Répartition, Réplication, Nomadisme, Hétérogénéité dans les SGBDs

Didier DONSEZ
Université Joseph Fourier
IMA – IMAG/LSR/ADELE
Didier.Donsez@imag.fr
Didier.Donsez@ieee.org

Les Développements Technologiques

- Amélioration des communications
  - Réseaux LAN et WAN plus rapides, plus surs

    FDDI, Fiber Channel, GigaEthernet

    RNIS, ATM

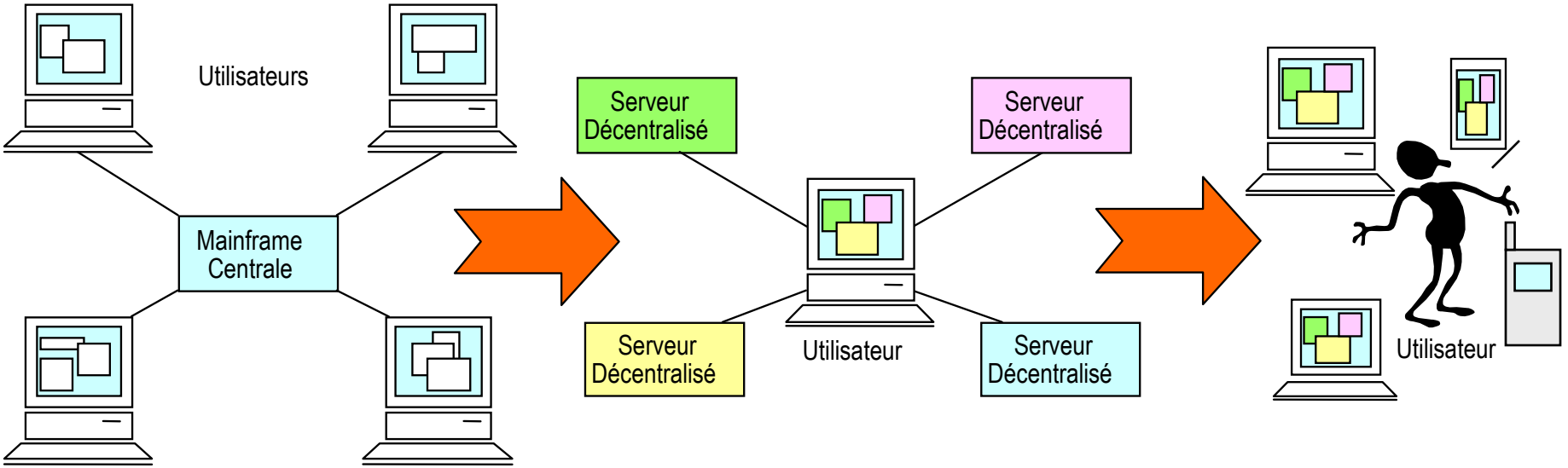
- Amélioration des postes de travail
  - meilleur prix/performance
  - amélioration des possibilités

    stations parallèles

Les Pressions pour la Distribution

- Pression des entreprises
  - 90% des accès concernent des données locales
  - impératif de décentraliser l'information

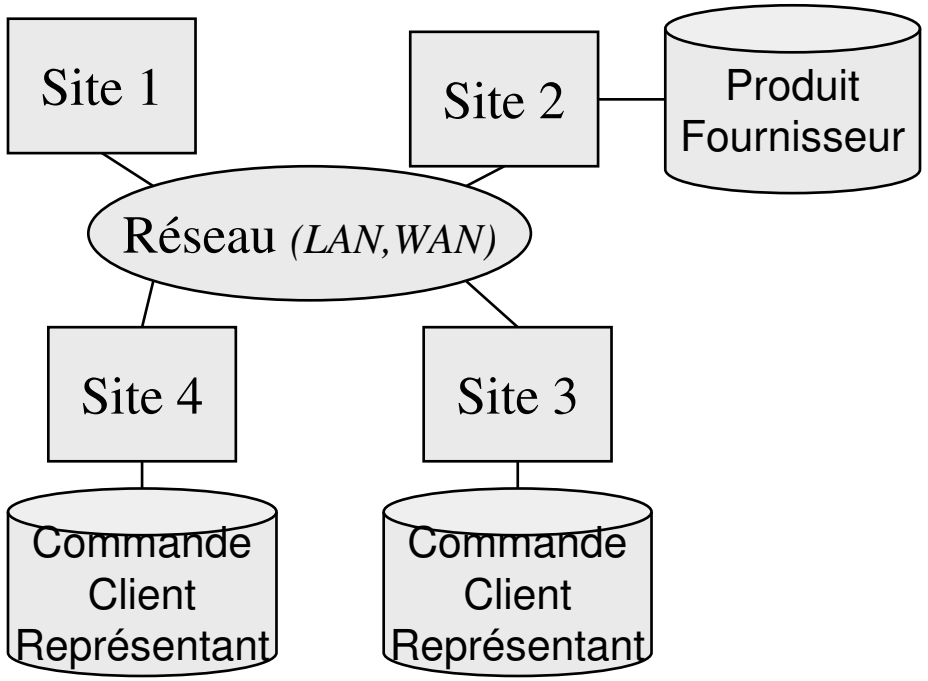
    cas des Multinationales
- Pression des utilisateurs
  - bases de données directement adaptées à leurs besoins
  - Vers l'Ubiquitous Computing (Informatique OmniPrésente)



Qu'est qu'un SGBD Répartie

- BD Répartie (*Distributed Database*)
  - un schéma global
  - une collection de BDs logiquement reliées et réparties entre plusieurs sites

Produit(NP,Designation, PrixUnit, NF)
Fournisseur(NF,Nom,Ville)
Client(NCL,Nom,Ville)
Représentant(NR,Nom,Ville)
Commande(NP,NCL,Date,Qte,NR)



Evaluation de l ’approche BD-R

- Avantages
  - Extensibilité
  - Partage de données hétérogènes et réparties
  - Meilleures performances
  - Meilleure disponibilité
  - Economie
- Inconvénients
  - Complexité
  - Manque d ’expérience
  - Distribution du Contrôle
  - Difficulté de Migration

Avantages comparés

BD Centralisée

☺ Organisation Simple

- contrôle et planification par une seule équipe

☺ Expérience

- Techniques BD simples
- Sécurité, Performances

☺ Applications

- Temps d ’accès uniforme à toute la BD

BD Répartie

☺ Organisation Déjà Répartie

- besoins locaux différents
- autonomie locale

  gestion et contrôle des données

☺ Expérience

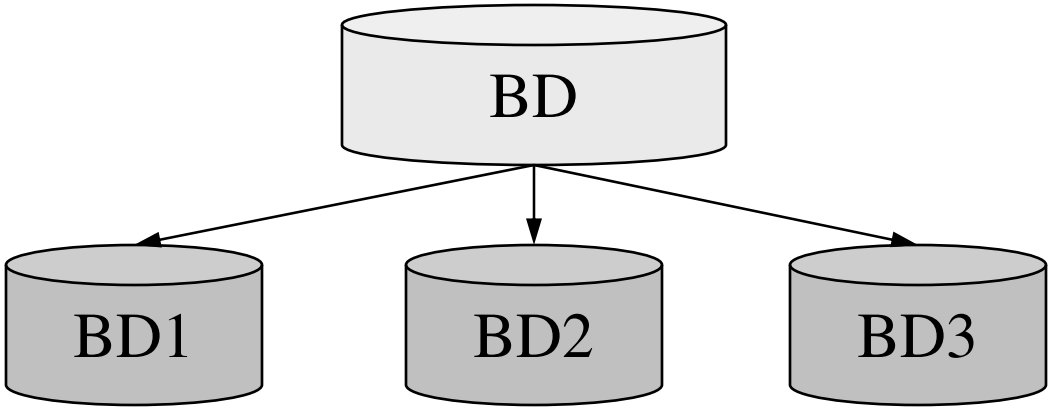
- Traitements principalement locaux
- Accès multi-site occasionnel

☺ Applications

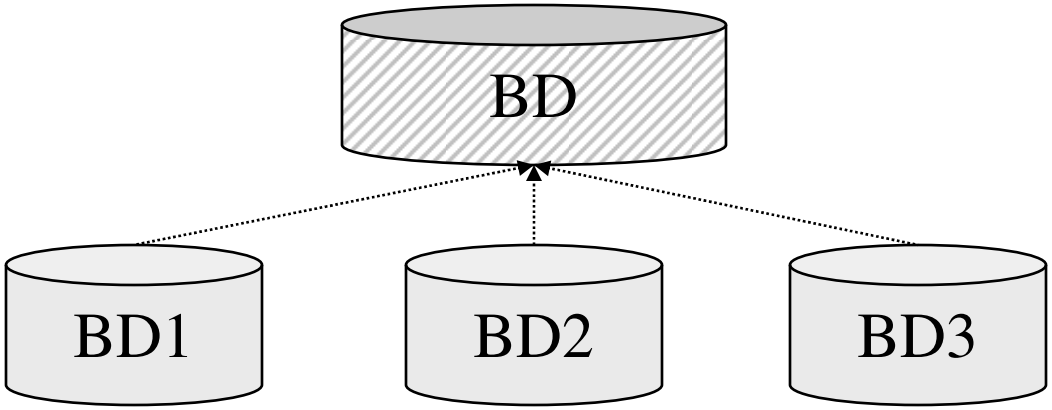
- Très diverse (besoins spécifiques)

Migration vers une BDR

- Décomposition en BD locales

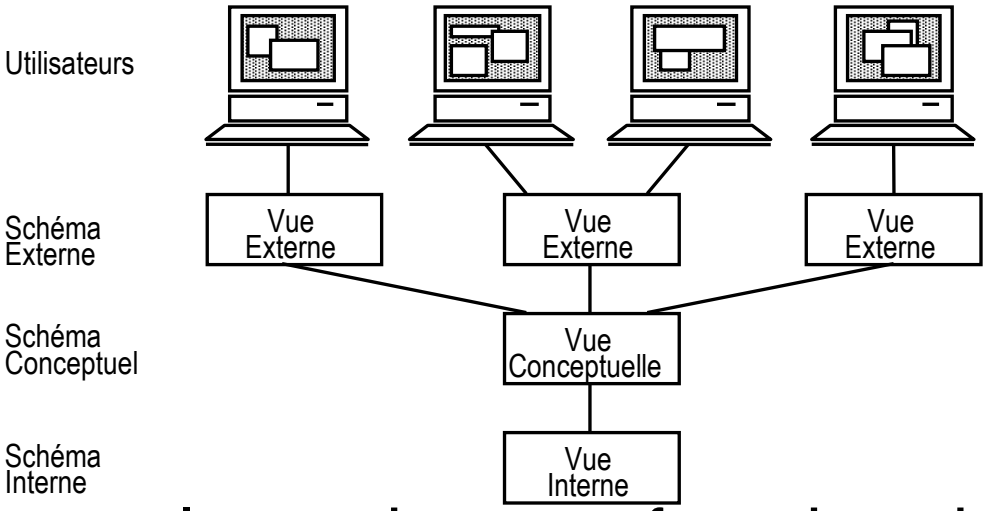


- Intégration logique des BDs locales existantes



Gestion des BD-R

- Rappel sur l'architecture ANSI/SPARC



- Plusieurs alternatives en fonction de l 'existence et de la réalisation du schéma global
  - Couplage fort
  - Couplage Faible
  - Fédéré

Couplage Fort

- Schéma Global
  - ☺ Indépendance Applications/Bases de Données
  - ☹ Schéma global lourd à gérer

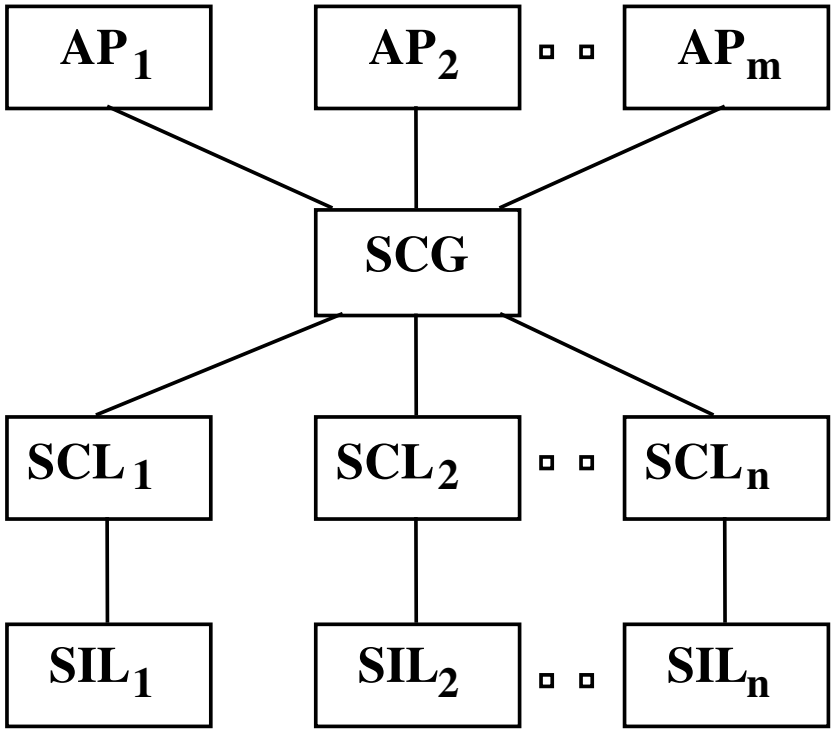


Schéma Global

- Schéma Conceptuel Global
  - Description unifiée et globale de toutes les données du SGBD-R
  - Indépendance à la répartition

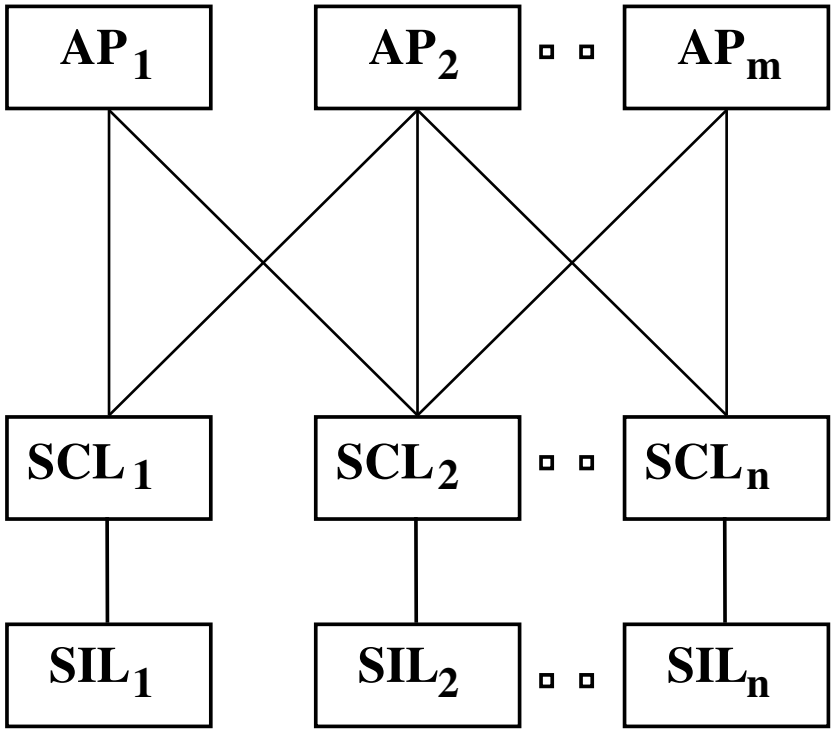
Produit(NP,Designation, PrixUnit, ...)
Client(NCL,Nom,Ville)
Commande(NP,NCL,Date,Qte,...)

