Le SGBD relationnel pour microordinateur Pepin:un outil pour l'evaluation de performance

P. Bouchet, Alain Chesnais, J.M. Feuvre, G. Jomier

INRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P 105
78153 Le Chesnay Cedex
France
Tél. 954 90 20

Rapports de Recherche

N° 125

LE SGBD RELATIONNEL POUR MICROORDINATEUR *PEPIN* : UN OUTIL POUR L'ÉVALUATION DE PERFORMANCE

Pierre BOUCHET
Alain CHESNAIS
Jean-Michel FEUVRE
Geneviève JOMIER

Avril 1982

LE SGBD RELATIONNEL POUR MICROORDINATEURS PEPIN : UN OUTIL POUR L'EVALUATION DE PERFORMANCE

THE RELATIONAL DBMS FOR MICROCOMPUTERS PEPIN : A TOOL FOR PERFORMANCE EVALUATION

Pierre BOUCHET (Univ. Paris-Nord)
Alain CHESNAIS (Univ. Paris-Sud)
Jean-Michel FEUVRE (INRIA)
Geneviève JOMIER (INRIA)

INRIA
BP. 105
78153 Le Chesnay Cédex
Tél. 954.90.20

Résumé

Le SGBD relationnel pour microordinateurs PEPIN a été développé à l'INRIA dans le but d'obtenir une maquette pour l'évaluation de performance des SGBD centralisés et répartis.

Ce papier présente d'abord les principales caractéristiques du système actuellement opérationnel, puis un ensemble de mesures sur le système, destinées à servir de base à des évaluations de performance.

Abstract

The Relational DBMS for microcomputers PEPIN has been developped at INRIA. The aim is to design an experimental tool to study performances of centralised and distributed DBMS.

In this paper, we first present the main characteristics of the system which is now running. Then we present some measurements on the system, which should be used in future performance evaluation.

mots clés : système de gestion de bases de données relationnel, microordinateur, performance, évaluation.

key words : Data Base Management system, Relational, Microcomputer, Performance, Evaluation.

INTRODUCTION

Le système de gestion de base de données relationnelles PEPIN a été développé par le projet "Modélisation et Evaluation de Performance de Systèmes Informatiques et Téléinformatiques" de l'INRIA afin de disposer d'une maquette expérimentale pour l'évaluation de performance .

En effet l'évolution de la technologie, et en particulier la diffusion importante des microordinateurs conduit à une forte demande de systèmes de gestion de base de données (SGBD) [Ever.81]. D'autre part les SGBD sont au coeur des nouveaux systèmes qui se développent actuellement [Tsic.81] : burotiques [Giff.80], archivage, messagerie, systèmes d'information, systèmes experts etc. On leur demande de nombreuses qualités de fiabilité, sécurité, résistance aux pannes, rapidité d'exécution des requêtes, etc.

Ceci explique l'intérêt de l'évaluation de leurs performances. Cependant la grande diversité des SGBD existants, tant dans leur conception que dans les services qu'ils offrent [Ever.81, LoMa.81] limite les possibilités et l'intérêt de leur modélisation globale [Sevc.81] à des fins d'évaluation quantitative. Ainsi la plupart des travaux se sont-ils attachés à des aspects particuliers qui ont été l'objet le plus souvent de modélisation analytique et/ou de simulation. L'accent a été mis essentiellement sur le contrôle de la concurrence [ChGM.81], [PoLe.80], [Ries.81] [RiSt.77], [ShSp.81], les performances liées à l'organisation des données [GoVe.80] ou à différents chemins d'accès aux données [AoKS.80]. Dans le cas du modèle relationnel le problème de l'optimisation des questions est d'une grande importance sur les performances du SGBD [Kim.81] et motive des travaux sur l'évaluation de la taille de résultats de requête, [Rich.81], [GeGa.82]. D'autres travaux d'évaluation théorique ont été développés dans le cadre des SGBD répartis, l'intérêt étant porté essentiellement sur la comparaison d'algorithmes de maintien de la cohérence [Garc.79], [Wilm.79], ou l'évaluation du nombre de copies des données [CoPG.80].

Des mesures sur des systèmes de gestion de base de données ont été publiées [HeWY.81 , HaSt.79, LoMa.81]. Cependant beaucoup reste à faire dans ce domaine. En effet des résultats de mesure sont intéressants s'ils permettent de tirer des conclusions dépassant le cadre strict de l'expérience qui les a produits, c'est-à-dire lorsqu'on peut les utiliser comme paramètres d'un modèle, ce qui va permettre en particulier des comparaisons. De telles données, tant sur le comportement du logiciel de SGBD que sur sa charge (caractérisation des données et des requêtes) font défaut. Cette situation est particulièrement criante pour les SGBD sur microordinateurs, [Mira.82] pour lesquels on ne dispose d'aucune mesure utilisable.

Face à cette situation il nous a semblé interessant de développer une maquette expérimentale pouvant servir l'évaluation de performance, c'est-à-dire offrant la possibilité de mesurer les paramètres que l'on veut connaître, et suffisamment souple pour pouvoir faire varier les choix d'implantation. Cette possibilité de variété offerte par une écriture très modulaire et en couches du logiciel est envisagée tant du point de vue logiciel (mise en oeuvre de différents choix d'algorithmes de gestion de la concurrence) que du point de vue matériel : en effet PEPIN, quoiqu'existant en

version monomachine, a été conçu pour une implantation répartie multi-microordinateurs.

Ce papier a donc un double but, présenter PEPIN, c'est l'objet de la première partie, et ensuite présenter un certain nombre de résultats des mesures prélevées dans la version actuelle pour l'ensemble des opérations fondamentales sur les nuplets. Ces deux parties sont nécessairement liées puisque la première permet l'interprétation de la seconde.

I. LE SYSTEME DE GESTION DE BASES DE DONNEES

Dans cette section nous présentons le SGBD PEPIN-2 indépendamment de son implémentation. PEPIN-2 est issu après refonte et réécriture complètes de PEPIN-Ø et PEPIN-1 présentés dans [BCFJK.81a] et [BCFJK.81b]. Ses caractéristiques essentielles peuvent se résumer par les termes suivants :

- relationnel
- orienté vers le mode conversationnel
- portable sur la plupart des mini et micro ordinateurs (écrit en PASCAL UCSD)
- conçu pour environnement multimicroordinateur.

La configuration globale du logiciel du SGBD est représentée à la figure 1.

