Module BDR
Master d'Informatique (SAR)

Cours 5- Bases de données réparties

Stéphane Gançarski
Stephane.Gancarski@lip6.fr

Bases de Données Réparties

- Définition
- Conception
- Décomposition
- Fragmentation horizontale et verticale
- Outils d'interface SGBD
  - extracteurs, passerelles
- Réplication
- SGBD répartis hétérogènes

BD réparties (1)

- Principe : BD locales, avec des accès locaux rapides, et possibilité d'accéder aux autres SGBD du réseau (accès globaux)
- Plusieurs niveaux d'intégration :
  - Client/serveur : BD centralisée, seuls certains traitements (interface, p.ex.) sont locaux.
  - Accès distant (Remote Data Access)
  - Vues réparties : extension du mécanisme de vues pour définir des vues sur plusieurs sites.
  - médiateurs
  - BD réparties/fédérées

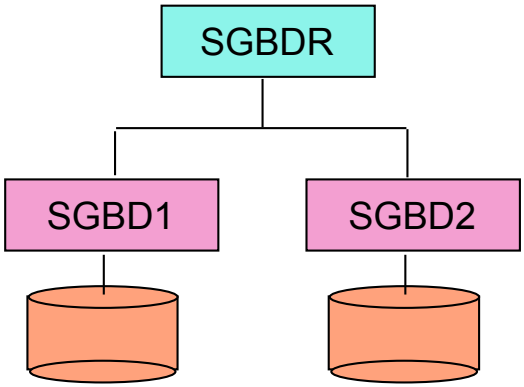
BD Réparties (2)

- BD réparties :
  - Plusieurs bases sur plusieurs sites, mais une seule BD « logique ».
  - Fédérée : intègre des bases et des schémas existant
  - Réparti « pur » : conçue répartie. Pas d'accès locaux

- Les ordinateurs (appelés sites) communiquent via le réseau et sont faiblement couplés (pas de partage de MC, disk, au contraire de *BD parallèles*)

- Chaque site
  - contient des données de la base,
  - peut exécuter des transactions/requêtes locales et
  - participer à l'exécution de transactions/requêtes globales

SGBD réparti



Rend la répartition (ou distribution) *transparente*

- dictionnaire des données réparties
- traitement des requêtes réparties
- gestion de transactions réparties
- gestion de la cohérence et de la sécurité

Paramètres à considérer

- Coût et temps de communication entre deux sites
  - Accès réseau (longue distance, WAN, MAN) beaucoup plus coûteux que accès disque
- Fiabilité
  - fréquence des pannes des sites, du réseau (cf. P2P)
- Accessibilité aux données
  - accès aux données en cas de panne des sites, du réseau.
  - accès aux sites les moins encombrés, les plus puissants
  - Réplication : exploiter la localité

Evaluation de l'approche BDR

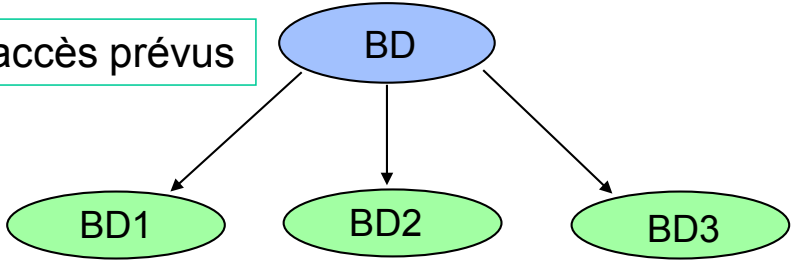
- avantages
  - extensibilité
  - partage des données hétérogènes et réparties
  - performances avec le parallélisme
  - Disponibilité et localité avec la réplication

- inconvénients
  - administration complexe
  - complexité de mise en œuvre et de développement
  - distribution du contrôle
  - difficulté de migration
  - surcharge (l'échange de messages augmente le temps de calcul)

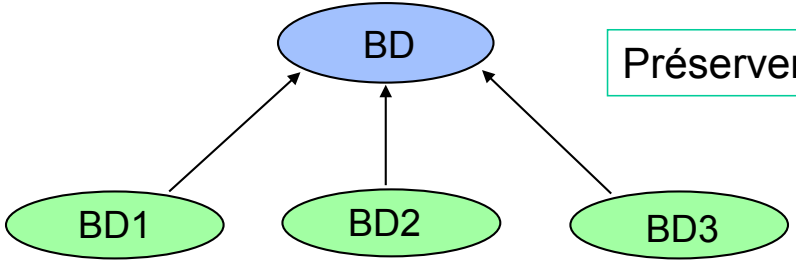
Migration vers une BDR

Décomposition en BD locales : réparti « pu r »

