



INSTITUT AFRICAIN
D'INFORMATIQUE

07 BP 12456 Lomé-TOGO

Tel : (00228) 22 20 47 00

Email: iaitogo@iai-togo.com

Site web: www.iai-togo.tg

Exposé de 3ème année GLSI
Bases de données distribuées et
réparties

**Thème : FRAGMENTATION ET ALLOCATION
DES BASES DE DONNEES**

MEMBRES DU GROUPE :

AFATOLO

AFIDEGNON

BINESSI

FOUSSENI

TRAORE

Chargé du cours :

M. AZOTI Hodabalo Pierre

I.	FRAGMENTATION	2
1.	Définition	2
2.	Objectif de la fragmentation	2
3.	Les 12 principaux objectifs d'une base de données répartie définis par C.J. Date	2
4.	Les règles de la fragmentation	3
5.	Les types de fragmentation	3
a.	Fragmentation Horizontale.....	3
i.	Fragmentation horizontale Primaire	3
ii.	Fragmentation horizontale Dérivée	4
b.	Fragmentation verticale	10
c.	Fragmentation mixte	16
6.	Problèmes de la Fragmentation	18
II.	ALLOCATION	19
1.	La réplication totale des données	19
2.	L'absence de réplication.....	19
3.	La méthode hybride	19
4.	Les types d'allocation	19
a.	Allocation centralisée	19
b.	Allocation partitionnée	19
c.	Allocation répliquée	20
d.	Allocation répliquée sélective.....	20
5.	Problèmes de l'allocation	20
III.	Mise à jour d'une base de données reparties.....	20
1.	Insertion	20
2.	Suppression	21
3.	Modification	21
IV.	Répartition avancée	21
1.	L'allocation avec duplication de fragments.....	21
2.	L'allocation dynamique des fragments	22
3.	La fragmentation dynamique	22
	Conclusion	22

I. FRAGMENTATION

1. Définition

La fragmentation est le processus de décomposition d'une base de données en un ensemble de sous bases de données. Cette décomposition doit être sans perte d'information. La fragmentation peut être coûteuse s'il existe des applications qui possèdent des besoins opposés.

2. Objectif de la fragmentation

- Les applications ne travaillent que sur des sous-ensembles des relations.
- Une distribution complète générerait soit trop de trafic soit une réplication avec :
 - Problème de mise à jour ;
 - Problème de stockage
- Equilibrer la charge de travail entre les sites (parallélisme)
- Contrôle de concurrence plus simple pour les accès à un seul fragment
- Favoriser les accès locaux
- Augmenter la disponibilité des données

3. Les 12 principaux objectifs d'une base de données répartie définis par C.J. Date

Les principaux objectifs sont :

1. Transparence pour l'utilisateur
2. Autonomie de chaque site
3. Absence de site privilégié
4. Continuité de service
5. Transparence vis à vis de la localisation des données
6. Transparence vis à vis de la fragmentation
7. Transparence vis à vis de la réplication
8. Traitement des requêtes distribuées
9. Indépendance vis à vis du matériel
10. Indépendance vis à vis du système d'exploitation
11. Indépendance vis à vis du réseau
12. Indépendance vis à vis du SGBD

4. Les règles de la fragmentation

- **La complétude** : Pour toute donnée d'une relation R, il existe un fragment Ri de la relation qui possède cette donnée.
- **La reconstruction** : Pour toute relation R décomposée en un ensemble de fragments Ri, il existe une opération de reconstruction pour trouver R à partir de ses fragments Ri.
- **La disjonction** : Les différents fragments doivent être exclusifs. C'est-à-dire qu'aucune donnée ne doit se retrouver dans plus d'un fragment sauf dans le cas d'une fragmentation verticale où la clé primaire doit se trouver dans tous les fragments.