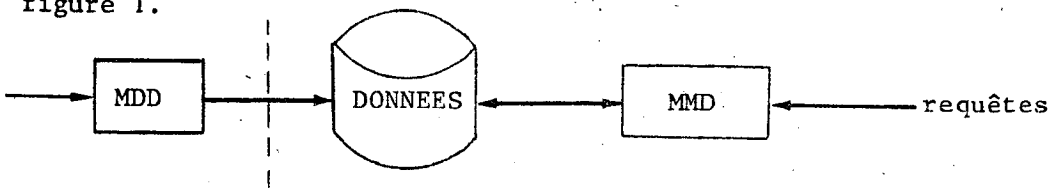

Figure 1

Dans ce diagramme deux parties jouent des rôles distincts. La première, exécutée à la création d'une nouvelle base de données et formée des modules de définition des données (MDD), permet de définir les relations qui composent la base et d'initialiser les catalogues permettant la manipulation ultérieure de ces relations. La deuxième, active pendant toute la vie de la base, formée des modules de manipulation des données (MMD), permet de manipuler les relations de la base.

Les Modules de Définition des Données

Cet ensemble de modules établit la configuration initiale de la base pour l'utilisateur, prépare la mémoire secondaire et initialise les catalogues qui permettent d'accéder aux données.

Une base de données est un ensemble de relations. Chaque relation est composée d'un certain nombre d'attributs. Un attribut prend ses valeurs dans un ensemble appelé son domaine et peut être déclaré clé primaire.

Un domaine peut être soit un domaine système (entier, réel, chaine de caractères de longueur fixe ou variable), soit défini explicitement par l'utilisateur (liste de valeurs, intervalle).

L'utilisateur peut classer une relation ou définir un nombre arbitraire d'index sur un ou plusieurs attributs. Les index sont traités par le système comme des relations classées.

Une fois ces informations recueillies le système réserve de l'espace secondaire en fonction du volume prévu des données à stocker. Les catalogues et le schéma sont ensuite initialisés.

Ces opérations faites nous pouvons commencer à insérer des valeurs dans la base et les manipuler. Regardons brièvement la structure des données sousjacentes.

Structure des Données

La totalité de la mémoire secondaire est composée de pages. Une page est l'unité de transfert entre mémoire centrale et mémoire secondaire. Nous définissons six types de pages :

- les pages de nuplets qui contiennent les valeurs des nuplets de la base de données
- les pages de chaine variable qui contiennent les valeurs des attributs dont le domaine est chaine de caractères de longueur variable.
- les pages de catalogue de relation qui contiennent des pointeurs vers les pages de nuplets et les pages de chaine variable d'une relation.
- les pages de catalogue global qui contiennent des pointeurs vers les pages de catalogue de relation de toutes les relations de la base de données
- les pages de schéma contenant les informations permettant d'accéder aux nuplets de la relation. En particulier une page de schéma référence pour chaque relation la partie du catalogue global qui la concerne.
- les pages d'état qui indiquent les états des pages en mémoire secondaire et permettent de gérer l'allocation d'espace et d'assurer l'atomicité des transactions ainsi que la reprise sur pannes.

Un nuplet est formé par la concaténation des chaines d'octets représentant chaque attribut. Le codage utilisé est tel que la relation d'ordre entre deux valeurs d'un attribut se traduit par l'ordre lexicographique sur les chaines d'octets, sauf dans le cas des attributs de domaine chaine de caractères de longueur variable. En effet, pour avoir des nuplets de longueur fixe, la valeur d'un tel attribut est remplacé par un pointeur vers son adresse (adresse logique de la page dans laquelle la valeur est stockée et déplacement logique à l'intérieur de la page).

Expliquons rapidement comment l'on accède aux nuplets d'une relation avec cette structure. La figure 2 représente le principe de l'accès. Avec le nom de la relation nous trouvons l'entrée dans la page de schéma la concernant. Cette entrée contient un pointeur (PTR) vers la première entrée dans le catalogue global concernant la relation ainsi que son nombre total (NB) d'entrées. Toutes les entrées dans le catalogue global concernant une relation sont rangées consécutivement.

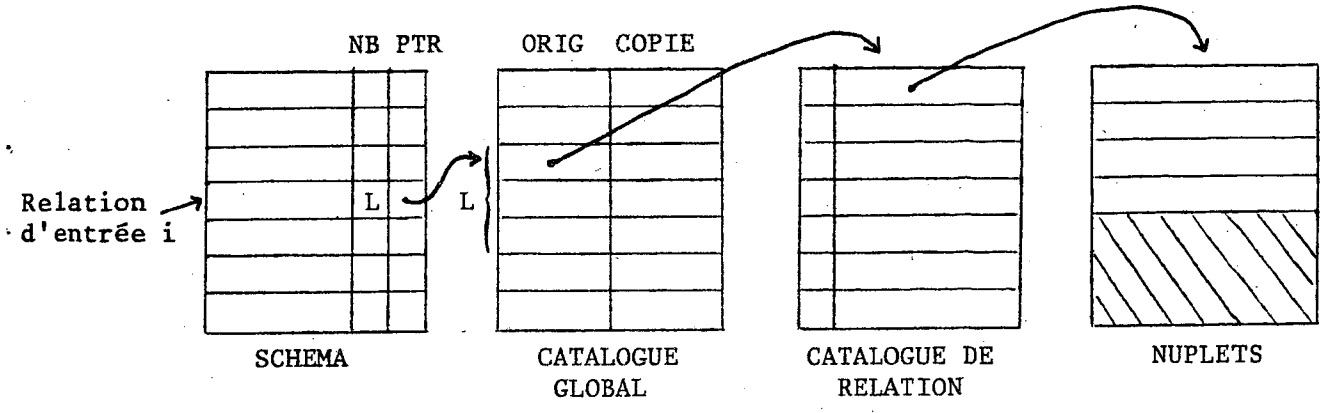

Figure 2 : Accès aux nuplets d'une relation

Chaque entrée du catalogue global est un doublet de pointeurs vers des pages de catalogue de relation (nous reviendrons sur l'utilisation du deuxième champ dans le chapitre concernant la cohérence des données). La page de catalogue de relation contient des pointeurs vers des pages de nuplets dans lesquelles sont stockées, sous forme de tableau, les nuplets de la relation. Pour permettre aux pages les plus souvent référencées (pages de schéma, de catalogue global, de catalogue de relation) de rester autant que possible en mémoire centrale nous avons implémenté un algorithme de pagination suivant le principe LRU.

Les six types de pages définies précédemment peuvent être divisés en deux classes suivant leur localisation en mémoire secondaire (voir la Figure 3) : les pages statiques qui restent à un emplacement fixe pandant toute la vie de la base (ce sont les pages de schéma, de catalogue global et d'état) et les pages dynamiques qui peuvent être successivement allouées et désallouées (ce sont les pages de catalogue de relation, de nuplets et de chaines variables).