- \+ Schéma de placement
  - Régle de correspondance avec les données locales
  - Indépendance à la localisation et à la décomposition

Produit = Produit@Site2
Commande = Commande@Site3 ∪ Commande@Site4
Client = Client@Site3 ∪ Client@Site4

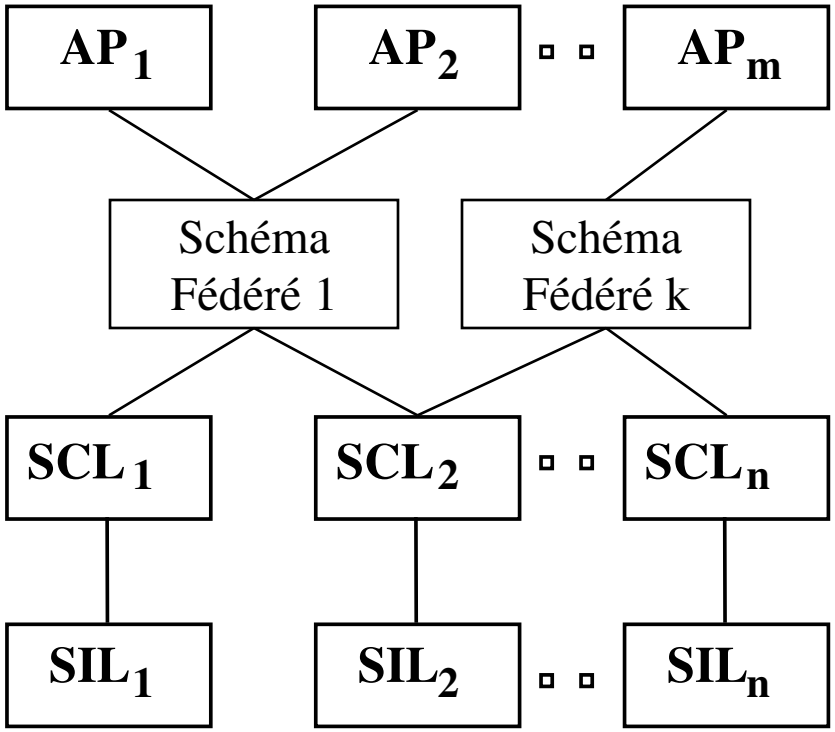
Couplage Faible

- N Schémas Locaux
  - ☺ Pas de Schéma global
  - ☹ Pas d ’Indépendance Applications/Bases de Données



Fédéré

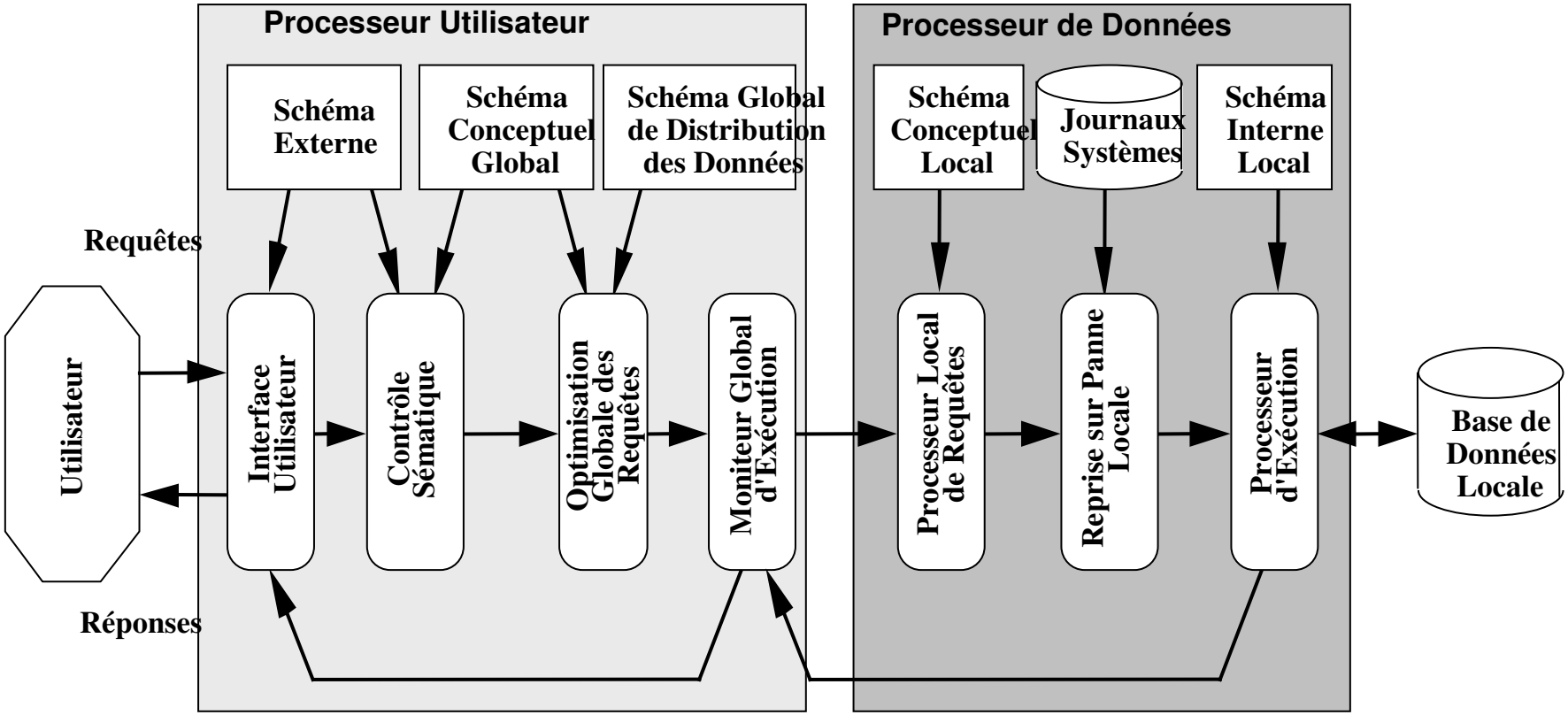
- Fédération de Schémas Hétérogènes
  - ☺ Moyen contrôlé de migration
    depuis les SGBDs locaux vers un SGBD-R



Notions Complémentaires

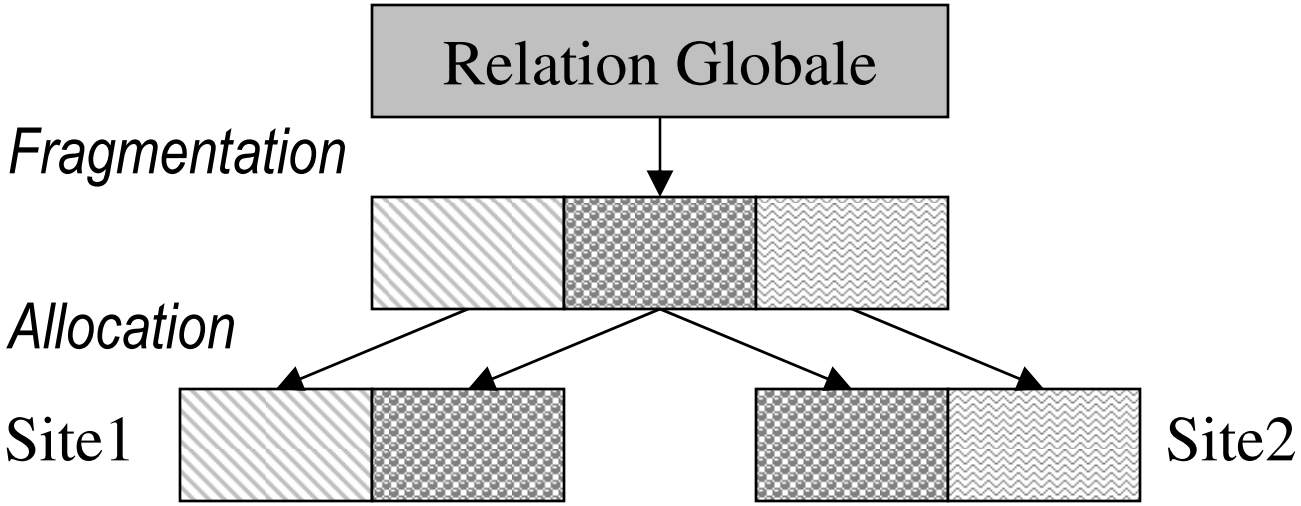
- SGBD Réparti (*Distributed DBMS*)
  - gère une BD-R et fournit les mécanismes d'accès rendant la distribution transparente à l'utilisateur.
- BD Interopérable (*Interoperable Database*)
  - BD capable d 'échanger des données en comprenant mutuellement ce qu 'elles représentent
- Multibase (*Multibase*)
  - Plusieurs BDs hétérogènes capables d 'interopérer avec une application via un langage commun et sans modèle commun
- BD fédérée (*Federated Database*)
  - Plusieurs BDs hétérogènes accédées comme une seule via une vue commune

L'architecture physique d'un SGBD-R



- Décomposition de la BD-R
  - Fragmentation
  - Duplication
- Requêtes Réparties
  - Optimisation et évaluation de requêtes reparties
- Transactionnement Distribué
  - Fiabilité, Résistance aux Pannes
    - Terminaison distribuée
  - Contrôle de concurrence
    - Résolution des interblocages

Objectifs de la Décomposition



- Fragmentation
  - Trois 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="Paris"
  - Client2 = select * from Client where ville<>"Paris"

- Reconstruction
  - Client = Client1 ∪ Client2

Client	NCL	Nom	Ville
	C1	Dupont	Paris
	C2	Durant	Lille
	C3	Martin	Nice
	C4	Olivier	Paris

Client1	NCL	Nom	Ville
	C1	Dupont	Paris
	C4	Olivier	Paris

Client2	NCL	Nom	Ville
	C2	Durant	Lille
	C3	Martin	Nice

Fragmentation Horizontale Dérivée

- Fragments définis par jointure
  - Commande1 = select * from Commande
    where NCL in (select NCL from Client1)
  - Commande2 = select * from Commande
    where NCL in (select NCL from Client2)

- Reconstruction
  - Commande = Commande1 ∪ Commande2

Cmd	NP	NCL	Date	Qte
	P1	C1	1/1/99	10
	P2	C2	1/1/99	20
	P2	C2	2/1/99	30
	P3	C4	4/1/99	30

Cmd1	NP	NCL	Date	Qte
	P1	C1	1/1/99	10
	P3	C4	4/1/99	30

Cmd2	NP	NCL	Date	Qte
	P2	C2	1/1/99	20
	P2	C2	2/1/99	30

Fragmentation Verticale

- Fragments définis par projection
  - Produit1 = select NP, Désigantion, PrixUnit from Produit
  - Produit2 = select NP, NF from Produit
- Reconstruction
  - Produit = select NP, Désigantion, PrixUnit, NF
    from Produit1 join Produit2 using(NP)