5. Les types de fragmentation

- Fragmentation Horizontale
- Fragmentation verticale
- Fragmentation mixte

a. Fragmentation Horizontale

C'est un découpage d'une table en sous tables par utilisation de prédicats permettant de sélectionner les lignes appartenant à chaque fragment. L'opération de fragmentation est obtenue grâce à la sélection des tuples d'une table selon un ou des critères bien précis et la reconstitution de la relation initiale se fait grâce à l'union (U) des sous-relations.

Il existe deux types de fragmentation horizontale :

- Fragmentation horizontale Primaire
- Fragmentation horizontale Dérivée

i. Fragmentation horizontale Primaire

La Fragmentation horizontale est définie par l'opération de sélection.

Exemple :

Client (NCL, Nom, Ville) peut être fragmenté :

Client1= SELECT * FROM Client WHERE Ville = «Lomé»

Client2= SELECT * FROM Client WHERE Ville <> «Lomé»

Reconstruction de la relation initiale :

Client = Client1 U Client2

ii. Fragmentation horizontale Dérivée

La Fragmentation d'une table en fonction des fragments horizontaux d'une autre table. (Cette fragmentation est obtenue dans le cas de lien père-fils)

Exemple

Commande (NCL, N°Produit, Date, Qte, N°Représentant)

Commande1= SELECT * FROM Commande

WHERE NCL IN

(SELECT NCL FROM CLIENT1)

Commande2= SELECT * FROM Commande

WHERE NCL IN

(SELECT NCL FROM CLIENT2)

Reconstruction de la relation initiale :

Commande = Commande1 U Commande2

Comment faire la fragmentation Horizontale ?

Considérons la table ETUDIANT suivante :

numE	nomE	pnomE	numC
1	DUPONT	Jean	1
2	DUPOND	Jeanne	2
3	DUBOIS	Robert	1
4	DUBALAI	Aline	1
5	DUGENOU	Jean	2
6	DURAND	Aline	2
7	DURACUIRE	Robert	2
8	DURALUMIN	Roberte	1
9	DURDUR	Jean	2
10	DURALEX	Jean	1

Les prérequis

- i) $\neg C = D$ alors $\neg D = C$ et $\neg C \wedge \neg D = 0$.
- ii) Premier théorème de Morgane $\overline{A \times B} = \bar{A} + \bar{B}$
- iii) Deuxième théorème de Morgane $\overline{A + B} = \bar{A} \times \bar{B}$

D'après les lois fondamentales de l'algèbre de Boole :

- iv) $A + A = A$
- v) $A.A = A$
- vi) $1 + A = 1$
- vii) $A.\bar{A} = 0$

Les requêtes les plus souvent formulées :

- R1 : SELECT * FROM ETUDIANT WHERE numC = 1 ;
- R2 : SELECT * FROM ETUDIANT WHERE nomE LIKE '%R%' ;
- R3 : SELECT * FROM ETUDIANT WHERE pnomE = "Jean" AND numC=2;

Pour la fragmentation horizontale on exploite les clauses WHERE des requêtes.

Exploitation des clauses WHERE des requêtes :

Formalisation

On considère les conditions suivantes :

- $A : \text{numC} = 1$
- $B : \text{nomE LIKE '%R%'}$
- $C : \text{pnomE} = \text{"Jean"}$
- $D : \text{numC} = 2 : \neg A$

Expression des clauses where des requêtes en fonction de A, B, C et D :

- $C1 : A$
- $C2 : B$
- $C3 : C \wedge D = C \wedge \neg A$

Nous avons trois (03) contraintes donc le nombre de conjonctions est $2^3 = 8$.
A partir des conditions C_i , on peut construire l'ensemble des conjonctions CC_j ($i=1..8$) de conditions en utilisant :

$$CC = \left\{ \bigwedge_{i=1..n} C_i^* \text{ où } C_i^* \text{ est soit } C_i \text{ soit } \neg C_i \right\}$$