En fonction des accès prévus

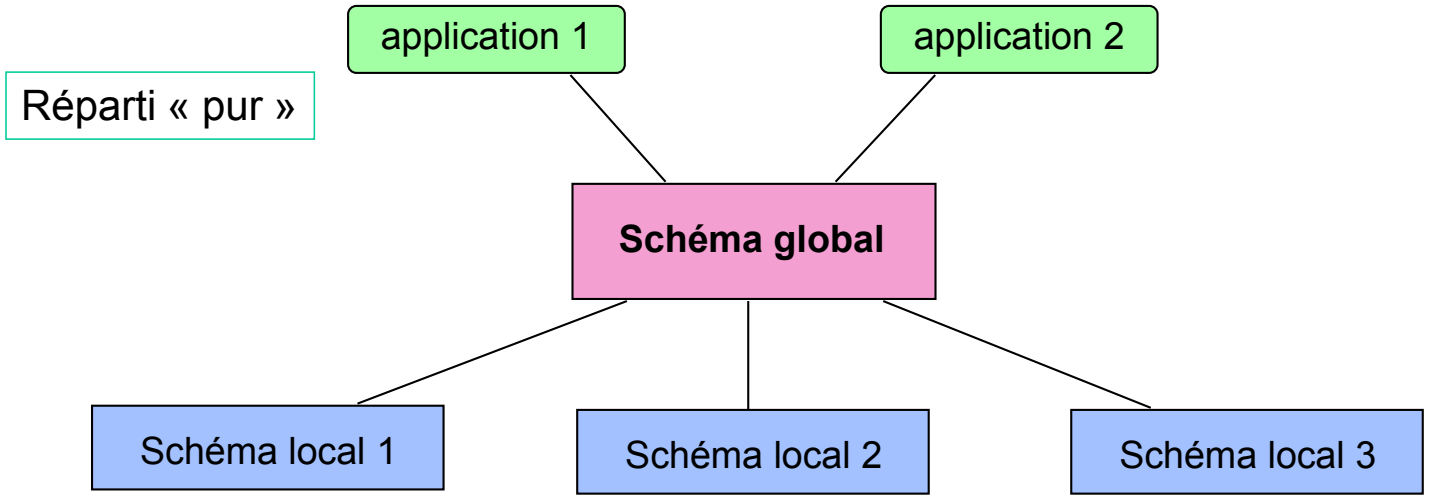


Intégration logique des BD locales existantes (fédérée, médiateur)