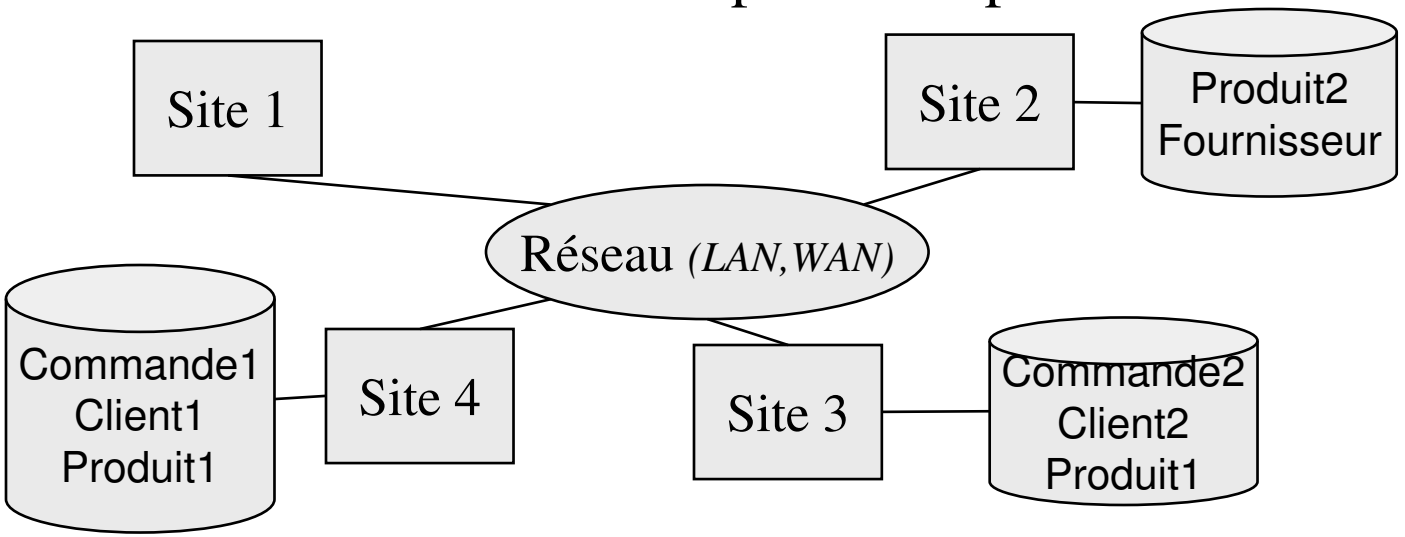
Produit	NP	Desig.	PrixUnit	NF
	P1	Ciment	100	F1
	P2	Bois	200	F1
	P3	Vis	50	F2
	P4	Clou	10	F2

Produit1	NP	Desig.	PrixUnit
	P1	Ciment	100
	P2	Bois	200
	P3	Vis	50
	P4	Clou	10

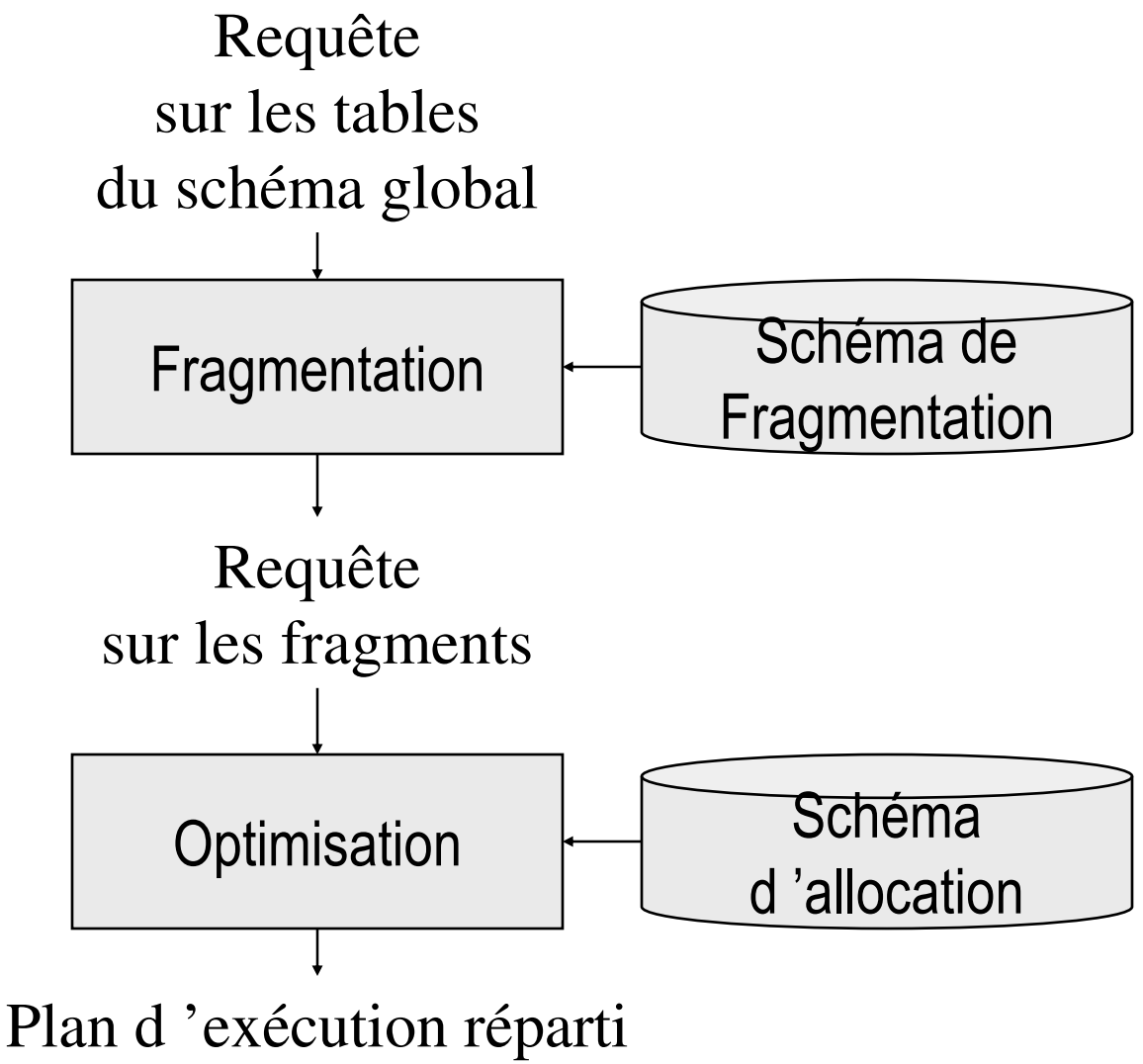
Produit2	NP	NF
	P1	F1
	P2	F1
	P3	F2
	P4	F2

Allocation des 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



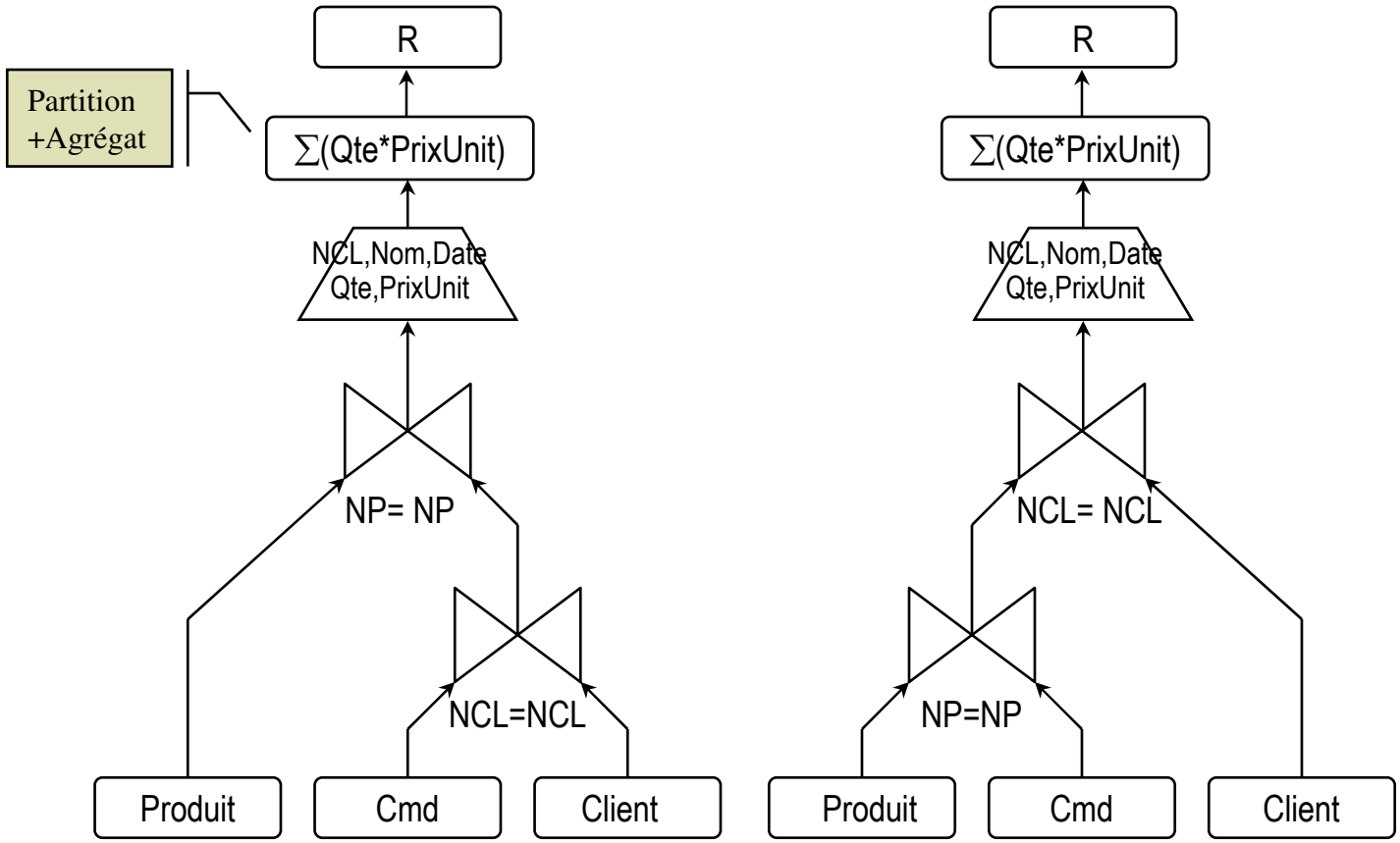
Evaluation des Requêtes Réparties



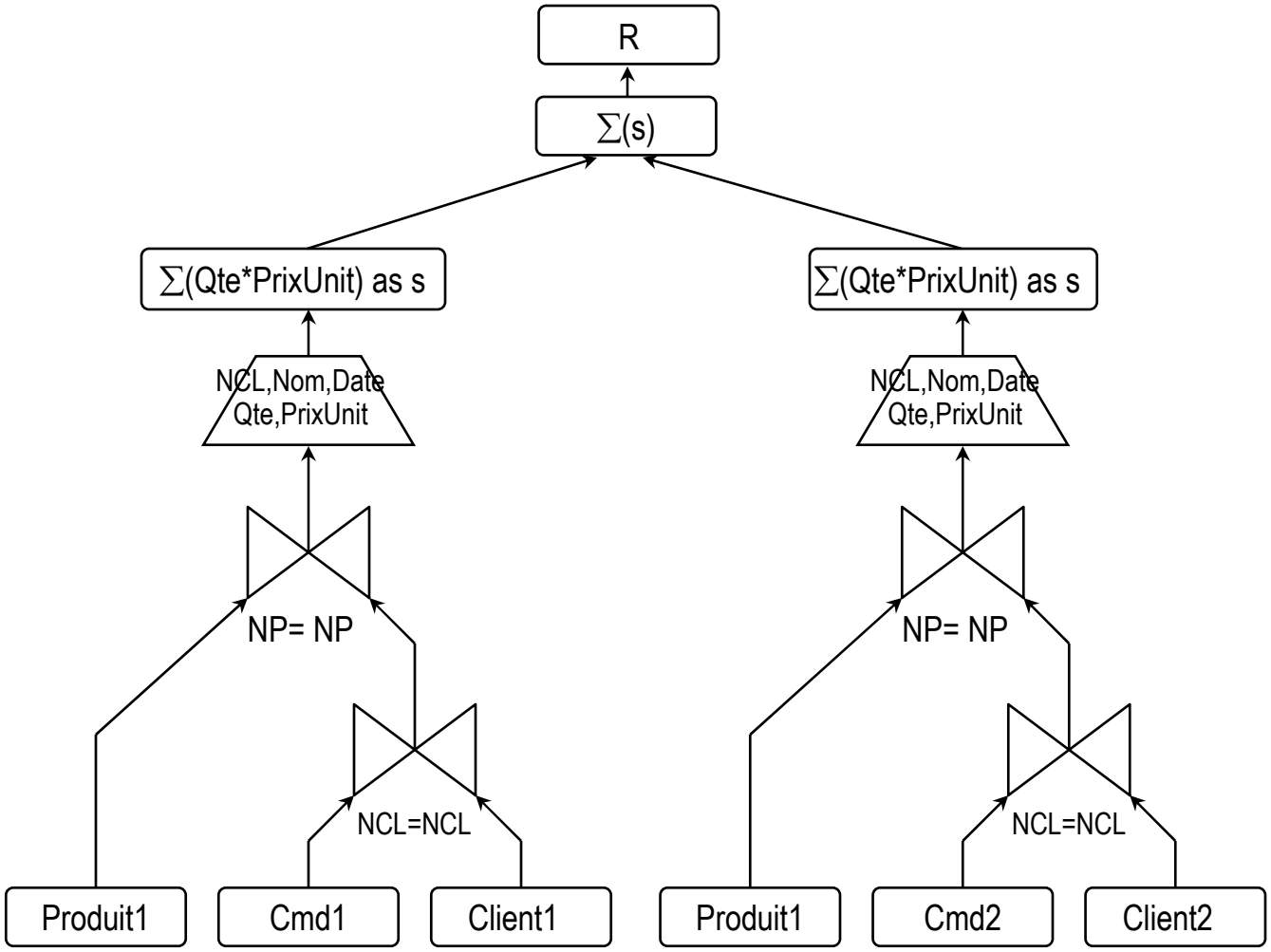
Exemple I
2 arbres

Produit(NP,Designation, PrixUnit, ...)
Client(NCL,Nom,Ville)
Commande(NP,NCL,Date,Qte,...)