Donc on a :

$$CC = \{C1 \wedge C2 \wedge C3, \neg C1 \wedge C2 \wedge C3, C1 \wedge \neg C2 \wedge C3,$$

$$C1 \wedge C2 \wedge \neg C3, \neg C1 \wedge \neg C2 \wedge C3, C1 \wedge \neg C2 \wedge \neg C3, \neg C1 \wedge C2 \wedge \neg C3, \neg C1 \wedge \neg C2 \wedge \neg C3\}$$

Expression et simplification des conjonctions en fonction de A, B et C :

Rappel

- $C1 : A$
- $C2 : B$
- $C3 : C \wedge D = C \wedge \neg A$

SIMPLIFICATION

$$1) C1 \wedge C2 \wedge C3 = A \wedge B \wedge (C \wedge \neg A)$$

$$= A.B.C.\bar{A}$$

$$\underline{C1 \wedge C2 \wedge C3 = 0} \quad \text{car } A.\bar{A} = 0$$

$$2) C1 \wedge C2 \wedge \neg C3 = A \wedge B \wedge \neg (C \wedge \neg A)$$

$$= A.B.(\overline{C.A})$$

$$= A.B.(\bar{C} + A)$$

$$= A.B.\bar{C} + A.B.A$$

$$= A.B.\bar{C} + A.B$$

$$= A.B.(\bar{C} + 1)$$

$$\underline{C1 \wedge C2 \wedge \neg C3 = A.B \quad \text{car } \bar{C} + 1 = 1}$$

$$3) C1 \wedge \neg C2 \wedge C3 = A \wedge \neg B \wedge (C \wedge \neg A)$$

$$= A.\bar{B}.C.\bar{A}$$

$$\underline{C1 \wedge \neg C2 \wedge C3 = 0 \quad \text{car } A.\bar{A} = 0}$$

$$4) C1 \wedge \neg C2 \wedge \neg C3 = A \wedge \neg B \wedge \neg (C \wedge \neg A)$$

$$= A.\bar{B}.(\overline{C.A})$$

$$= A.\bar{B}.(\bar{C} + A)$$

$$= A.\bar{B}.\bar{C} + A.\bar{B}$$

$$= A.\bar{B}.(\bar{C} + 1)$$

$$\underline{C1 \wedge \neg C2 \wedge \neg C3 = A.\bar{B} \quad \text{car } \bar{C} + 1 = 1}$$

$$5) \neg C1 \wedge C2 \wedge C3 = \neg A \wedge B \wedge (C \wedge \neg A)$$

$$= \bar{A}.B.C.\bar{A}$$

$$\underline{\neg C1 \wedge C2 \wedge C3 = \bar{A}.B.C \quad \text{car } \bar{A}.\bar{A} = \bar{A}}$$

$$6) \neg C1 \wedge C2 \wedge \neg C3 = \neg A \wedge B \wedge \neg(C \wedge \neg A)$$

$$= \bar{A}.B.(\overline{C.A})$$

$$= \bar{A}.B.(\bar{C}+A)$$

$$= \bar{A}.B.\bar{C} + \bar{A}.B.A$$

$$\underline{\neg C1 \wedge C2 \wedge \neg C3 = \bar{A}.B.\bar{C} \quad \text{car } A.\bar{A} = 0}$$

$$7) \neg C1 \wedge \neg C2 \wedge C3 = \neg A \wedge \neg B \wedge (C \wedge \neg A)$$

$$= \bar{A}.\bar{B}.C.\bar{A}$$

$$\underline{\neg C1 \wedge \neg C2 \wedge C3 = \bar{A}.\bar{B}.C \quad \text{car } \bar{A}.\bar{A} = \bar{A}}$$

$$8) \neg C1 \wedge \neg C2 \wedge \neg C3 = \neg A \wedge \neg B \wedge \neg(C \wedge \neg A)$$

$$= \bar{A}.\bar{B}.(\overline{C.A})$$

$$= \bar{A}.\bar{B}.(\bar{C} + A)$$

$$= \bar{A}.\bar{B}.\bar{C} + \bar{A}.\bar{B}.A$$

$$\underline{\neg C1 \wedge \neg C2 \wedge \neg C3 = \bar{A}.\bar{B}.\bar{C} \quad \text{car } \bar{A}.A = 0}$$