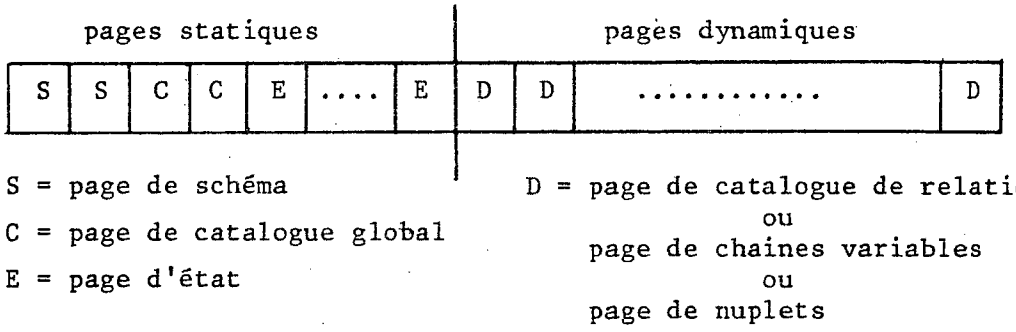

Figure 3 : Zones statique et dynamique de la mémoire secondaire

Les pages dynamiques peuvent être dans un des quatre états suivants :

- LIBRE si elle est inutilisée
- OCCUPEE si elle est utilisée et n'est pas en cours de modification
- TRANSITOIRE D'ORIGINE si c'est la version d'origine d'une page en cours de modification
- TRANSITOIRE DE COPIE si c'est la version résultante d'une page en cours de modification.

Cette notion d'état des pages dynamiques est utilisée pour assurer l'atomicité des transactions et la reprise sur pannes.

Maintien de la Cohérence et Reprise sur Pannes

Pour assurer l'intégrité des données dans le cas multi-usager, un algorithme de verrouillage au niveau de la relation a été implémenté (une modification de la structure d'adressage dans PEPIN permettant de verrouiller au niveau de la page est à l'étude). En cas de conflit on permet à la transaction de redemander le verrou refusé un certain nombre de fois, si elle ne l'obtient pas au bout du nombre permis de réessais la transaction est avortée. Cette stratégie de verrouillage a fait l'objet d'une étude théorique [ChGM.81] pour déterminer le nombre optimal de réessais permis.

Espace fantôme

Comme nous l'avons vu au paragraphe précédent, une transaction peut être avortée par le système après avoir entamé ses modifications sur la base de données. Pour nous permettre de le faire aisément et d'assurer l'atomicité des transactions, même en cas de panne du système, nous avons adopté un mécanisme de mises à jour différées [Lori.77].

Aucune modification n'est apportée directement sur la base. Quand une transaction veut modifier des données de la base elle fait une copie des données à modifier et apporte ses modifications à la copie, gardant ainsi la version d'origine des données. L'ensemble de ces copies s'appelle l'espace fantôme. Ce n'est qu'à la fin d'une transaction que ces copies sont validées et les anciennes valeurs perdues. En cas d'erreur ou d'avortement il suffit de libérer les copies et de reprendre les valeurs d'origine.

Notons que dans notre système les copies sont créées au niveau du mécanisme de pagination : pour "copier" une page résidant en mémoire centrale il suffit de changer son adresse disque. Si la page doit quitter la mémoire centrale, elle sera alors écrite à cette nouvelle adresse.

Donnons un exemple pour illustrer le fonctionnement de ce mécanisme : on supposera que la configuration initiale de la base est celle indiquée sur la Figure 4.

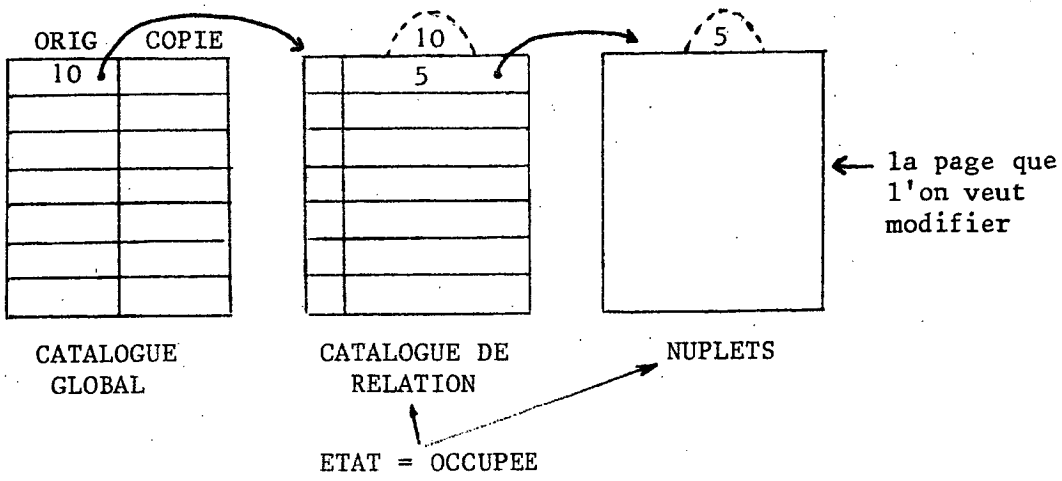

Figure 4 : Etat initial de la base

Avant d'apporter la modification dans la page 5 on cherche une page à l'état libre. Supposons que la page 6 le soit. On change alors son état pour aboutir à l'état transitoire de copie, la page 5 est mise à l'état transitoire d'origine et copiée en 6. Cette modification entraine une modification du catalogue de la relation : on cherche donc une page libre pour copier le catalogue de relation à l'adresse 10. Supposons que la page 11 soit libre. On la met à l'état transitoire de copie. La page 10 est mise à l'état transitoire d'origine et copiée en 11. On modifie alors le pointeur qui référençait la page 5 pour lui faire référencer la copie en 6 et on positionne le bit de modification pour indiquer que la page 6 est une copie (ce qui nous permet de ne pas en refaire une copie si on veut y apporter une nouvelle modification). On fait ensuite pointer le champ COPIE correspondant dans le catalogue global sur la page 11 pour aboutir à l'état illustré à la Figure 4bis.