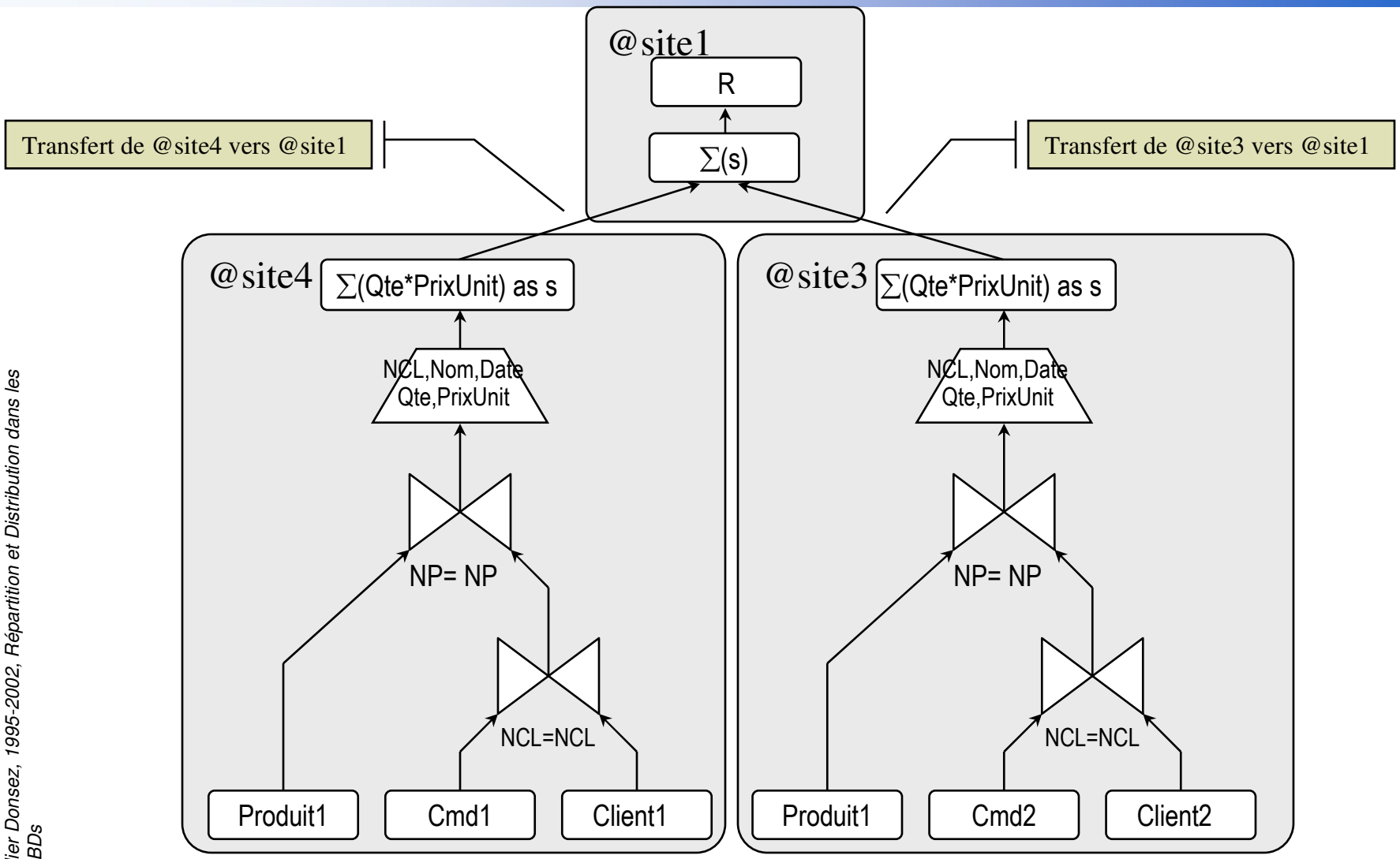
```
select NCL,Nom,Date, Sum(Qte*PrixUnit)
from (Cmd join Client using (NCL)) join Produit using (NP)
group by NCL,Nom,Date;
```



Exemple I :
Fragmentation de l ’arbre 1



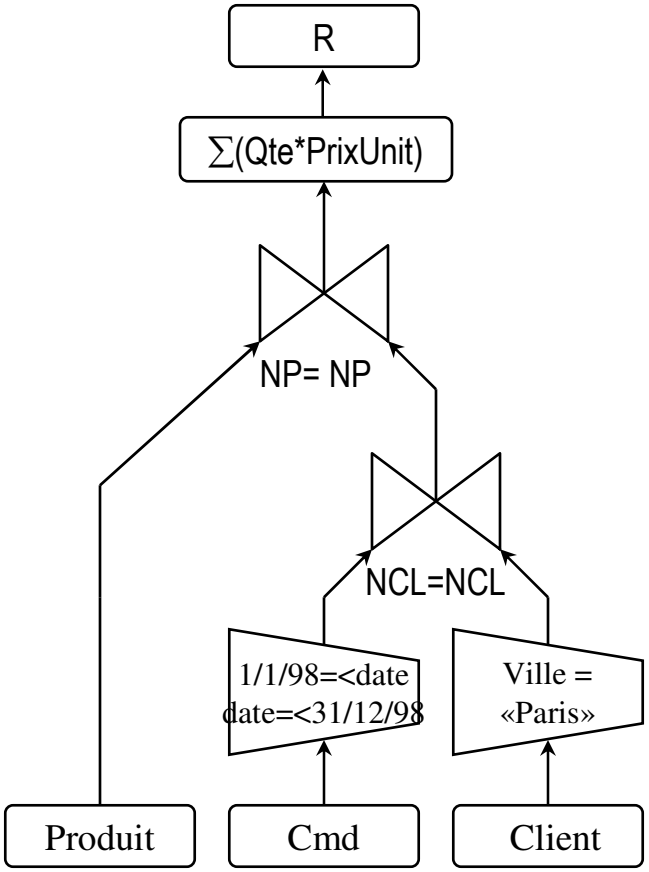
Exemple I :
Allocation des fragments de l 'arbre 1

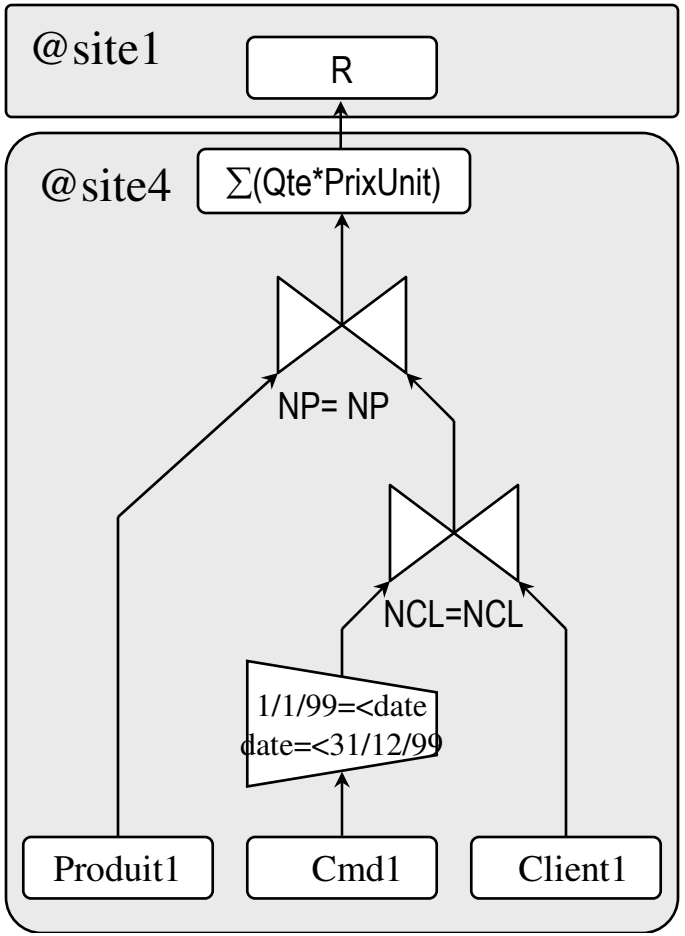


Exemple II

Produit(NP,Designation, PrixUnit, ...)
Client(NCL,Nom,Ville)
Commande(NP,NCL,Date,Qte,...)

select Sum(Qte*PrixUnit) as CA99Paris
from (Cmd join Client using (NCL)) join Produit using (NP)
where Date between 1/1/99 and 31/12/99 and Ville="Paris"





Optimisation

- Plan d ’exécution
  - ensemble des Sous-Requêtes sur les schémas locaux
    et des opérations de transferts des résultats intermédiaires

- Recherche d ’un plan d ’exécution de coût minimal
  - Coût = f (temps de réponse, $)
    est une fonction sur l ’espace des plans d ’exécution

Coût = α*CoûtCPU + β*CoûtES + δ*CoûtComm

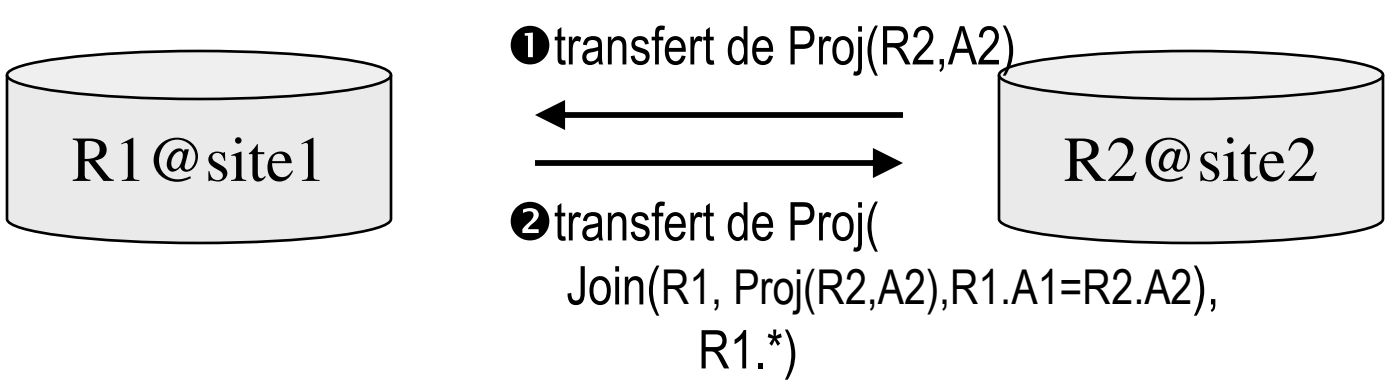
En WAN, le coût de communication est majoritaire

δ*CoûtComm >> α*CoûtCPU + β*CoûtES

Optimisation : Jointure Répartie

- Jointure(R1,R2,R1.A1=R2.A2) @ site2
  - Direct

  - Par semi-jointure



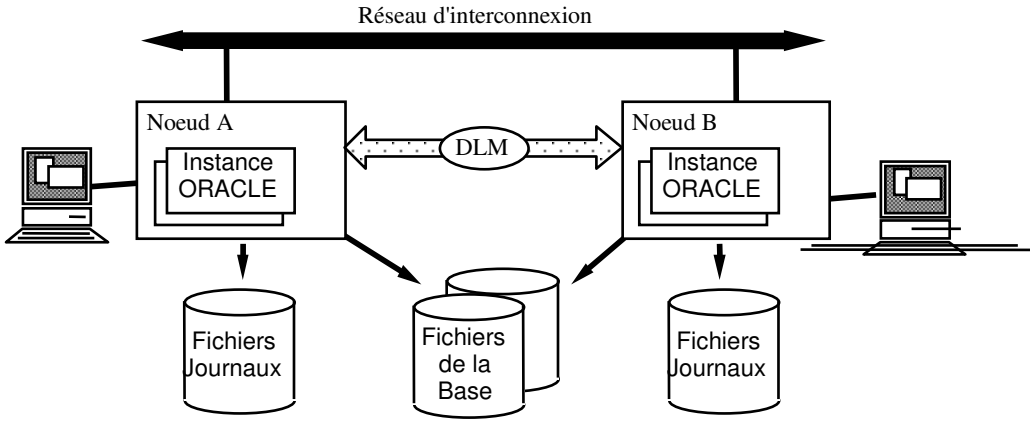
  - Exercice : Calculez le gain

Gestion des Transactions Réparties

- Contrôle de Concurrence Répartie
  - Verrouillage
  - Estampillage
  - Certification
- Reprise sur panne Répartie
  - Validation à 2 phases
  - Validation à 3 phases
  - Moniteur Transactionnel

Verrouillage distribué (i)

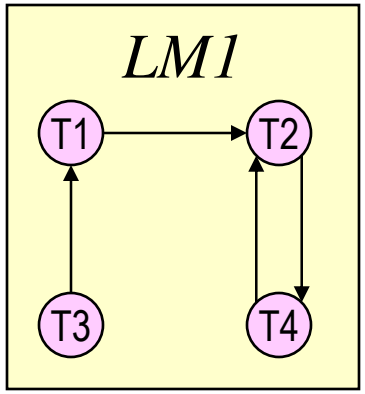
- Gestion des Verrous (LM Lock Manager) Centralisé
  - Distributed Lock Manager d 'Oracle

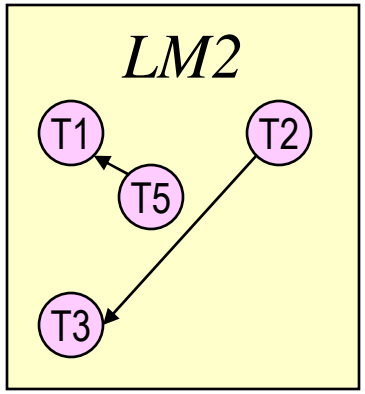


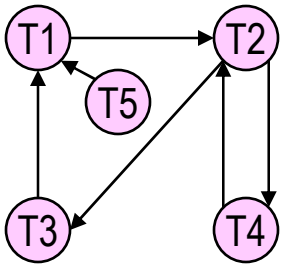
- Gestion des Verrous (LM Lock Manager) Répartie
  - Chaque site a un LM local
  - Le LM local maintient un graphe d 'attente local

Verrouillage distribué (ii)
LM Répartie

- Principe
  - Chaque sous-requête sur un site verrouille sur le GT local
  - Le GT local maintient un graphe d 'attente local
- L 'interblocage (*ou Verrous Mortels*)
  - plusieurs sites utilisent le 2PC
  - chaque site est capable de détecter un interblocage local (*T2-T4*)
  - mais un interblocage distribué est difficile (*T1-T2-T3*)