On obtient alors 6 fragments horizontaux

F1 : $C1 \wedge C2 \wedge \neg C3 = A \wedge B$: SELECT * FROM ETUDIANT WHERE numC = 1 AND nomE LIKE '%R%';

F2 : $C1 \wedge \neg C2 \wedge \neg C3 = A \wedge \bar{B}$: SELECT * FROM ETUDIANT WHERE numC = 1 AND nomE NOT LIKE '%R%';

F3 : $\neg C1 \wedge C2 \wedge C3 = \bar{A} \wedge B \wedge C$: SELECT * FROM ETUDIANT WHERE numC = 2 AND nomE LIKE '%R%' AND pnomE = "Jean";

F4 : $\neg C1 \wedge C2 \wedge \neg C3 = \bar{A} \wedge B \wedge \bar{C}$: SELECT * FROM ETUDIANT WHERE numC = 2 AND nomE LIKE '%R%' AND pnomE <> "Jean";

F5 : $\neg C1 \wedge \neg C2 \wedge C3 = \bar{A} \wedge \bar{B} \wedge C$: SELECT * FROM ETUDIANT WHERE numC = 2 AND nomE NOT LIKE '%R%' AND pnomE = "Jean";

F6 : $\neg C1 \wedge \neg C2 \wedge \neg C3 = \bar{A} \wedge \bar{B} \wedge \bar{C}$: SELECT * FROM ETUDIANT WHERE numC = 2 AND nomE NOT LIKE '%R%' AND pnomE <> "Jean";

y

F1 : $A \wedge B$

numE	nomE	pnomE	numC
8	DURALUMIN	Roberte	1
10	DURALEX	Jean	1

F4 : $\neg A \wedge B \wedge \neg C$

numE	nomE	pnomE	numC
6	DURAND	Aline	2
7	DURACUIRE	Robert	2

F2 : $A \wedge \neg B$

numE	nomE	pnomE	numC
1	DUPONT	Jean	1
3	DUBOIS	Robert	1
4	DUBALAI	Aline	1

F5 : $\neg A \wedge \neg B \wedge C$

numE	nomE	pnomE	numC
5	DUGENOU	Jean	2

F3 : $\neg A \wedge B \wedge C$

numE	nomE	pnomE	numC
9	DURDUR	Jean	2

F6 : $\neg A \wedge \neg B \wedge \neg C$

numE	nomE	pnomE	numC
2	DUPOND	Jeanne	2

b. Fragmentation verticale

La fragmentation verticale est obtenue par décomposition de la table en groupes de colonnes. Fragmentation verticale est définie par l'opération de projection.

Table Client (N°Client, Nom, Sexe, Ville)

Exemple

Client (N°Client, Nom, Sexe, Ville) peut être fragmentée :

Client1= SELECT N°Client, Nom FROM Client

Client2= SELECT N°Client, Sexe, Ville FROM Client

Reconstruction de la relation initiale :

Client = Client1 ⋈ Client2

Comment faire la fragmentation Verticale ?