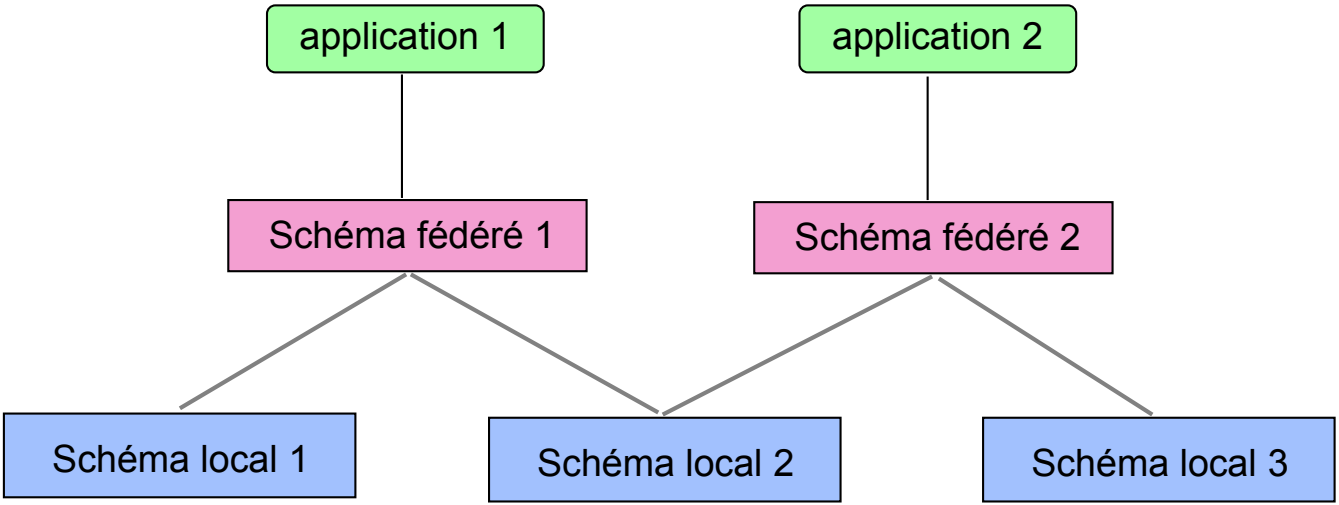
Préserver l'autonomie

Architecture de schémas



– indépendance applications/BDR
– schéma global lourd à gérer

Architecture fédérée



moyen contrôlé de migration

Schéma global

schéma conceptuel global

- donne la description globale et unifiée de toutes les données de la BDR (e.g., des relations globales)
- indépendance à la répartition

schéma de placement

- règles de correspondance avec les données locales
- indépendance à la localisation, la fragmentation et la duplication

Le schéma global fait partie du dictionnaire de la BDR et peut être conçu comme une BDR (dupliqué ou fragmenté)

Exemple de schéma global

Schéma conceptuel global

Client (nclient, nom, ville)

Cde (ncde, nclient, produit, qté)

Client1 ∩ Client2 = Client « répli pur »
= Ø « fragmenté pur »

Schéma de placement

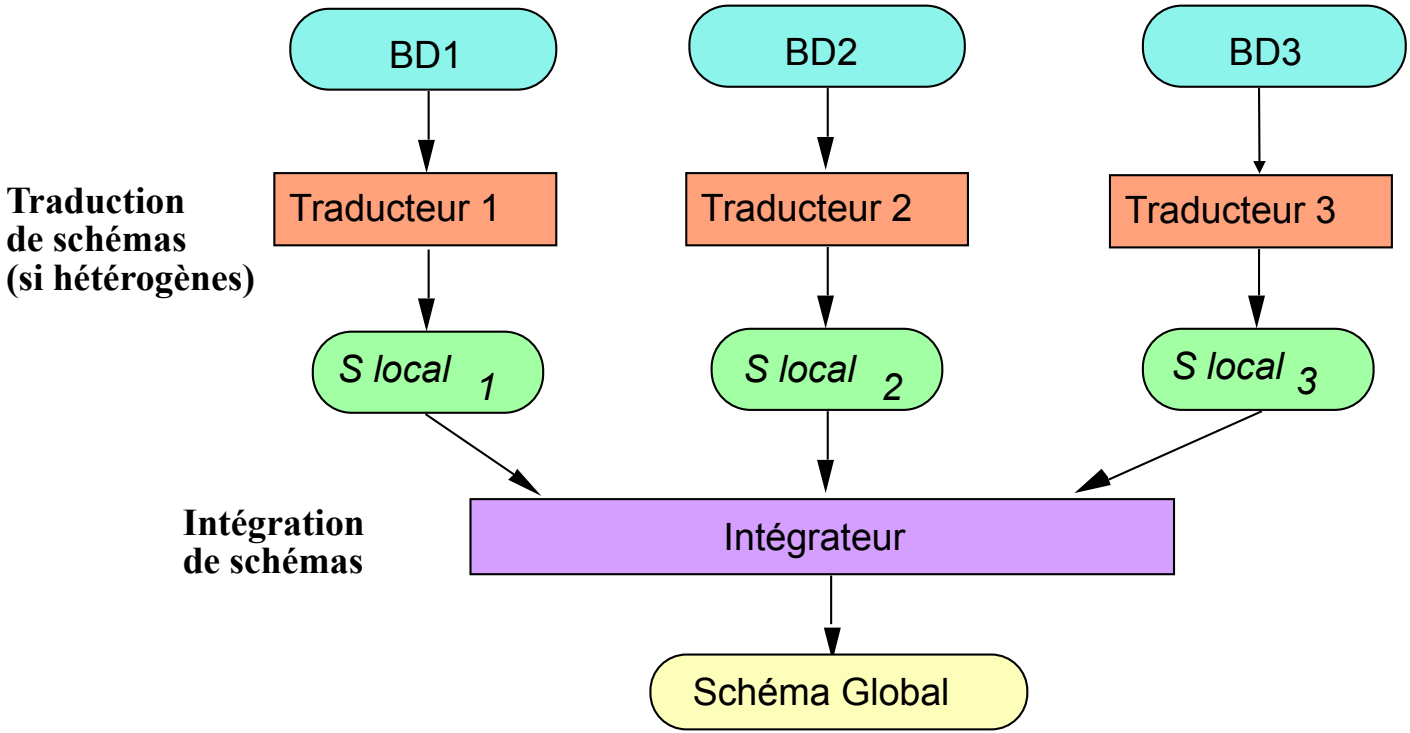
Client = Client1 @ Site1 U Client2 @ Site2

