

Module BDR Master d'Informatique (SAR)

Cours 10- Gestion de données réparties

Stephane Gançarski
Stephane.Gancarski@lip6.fr

1

Plan

- Intégration de données
- Architectures
 - Approche matérialisée (Entrepôt de données)
 - Approche virtuelle (média-teurs)
 - Architectures P2P

2

Intégration de données

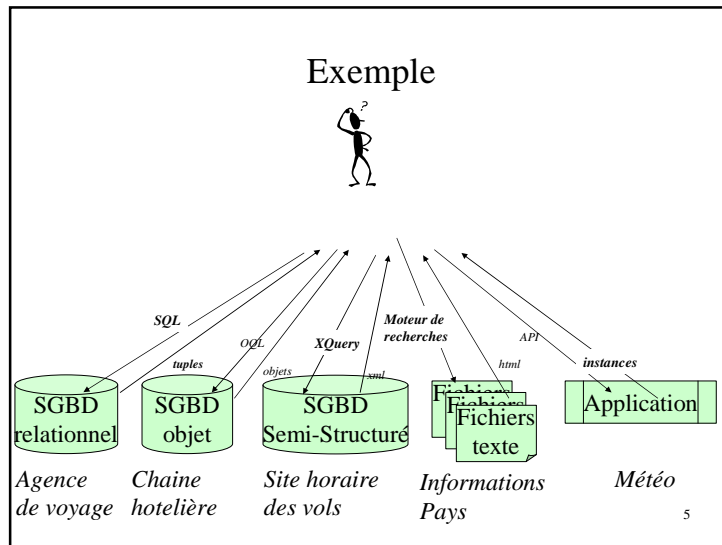
- Contexte
- Caractéristiques
- Processus d'intégration de données

3

Contexte

- Sources d'informations nombreuses et très diversifiées (SGBD R, SGBDO, XML, fichiers texte, pages Web, etc.)
- Différents modes de consultation
 - Langages et modes de requêtes différents (SQL, moteurs de recherche, programme d'applications...)
 - Différentes façons de répondre (différentes présentations du résultat) : pages Web, tableurs, relations...
- Différents interactions avec la source
 - Protocoles de communication (JDBC, ODBC, IIOP)
 - Différentes interfaces

4



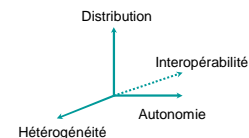
Intégration de données

Fournir

- un accès (requêtes, éventuellement mises à jour)
- uniforme (les sources sont transparentes à l'utilisateur)
- à des sources (pas seulement des BD)
- multiples (même 2 est un problème)
- autonomes (sans affecter le comportement des sources)
- hétérogènes (différents modèles de données, schémas)
- structurées (ou au moins semi-structurées)

Caractéristiques des sources

- Distribution
- Hétérogénéité
- Autonomie
- Interopérabilité



Distribution

- Les données sont stockées sur des supports répartis géographiquement.
- Offre disponibilité et amélioration des temps d'accès.
- Pbs:
 - Localiser la (ou les) source(s) contenant les données pertinentes.
 - Tenir compte de la puissance des sources et de leur charge
 - Les sources peuvent être temporairement indisponibles

Hétérogénéité

- L'hétérogénéité concerne les données, les modèles et les langages.
- Système homogène :
 - même logiciel gérant les données sur tous les sites
 - même modèle de données
- Système hétérogène :
 - n'adhère pas à toutes les caractéristiques d'un système homogène
 - langages de programmation et d'interrogation différents, modèles différents, SGBD différents

9

Autonomie

- Conception : sources locales avec des
 - modèles de données propres,
 - langage d'interrogation
 - Interprétation sémantique des données, contraintes, fonctions ...
- Communication : les sources de données locales décident quand et comment répondre aux questions d'autres sources
- Exécution : pas d'information provenant des sources locales sur
 - l'ordre d'exécution des transactions locales ou des opérations externes
 - pas de distinction entre les opérations locales et globales
- Association :
 - connexion et déconnexion des sources
 - partage de données et des fonctions

10

Interopérabilité

- Systèmes interopérables :
 - échange de messages et de requêtes
 - partagent les fonctions
 - Communiquent même avec des composants internes incompatibles
- Propriétés fondamentales à tout système interopérable :
 - Distribution
 - Hétérogénéité
 - Autonomie

