Modélisation et protection contre les pannes dans les bases de données

Soheir Abdel Hay Abdel Hamid

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR INGENIEUR

par

Soheir Abdel Hay Abdel Hamid

MODELISATION ET PROTECTION CONTRE LES PANNES DANS LES BASES DE DONNEES.

Soutenue le 24 janvier 1979 devant la commission d'examen

G. VEILLON	Président
L. BOLLIET	Examinateurs
G. SAUCIER	
E. GELENBE	
B. VAN CUTSEM	
M. ADIBA	
M. GAULENE	

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1977-1978

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET
M. Georges LESPINARD

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Electronique - automatique
	BESSON Jean	Chimie minérale
	BLOCH Daniel	Physique du solide - cristallographie
	BONNETAIN Lucien	Génie chimique
	BONNIER Etienne	Métallurgie
*	BOUDOURIS Georges	Electronique - automatique
	BRISSONNEAU Pierre	Physique du solide - cristallographie
	BUYLE-BODIN Maurice	Electronique - automatique
	COUMES André	Electronique - automatique
	DURAND Francis	Métallurgie
	FELICI Noël	Electronique - automatique
	FOULARD Claude	Electronique - automatique
	LANCIA Roland	Electronique - automatique
	LONGEQUEUE Jean-Pierre	Physique nucléaire corpusculaire
	LESPINARD Georges	Mécanique
	MOREAU René	Mécanique
	PARIAUD Jean-Charles	Chimie - physique
	PAUTHENET René	Electronique - automatique
	PERRET René	Electronique - automatique
	POLOUJADOFF Michel	Electronique - automatique
	TRAYNARD Philippe	Chimie - physique
	VEILLON Gérard	Informatique fondamentale et appliquée

* en congé pour études

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuël	Electronique - automatique
	BOUVARD Maurice	Génie mécanique
	COHEN Joseph	Electronique - automatique
	GUYOT Pierre	Métallurgie physique
	LACOUME Jean-Louis	Electronique - automatique
	JOUBERT Jean-Claude	Physique du solide - cristallographie

.../...

MM.	ROBERT André	Chimie appliquée et des matériaux
	ROBERT François	Analyse numérique
	ZADWORNY François	Electronique - automatique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	CHARTIER Germain	Electronique - automatique
	CHIAVERINA Jean	Biologie, biochimie, agronomie
	IVANES Marcel	Electronique - automatique
	LESIEUR Marcel	Mécanique
	MORET Roger	Physique nucléaire - corpusculaire
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	SABONNADIERE Jean-Claude	Informatique fondamentale et appliquée
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie Physique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahim	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

MM.	BISCONDI Michel	Ecole des Mines St. Etienne (dépt. Métallurgie)
	BOOS Jean-Yves	Ecole des Mines St. Etienne (Métallurgie)
	DRIVER Julian	Ecole des Mines St. Etienne (Métallurgie)

.../...

MM.	KOBYLANSKI André	Ecole des Mines St. Etienne (Métallurgie)
	LE COZE Jean	Ecole des Mines St. Etienne (Métallurgie)
	LESBATS Pierre	Ecole des Mines St. Etienne (Métallurgie)
	LEVY Jacques	Ecole des Mines St. Etienne (Métallurgie)
	RIEU Jean	Ecole des Mines St. Etienne (Métallurgie)
	SAINFORT	C.E.N. Grenoble (Métallurgie)
	SOUQUET	U.S.M.G.
	CAILLET Marcel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	COULON Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	GUILHOT Bernard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	LALAUZE René	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	LANCELOT Francis	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	SARRAZIN Pierre	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	SOUSTELLE Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	THEVENOT François	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	THOMAS Gérard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	TOUZAIN Philippe	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	TRAN MINH Canh	Ecole des Mines St. Etienne (Chim. Min. Ph.)

E.N.S.E.R.G.

MM.	BOREL	Centre d'études nucléaires de Grenoble
	KAMARINOS	Centre national recherche scientifique

E.N.S.E.G.P.

M.	BORNARD	Centre national recherche scientifique
Mme	CHERUY	Centre national recherche scientifique
MM.	DAVID	Centre national recherche scientifique
	DESCHIZEAUX	Centre national recherche scientifique

Je remercie vivement

Monsieur G. VEILLON, Professeur à l'Institut National Polytechnique de Grenoble, de m'avoir fait l'honneur de présider le jury de cette thèse.

Je suis également reconnaissante à Madame G. SAUCIER, Maître de Conférences à l'ENSIMAG, qui m'a accueillie dans son équipe, qui a guidé ce travail par des suggestions et des critiques constructives et qui par son enthousiasme et ses conseils m'a constamment encouragée.

Je remercie Monsieur E. GELENBE, Maître de Conférences à l'Université Paris-Sud, pour l'intérêt qu'il a accordé à ce travail et pour avoir accepté de le juger.

Je suis très honorée de la présence de Monsieur L. BOLLIET, Professeur à l'Institut Universitaire de Technologie de Grenoble, qui a bien voulu accepter de faire partie du jury malgré ses nombreuses occupations.

Je remercie également

Monsieur ADIBA, Maître Assistant à l'Université des Sciences Sociales,
Monsieur B. VAN CUTSEM, Professeur à l'Université I,
Monsieur GAULENE, Directeur Scientifique à la Société SEMS
qui ont bien voulu accepter de faire partie du jury.

Je voudrais remercier aussi tous les membres de l'équipe "Conception et Sécurité des Systèmes Logiques", et en particulier P. CASPI pour les fructueuses conversations qui m'ont permise d'améliorer ce travail.

Je tiens aussi à remercier Madame G. DUFFOURD qui a assuré la dactylographie de ce mémoire avec dévouement, ainsi que Monsieur D. IGLESIAS et le service de reprographie pour le soin qu'ils ont apporté à la réalisation matérielle de ce document.

SOMMAIRE

CHAPITRE III : GENERALITES SUR LA SURETE DE FONCTIONNEMENT DES SGBD. MODELES ANALYTIQUES DES STRATEGIES DE REDEMARRAGE DES SGBD

INTRODUCTION

Un système de Gestion de Bases de Données, SGBD est un ensemble de logiciel offrant un certain nombre de services pour gérer des données centralisées, partagées par plusieurs applications.

Un SGBD doit fournir les fonctions suivantes :

1) La réalisation des opérations de base sur les enregistrements des fichiers de la BD, il doit permettre la définition, la création, la modification, l'accès et la mise à jour de ces enregistrements.

2) Des procédures garantissant les récupérations de l'information contre tout type d'avarie, pour aboutir à une BD fiable et sûre.

3) Des techniques de discrétion évitant que n'importe qui puisse avoir accès à n'importe quoi.

4) Permettre l'interaction et l'exécution des programmes conversationnels.

5) Offrir un certain nombre d'outils pour que l'administrateur de la base puisse évaluer et améliorer les performances du système.

Notre attention se portera essentiellement sur les points 2 et 5. Notre travail est divisé en deux parties distinctes. La première partie, Chapitre I et II s'intéressera à l'évaluation du coût et des performances des SGBD. La sûreté de fonctionnement des SGBD sera le sujet de la deuxième partie, Chapitre III.

Dans tout ce travail, l'objectif sera de prendre en compte, au mieux, l'organisation matérielle du SGBD. La prise en compte de l'implémentation physique au niveau fonctionnel permettra seule une approche efficace pour l'étude des performances et de la fiabilité.

Dans le Chapitre I, on présentera brièvement quelques organisations physiques des BD. Un modèle graphique utilisant les Réseaux de Petri sera proposé dans ce chapitre pour la représentation fonctionnelle des SGBD.

Dans le second Chapitre, une étude des différents modèles pour l'évaluation des performances des SGBD est faite. On proposera une amélioration d'un de ces modèles.

Le Chapitre III donnera quelques généralités sur la sûreté de fonctionnement des SGBD avec une description succinte des causes de panne. Les différentes stratégies habituellement employées pour le redémarrage des SGBD après la détection d'une panne seront présentées. Les modèles analytiques pour le calcul du temps optimal entre les points de reprise pour minimiser le coût des pannes sont ensuite passés en revue. Un modèle analytique d'un SGBD utilisant des fichiers différentiels sera proposé. Il permet de calculer le temps optimal entre les fusions des fichiers différentiels et la BD, en minimisant le coût du système par unité de temps.

CHAPITRE I

ORGANISATIONS PHYSIQUES DES BD
MODELISATION FONCTIONNELLE DES SGBD

-o-o-

I.1 INTRODUCTION

Les organisations physiques des enregistrements des BD sont nombreuses et très variées [DATE 77], [MART 75]. Ces organisations ont une grande influence sur les performances des SGBD. Seules les organisations multilistes, listes inversées et listes cellulaires seront considérées. Dans le paragraphe I.2, une description résumée de ces trois organisations sera donnée.

Un modèle fonctionnel des SGBD sera proposé dans le paragraphe I.3. Il représente les chemins d'accès pour constituer la réponse aux requêtes des utilisateurs des B.D. Pour cette modélisation, les Réseaux de Petri, RdP, sont utilisés, facilitant la compréhension, l'évaluation et la comparaison des différentes implémentations des SGBD. Ceci est dû à la possibilité de la description fonctionnelle de ces systèmes à différents niveaux de détail très divers.

I.2 ORGANISATIONS PHYSIQUES

Les enregistrements des fichiers des BD ont plusieurs mots clés et ils sont normalement rangés sur le support physique dans l'ordre ascendant d'une seule clé, qui est une clé primaire unique qui identifie l'enregistrement. Les autres clés sont des clés secondaires. Les enregistrements ayant une clé secondaire de valeur K_v, susceptible d'être un critère de recherche lors de l'exploitation du fichier, forment un ensemble $E_r\{K_v\}$. Cet ensemble est défini comme possédant une certaine propriété P, ou un mot clé K_v.

I.2.1. L'ORGANISATION MULTILISTE (Fig.I.1)

Dans cette organisation, des pointeurs lient les enregistrements de chacun des ensembles $E_r\{K_v\}$, formant ainsi des listes de longueur variable. Un dictionnaire est composé, dans lequel on associe à chaque mot-clé, K_v, de chaque attribut, l'adresse du premier enregistrement et le nombre d'enregistrements dans l'ensemble $E_r\{K_v\}$.

L'organisation multiliste est bien adaptée à la recherche d'un ensemble $E_r\{K_v\}$, elle l'est moins bien lors de la recherche des enregistrements résultant de l'intersection de deux ou plusieurs ensembles $E_r\{K_{v1}\} \cap E_r\{K_{v2}\} \ldots \cap E_r\{K_{vm}\}$. Il faudra amener en mémoire centrale tous les enregistrements du plus petit ensemble $E_r\{K_{vi}\}$, et les examiner un à un pour extraire les enregistrements demandés. Ce nombre d'enregistrements amenés en mémoire centrale, peut évidemment excéder de beaucoup le nombre d'enregistrements recherchés.

Dans le cas de l'union, $E_r\{K_{v1}\} \cup E_r\{K_{v2}\} \ldots \cup E_r\{K_{vm}\}$, tous les ensembles $E_r\{K_{v1}\}$, $E_r\{K_{v2}\}$ et $E_r\{K_{vm}\}$ sont amenés en mémoire centrale. Un même enregistrement peut être amené plusieurs fois en mémoire centrale, ce qui ajoute un autre point de faiblesse à l'organisation multiliste.

Par contre, cette organisation est facile à mettre en oeuvre, l'addition et la suppression des enregistrements est simple.

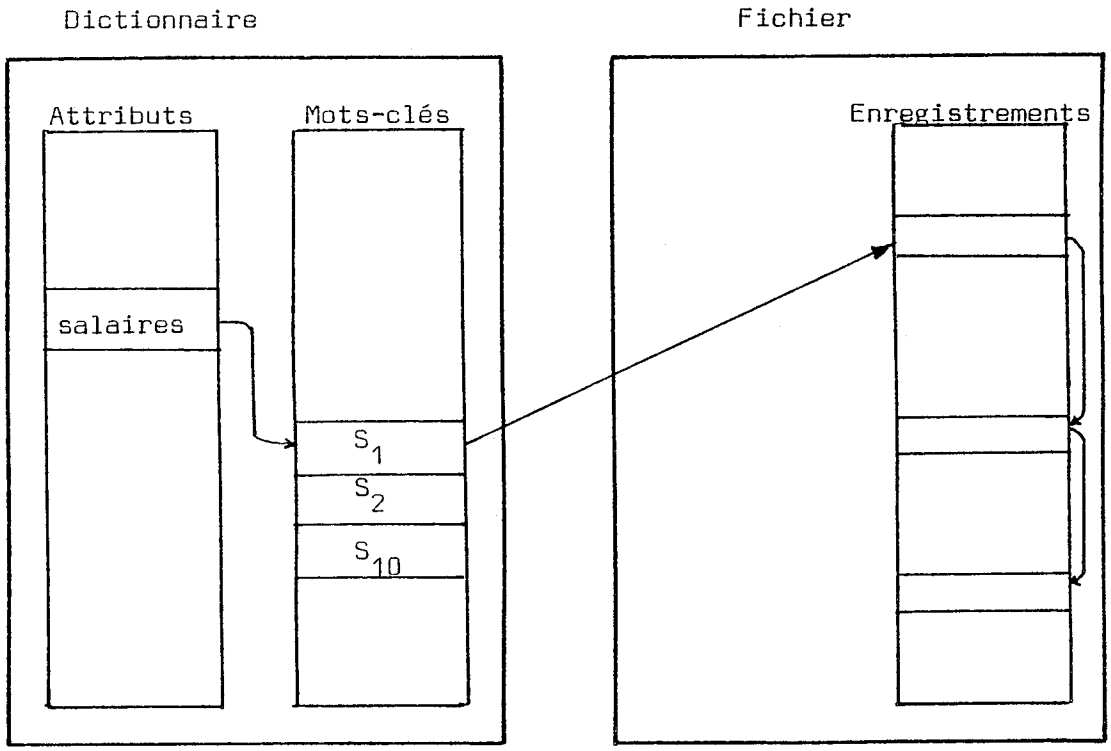


Figure I.1 : Organisation Multilistes

I.2.2. L'ORGANISATION LISTES INVERSEES (Fig.I.2)

Dans cette organisation, les adresses des enregistrements des ensembles $E_r\{K_v\}$ sont stockées en mémoire, formant des listes d'accès à ces ensembles. Dans le dictionnaire, on associe à chaque mot clé, l'adresse de sa liste d'accès.

Si l'on recherche des enregistrements possédant l'intersection ou la réunion de plusieurs propriétés, il suffit d'effectuer l'intersection ou la réunion des adresses contenues dans les listes d'accès, pour déterminer les adresses des enregistrements recherchés. Seuls ces enregistrements sont amenés en mémoire centrale, ce qui permet un temps de réponse trop court et l'utilisation de cette organisation dans les systèmes fonctionnant en temps réel.

Si la taille du dictionnaire et des listes d'accès augmente, ils sont stockés sur la mémoire secondaire en plusieurs niveaux, ce qui augmente leur temps de recherche et de mise à jour, faisant perdre en partie à l'organisation liste inversée son avantage essentiel.

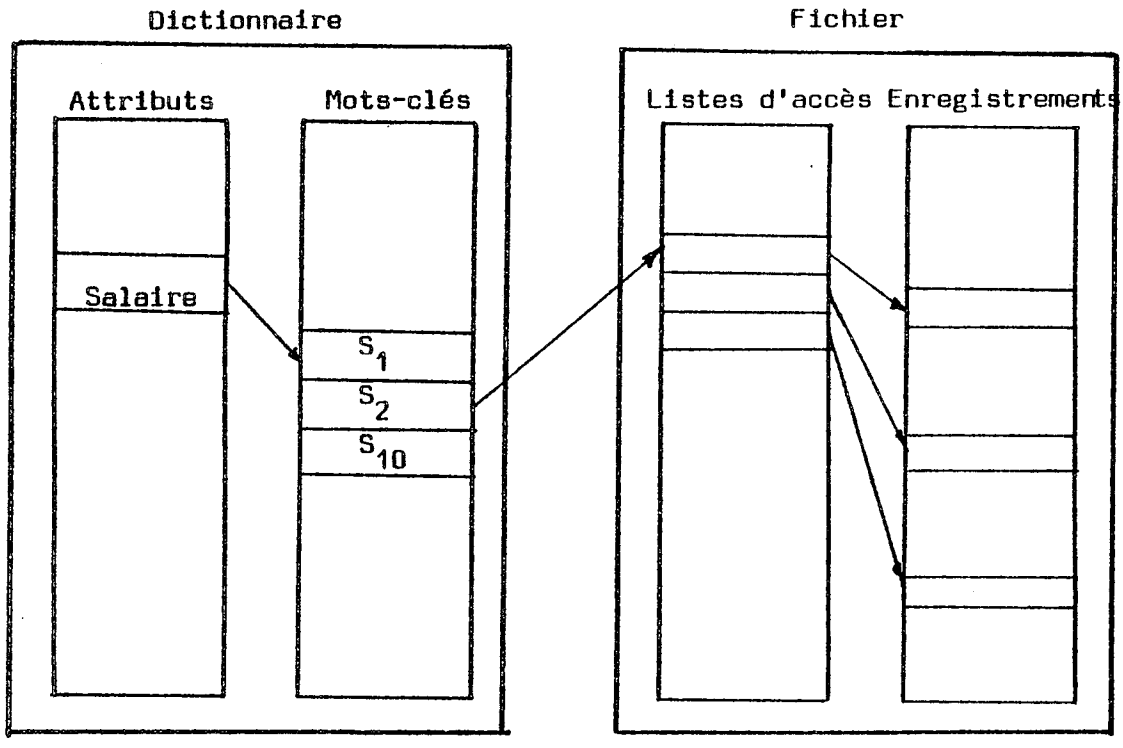


Figure I.2. Organisation listes inversées

I.2.3. L'ORGANISATION CELLULAIRE (Fig.I.3)

L'organisation cellulaire est un compromis entre les organisations multilistes et listes inversées. L'espace alloué au fichier est découpé en cellules distinctes. Ces cellules peuvent être des pistes, des cylindres ou des disques.

Comme dans l'organisation multiliste, les enregistrements possédant une même propriété sont chaînés les uns aux autres. Cependant, au lieu de les réunir en une seule liste, on les regroupe en listes partielles, chacune de ces listes partielles correspond à une cellule.

Comme dans l'organisation liste inversée, plusieurs adresses d'enregistrements correspondent à une même entrée du dictionnaire.

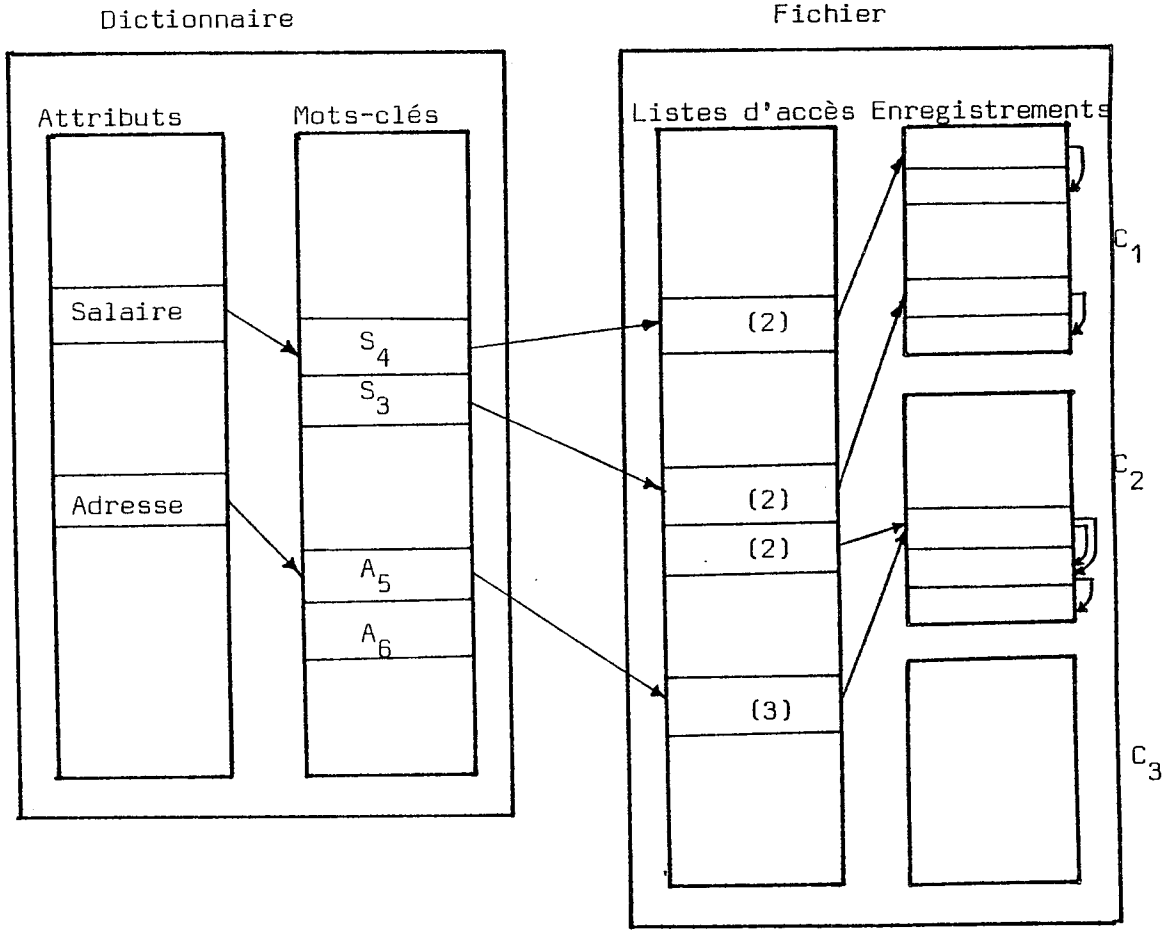


Figure I.3. Organisation cellulaire

Ici, ce sont les adresses des premiers enregistrements des listes partielles. Les remarques à faire sur cette organisation sont :

- Le dictionnaire et les listes d'accès sont moins encombrés que dans l'organisation listes inversées.
- La recherche d'enregistrement vérifiant l'intersection de plusieurs propriétés conduit à un nombre de transfert d'enregistrements inutiles moins grand que dans l'organisation multiliste.
- Les temps de transfert des enregistrements sont améliorés. En effet, d'une part le parcours d'une liste partielle se fait à l'intérieur d'une cellule, d'où une réduction des temps de positionnement du mécanisme d'accès, d'autre part, les cellules peuvent être rangées sur des unités de mémoire à accès sélectif différents, permettant un accès simultané à ces cellules.

I.3 MODELE FONCTIONNEL

Ce modèle permet une description fonctionnelle multiniveau des SGBD, facilitant la simulation complète des systèmes. La modélisation est fondée essentiellement sur les RdP, elle se compose de trois parties, la partie opérative du système est représentée par un graphe de données, la partie contrôle par un graphe appelé graphe de commande et d'une interprétation [VALE 76].

I.3.1. LE GRAPHE DE DONNEES

Ce graphe est une représentation abstraite d'une partie du matériel gérée par le SGBD, il décrit le support physique et la répartition du dictionnaire et des fichiers de la BD sur ce support. Comme il illustre les chemins d'accès à parcourir pour récupérer l'ensemble des enregistrements répondant à une requête quelconque il est formé de :

- cellules de stockages représentées par des rectangles. D'après le niveau de description fonctionnelle du SGBD. Ces cellules peuvent être des cylindres, des pistes, un ensemble d'éléments ayant une signification logique, par exemple les attributs de la BD, des mots clés d'un attribut,...

A chaque cellule de stockage, est associée une variable qui représente la taille de cette cellule en des unités qui seront précisées pour chaque niveau du graphe. La taille marquée entre parenthèse près de l'identificateur de la cellule.

- Un ensemble fini d'opérateurs représenté par des cercles. Dans notre cas, ce ne sont que des opérations d'accès aux cellules de stockage.
- Des prédicats qui sont des opérateurs particuliers s'exécutant normalement à la suite de l'exécution des opérateurs, leur valeur de sortie, "vraie" ou "faux" devient disponible pour le graphe de commande. Les prédicats sont représentés par des losanges.
- Des cellules d'E/S; ce sont des cellules de stockages particulières pour la communication directe avec le monde extérieur. Elles sont représentées par un rectangle avec une flèche entrante ou sortante.

Les cellules de stockage sont réparties dans le graphe de données en quatre niveaux essentiels. Chaque niveau représente des informations stockées sur le support physique et qui ont une seule signification logique. Par exemple, le niveau des attributs, le niveau des mots clés, le niveau des listes d'accès et le niveau des enregistrements. Au niveau le plus haut se situent les attributs et au niveau le plus bas se situent les enregistrements ou les blocs d'enregistrements.

D'après le nombre d'attributs, de mots clés ou de listes d'accès, ces niveaux sont découpés sur le support physique en un ou plusieurs sous niveaux; ce découpage est représenté sur le graphe de données. La recherche pour récupérer la réponse à une requête commence toujours au niveau le plus haut du graphe de données et s'achemine dans le graphe, pour rétrécir le plus possible les enregistrements à accéder pour constituer la réponse.

I.3.2. LE GRAPHE DE COMMANDE (Fig.I.4)

Le graphe de commande est un Réseau de Petri Temporisé (RdPT), c'est-à-dire un graphe biparti; les deux classes de noeuds étant appelées usuellement places et transitions (Cf. annexe II) :

- A chaque place Pi (représentée par un cercle) est associé un ensemble de doublets (p_i, τ_i) ou p_i est un opérateur du graphe de données et τ_i le temps d'exécution associé à cet opérateur.
- A chaque transition t_{ij} (représentée par une barre) est associée une condition $\psi(t_{ij})$ qui est une fonction logique des prédicats du graphe de données.

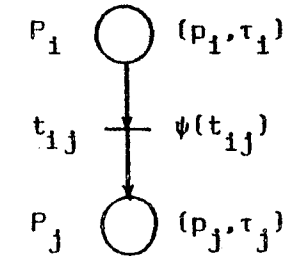


Figure I.4.
Graphe de commnande

Comme les opérateurs du graphe de données ne sont que l'accès et le transfert d'une partie des données en mémoire centrale, le temps de leur exécution dépend de la taille et de la répartition des données sur le support physique. Essentiellement, le temps d'accès est fonction de sa nature, s'il est aléatoire il prendra un temps noté τ_a, s'il est séquentiel il prendra un temps noté τ_s. Des expressions plus précises de ce temps seront données au chapitre II. Le transfert s'effectue dans un temps τ_t.

- L'opérateur "id" sera associé à toute place du RdPT pour laquelle il ne correspond aucune action dans le graphe de données.
- Pour toute transition dont la mise à feu ne dépend que de sa validation, la condition associée est omise.

Règles d'évolution dans le graphe de commande

Les règles d'évolution sont les suivantes :

- La mise à feu d'une transition validée t_{ij} n'est autorisée que si la condition $\psi(t_{ij})$ est vérifiée.
- L'arrivée d'une marque dans une place P_i déclenche l'exécution des opérateurs associés dans le graphe de données. Cette marque reste indisponible durant l'intervalle de temps entre l'instant τ_o de son arrivée à la place et l'instant $\tau_o+\tau_i$, puis devient disponible.

I.3.3. L'INTERPRETATION

L'interprétation donne un sens physique aux sommets du graphe de données, aux opérateurs, aux prédicats et aux cellules de stockages.

- L'opération réalisée par un opérateur donné.
- L'expression logique déterminant un prédicat donné.
- Les valeurs initiales des cellules de stockages.

