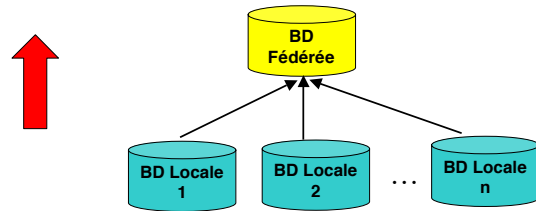
- Amélioration des performances (placer les traitements à l'endroit où se trouvent les données)
- Disponibilité en raison de l'existence de plusieurs copies
- Maintien d'une vision unique de la base de données malgré la répartition

Problèmes :

- Complexité de la répartition (fragmentation, duplication, placement)
- Définition des schémas locaux à partir du schéma global

Fédération de BD existantes : BD Fédérées

Approche ascendante : **intégration** logique de BD locales



Intérêts :

- **Amélioration des performances** (placer les traitements où se trouvent les données)
- **Donner aux utilisateurs une vue unique des données** implémentées sur plusieurs systèmes à priori hétérogènes (plates-formes et SGBD)
- **Cas typique rencontré lors de la concentration d'entreprises** : faire cohabiter les différents systèmes tout en leur permettant d'interopérer

Problèmes :

- **Hétérogénéité sémantique** (BD) et **syntactique** (SGBD, communications,...)
- **Intégration des schémas locaux** pour créer un schéma global

2. Bases de données Réparties

- **Problèmes majeurs dans les BD réparties**
- **Fragmentation et duplication**
- **Fragmentation horizontale**
- **Evaluation et optimisation des requêtes réparties**

Problèmes majeurs dans les BD réparties

Décomposition de la BD-R :

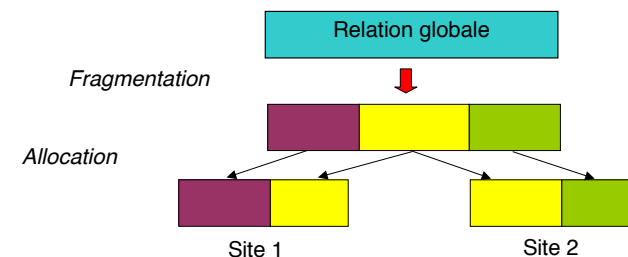
- **Fragmentation**
- **Duplication**

Evaluation et Optimisation de requêtes réparties

Transactionnel distribué :

- **Fiabilité et résistance aux pannes** : terminaison distribuée
- **Contrôle de concurrence** : résolution des interblocages (deadlocks) :

Fragmentation et duplication



Fragmentation :

- **3 types** : Horizontale, Verticale, Mixte
- **Performance** en favorisant les **accès locaux**
- **Equilibrage de la charge** entre les sites

Duplication :

- **Favoriser les accès locaux**
- **Augmenter la disponibilité** des données

Fragmentation horizontale

Fragments définis par sélection :

- Client1 = select * from Client where ville = "Marseille"
- Client2 = select * from Client where ville <> "Marseille"

Reconstruction :

Client = Client1 U Client2

Client	NoCli	NomCli	VilleCli
	C1	Marius	Marseille
	C2	Durant	Lyon
	C3	Duval	Paris
	C4	Olive	Marseille

Client 1	NoCli	NomCli	VilleCli
	C1	Marius	Marseille
	C4	Olive	Marseille

Client 2	NoCli	NomCli	VilleCli
	C2	Durant	Lyon
	C3	Duval	Paris

Fragmentation Verticale

Fragments définis par projection:

- Produit1 = select NoProd, DesProd, Prix from Produit
- Produit2 = select NoProd, NoFour from Produit

Reconstruction :

Produit = select NP, Désigantion, PrixUnit, NF from Produit1 join Produit2 using(NP)

Produit	NoProd	DesProd	Prix	NoFour
	P1	chaise	80	F1
	P2	bureau	150	F1
	P3	table	100	F2
	P4	fauteuil	200	F2

Produit 1	NoProd	DesProd	Prix
	P1	chaise	80
	P2	bureau	150
	P3	table	100
	P4	fauteuil	200