Cde = Cde @ Site3

Cde ni fragmenté ni répliqué

Conception d'une BDR par intégration

Approche médiateur



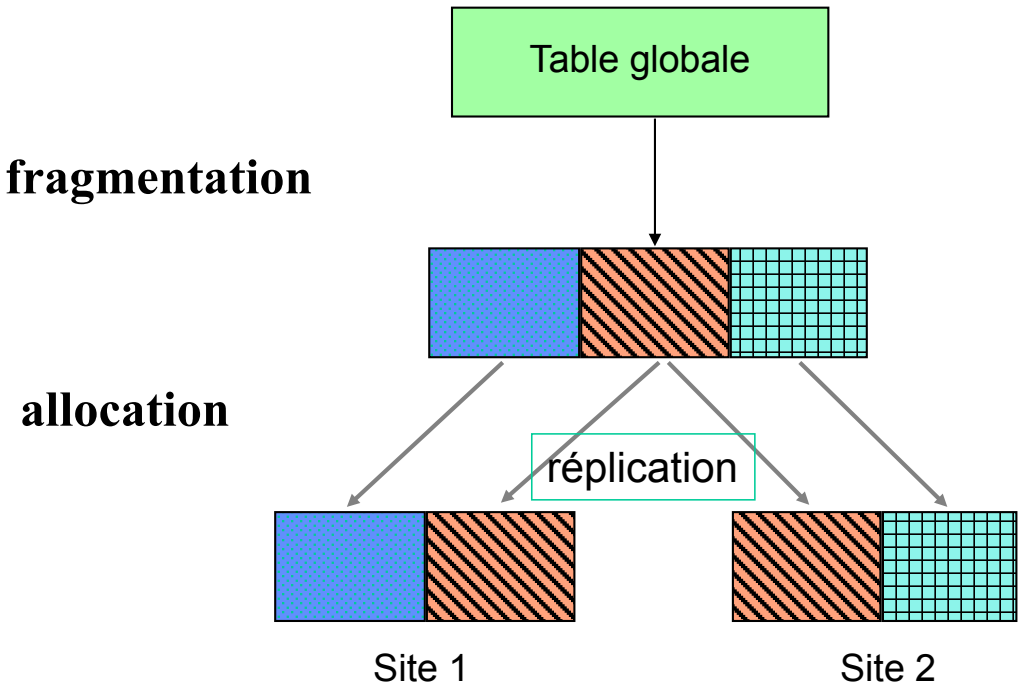
Intégration de schémas

1. pré-intégration
   - Les schémas sont transformés pour les rendre plus homogènes
   - *identification* des éléments reliés (e.g. domaines équivalents) et établissement des règles de conversion (e.g. 1 inch = 2,54 cm)
   - Pbs : hétérogénéité des modèles de données, des puissances d'expression, des modélisations

2. comparaison
   - *identification* des conflits de noms (synonymes et homonymes) et des conflits structurels (types, clés, dépendances)

3. conformance
   - *résolution des* conflits de noms (renommage) et des conflits structurels (changements de clés, tables d'équivalence)
   - Définition de règles de traduction entre le schéma intégré et les schémas initiaux.

Conception par décomposition



Objectifs de la décomposition

fragmentation

- trois types : horizontale, verticale, mixte
- performances en favorisant les accès (et traitements) locaux
- équilibrer la charge de travail entre les sites (parallélisme)

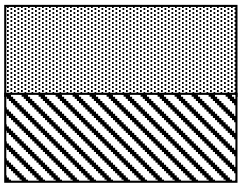
Trop fragmenter : BD éclatée, nombreuses jointures réparties

duplication (ou réplication)

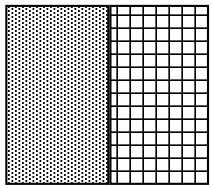
- favoriser les accès locaux
- augmenter la disponibilité des données

Trop répliquer : surcoût de maintenir cohérence des répliques

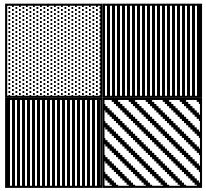
Types de Fragmentation



horizontale



verticale



mixte

Fragmentation correcte

Complète

- chaque élément de R doit se trouver dans un fragment

Reconstructible

- on doit pouvoir recomposer R à partir de ses fragments (ressemble à décomposition de schéma vue en Li341 pour fragmentation verticale)