I.3.4. EXEMPLE

Prenons comme exemple, le fichier des employés d'une entreprise. L'ensemble des attributs de la BD : {Code du travail, Salaire, Code de l'adresse}.

L'ensemble des mots clés de chaque attribut :

$V(\text{Code du travail}) = \{T_1, T_2, \ldots T_{10}\}$

$V(\text{Salaire}) = \{S_1, S_2, \ldots S_{10}\}$

$V(\text{Adresse}) = \{A_1, A_2, \ldots A_{15}\}$

Considérons les enregistrements de la BD :

r_1 = {Code du travail = T_4, Salaire = S_9, Adresse = A_2, A_4}

r_2 = {Code du travail = T_2, Salaire = S_{10}, Adresse = S_s}.

.

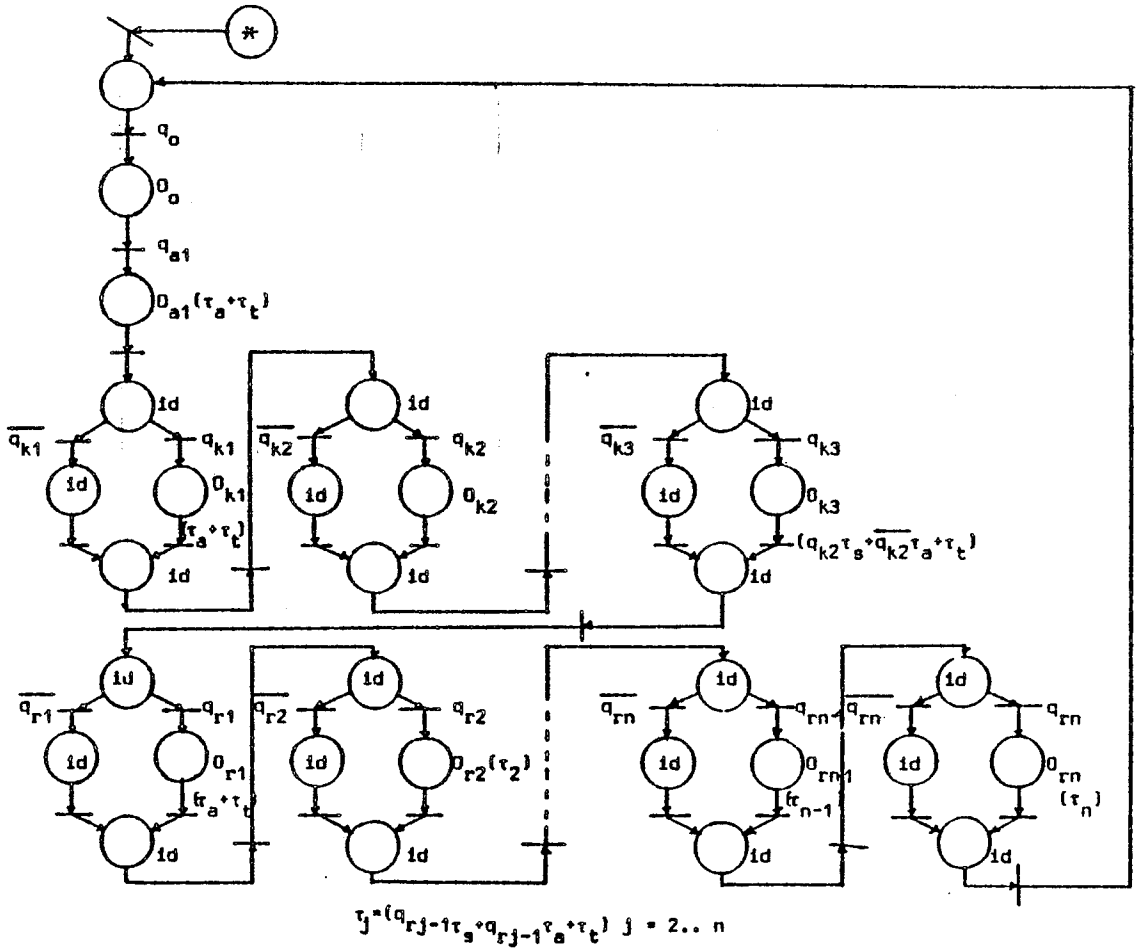
.

La modélisation du système gérant cette BD est illustrée dans la figure I.5. L'organisation physique adoptée est l'organisation multi-liste.

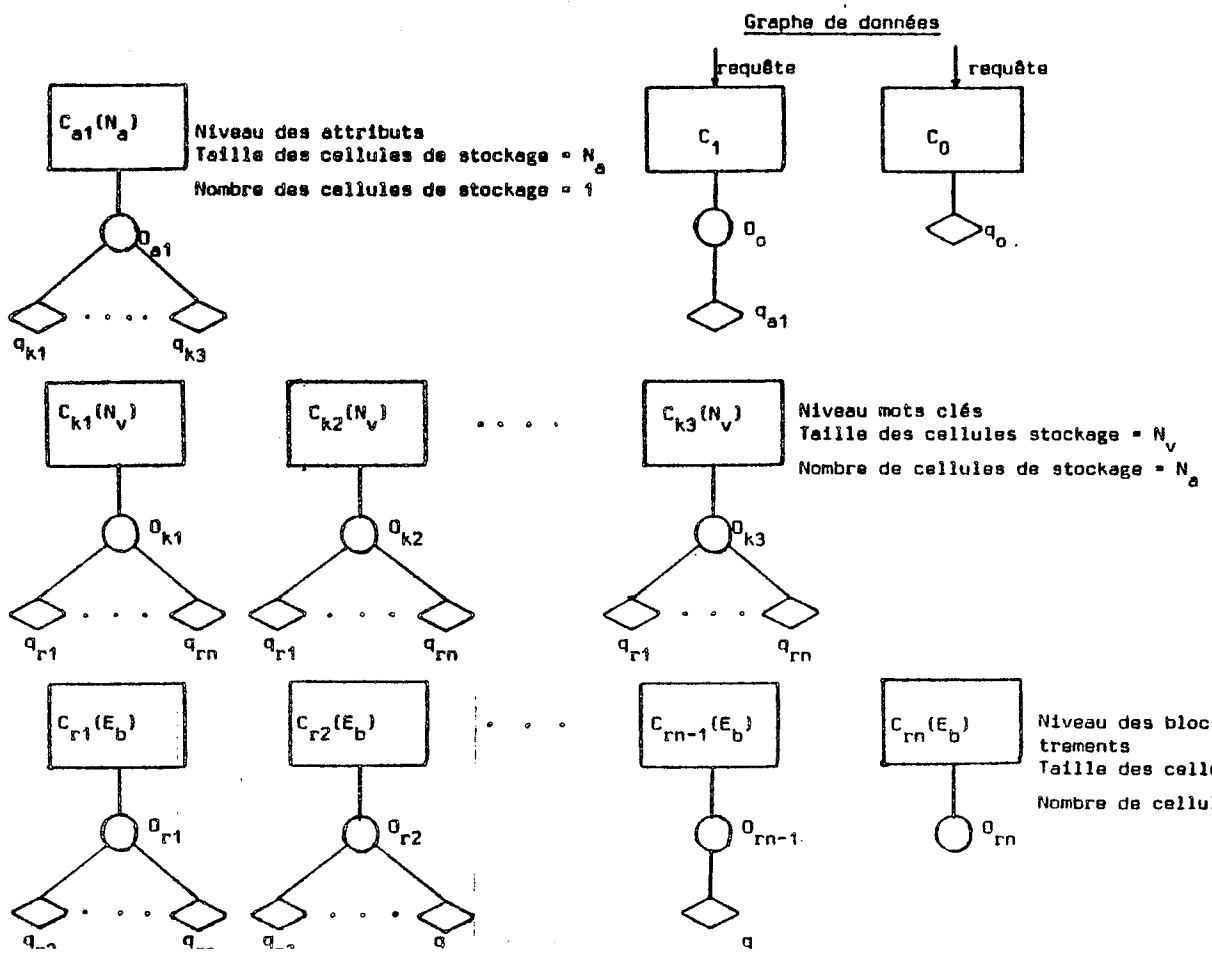
Interprétation

- L'arrivée d'une requête est l'ordre de départ au SGBD, q_o est mis à 1.
- Stockage de la requête dans C_1.
- O_o : décodage et tri des attributs et mots clés de la requête, q_{a1} est mis à 1.

Figure I.5 : Représentation graphique de l'organisation multiliste



Graphe de commande



- C_{a1} est la partie du support physique contenant les attributs de la BD, chaque entrée de cette cellule contient un attribut et un pointeur vers la zone des mots clés de cet attribut et sa longueur.

- O_{a1} : accès et lecture de C_{a1}.

Comparaison avec les attributs de la requête pour mettre à 1 les q_{ki} correspondants.

Le temps d'exécution de cette opération est $(\tau_a+\tau_t)$.

- C_{k1}; C_{k2} et C_{k3} contiennent les mots clés des trois attributs de la BD, et des pointeurs vers les listes d'enregistrements contenant les mots clés et leur longueur.

- O_{ki}, i = 1 ... Na, où Na est le nombre d'attribut de la BD. C'est l'accès au zone des mots clés correspondant puis la comparaison avec les mots clés de la requête et la mise à 1 des q_{rj}, pour initialiser la lecture des zones contenant le premier enregistrement des listes d'enregistrements répondant à la requête.
  Le temps d'exécution de l'opérateur O_{ki} est $(q_{ki-1}\ \tau_s+\bar{q}_{ki-1}\ \tau_a+\tau_t)$.
  Si le bloc précédent C_{ki-1} est lu, un accès séquentiel est effectué pour aboutir au bloc C_{ki}, sinon un accès aléatoire est nécessaire. Le temps de transfert est toujours τ_t.

- C_{ri} = i = 1,2,.. Be où Be est le nombre de blocs contenant les enregistrements. Ces cellules sont des blocs d'enregistrements qui peuvent être accéder par une seule opération d'E/S. Chaque cellule contient un nombre d'enregistrement dépendant de la taille de la cellule et de l'enregistrement. Chaque enregistrement contient un nombre de pointeurs suffisant pour son chaînage dans les differentes listes auxquelles il peut appartenir.

- O_{ri} i = 1,2,... B_e.

C'est l'accès aux blocs d'enregistrements, et la mise à 1 des q_{rk} k > i, pour permettre l'accès aux blocs contenant la suite des listes. Le temps d'exécution $=(q_{rj-1}\ \tau_s + \bar{q}_{rj-1}\ \tau_a+\tau_t)$

La succession d'exécution de ces opérateurs est clarifiée dans le graphe de commande.

I.3.5. MODELISATION DES LISTES INVERSEES

Les graphes de données et de commande de cette organisation sont montrés dans la figure I.6., l'interprétation correspondante suit.

Interprétation

q_o= (C_o=1?), arrivée d'une requête ?

O_o : décodage et tri des attributs et mots clés de la requête.

O_{a1} : l'accès et la recherche dans C_{a1} des attributs de la requête et la mise à 1 des prédicats correspondants.

q_{ki} (i=1,2..., Na) = 1 pour les attributs spécifiés dans la requête.

O_{ki} (i=1,2..., Na) : accès et recherche dans les zones de mots clés des attributs de la requête et la mise à 1 des prédicats correspondants.

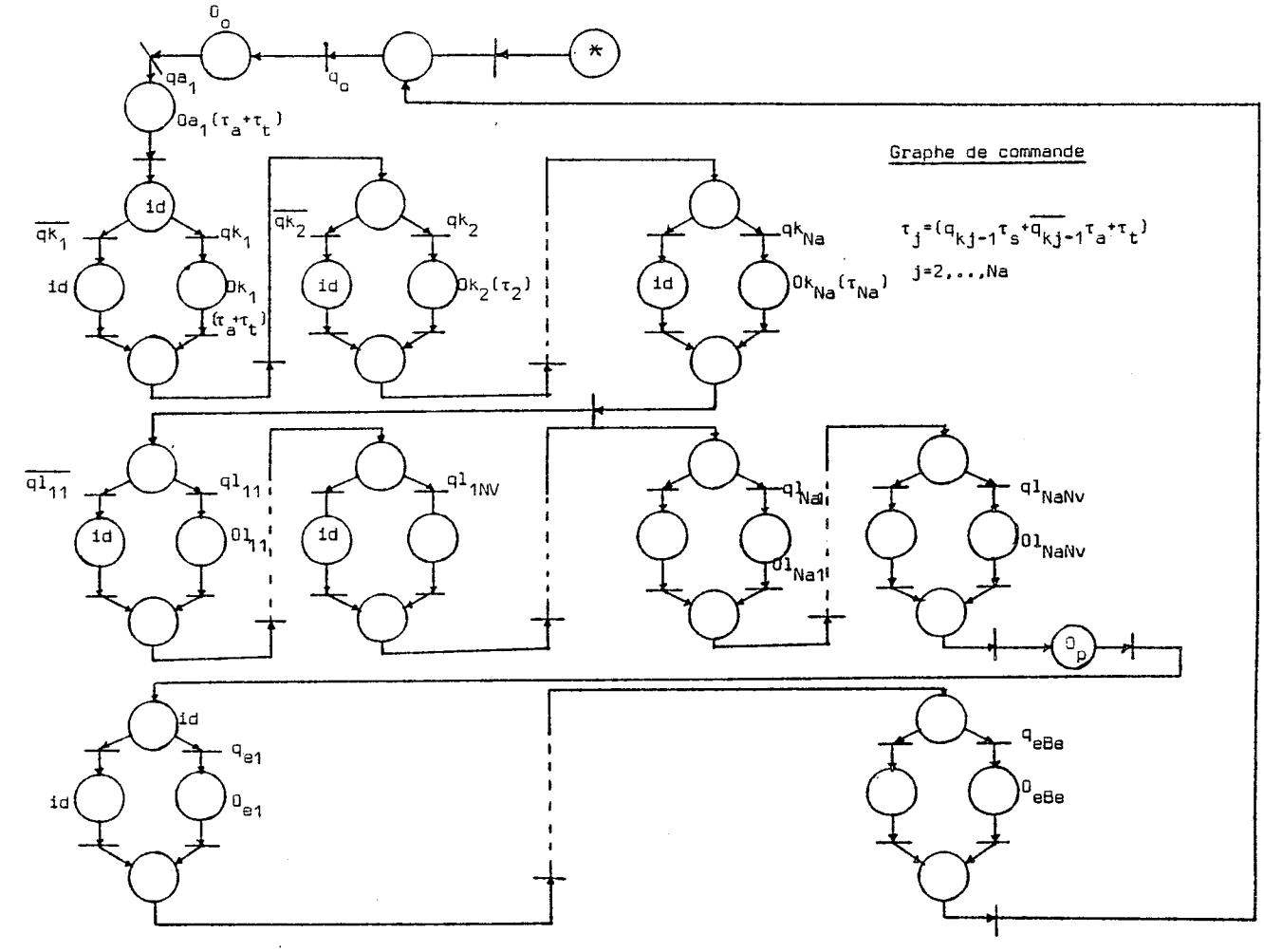
ql_{ij}[(i=1,..., Na) et (j=(1.. Nv)] = 1 pour le $j^{ième}$ mot clé du $i^{ème}$ attribut de la BD.

Ol_{ij} (a_i i=1.. Na et j = 1... Nv) accès et lecture des listes d'accès et communication des adresses des enregistrements qui existent dans ces listes au système.

O_p : Intersection, fusion et tri des adresses des enregistrements pour restreindre l'accès au niveau des enregistrements à ceux qui répondent uniquement à la requête, et la mise à 1 des prédicats correspondant à leur bloc.

q_{ei} (i=1...B_e) : = 1 pour les blocs contenant les enregistrements répondant à la requête.

O_{ei} (i=1... B_e) accès aux blocs et extraction de la réponse.



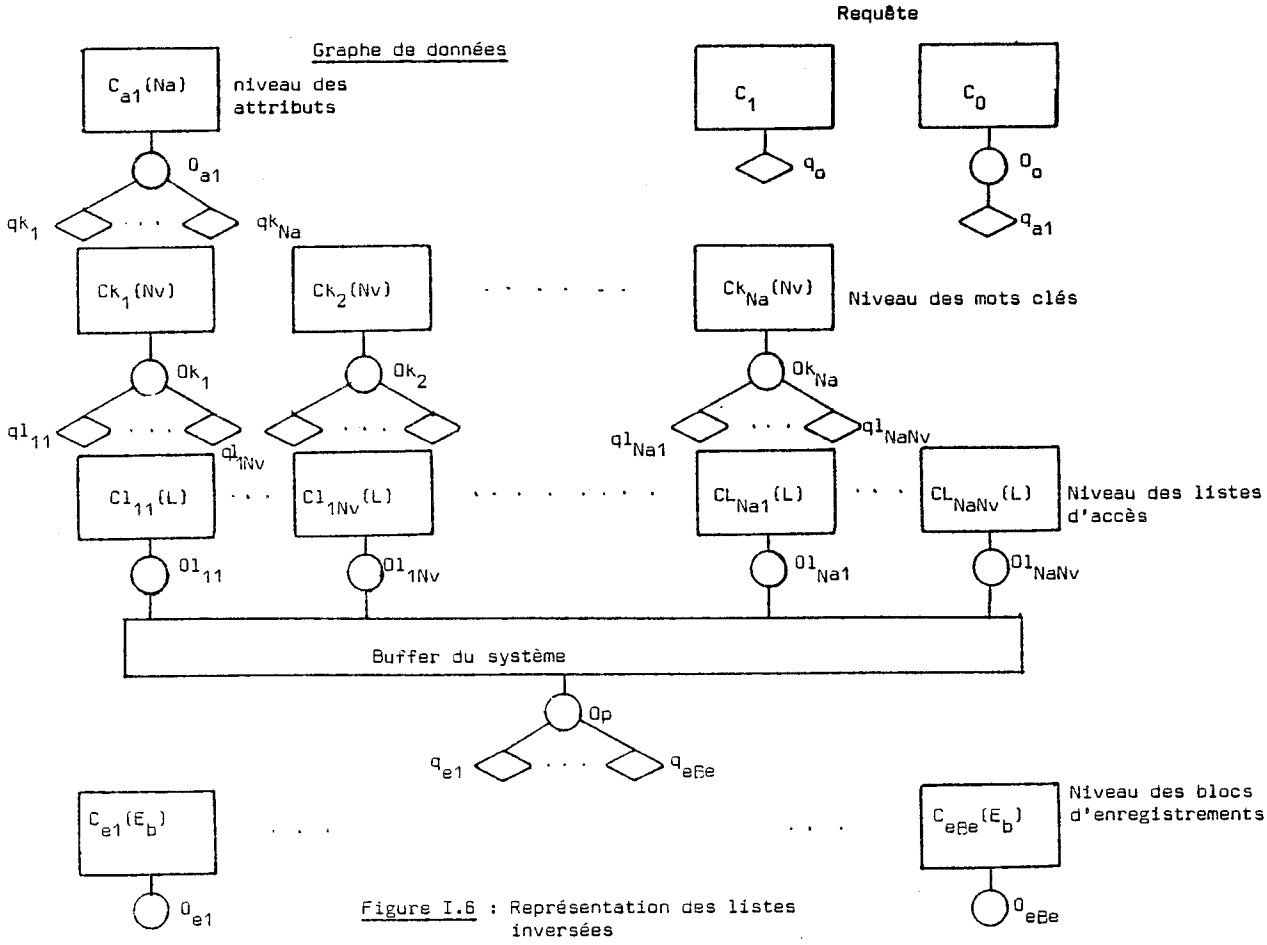


Figure I.6 : Représentation des listes inversées

I.3.6. MODELISATION DES LISTES CELLULAIRES

Le graphe de données de cette organisation est montré dans la figure I.7. Le graphe de commande est identique à celui des listes inversées.

L'interprétation est identique à celle des listes inversées jusqu'à la lecture des listes d'accès, ces listes contiennent les adresses des premiers enregistrements des listes cellulaires qui peuvent contenir la réponse.

O_p : Intersection, fusion et tri des cylindres à accéder au niveau des enregistrements et la mise à 1 des prédicats correspondants.

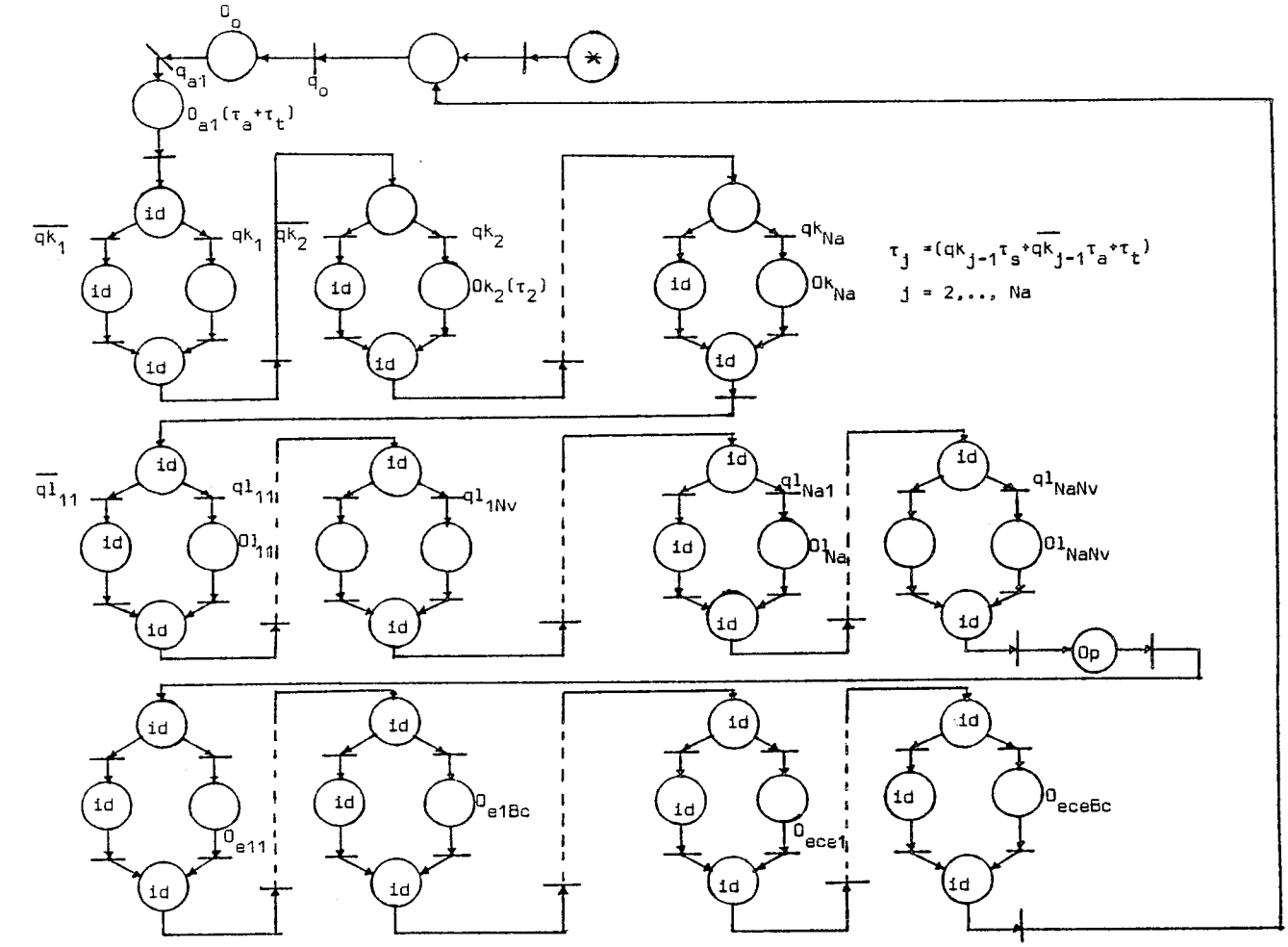
q_{eij} (i=1... Ce, et j=1... Bc) : = 1 pour le cylindre contenant une partie de la liste des enregistrements de la réponse. Ce est le nombre de cylindres contenant les enregistrements. Bc est le nombre de bloc par cylindre.

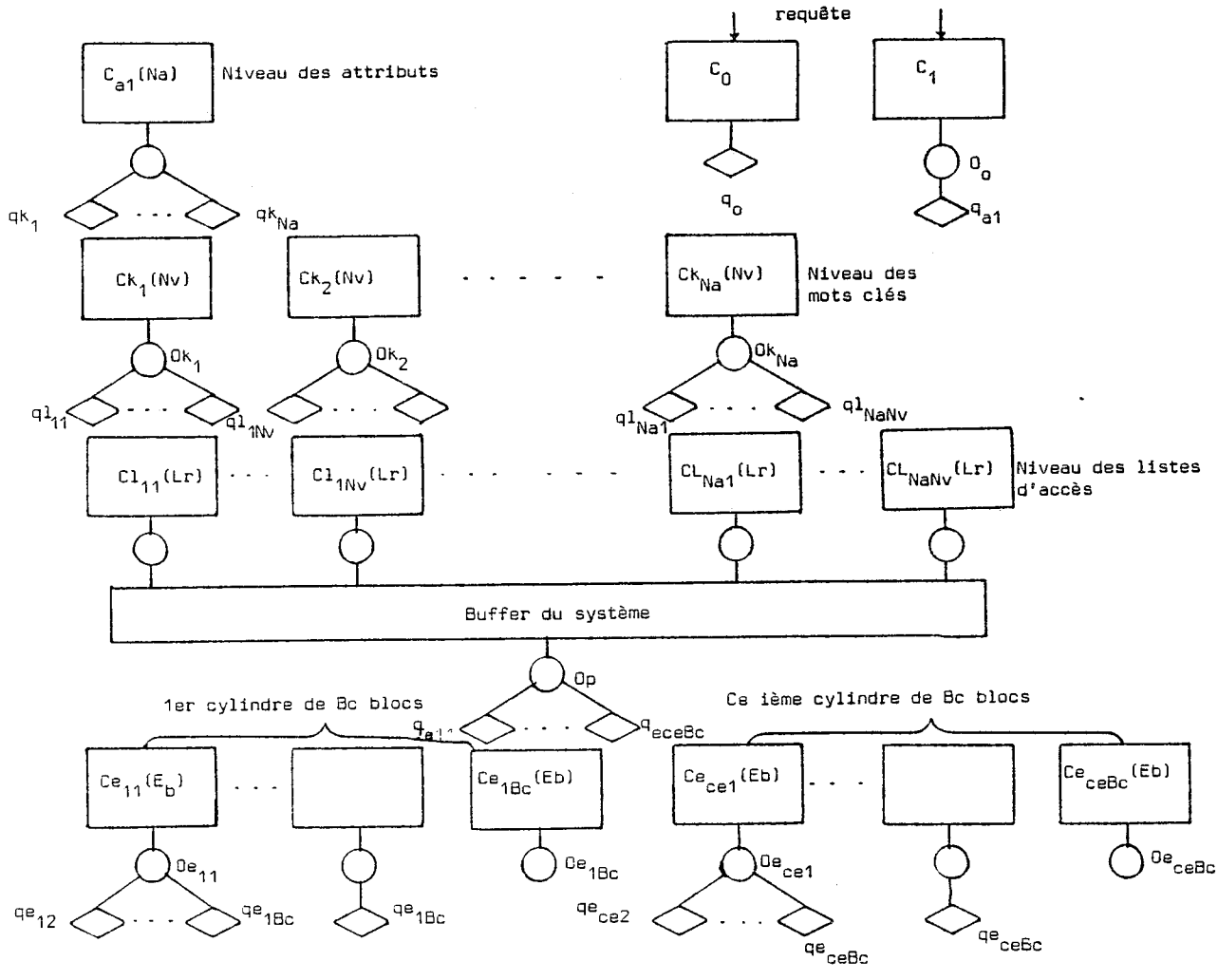
O_{eij} = mouvement du mécanisme d'accès jusqu'au cylindre numéro i, lecture du premier bloc contenant le premier enregistrement de la liste, et la mise à 1 des prédicats pour le parcours de la liste partielle dans ce cylindre.