Produit 2	NoProd	NoFour
	P1	F1
	P2	F1
	P3	F2
	P4	F2

Fragmentation horizontale dérivée

Fragments définis par jointure:

- Commande1 = select * from Commande where NoCli in (select NoCli from Client1)
- Commande2 = select * from Commande where NoCli in (select NoCli from Client2)

Reconstruction :

Commande = Commande1 U Commande2

Commande	NoProd	NomCli	date	Qte
	P1	C1	10/1/04	10
	P2	C2	10/1/04	20
	P2	C2	21/3/04	15
	P3	C4	15/4/04	20

Commande1	NoProd	NomCli	date	Qte
	P1	C1	10/1/04	10
	P3	C4	15/4/04	20

Commande2	NoProd	NomCli	date	Qte
	P2	C2	10/1/04	20
	P2	C2	21/3/04	15

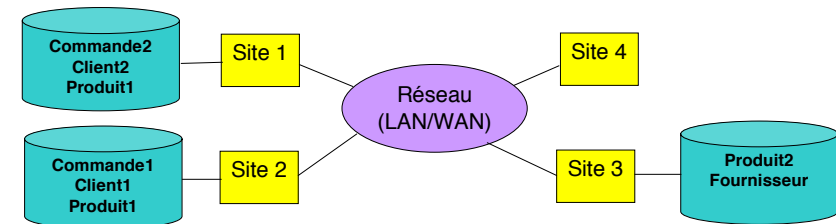
Allocation de fragments aux sites

Non-Dupliquée :

- **partitionnée** : chaque fragment n'existe que sur un seul site

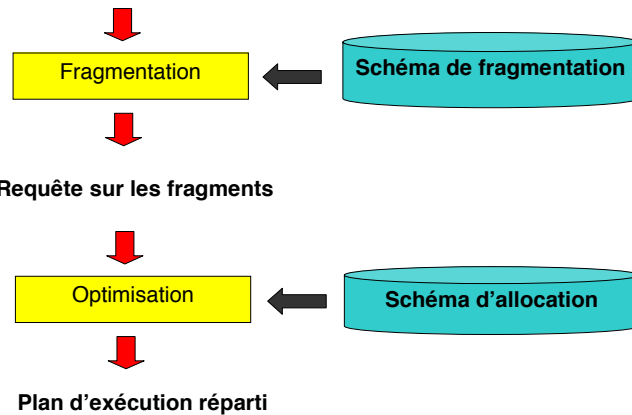
Dupliquée :

- chaque fragment est **sur plus d'un site**
- maintien de la **cohérence** des copies multiples



Evaluation et optimisation des requêtes réparties

Requêtes sur les tables de schéma global

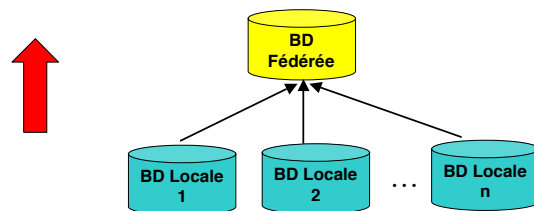


3. Bases de données Fédérées

- Procédure d'intégration
- Traitement de l'hétérogénéité sémantique
- Traduction des schémas
- Intégration des schémas
- Cas de conflits dans l'intégration
- Architecture de référence

Fédération de BD existantes : BD Fédérées

Approche ascendante : **intégration** logique de BD locales



Intérêts :

- **Amélioration des performances** (placer les traitements où se trouvent les données)
- **Donner aux utilisateurs une vue unique des données** implémentées sur plusieurs systèmes à priori hétérogènes (plates-formes et SGBD)
- **Cas typique rencontré lors de la concentration d'entreprises** : faire cohabiter les différents systèmes tout en leur permettant d'interopérer

Procédure d'intégration

Procédure d'intégration :

1. **Traitement de l'hétérogénéité sémantique** (données de la base)
2. **Traitement de l'hétérogénéité syntaxique : Traduction des schémas** (SGBD, communication)
3. **Intégration des schémas** (pour créer un schéma global)

1 - Traitement de l'hétérogénéité sémantique

Hétérogénéité sémantique :

▪ Origine :

Résulte des conceptions indépendantes des différentes BD

▪ Effet :