Considérons la table ETUDIANT suivante :

numE	nomE	pnomE	numC
1	DUPONT	Jean	1
2	DUPOND	Jeanne	2
3	DUBOIS	Robert	1
4	DUBALAI	Aline	1
5	DUGENOU	Jean	2
6	DURAND	Aline	2
7	DURACUIRE	Robert	2
8	DURALUMIN	Roberte	1
9	DURDUR	Jean	2
10	DURALEX	Jean	1

Les requêtes les plus souvent formulées :

- R1 : SELECT nomE, pnomE FROM ETUDIANT WHERE numC = 1 ;
- R2 : SELECT pnomE FROM ETUDIANT WHERE nomE LIKE '%R%' ;
- R3 : SELECT numE FROM ETUDIANT WHERE pnomE="Jean" AND numC=2;
- Définissons un univers : Univers U = {numE, nomE, pnomE, numC}

Pour la fragmentation verticale on exploite les clauses SELECT des requêtes.

Exploitation des clauses SELECT des requêtes :

Formalisation

On considère les projections suivantes :

- $P1 = \text{nomE}, \text{pnomE}$ pour $R1$
- $P2 = \text{pnomE}$ pour $R2$
- $P3 = \text{numE}$ pour $R3$

Construction d'un dual : $\sim P_i = U - P_i$ (Univers U privé de P_i)

- $\sim P1 = \text{numE}, \text{numC}$
- $\sim P2 = \text{numE}, \text{nomE}, \text{numC}$
- $\sim P3 = \text{nomE}, \text{pnomE}, \text{numC}$

On effectue alors le calcul des **Intersections de Projection**.

Mais qu'est-ce qu'une Intersection de projection ?

Intersections de projection

❖ Intersection de projection pour une requête satisfaite

Soit F un fragment satisfaisant $R1$ de projection $P1$

$$\rightarrow IP = \{ (P1), (\sim P1) \}$$

Attributs utiles pour $R1$ Attributs inutiles pour $R1$

❖ Intersection de projection pour deux requêtes satisfaites

Soit F un fragment satisfaisant $R1$ de projection $P1$ et $R2$ de projection $P2$

$$\rightarrow IP = \{ (P1 \cap P2), (\sim P1 \cap P2), (P1 \cap \sim P2), (\sim P1 \cap \sim P2) \}$$

attributs utiles pour $R1$ et $R2$ attributs utiles pour $R2$ attributs utiles pour $R1$ attributs inutiles pour $R1$ et $R2$

❖ **Généralisation à un fragment satisfaisant n requêtes**

$$IP = \{ \bigcap_{j=1}^n P_j^* \mid P_j^* = P_j \text{ ou } \sim P_j \}$$

RAPPEL

Requêtes

- R1 : SELECT nomE, pnomE FROM ETUDIANT WHERE numC = 1 ;
- R2 : SELECT pnomE FROM ETUDIANT WHERE nomE LIKE '%R%' ;
- R3 : SELECT numE FROM ETUDIANT WHERE pnomE="Jean" AND numC=2;

Univers U = {numE, nomE, pnomE, numC}

Projections

- P1 = nomE, pnomE pour R1
- P2 = pnomE pour R2
- P3 = numE pour R3

Duals de projections

- $\sim P1$ = numE, numC
- $\sim P2$ = numE, nomE, numC
- $\sim P3$ = nomE, pnomE, numC

Construction des intersections de projection

- **Pour F1 (satisfaisant R1 et R2) :**

On a : $IP1 = \{ P1 \cap P2, P1 \cap \sim P2, \sim P1 \cap P2, \sim P1 \cap \sim P2 \}$

→ $IP1 = \{ (pNomE), (nomE), (), (numE, numC) \}$

- **Pour F2 (satisfaisant R1) :**

On a : $IP2 = \{ P1, \sim P1 \}$

→ $IP2 = \{ (nomE, pnomE), (numE, numC) \}$

➤ **Pour F3 (satisfaisant R2 et R3):**

On a : $IP3 = \{ P2 \cap P3, P2 \cap \sim P3, \sim P2 \cap P3, \sim P2 \cap \sim P3 \}$