Ce modèle permet de simuler des SGBD pour différentes organisations physiques et complexité des requêtes.

Le temps de réponse aux requêtes peut être estimé par simulation. Cette simulation aide le concepteur des SGBD en cours de développement au choix de l'organisation physique qui correspond le mieux à la répartition de la complexité des requêtes des utilisateurs du système.

Ce modèle est utile aussi pour la représentation des grandes BD dont les fichiers sont stockés sur plusieurs supports physiques. Ces fichiers peuvent avoir différentes organisations physiques. L'accès parallèle à ces supports physiques peut être représenté par le modèle. En particulier, la détection de conflit peut être faite sur ce type de modèle. Une simulation de ces systèmes donne en outre le temps de réponse à des requêtes interrogeant les fichiers de la BD. Ceci est illustré dans l'exemple suivant.


O_o
q_{a1}
q_o
$O_{a1}(\tau_a+\tau_t)$
id
$\overline{qk_1}$
qk_1
$\overline{qk_2}$
qk_2
$Ok_2(\tau_2)$
qk_{Na}
Ok_{Na}
$\tau_j = (qk_{j-1}\tau_s + \overline{qk}_{j-1}\tau_a + \tau_t)$
$j = 2,..., Na$
$\overline{ql}_{11}$
ql_{11}
Ol_{11}
ql_{1Nv}
ql_{Na1}
Ol_{Na1}
ql_{NaNv}
Ol_{NaNv}
Op
O_{e11}
O_{e1Bc}
O_{ece1}
O_{eceBc}


requête
$C_{a1}(Na)$
Niveau des attributs
C_0
C_1
O_o
q_o
q_{a1}
qk_1
qk_{Na}
$Ck_1(Nv)$
$Ck_2(Nv)$
$Ck_{Na}(Nv)$
Niveau des mots clés
Ok_1
Ok_2
Ok_{Na}
ql_{11}
ql_{1Nv}
ql_{Na1}
ql_{NaNv}
$Cl_{11}(Lr)$
$Cl_{1Nv}(Lr)$
$CL_{Na1}(Lr)$
$CL_{NaNv}(Lr)$
Niveau des listes d'accès
Buffer du système
Op
1er cylindre de Bc blocs
Ce ième cylindre de Bc blocs
q_{eceBc}
$Ce_{11}(E_b)$
$Ce_{1Bc}(Eb)$
$Ce_{ce1}(Eb)$
$Ce_{ceBc}(Eb)$
Oe_{11}
Oe_{1Bc}
Oe_{ce1}
Oe_{ceBc}
qe_{12}
qe_{1Bc}
qe_{ce2}
qe_{ceBc}

I.3.7. EXEMPLE

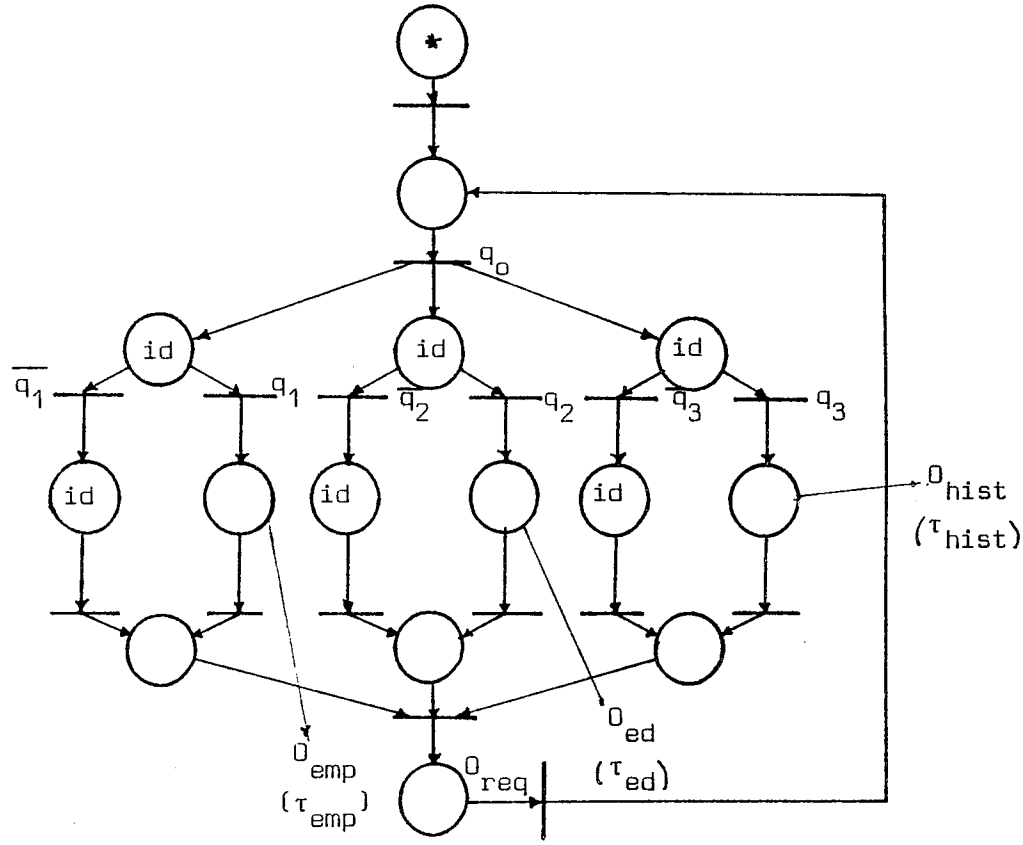
Soit une grande BD contenant trois fichiers, le premier contenant les employés, adresse, date de naissance..., le second la formation des employés, les cours et stages suivis,... le troisième, l'histoire de ces employés dans le travail, leurs expériences, leur salaire,...

Ces fichiers sont stockés séparément sur trois supports physiques que l'on peut accéder parallèlement.

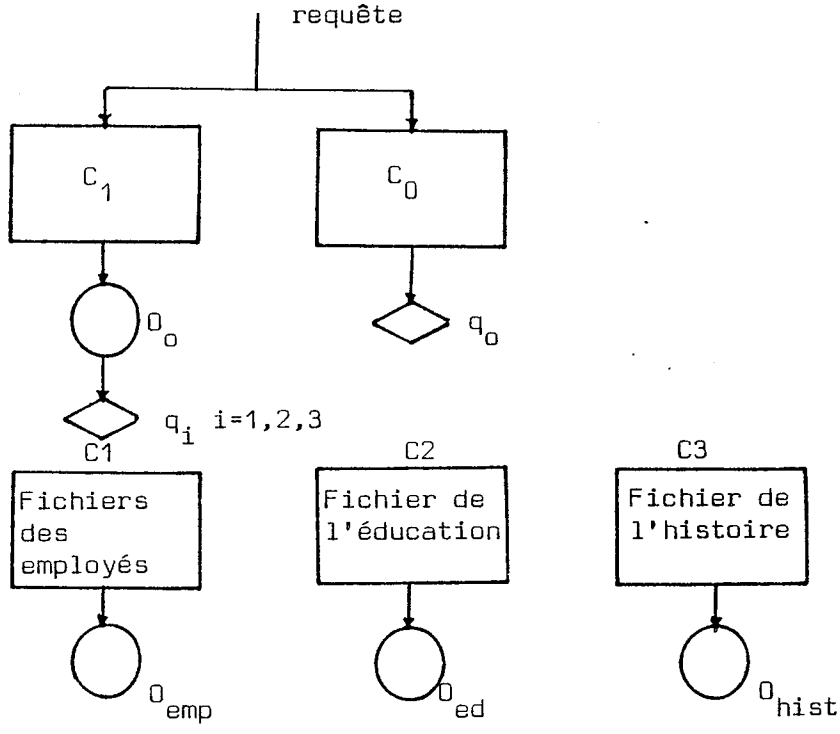
Une simple requête cherchant un employé ayant une certaine formation et une certaine expérience nécessitera un accès à ces trois fichiers.

Le temps de réponse pourra être obtenu par une simulation du modèle présenté dans la figure I.8.

Les opérateurs du graphe de commande décrivent à un niveau macroscopique l'accès parallèle, chacun représente la recherche dans le fichier correspondant, cette recherche dépendant de l'organisation physique du fichier est représentée à un niveau plus détaillé dans les figures I.5, I.6 et I.7. Les cellules de stockage du graphe de données de la figure I.8 sont les fichiers eux-mêmes.



Graphe de Commande



Graphe de donnée

Figure I.8

Interprétation

q_0 : (Co=1?) arrivée d'une requête?

Co : décodage de la requête et précision des fichiers à accéder, et les différents attributs et mots clés de chaque fichier.

q_i=1 (i = 1,2,3) pour les fichiers à accéder.

O_{emp} : accès au fichier des employés et extraction des informations nécessaires, ceci est exécuté dans un temps τ_{emp}. De même pour O_{ed} et O_{hist}, les temps τ_{emp}, τ_{ed} et τ_{hist} peuvent être obtenus des figures détaillées I.5, I.6 et I.7.

O_{req} : la constitution de la réponse à la requête.

Une simulation du SGBD à l'aide de ce modèle permet de choisir l'organisation physique la plus performante pour chacun des trois fichiers. Le critère de comparaison sela le temps de réponse à des requêtes de différentes complexités.

Une simulation de réseaux de Petri Temporisés a été conçue dans l'équipe "Conception et Sécurité des Systèmes" de l'ENSIMAG et permet la validation fonctionnelle, l'évaluation des performances à ce niveau de modélisation [ZACH 77] et [ALVA 78], [MOAL 76-B].

En fait, ce type de modélisation permet à la fois la détection d'erreur de conception, ainsi que la mise en place de test fonctionnel contre les pannes matérielles [BELL 77]. En particulier, une étude de la propagation des erreurs dans les systèmes distribués (phénomène de contamination) peut être appliquée aux bases de données réparties [BELL 78].

CHAPITRE II

EVALUATION DE PERFORMANCES DES SGBD

-o-o-

II.1 INTRODUCTION

La performance des SGBD est mesurée essentiellement par le temps moyen de réponse aux requêtes des utilisateurs du système. Ce temps est divisé en deux parties, la première est le temps passé dans les différentes files d'attente du système et la seconde est le temps d'accès et de transfert des informations.

L'attente dans les différentes files dépend de la charge du système et des stratégies du système pour le service des demandes, FCFS,...

La figure II.1 montre le graphe de temps du disque à tête mobile [COFF 73], [IBM 74], [KNUT 73] et [LEFK 69].

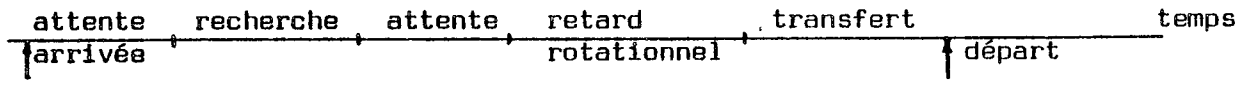


Figure II.1. Graphe de temps du disque à tête mobile

Le temps de recherche pris pour le positionnement du mécanisme d'accès sur le cylindre contenant le bloc à lire est fonction du nombre de cylindres traversés pour atteindre le cylindre recherché, cette fonction n'est pas linéaire.

La sélection de la tête de lecture/écriture est effectuée par des circuits électroniques et prend un temps négligeable par rapport aux autres facteurs.

Le retard rotationnel est le temps passé pour que le début de l'enregistrement ou le bloc à accéder aboutisse devant la tête de lecture/écriture correspondante. Il a comme moyenne le temps d'une demi rotation du disque.

Le transfert des données entre le disque et la mémoire centrale dépend de la taille de ces données. Le débit de transfert est fonction de la vitesse de rotation du disque et de la densité d'enregistrement sur ce disque. Une rotation complète est nécessaire pour le transfert des informations occupant une piste d'un disque.

Les modélisations étudiées ne prenant le compte de tous les détails de l'environnement, les valeurs moyennes des différentes composantes des temps d'accès seront considérées.

Dans les paragraphes (II.2) à (II.6) de ce chapitre, on résumera les différents travaux et modèles apparus pour l'évaluation des performances des SGBD. Ces modèles peuvent être classés en deux groupes distincts. Le premier évalue le temps moyen de réponse aux requêtes des SGBD en fonction de la complexité de ces requêtes et de l'organisation physique des données. Ce groupe n'inclut pas dans son évaluation l'attente dans les files d'attente des supports physiques (II.2, II.3, II.4). Le deuxième évalue le temps moyen de réponse pour les demandes d'accès à un enregistrement ou à un bloc de données dans un SGBD. L'attente dans les files des supports physiques est considérée (II.5, II.6).

Dans le paragraphe II.7, on propose une amélioration du modèle de YAO [YAO 77.B]. Des expressions plus précises que celles de YAO pour l'évaluation du coût d'accès aux fichiers des BD seront données pour les organisations listes inversées, listes cellulaires et multilistes. Plus de détails de l'implémentation des SGBD seront considérés.

II.2 MODELE DE CARDENAS [CARD 73] et [CARD 75]

Une évaluation de la quantité de stockage, et du temps moyen d'accès aux informations de l'organisation listes inversées est présentée dans [CARD 73] et [CARD 75]. Le dictionnaire et les listes d'accès sont vues comme étant une autre BD qui peut être aussi inversée.

Le modèle prend en considération, la structure de stockage, la recherche dans le dictionnaire et les listes d'accès, ainsi que des données statistiques sur le contenu de la BD, la complexité des requêtes interrogeant la BD, et les paramètres macroscopiques du support physique de la BD.

La mémoire secondaire est divisée en blocs ou pages, qui représentent l'unité de données qui peut être transférée entre cette mémoire et la mémoire centrale à la suite d'une seule opération d'E/S. Ce bloc ou page peut être une piste d'un disque.

La figure II.2 montre l'organisation physique de la BD. Le dictionnaire est organisé en deux niveaux, le niveau 0 est un index au niveau 1, chacune de ses entrées correspond à la première propriété, ou mot-clé, de chaque bloc du niveau 1, elle contient le mot clé et un pointeur sur son bloc.

Les paramètres et les symboles utilisés sont groupés dans le tableau II.1 en trois groupes : les paramètres de la BD, du support physique et des requêtes.

On essaiera d'unifier les notations tout le long de ce chapitre pour faciliter la comparaison entre les différents modèles.

Un taux d'occupation F est défini pour chaque niveau, il représente le rapport entre l'espace initialement utilisé lors d'une réorganisation du fichier sur l'espace total alloué au fichier.

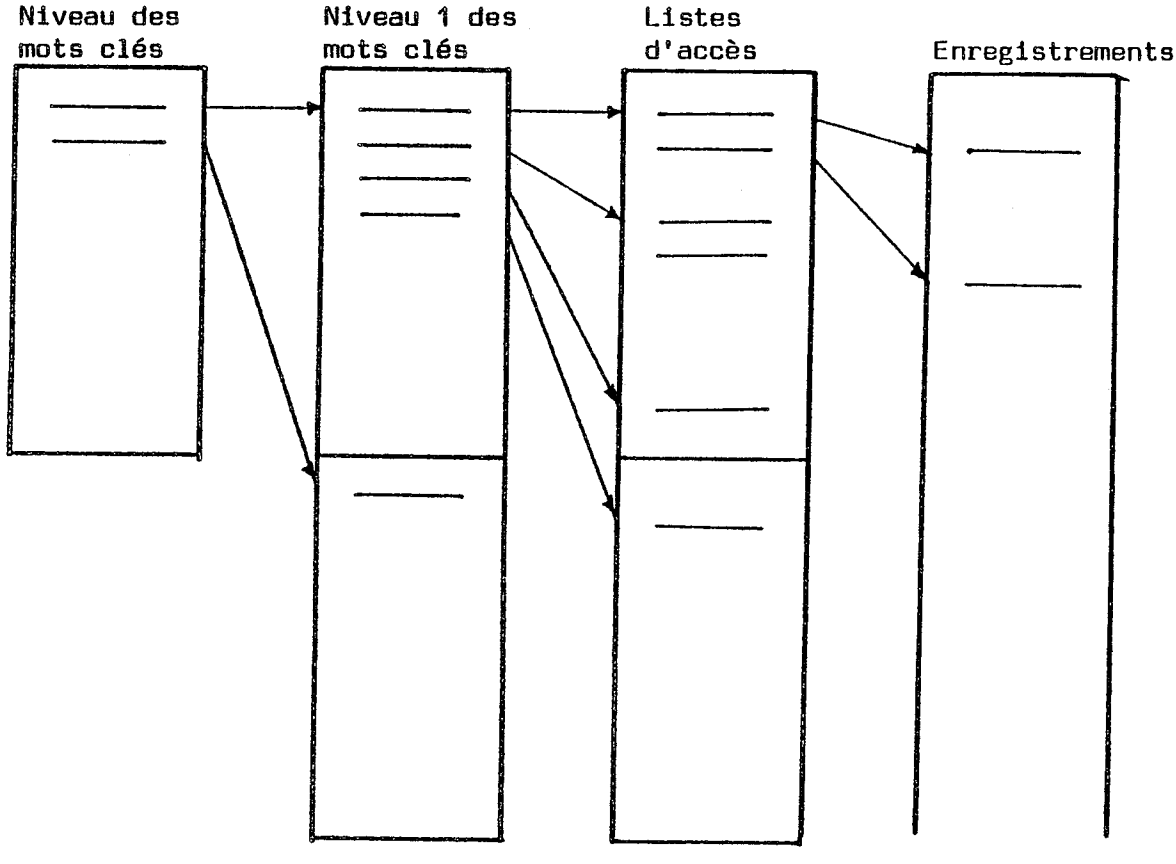


Fig.II.2. Organisation listes inversées

Statistiques de la BD

N : nombre d'enregistrements de la BD

r : longueur moyenne des enregistrements

K_1 : longueur moyenne des mots clés

$(N_v)i$: nombre de mots clés pour l'ième attribut

N_a : nombre d'attributs inversés dans la BD.

Paramètres du support physique

b : taille d'un bloc en mots

T_τ : temps moyen d'accès à un bloc, page ou piste

T_c : temps moyen de comparaison d'une clé accédée

T_I : temps moyen de l'intersection de deux blocs des listes d'accès.

Caractéristiques des requêtes (Cf. Annexe 1)

ACI : nombre moyen de conditions élémentaires/disjonction

c : nombre moyen d'attribut/conjonction

d : nombre moyen de conjonctions dans la requête.

Paramètres estimés

F_k : taux d'occupation au niveau 1 des mots clés

F_1 : taux d'occupation des blocs des listes d'accès

F_e : taux d'occupation des blocs d'enregistrements

B_k : nombre de blocs occupés par le niveau 1 des mots clés

B_1 : nombre de blocs occupés par les listes d'accès

B_e : nombre de blocs occupés par les enregistrements

B_r : nombre moyen de blocs d'enregistrements accédé par requête.

Tableau II.1

	Condition élémentaire ACI = 1	Disjonction des conditions élémentaires ACI ≥ 2	Remarques
1. Lecture du niveau 0 des mots clés	T_T	T_T	Lecture d'un bloc
2. Recherche de l'index au niveau 1	$T_c \log_2 B_K$	$T_c \log_2 B_K$	Recherche binaire dans une table ordonnée de B_K entrées
3. Lecture du bloc contenant le mot clé recherché	T_T	T_T	Lecture d'un bloc
4. Recherche du pointeur à la liste d'accès	$T_c \log_2 \left\lfloor \frac{b}{K_L+2} \right\rfloor$	$T_c \log_2 \left\lfloor \frac{b}{K_L+2} \right\rfloor$	Recherche binaire dans une table de $\left\lfloor \frac{b}{K_L+2} \right\rfloor$ entrées
5. Répéter 4 q fois	-	-	q nombre de mots clés pour une condition élémentaire $q = \frac{1}{2} \sum_1^{N_a} \frac{(N_v)_i}{N_a}$
6. Répéter 4 et 5 pour chaque condition élémentaire	-	-	Les mots clés d'un même attribut se trouvent dans un même bloc
7. Lecture et fusion des listes d'accès	$(T_I+T_T) * \left\lceil \frac{L}{b} \right\rceil$	$(T_I+T_T)*ACI*\left\lceil \frac{L}{b} \right\rceil$	L : longueur moyenne des listes d'accès $L = N_e / \sum_1^{N_a} (N_v)_i * N_a$
8. Lecture des blocs contenant les données	$T_T * B_r$	$T_T * B_r$	$B_r = B_e(1-(1-\frac{1}{B_e})^K)$ et $K = L * ACI$

Tableau II.2

	Conjonction	Disjonction de conjonctions	Remarques
1. Lecture du niveau D des mots clés	T_T	T_T	
2. Recherche de l'index au niveau 1	$T_c \log_2 B_K$	$T_c \log_2 B_K$	
3. Lecture du bloc contenant le mot-clé recherché	T_T	T_T	
4. Recherche du pointeur à la liste d'accès	$T_c \log_2 \left\lfloor \frac{b}{K_L+2} \right\rfloor$	$T_c \log_2 \left\lfloor \frac{b}{K_L+2} \right\rfloor$	
5. Répéter 4 pour chaque mot-clé de la condition élémentaire			c : nombre d'attribut/ conjonction
6. Répéter 2 à 5 pour chaque attribut de la requête	$\Sigma_{i=1}^{c} ACI_i * \left\lceil \frac{L}{b} \right\rceil (T_I+T_T)$	$\Sigma_{i=1}^{c} (ACI)_i * d * \left\lceil \frac{L}{b} \right\rceil (T_I+T_T)$	d : nombre de conjonction dans la requête
7. Lecture, fusion et intersections des listes d'accès			
8. Lecture des blocs de données	$T_T * B_r$	$T_T * B_r$	$B_r = B_e(1-(1-\frac{1}{Be})^K)$ $K = (ACI)min * L$ $= \Sigma_{j=1}^{d} (ACImin)j * L$

Tableau II.3

La stratégie de la recherche des enregistrements, et par conséquence le calcul du temps moyen d'accès dépend de la complexité des requêtes (Cf. Annexe 1).

La dérivation du temps d'accès pour chaque type de requêtes est résumée dans les tableaux II.2 et II.3.

Cardena a fait beaucoup d'hypothèses qui empêchent l'application de son modèle sur les grands systèmes utilisant les listes inversées. Ses hypothèses sont :

1. L'organisation du dictionnaire en deux niveaux.
2. La taille du premier niveau est d'un bloc.
3. Les mots clés d'un même attribut sont stockés sur un même bloc, ce qui n'est pas évident pour toutes les BD.
4. L'expression pour le calcul du nombre de bloc à accéder pour récupérer la réponse est trop approximative si le nombre d'enregistrements par bloc est inférieur à 10 [YAO 77.A].

II.3 MODELE DE SILER [SILE 76]

Siler représente un modèle stochastique pour l'évaluation de trois organisations physiques, les listes inversées, les multilistes et les listes cellulaires, et des combinaisons hybrides de ces trois organisations. Il évalue le temps de réponse T, associé à une séquence de requêtes des utilisateurs pour une organisation physique et pour une BD donnée.

Le temps de réponse est une équation linéaire ayant comme variable le mouvement du mécanisme d'accès A à la rotation du disque R. Des coefficients c_1 et c_2 qui dépendent des paramètres du support physique de la BD, convertissent le mouvement en temps.

$$T = c_1 A + c_2 R \qquad (II.1)$$

Pour le disque IBM 2314 $c_1 = 0.075$ et $c_2 = 0.025$. Les requêtes prennent une forme conjonctive normale. Le nombre d'attributs énumérés dans la conjonction est défini comme la complexité, c de la requête.

Pour la simulation du système, chaque cylindre dans le modèle est représenté par la fréquence, $F_c(i, K_{vi})$, d'occurrence des enregistrements, ayant une propriété ou des mots-clés K_{vi} pour chaque attribut i dans ce cylindre.

Le modèle considère que les mots clés sont indépendants et estime que le nombre d'enregistrements par cylindre satisfaisant une requête peut être obtenu de l'équation II.2.

$$E_{rc} = E_c \prod_{i=1}^{c} (F_c(i,K_{vi})/E_c) \ldots \quad II.2$$

où c est le complexité de la requête;
E_c est le nombre d'enregistrements par cylindre.

L'accès à un cylindre nécessite un seul mouvement du mécanisme d'accès et le nombre de rotation du disque J(x) est calculé par une équation récurrente dans [SILE 76].

Pour accéder à un nombre x d'enregistrements d'un cylindre ayant B_c tête de lecture/écriture, i.e. B_c blocs, et E_b enregistrements par tête de lecture/écriture, i.e E_b enregistrements par bloc, J(x) est donné par :

$$J(x) = \frac{\sum_{i=\lfloor \frac{x}{E_b} \rfloor}^{\min(x,B_c)} C_i^{B_c} \; H(i/x,E_b)}{C_x^{E_b}} \qquad II.3$$

où

$$H(i/x,E_b) = C_x^{E_b} - \sum_{j=\lfloor \frac{x}{E_b} \rfloor}^{i-1} C_j^i \; H(j/x,E_b)$$

Pour une organisation listes inversées, le mouvement du mécanisme d'accès et la rotation du disque sont donnés par :

$$A_I = c + \sum_{cyl=1}^{Ce} (E_{rc} > 0) \qquad II.4a$$

$$R_I = \frac{1}{2} c + \sum_{i=1}^{c} \left[\left(\frac{1}{L_b}\right) \sum_{cyl=1}^{Ce} F_c(i,K_{vi}) \right] + \frac{1}{2} A_I + \sum_{cyl=1}^{C_e} J(E_{rc}) \qquad II.4b$$

C_e : nombre de cylindres contenant les enregistrements.

L_b : nombre de pointeurs dans les listes d'accès par bloc.

L'équation II.4a est directe, c accès sont nécessaires pour accéder aux listes d'accès, et un accès pour chaque cylindre contenant des enregistrements recherchés.

Les deux premiers termes dans l'équation II.4.b représentent la rotation pour l'accès aux listes d'accès, les deux autres expriment la rotation pour l'accès aux enregistrements.

Pour les multilistes :

$$A_T = \sum_{cyl=1}^{C_e} \{F_c(m,K_m) > 0\} \qquad \text{II.5a}$$

$$R_T = \frac{1}{2} A_T + \sum_{cyl=1}^{C_e} J(F_c(m,K_m)) \qquad \text{II.5b}$$

m est l'attribut ayant la liste la plus courte dans le cylindre considéré.

Pour les listes cellulaires

$$A_c = c + \sum_{cyl=1}^{C_e} \{(\min_i F_c(i,K_{vi})) > 0\} \qquad \text{II.6a}$$

$$R_c = \frac{1}{2} c + \sum_{i=1}^{c} \left[\left(\frac{1}{L_b}\right) \sum_{cyl=1}^{C_e} (F_c(i,K_{vi}) > 0) \right] + \frac{1}{2} A_c + \sum_{cyl=1}^{C_e} J(\min_i F_c(i,K_{vi})) \qquad \text{II.6b}$$

Des répartitions des requêtes et des mots clés de la BD sont considérées et le temps de réponse est calculé pour différentes complexités de requêtes. Les résultats de la simulation sont trouvés dans [SILE 76].

Siler évalue le temps d'accès aux listes d'accès et aux enregistrements en considérant que la recherche des mots clés et des attributs est identique pour les trois organisations.