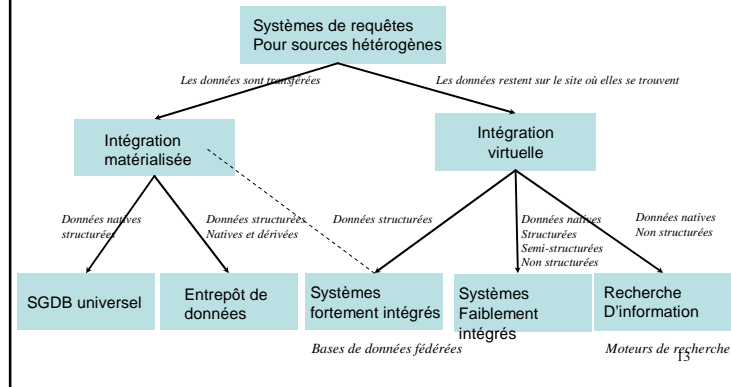
11

Architectures d'intégration

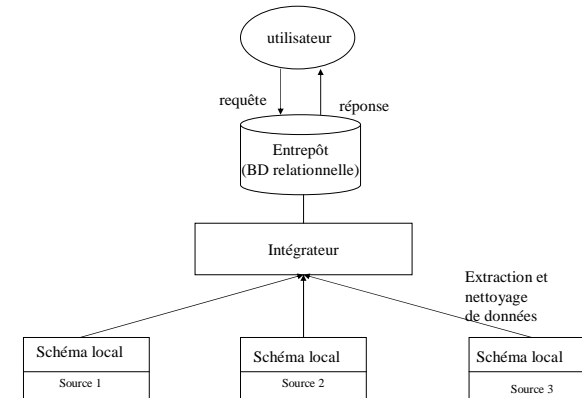
- Intégration matérialisée
 - Les données provenant des sources à intégrer sont stockées sur un support spécifique (entrepôt de données).
 - L'interrogation s'effectue comme sur une BD classique (relationnelle).
- Intégration virtuelle
 - Les données restent dans les sources
 - Les requêtes sont faites sur un schéma global, puis décomposées en sous-requêtes sur les sources. Les différents résultats des sources de la requête sont combinés pour former le résultat final.

12

Architectures d'intégration : Localisation de données

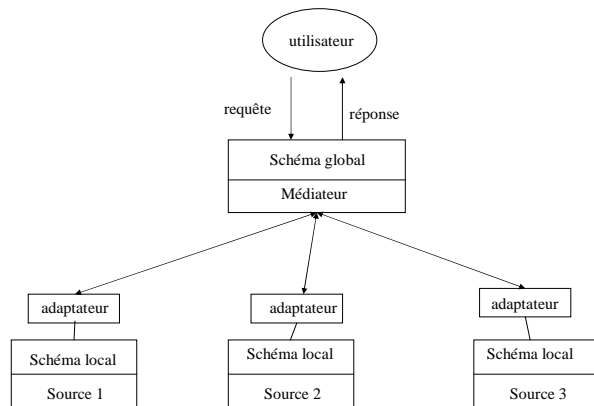


Architecture d'entrepôt de données



14

Architecture de médiateur



15

Matérialisé vs. virtuel

- Architectures matérialisées
 - Bonnes performances
 - Données pas toujours fraîches
 - Nettoyage et filtrage des données
- Architectures virtuelles
 - Les données sont toujours fraîches
 - Traitement de requêtes peut être coûteux
 - Défi principal : performances

16

Entrepôts de données

17

Motivations

- Réconciliation sémantique
 - Dispersion des sources de données au sein d'une entreprise
 - Différents codage pour les mêmes données
 - L'entrepôt rassemble toutes les informations au sein d'un unique schéma
 - Conserve l'historique des données
- Performance
 - Les données d'aide à la décision nécessitent une autre organisation des données
 - Les requêtes complexes de l'OLAP dégradent les performances des requêtes OLTP.
- Disponibilité
 - La séparation augmente la disponibilité
 - Une bonne façon d'interroger des sources de données dispersées
- Qualité des données