Détection des interblocages

- Prévention
  - Garantir que le problème ne survient jamais
  - Combinaison de Verrouillage et d ’Estampillage

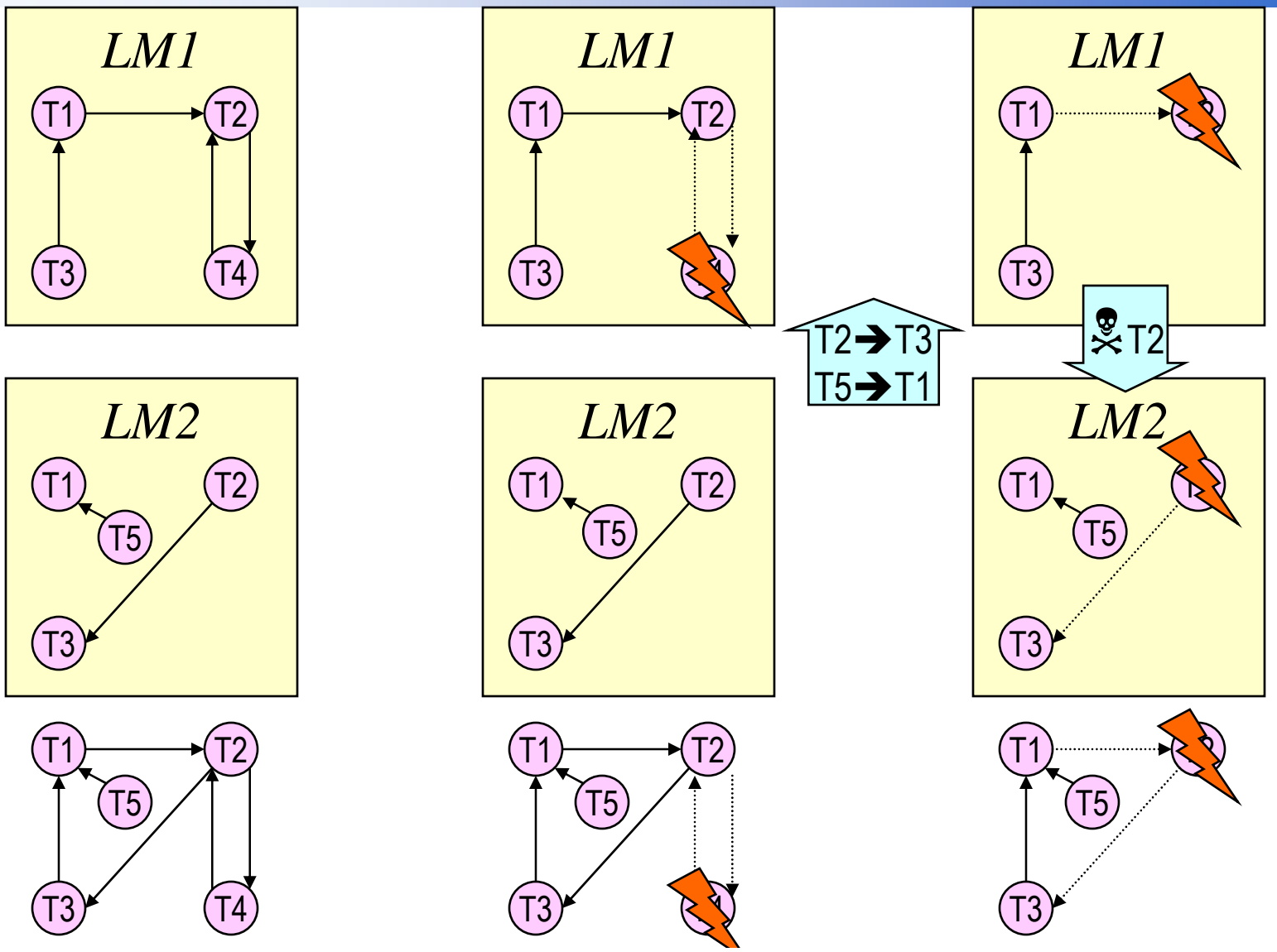
    DieWait, WoundWait
- Détection
  - Graphe d ’attente local complété périodiquement par les graphes locaux des autres sites
  - Détection de cycle sur l ’union des graphes locaux
- Présomption
  - Annulation des transactions

    trop longtemps (timeout) en attente

Plus d ’info:

- *Edgar Knapp, « Deadlock detection in distributed databases », ACM Computing Surveys, Vol. 19, No. 4 (Dec. 1987), Pages 303-328*

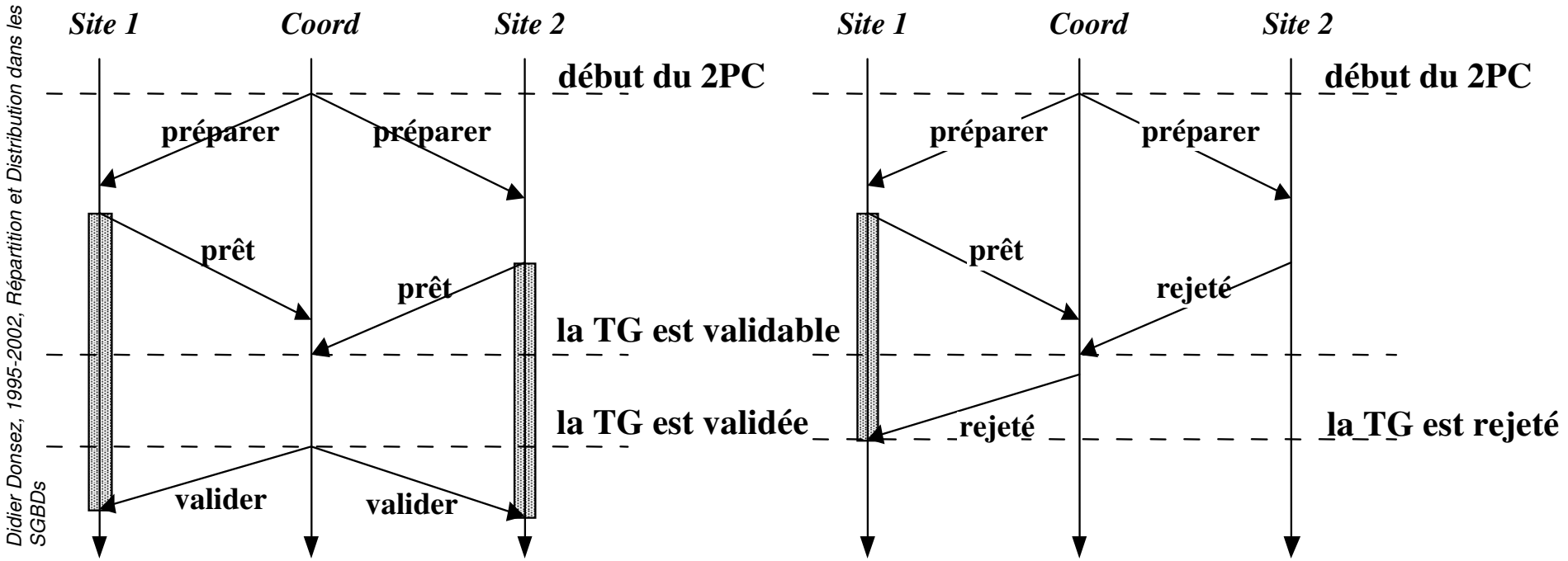
Détection des interblocages répartis



Terminaison d'une transaction distribuée

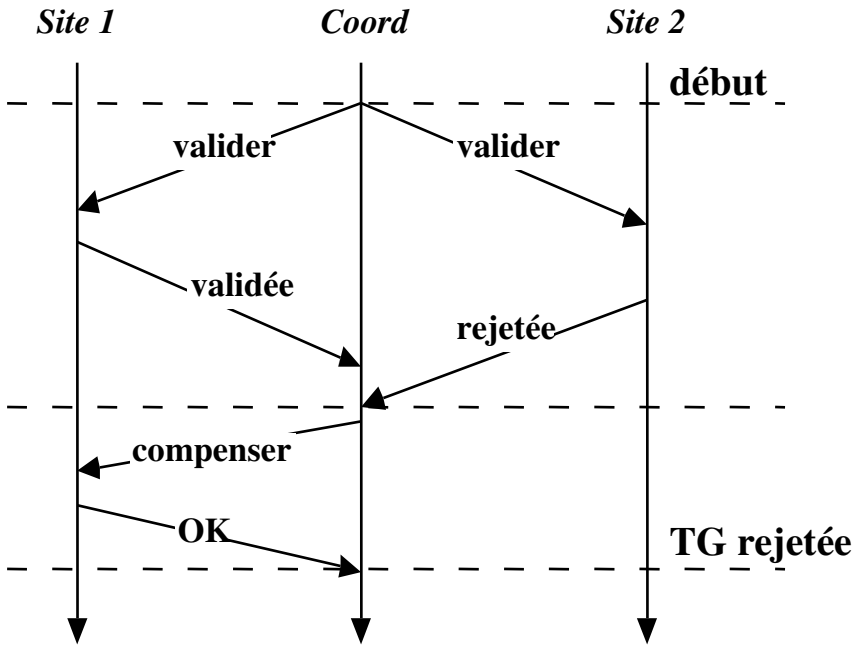
⇨Garantir l'atomicité d'une transaction distribuée

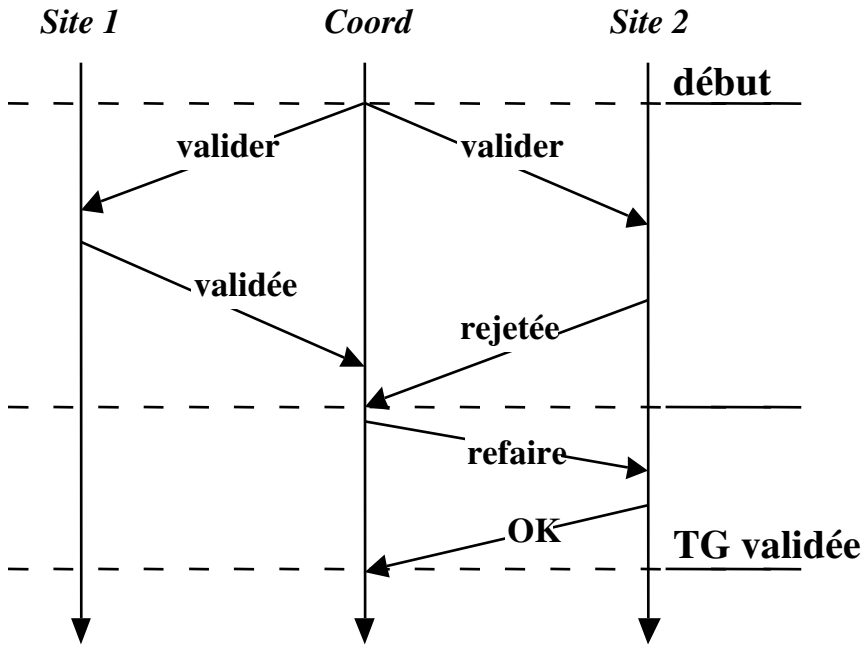
- Protocole de validation à 1 phase
  - Quand un seul des sites a des modifications à valider
- Protocole de validation à 2 phases (*Two Phase Commit*)



Protocoles Asynchrones de Validation

- le protocole Compenser
  - difficile de définir une transaction de compensation
  - compensation sur plusieurs sites
  - certains effets sont incompensables
- le protocole Refaire :
  - sous transactions re-essayables
  - exécution dépendante des valeurs





Objectifs de la réplication

- Performance
  - ☺ localité des accès en consultation (lecture)
  - ☹ mise en cohérence des mises à jour (écriture)
- Disponibilité (*Availability*)

1 heure d 'arrêt /jour	95,8%
1 heure d 'arrêt /semaine	99,41%
1 heure d 'arrêt /mois	99,86%
1 heure d 'arrêt /an	99,9886%
1 heure d 'arrêt /20 ans	99,99942%

Redondance multi sites

- Disponibilité(N serveurs) = 1 - ProbPanne^N

  1 serveur = 95 % de disponibilité

  2 serveurs = 99,75 % de disponibilité

Notions complémentaires

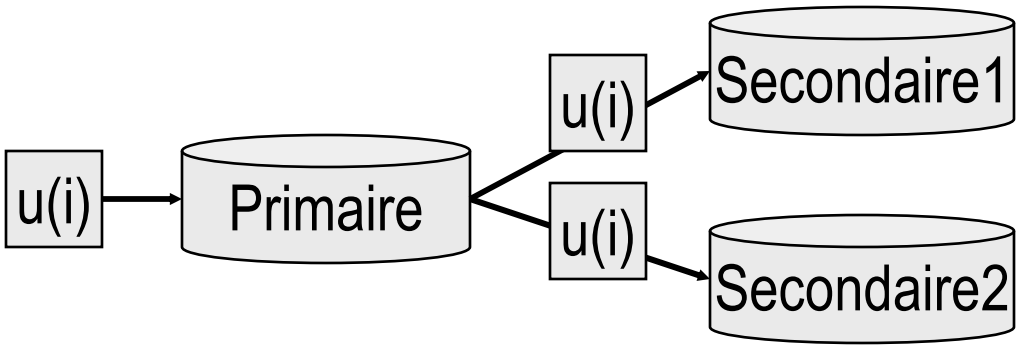
- Disponibilité
  - Disponible quand prét à fonctionner
    et apte à accomplir sa fonction de manière fiable
- Fiabilité
  - Aptitude à accomplir sa fonction sans défaillance dans des conditions données pour une durée déterminée
- Maintenabilité
  - Possibilité d'être maintenu ou rétabli en un temps donnée dans un état d'aptitude à accomplir sa fonction
- Maintenance
  - Ensemble des opérations qui permettent
    de maintenir (m. préventive) ou de rétablir (m. corrective)
- Sûreté de fonctionnement
  - Fiabilité et sécurité vis à vis des personnes et des biens