II.4 MODELE DE YAO [YAO 77-B]

YAO propose un modèle général applicable pour l'analyse et l'évaluation des différentes organisations des BD systématiquement, il donne une expression générale pour le coût d'accès moyen à une BD.

YAO définit une requête comme étant une fonction booléenne qui unit des mots clés (attribut=valeur) ou des domaines (v_1< attribut< v_2) par des opérateurs ∩ et ∪ et elle est structurée dans une forme disjonctive normale [MEND 64].

La complexité des requêtes est représentée par un tuple (d,c,p,q) où d est le nombre moyen de conjonctions par requête, c le nombre moyen d'attributs dans chaque conjonction, p le nombre total d'attributs spécifiés, et q le nombre moyen de mots clés spécifiés pour chaque attribut.

L'accès aux enregistrements de la BD est représenté par une arborescence. Dans cet arbre, le niveau s est le niveau des attributs, t est le niveau des mots clés, r est le niveau des listes d'accès et n est celui des enregistrements.

La figure II.3 illustre l'arbre d'accès d'un fichier des employés définit par les attributs (code du travail, code de l'adresse, salaire).

Dans un arbre, l'ensemble filial est défini comme étant l'ensemble de noeuds ayant le même parent immédiat.

Chaque niveau de l'arbre est constitué des ensembles filiaux de ce niveau i.

Chacun des niveaux de l'arbre est complètement défini par quatre paramètres W_i, P_i, F_i et Q_i.

W_i : taille moyenne de l'ensemble filial à ce niveau

P_i : taux de débordement des noeuds à ce niveau

$P_i = \frac{\nu i}{\mu_i + \nu_i} = \frac{\nu i}{W_i}$. Dans un ensemble de W_i noeuds, μ_i noeuds sont stockés séquentiellement, les autres noeuds forment une chaîne de longueur ν_i.

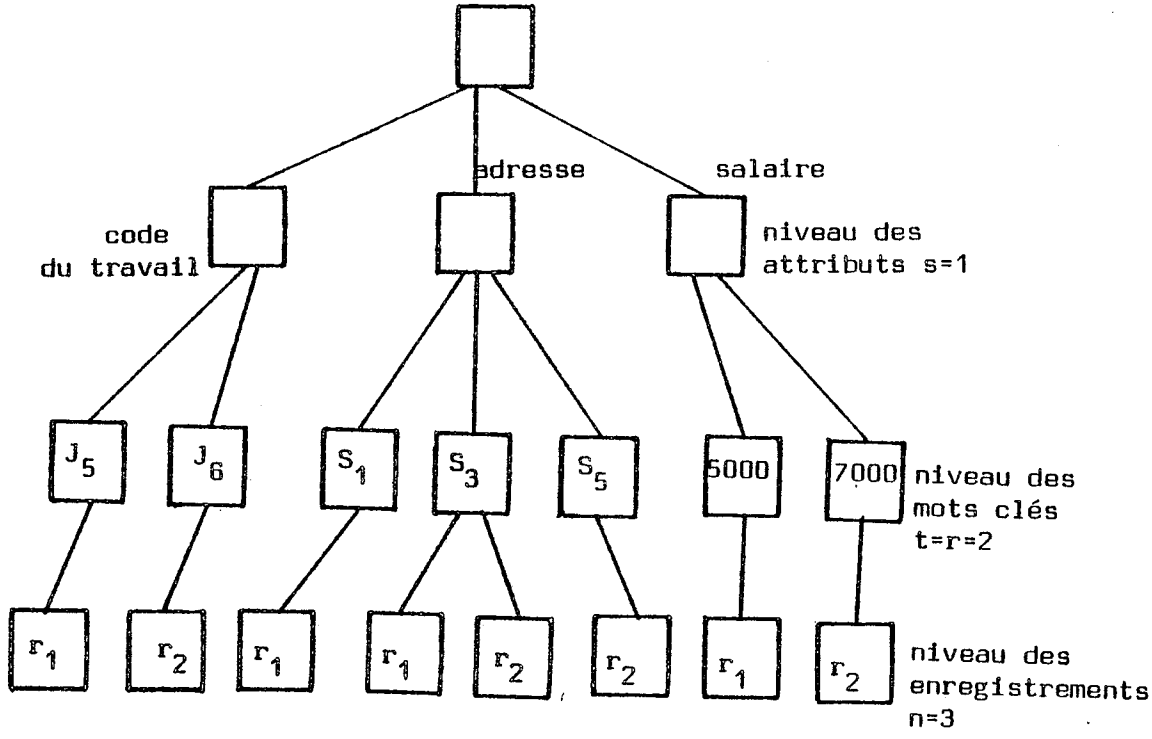


Fig.II.3. Modèle d'accès

F_i = taux d'occupation = $\mu_i/(\mu_i+w_i)$

où w_i est le nombre de noeuds libres permettant l'extension du niveau. La variable Q_i est le rapport de débordement permis dans un ensemble pendant son extension.

Equation générale du coût d'accès

A chaque piste et cylindre du support physique s'associent deux paramètres, le temps d'accès aléatoire et le temps d'accès séquentiel, t_r, t'_r et t_s, t'_s respectivement. Nous rappelons que pour accéder à un ensemble de pistes ou blocs séquentiels, les blocs et cylindres qui les contiennent sont accédés séquentiellement, sauf le premier cylindre et le premier bloc de chaque cylindre sont accédés aléatoirement. Dans le cylindre, les blocs consécutifs sont accédés sans retard.

Si le système de stockage contient N enregistrements, stockés séquentiellement sur B_e blocs et C_e cylindres, et si ces enregistrements sont également susceptibles d'être recherchés, le nombre moyen de blocs accédés séquentiellement pour la recherche de β enregistrements est

$$E(E_b,B,\beta) = 1+B'_e - \left[\frac{1}{C_\beta^N}\right] \sum_{i=1}^{B'_e} C_\beta^{iE_b} \qquad (II.7)$$

où E_b est le nombre d'enregistrements par bloc,

$B'_e = \left\lfloor \frac{N}{Eb} \right\rfloor$, et $B_e = \left\lceil \frac{N}{Eb} \right\rceil$ est le nombre de blocs contenant les enregistrements.

De même, le nombre de cylindres accédés séquentiellement pour la recherche des β enregistrements est :

$$E(B_c,N,\beta) = 1 + C'_e - \left[\frac{1}{C_\beta^N}\right] \sum_{i=1}^{C'_e} C_\beta^{iB_c} \qquad (II.8)$$

où B_c est le nombre de blocs par cylindre

$$C_e = \left\lceil \frac{N}{B_e\ B_c} \right\rceil \quad \text{et} \quad C'_e = \left\lfloor \frac{N}{B_e\ B_c} \right\rfloor$$

Le temps moyen d'accès pour la recherche des β enregistrements sera S'(N,β)

$$S'(N,\beta) = t_s+t_r+ [E(B_c,N,\beta)-1] (t'_s+t_r) + E(E_b,N,\beta).\ T \qquad (II.9)$$

Le temps moyen d'accès des N enregistrements séquentiellest est

$$S'(N,N) = S(N) = t_s+t_r + (C_e-1) (t'_s+t_r) + B_e.\ T \qquad (II.10)$$

Pour calculer le temps moyen d'accès à L_e enregistrements chaînés, si la chaîne est répartie aléatoirement sur une zone de B_e blocs ou C_e cylindres, le nombre moyen de blocs qui peuvent contenir au moins un enregistrement de la chaîne L_e est approximativement donné par :

$$G(B_e,L_e) = B_e\ [1 - (1- \frac{1}{B_e})^{L_e}] \qquad (II.11)$$

Le nombre moyen approximatif de cylindres qui contiennent au moins un enregistrement de la chaîne est donné par :

$$G(C_e,L_e) = C_e\ [1 - (1- \frac{1}{C_e})^{L_e}] \qquad (II.12)$$

Le temps moyen d'accès pris pour le balayage d'une liste ou chaîne de longueur L_e est :

$$V(L_e) = G(C_e, L_e)\ t_s + G(B_e, L_e)(t_r + T) \qquad \text{(II.13)}$$

Le temps moyen pour la recherche d'un enregistrement dans la liste

$$V'(L_e) = G(c_e, L_e/2)t_s + G(B_e, L_e/2)(t_r + T) \qquad \text{(II.14)}$$

Le temps de réponse à une requête de complexité (d,c,p,q) sera le critère de comparaison entre les différentes organisations physiques. Le temps de réponse est la somme du temps de recherche dans le dictionnaire, D et le temps I de l'interrogation du fichier.

Généralement, la recherche dans le dictionnaire du niveau 1 au niveau j de m clés prend un temps :

$$S(1,j,m) = \sum_{i=1}^{j} S_i(1,m)$$

où

$$S_i(1,m) = (1-P_i)G(\Pi_{k=1}^{i-1} W_k, m)S'((1-P_i)\ W_i, \beta_i) + P_i\ V'(G(\Pi_{k=1}^{i-1} W_k, m)\ P_i\ W_i) \qquad \text{(II.15)}$$

Le nombre d'ensemble filiaux au niveau i du dictionnaire est $\Pi_{k=1}^{i-1} W_k$, par notation $W_k = 1$ si $1 > i-1$.

Le nombre d'ensembles filiaux au niveau i qui peuvent contenir les m clés est $G(\Pi_{k=1}^{i-1} W_k, m)$, en considérant que les m clés sont triées avant l'exécution de la requête et qu'elles sont réparties aléatoirement sur les ensembles filiaux.

β_i est le nombre moyen de clés recherchées par ensemble filial, au niveau i du dictionnaire, $\beta_i = m/G(\Pi\ W_k, m)$.

La recherche des portions séquentielles des ensembles filiaux au niveau i prend un temps égal à :

$$G(\Pi_{k=1}^{i-1}\ W_k, m)\ S'((i-P_i)\ W_i, \beta_i)$$

Une zone de débordement est commune pour toutes les portions chaînées des ensembles filiaux au niveau i, d'où le temps d'accès à ces chaînes est obtenu à partir de :

$$V'(G(\Pi_{k=1}^{i-1} W_k, m)\ P_i\ W_i)$$

Le temps de recherche des m clés dans les ensembles filiaux du niveau i est la somme pondérée des temps d'accès séquentiel et chaîné, donnant ainsi l'équation (II.15).

Le temps de recherche D dans le dictionnaire est constitué de la recherche des p attributs de la requête du niveau 1 au niveau s, puis la recherche de q mots clés pour chaque attribut du niveau s+1 au niveau t de l'arbre d'accès, donc

$$D = S(1,s,p) + P\ S(s+1,t,q) \qquad \text{(II.16)}$$

L'interrogation du fichier commençant de m adresses initiales au niveau 1, jusqu'au j prend un temps $R(1,j,m)$ où :

$$R(1,j,m) = \Sigma_{i=1}^{j} R_i(1,m)$$

et
$$R_i(1,m) = (1-P_i)(\Pi_{k=1}^{i-1} W_k)\ m\ S((1-P_i)\ W_i) +$$

$$P_i\ V((\Pi_{k=1}^{i-1} W_k)\ m\ P_i\ W_i) \qquad \text{(II.17)}$$

La différence entre II.15 et II.17 est que chaque ensemble filial interrogé est lu complètement. Une analyse similaire peut aider à la déduction de II.17.

Si la requête contient pq mot clé, le temps nécessaire pour l'interrogation des listes d'accès est $R(t+1,r,pq)$.

Le calcul du nombre de blocs et cylindres nécessaires pour récupérer l'ensemble des enregistrements répondant à la requête est un problème difficile.

YAO fait une approximation simplifiant le calcul de ce nombre qu'il appele "nombre de blocs essentiels".

La ième conjonction de la requête est constituée de c_i attributs. Le nombre de mots clés spécifiés pour les attributs sont q_{i1}, q_{i2},... q_{ici}. La longueur moyenne de la liste d'accès pour chaque mot clé est $y_r = \Pi_{k=t+1}^{r} W_k$.

Les longueurs des listes d'accès pour les C_i attributs sont :

$q_{i1}\ y_r$, $q_{i2}\ y_r$,... $q_{ici}\ y_r$.

Le nombre de blocs essentiels pour la ième conjonction est inférieur à la liste d'accès la plus courte, $\min_j \{q_{ij}\}\ y_r$.

En prenant la moyenne, le nombre de blocs essentiels pour chaque conjonction est limité à : $\underset{i}{\text{moyenne}} \{\min_j(q_{ij})\ y_r\} = \hat{q}y_r$.

En considérant que les mots clés sont répartis uniformément, les mots clés de chaque attribut qui n'ont pas la liste la plus courte, la réduisent par un facteur q/N_v. D'où :

$$\varepsilon = \left\lceil \hat{q}y_r\ (q/N_v)^{c-1} \right\rceil d \qquad \text{(II.18)}$$

Le temps d'interrogation du fichier est alors :

$$I = R(t+1,r,pq) + R(r+1,n,\varepsilon) \qquad \text{(II.19)}$$

Le temps moyen de réponse à une requête de complexité (d,c,p,q) est G

$$G = S(1,s,p) + p\,S(s+1,t,q) + R(t+1,r,pq) + R(r+1,n,\varepsilon) \qquad \text{(II.20)}$$

où $\varepsilon = \left\lceil \hat{q}y_r\ (q/N_v)^{c-1} \right\rceil d$.

En appliquant cette équation générale pour le calcul du coût d'accès des trois organisations des figures (II.3), (II.4) et (II.5), une comparaison entre ces organisations peut être faite.
Les hypothèses suivantes sont considérées :

E(Bc,Na,p) = E(Eb,Na,p) = 1, les attributs sont stockés sur un seul bloc.

E(Bc,Nv,q) = E(Eb,Nv,q) = 1, les mots clés d'un attribut occupent un seul bloc.

Au niveau r, chaque liste d'accès occupe un seul cylindre et pas plus.

Au niveau n; la taille de l'enregistrement est inférieure à la taille du bloc.

Pour l'organisation liste inversée, les paramètres sont : (Fig.II.4)

$W_1 = Na$, $W_2 = Nv$, $W_3 = L$, $W_4 = 1$

$Pi = 0$ $i=1,2,3$ $P4 = 1$

D'où

$$C_L = S'(Na,p) + p\,S'(Nv,q) + pq\,S(L) + V(\varepsilon)$$

$$C_L = T_T + p\,T_T + pq\,(t_s+t_r+\left\lceil\frac{L}{Eb}\right\rceil T)+G(Ce,\varepsilon)\,t_s+G(Be,\varepsilon)\,(t_r+T)\ldots$$

où $T_T = t_s+t_r+T$ temps d'accès et transfert d'un bloc...

et $\varepsilon = \left\lceil \hat{q}(q/Nv)^{c-1} \right\rceil d$ (II.21)

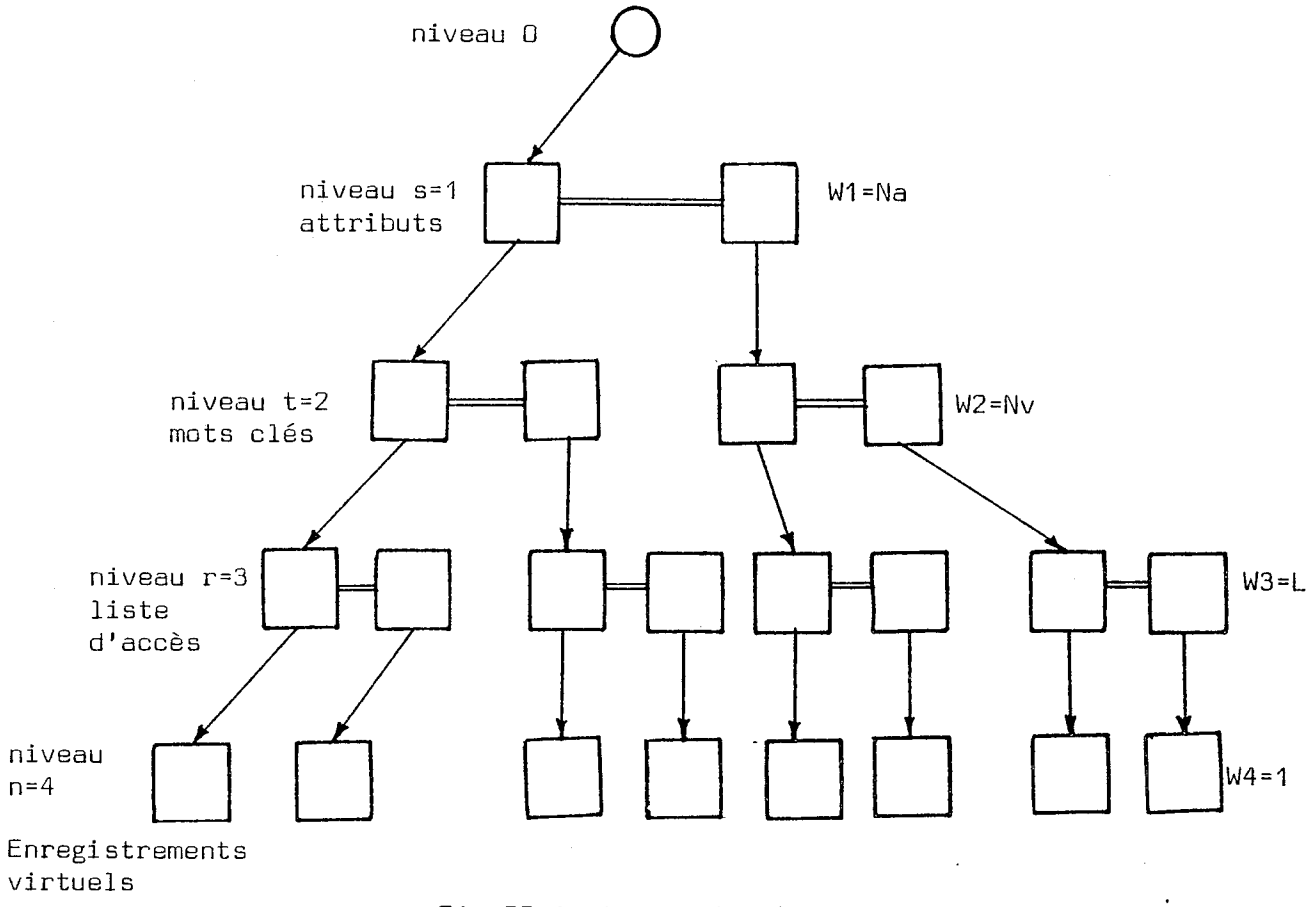


Fig.II.4. Arbre d'accès des listes inversées

où Na : nombre d'attributs de la BD

$Na = \Pi_{j=1}^{s} W_j$

Nv : nombre de mots clés/attributs

$Nv = \Pi_{j=s+1}^{t} W_j$

L = Longueur moyenne d'une liste

L = taille moyenne des ensembles E(kv)

$L = \Pi_{j=t+1}^{n} W_j$

Les paramètres de l'organisation multiliste sont :(Fig.II.5)

$s = 1,\ t = r = 2,\ n = 3 \quad W_1 = Na,\ W_2 = Nv,\ W_3 = L$

$P_1 = P_2 = 0 \quad P_3 = 1$

Le coût de la requête sera :

$$C_M = S_1(1,p) + p\ S_2(2,q) + R_3(3,\varepsilon)$$

$$= S'(Na,p) + p\ S'(Nv,q)) + V(\varepsilon L)$$

$$= T_T + p\ T_T + V\ \{\lceil \hat{q}(q/Nv)^{c-1} \rceil\ d\ L\} \qquad (II.22)$$

Pour l'organisation cellulaire, s=1, t=2, r=3 et n=4 (Fig.II.6)

$W_1 = Na,\ W_2 = Nv,\ W_3 = \left\lceil \frac{L}{R_s} \right\rceil = Lr$ et

$W_4 = R_s$ = longueur moyenne des chaînes dans une cellule.

$P_i = 0 \quad i = 1,2,3$ et $P_4 = 1$.

$$C_c = S'(Na,p) + p\ S'(Nv,q) + pq\ S(Lr) + V(\varepsilon\ Rs)$$

$$= T_T + p\ T_T + pq\left(ts + tr + \left\lceil \frac{Lr}{Eb} \right\rceil T\right) + G(Ce, \varepsilon\ Rs)\ ts +$$

$$G(Be, \varepsilon\ Rs)(tr+T) \qquad (II.23)$$

où $\varepsilon = \lceil \hat{q} Lr (q/Nv)^{c-1} \rceil\ d$

et $Lr = \lceil L/Rs \rceil$

Dans ce modèle, le calcul du nombre de blocs essentiels est très approximatif, des expressions plus proches de la réalité seront données dans le paragraphe II.7.

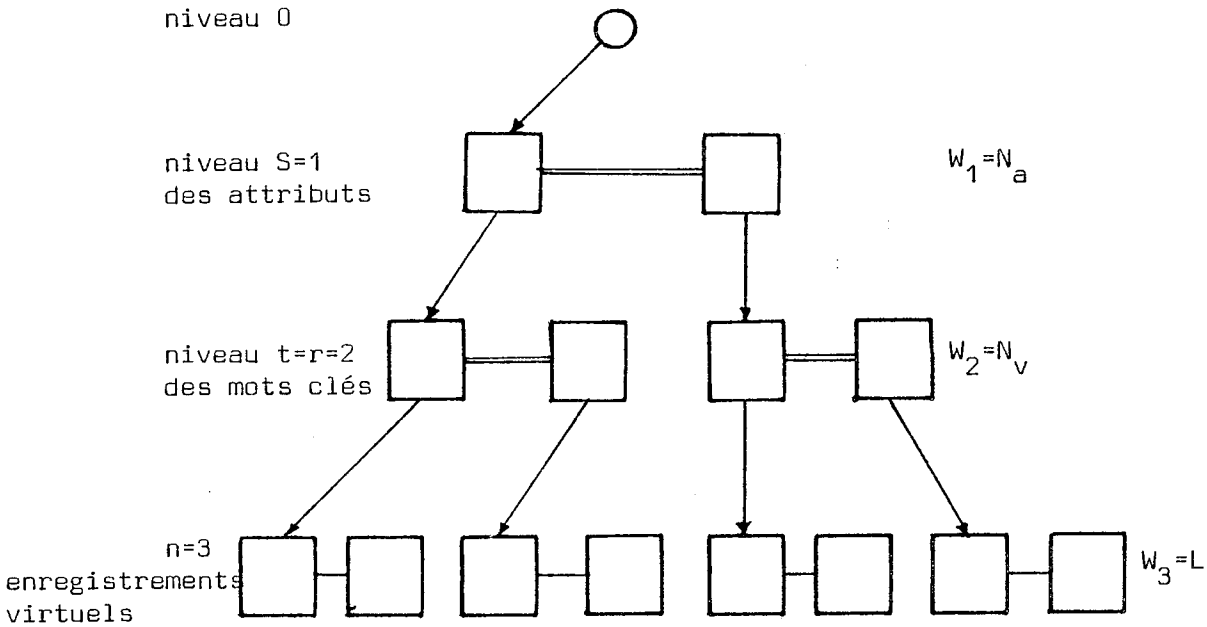


Figure II.5 : Arbre d'accès des multilistes

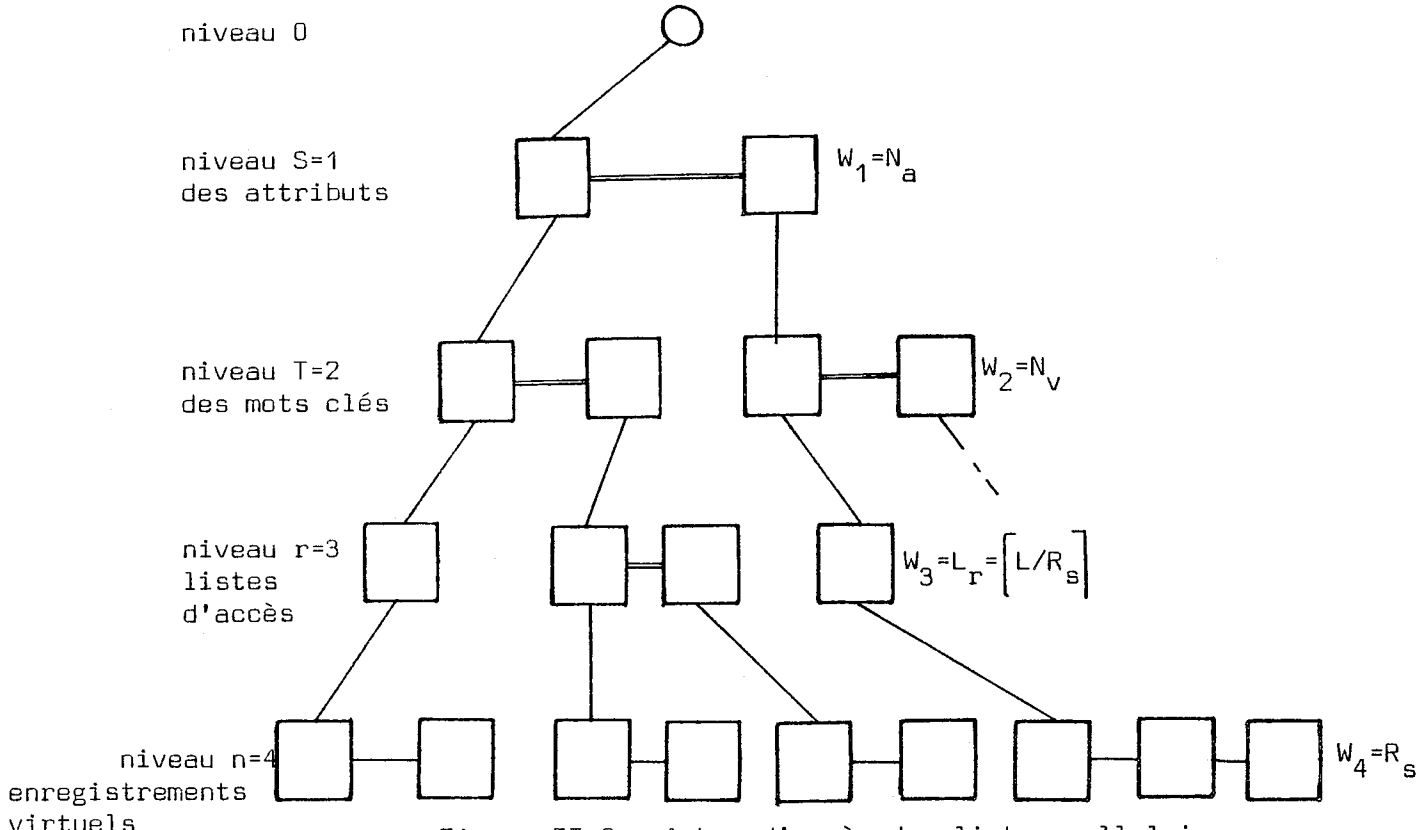


Figure II.6 : Arbre d'accès des listes cellulaires

III.5 EVALUATION DU TEMPS DE REPONSE A UNE DEMANDE DE LECTURE D'UNE PAGE D'UNE MEMOIRE SECONDAIRE [GELE 77-A]

Un modèle général pour évaluer le temps d'attente dans une file d'attente d'un serveur autonome[1] est présenté dans [GELE 77.A]. Ce modèle est applicable aux unités de mémoires secondaires, disques à tête fixe ou à bras mobile. L'application de ce modèle pour l'évaluation du temps de réponse à une demande de lecture d'une page d'un tambour magnétique est faite.

Pour le modèle général, les intervalles d'arrivée des clients ou demandes au serveur sont $A_1, A_2, \ldots A_n$.

Le temps de service des clients est $S_1, S_2, \ldots S_n$.