Désaccord sur la signification des données

▪ Solution :

Analyse sémantique comparée des données, préalable à la fédération (souvent avec la phase de traduction)

2 - Traduction des schémas

Permet une résolution de l'hétérogénéité syntaxique

- **Origine** : utilisation de **modèles différents** dans les **BD composantes**
- **Effet** : nécessite des traductions de tous les modèles vers tous les modèles

Solution : traduction de tous les schémas dans un **modèle commun** dit **canonique** ou **pivot**

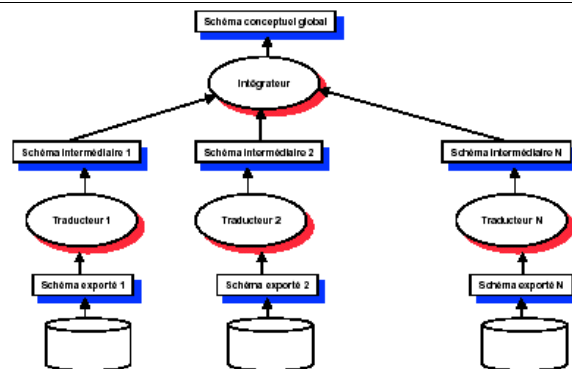
• Problématique :

- Le **modèle canonique** doit avoir un **pouvoir de modélisation** supérieur ou égal à ceux des modèles des BD composantes
- Nécessité de **compléter sémantiquement** des modèles de BD composantes qui seraient trop pauvres

• Choix du modèle canonique :

- **Entité – Association, Relationnel**
- **Objet**

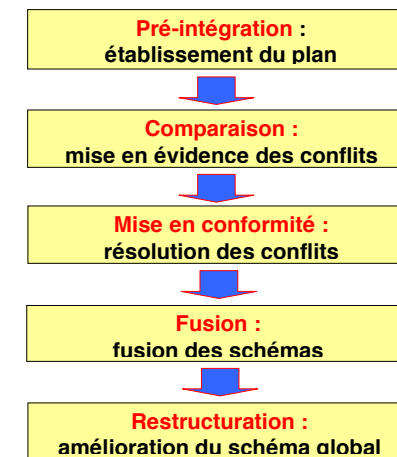
3 - Intégration des schémas



Procédure :

- Identifier les éléments de base qui sont liés
- Choisir la représentation la plus adéquate pour le schéma global
- Intégrer les éléments des schémas intermédiaires

Démarche d'intégration de schémas (1)



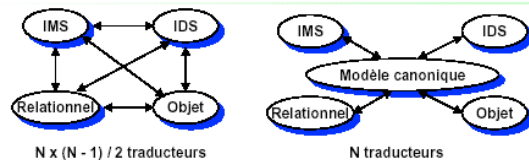
Démarche d'intégration de schémas (2)

- **Pré-intégration :**
 - mise en évidence des dépendances induites par les schémas
 - définitions des équivalences entre domaines
 - convention de désignation
- **Comparaison ou analyse - mise en évidence des conflits :**
 - de désignation (homonymie, synonymie)
 - structurels
 - de domaine
 - de contraintes
- **Mise en conformité : résolution des conflits :**
 - renommage pour les conflits de noms
 - étude au cas par cas pour les conflits structurels
- **Fusion des schémas - Qualités recherchées :**
 - complétude (pas de perte d'information)
 - minimalité (absence de redondance)
 - clarté
- **Restructuration - Amélioration du schéma global :**
 - pour l'essentiel recherche de clarté sans remise en cause des qualités recherchées

Cas de conflits dans l'intégration

- **Conflits d'attributs :**
 - **Conflit de nom :** -> renommage
 - **Conflit de type :** -> conversion
- **Attribut sans équivalent dans l'autre relation :**
 - **Attribut optionnel :** -> valeur nulle
 - **Attribut indispensable :** -> relation auxiliaire
- **Conflit de relation :**
 - **Conflit multi-attribut :** un attribut correspond à plusieurs dans l'autre relation (ex. adresse et N°, rue, code, ville) : -> utilisation d'un calcul sur les attributs (ex. extraction)
 - **Conflit de clé :**
 - pas la même clé : -> changement de clé
 - la clé d'une des relations composantes n'est pas une clé générale : -> génération d'une nouvelle clé par ajout d'un élément (ex. nom de commune pas déterminant au niveau national ajout du numéro de département au nom de la commune pour créer la nouvelle clé)
- ...