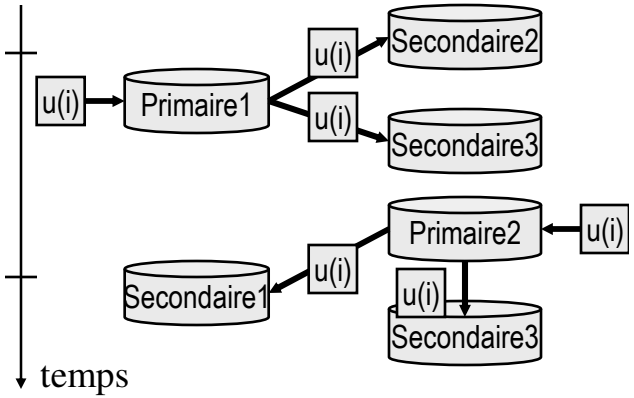
Modèle de réplication

- 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
- Mode de réplication
  - Asymétrique
    - une copie primaire / N copies secondaires
  - Symétrique
    - N copies primaires

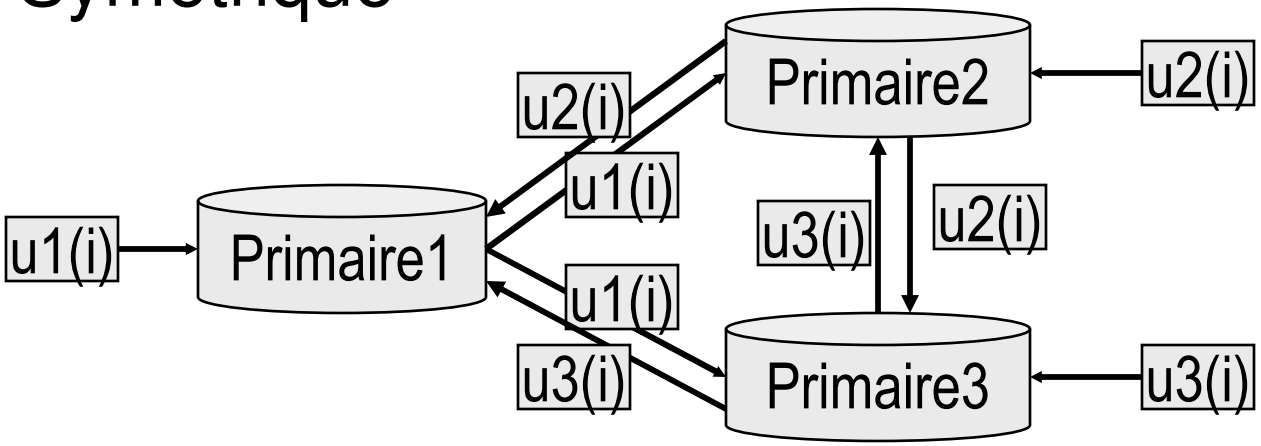
Mode de réplication

- Asymétrique





- Symétrique



Propagation des Mises à Jour

- *de la Source vers la Cible*

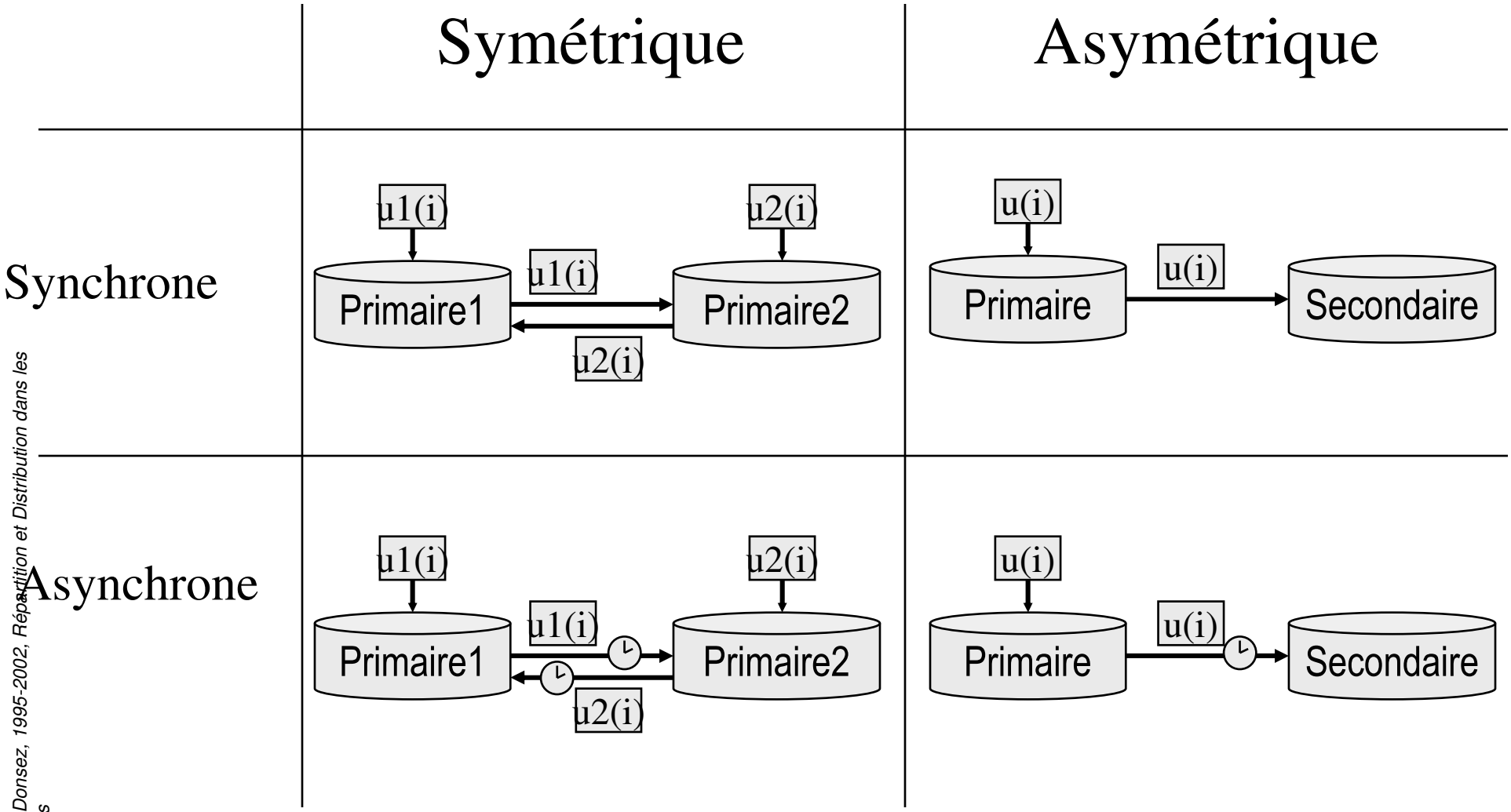
Synchrone

- Mises à jour globales dans une même transaction
  - ☺ Cohérence forte
    - Contrôle de Concurrence : Verrouillage (Maître/Esclave,Quorum), Estampillage, Certification
    - Terminaison : Two Phase Commit
  - ☹ Ralentit la transaction et le débit

Asynchrone 

- Mises à jour dans des transactions différées
  - ☺ Pas de retard
  - ☺ BD Nomades à la reconnexion
  - ☹ Fusion (manuelle) des copies divergentes

Modèle de Réplication et Propagation des Mises à jour

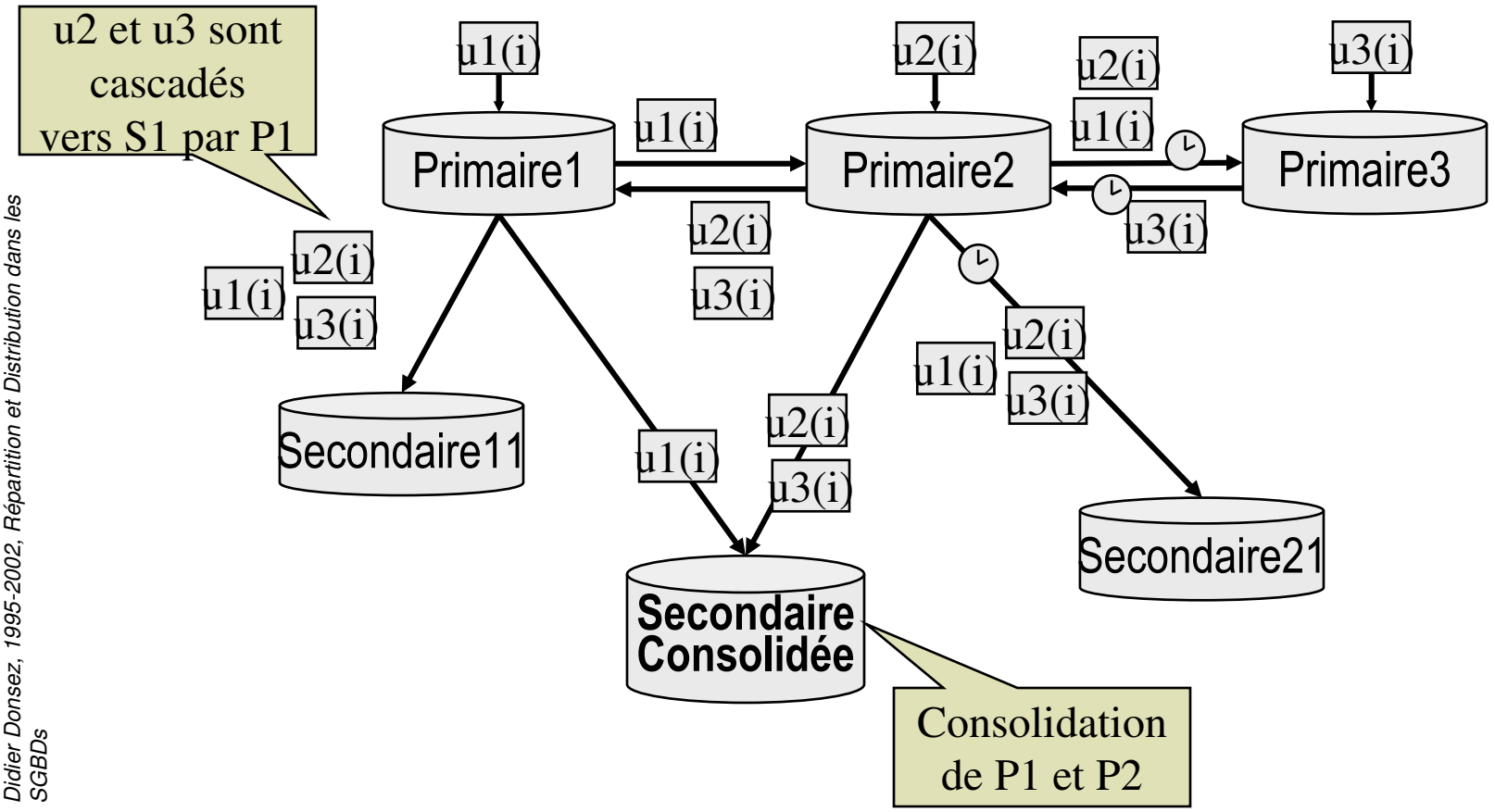


Configuration Hybride (i)

- Définition d 'un schéma de réplication

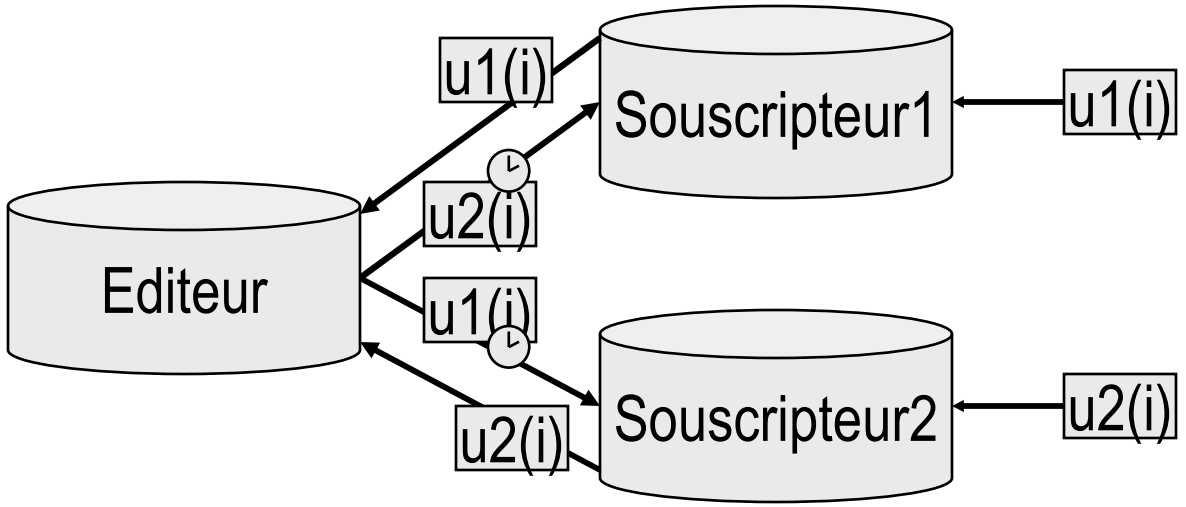
  par association symétrique/asymétrique synchrone/asynchrone

  - Adaptation à des problèmes spécifiques



Configuration Hybride (ii)

- 1 Editeur / N Souscripteurs (MS SQL Server v7)
  - mise à jour synchrone des souscripteurs vers l 'éditeur
    - Two Phase Commit
  - mise à jour asynchrone de l 'éditeur vers les sous-souscripteurs
    - Convergence des copies divergentes



Détection des Mises à Jour sur les Données Répliquées

- Utilisation des Journaux
  - les transactions qui modifient écrivent une marque spéciale dans le journal
  - ☺ Détection périodique en lisant le journal
  - ☹ Modification de la gestion du journal