Après le service du $n^{ième}$ client, le serveur est indisponible pour un temps T_n.

On définit $S_n = s_n + T_n$ (II.24)

Si la file d'attente est vide après le départ du client dont le service a commencé à l'instant τ_k, le serveur est alors disponible aux instants $\tau_k + \overline{s_1}$, $\tau_k + \overline{s_1} + \overline{s_2}, \ldots \tau_k + \overline{s_1} + \ldots + \overline{s_n}$

Les temps d'attente du 1er, 2e,... $n^{ième}$ client sont respectivement $w_1, w_2, \ldots w_n$.

Ces temps satisfont l'équation récurrente

$$w_{n+1} = \Pi_n(-w_n - \varepsilon_n), \quad n \geq 1 \ldots \quad (II.25)$$

où $\varepsilon_n = S_n - A_{n+1}$

$$\Pi_n(x) \begin{cases} -x & \text{si } x \leq 0 \\ \sum_1^{l(x)} \overline{S_j} - x & \text{si } x > 0 \end{cases}$$

et

$$l(x) = \min \{l : \sum_1^l \overline{S_i} \geq x, \ l > 0\}$$

1. Un serveur autonome est un serveur qui devient indisponible pour un certain temps après chaque service rendu aux clients

Pour un tambour magnétique divisé en N secteurs, et accomplissant une rotation complète en un temps T, le temps de service d'un client est T/N.

Les demandes de lecture/écriture sont adressées aux N secteurs, formant N files d'attente.

Le transfert d'une page s'effectue quand le secteur correspondant passe devant la tête de lecture/écriture. Ce transfert s'exécute en un temps T/N, puis le service à ce secteur est interrompu pour un temps $T(\frac{N-1}{N})$.

Le temps d'attente W dans la file d'attente d'un secteur est donné par l'équation (II.26).

$$W = V+Y \qquad \text{(II.26)}$$

où V est le temps d'attente dans une file correspondante GI/D/1[1] ayant un temps constant de service T.
Y est une variable aléatoire répartie uniformément entre [0,T]. Un cas particulier où l'arrivée des demandes suit une loi de Poisson, donne une moyenne du temps d'attente

$$EW = T/2 + \lambda T^2/2(1-\lambda T) \qquad \text{(II.26)}$$

où λ est le taux d'arrivée des demandes.
Le temps de réponse est :

$$R = V+Y+T/N \qquad \text{(II.27)}$$

(1) Dans la notation GI/D/1 des files d'attente, le premier terme exprime la répartition des intervalles de temps entre les arrivées des clients, le 2e exprime la répartition du temps de service, et le 3e le nombre de serveur.
GI : général indépendant
D : déterministe

II.6 EVALUATION DU TEMPS DE REPONSE AUX DEMANDES DE TRANSFERT DES ENREGISTREMENTS DE LONGUEUR VARIABLE [GELE 75]

Un modèle présenté dans [GELE 75] évalue le temps moyen de réponse aux demandes de transfert des enregistrements de longueur variable stockés sur un disque à tête fixe ou un tambour magnétique. Le début de tous les enregistrements est à une position angulaire fixe sur la surface du disque ou tambour.

Les longueurs des enregistrements à tranférer sont indépendantes et identiquement réparties suivant une fonction de répartition F(x), dont le premier et le second moment ont des valeurs finies.

$$F(x) = 1-e^{-\mu x} \qquad \text{(II.28)}$$

De même les arrivées des demandes de transfert sont indépendantes et identiquement réparties suivant une fonction de répartition G(t)
où $G(t) = 1-e^{-\lambda t}$ (II.29)

Ces demandes sont servies dans l'ordre de leur arrivée (FCFS).
La probabilité d'arrivée de k demandes dans un temps T

$$a_k = \frac{e^{-\lambda T}(\lambda T)^k}{k!} \qquad k = 0,1,2... \qquad \text{(II.30)}$$

Le temps moyen de réponse W_F est déduit dans [GELE 75]

$$W_f = T\left[E(l)+\frac{1}{2}+\lambda T\ \frac{E(n^2)}{2(1-\lambda T\ E(n))}\right] \qquad \text{(II.31)}$$

où T est le temps d'une rotation complète du disque,
n le nombre de rotation du disque pour le transfert d'un enregistrement
l est la longueur d'un transfert.

$$E(l) = 1/\mu$$

$$E(n) = 1/(1-e^{-\mu})$$

$$E(n^2) = \frac{1+e^{-\mu}}{(1-e^{-\mu})^2}$$

Si les enregistrements sont longs, $\mu \ll 1$,

$$E(n) \simeq \frac{1}{\mu}+\frac{1}{2}$$

$$E(n^2) = \frac{2}{\mu 2}+\frac{1}{\mu}-1.$$

II.7 PROPOSITION D'AMELIORATION

On s'intéresse à l'étude des trois organisations multilistes, listes cellulaires et listes inversées.

Rappelons que le temps d'accès à une BD pour constituer la réponse à une requête est nettement divisé en une recherche dans le dictionnaire pour retrouver les attributs et mots clés de la requête, puis une lecture des listes d'accès et des enregistrements. Une équation du temps de recherche dans le dictionnaire de la BD pour les trois organisations sera donnée.

Le temps de la lecture des listes d'accès des enregistrements sera donné séparément pour chacune de ces trois organisations, car une équation générale ne sera pas utile; de grandes approximations devront être faites et écarteront le modèle de la réalité et de l'implémentation réelle du SGBD.

Pour calculer le nombre de blocs et de cylindres à accéder séquentiellement pour la recherche de β enregistrements parmi N enregistrements, et le temps de cet accès on utilisera les équations II.7, II.8 et II.9 de YAO. Pour la lecture des N enregistrements l'équation II.10 sera utilisée.

De même, pour la recherche dans une chaîne de longueur Le, le nombre de blocs et de cylindres, et le temps d'accès seront donnés par les équations II.11, II.12, II.13 et II.14. Pour le calcul du nombre de blocs à accéder pour la recherche des β enregistrements dans une chaîne répartie sur Be blocs, YAO a donné une expression plus précise dans [YAO 77-A] qui est :

$$G(Be,\beta) = Be \left[1 - \prod_{i=1}^{\beta} \frac{Nd - i+1}{N - i+1} \right] \qquad (II.32)$$

ou d = 1-1/Be

N est le nombre d'enregistrements stockés dans les Be blocs.

Une bonne simulation du mouvement du mécanisme d'accès aux fichiers de la BD est obtenue en calculant le nombre moyen de cylindres à accéder séquentiellement pour la recherche dans (ou la lecture de) e ensembles filiaux parmi E ensembles filiaux enregistrés sur C_F cylindres; ce nombre est :

$$E(F_c,E,e) = 1+C'_F - \frac{1}{C_e^E} \sum_{i=1}^{C'_F} C_e^{iF_c} \qquad \text{(II.33)}$$

où F_c est le nombre d'ensembles filiaux par cylindre

$$F_c = \left\lceil \frac{\text{taille d'un cylindre}}{\text{taille d'un ensemble filial}} \right\rceil = \left\lceil \frac{C}{W_i} \right\rceil$$

Ce nombre $E(F_c,E,e)$ représente le nombre de cylindres traversés par le bras d'accès durant la recherche ou la lecture. Les deux fonctions suivantes sont ajoutées :

La première est utilisée pour calculer le temps de recherche dans le dictionnaire de la BD.

Considérons qu'un niveau du dictionnaire composé de E ensembles filiaux stockés séquentiellement, chacun de taille W_i, si e ensembles sont à accéder pour la recherche de ß éléments dans chaque ensemble, le temps d'accès et de transfert sera :

$$Z'(E,e,\beta) = t_s + [E(F_c,E,e)-1]\, t'_s + e\,[t_r + E(E_b,w_i,\beta)T] \qquad \text{(II.34)}$$

Le système accède séquentiellement un nombre de cylindres $E(F_c,E,e)$ sauf le premier, puis pour chaque ensemble filial un accès aléatoire au premier bloc d'un groupe de $E(E_b,W_i,\beta)$ blocs stockés séquentiellement, puis le transfert de ces $E(E_b,W_i,\beta)$ blocs est effectué.

La seconde fonction calcule le temps moyen de lecture de e ensembles filiaux au niveau des listes d'accès composés de E ensembles filiaux

$$Z(E,e) = t_s + [E(F_c,E,e)-1]\, t'_s + e\,[t_r + \left\lceil \frac{W_i}{b} \right\rceil T] \qquad \text{(II.35)}$$

où b est la taille d'un bloc et $\left\lceil \frac{W_i}{b} \right\rceil$ est le nombre de blocs contenant un ensemble filial.

Le système accède séquentiellement $E(F_c,E,e)$ cylindres sauf le premier, puis pour chacun des e ensembles filiaux, un accès aléatoire au premier bloc d'un groupe de $\left\lceil \frac{W_i}{b} \right\rceil$ blocs stockés séquentiellement puis le transfert de ces blocs est effectué.

Dans ce qui suit le temps de recherche, d'accès et de transfert pour constituer la réponse à une requête de complexité (d,c,p,q) sera donné.

Rappelons que la requête prend une forme disjonctive normale, elle est composé de d conjonctions, c attributs sont en moyenne spécifiés par conjonction, p est le nombre total d'attributs de la requête, et q est le nombre de mots clés spécifiés par attribut.

Temps de recherche des p attributs : Ra

Généralement, les Na attributs de la BD sont enregistrés en un seul niveau sur le support physique. D'où le temps de la recherche sera simplement :

$$Ra = (1-Pa)S'((1-Pa),p) + Pa\ V'(Pa\ Na) \qquad (II.36)$$

où Pa est le rapport de débordement au niveau des attributs.

Temps de recherche des pq mots clés : Rk

Le nombre d'ensembles filiaux au niveau des mots clés est Na, la taille de chaque ensemble est Nv.

Le temps de recherche des pq mots clés est obtenu en appliquant l'équation (II.34); on mettra E = Na, le nombre d'ensembles filiaux dans le niveau e = p. est le nombre moyen d'ensembles filiaux à accéder pour la recherche de q mots clés par ensemble filial. D'où

$$Rk = (1-Pk)\ Z'(Na,p,q) + Pk\ V'(p\ Pk\ Nv) \qquad (II.37)$$

où Pk est le rapport de débordement au niveau des mots clés.

Le temps de lecture des listes d'accès et des enregistrements sera donné séparément pour chacune des trois organisations listes inversées, cellulaires et multilistes.

A) ORGANISATION LISTES INVERSEES

Temps de lecture des listes d'accès : RL

Le nombre de listes d'accès à lire est pq, chacun a une taille moyenne de L. Le nombre total de listes d'accès à ce niveau est Na.Nv. D'où le temps de lecture sera :

$$RlL = (1-PL)Z(Na\ Nv, pq) + PL\ V(pq\ PL\ L) \qquad (II.38)$$

où PL est le rapport de débordement au niveau des listes d'accès.

Temps de lecture des enregistrements : ReL

Si N est le nombre total d'enregistrements de la BD. La taille E_r de l'ensemble des enregistrements répondant à une requête composée de d conjonctions chacune de c attributs et chaque attribut q mots clés est donné par l'équation II.39. Les attributs et les mots clés sont également demandés dans les requêtes.

$$E_r = N(q\frac{1}{Nv})^c : d \qquad (II.39)$$

Cet ensemble d'enregistrements est réparti aléatoirement sur le support physique dans une zone de Be blocs.

Le nombre de blocs qui peuvent contenir au moins un de ces E_r enregistrements est $G(B_e, E_r)$.

Si le système effectue un tri des adresses contenues dans les listes d'accès avant l'accès aux enregistrements, la récupération de cet ensemble prendra un temps

$$ReL = t_s + \left[E(B_c, B_e, E_r) - 1\right] t'_s + G(B_e, E_r)(T + t_r) \qquad (II.40)$$

Si le tri n'est pas effectué, l'accès aux cylindres sera aléatoire

$$ReL = G(C_e, E_r)\ t_s + G(B_e, E_r)(T + t_r) \qquad (II.41)$$

B) ORGANISATION LISTES CELLULAIRES

Temps de lecture des listes d'accès : RLc

Dans ce cas les listes ont une longueur Lc, d'où le temps de lecture des pq listes sera :

$$RlC = (1-PL)Z(Na\ Nv, pq) + PL\ V(pq\ PL\ Lc) \qquad (II.42)$$

Temps de lecture des enregistrements : Rec

L'ensemble des enregistrements contenant q mots clés d'un attribut de la BD a une taille $e_r = q\ N/Nv$

à rappeler que N est le nombre d'enregistrements de la BD

Nv : le nombre de mot clé/attribut.

Cet ensemble est réparti aléatoirement sur les Ce cylindres de la BD, et le nombre de cylindre contenant au moins un enregistrement de cet ensemble est

$$k = G(Ce,er) = Ce(1-(1-\frac{1}{Ce})^{er})$$

Considérons que les attributs sont également interrogés par les requêtes, le nombre de cylindre à accéder par conjonction est :

$$Ce \times [1-(1-\frac{1}{Ce})^{er}]^{c}$$

Le nombre total de cylindres accédés pour obtenir la réponse des d conjonctions est $Cr = d\ Ce\ [1-(1-\frac{1}{Ce})^{er}]^{c}$ (II.43)

Dans chaque cylindre accédé, une liste de longueur R_s est répartie sur $Bc\ [1-(1-\frac{1}{Bc}^{Rs}]$ blocs.

où Bc est le nombre de blocs par cylindre.

D'où le nombre de blocs contenant la réponse est Br

$$Br = Cr\ \ Bc\ [1-(1-\frac{1}{Bc})^{Rs}] \qquad (II.44)$$

Le temps d'accès aux enregistrements est alors

$$Rc = ts\ Cr + Cr\ Bc\ [1-(1-\frac{1}{Bc})^{Rs}]\ \ (tr+T) \qquad (II.45)$$

C) ORGANISATION MULTILISTES

Pour la lecture des enregistrements, pour chacune des d conjonctions de la requête, une liste de $\hat{q}L$ enregistrements est parcourue. Cette liste est répartie aléatoirement sur

$$Cr = Ce\ [1-(1-\frac{1}{Ce})^{\hat{q}L}] \text{ cylindres}$$

et $Br = Be\ [1-(1-\frac{1}{Be})^{\hat{q}L}]$ blocs.

D'où le temps d'accès aux enregistrements

$$RM = d\ Cr\ Ts + d\ Br\ (tr+T) \qquad (II.46)$$

L'accès aléatoire aux cylindres et aux blocs provient de la nature du parcours des listes.

Le temps de réponse calculé utilisant notre modèle prend mieux en compte la nature de l'accès aux dictionnaires et aux fichiers des BD que le modèle de YAO.

Une simulation de ce modèle présente une aide au concepteur des SGBD pour le choix de l'organisation physique qui convient aux applications gérées par le système.

II.8 CONCLUSION

Une étude plus poussée dans le domaine de l'évaluation des SGBD est à effectuer, car toutes les évaluations apparues calculent le temps de réponse à une certaine requête, ou la moyenne du temps de réponse pour le transfert d'un bloc ou d'un enregistrement pour une certaine répartition d'arrivées de ces demandes de transfert.

Le calcul du temps moyen de réponse à une requête en prenant en considération plus de détails sur la répartition de ces requêtes avec le temps, et plus de détails sur l'attente dans les files d'attente du système sera un bon critère de comparaison entre les différents SGBD.

CHAPITRE III

GENERALITES SUR LA SURETE DE FONCTIONNEMENT DES SGBD

MODELES ANALYTIQUES DES STRATEGIES DE REDEMARRAGE DES SGBD

III.1 INTRODUCTION

Le problème de la sûreté de fonctionnement doit être pris en compte dès les premières étapes de la conception d'un SGBD (cahier de charges). Ce problème concerne le concepteur au niveau de la définition du logiciel et du matériel gérant les BD, mais aussi l'utilisateur qui en supporte les conséquences.

Les différentes techniques adoptées pour obtenir une bonne sûreté de fonctionnement des SGBD ont évolué, ceci est dû à des progrès importants :

- du coût du matériel,
- du progrès du logiciel et du matériel,
- de l'extension du domaine d'application des BD.

Ces techniques consistent essentiellement :

- à utiliser du matériel à haute fiabilité et à concevoir un logiciel plus fiable, pour minimiser l'apparition des défauts,
- à introduire des redondances afin d'assurer la survie de la BD à la majorité des pannes.

La mise en oeuvre de chacune de ces techniques est étroitement liée aux applications utilisant les BD et aux objectifs fixés dans le domaine de la sûreté que dans celui des performances.

Dans ce chapitre sont exposés les principaux concepts utilisés dans l'étude de la sûreté de fonctionnement d'un SGBD. Les définitions et les explications de ces concepts sont présentées.

Les causes des pannes et les différentes stratégies de redémarrage des SGBD sont brièvement présentées dans le paragraphe III.3 et III.4 respectivement.

Dans le paragraphe III.5 de ce chapitre, on présentera les modèles analytiques apparus pour modéliser la stratégie classique de redémarrage des SGBD après la détection d'une panne. Cette stratégie est fondée sur la création des points de reprise et la construction d'un audit-trail. Ces modèles calculent le temps optimal entre les points de reprise.

Dans le paragraphe III.6, un modèle analytique est proposé, ce modèle diffère des précédents, car il étudie des SGBD qui utilisent des fichiers différentiels pour stocker les modifications à effectuer aux fichiers des BD.
Cette étude concerne donc les applications qui utilisaient les fichiers différentiels pour déterminer le temps optimal entre les fusions de ces fichiers et la BD. Notre modèle calcule ce temps et notre but est de minimiser le coût du système durant sa vie.

III.2 GENERALITES SUR LA SURETE DE FONCTIONNEMENT DES SGBD

Dans ce paragraphe la terminologie et les définitions des différentes composantes de la sûreté de fonctionnement d'un SGBD sont présentées [MOAL 76-A, BELL 77, RICH 78, ADIB 76, SOUL 76, KATZ 73].

Le SGBD est un ensemble de logiciel collaborant avec un système hôte permettant l'exploitation des BD et ceci pour remplir une ou plusieurs fonctions précises (contrôle des processus, gestion,...).

Le terme défaut désigne une défectuosité du matériel ou du logiciel dûe à une imperfection de conception ou à un non respect des règles technologiques, ou à l'usure du matériel.

Le terme panne indique l'effet produit par un défaut. Dans un SGBD, une panne peut soit bloquer le système ne permettant pas l'accès aux données, soit bloquer un ou plusieurs utilisateurs, soit menacer l'intégrité de la BD.

Le terme erreur, désigne le résultat incorrect généré lors du fonctionnement du système, cette valeur erronée est enregistrée dans la BD ou communiquée au monde extérieur.

Rappelons la définition de la sûreté de fonctionnement d'un système informatique : "La sûreté de fonctionnement est l'aptitude d'un système à minimiser l'apparition des défauts et/ou à minimiser leurs effets".

La sûreté de fonctionnement d'un système informatique est une grandeur vectorielle ayant comme composantes la fiabilité, sécurité, disponibilité, crédibilité, maintenabilité et réparabilité. Les définitions de ces composantes se trouvent dans [BELL 77].

Dans le cas d'un SGBD, la sûreté de fonctionnement englobe en plus de ces concepts classiques, des concepts d'intégrité, de confidentialité, d'intimité et de contrôle d'accès [RICH 78]. Suivant le type de système et ses applications, l'importance que l'on attache à chaque concept varie.

Pour définir les composantes de la sûreté de fonctionnement d'un SGBD on distingue les états dans lesquels il peut se trouver (Fig.III.1).

Etat $[Q_n]$: état de fonctionnement normal, le système peut recevoir et exécuter les requêtes des utilisateurs. Cet état peut comporter une panne mais elle ne s'est pas encore manifestée.

Etat $[Q_e]$: état de fonctionnement erroné. Une ou plusieurs erreurs existent, mais elles ne sont pas détectées. Cet état est divisé en deux sous états :

- Etat $[Q_{e1}]$: la BD est contaminée, et le support physique contient des valeurs erronées.
- Etat $[Q_{e2}]$: la BD est encore intacte.

Etat $[Q_l]$: état dans lequel la panne est localisée et réparée.

Si la localisation et l'identification de la panne montre que le système était à l'état Q_{e2}, le système passe à l'état Q_n, sinon il passe à l'état Q_r.

Etat $[Q_r]$: redémarrage du système

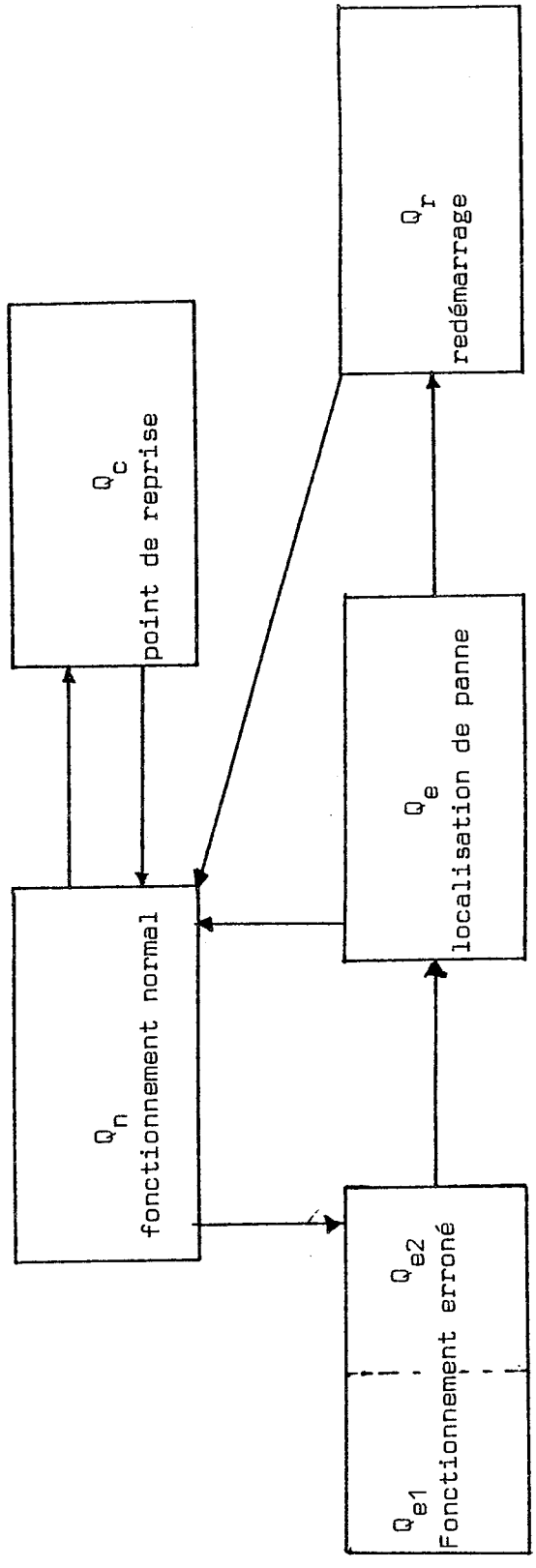


Figure III.1 : Etats d'un SGBD

Etat $[Q_c]$: état de création d'un point de reprise, c'est-à-dire d'enregistrer une copie de la BD sur un autre support physique.

Dans les états $[Q_l]$, $[Q_r]$ et $[Q_c]$ le système retarde l'exécution des requêtes qui lui arrivent jusqu'à ce qu'il revienne à l'état $[Q_n]$.

La fiabilité R(T) d'un SGBD à un instant T est la probabilité que le système soit en bon fonctionnement, c'est-à-dire qu'il soit aux états Q_n ou Q_c ou Q_r entre 0, T.

La disponibilité A(T) d'un SGBD à un instant T est la probabilité que le système soit en état de fonctionnement normal Q_n à l'instant T.

La sécurité d'un SGBD à un instant T est la probabilité que le système n'ait pas communiqué des valeurs erronées au monde extérieur. C'est la probabilité que le système ne soit pas à l'état Q_e, entre 0 et T. Une expression explicite de la sécurité d'un SGBD n'est pas facile à déduire car elle dépend de la nature de la communication entre le système et les applications lors de l'occurrence de la panne.

Pour les SGBD, on inclura le concept de maintenabilité dans la réparabilité. Elle est définie comme la probabilité que le système ne soit plus à l'état Q_l ou à l'état Q_r à l'instant T sachant qu'il y était à l'instant 0.

On définit l'intégrité à l'instant T par la probabilité que le système ne soit pas à l'état Q_{e1} entre 0 et T. Le terme cohérence sera considéré comme synonyme de l'intégrité.

La cohérence peut être interne ou externe, cohérence des données entre elles, ou cohérence des données avec le monde extérieur.

La confidentialité dans un SGDB désigne la protection des données contre toute manipulation non autorisée.

La définition des autres concepts se trouve dans [RICH 78].

Un des buts essentiels des concepteurs des SGBD est de garantir l'intégrité de leur BD et d'offrir des moyens pour protéger la BD contre tout défaut matériel ou logiciel dans le système. Ce défaut matériel ou logiciel produit une panne, qui peut causer une incohérence de la BD.

La panne peut alors se manifester différemment d'après sa provenance. Pour éviter la contamination de toute ou une grande partie des informations, après la détection d'une panne, il faut arrêter le système et le redémarrer d'un point sûr. Un point sûr est un point auquel le système était à l'état Q_n. La stratégie de redémarrage est fixée d'après les critères jugés nécessaires par les concepteurs des SGBD.

Pour les SGBD, ces critères peuvent être classés en trois catégories [YOUR 72, MART 73].

1) *Exigence d'une haute sécurité et d'une haute disponibilité*

Certain SGBD fonctionnent dans des environnements où les pannes peuvent avoir des conséquences catastrophiques. La sécurité et la disponibilité du système deviennent des facteurs très importants dans la conception. Comme exemple de ces systèmes, citons les systèmes de contrôle de processus industriels ou de contrôle du trafic aérien.

2) *Exigence d'une bonne disponibilité (la sécurité étant moins importante)*

Ces systèmes nécessitent un redémarrage rapide après la détection d'une panne, pour retourner à une BD cohérente. Ces systèmes demandent un temps moyen de réparation d'une panne MTTR ne dépassant pas une certaine limite. Comme exemple citons les systèmes de réservation des billets d'une société aérienne, le client ne peut pas attendre des heures pour obtenir son billet si une panne se manifeste dans le système, un MTTR de quelques minutes est alors exigé.

3) Exigence d'une bonne performance

Le concepteur s'attachera à obtenir un rendement maximum, en minimisant l'overhead dû aux redondances introduites dans le système. Comme exemple, citons un centre de calcul gérant plusieurs BD d'utilisateurs différents en mode "batch".

III.3 CAUSES DES PANNES DANS LES SGBD

Fonctionnellement, nous pouvons distinguer deux sortes de pannes, une panne grave et une panne bénigne.
Une panne grave concerne le système entier et tous les utilisateurs. Le blocage du système ou la contamination d'une grande partie de la BD sont des pannes graves. Une panne bégnine concerne un ou plusieurs utilisateurs : soit ils sont isolés du système et ne peuvent pas accéder à la BD, soit une valeur erronée a été stockée dans leurs enregistrements.

Les causes de pannes d'un SGBD peuvent se classer comme ce qui suit [YOUR 72, DAVE 76, ABRI 72].

III.3.1 PANNE DES PROCESSEURS

L'U.C peut ne pas exécuter correctement une instruction ou un groupe de plusieurs instructions. Cette panne peut être catastrophique car le système peut stocker des informations correctes dans des adresses fausses ou stocker des informations erronées dans des adresses correctes ou fausses.