[Disjointe] /*si on veut éviter réplication pour cohérence */

- chaque élément de R ne doit pas être dupliqué

Fragmentation Horizontale

Fragments définis par sélection

Client1 = Client where ville = Paris
Client2 = Client where ville not= Paris

Inférence : correcte

Reconstruction

Client =Client1 U Client2

Client

nclient	nom	ville
C 1	Dupont	Paris
C 2	Martin	Lyon
C 3	Martin	Paris
C 4	Smith	Lille

Client1

nclient	nom	ville
C 1	Dupont	Paris
C 3	Martin	Paris

Client2

nclient	nom	ville
C 2	Martin	Lyon
C 4	Smith	Lille

Fragmentation Horizontale Dérivée

Fragments définis par jointure

Cde1 = Cde where

Cde.nclient =Client1.nclient

Cde2 = Cde where

Cde.nclient = Client2.nclient

Cde

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Reconstruction

Cde = Cde1 U Cde2

Cde1 en fonction de cde et client1: *Semi-jointure*

Cde1

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20

Cde2

ncde	nclient	produit	qté
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Fragmentation Verticale

Fragments définis par projection

Cde1 = Cde (ncde, nclient)

Cde2 = Cde (ncde, produit, qté)

Cde

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Reconstruction

Cde = [ncde, nclient, produit, qté]

where Cde1.ncde = Cde2.ncde

Cde1

ncde	nclient
D 1	C 1
D 2	C 1
D 3	C 2
D 4	C 4

Cde2

ncde	produit	qté
D 1	P 1	10
D 2	P 2	20
D 3	P 3	5
D 4	P 4	10

Allocation des Fragments aux Sites

Non-dupliquée

- partitionnée : chaque fragment réside sur un seul site

Dupliquée

- chaque fragment sur un ou plusieurs sites
- maintien de la cohérence des copies multiples : coûteux
- **(le fameux) Compromis Lecture/écriture**:

  + le ratio Lectures/màj est > 1, + la duplication est avantageuse

Allocation de Fragments

Problème: Soit

F un ensemble de fragments

S un ensemble de sites

Q un ensemble d'applications et leurs caractéristiques

trouver la distribution "optimale" de *F* sur *S*

Optimum

- coût minimal de communication, stockage et traitement
- Performance = temps de réponse ou débit

Solution

- allouer une copie de fragment là où le bénéfice est supérieur au coût

Exemple d'Allocation de Fragments

Client1

nclient	nom	ville
C 1 C 3	Dupont Martin	Paris Paris

Client2

nclient	nom	ville
C 2 C 4	Martin Smith	Lyon Lille

Cde1

ncde	client	produit	qté
D 1 D 2	C 1 C 1	P 1 P 2	10 20

Site 1

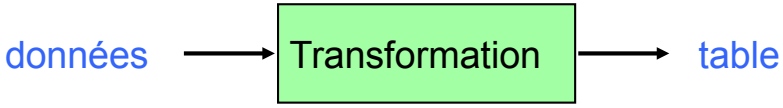
Cde2

ncde	client	produit	qté
D 3 D 4	C 2 C 4	P 3 P 4	5 10

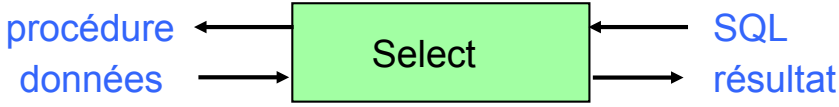
Site 2

Outils d'interface SGBD

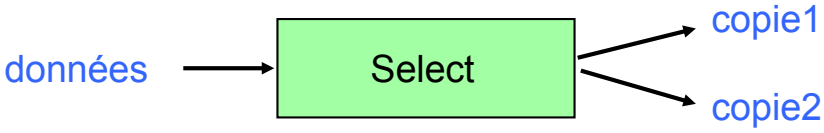
Extracteur



Passerelle



Réplicateur



Extracteurs et Passerelles

- Fonctions
  - définition des procédures de transformation (dictionnaire) et exécution dans l'environnement cible
  - conversion de formats et de valeurs
  - filtrage et fusion de fichiers ou de tables
  - données calculées et résumés
- Fournisseurs indépendants
  - Evolutionary Technology Inc. (ETI), Information Builders Inc. (IBI), Prism, Carleton, etc.
  - extracteurs indépendants entre les données sources et les outils cibles
- Editeurs de SGBD
  - Oracle, DB2, Sybase, etc.
  - passerelles entre le SGBD et les données sources