→ $IP3 = \{ (), (pNomE), (numE), (numC, nomE) \}$

➤ **Pour F4 (satisfaisant R2) :**

On a : $IP4 = \{ P2, \sim P2 \}$

→ $IP4 = \{ (pnomE), (numE, nomE, numC) \}$

➤ **Pour F5 (satisfaisant R3):**

On a : $IP5 = \{ P3, \sim P3 \}$

→ $IP5 = \{ (numE), (nomE, pnomE, numC) \}$

➤ **Pour F6 (satisfaisant personne) :**

→ Ce fragment n'intéresse personne, il est donc inutile de le fragmenter verticalement

Pour chaque élément d'un IP, ajouter la clé si elle n'existe pas déjà et définir la projection

Pour F1 (satisfaisant R1 et R2):

On a : $IP1 = \{(pNomE), (nomE), (), (numE, numC)\}$

$F11 = \Pi_{numE, pnomE} (F1)$

$F12 = \Pi_{numE, nomE} (F1)$

$F13 = \Pi_{numE, numC} (F1)$

Pour F2 (satisfaisant R1) :

On a : $IP1 = \{(nomE, pnomE), (numE, numC)\}$

$F21 = \Pi_{numE, nomE, pnomE} (F2)$

$F22 = \Pi_{numE, numC} (F2)$

Pour F3 (satisfaisant R2 et R3):

On a : $IP3 = \{(), (pNomE), (numE), (numC, nomE)\}$

$F31 = \Pi_{numE, pnomE} (F3)$

~~$F32 = \Pi_{numE} (F3)$~~

(On n'a pas besoin de ce fragment car il n'est composé que de la clé primaire)

$F33 = \Pi_{numE, nomE, numC} (F3)$

Pour F4 (satisfaisant R2) :

On a : $IP4 = \{(pnomE), (numE, nomE, numC)\}$

$F41 = \Pi_{numE, pnomE} (F4)$

$F42 = \Pi_{numE, nomE, numC} (F4)$

Pour F5 (satisfaisant R3):

On a : : IP5= {(numE), (nomE,pnomE,numC)}

~~F51 = Π_{numE} (F5)~~

~~F52 = $\Pi_{\text{numE, nomE,pnomE,numC}}$ (F5)~~

(On n'a pas besoin de ces fragments)

Pour F6 (satisfaisant personne) :

Ce fragment n'intéresse personne, il est donc inutile de le fragmenter verticalement

Après les opérations ci-dessus les fragments verticaux obtenus sont :

F11 = $\Pi_{\text{numE, pnomE}}$ (F1)

F12 = $\Pi_{\text{numE, nomE}}$ (F1)

F13 = $\Pi_{\text{numE, numC}}$ (F1)

F21 = $\Pi_{\text{numE, nomE,pnomE}}$ (F2)

F22 = $\Pi_{\text{numE, numC}}$ (F2)

F31 = $\Pi_{\text{numE,pnomE}}$ (F3)

F33 = $\Pi_{\text{numE, nomE, numC}}$ (F3)

F41 = $\Pi_{\text{numE, pnomE}}$ (F4)

F42 = $\Pi_{\text{numE,nomE,numC}}$ (F4)

c. Fragmentation mixte

La Fragmentation mixte résulte de l'application successive d'opérations de fragmentation horizontale et de fragmentation verticale. Par exemple on peut décider de fragmenter verticalement les fragments horizontaux créés précédemment ou aussi fragmenter horizontalement les fragments verticaux créés précédemment.