Traduction des schémas



Passage par un modèle canonique :

- Chaque site possède un traducteur local/canonique
- Chaque traducteur réalise 3 conversions :
 - **schéma local :** schéma équivalent en modèle canonique
 - **données locales :** données équivalentes en modèle canonique requêtes en langage du modèle canonique
 - **requêtes équivalentes en modèle local**

+ Développement d'un seul traducteur par SGBD

+ Simplification de la modélisation

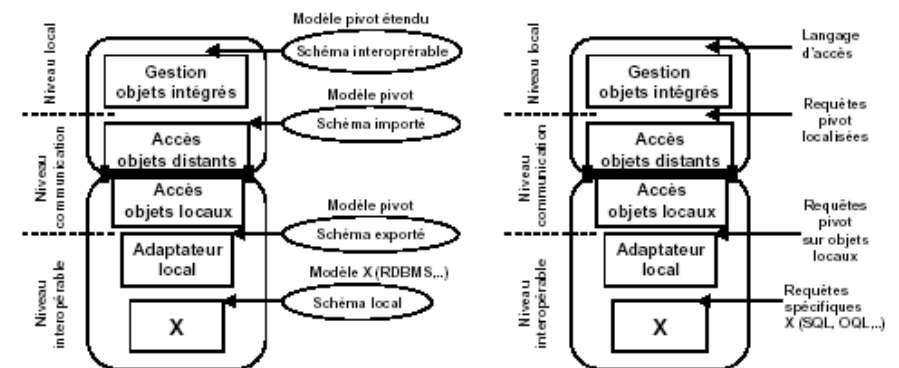
+ Transparence

- Difficulté de définir un modèle canonique aussi riche que les modèles locaux

- Temps de réponse accru pour les interrogations locales

Architecture de référence

Organisation des schémas & Niveaux des langages :



4. Accès aux données réparties

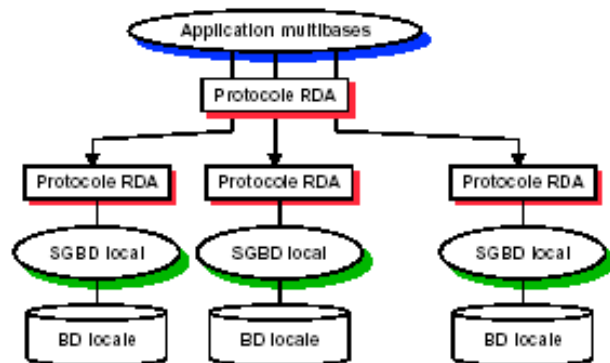
- Types d'accès aux données réparties
- RDA - Remote Data Access
- SQL-CLI
- ODBC - Open Data Base Connectivity
- JDBC - Open Data Base Connection
- Vues réparties
- SGBD répartis

Types d'accès aux données réparties

3 types d'accès :

- **Client / Multibases :**
 - RDA (Remote Data Access) - Standard ISO
 - DRDA (Distributed Relational DatabaseArchitecture) d'IBM (semble être en voie de disparition)
 - SQL-CLI (Call Level Interface) de l'Open Group
 - ODBC (Open Database Connectivity) de Microsoft
 - JDBC (Java Database Connection) de SUN
- **Vues réparties (sur BD fédérées)**
- **SGBD réparti**

RDA - Remote Data Access



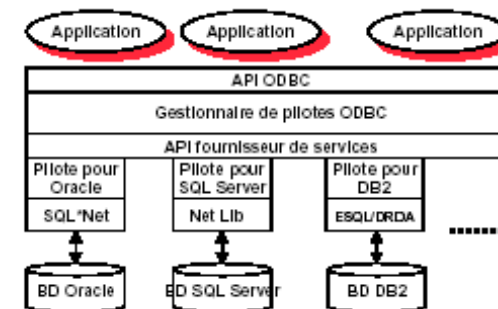
- Les usagers connaissent la localisation
- Si une jointure est nécessaire, elle doit être réalisée par l'application