18

Systèmes légués

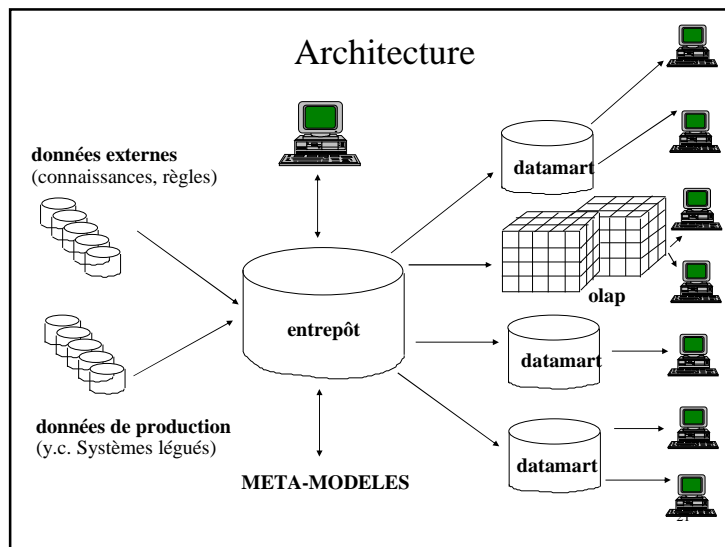
- *gros système, critique*, sur environnement ancien. Souvent *peu documenté*. *Interactions* entre les différents modules *peu claires*. *Très cher* à maintenir.
- Il faut l'intégrer (migration) au système actuel (Entrepôt) = architecture cible.
- Contraintes : migration sur place, garder opérationnel, corriger et améliorer pour anticiper, le moins de changements possibles (diminuer le risque), flexible sur les évolutions futures, utiliser les technologies modernes.
- Approche classique : tout réécrire dans l'architecture cible
 - promesses à tenir dans des conditions changeantes
 - problème de transfert de très gros fichiers (plusieurs jours) dans système critique
 - gros projet, retard mal vus, risque d'abandon
- Approche incrémentale :
 - isoler des sous-systèmes à migrer
 - établir des *passerelles* pour que les modules déjà migrés puissent communiquer avec les modules encore dans le système légué (traducteur de requêtes et de données).
 - coordonner les mises à jour pour garder la cohérence.

19

Bases de Données/Entrepôts de données

	BD- OLTP	Entrepôts
Objectif	collecte de données opérations au jour le jour	consultation et analyse
Utilisateurs	un département (Employé)	transversal (Gestionnaire)
Types de données	données de gestion (données courantes)	données d'analyse (données historiques)
Informations	détaillées	détaillées + agrégées
n-uplets accédés	dizaines	millions
Opérations	requêtes simples, pré-déterminées sélections et mises à jour nombreuses transactions transactions courtes temps réel recherche d'enregistrements détaillés	requêtes complexes, ad-hoc sélections peu de transactions transactions longues batch agrégations et group by

20



Architecture à 3 niveaux

- Serveur de la BD de l'entrepôt
 - Presque toujours relationnel
- Data marts /serveur OLAP
 - Relationnel (ROLAP)
 - Multidimensionnel (MOLAP)
- Clients
 - Outils d'interrogation et de rapports
 - Outils d'analyse et d'aide à la décision

22

Construction d'un entrepôt de données

Trois phases principales

1. Acquisition:

Extraction : collection de données utiles

Préparation : transformation des caractéristiques des données du système opérationnel dans le modèle de l'entrepôt

Chargement : nettoyage (élimination des doublés, incomplètes, règles d'intégrité, etc.) et chargement dans l'entrepôt (trier, résumer, calculs, index).

2. Stockage :

Les données sont chargées dans une base de données pouvant traiter des applications décisionnelles.

3. Restitution des données :

Il existe plusieurs outils de restitution (tableaux de bord, requêteurs SQL, analyse multidimensionnelle, data mining ...)

23

Maintenance

- Les données de l'entrepôt sont stockées sous forme de vues matérialisées sur les différentes sources de données.
- Quand répercuter les mises à jour des sources ?
 - À chaque modification ?
 - Périodiquement ?
 - À définir par l'administrateur
- Comment les répercuter ?
 - Tout recompiler périodiquement ?
 - Maintenir les vues de façon incrémentale
 - Détecter les modifications (transactions, règles actives, etc.)
 - Les envoyer à un intégrateur qui détermine les vues concernées, calcule les modifications et les répercute.