Le système peut exécuter des programmes de test dans le temps d'oisiveté de l'U.C ou sur un autre processeur pour un système multiprocesseur pour tester constamment le bon fonctionnement de l'UC [ROBA 75, ROBA 76, ROBA 78].

III.3.2 PANNE DE LA MEMOIRE CENTRALE

Dans la plupart des mémoires, un bit de parité est associé à chaque mot ou octet, ce qui rend possible la détection d'une certaine classe de pannes dans ces mémoires et l'initialisation d'une interruption pour la reconfiguration des modules de la mémoire et le redémarrage du système.

Avec l'évolution de la technologie, la fiabilité des processeurs et des mémoires s'est améliorée, ce qui rend très rare mais toujours possible la probabilité de l'occurrence d'une de ces pannes.

III.3.3 PANNE DANS LE RESEAU DE COMMUNICATION ET DES PERIPHERIQUES

Dépendant de la localisation de la panne, un ou plusieurs utilisateurs seront concernés. La figure III.2 montre les différentes pannes susceptibles de menacer l'intégrité de la BD.

Une panne d'un terminal, des modems et des lignes de communications à faible vitesse peut nuire à un nombre restreint d'applications ou d'utilisateurs, et empêcher leur accès à la BD. Une panne dans cette zone n'est pas trop dangereuse si le terminal isolé ne gère pas des applications principales dans le système.

Une panne dans la zone 2 peut nuire à tous les utilisateurs de la BD.

Si des techniques logicielles évoluées de détection des erreurs, pendant la transmission des informations des terminaux jusqu'à la BD, sont préservées, même après l'occurrence d'une panne dans la zone 1 ou la zone 2, seul l'accès à la BD est perturbé.

III.3.4 PANNE DU CANAL

Une panne du canal disque peut causer un décalage d'un enregistrement d'un ou plusieurs caractères à gauche ou à droite, ou peut mettre à zéro 2 à 3 caractères d'un enregistrement pendant son transfert. Ce type de panne même s'il est rare peut causer un comportement imprévisible

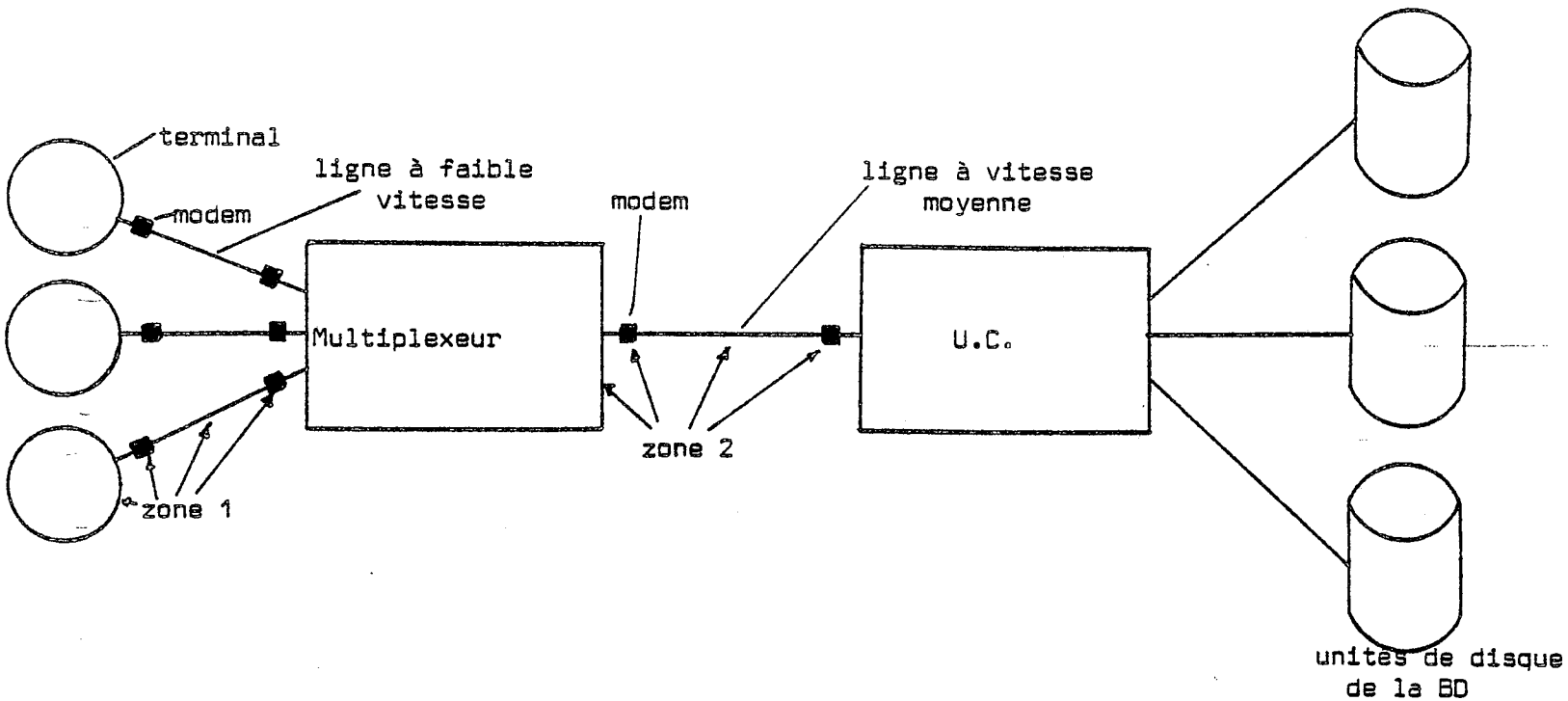


Figure III.2 : Réseau de communication dans un système de BD

du système et des applications si le transfert d'informations va du disque vers la mémoire centrale et peut nuire aussi à l'intégrité de la BD si le transfert est dans l'autre sens.

III.3.5 PANNE DU SYSTEME ELECTROMECANIQUE D'ACCES AU DISQUE

Une panne dans le système électromécanique d'accès au disque peut causer la chute brusque "head-crash" de la tête de lecture/écriture sur la surface magnétique du disque tournant, et rend ainsi une partie de la BD illisible. Si les dictionnaires de la BD sont stockés sur cette zone du disque, le système doit être arrêté et un redémarrage d'un point sûr est nécessaire. Si des informations concernant un utilisateur sont stockées sur cette zone, le système peut continuer son fonctionnement normal et récupérer ces informations soit en les demandant à l'utilisateur, soit par un redémarrage limité.

III.3.6 ERREUR DE L'OPERATEUR

Les erreurs que peut commettre l'opérateur peuvent se classifier en ce qui suit :

- durant l'initialisation du système,
- durant la marche du système,
- avant, durant et après une panne,
- fermeture du système à la fin de la journée.

III.3.7 PANNE D'ALIMENTATION

Cette panne cause un arrêt brusque des divers composants physiques du système, si les précautions nécessaires ne sont pas prises pour sauvegarder les informations par des interrupteurs spéciaux, la BD et le système peuvent ne pas être cohérents.

III.3.8 ENVIRONNEMENT

Dans ce paragraphe on inclue tous les désastres qui peuvent causer une dégradation physique du support de la BD; lors d'un transfert un disque peut tomber et se briser, l'effet de l'humidité, température, feu etc peut nuire aux contenus des unités de disques de la BD.

III.3.9 PANNES LOGICIELLES

Un utilisateur seulement peut être en panne logicielle car dans son application il a fait des erreurs qui causent par exemple un bouclage. Plusieurs utilisateurs peuvent être en panne dans un "deadlock", un conflit existe entre l'exécution de leur application.

III.3.10 DES PANNES INEXPLICABLES

Certaines pannes peuvent se manifester, le système est perturbé sans aucun message d'erreur et l'identification de la source de la panne est impossible.

III.4 DIFFERENTES PHILOSOPHIES DE PRESERVATIONS DE L'INTEGRITE DE LA BD

Il y a quatre philosophies distinctes pour préserver l'intégrité de la BD, elles sont résumées dans ce qui suit [YOUR 72, DAVE 76, FOSS 74, SEVE 76, WILK 72, FRAZ 69].

III.4.1 GENERATION

Dans cette méthode, souvent appelée grand-père/père/fils, les requêtes sont exécutées à partir d'une version physique de la BD pour produire une nouvelle version qu'on enregistre sur un autre support physique. La version précédente et les requêtes sont gardées, de même que la nouvelle version. Si une panne se manifeste, le fichier ou la BD est recréé à partir de l'ancienne version et des requêtes. Plus qu'une génération de la BD et les requêtes associées sont préservées.

III.4.2 "DUMPING"

Dans cette méthode, la BD est recopiée à des intervalles de temps prédéterminés sur un autre support physique, point de reprise, et il y a constitution d'un "audit-trail".

L'audit-trail est un enregistrement normalement stocké sur une bande magnétique et il peut contenir :

- le texte d'une requête,
- une copie d'un enregistrement avant sa modification par une requête que nous appelerons brièvement "image avant requête",
- une copie d'un enregistrement après l'exécution d'une requête, que nous appelerons brièvement "image après requête".

Si une panne se manifeste, le redémarrage du système peut s'effectuer selon une des quatre stratégies suivantes :

A) L'audit-trail est un journal d'ordre contenant l'ensemble des requêtes arrivées au système depuis la dernière recopie de la BD. Après l'occurrence d'une panne, il est difficile de préciser pendant l'exécution de quelle requête cette panne est arrivée, et quelles sont les informations erronées qui ont été stockées sur la BD. Pour restituer la BD, il faut d'abord recharger la dernière copie de la BD et reexécuter toutes les requêtes enregistrées depuis cette copie.

Cette méthode devient inefficace si l'une des trois conditions suivantes est vraie :

1) La BD est trop grande, et le rechargement de la copie prend un temps considérable.

2) Un grand nombre de requêtes s'accumulent sur l'audit-trail et/ou les requêtes nécessitent un temps de reexécution qui n'est pas négligeable.

3) Les pannes sont fréquentes, et la méthode de rechargement de la BD et reexécution des requêtes devient inefficace.

Il est à noter que cette méthode est effective pour les deux types de pannes graves et bénignes.

B) Dans cette méthode, l'audit-trail contient les requêtes arrivées au système depuis la dernière recopie et les "images avant requête".

Si une panne bénigne se manifeste, le rédémarrage du système n'est qu'un balayage de l'audit-trail pour préciser les requêtes qui n'ont pas terminé leur exécution, puis le rechargement des enregistrements demandés par ces requêtes à partir des "images-avant-requête" et la reexécution de ces requêtes.

Si la panne est grave, le redémarrage se fait comme dans A), le gain de cette méthode est que le rechargement de la BD ne se fait que pour les pannes graves.

C) Pour la troisième méthode, l'audit-trail est composé des "images-avant-requête". Le redémarrage se fait par un chargement de la copie de la BD et sa fusion avec l'audit-trail. Cette méthode est appréciée si la re-exécution des requêtes prend un temps considérable mais elle devient inefficace si le rechargement de la BD est lent.

D) Pour grouper les avantages des trois méthodes précédentes, l'audit-trail est formé d'un journal d'ordres, des "images-avant-requête" et des "images-après-requête". D'après le type et la localisation de la panne, la restitution de la BD s'effectue soit en utilisant les "images-avant-requête", soit la copie de la BD. Puis la fusion avec les "images-après-requête" ou la reexécution des requêtes donne une BD sûre.

III.4.3 FICHIERS DIFFERENTIELS [SEVE 76]

Une modification de ce qui précède donne naissance à une nouvelle philosophie, utilisant des fichiers différentiels. Dans cette méthode, l'insertion, la suppression et la mise à jour des enregistrements de la BD ne sont pas effectuées directement, mais un fichier différentiel est construit sur lequel sont enregistrées toutes les modifications de la BD. Ce fichier doit être accessible aux requêtes.

A des intervalles de temps prédéterminés, la BD et le fichier différentiel sont fusionnés pour recréer une nouvelle BD mise à jour.

Le "dumping" de cette BD peut être effectué à des intervalles de temps multiple du temps entre les fusions.

Si une panne se manifeste, la BD joue le rôle des "images-avant-requête" et la reexécution des requêtes donne une BD sûre. Une trace des enregistrements modifiés doit être gardée.

III.4.4 DUPLICATION

Des versions identiques du même fichier sont parallèlement modifiées. Ces fichiers sont stockés sur des unités différentes de disque. Cette méthode est utile seulement si la majorité des pannes du système sont associées aux disques de la BD. Une erreur, un "head-crask", même une brisure d'une unité de disque, ne causent pas une perte de la BD.

Cette méthode protège l'intégrité de la BD pour un nombre limité de pannes, de plus elle est trop coûteuse, car la taille du support physique est doublée, et de même pour la tâche des canaux effectuant les entrées-sorties. Pour diminuer ce coût, seule une duplication des parties critiques de la BD et des dictionnaires est effectuée.

III.5 MODELES ANALYTIQUES POUR LA STRATEGIE DE REDEMARRAGE DES SGBD CREANT DES POINTS DE REPRISE

Tous ces modèles font une analyse des SGBD créant des points de reprise, ils calculent le temps optimal entre ces points de reprise. Ces modèles ont des critères différents pour le calcul du temps optimal, certains minimisent le coût des pannes, le temps de réponse à une requête, d'autres cherchent une disponibilité maximum. Une bibliographie de ces modèles suit.

III.5.1 MODELE DE YOUNG

Le modèle de Young exposé dans [YOUN 73] est applicable pour les fichiers des BD de même pour les differentes tâches d'un système. A des points de reprises, on sauvegarde les fichiers de la BD et des informations suffisantes pour le redémarrage du système en cas de panne.

Le but du modèle est de minimiser le coût des pannes. Ces pannes sont aléatoires et le coût de la machine est essentiellement proportionnel au temps.

Le temps optimal entre les points de reprise T_c est obtenu en faisant une approximation du premier ordre.

$$T_c = \sqrt{2 T_s T_f} \qquad \text{(III.1)}$$

où T_s est le temps nécessaire pour sauvegarder la BD

T_f est le temps moyen entre les pannes.

III.5.2 MODELES DE CHANOY [CHAN 75]

La stratégie de redémarrage des systèmes modélisés par CHANOY est fondée sur la sauvegarde des fichiers de la BD à des intervalles égaux et la construction d'un audit-trail sur lequel sont stockées toutes les requêtes arrivées au système depuis la dernière sauvegarde. Trois modèles sont proposés; quatre hypothèses communes aux trois modèles sont faites :

1) La détection des pannes est aléatoire et suit une loi de Poisson.

2) Le temps de reexécution des requêtes sur l'audit-trail est proportionnel au nombre de requêtes qu'il contient.

3) Les requêtes arrivant au système pendant la sauvegarde des fichiers de la BD ou le redémarrage du système après une panne sont stockées jusqu'à la fin de cette tâche.

4) La disponibilité du système est considérée élevée.

Le modèle A) ajoute deux autres hypothèses :

1) Pendant la sauvegarde des fichiers ou le redémarrage du système la probabilité d'occurrence d'une panne est nulle.

2) Le débit d'arrivée des requêtes est constant.

Le modèle B) permet l'occurrence d'une panne pendant la sauvegarde ou le redémarrage du système mais pour lui, le débit des requêtes est toujours constant.

Le modèle C) est plus réaliste, car il considère une répartition des requêtes variable avec le temps, avec des sommets et des creux. Mais, les pannes durant le redémarrage ou les points de reprises sont négligées.

Le but de ces modèles est de minimiser l'overhead au système par unité de temps.

Pour le modèle A), le temps optimal entre les points de reprise est donné par l'équation

$$Topt = \sqrt{2F/\lambda k} \qquad \text{(III.2)}$$

où F est le temps de sauvegarde des fichiers de la BD

λ est le taux de panne

$$k = \mu/b = \frac{\text{taux d'arrivée des requêtes}}{\text{débit de reexécution des requêtes de l'audit-trail}}$$

L'overhead par unité de temps ropt prend la valeur suivante :

$$ropt = \lambda R + \sqrt{2\lambda kF} \qquad \text{(III.3)}$$

où R est le temps nécessaire pour réparrer les pannes et de charger la dernière copie de la BD.

Pour le modèle B), Topt satisfait l'équation :

$$\exp(\lambda k\ Topt)(1-\lambda k\ Topt) = 1-[\exp(\lambda F)-1]\ k.\exp(-\lambda k) \qquad \text{(III.4)}$$

Le coût pour le troisième modèle est le nombre de requêtes qui arrivent au système et le trouvent dans un état disponible. Un groupe d'équation est déduit, et le temps optimal entre les points de reprise est calculé dans [CHAN 75].

III.5.3 MODELES DE GELENBE

Des modèles possédant des différences essentielles des modèles précédents sont présentés dans [GELE 77] et [GELE 78]. Pour Gelenbe, le SGBD est un serveur qui présente un service à des clients (les requêtes) dans l'ordre de leur arrivée.

Ces clients arrivent au système suivant une loi de Poisson, et le temps de leur service est exponentiel. Une maintenance régulière (point de reprise) est effectuée au serveur à des temps prédéterminés.

Le serveur tombe en panne indépendamment de l'arrivée des requêtes, du service et de la maintenance, suivant une loi de Poisson. Le temps de redémarrage du système est fonction de l'âge de la panne qui est le temps entre le point de reprise le plus récent et l'occurrence de la panne.

Ces modèles ne négligent pas les requêtes arrivant au système durant les points de reprise et son redémarrage mais retardent leur exécution.

Une expression de la disponibilité du système est donnée dans [GELE 77] par l'équation (III.5). C'est la probabilité stationnaire Π_o que le système est disponible pour le service des clients.

$$\Pi_o = \{1 + \frac{Ec}{Ey} + \frac{\gamma}{Ey} \int_o^\infty h(y)\,[1-F(y)]\,dy\}^{-1} \qquad \text{(III.5)}$$

où Ec est le temps moyen pour créer un point de reprise.

Ey est le temps moyen de fonctionnement normal entre deux points de reprise.

h(y) est le temps de redémarrage du système si une panne est détectée après un temps y de fonctionnement normal

$$h(y) = \beta + \alpha y$$

où β est le temps de recopie de la BD sur la mémoire centrale.

F(y) est la répartition du temps y de fonctionnement normal entre deux points de reprise.

Le temps optimal "â" de fonctionnement normal entre deux points de reprise est obtenu en maximisant la disponibilité du système.

Ce temps est :

$$\hat{a} = \frac{Ec}{1+\beta\gamma}\left[\sqrt{1+\frac{2(1+\beta\gamma)}{\rho\gamma\, K\, Ec}} - 1\right] \qquad \text{(III.6)}$$

où γ est le taux de panne.

K est la proportion de requête à reexécuter après la détection d'une panne.

$\rho = \lambda/\mu$ où λ est le taux d'arrivée des requêtes au système, μ est le taux de service du système.

L'équation (III.6) se rapproche de celle de [YOUN 73] et [CHAN 75] si $2(1+\beta\gamma)/\rho\gamma KEc >> 1$.

Le temps optimal devient

$$\hat{a} \simeq \sqrt{\frac{2Ec}{\rho\gamma K(1+\beta\gamma)}} \qquad \text{(III.7)}$$

Un cas intéressant est obtenu si

$$2(1+\beta\gamma) << \rho\gamma\, KEc \qquad \text{(III.8)}$$

Le temps optimal est

$$\hat{a} = \frac{1}{\rho\gamma K}$$

Le temps total entre deux points de reprise est

$$E\xi = \left(\frac{2Ec}{\alpha\gamma}\right)^{1/2} . (1+\gamma\beta) + Ec \qquad \text{(III.9)}$$

Ce temps inclut le temps de redémarrage du système après la détection des pannes.

Une analyse de la file d'attente du serveur est présentée dans [GELE 77].

Le modèle présenté dans [GELE 78] est fondé sur les mêmes hypothèses concernant l'arrivée et l'exécution des requêtes, le temps de service et l'occurrence des pannes. Les hypothèses suivantes sont aussi considérées :

1) Le temps de redémarrage après la détection d'une panne suit une loi exponentielle ayant comme paramètre $(\mu KT)^{-1}$.

2) Le temps d'opération normal entre deux points de reprise successifs est une variable aléatoire répartie exponentiellement et sa moyenne est T.

3) Le temps de création d'un point de reprise suit une loi exponentielle et sa moyenne est Z.

   La disponibilité A du système est donnée par l'expression suivante

$$A = (1 + \mu KT\gamma + Z/T)^{-1} \qquad \text{(III.10)}$$

   et la moyenne du temps de réponse à une requête W est :

$$W = \frac{1/\mu + A^2\{\gamma(\mu KT)^2 + Z^2/T\}}{A-\rho} \qquad \text{(III.11)}$$

où $\rho = \lambda/\mu$

La disponibilité du système est moins sensible aux variations de T que la moyenne du temps de réponse W. (Fig.III.3).

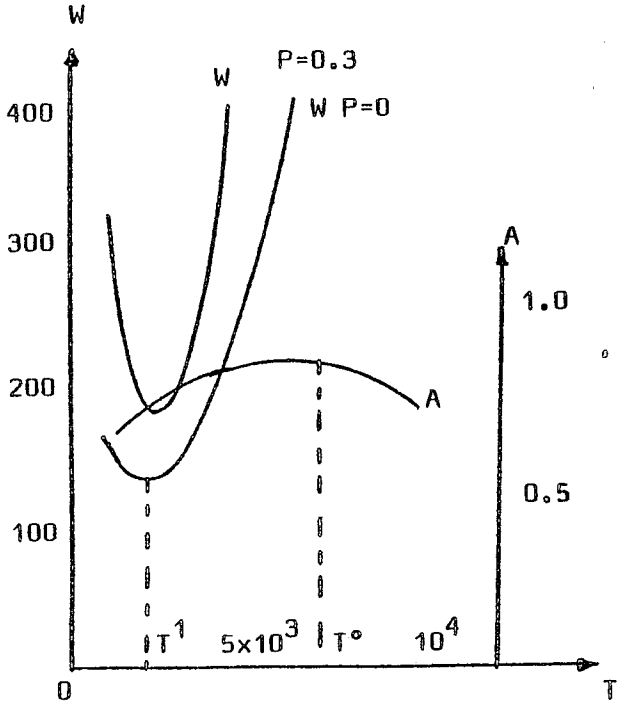


Figure III.3 : Sensibilité du temps de réponse W et la disponibilité A avec l'intervalle entre deux points de reprise T

Deux critères peuvent se présenter pour le choix de T, l'intervalle entre deux points de reprise.

Le premier critère offre un système ayant une disponibilité maximum, d'où

$$T^o = \sqrt{\frac{Z}{\gamma K \mu}} \qquad \text{(III.12)}$$

Le deuxième critère minimise la moyenne du temps de réponse aux requêtes, T^1 est approximativement donné par l'expression

$$T^1 = \left(\frac{Z^2}{2\gamma(\mu K)^2}\right)^{1/3} \qquad \text{(III.13)}$$

Selon les objectifs fixés au système, l'une des deux valeurs T^o ou T^1 peut être considérée pour initier la création des points de reprise.

Une application des deux modèles précédents est présentée dans [BOUC 78], elle offre une méthode facilitant le choix entre deux architectures qui assurent le redémarrage du système après la détection d'une panne.

Le premier système utilise une bande magnétique pour stocker une copie de la BD et de la mémoire centrale à chaque point de reprise, ainsi que les requêtes arrivant au système durant son fonctionnement normal.

Le deuxième système utilise, en plus de la bande magnétique, une unité de disque pour stocker une copie de la BD, cette copie est mise à jour parallèlement avec la BD.

Une expression de la disponibilité des deux systèmes est obtenue dans [BOUC 78]. Une comparaison entre le coût de ces deux systèmes peut être effectuée.

III.6 SGBD UTILISANT DES FICHIERS DIFFERENTIELS

III.6.1 LES FICHIERS DIFFERENTIELS ET LEURS AVANTAGES

L'organisation des BD larges et volatiles en fichiers différentiels est très efficace, ceci est dû au fait que les modifications de la BD sont enregistrées dans une zone relativement petite appelée fichier différentiel, et la BD elle-même est gardée sans modification. A des intervalles de temps prédéterminés, une mise à jour de la BD à partir du fichier différentiel est effectuée. La localisation des modifications augmente la vitesse de redémarrage du SGBD en cas de panne, et minimise la probabilité de l'occurrence d'une panne grave. En plus, la disponibilité des données augmente avec une réduction du coût de stockage et de l'accès à la BD.

Une organisation utilisant un fichier différentiel peut être la suivante : chaque enregistrement possède un identificateur unique, et le système utilise un dictionnaire pour faire l'accès à ces enregistrements. Tous les enregistrements modifiés sont stockés séquentiellement sur le fichier différentiel. Chaque nouvelle image d'un enregistrement pointe à son antécédent. Une trace de ces modifications est obtenue en construisant un index séparé pour le fichier différentiel pointant sur la copie la plus récente des enregistrements.

Les requêtes arrivées au système sont stockées séparément. A des intervalles de temps qui seront le sujet de notre étude, une fusion de la BD et du fichier différentiel est effectuée.

Si une panne se manifeste, la BD considérée comme une "image-avant-requête" et le journel des requêtes sont utilisés pour redémarrer le système et restituer le fichier différentiel.

Une copie de la BD est produite à des intervalles de temps multiple des intervalles entre les fusions.

Les avantages de cette organisation sont classés en ce qui suit :

1) Réduction du coût de "dumping" de la BD : le dumping de la BD n'est qu'une opération de mise à jour de la copie de la BD. Pour une organisation utilisant des fichiers différentiels, le dumping n'est qu'une fusion de ces fichiers avec la BD. Ceci est effectué dans un temps moins important que le dumping de toute la BD et diminuant ainsi l'overhead au SGBD.

2) Facilite un dumping différentiel.

3) Permet une fusion temps réel puisque la taille du fichier est petite.

4) Redémarrage rapide des pannes matérielles et logicielles, puisque le dumping peut s'effectuer fréquemment et le nombre de requêtes à reexécuter après la détection d'une panne sera moins important que dans le cas traditionnel.

5) Réduit le risque d'une perte sérieuse des données. En concentrant les modifications de la BD dans une petite zone, la partie exposée aux pannes est minimisée, elle peut aussi être stockée sur une unité à haute fiabilité, ce qui n'est pas faisable pour une large BD.
   Le fichier différentiel peut être dupliqué avec un coût raisonnable.

6) Puisqu'une grande partie de la BD est statique, le SGBD sera simple diminuant la probabilité de l'occurrence d'une panne logicielle.

7) La majeure partie de la BD est demandée pour une lecture seulement, et ceci élimine le risque d'occurrence d'un grand nombre de pannes.

III.6.2 LE MODELE ANALYTIQUE PROPOSE

Le modèle analytique proposé détermine le nombre optimal de requêtes arrivées au système pour initier une fusion du fichier différentiel et la BD.

Notre critère est de minimiser le coût par unité de temps de cette fusion durant la vie du SGBD. Ce coût est le rapport du coût moyen d'un intervalle entre deux fusions et la durée moyenne de cet intervalle [LOHM 77].

Le modèle est fondé sur les hypothèses suivantes :

1) La répartition des pannes suit une loi de Poisson ayant un taux moyen λ.

2) La répartition d'arrivée des requêtes de mise à jour, MAJ est constante avec le temps, avec un taux μ.

3) Les phénomènes d'arrivée des requêtes de MAJ et l'occurrence des pannes sont indépendants mutuellement.

4) La probabilité d'occurrence des pannes pendant la fusion du fichier différentiel et pendant le redémarrage du système est nulle.

5) La durée de vie du système est infinie.

6) Le temps d'exécution des requêtes est négligeable par rapport au temps des entrées-sorties.