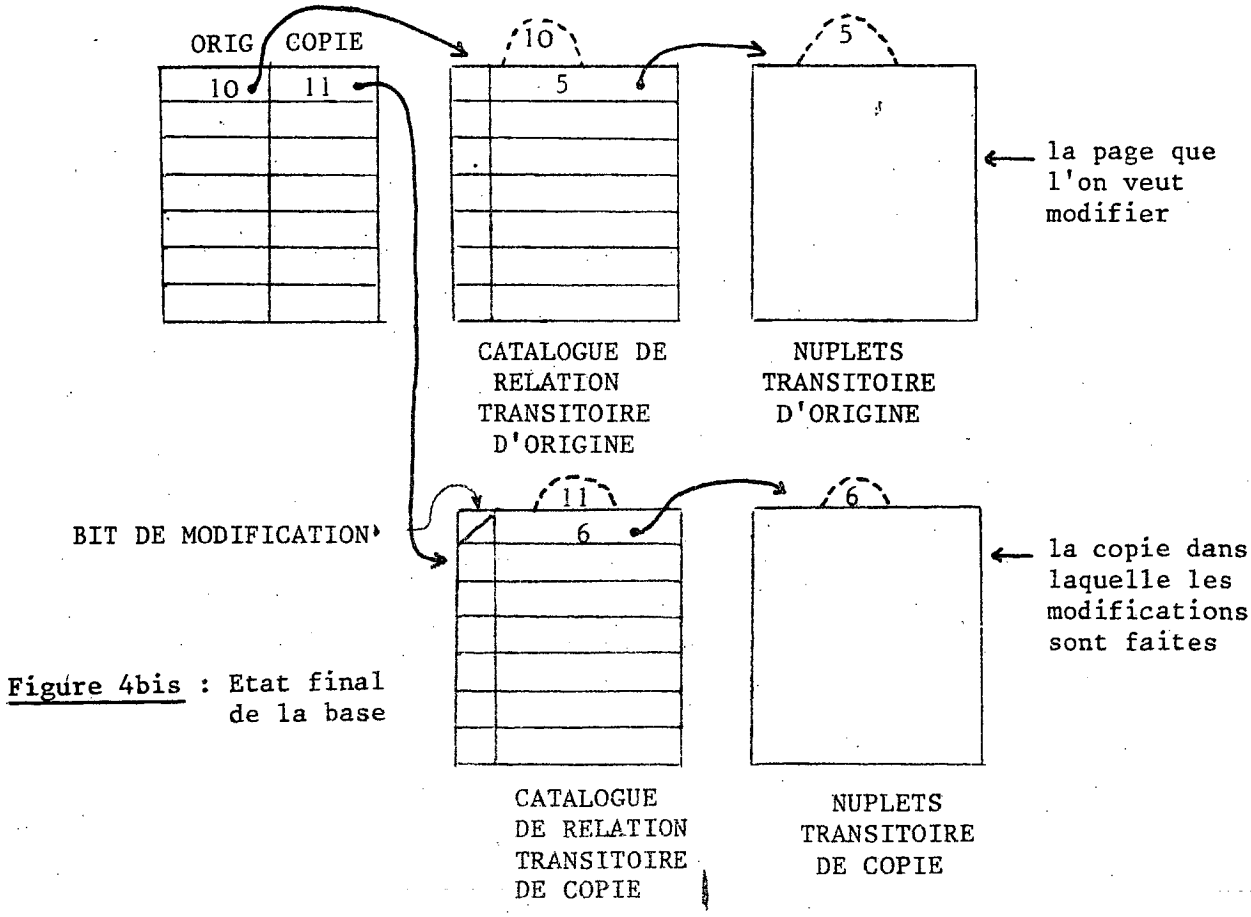

Figure 4bis : Etat final de la base

On peut maintenant apporter nos modifications dans la page 6 qui est une copie de la page 5 d'origine.

La transaction se terminera ensuite soit par une validation si tout s'est bien passé, soit par un avortement.

Avortement

Pour avorter une transaction il suffit d'effacer les champs COPIE qu'elle a rajouté dans le catalogue global et de nettoyer les états des pages modifiées en mettant à l'état LIBRE les pages à l'état TRANSITOIRE DE COPIE et à l'état OCCUPEE les pages à l'état TRANSITOIRE D'ORIGINE.

Validation

Pour assurer l'atomicité de la validation,celle-ci se fait en deux phases [LaSt.76]. La première phase consiste à mettre de côté toutes les informations nécessaires à la validation proprement dite On force l'écriture sur disque de toutes les pages touchées par la transaction,puis on marque dans la page de schéma, toutes les relations touchées par la transaction. La page de schéma est alors écrite sur disque,ce qui termine la première phase de validation. A partir de cet instant la transaction est logiquement validée. La deuxième phase consiste à faire effectivement la validation en parcourant la page de schéma et pour chaque relation marquée, remplacer ses champs ORIG par les champs COPIE. Ensuite il faut nettoyer les états des pages modifiées en mettant à l'état OCCUPEE les pages à l'état TRANSITOIRE DE COPIE et à l'état LIBRE les pages à l'état TRANSITOIRE D'ORIGINE.

Reprise

En cas de panne il faut pouvoir redémarrer le système à partir d'un état cohérent en perdant le moins de travail possible. Avec la technique de mises à jour différées nous pouvons repartir de l'état dans lequel se serait trouvé la base si on avortait toutes les transactions n'ayant pas terminé leur première phase de validation à l'instant de la panne. Pour ce faire on lance la deuxième phase de validation décrite ci-dessus, afin de terminer une éventuelle validation en cours au moment de la panne, puis on avorte toutes les autres. Cette procédure est lancée à chaque initialisation du système.

Modules de Manipulation des Données

L'organisation des couches de logiciel du système PEPIN est illustrée à la Figure 5 .

Détaillons rapidement les fonctions de chacune de ces couches. La couche MEMVIRT comprend les modules de gestion de la mémoire virtuelle Ces modules permettent d'adresser les pages de la base décrites à la Figure 2, de les amener en mémoire centrale si elle ne s'y trouvent pas, de forcer des pages en écriture sur la mémoire secondaire et d'en créer des copies pour le mécanisme des mises à jour différées. La couche PHYLOG s'occupe de la traduction d'adresse logique de pages d'une relation en

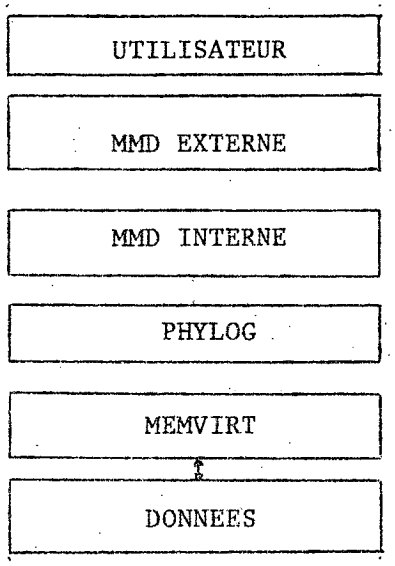

Figure 5 : Couches du système PEPIN

adresses physiques, de la gestion des mises à jour différées et de l'atomicité des transactions. En effet les couches supérieures voient les relations comme des suites de pages de nuplets et manipulent des adresses logiques de pages formées de couples (REL,CURS) où REL est un identificateur de relation et CURS un curseur. C'est la couche PHYLOG qui, avec la structure des catalogues et schémas illustrée à la Figure 2, retrouvera la page physique correspondant à l'adresse logique fournie, créera les copies nécessaires et mettra à jour les états des pages physiques.