Exercice :

R1: SELECT numero, numeq FROM CUISINIER WHERE prenom = 'Jean' AND nom LIKE '%R%';

R2: SELECT * FROM CUISINIER WHERE numeq = '1';

R3: SELECT numero, nom FROM CUISINIER WHERE numeq = '2' AND prenom = 'Jean';

Table

Cuisinier

NUMERO	NOM	PRENOM	NUMEQ
12	DUPONT	Jean	1
34	DUPONT	Jeanne	2
17	DUBOIS	Robert	1
22	DUBALAI	Aline	1
26	DUGENOU	Jean	2
11	DURAND	Aline	2
38	DURACUIRE	ROBERT	2
9	DURALUMIN	Roberte	1
13	DURDUR	Jean	2
20	DURALEX	Jean	1

Fragmentation Horizontale :

F1	NUMERO	NOM	PRENOM	NUMEQ
	13	DURDUR	Jean	2
F2	NUMERO	NOM	PRENOM	NUMEQ
	20	DURALEX	Jean	1
F3	NUMERO	NOM	PRENOM	NUMEQ
	26	DUGENOU	Jean	2
F4	NUMERO	NOM	PRENOM	NUMEQ
	12	DUPONT	Jean	1
	17	DUBOIS	Robert	1
	22	DUBALAI	Aline	1
	9	DURALUMIN	Roberte	1
F5	NUMERO	NOM	PRENOM	NUMEQ
	34	DUPONT	Jeanne	2
	11	DURAND	Aline	2
	38	DURACUIRE	ROBERT	2

Fragments
horizontaux

Fragmentation verticale des fragments Horizontaux :

F11	F12	F13	F14
NUMERO	NUMERO NOM	NUMERO NUMEQ	NUMERO PRENOM
13	13 DURDUR	13 2	13 Jean
F21	F22		
NUMERO NUMEQ	NUMERO NOM PRENOM		
20 1	20 DURALEX Jean		
F31	F32		
NUMERO NOM	NUMERO PRENOM NUMEQ		
26 DUGENOU	26 Jean 2		
F41			
NUMERO NOM PRENOM NUMEQ			
12 DUPONT Jean 1			
17 DUBOIS Robert 1			
22 DUBALAI Aline 1			
9 DURALUMIN Roberte 1			
F51			
NUMERO NOM PRENOM NUMEQ			
34 DUPONT Jeanne 2			
11 DURAND Aline 2			
38 DURACUIRE ROBERT 2			

6. Problèmes de la Fragmentation

- **Coût** : la distribution entraîne des coûts supplémentaires en termes de communication, et en gestion des communications (hardware et software à installer pour gérer les communications et la distribution).
- **Sécurité** : la sécurité est un problème plus complexe dans le cas des bases de données réparties que dans le cas des bases de données centralisées.

II. ALLOCATION

Lorsque le concepteur a fini de fragmenter sa base, il lui faut ensuite allouer chaque fragment sur son site correspondant.

But : placer les fragments afin de minimiser le transfert de données entre les sites et ainsi optimiser le système. Cette affectation sur les sites est décidée en fonction de l'origine prévue des requêtes qui ont servi à la fragmentation.

L'allocation peut être faite de plusieurs façons. Les méthodes d'allocation sont :

- **La réplication totale des données**
- **L'absence de réplication**
- **La méthode hybride**

1. La réplication totale des données

Cette méthode n'est pas très efficace lorsque les données sont régulièrement mises à jour car il se pose le problème de cohérence de données

2. L'absence de réplication

Chaque donnée est mise à jour sur un seul site. Cette méthode est plus efficace quand les données sont beaucoup plus modifiées que lues.

3. La méthode hybride

Afin de bénéficier des deux méthodes citées à la fois, celle hybride peut être utilisée. Ainsi les données en Read Only (Lecture seule) peuvent être répliquées et les données en Read Write (Lecture et Ecriture) pas du tout.

4. Les types d'allocation

a. Allocation centralisée

- Taille base limitée par les capacités du site central
- Disponibilité faible

b. Allocation partitionnée

- Limite l'espace disque local
- Meilleur si la fiabilité de la solution centralisée n'est pas suffisante
- Intéressant s'il y a une partie importante de traitement local

c. Allocation répliquée

- Une copie de la base sur chaque site
- Maximise la fiabilité
- Possible si la base est petite et si l'inefficacité des mises à jour est tolérable

d. Allocation répliquée sélective

- Les fragments critiques sont répliqués
- Les fragments non critiques sont localisés sur un seul site
- Grande flexibilité

5. Problèmes de l'allocation

Soient :

- $F = \{F1, F2, \dots, Fn\}$ - ensemble de fragments
- $S = \{S1, S2, \dots, Sm\}$ - ensemble de sites
- $Q = \{Q1, Q2, \dots, Qp\}$ - ensemble de requêtes

Afin d'améliorer la performance de Q, il faut trouver une distribution optimale de F sur S.