7) La répartition des intervalles de temps entre l'arrivée des requêtes est indépendante des pannes et du redémarrage du système.

8) Chaque requête effectue une seule mise à jour au fichier. Pour analyser le cycle ou la période entre deux fusions on considère i est l'index des requêtes de MAJ arrivées au système dans une période ou un cycle.
Xi = le temps entre l'arrivée de la $(i-1)^{ème}$ et la $i^{ème}$ requête, considérant que dans cet intervalle il n'y a pas eu de panne.

$T = \sum_{i=1}^{n} Xi$ est le temps dans le cycle pris entre l'arrivée des n requêtes.

Ki = le nombre de panne pendant l'intervalle Xi

M = le nombre de panne dans le cycle

R = le temps nécessaire pour le redémarrage du système dans un cycle.

Les paramètres suivants sont utilisés dans le modèle

N = taille de la BD

λ = taux de panne

μ = taux d'arrivée des requêtes de MAJ

t_F = temps moyen de fusion d'un enregistrement du fichier différentiel sur la BD

α = le temps moyen pour restituer un enregistrement du fichier différentiel à partir de la BD, la reexécution de la requête est négligée

C_F = coût moyen de fusion d'un enregistrement du fichier differentiel sur la BD

a = coût moyen pour restituer un enregistrement du fichier différentiel à partir de la BD

s = coût de stockage d'un enregistrement du fichier differentiel par unité de temps.

La figure III.4 montre l'arrivée des requêtes, l'occurrence des pannes et la fusion avec le temps. N(t) est le nombre de MAJ arrivés au système au temps t depuis la dernière fusion.

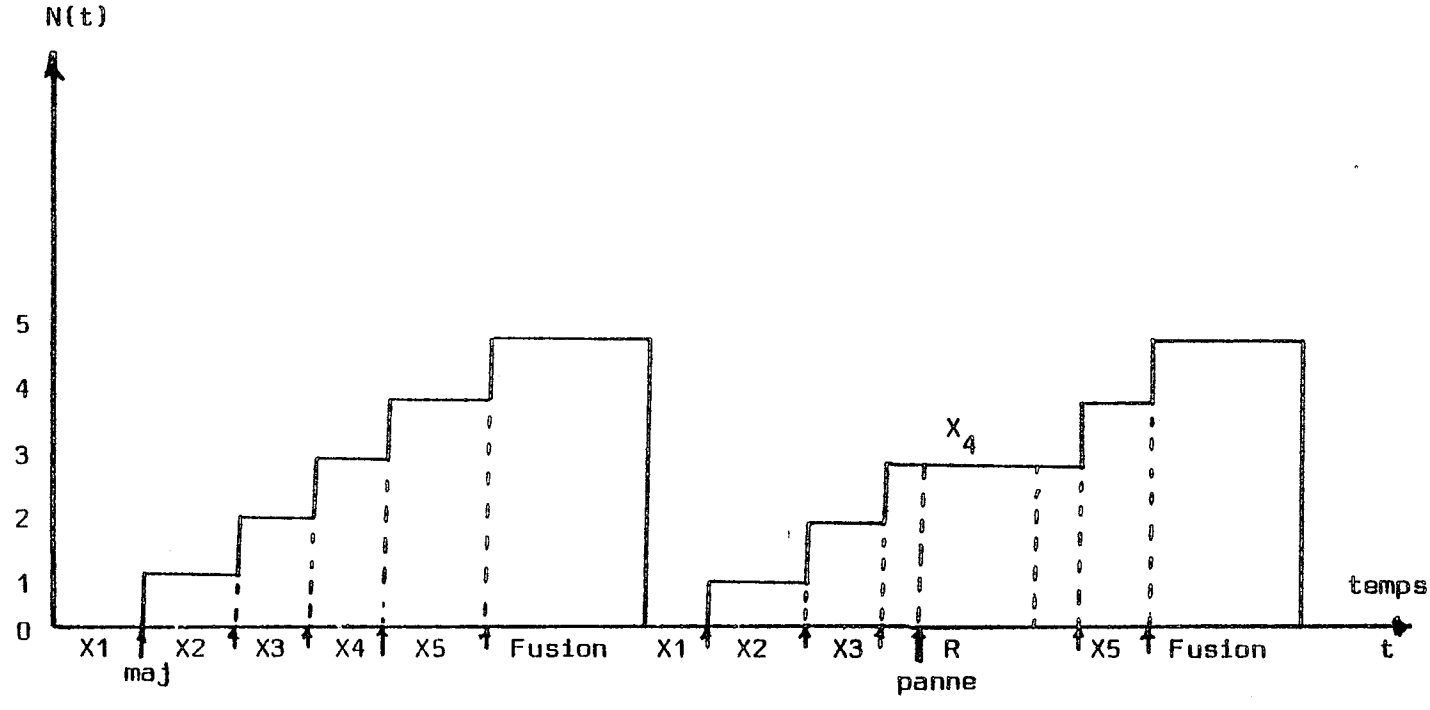


Figure III.4 : Arrivée des requêtes, occurrence des pannes

Le but de notre modèle est d'obtenir une valeur optimale de n (nombre de requêtes dans un cycle) pour minimiser le rapport C(n)/τ(n),où C(n est le coût moyen du cycle et τ(n) est la longueur moyenne du cycle.

τ(n) *La longueur moyenne du cycle*

Le temps moyen du cycle τ(n) est la somme de la moyenne du temps dûe à l'intervalle entre l'arrivée des n requêtes de MAJ par cycle, $E\{\Sigma^{n}Xi\}$, de la moyenne du temps de fusion du fichier differentiel et la BD après n MAJ, E{F/n} et de la moyenne du temps de redémarrage du système par cycle E{R/n}.

$$\tau(n) = E\{\Sigma_{i}^{n} Xi\} + E\{F/n\} + E\{R/n\} \qquad \text{(III.14)}$$

Puisque la moyenne du temps entre deux requêtes est 1/μ

$$E\{\Sigma_{i}^{n} Xi\} = n/\mu \qquad \text{(III.14.1)}$$

Pour obtenir E{F/n}, seules les copies les plus récentes des enregistrements du fichier différentiel sont fusionnées avec la BD.

Si les enregistrements de la BD sont également susceptibles d'être mis à jour par les requêtes, la moyenne du nombre d'enregistrements à fusionner après n MAJ est approximativement donnée dans [SEV 76].

$$N_F = N\left[1 - \left(\frac{N-1}{N}\right)^{n}\right]$$

$$= N\left[1-X^{n}\right] \qquad \text{où } X = \frac{N-1}{N}$$

d'où

$$E\{F/n\} = t_F\, N\left[1-X^{n}\right] \qquad \text{(III.14.2)}$$

Le temps moyen de redémarrage du système dans une période contenant n MAJ E{R/n} peut être exprimé de la manière suivante :

$$E\{R/n\} = \Sigma_{m=o}^{\infty} E\{R/M = m,n\}.\ P\{M=m/n\} \qquad \text{(III.14.3)}$$

Puisque l'arrivée des requêtes et l'occurrence des pannes sont mutuellement indépendantes :

$$P\{M=m/n\} = \int_{t=o}^{\infty} P\{M=m\}\ f(t)dt \qquad \text{(III.14.4)}$$

où P{M=m} est la probabilité d'occurrence de m pannes dans un intervalle de temps de longueur t,

f(t) est la densité de la probabilité de la longueur du cycle contenant n requête de MAJ.

$$P\{M=m\} = \frac{(\lambda t)^m e^{-\lambda t}}{m!} \qquad \text{(III.14.5)}$$

d'où

$$P\{M=m/n\} = \int_0^\infty \frac{(\lambda t)^m e^{-\lambda t}}{m!} f(t)\,dt \qquad \text{(III.14.6)}$$

Après la détection d'une panne au $i^{ème}$ intervalle, le temps de redémarrage du système après i-1 MAJ est $\alpha(i-1)$. En moyenne, si m pannes sont réparties sur les n intervalles du cycle, le nombre moyen de pannes dans chaque intervalle sera m/n.
D'où le temps moyen de rédémarrage au $i^{ème}$ intervalle est $\frac{m}{n}\alpha(i-1)$.
Et le temps moyen de redémarrage par cycle contenant m pannes et n MAJ sera :

$$E\{R/M=m,n\} = \Sigma_{i=1}^{n} \frac{m}{n} \alpha(i-1) \qquad \text{(III.14.7)}$$

En substituant (III.14.6) et (III.14.7) dans (III.14.3) on obtient :

$$E\{R/n\} = \Sigma_{m=0}^{\infty} \Sigma_{i=1}^{n} \frac{m}{n} \alpha(i-1) \int_0^\infty \frac{(\lambda t)^m e^{-\lambda t}}{m!} f(t)dt \qquad \text{(III.14.8)}$$

$$= \frac{\alpha}{n} \Sigma_{i=1}^{n} (i-1) \Sigma_{m=0}^{\infty} m \int_0^\infty \frac{(\lambda t)^m e^{-\lambda t}}{m!} f(t)dt \qquad \text{(III.14.9)}$$

$$= \frac{\alpha}{2}(n-1)\lambda \int_0^\infty t\, f(t)dt \qquad \text{(III.14.10)}$$

La quantité $\int_0^\infty t\, f(t)dt$ est le temps moyen dans le cycle dû à l'arrivée des n requêtes, elle est égale à n/μ.

$$E\{R/n\} = \frac{\alpha\lambda}{2\mu}(n-1)n \qquad \text{(III.14.11)}$$

D'où

$$\tau(n) = \frac{n}{\mu}\left[\frac{\alpha\lambda}{2} n + \left(1 - \frac{\alpha\lambda}{2}\right)\right] + Nt_f(1-X^n) \qquad \text{(III.14.12)}$$

De même pour le coût moyen du cycle durant la vie du système C(n).
C(n) = E{coût de stockage du fichier differentiel par cycle/n}, E{CS/n}.

E{coût de fusion du fichier différentiel/n}, E{CF/n}+
E{coût de redémarrage par cycle/n}, E{CR/n}, (III.15)

Le coût de stockage de i-1 enregistrements du fichier différentiel durant l'$i^{ème}$ intervalle = $s(i-1)\frac{1}{\mu}$ (III.15.1)

$$E\{CS/n\} = \Sigma_{i=1}^{n} s(i-1).\frac{1}{\mu} = \frac{s}{\mu}\, n\frac{(n-1)}{2} \qquad \text{(III.15.2)}$$

$E\{CF/n\}$ = C_f x nombre d'enregistrements du fichier différentiel à fusionner avec la BD après n MAJ.

$$E\{CF/n\} = C_F \times N\,[1-X^n] \qquad \text{(III.15.3)}$$

En procédant de la même façon que le calcul du temps on obtient :

$$E\{CR/n\} = \frac{a\lambda}{2\mu}(n-1)n \qquad \text{(III.15.4)}$$

D'où

$$C(n) = \frac{n}{2\mu}\,[(n-1)(a\lambda+S) + C_F\, N(1-X^n)] \qquad \text{(III.15.5)}$$

Le rapport $C(n)/\tau(n)$ donne le coût moyen par unité de temps durant la vie du système.

$$\frac{C(n)}{\tau(n)} = \frac{(n^2-n)(a\lambda+S) + 2\,C_F\,N\mu(1-X^n)}{\alpha\lambda n^2 + (2-\alpha\lambda)n + 2t_F N\mu(1-X^n)} \qquad \text{(III.16)}$$

La valeur optimale de n est obtenue de la différentiation :

$$\frac{d}{dn}\left[\frac{C(n)}{\tau(n)}\right] = 0$$

L'équation III.17 résulte

$$\hat{n}^2\frac{w}{\mu} + 2\,\hat{n}\,y + Z = X^{\hat{n}}\left[\hat{n}^2\, y \ln\frac{1}{X} + \hat{n}\left(2y + Z\ln\frac{1}{X}\right)+Z\right] \qquad \text{(III.17)}$$

où $w = S + a\lambda$ (III.17.1)

$y = t_f\, N\, w - \alpha\lambda\, C_F\, N$ (III.17.2)

$Z = -y - 2C_F\, N$ (III.17.3)

La valeur optimale $\hat{n}$ est l'intersection des deux courbes de l'équation (III.17). Ces deux courbes prennent la valeur Z pour n=0. Elles ont une même tangeante qui est égale à 2y à l'origine. La courbe de droite a pour assymptote l'axe horizontale et l'autre augmente jusqu'à l'infini.

Pour que ces deux courbes se coupent, une condition nécessaire et suffisante sera imposée sur les paramètres de l'équation (III.17); la dérivée seconde de la courbe de gauche $F_1(n)$ doit être inférieure à la dérivée seconde de la courbe de droite $F_2(n)$ à l'origine

$$F_1''(0) < F_2''(0) \qquad \text{(III.18)}$$

d'où

$$\frac{2w}{\mu} < -1_n \frac{1}{X} \{Z \, 1_n \frac{1}{X} + 2y\}$$

En substituant w, y et z, on obtient la condition d'optimisation

$$\mu > \frac{S + a\lambda}{N \ln \frac{1}{X} \left[C_F \ln \frac{1}{X} + (1 - \frac{1}{2} \ln \frac{1}{X})[\lambda(\alpha C_F - t_F a) - t_F S]\right]} \qquad \text{(III.19)}$$

Un cas spécial est considéré, où le coût est proportionnel au temps dans le système;

$$\alpha/a = t_F/C_F$$

L'équation (III.17) prendra la forme suivante :

$$\hat{n}^2 \frac{w}{\mu} + 2\hat{n}\, y + Z = X^{\hat{n}}[\hat{n}^2 y \ln \frac{1}{X} + \hat{n}(2y + Z \ln \frac{1}{X}) + Z] \qquad \text{(III.20)}$$

où $w = S + a\lambda$

$y = t_F N_S$

$Z = -N(t_F S + 2 C_F)$

et la condition d'optimisation deviendra

$$\mu > \frac{S + a\lambda}{N \ln \frac{1}{X} \left[C_F \ln \frac{1}{X} - t_F S(1 - \frac{1}{2} \ln \frac{1}{X})\right]} \qquad \text{(III.21)}$$

Exemple

Considérons un système de BD contenant des enregistrements de 500 octets chacun. Le coût d'exécution d'une certaine tâche est proportionnel au temps de son exécution. La BD ainsi que les fichiers différentiels sont stockés sur des unités de disques IBM 3330 [LOHM 77].

Les paramètres du système sont :

$S = 16 \times 10^{-9}$ Francs/mise à jour/sec.

$a = 0.10$ Francs

α = 0.0934 sec

t_F = 0.0734 sec

C_F = 0.102 Francs.

La figure III.5 montre les zones d'optimisation pour des BD de différentes tailles contenant 10^3, 10^4 et 10^5 enregistrements. Pour les surfaces au dessous des courbes, d'autres critères de l'environnement du système doivent être considérés pour le calcul du nombre de mise à jour dans le cycle entre deux fusions.

Les familles de courbes des figures (III.6.a), (III.6.b) et (III.6.c) donnent le nombre de mise à jour qui doivent arriver au système pour initier une fusion. Ce nombre $\hat{n}$ augmente avec le taux d'arrivée des requêtes , cette augmentation est plus annoncée pour les taux faibles d'arrivée de requêtes.

Le nombre de mise à jour formant un cycle augmente avec le temps moyen entre les pannes $1/\lambda$.

Les familles de courbe des figures (III.7.a); (III.7.b) et (III.7.c) donnent le temps optimal entre deux fusions.

Ce temps n'est plus proportionnel à la racine carrée du temps moyen entre les pannes comme pour les modèles précédents, mais il dépend aussi de la taille de la BD et du taux d'arrivée des requêtes.

Ce temps augmente avec μ, figures III.7, tant que le nombre de requêtes accumulé dans le cycle ne nécessite pas une longue période de reexécution, mais arrivé à un certain μ, l'accumulation des requêtes est trop dense, et l'occurrence des pannes, si la longueur du cycle n'est pas diminuée, devient trop coûteuse. Ceci est illustré dans la figure III.7.c pour des BD de 10^5 enregistrements et des systèmes ayant un taux de panne de 2×10^{-6}/sec, dans la partie AB, la reexécution des requêtes est peu coûteuse, puis elle devient un facteur important dans la détermination de $\tau(n)$ dans la partie BC.

Figure III.5 : Zones d'optimisation

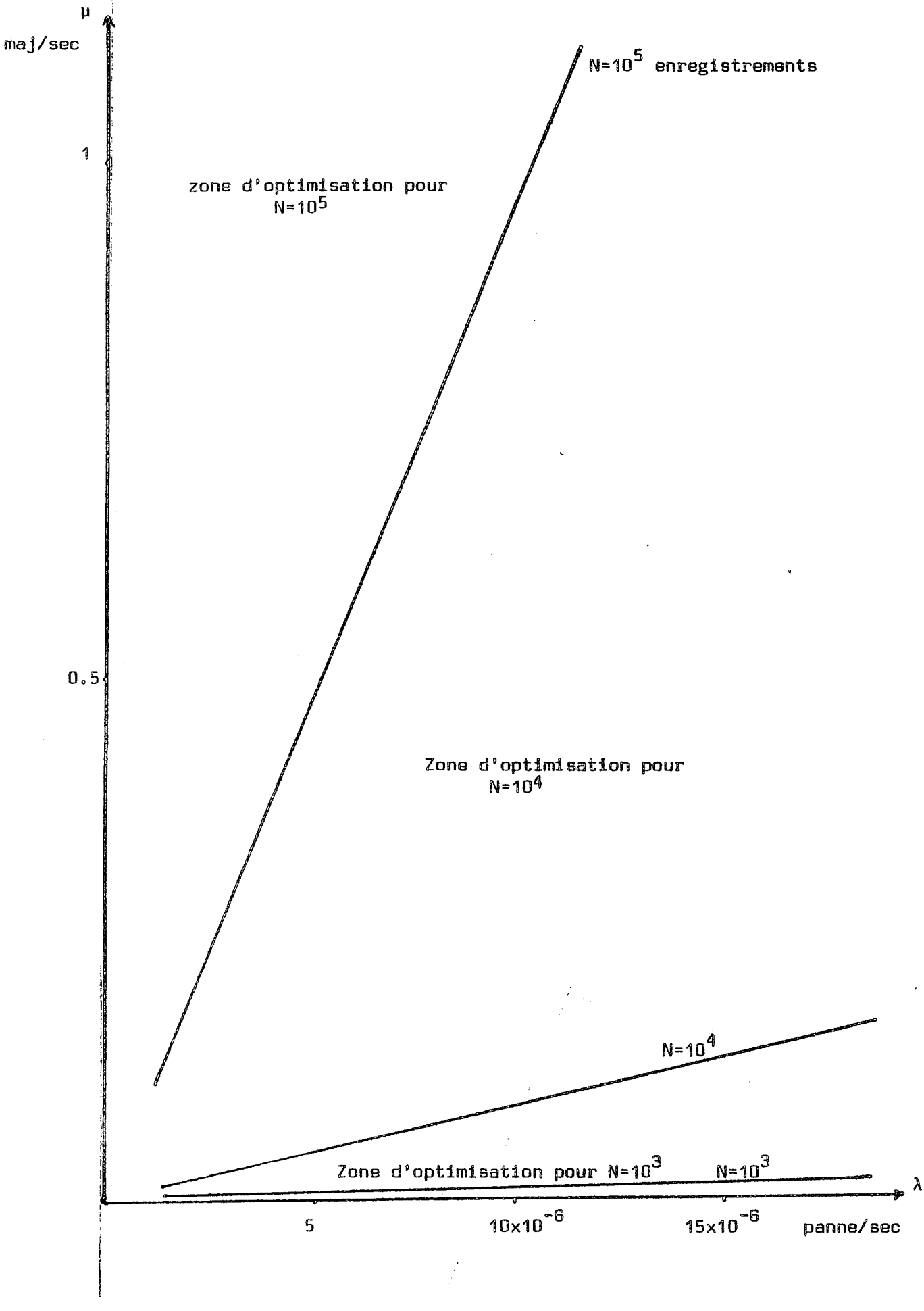


Figure III.6.a : Nombre optimal de maj/sec pour une BD de 10^3 enregistrements

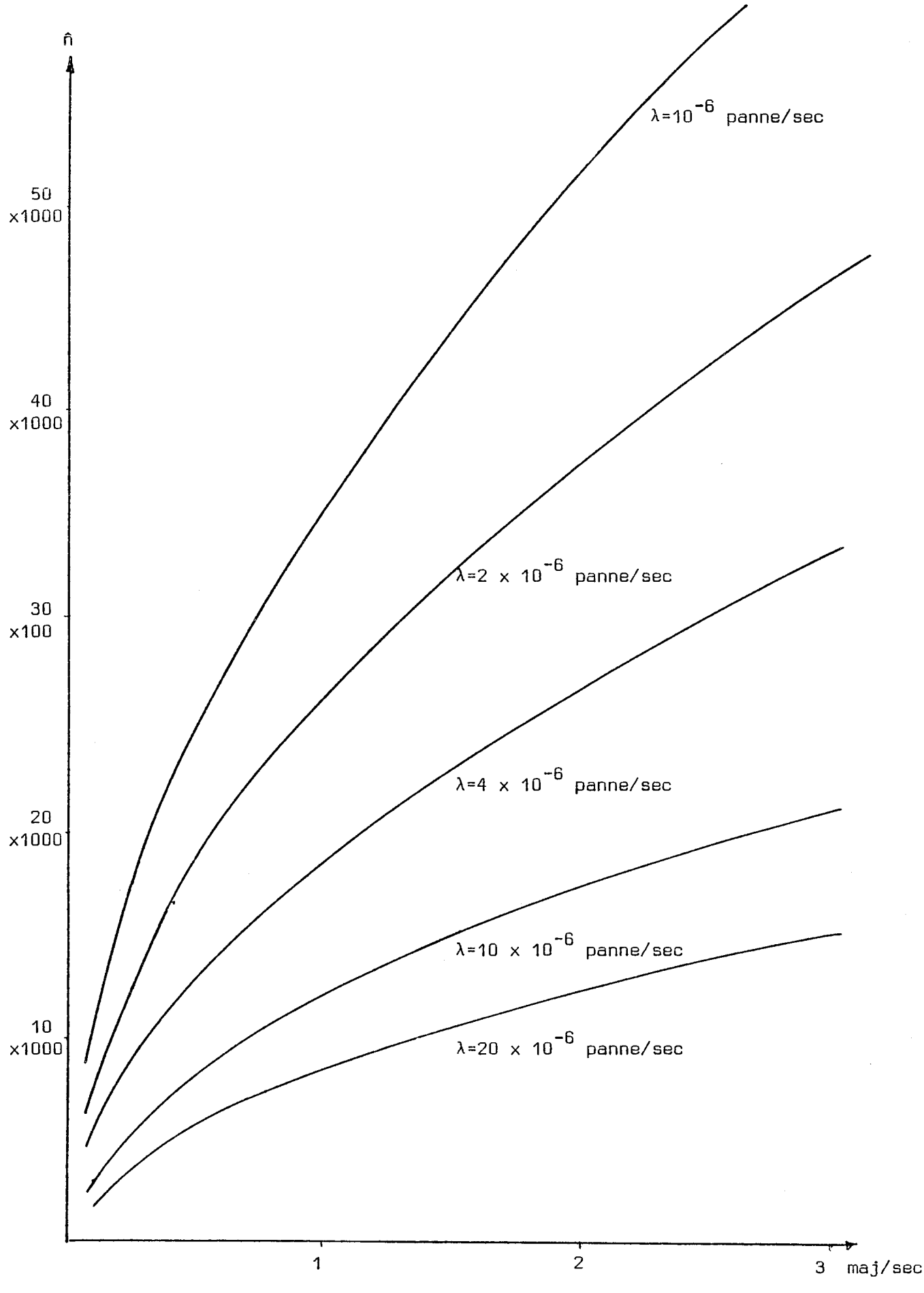


Figure III.6.b : Nombre optimal de maj/sec pour une BD de 10^4 enregistrements

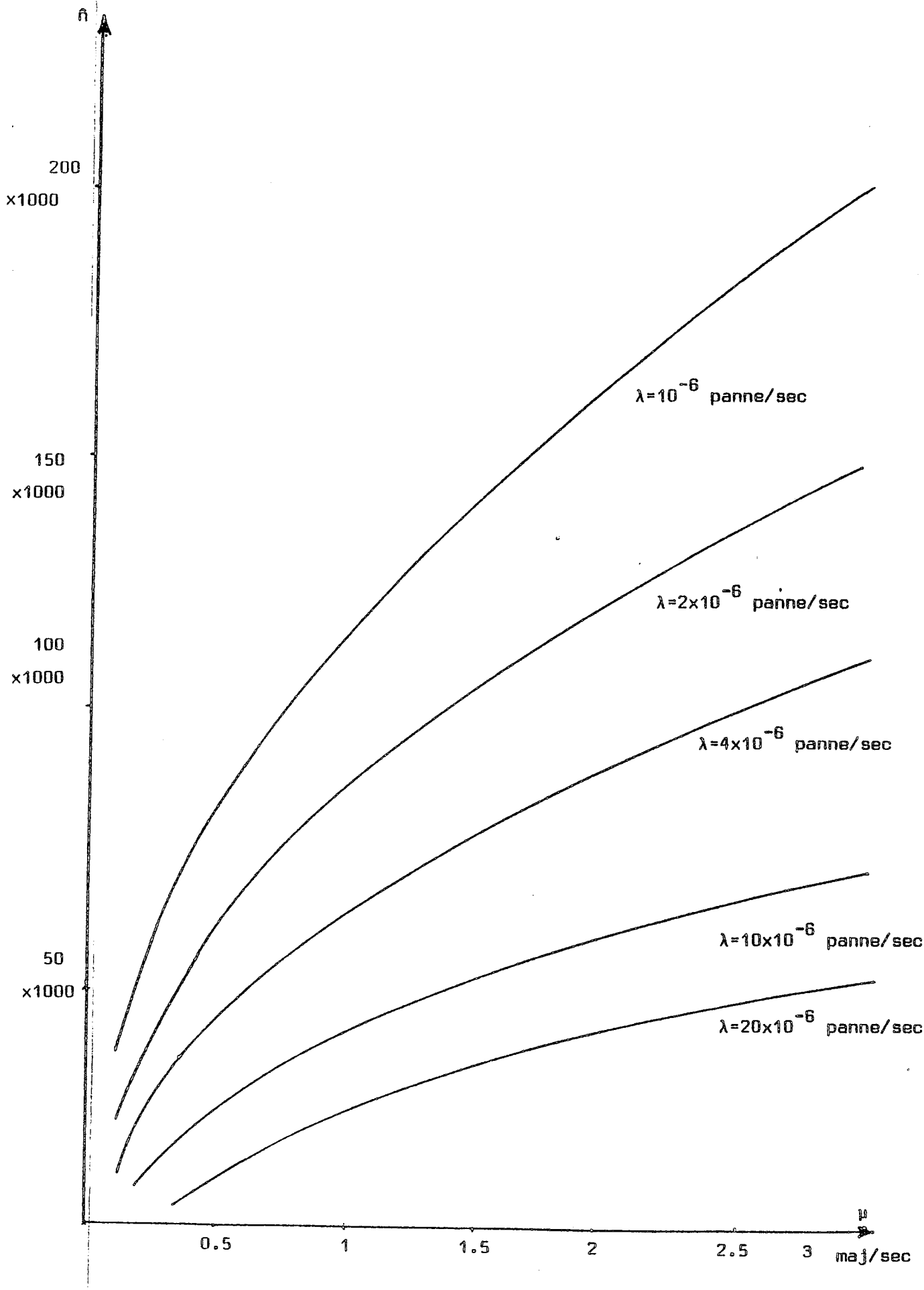
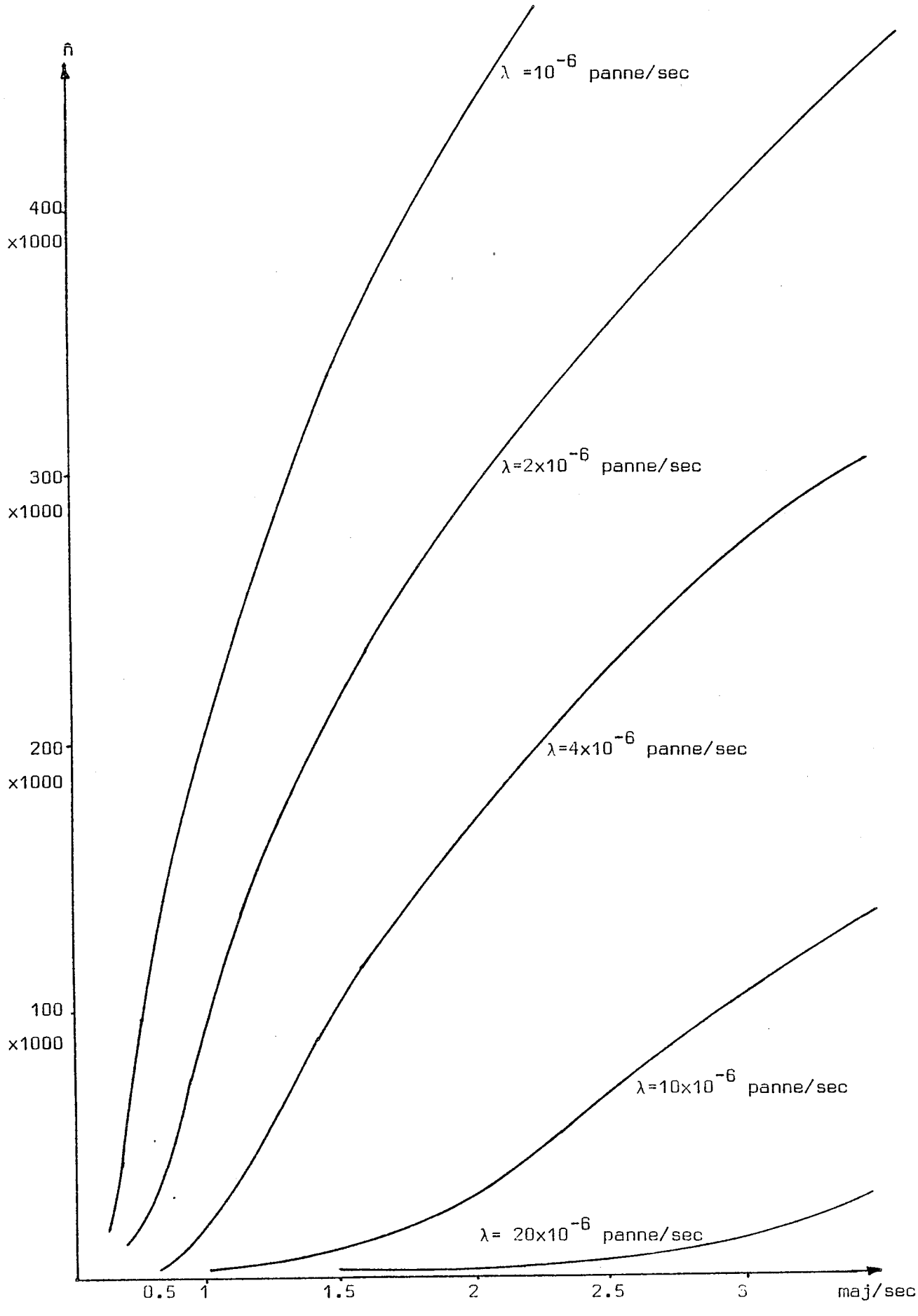