La réplication

Objectifs

Fonctions

Modèles d'appartenance

- fixe, dynamique ou partagé

Détection des modifications

Produits

- Data Propagator Relational (DProp R) d'IBM
- Informix OnLine
- CA-OpenIngres Replicator
- Oracle RAC
- Sybase Replication Server
- Multi Server Option d'Afic Technologies

Objectifs de la réplication

+ Accès simplifié, plus performant pour les lectures
+ Résistance aux pannes
+ Parallélisme accru
+ Evite des transferts

- Overhead en mise à jour
- Cohérence des données
- Toujours bien si on privilégie les lectures et/ou si peu de conflit entre màj

Objectifs de la réplication

Problème : comment partager des données entre p sites ?

- Solution 1 : sans duplication
  - stockage sur un site et accès réseau depuis les autres sites
  - problèmes de performances et de disponibilité

- Solution 2 : duplication synchrone
  - propagation des mises à jour d'un site vers les autres par une transaction multi-site avec validation 2PC ou communication de groupe
  - problèmes liés au 2PC : bloquant et cher. Communication de groupe seulement pour LAN

- Solution 3 : réplication asynchrone
  - Non bloquant mais uniquement cohérence à terme
  - Contrôle de la fraîcheur, traitement spécifique des requêtes read-only

Aussi mono-maître,multi-maîtres

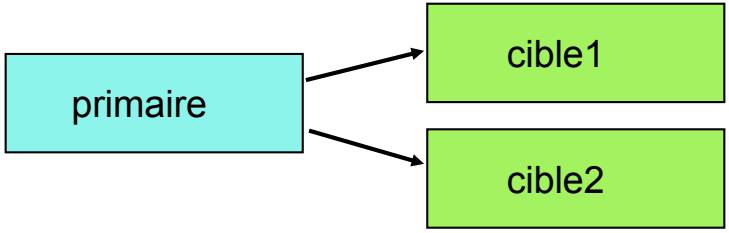
Fonctions d'un réplicateur

- Définition des objets répliqués
  - table cible = sous-ensemble horizontal et/ou vertical d'une ou p tables
- Définition de la fréquence de rafraichissement
  - immédiat (après mise à jour des tables primaires)
    - Synchrone (avant validation sur primaire)
    - Asynchrone (transaction séparée)
  - à intervalles réguliers (heure, jour, etc.)
  - à partir d'un événement produit par l'application
- Rafraichissement
  - complet ou partiel (propagation des modifications)
  - push (primaire -> cibles) ou pull (cible -> primaire)
- Support des données hétérogènes via les passerelles

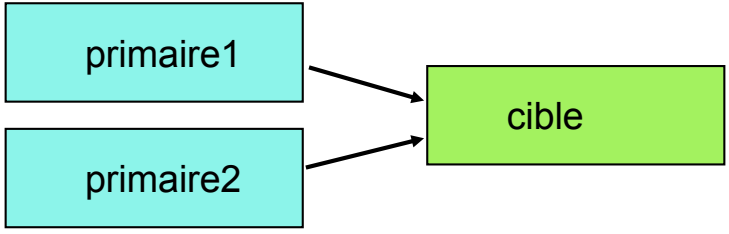
Modèle d'appartenance fixe (monomaître)

Seul le site primaire peut mettre à jour, les sites cibles ne recevant que des copies en lecture

Diffusion



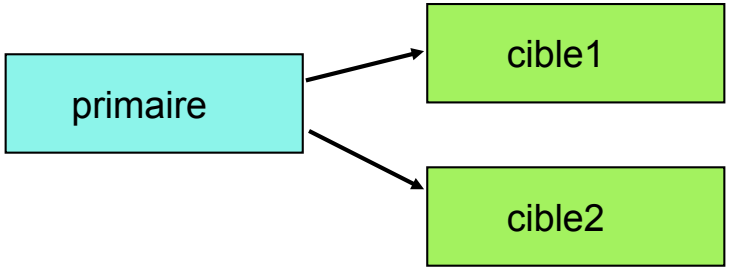
Consolidation



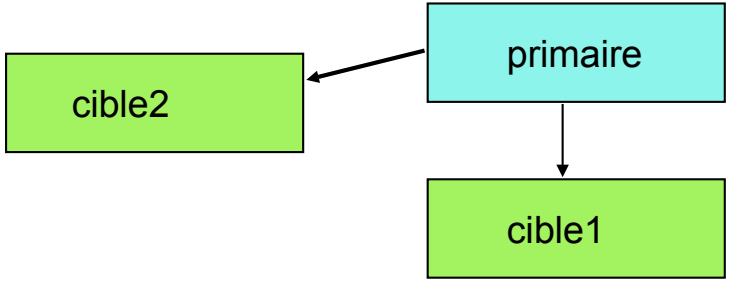
Modèle d'appartenance dynamique (monomaître)

Le site primaire peut être différent au cours du temps, en fonction d'événements: panne d'un site, état de la données, etc.