24

Evolution

Les entrepôts sont amenés à évoluer **souvent** et **considérablement**.
La taille d'un entrepôt **croît rapidement** (de 20giga à 100giga en 2 ans).

Pourquoi ?

- nouvelles données (extension géographique, changement de fréquence des historiques, changement du niveau de détail, etc.)
- ajout de nouveaux éléments de données au modèle (l'ajout d'un attribut pour 2millions de n-uplets représente une augmentation considérable!)
- création de nouveaux index, résumés
- ajout de nouveaux outils (générateurs de requêtes, outils OLAP, etc.)
- nouveaux utilisateurs
- complexité des requêtes

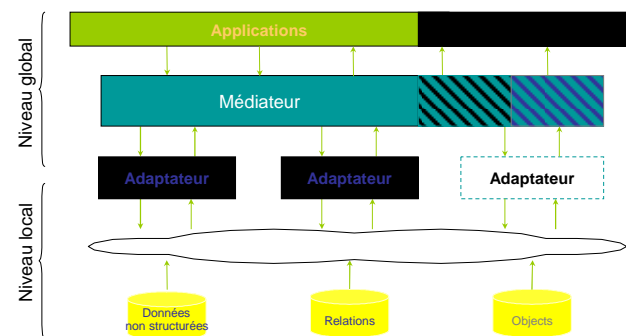
Comment garantir l'**extensibilité**, la **disponibilité**, la **maintenabilité** ?

25

Médiateurs

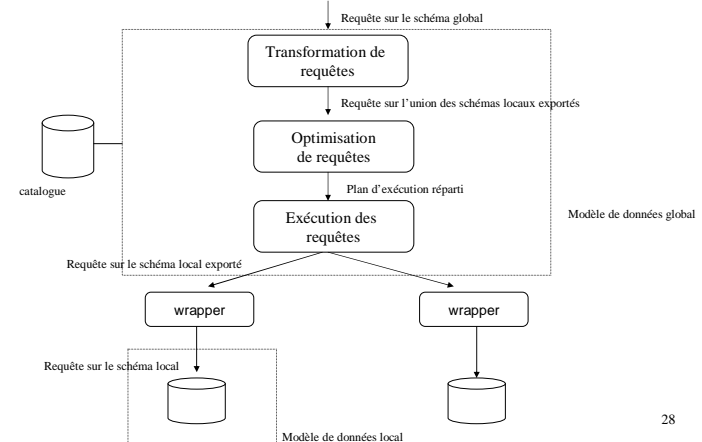
26

Médiateurs



27

Médiateur



28

Médiateur

Le médiateur s'occupe de la répartition des sources :

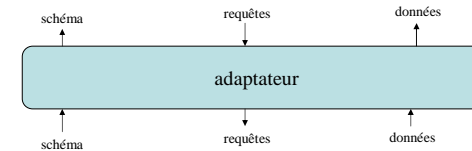
- Localisation des sources
- Accepte les requêtes des clients
- Réécrit (décompose) et optimise les requêtes (optimisation répartie)
- Envoie les plans d'exécution à faire exécuter par les wrappers des différentes sources
- Combine (recompose) les résultats des wrappers et effectue éventuellement quelques opérations supplémentaires

Attention : le médiateur ne comprend pas de code spécifique aux sources !

29

Adaptateur (Wrapper)

- Cache l'hétérogénéité au médiateur



- Traduit le schéma des sources en termes du schéma global
- Traduit les requêtes du médiateur en termes compréhensibles par les sources
- Traduit les résultats renvoyés par la source en termes du schéma global
- Un adaptateur par source (peut constituer un obstacle à l'intégration d'un nombre important de sources)
- Assez difficiles à écrire
- Peuvent être « intelligents » : effectuer des optimisations spécifiques aux sources
- Sont généralement associés aux sources, mais peuvent aussi se trouver dans le médiateur

30

Exemple

Transformer :

```
<b>Data on the Web</b>
<i>Abiteboul S.</i>
<i>Buneman P.</i>
<i>Suciu D.</i>
Morgan Kaufman, 1999
```

En :

```
<livre>
<titre>Data on the Web</titre>
<auteur>Abiteboul S.</auteur>
<auteur>Buneman P.</auteur>
<auteur>Suciu D.</auteur>
<editeur>Morgan Kaufman </editeur>
<annee> 1999</annee>
</livre>
```