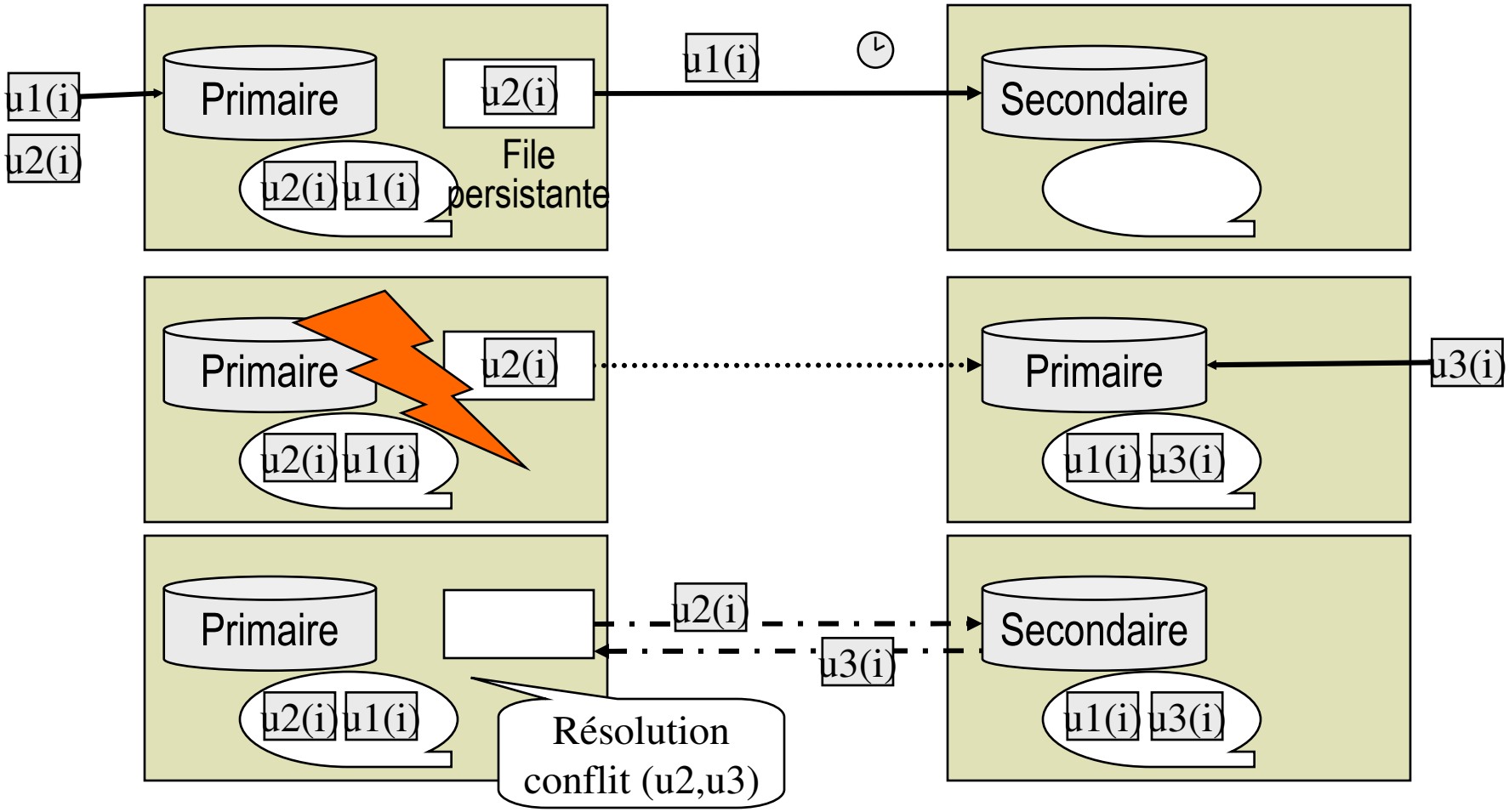
- Utilisateur de Triggers
  - La modification d ’une donnée répliquée déclenche un trigger
  - ☺ Mécanisme général et extensible
  - ☹ la détection fait partie de la transaction et la ralentit

Rafraîchissement des Copies Cibles

- Propagation de mises à jour
  - Total
  - Différentiel (propogation des dernières mises à jour)
- Fréquence des Rafraîchissements
  - Immédiat : après la transaction
  - A intervalle régulier : minute, heure, jour, semaine, mois
  - Evénementiel : provoqué par une application
- Initiative du rafraîchissement
  - Push (source vers cible)
  - Pull (cible vers source) - cas des BD Nomades à la reconnexion
- Outils
  - files d 'attente persistantes (Store & Forward)
  - passerelles pour les données hétérogènes

Exemple

- Hot Standby d ’Oracle



Résolution des Conflits (i)

- Cas
  - 2 copies primaires rafraîchies en asynchrone
  - copie primaire en panne en asynchrone (Hot Standby)
- Détection des conflits
  - Solution

    1- transporter la nouvelle (rien si DELETE)
    et l 'ancienne valeur (rien sur INSERT) sur la cible

    2- vérifier ancienne valeur source = valeur courante cible

    SINON conflit
- Type de Conflits
  - Unicité
    - violation d 'une contrainte UNIQUE
  - Mise à Jour
    - différence entre l 'ancienne valeur source et la valeur courante cible
  - Suppression
    - modification d 'une ligne détruite

Résolution des Conflits (ii)

- Fusion des copies divergentes
  - Actions possibles
    - Ignorer les ordres en conflits

      DISCARD de la nouvelle valeur, OVERWRITE de la valeur courante
    - Répliquer uniquement la dernière mise à jour

      EARLIEST TIMESTAMP, LASTEST TIMESTAMP
    - Traiter les conflits en fonction d 'une priorité prédéfinie

      SITE PRIORITY
    - Routines pré-définies
    - Traitement manuel

      effectué par le DBA à partir d 'une liste de conflits
      (table DEFERROR d 'Oracle sur le nœud cible)
- Remarque

  différence entre Cohérence séquentielle et Cohérence transactionnelle

Résolution des Conflits (iii)

- Routines pré-définies
  - faciliter la tache du DBA
  - Routines
    - MAXIMUM, MINIMUM, AVERAGE
      de la valeur courante et de la nouvelle valeur
    - ADDITIVE

      valeur cible += (nouvelle valeur source - ancienne valeur source)
- Inconvénients
  - convergence pour au maximum deux sites « maîtres »
  - impact sur le schéma et sur le codage des procédures
    - le développeur doit prévoir la gestion du conflit
  - ne règlent pas tous les types de conflit
    - suppression, modification de la clé primaire, valeurs nulles, violation de CI

Gestion de la Cohérence dans la Réplication Symétrique

- Gestion Centralisée
  - un site est élu maître, les autres sont esclaves
  - Inconvénients
    - En cas de panne du maître, il faut élire un nouveau maître
    - En cas de partition réseau : 2 maîtres donc 2 copies divergentes
- Gestion Décentralisée
  - Les Quorums

les Quorums

- But:
  - Résister à k pannes de sites parmi N sites
    ou à un partitionnement du réseau
  - Approche sans maître
    - les verrous sont obtenus après avis d 'un quorum de site
- Quorum

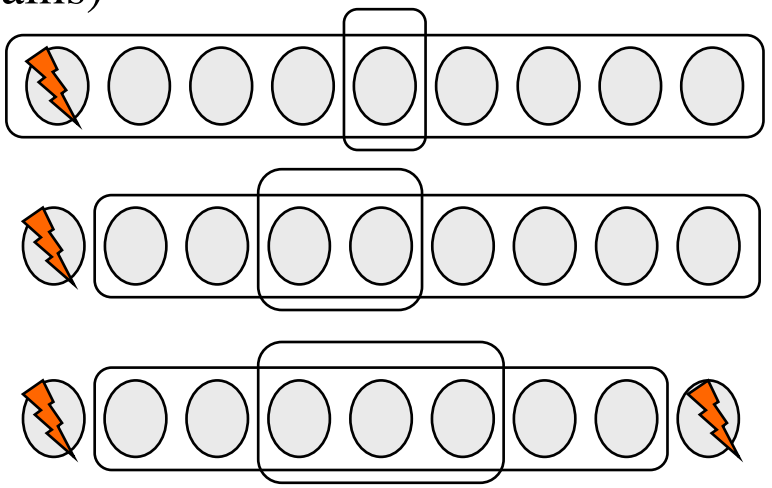
  R + W > N (exclusion des lecteurs/Ecrivains)

  2W > N (exclusion de 2 Ecrivains)

  ex: 9 sites R=1 et W=9

  ex: 9 sites R=2 et W=8

  ex: 9 sites R=3 et W=7



les Quorums Topologiques

- But : limiter la taille des quorums possibles
  - Topologie logique
  - Topologie liée au réseau
- Quorum en Grille (Grid)

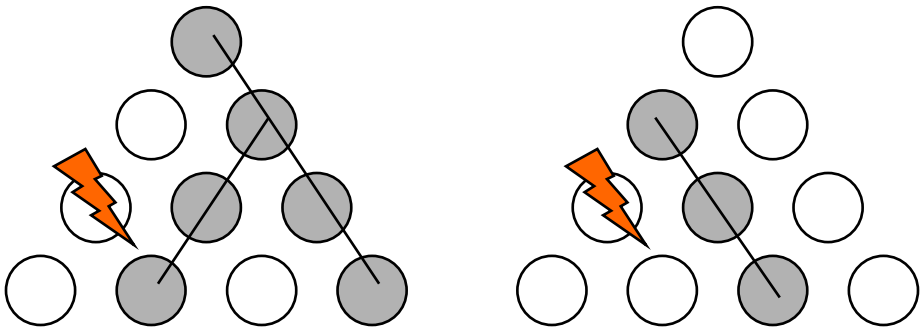
  Lecture : une ligne σ(N)

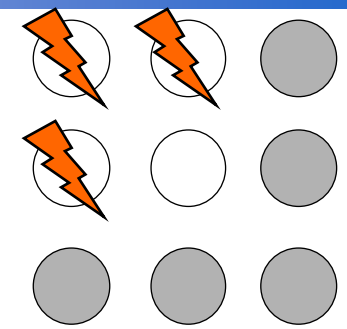
  Ecriture : une ligne + une colonne 2σ(N)-1

  ex : 9 sites R=3 W=5
- Quorum en Arbre (Tree)

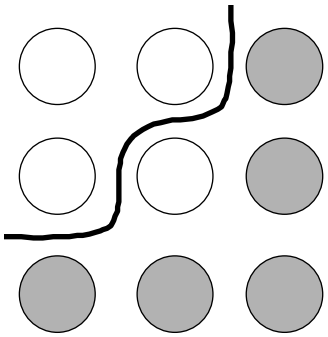
  Lecture : une branche droite

  Ecriture : une branche droite + une branche gauche





Panne



Partition

Réplication procédurale

- Au lieu de transférer des nouvelles (et anciennes) valeurs, la procédure exécutée sur la source est propagée vers le site cible pour y être exécutée

Avantages

- Evite le transfert d 'importants volumes de données

  quand CoûtComm >> CoûtCPU + CoûtDisque

- Conflits de mises à jour
  - évite l 'usage des routines pré-définies

Mécanisme

- Wrapper (Oracle) pour transférer la procédure par une file (Store & Forward)

Notions Complémentaires

- Copie Dérivée (*Derived Copy*)
  - Copie des sous-ensembles de plusieurs tables définis par une requête
- Cliché (*Snapshot*)
  - Copie dérivée asynchrone rafraîchie périodiquement

Produits - Réplication

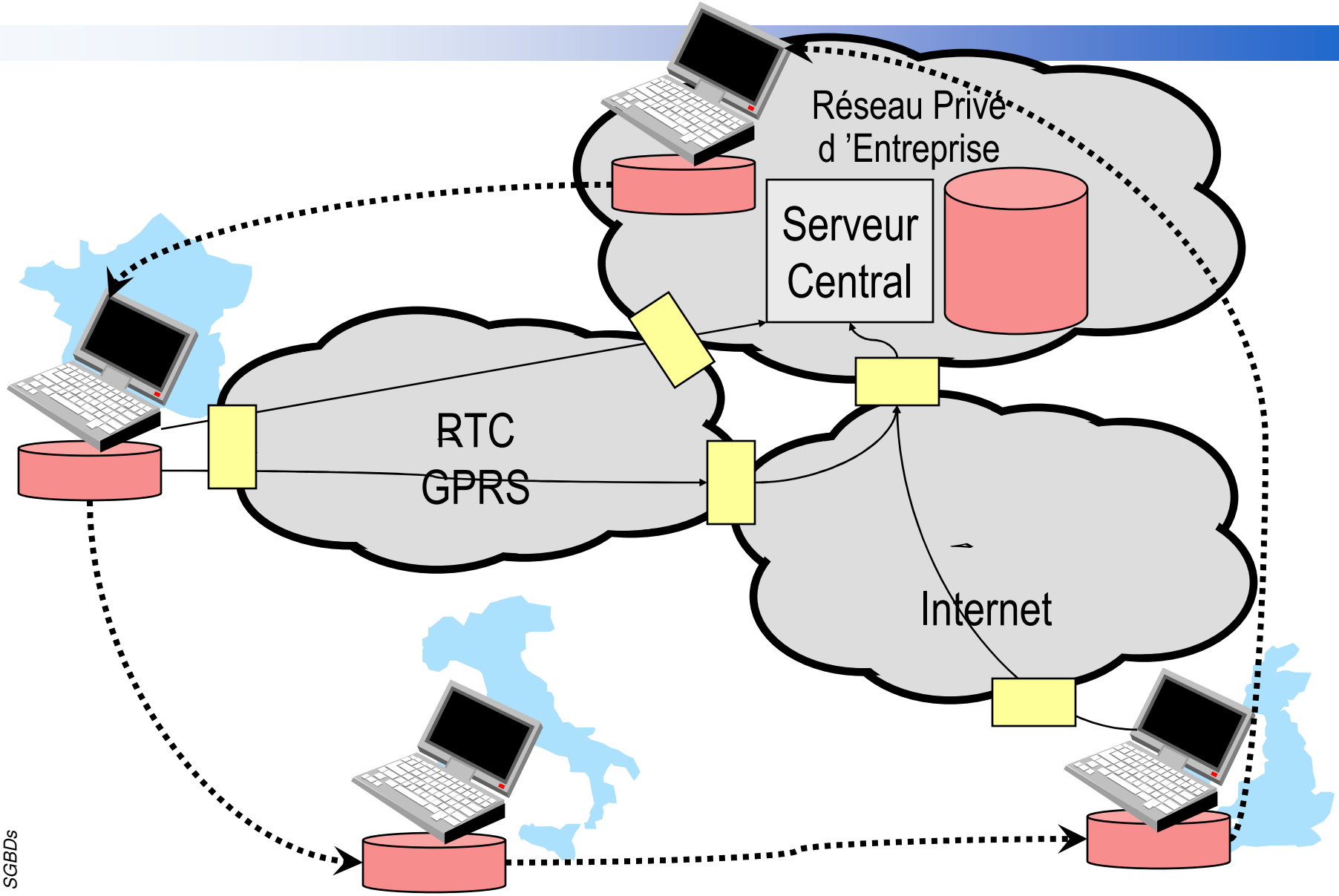
- IBM Data Propagator Relational (Dprop-R)
- Informix
- CA OpenIngres
- Oracle Symetric Replicator
- Sybase Replicator Server
- MicroSoft SQL Server v7
- AFIC Tech. Multi Server Option