SQL-CLI

1988 : consortium SAG (SQL Access Group) regroupant 44 éditeurs de SGBD avec pour objectif de définir :

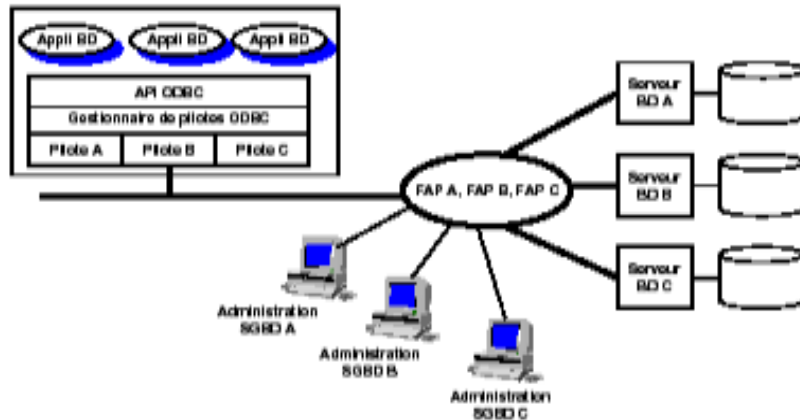
- un **standard d'interopérabilité entre clients et SGBD**;
- une **interface (CLI Call Level Interface) définissant un ensemble d'API** (Application Programming Interface) **communes pour les différents SGBD**.

Ex : Composants de la CLI ODBC de Microsoft



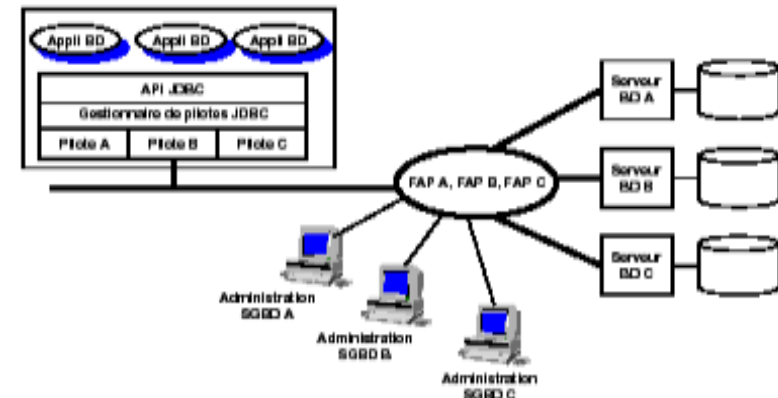
ODBC - Open Data Base Connectivity

- Spécification contrôlée par **Microsoft** et supportée par les principaux fournisseurs de SGBD
- **Difficulté** : niveau de SQL supporté, développement des pilotes



JDBC - Open Data Base Connection

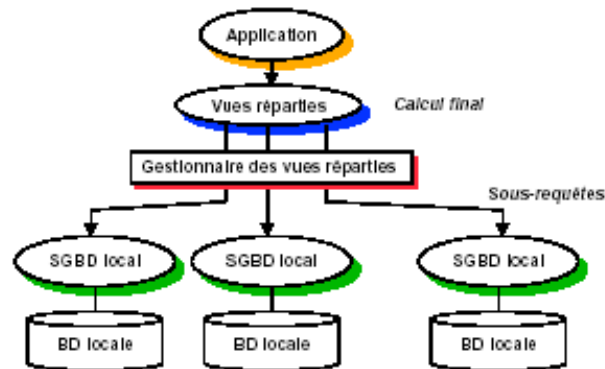
- Spécification commune à **Sun** et différents **fournisseurs de SGBD**
- **Difficulté** : risque potentiel d'intrusion dans des systèmes par l'intermédiaire du code mobile (byte-code)



Vues réparties (1)

Transparence de la localisation assurée par la définition des vues réparties :

- Les **jointures inter-bases** sont exécutées par le système
- Les **misés à jour** sont supportées au moyen des vues réparties
- Un **protocole de validation à 2 phases** est supporté



Vues réparties (2)