Appartenance à l'instant t1



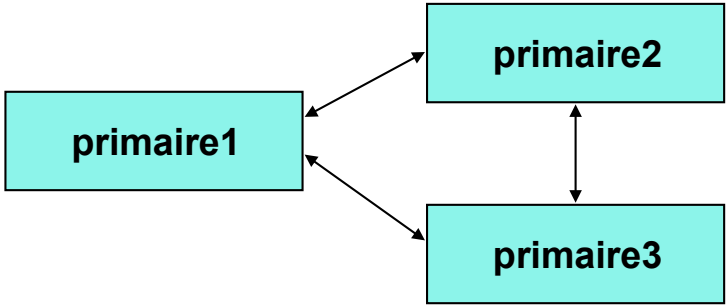
Appartenance à l'instant t2



Modèle d'appartenance partagée (multimaître)

Une donnée appartient à plusieurs sites, qui peuvent chacun mettre à jour et diffuser aux autres sites

- **augmente la disponibilité**
- **peut produire des conflits, qui doivent être détectés et résolus**

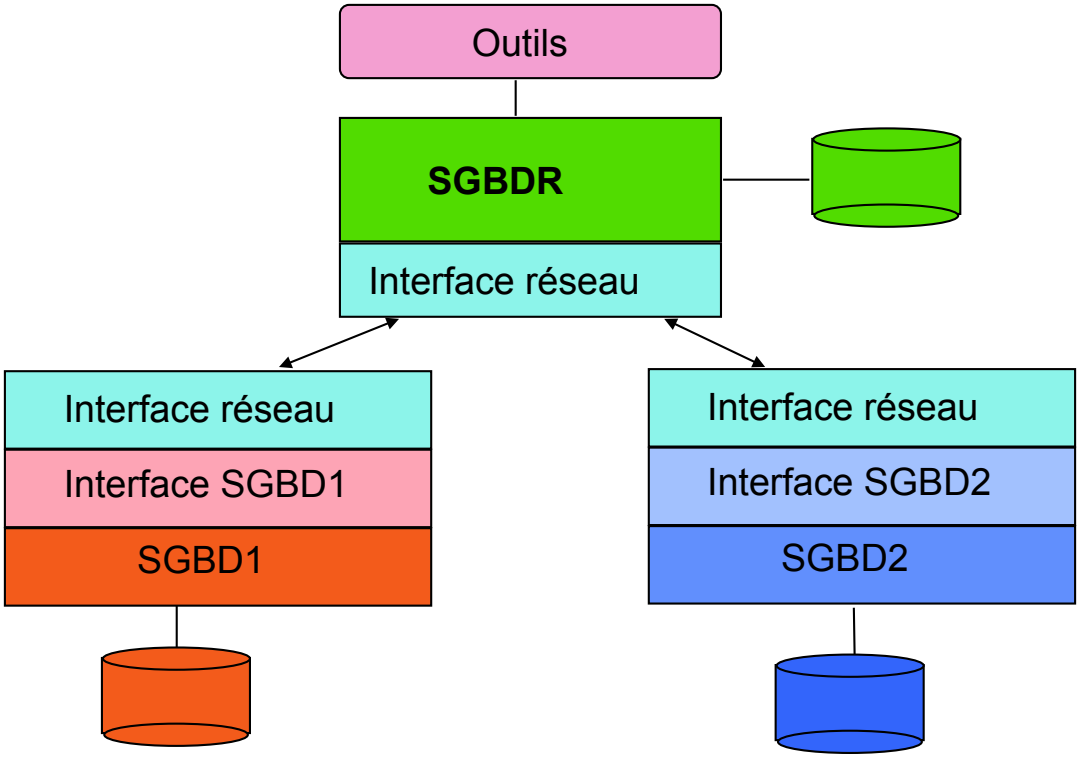


Détection des modifications

- Solution 1 : utilisation du journal
  - les transactions qui modifient écrivent une marque spéciale dans le journal
  - détection périodique en lisant le journal, indépendamment de la transaction qui a modifié
  - modification de la gestion du journal