La couche MMD INTERNE contient les modules de manipulation des données sur la structure interne de la base. Elle manipule les nuplets à l'intérieur des pages à partir d'informations internes transmises par la couche MMD EXTERNE.

La couche MMD EXTERNE contient les modules de manipulation des données qui sont proposées à l'utilisateur. Cette couche dispose du schéma externe de la base et se charge de traduire les requêtes sur ce schéma externe en requêtes sur la structure interne, compréhensible par la couche MMD INTERNE.

Les modules de manipulations, disponibles sous la forme d'une bibliothèque de procédures PASCAL afin de permettre l'écriture par l'utilisateur de programmes d'applications, sont les suivants :

- INSERTION d'un nuplet dans une relation
- MODIFICATION sur condition dans une relation
- SUPPRESSION sur condition dans une relation
- RECHERCHE sur condition dans une relation (= SELECTION)
- PROJECTION d'une relation sur un sous-ensemble de ses attributs
- JONCTION de deux relations sur un attribut commun. (En cours d'implémentation).

D'autres opérations sur relations (∩, ∪, / etc.) seront implantées ultérieurement. Les trois dernières opérations créent de nouvelles relations résultats.

Les conditions de recherche sont,soit des conditions élémentaires du type : ATTRIBUT opérateur de comparaison CONSTANTE,où les opérateurs sont : =, ≠, >, ≥, <, ≤,,soit des conditions complexes obtenues par conjonctions et disjonctions de conditions élémentaires.

La couche UTILISATEUR permet à l'utilisateur d'accéder aux procédures de la couche inférieure en mode interactif, s'il ne désire pas écrire de programme de manipulation.

Création de Relations et Vues de la Base

Comme nous l'avons vu au paragraphe précédent,un utilisateur peut créer des relations résultats d'opérations sur la base. Ces relations sont temporaires, c'est-à-dire qu'elles disparaissent à la prochaine validation ou avortement. Il est parfois utile de pouvoir garder ces relations ou de rajouter de nouvelles relations à la base de données. Le système PEPIN permet à l'utilisateur de le faire. Le système offre en outre un mécanisme de sécurité simplifié permettant à l'utilisateur de déclarer PRIVEE une relation qu'il a créée, il sera alors le seul à "voir" la relation et pouvoir y accéder.

Expliquons brièvement comment un utilisateur "voit" la base. Pour accéder aux nuplets d'une relation nous avons vu (cf. Figure 2) qu'il faut disposer de l'entrée dans le schéma correspondant à la relation. Il faut donc disposer d'une table associant le nom d'une relation à son numéro d'entrée. Cette table existe au niveau de la couche MMD EXTERNE et l'utilisateur ne "voit" que les relations qui figurent dans sa table.

Pour permettre d'identifier les utilisateurs et pour générer leur table quand ils se connectent au système,il existe deux relations privilégiées dans la base : UTILISATEURS et METARELATION. La relation UTILISATEURS est composée de deux attributs : IDENTITE et MOT-PASSE permettant au système d'identifier et de vérifier l'identité d'un utilisateur lorsqu'il se connecte. Remarquons qu'il existe un utilisateur privilégié appelé ADMINISTRATEUR qui a le droit de tout faire et de tout voir.

La METARELATION est en fait la relation contenant les informations sur les relations présentées dans la base de données. Entre autres elle a quatre attributs permettant de gérer les vues : NOM, ENTREE, CREATEUR et DROITS. L'attribut NOM donne le nom de la relation et l'attribut ENTREE donne l'entrée dans le schéma correspondant. L'attribut CREATEUR donne l'identité de l'utilisateur qui a créé la relation et l'attribut DROITS indique si la relation est PUBLIQUE ou PRIVEE. Quand un utilisateur se connecte au système, après l'avoir identifié et vérifié son identité, on génère sa table en y introduisant les relations PUBLIQUES et toutes les relations PRIVEES qu'il a créées. Les relations UTILISATEURS et METARELATION sont toutes deux PRIVEES et créées par l'ADMINISTRATEUR.

II. MESURES EFFECTUEES SUR LE SGBD PEPIN.

Dans cette partie, nous présentons le résultat de mesures sur le système PEPIN. Nous avons mesuré les temps moyens d'exécution de certaines opérations de manipulations des données (insertion, sélection ...) et des opérations de validation et avortement qui viennent en complément des précédentes, car le système effectue des mises à jour différées.

L'intérêt de cette série de mesures est double. D'une part elle doit prouver la facilité d'utilisation du système comme maquette d'évaluation de performances de Systèmes de Gestion de Bases de Données. D'autre part elle doit fournir u ordre de grandeur des performances d'un SGBD sur micro-ordinateurs, ainsi que le comportement de ces performances en fonction de certains paramètres de la base de données (taille de la base, taille des nuplets, chemins d'accès ...).

Nous allons d'abord décrire la méthode utilisée pour prélever ces mesures puis nous présenterons les résultats obtenus et analyserons leur comportement.

1. La méthode utilisée

Les résultats obtenus sont des durées moyennes d'exécution de certaines opérations de manipulation des données (insertion, recherche, validation, reprise ...), opérations situées dans la couche du système PEPIN décrite dans la première partie.

Par cela nous avons écrit un scénario de manipulation de la base formée des opérations ci-dessus, et comprenant des points de mesure logiciels de part et d'autre des opérations à mesurer. Ces points de mesure sont des appels à une fonction système qui permet d'accéder à l'horloge de la machine. Le manque de précision de cette fonction ($1/60^e$ de seconde) explique qu'il a fallu effectuer de nombreuses opérations pour obtenir des moyennes sans pouvoir fournir d'intervalle de confiance. Les mesures effectuées sont stockées en mémoire jusqu'à la fin de l'exécution du scénario afin de minimiser l'influence de ce logiciel de mesure sur les performances réelles du système. (En effet une écriture de ces mesures sur disque encours d'exécution modifie la position de la tête de lecture et donc la durée de l'accès suivant).