Définition de la vue répartie (sur le site 3) :

```
CREATE VIEW Accidenté-grave {N°personne, nom, prénom, adresse,gravité,
département, N° véhicule, marque, type}
AS SELECT P.N°personne, P.nom, P.prénom, P.adresse,
B.gravité, A.département,V.N° véhicule, V.marque, V.type
FROM S1.Personne P, S2.Blessé B, S2.Accident A, S1.Voiture V
WHERE P.N°personne = B.N°personne
AND B.gravité > « commotions »
AND A.N°véhicule = V.N°véhicule
AND A.N°.accident = B.N°accident
```

Requête sur la vue répartie (sur le site 3) : « **liste des blessés graves dans une voiture yyy de marque xxx dans la région parisienne** »

```
SELECT N°personne,nom,prénom,adresse
FROM Accidenté-grave
WHERE marque = « xxx »
AND type = « yyy »
AND département IN (75, 78, 91, 92, 93, 94, 95)
```

Vues réparties (3)

Fonctions réalisées par le gestionnaire des vues réparties :

- La **transformation de la requête sur les relations de base**
- La **décomposition de la requête en requêtes mono-site** :
 - Requête sur site 1 : `SELECT N°véhicule FROM Voiture`
 - Requête sur site 1 : `SELECT * FROM Personne ...`
 - Requête sur site 2 : `SELECT B.N°personne, A.N°véhicule FROM Blessé B, Accident A, ...`
- Le **contrôle de l'exécution des requêtes**
- **L'intégration du résultat** en effectuant les différentes opérations (dont les jointures)

Conclusion :

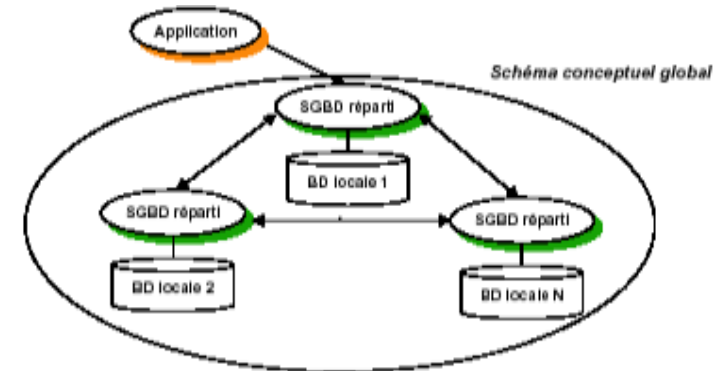
Le système apparaît à l'application comme un vrai SGBD réparti mais :

- Il y a toujours 3 **requêtes différentes** pour 2 sites
- La **totalité de la relation personne doit être transférée**

SGBD répartis

La transparence à la localisation est assurée par la définition de la base répartie :

- Les différentes opérations sont prises en charge par les différents SGBD
- Un protocole de validation à 2 phases est supporté



SGBD répartis

(RJ Chevance)

Schéma conceptuel de la base :

Personne (N° personne, nom, prénom, adresse, ...)
Voiture (N° véhicule, marque, type, ...)
Conducteur (N° personne, N° véhicule, NB_accidents,...)
Accident (N° accident, date, département, N° véhicule, N° personne, ...)
Blessé (N° accident, N° personne, gravité,)

Implémentation de la base :

- **Sites 75, 78, 91, 92, 93, 94, 95** : Bases préfectorales avec Voitures, Conducteur et Personne pour les voitures immatriculées dans le département (Personne, Voiture, Conducteur)
- **SAMU** : base SAMU de la région parisienne (Accident, Blessé)

Requête :

« liste des blessés graves dans une voiture type yyy de marque xxx dans la région parisienne »

SGBD réparti : Plan d'exécution répartie

(RJ Chevance)

Requête sur site SAMU :

```
SELECT B.N°personne, A.N°véhicule FROM Blessé B, Accident A
WHERE B.N°accident = A.N°accident AND B.gravité > « commotions »
AND A.département IN (75, 78, 91, 92, 93, 94, 95) INTO temp1
SEND temp1 to S75, S78, S91, S92, S93, S94, S95
```