Autres contraintes :

- La gestion du stockage
- L'équilibrage de charge entre les sites

III. Mise à jour d'une base de données reparties

La principale difficulté réside dans le fait qu'une mise à jour dans une relation du schéma global se traduit par plusieurs mises à jour dans différents fragments. Il faut donc identifier les fragments concernés par l'opération de mise à jour, puis décomposer en conséquence l'opération en un ensemble d'opération de mise à jour sur ces fragments.

1. Insertion

Retrouver le fragment horizontal concerné en utilisant les conditions qui définissent les fragments horizontaux, puis insertion du tuple dans tous les fragments verticaux correspondants.

2. Suppression

Rechercher le tuple dans les fragments qui sont susceptibles de contenir le tuple concerné, et supprimer les valeurs d'attribut du tuple dans tous les fragments verticaux.

3. Modification

Rechercher les tuples, les modifier et les déplacer vers les bons fragments si nécessaire.

IV. Répartition avancée

1. L'allocation avec duplication de fragments

Dans ce type de répartition avancée, Certains fragments peuvent être dupliqués sur plusieurs sites (éventuellement sur tous les sites)

Avantages :

- ❖ Elle permet d'améliorer les performances en termes de temps d'exécution des requêtes (en évitant certains transferts de données).
- ❖ Elle permet une meilleure disponibilité des informations (connues de plusieurs sites),
- ❖ Elle permet une meilleure fiabilité contre les pannes.

Inconvénient :

- Les mises à jour doivent être effectuées sur toutes les copies d'une même donnée.

En conséquence :

- Moins un fragment est sujet à des modifications et plus il est prédisposé à la duplication.

Il existe deux types de duplication : Duplication synchrone et Duplication asynchrone.

Duplication synchrone :

- La sérialisabilité est assurée sur l'ensemble des nœuds
- Une transaction est confirmée seulement lorsque tous les sites ont été mis-à-jour

Duplication asynchrone :

- Les mises-à-jour sont d'abord faites sur une copie primaire
- Les sites de réplication sont mis-à-jour en différé, à partir de la copie primaire, après la confirmation de la transaction
- Ex. d'implémentation : vues matérialisées (ou SNAPSHOTS)

2. L'allocation dynamique des fragments

Ici, l'allocation d'un fragment peut changer en cours d'utilisation de la BDR :

- Le schéma d'allocation et les schémas locaux doivent être tenus à jour.
- Elle peut être utilisée comme une alternative à la duplication : Elle se révèle plus efficace lorsque la base de données est disposée à de nombreuses mises à jour

3. La fragmentation dynamique

- Dans le cas où le site d'allocation peut changer dynamiquement, il est possible que deux fragments complémentaires (verticalement ou horizontalement) se retrouvent sur le même site. Il est alors normal de les fusionner.
- A l'inverse, si une partie d'un fragment est appelée sur un autre site, il peut être intéressant de décomposer ce fragment et de ne faire migrer que la partie concernée.
 - Ces modifications du schéma de fragmentation se reflètent sur le schéma d'allocation et sur les schémas locaux.

Conclusion

Les SGBD distribués offrent une autonomie des sites ainsi qu'une distribution de l'administration des données. Ainsi la fragmentation et la répartition de ces données sur plusieurs sites permettent de répartir la charge des SGBD sur plusieurs sites et ainsi d'optimiser le trafic de données.