Ce scénario a été exécuté sur différentes bases, et dans chaque cas, successivement sur chacune des relations. La structure exacte des bases et des relations est exposée dans le paragraphe suivant avant la présentation de chaque résultat. Nous allons maintenant détailler le scénario utilisé puis présenter les mesures obtenues.

SCENARIO de MESURES.

- INITIALISATION PEPIN
  Remplissage de la Base jusqu'à 10 K octets
  RECHERCHES ⇒ durée moyenne de recherche
  VALIDATION ⇒ durée de validation
  INITIALISATION PEPIN
  Séquence d'INSERTIONS ⇒ durée moyenne d'insertion

Cette séquence d'opérations est ensuite effectuée pour un remplissage jusqu'à 50 Koctets, puis 100 Koctets.

- REMISE A ZERO de la BASE.

Nous allons faire quelques remarques sur ce scénario :

- Le remplissage de la base s'éffectue par insertion de nuplets dont le contenu est généré aléatoirement de façon uniforme. Il en est de même pour les nuplets insérés lors des séquences de mesures.

- Le scénario ne comporte pas de suppression, donc les relations non classées vont être remplies dans des conditions idéales si l'on se réfère à la structure de données utilisée. En effet, lorsqu'on insère des nuplets dans une relation non classée, le système recherche la "prochaine" place d'insertion d'un nuplet. Celle-ci est donc trouvée de manière immédiate lorsqu'on insère dans une relation dans laquelle aucune suppression n'a été faite. En revanche la suppression de nuplets va occasionner des "trous" dans des pages. Pour éviter la fragmentation , ces trous seront utilisés lors des prochaines insertions avant d'aller chercher de la place en fin de relation : par la suite, cette recherche des "trous" occasionnera des lectures de pages.

- Les RECHERCHES effectuées sont des recherches portant soit sur une condition élémentaire, soit sur une condition complexe. Dans la condition élémentaire ATTRIBUT = VALEUR, la valeur existe dans la base mais n'est peut-être pas unique. Dans toutes les recherches effectuées, le nombre de nuplets vérifiant cette condition est resté très faible.

Dans la recherche sur une condition complexe, la recherche est formée par un balayage des pages d'une relation et d'un filtrage sur chacune de ces pages. Pour mieux distinguer ces deux phases nous avons effectué des recherches selon une condition non élémentaire donc exigeant un filtrage plus important. La condition de recherche utilisée est du type :

(C1 et C2) et (C3 et C4) où les Ci désignent des condition élémentaires.

- Afin d'assurer une cohérence de chacune des mesures effectuées durant le scénario et de permettre des comparaisons significatives des résultats, le scénario comporte plusieurs réinitialisations.

Les Initialisations avant les mesures sur les opérations d'insertion permettent de travailler à partir d'une mémoire centrale ne contenant aucune page. La remise à zéro finale permet de réexécuter le scénario sur une autre relation en repartant d'une base vide .

2. Mesures effectuées :

Les mesures ont été effectuées sur un LSI 11/2 disposant d'un disque dur de 2 × 5 M octets dont le temps d'accès moyen est de 40 ms. Dans ce paragraphe tous les résultats présentés sont exprimés en secondes.

a. Relations non classées

La structure de la base utilisée est la suivante :

R1 : nuplet de longueur 10 octets
R2 : nuplet de longueur 50 octets
R3 : nuplet de --- 100 octets
R4 : nuplet - --- 200 octets

Durées d'insertion

Les résultats des mesures des durées d'une insertion de nuplet dans les relations non classées (R1, R2, R3 et R4) en fonction de la taille de la base sont présentées sur les figures 6 et 7.

Taille de la base \ Relation	R1	R2	R3	R4
0 K	0.145	0.158	0.172	0.205
10 K	0.148	0.162	0.175	0.208
50 K	0.155	0.168	0.180	0.215
100 K	0.158	0.170	0.183	0.220

Fig 6. Tableau des durées d'insertion dans les relations non classées R1, R2, R3 et R4 en fonction de la taille de la base.