Requêtes sur S75, S78, S91, S92, S93, S94, S95 :

```
RECEIVE temp1 FROM SAMU
SELECT P.nom, P.prénom FROM Personne P, temp1 T, Voiture V
WHERE P.N°personne = T.N°personne AND T.N°véhicule = V.N°véhicule
AND V.marque = « xxx » AND V.type = « yyy » INTO temp2.i
SEND temp2.i TO Interrogation
```

Requête sur site Interrogation :

```
RECEIVE temp2.75 FROM S75
.....
RECEIVE temp2.95 FROM S95
UNION temp2.75, temp2.78, .....temp2.95
INTO résultat
```

- Le SGBD réparti a pris en charge tous les problèmes liés à la répartition
- Les transferts sont minimisés : seuls les N° des blessés et des véhicules sont transférés

SGBD réparti : les problèmes & solutions

- **Validation à 2 phases** : utilisé dès qu'une mise à jour s'adresse à plus d'un site ou à plus d'une base sur un même site
- **Verrouillage à 2 phases** : pour assurer la sérialisation des transactions (phase d'acquisition des verrous puis phase de relâchement des verrous)
- **Détection des deadlocks**
- **Partitionnement et placement des données :**

Objectifs :

- Réduction de la charge (accès aux données, communication, espace de recherche)
- Équilibrage de charge
- Accroissement du travail utile (e.g. effet de cache)

Solutions :

- Partitionnement vertical
- Partitionnement horizontal

SGBD répartis du marché

Ingres/STAR

- le premier

IBM DB2

- 3 produits : DataLinks, DataJoiner, DataPropagator

Informix

- 2 produits : Virtual Table Interface, Enterprise Replication

Oracle

- 2 produits : Transparent Gateways, Database Replication

Sybase

- Replication Server

...

5. Réplication

- Objectifs de la réplication
- Modes de réplication

Objectifs de la réplication

Objectifs de la réplication :

- Améliorer la **disponibilité** des données
- Respect de l'**intégrité** des données
- Améliorer les **performances**
- **Administration centralisée**

Difficultés de la réplication :

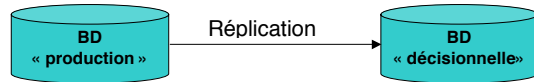
- **Synchronisation des copies**
- **Transparence de la gestion**

Copies :

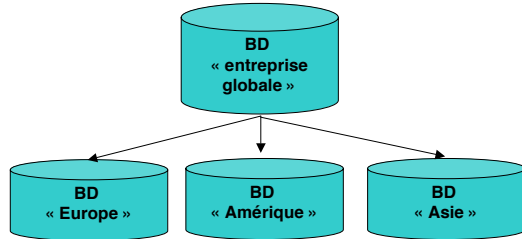
- **Copie Primaire** (ou Maître ou Source) : reçoit les mises à jour
- **Copie Secondaire** (ou Esclave ou Cible) :
 - en **consultation seulement**
 - peut être désigné Primaire en cas d'arrêt de la copie primaire

La Réplication : Exemples (1)

1. Mouvement d'information (OLTP DSS) :

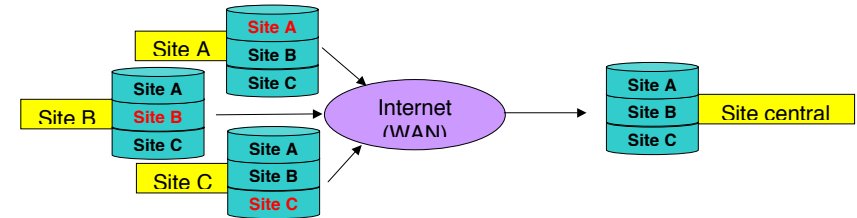


2. Distribution d'information :

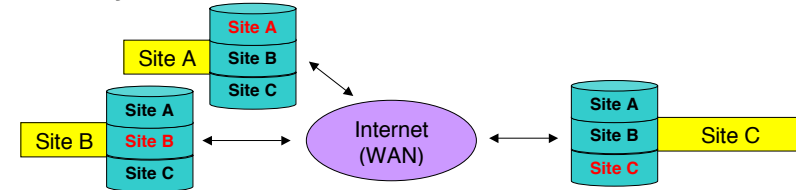


La Réplication : Exemples (2)