Figure III.6.c : nombre optimal de maj/cycle pour une BD de 10^5 enregistrements



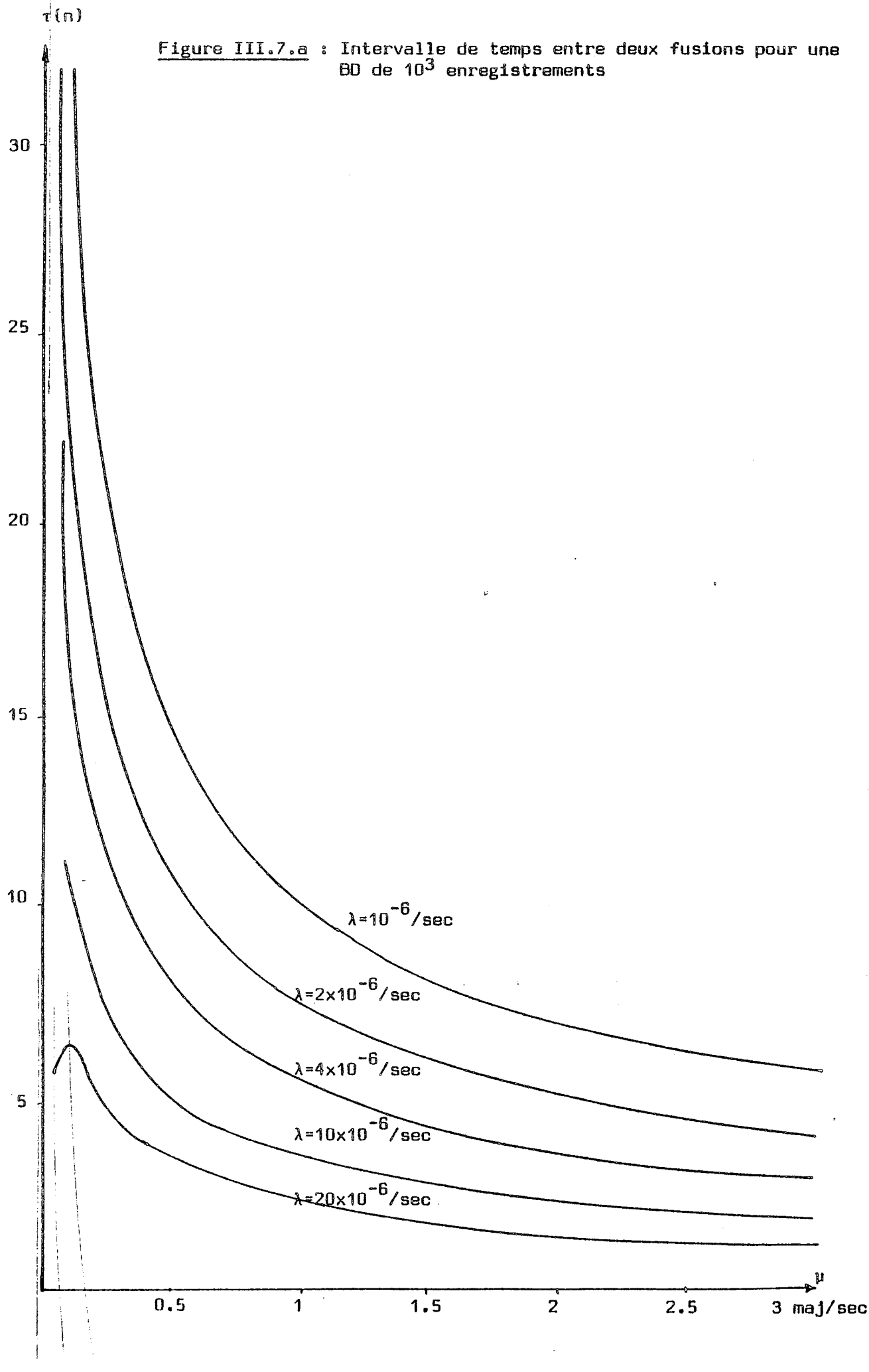


Figure III.7.a : Intervalle de temps entre deux fusions pour une BD de 10^3 enregistrements

Figure III.7.b : Intervalle de temps entre deux fusions pour une BD de 10^4 enregistrements

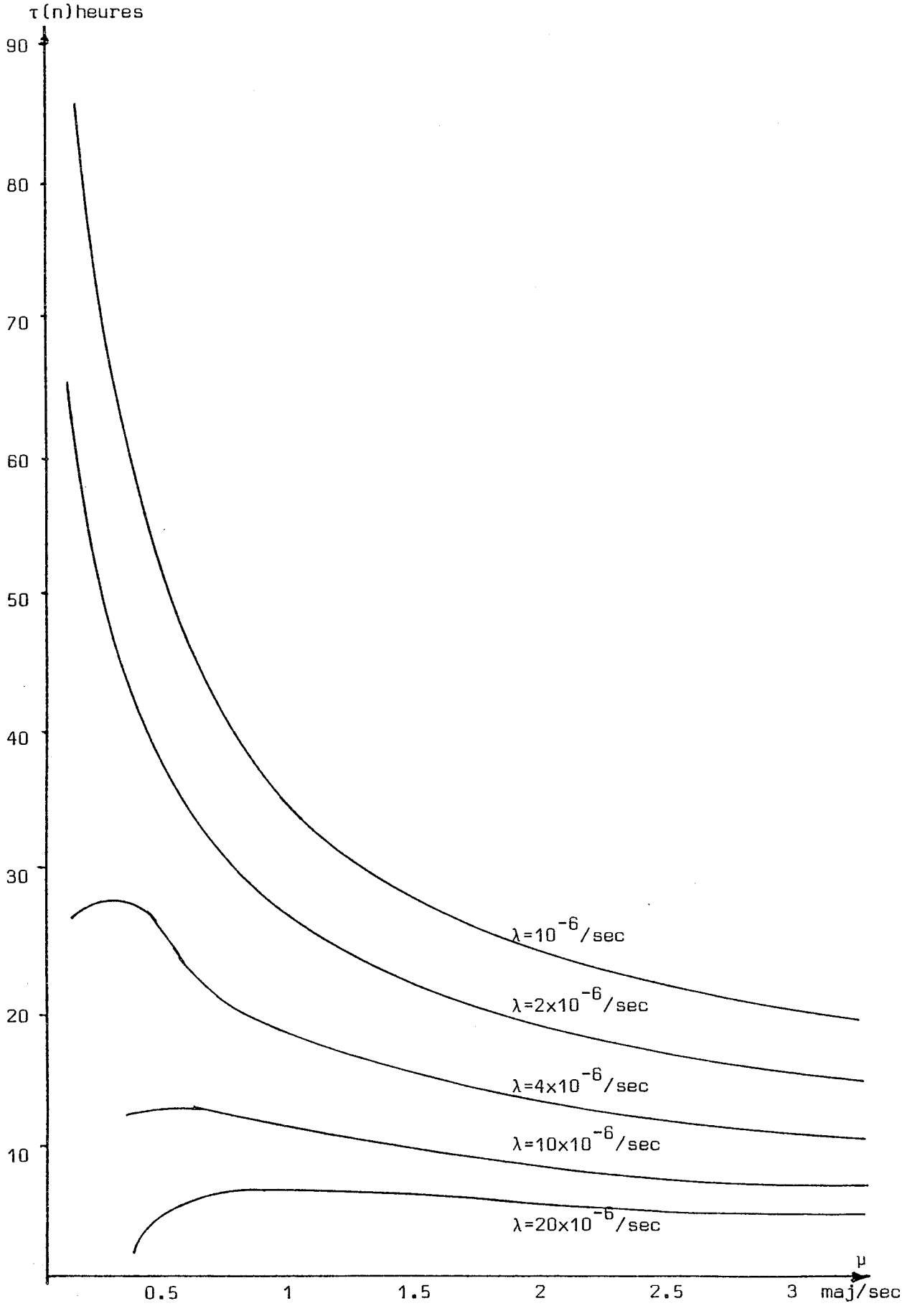
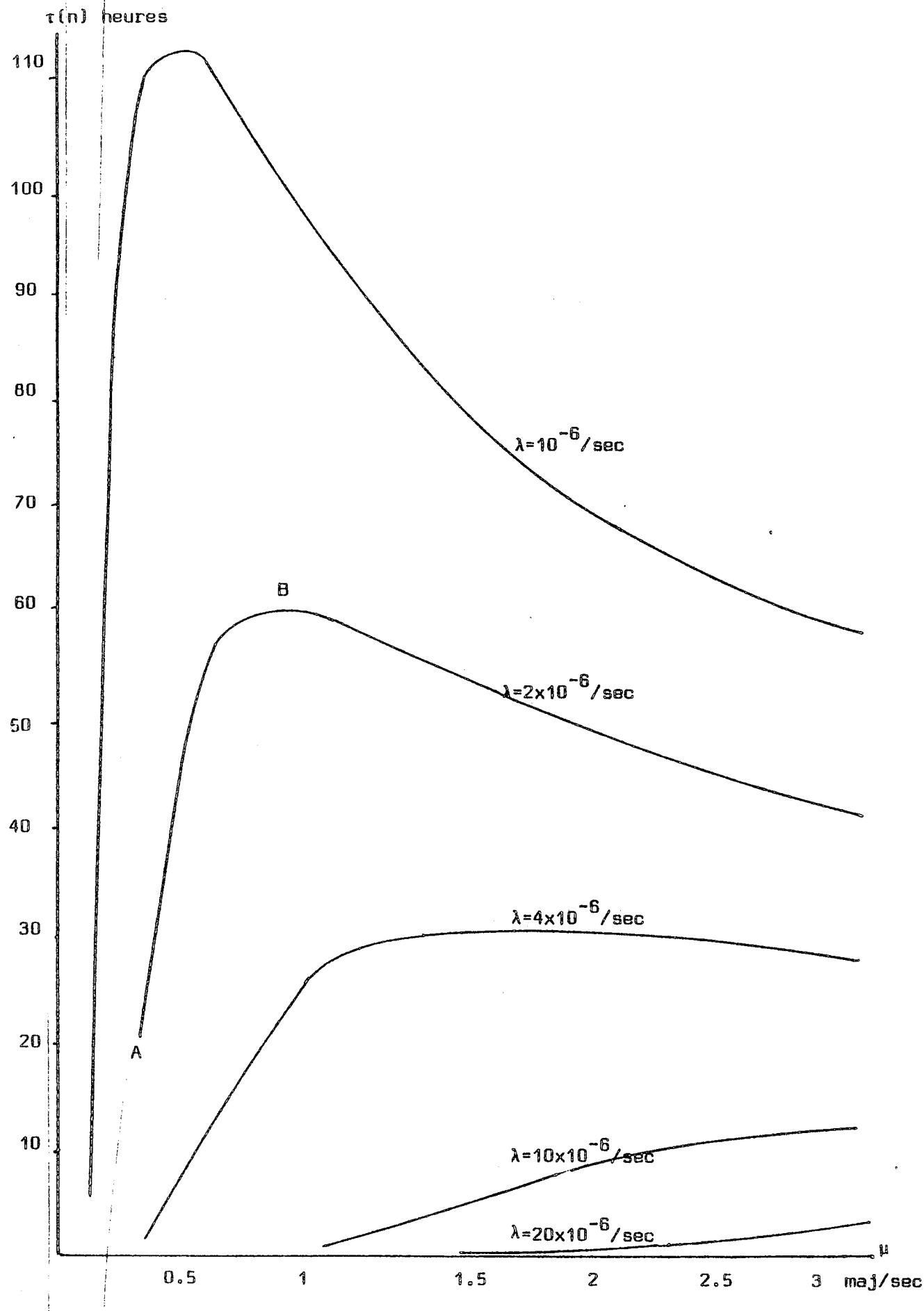


Figure III.7.c : Intervalle de temps entre deux fusions pour une BD de 10^5 enregistrements



III.7 CONCLUSION

Le modèle proposé est appliqué aux SGBD utilisant les fichiers différentiels. Il a permis de préciser n le nombre optimal de requêtes de MAJ qui doivent arriver au système pour initialiser une fusion du fichier differentiel et la BD. Les auteurs de l'article [TURN 63] avaient recommandé que la fusion du fichier différentiel et la BD puisse être initialiser quand la moitié de la BD était mise à jour. Cette proposition n'était pas basée sur une étude analytique, mais plutôt basée sur l'expérience.

Mais, notre modèle montre que le coût du système par unité de temps est minimum à des valeurs de n beaucoup plus grandes que la proposition de [TURN 63].

CHAPITRE IV

CONCLUSION

-o-o-

Nous avons présenté, dans notre travail, des outils d'aide à la conception des SGBD performants, fiables et sûrs.

Avec le modèle analytique du Chapitre III un SGBD fiable et sûr est obtenu avec un coût minimum par unité de temps. Ce système utilise un fichier différentiel pour stocker les modifications à effectuer sur la BD. Le nombre optimal de requêtes de mise à jour qui doivent arriver au système pour initier une fusion entre ce fichier différentiel et la BD est calculé.

De même une bonne performance des SGBD peut être acquise en choisissant l'organisation physique des fichiers qui s'adapte le mieux aux requêtes des utilisateurs de ces systèmes.

Une simulation des SGBD avec des requêtes de complexité ayant une même répartition que les applications peut être effectuée en utilisant les modèles analytique et graphique des chapitre I et II. Le coût des différentes organisations est obtenu. Ce coût est la somme des temps de réponse aux requêtes. Ce coût est le critère de comparaison entre ces organisations.

Le modèle graphique, présenté dans le chapitre I est utile pour obtenir une bonne répartition des fichiers des grandes BD sur les différents supports physiques pour acquérir une bonne performance du système.

ANNEXE 1

Les requêtes interrogeant les BD peuvent se classifier comme ce qui suit : elles peuvent être :

A) Une condition élémentaire A, prenant la forme

nom d'une clé $\begin{matrix} > \\ < \\ = \\ \neq \end{matrix}$ valeur

par exemple la recherche des employés d'une entreprise ayant

AGE > 20

B) Disjonction des conditions élémentaires, I, formée de l'union de deux ou plusieurs conditions élémentaires ayant une même clé. ACI est défini comme étant le nombre de conditions élémentaires par disjonction

exemple AGE = 20 ou AGE = 21 , ACI = 2

C) Conjonction C, la requête est formée de l'intersection des disjonctions I, I_1 ET I_2... ET I_n, tel que chaque I interroge une clé différente. ICR est défini comme étant le nombre de disjonction dans la conjonction.

D) La disjonction des conjonctions, d. C'est l'union des conjonctions C. CD est défini comme étant le nombre de conjonction dans la disjonction.

ANNEXE 2 : Définition du Réseau de Petri Temporisé (RdPT)

Un RDPT est défini par un sextuplet $(P,T,R,M_0,\mathcal{T},\nu)$ où

- (P,T,R,M_0) est un RdP.
- $\mathcal{T}$ est un sous ensemble complètement ordonné de R, appelé base de temps,
  (R = ensemble des réels)
- ν: $P \times T \rightarrow \mathcal{T}$ telle que si
  $\nu(P,\tau_i) = \tau_j$ alors $\tau_j \geq \tau_i$.

On représente un RdPT par le RdP associé en indiquant sur chaque place P, l'application $\nu(P,\tau)$ (Fig. 2.1)

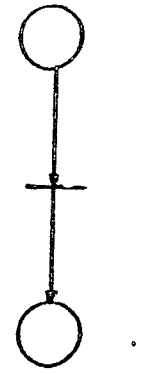


Figure 2.1 : RdPT

Règles d'évolution dans les RdPT

Une marque dans un RdPT peut se trouver dans l'un des deux états disponible ou indisponible. Initialement, toute place contient $M_0(P)$ marques disponibles.

- Une transition t_j est validée si toutes ses places d'entrée contiennent au moins une marque disponible. Toute transition validée peut être mise à feu.

- La mise à feu d'une transition validée t_j consiste à enlever une marque disponible de chaque place d'entrée et à mettre une marque indisponible dans chaque place de sortie.
  Une marque reste indisponible dans une place p durant l'intervalle de temps entre l'instant τ_0 de son arrivée à la place et l'instant $\nu(p,\tau_0)$, puis devient disponible.
  Par convention on considère que toute mise à feu est instantanée et qu'elle doit être effectuée dès la validation de la transition.

REFERENCES

ABRI 72 : J.R. ABRIAL, J.P. CAHEN, J.C. FAVRE, D. PORTAL, G. MAZARE, R. MORIN : "Project socrate", nouvelles spécifications, Grenoble, Sept. 1972.

ADIB 76 : M. ADIBA, C. DELOBEL : "Les modèles relationnels des bases de données", Grenoble, Avril 1976.

ALVA 78 : J. ALVAREZ : "CAO des systèmes à haute sécurité" Projet DEA, USMG, Sept.1978.

BELL 77 : C. BELLON : "Etude de la dégradation progressive dans les systèmes répartis", Thèse 3e cycle, Grenoble, Sept. 1977.

BELL 78 : C. BELLON, G. SAUCIER : "Test, contamination et reprise dans les systèmes distribués à haute disponibilité", Rapport de recherche, ENSIMAG, RR. n°129, Août 1978.

BOUC 78 : P. BOUCHET, E. GELENBE, S. TUCCI : "Le choix d'une architecture permettant la reprise sur panne dans un système de gestion de données", RAIRO, Informatique/computer science, vol.12, n°3, 1978.

CARD 73 : A. CARDENAS, "Evaluation and selection of file organisation - a model and system", CACM vol.16, n° 9, Sept. 1973.

CARD 75 : A. CARDENAS : "Analysis and performance of inverted data base structures", CACM, vol.18, n°5, May 1975.

CHANG 75 : K.M. CHANDY, J.C. BROWNE, C.W. DISSLY, W.R. UHRIC : "Analytic models for rollback and recovery strategies in DB systems", IEEE Trans. on software engineering vol. Se1, n°1, March 1975.

COFF 73 : E.G. COFFMAN, P.J. DENNING : "Operating systems theory", Prentice Hall, 1973.

DATE 77 : C.J. DATE : "An introduction to data base systems", Addison Wesley, Massachusetts 1977.

DAVE 76 : R.A. DAVENPORT : "Data base integrity", Computer Journal, Vol.19, n° 2, May 1976.

FOSS 74 : B.M. FOSSUM : "Data base integrity as provided for by a particular DBMS", IFIP 74, Northholland.

FRAZ 69 : A.G. FRAZER : "Integrity of a mass storage filing system", Computer Journal, Vol.12, n° 1, February 1969.

GELE 75 : E. GELENBE, J. LENFANT, D. POTIER : "Response time of a fixed-head disk to transfer of variable length", SIAM, J. Comput. Vol.4, n° 4, Décember 1975.

GELE 77-A : E. GELENBE, R. YASNOGORODSKY : "A queue with server of walking type", Rapport de recherche n° 227, Laboratoire de recherche en informatique et automatique, IRIA, Avril 1977.

GELE 77.B : E. GELENBE : "On the optimum checkpoint interval", Rapport de recherche n0 232, Laboratoire de recherche en informatique et automatique, IRIA, May 1977.

GELE 78 : E. GELENBE, D. DEROCHETTE : "Performance of Rollback recovery systems under intermittent failures", CACM, Vol.21, n° 6, June 1978.

IBMC 74 : IBM Corp. : "Introduction to IBM system/360, Direct access storage devices and organisation methods", GC 20-1649-8 IBM Corp. White plains N.Y., February 1974.

KATZ 73 : H. KATZAN : "Computer data security", Van Nostrand Reinhold Company, 1973.

KNUT 73 : D.E. KNUTH : "The art of computer programming, Vol.3, Sorting and searching" Addison Wesley, Reading Massachusetts 1973.

LEFK 69 : D. LEFKOVITZ : "File structure for on line systems", Spartan Books, 1969.

LOHM 77 : G.M. LOHMAN, J.A. MUCKSTADT : "Optimal policy for batch operations : Backup, checkpointing, Reorganisation and updating", ACM Trans on DB systems, Vol.2, n° 3, Sept. 1977.

MART 73 : J. MARTIN : "Security, Accuracy and privacy in Computers Systems", Prentice Hall, Englewood cliffs, N.Y. 1973.

MART 75 : J. MARTIN : "Computer data base organisation" Prentice-Hall, Englewood Cliffs 1975.

MEND 64 : MENDELSON : "Introduction to mathematical logic", Von Nostrand, N.Y. 1964.

MOAL 76.A : M.MOALLA, G. SAUCIER, J. SIFAKIS, M. ZACHARIADES : "A design tool for the multilevel description and simulation of systems of interconnected modules", 3rd annual symp. architecture, Tanapa, Jan.1976.

MOAL 76.B : M. MOALLA : "L'approche fonctionnelle dans la vérification des systèmes informatiques, proposition d'un ensemble de méthodologies", Thèse de Docteur-Ingénieur, INPG, Grenoble, Dec.1976.

RICH 78 : H. RICHY : "Confidentialité, Bases de données et réseaux d'ordinateurs", Thèse de Docteur-Ingénieur, Grenoble 1978.

ROBA 75 : Ch. ROBACH : "Méthodologie de test de processeurs, impacts sur la conception", Thèse de Docteur-Ingénieur, Univ. de Grenoble, Mai 1975.

ROBA 76 : Ch. ROBACH, G. SAUCIER, J. LEBRUN : "Processor testability and design consequences", IEEE trans on comp. June 1976.

ROBA 78 : Ch. ROBACH, G. SAUCIER : "Dynamic testing of control units", IEEE Trans. on computers", juillet 1978.

SEVE 76 : D.G. SEVERANCE and G.M. LOHMAN : "Differential files : their application to the maintenance of large data bases", ACM trans. on DB systems, Vol.1, n° 3, Sept. 1976.

SILE 76 : K. SILER : "A stochastic evaluation model for DB organizations in data retrieval systems", CACM, Vol.19, n° 2, February 1976.

SOUL 76 : G. SOULA : "Sûreté de fonctionnement des systèmes informatiques, la reconfiguration automatique avec dégradation progressive", Thèse, Univer. Paul Sabatier, Toulouse 1976.

TURN 63 : V.P. TURNBURKE : "Sequential date processing design" IBM system Journal, 2, March 1963.

VALE 76 : VALETTE : "Sur la description, l'analyse et la validation des systèmes de commandes parallèles", Thèse d'Etat, Univ. Paul-Sabatier, Toulouse 1976.

WILK 72 : M.V. WILKES : "On preserving the integrity of DB", Computer Journal, Vol.15, n° 3, August 1972.

YAO 77.A : S.B. YAO : "Approximating block accesses in DB organizations", CACM, Vol.20, n° 4, April 1977.

YAO 77.B : S.B. YAO : "An attribute based model for DB access cost analysis", ACM trans. on D.B. systems, Vol.2, n° 1, March 1977.

YOUN 73 : J.W. YOUNG : "A first order approximation to the optimum checkpoint interval", CACM Vol.17, n°9, Sept.1973.

YOUR 72 : E. YOURDON : "Design of on line computer systems", Prentice Hall, Englewood Cliffs, N.Y. 1972.

ZACH 77 : M. ZACHARIADES : "MAS : Réalisation d'un Langage d'Aide à la Description et à la Conception des Systèmes Logiques", Thèse 3e cycle, INPG, Grenoble, Sept.1977.

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de :

- Madame G. SAUCIER, Maître de Conférences à l'Institut National Polytechnique de GRENOBLE.

- Monsieur E. GELENBE, Maître de Conférences à l'Université Paris-Sud -ORSAY-

Madame ABDEL HAY ABDEL HAMID MOUSTAFA épouse TAHER Soheir

est autorisée à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Grenoble, le 5 Janvier 1979

Le Président de l'I.N.P.G.

Ph. TRAYNARD
Président
de l'Institut National Polytechnique
P.O. le Vice-Président,