31

Communication médiateur/adaptateur

- Pour faciliter le travail d'intégration, on définit
 - Un langage commun dans lequel le médiateur interrogera les adaptateurs
 - Un format de résultat commun dans lequel les adaptateurs répondront au médiateur
- Le langage et le format du résultat peuvent être standardisés ou propriétaires

32

Schéma global

- Choix d'un modèle commun (et du langage de requêtes)
 - Relationnel (Information Manifold, Le Select, XPeranto),
 - orienté-objet (Garlic, Disco),
 - semi-structuré (Tsimmis, Yat, Nimble, Xylème),
 - Logiques de description (SIMS, Observer)
- Plusieurs approches pour définir le schéma global :
 - Global as View (Tsimmis, Disco, Yat, Garlic, XPeranto)
 - Local as View (AGORA, Information Manifold)
 - Combinaison des deux ?
- Ces approches sont déterminantes pour la réécriture des requêtes, et pour l'évolution du système d'intégration (ajout de sources)

33

Bilan

- Entrepôts:
 - efficacité
 - qualité des données
 - adapté aux applications d'aide à la décision
 - maintenance
 - intégration de nouvelles sources
- Médiateurs
 - fraîcheur des données
 - Intégration de nouvelles sources (dépend de la conception)
 - Performances
 - Réécriture des requêtes

Pb: passage à l'échelle

34

Architectures Pair-à-pair

35

Définition

- Chaque nœud (peer) participant peut être client ou serveur
- Chaque nœud « paye » sa participation en donnant accès à une partie de ses ressources
- Propriétés :
 - Pas de coordination centralisée
 - Pas de BD centralisée
 - Aucun nœud n'a une vision globale du système
 - Le comportement global émerge à partir des interactions locales
 - Tous les services et données sont accessibles de n'importe quel nœud
 - Les nœuds sont autonomes
 - Les nœuds et les connections ne sont pas fiables

36

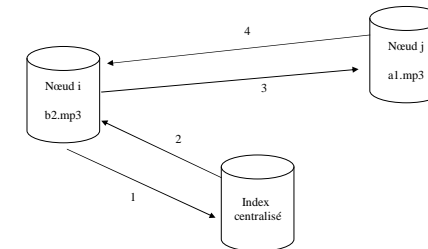
Différents types de systèmes P2P

- Systèmes de partage de fichiers
 - Napster, Gnutella, Freenet
- Commerce électronique
 - eBay, serveurs d'intégration B2B
- Bases de données réparties
 - Mariposa [Stonebraker96]
- Réseaux
 - Réseaux mobiles ad-hoc

L'approche P2P se trouve à tous niveaux du système : utilisateur, application, gestion de l'information, réseau.

37

Napster



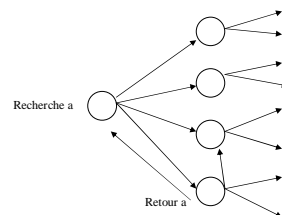
1: enregistrement de i
recherche de a1.mp3
ajout des fichiers de i dans l'index
2: retour des noeuds possédant a1.mp3

3: demande directe de i à j pour télécharger a1.mp3
4: téléchargement de a1.mp3 et
ajout dans la base de ressources partagées

38

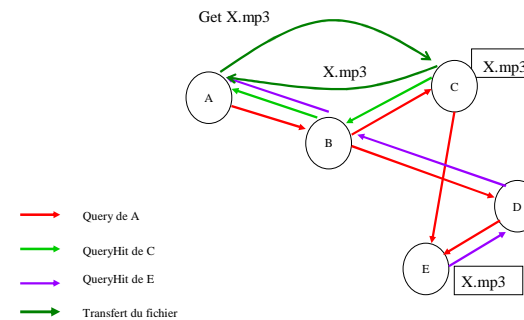
Gnutella

- Chaque nœud propage la requête à ses voisins (en général 4)
- Le nombre de propagations est limité (en général à 7)
- Détection de cycles grâce à l'identificateur des paquets



39

Gnutella (recherche)



40

Conclusion

- De plus en plus de répartition (de données, de ressources)
- Plusieurs architectures :
 - BD réparties
 - Entrepôts de données
 - Médiateurs
 - Systèmes pair-à-pair
- Choix en fonction des applications

41