3. Consolidation d'information :



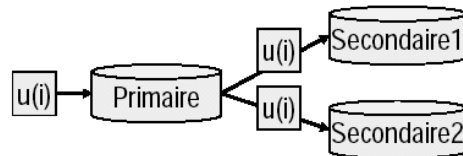
4. Mises à jour sans conflits :



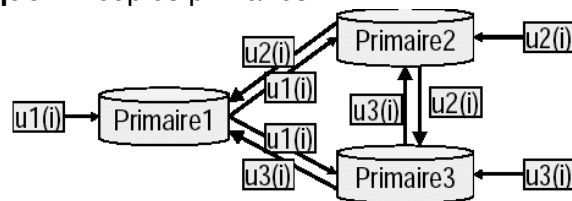
Modes de réplication

2 Modes de réplication :

- **Asymétrique** : une copie primaire / N copies secondaires



- **Symétrique** : N copies primaires



Modèles de réplication : modes de propagation des mises à jour

2 modes de propagation des Mises à Jour (source -> cible):

- **Synchrone** : **Mises à jour globales dans une même transaction**

+ : Cohérence forte : Contrôle de Concurrence (Verrouillage) & Terminaison : 2 Phase Commit

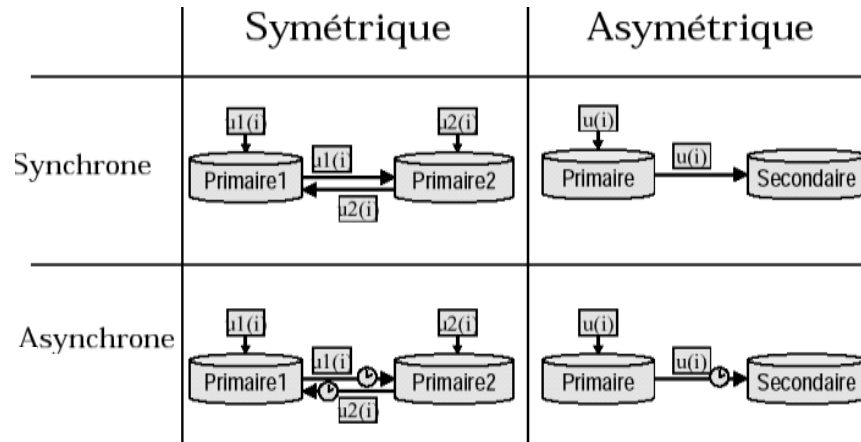
- : Ralentit la transaction et le débit (mise en œuvre de la validation à 2 phases)

- **Asynchrone** : **Mises à jour dans des transactions différées**

+ : incidence faible sur les performances

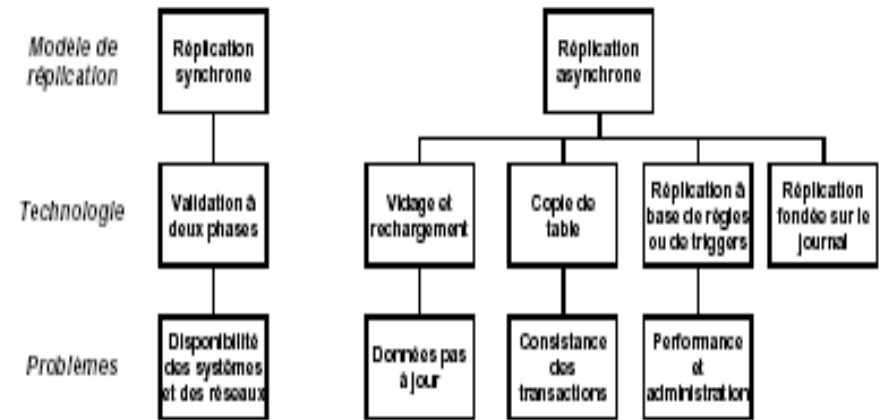
- : Nécessité de mise à niveau de la ou des copie(s) en cas de reprise

Modèles de réplication



(D. Donsez)

Modèles de réplication : technologies et problèmes



(D'après RJ Chevance)