Réplicateur	*Sym*	*ASym*	*Sites Hétérogènes*	*Sync*	*Async*	*Fréq. Async Min/Max*	*Initiative Rafraich.*
IBM		Fixe	DRDA	2PC	✓	1min/9sem	cible
Informix		Fixe	Non	2PC	✓	1sec/manuel	source
OpenIngres	✓	Dyn.	cible	2PC	✓	immédiat/manue	source
Oracle	✓	Dyn.	tous	2PC	✓	1sec/manuel	cible
Sybase		Dyn.	cible	2PC	✓	immediat	source
MicroSoft	Ed/So	Fixe		2PC	✓		les 2

BD embarquées et BD Nomades

- Cibles
  - Informatique portable : LapTop
  - Informatique nomade : PDA, Téléphone mobile
  - Informatique enfouie : WebPhone, iTV, Automate monétique …
- Usagers
  - Des collaborateurs mobiles et nomades
    - Représentants, Ingénieurs d 'affaire, Télé-travailleurs, …
- Problème
  - Des communications intermittentes, coûteuses, peu sures
    - Réseaux éphémères add-hoc
  - Faible empreinte mémoire
  - Mémoire stable différente des Disques durs
- Solutions
  - Réplication
  - Synchronisation

Motivations : Usagers Nomades



Les Produits

- Les Personal Editions
  - Open Ingres Desktop
  - IBM/DB2 Universal Server Personal Edition
  - Informix Dynamic Server Personal Edition
  - Microsoft SQL Server (Win2000) et Pocket Access (WinCE)
  - Oracle Lite
  - Sybase Adaptive Server AnyWhere
- Le créneau
  - Centura,
  - Pervasive,
  - …

Gestion des Bases Hétérogènes

- Les Ilots d'Information de l'Entreprise
  - différentes machines :

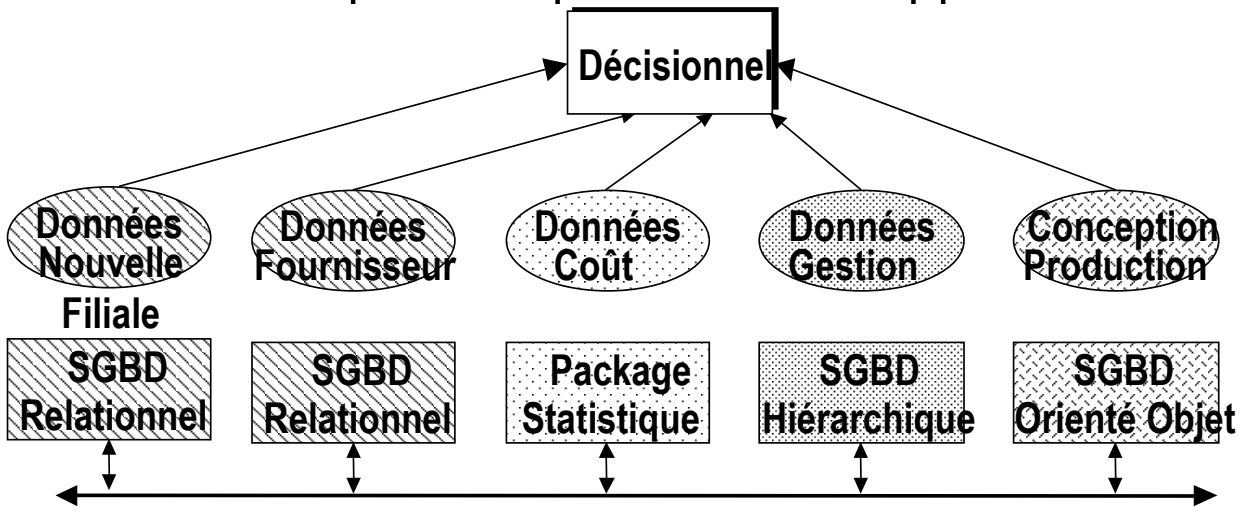
    mainframes, minis, micros, stations de travail
  - différents SGBDs : hiérarchique, relationnel, objet
  - Besoins spécifiques des départements
- Conséquences
  - Redondance des données et inconsistance
  - difficulté de localiser les données "utiles" et de les intégrer rapidement
  - Données "Utiles"

    = consistantes, à jour, et facilement accessibles

⇨Intégration logique des données distribuées et hétérogènes de l'entreprise

L ’intégration des données

- Relationnel ⇒ applications traditionnelles
- Modèles sophistiqués ⇒ applications avancées
- Héritage d’anciens systèmes (legacy systems)
  - 45% des SGBDs sur Mainframe sont IMS (hiérarchique)
  - De nombreuses applications utilisent encore des SGFs (VSAM)
- Intégration des systèmes "non-gestionnaires" de données
  - Traitement de texte, Feuille de calcul, Traitement d’images.
- Utilisation du Web pour simplifier le développement des applications



Les Différents Degrés d'Hétérogénéité

chaque BD est gérée

- par le même SGBD sur des systèmes différents
  - ex: ORACLE sur Linux/Unix, VMS, DOS
- par un SGBD différent mais avec le même modèle
  - ex: ORACLE, DB2, INGRES, SQLServer, MySQL
- par un SGBD différent avec un modèle de données différent
  - ex: DB2 et IMS (problème pratique pour IBM)
- chaque BD est une source de données structurées ou non

Les différentes Formes d'Autonomie

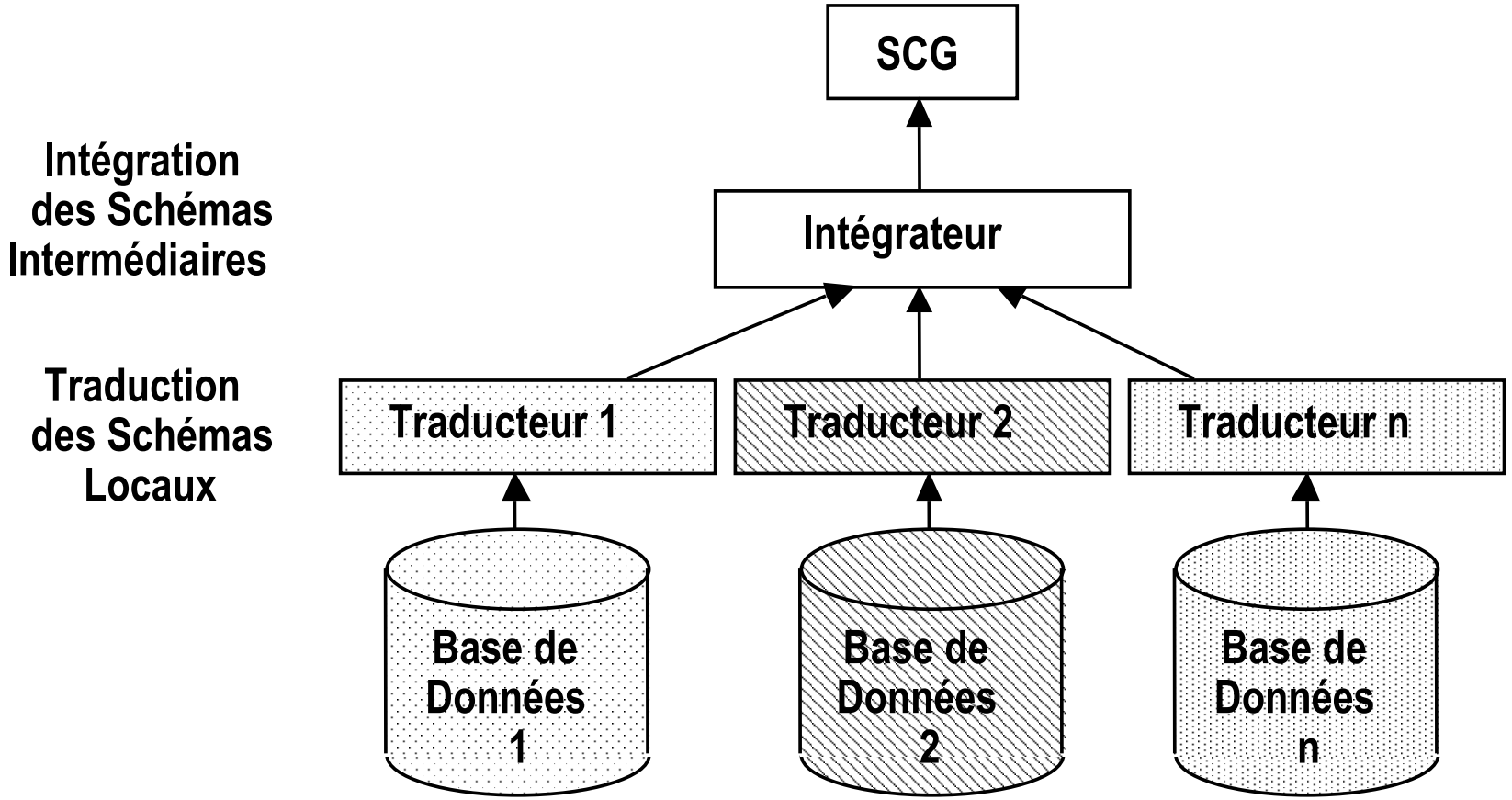
- Autonomie de Conception (*ou Autonomie physique*)
  - un SGBD décide d'une conception reliée à ses propre besoins
    - données, représentation, interprétation, implantation,…
- Autonomie de Communication
  - un SGBD décide de comment et avec quel autre SGBD il doit communiquer
- Autonomie d'Exécution (*ou Autonomie de Site*)
  - un SGBD peut exécuter des opérations localement de la manière qu'il veut

Les Problèmes à Résoudre

- Intégration de Bases de Données
- Calcul des Requêtes
- Gestion des Transactions

Intégration de Bases de données

- c'est le processus d'intégration conceptuelle
  des informations provenant des différentes bases composantes



Intégration de Schéma

- Pré-intégration
  - identification des éléments reliées
    - ex: domaines Equivalents
  - spécification des règles de conversion
    - ex: 1 pouce = 2,54 cm
    - ex: salaire en $ ⇔ DM ⇔ £ ⇔ FF ⇔ €
- Comparaison
  - conflits de nommage
    - synonymes, homonymes
  - conflits structurels
    - types, clés, dépendances
- Conformité
  - résolution des conflits de noms et structurels
- Fusion et Restructuration
  - fusion des schémas intermédiaires
  - fournir le “meilleur” schéma intégré

Calcul des requêtes

- La complexité du calcul des Requêtes est élevée
  - variations d'un SGBD à l'autre
    - pour les possibilités
    - pour les coûts
    - pour l'optimisation
  - difficulté de déplacer les données entre SGBDs

Gestion de Transactions

- Plusieurs transactions à coordonner
  - Terminaison des transactions
    - 2 phases commit largement utilisé
  - Contrôle de Concurrence
    - interopérabilité du Verrouillage distribué
    - interopérabilité de la Détection des interblocages
- Moniteurs Transactionnels
  - Protocoles : *OSI/TP, X/Open, OMG OTS, ...*
  - Produits : *Tuxedo, Encina, CICS …*
    - pas toujours interopérables entre eux

Produits - Gestion BD Hétérogènes

- Oracle
  - SQL*Net : interface réseau
  - SQL*Connect : passerelle vers BD relationnel ou non
- CA-OpenIngres
- ...

Bibliographie

- M. Tamer Özsu, Patrick Valduriez, "Principles of Distributed Database Systems", Prentice-Hall Intl Eds., Second Edition ISBN 0-13-659707-6,1999
- Miranda & Ruols, "Client-Serveur : Concepts, moteurs SQL et architectures paralléles", Ed Eyrolles, ISBN 2-212-08816-7, 1994
- Gardarin & Gardarin, "Le Client-Serveur", Ed Eyrolles, ISBN 2-212-08876-0, 1996
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database System Implementation", 2000, Ed Prentice Hall, ISBN 0-13-040264-8
- Donald Kossmann , « The state of the art in distributed query processing », ACM Computing Surveys , Volume 32 , Issue 4 (December 2000) , pp 422 - 469

Bibliographie

- Saito, Y. and Shapiro, M. 2005. Optimistic replication. ACM Comput. Surv. 37, 1 (Mar. 2005), 42-81.
  - DOI= http://doi.acm.org/10.1145/1057977.1057980