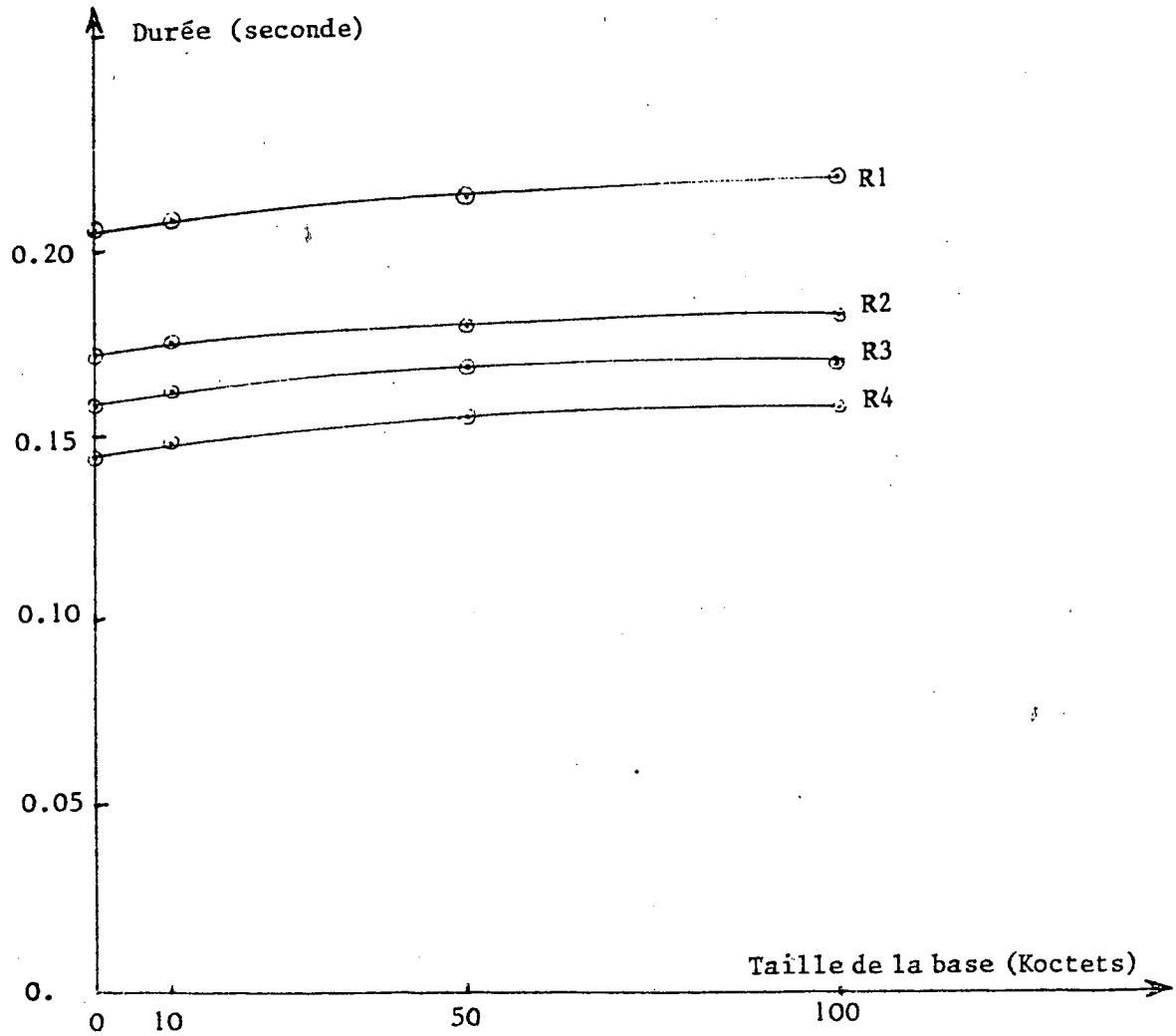

Figure 7 : Durée d'insertion d'un nuplet dans les relations non classées R1, R2, R3 et R4 en fonction de la taille de la base.

La durée d'insertion dans une relation non classée dépend très peu de la taille de la base. Il faut cependant rappeler que les conditions idéales de remplissage de la base font que la place d'une insertion suit immédiatement celle de la précédente insertion.

L'influence de la taille d'une nuplet se limite à une translation entre les courbes précédentes. Ceci est dû à la durée de copie effective du nuplet dans la base.

Durées de recherche avec condition élémentaire

Les résultats des mesures des durées de recherche avec condition élémentaire dans les relations non classées R1, R2, R3 et R4 en fonction de la taille de la base sont présentés sur les figures 8 et 9.

Taille de la base \ Relation	R1	R2	R3	R4	R4*
10 K	9.6	4.9	4.3	4.9	3.9
50 K	39.4	18.3	15.1	18.3	14.6
100 K	79.	35.1	28.4	35	28

Fig 8. Tableau des durées de recherche avec condition élémentaire dans les relations non classées R1, R2, R3 et R4 en fonction de la taille de la base.

La recherche (balayage des pages de la relation dans le cas d'une relation non classée) est une fonction affine de la taille de la base, donc du nombre de pages à parcourir.

Les résultats obtenus pour R4 semblent anormaux. Ceci est dû à un problème de fragmentation. Les nuplets des relations R1, R2 et R3 remplissent parfaitement la page tandis que ceux de la relation R4 laissent un espace de 100 octets innoccupés (soit 1/5 de la page). La relation R4 utilise donc plus de pages que les autres relations pour une même quantité d'information. La colonne R4* contient des résultats déduits par affinité des précédents pour tenir compte de cette expansion en nombre de pages.

Durées de recherche avec condition complexe

Les résultats des mesures des durées de recherche avec une condition complexe sont présentés sur la figure 10.

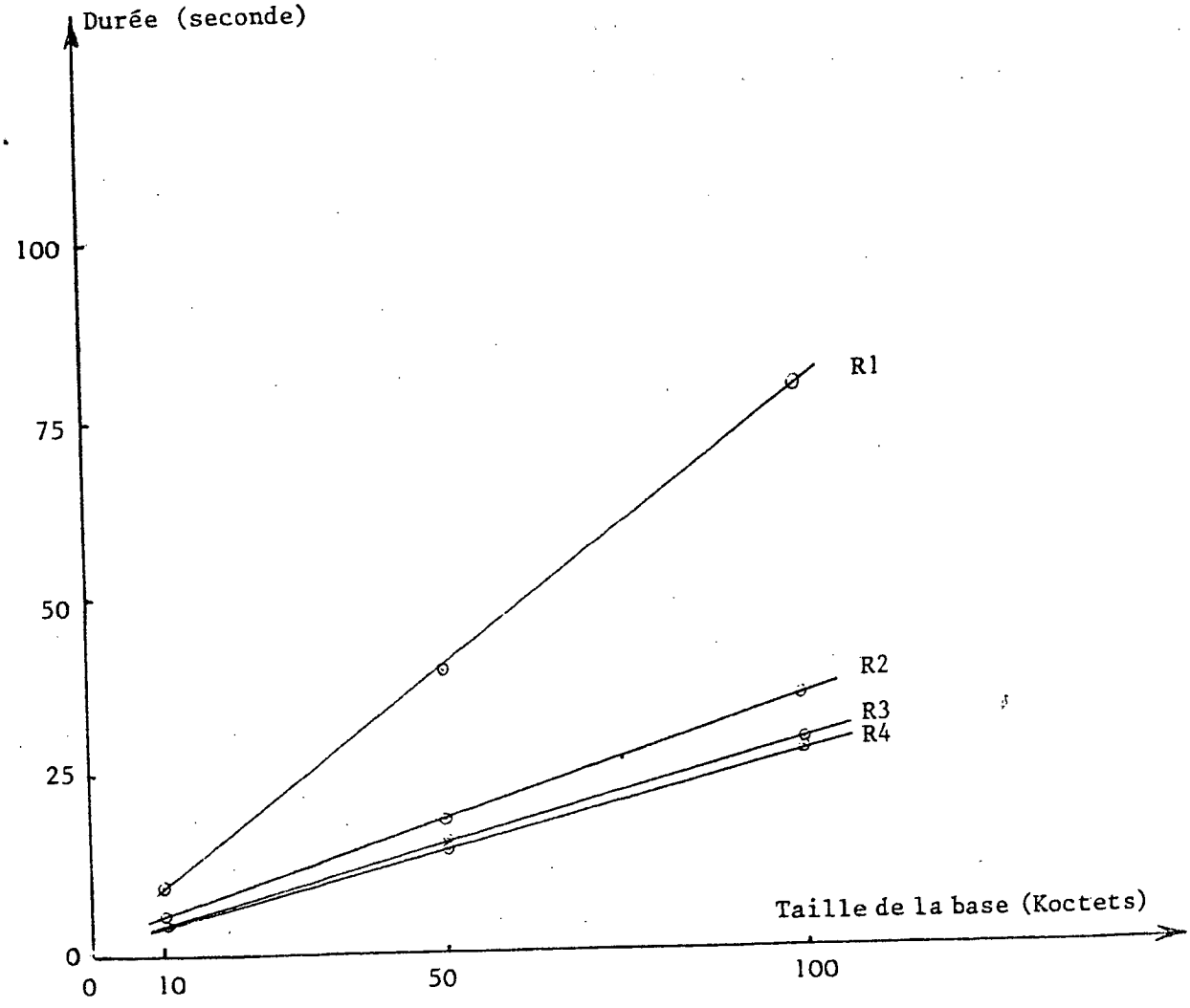

Figure 9 : Durée de recherche sur condition élémentaire dans les relations non classées R1, R2, R3 et R4 en fonction de la taille de la base.