- Solution 2 : utilisation de triggers
  - la modification d'une donnée répliquée déclenche un trigger
  - mécanisme général et extensible
  - la détection fait partie de la transaction et la ralentit
- Solution 3: détecter les conflit potentiels (a priori)
  - Parser le code (pas possible pour transactions interactives)
  - Graphe d'ordonnancement global

SGBD réparti hétérogène



Produits

- SGBD relationnels
  - Oracle, Ingres, Sybase, DB2, Informix
- DataJoiner (IBM)
  - basé sur DB2
- VirtualDB (Enterworks)
  - basé sur GemStone, vue objet des tables
- Open Database Exchange (B2Systems)
- Disco (GIE Dyade Bull-Inria)
  - utilisé par Kelkoo

Oracle/Star

- SGBD Oracle
  - gestion du dictionnaire de la BDR
- SQL*Net
  - transparence au réseau
  - connexion client-serveur, login à distance automatique
  - évaluation de requêtes réparties
  - validation en deux étapes et réplication
- SQL*Connect : passerelle vers les bases non-Oracle

Database link

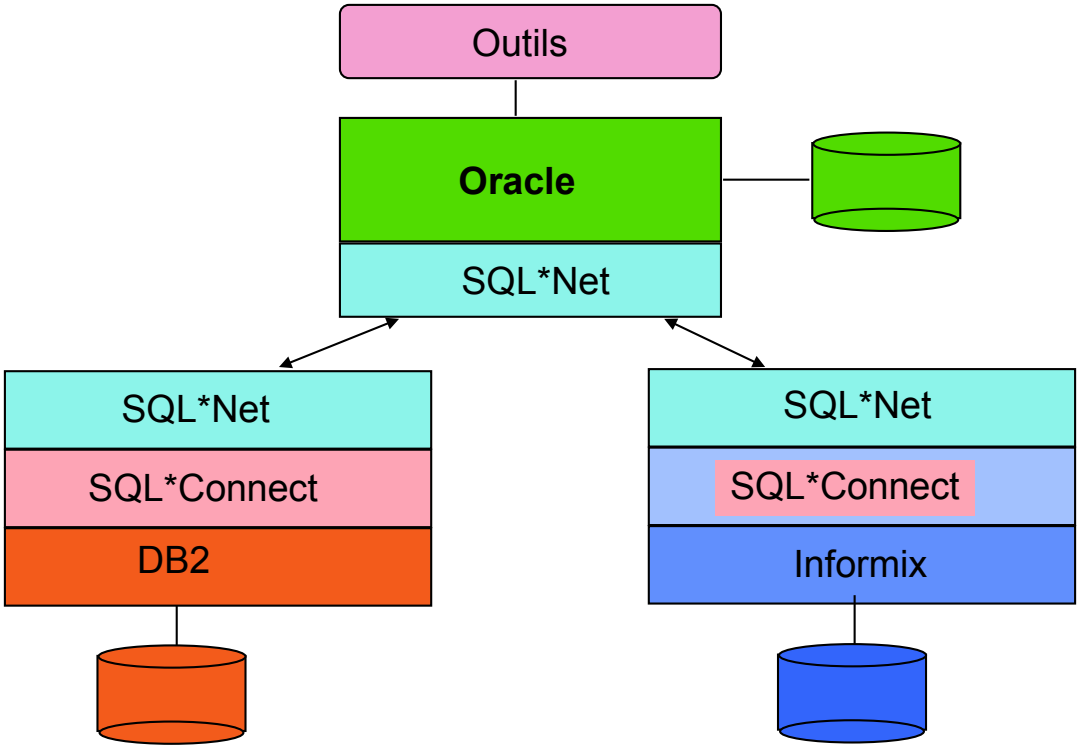
Lien à une table dans une BD distante specifié par :

- nom de lien
- nom de l'utilisateur et password
- chaîne de connexion SQL*Net (protocole réseau, nom de site, options, etc…)

Exemple

```
CREATE DATABASE LINK empParis
CONNECT TO toto  IDENTIFIEDBY monPW
USING Paris.emp
```

Oracle/Star : architecture



Conclusions et perspectives

Applications classiques

- décisionnel (data warehouse)
- transactionnel

Applications nouvelles

- intégration de données du Web
  - grand nombre de sources
  - hétérogénéité très forte
- intégration des données semistructurées (HTML, XML)
- intégration de la recherche documentaire
- Intégration de services Web (ex. agence de voyage)