Taille de la base \ Relation	R1	R2	R3	R4
10 K	34.5	9.7	6.8	5.7
50 K	166.	42.5	28.1	22.6
100 K	331	84.1	54.8	43.5

Figure 10. Tableau des durées de recherche avec condition complexe dans les relations non classées R1, R2, R3 et R4 en fonction de la taille de la base.

La durée du filtrage est non négligeable voire même assez importante par rapport au balayage de la base. Il ne faut cependant pas oublier que le pascal UCSD utilisé, interprète le P.code.

b. Relations classées.

La structure de la base utilisée est la suivante :

R1 : nuplet de longueur 50 octets
R2 : " " " 100 "
R3 : " " " 200 "

Ces relations sont classées sur leur premier attribut. La longueur de cet attribut de classement a été prise égale à 2 octets dans un premier essai puis à 15 octets ensuite afin d'étudier son influence.

Durées d'insertion

Les résultats des mesures des durées d'insertion dans les relations classées R1, R2 et R3 en fonction de la taille de la base et de la longueur de l'attribut de classement sont présentés sur les Figures 11 et 12.

Long. attribut de classement / Taille de la base / Relations	2 R1	15 R1	2 R2	15 R2	2 R3	15 R3
0 K	0.706	0.715	0.837	0.827	1.04	1.04
10 K	1.06	1.08	1.10	1.11	1.43	1.4
50 K	1.19	1.27	1.37	1.35	1.56	1.6
100 K	1.61	1.73	1.73	1.77	1.85	1.9

Figure 11. Tableau des durées d'insertion dans les relations classées R1, R2 et R3 en fonction de la taille de la base et de la longueur de l'attribut de classement.

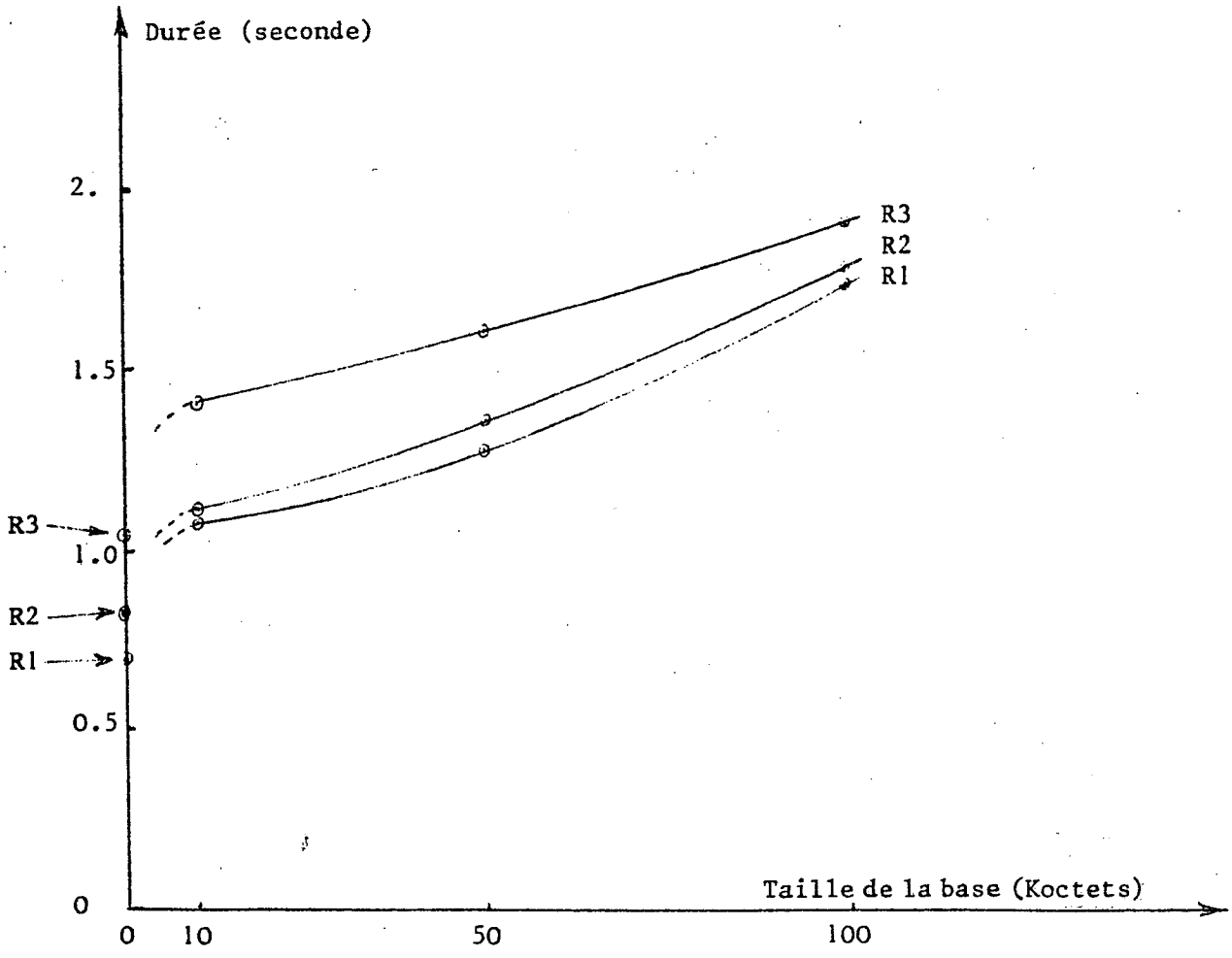

Figure 12 : Durée d'insertion d'un nuplet dans les relations classées R1, R2 et R3 en fonction de la taille de la base.

La recherche d'une place d'insertion effectuée par une dichotomie explique la croissance logarithmique de la durée d'insertion en fonction de la taille de la base.

Les variations en fonction de la taille du nuplet sont dues, d'une part à l'insertion effective du nuplet dans la page, mais surtout à une variation du taux de remplissage des pages. L'insertion dans une relation classée entrainent fréquemment un éclatement des pages , celles-ci ne sont jamais entièrement remplies. On constate que ce taux de remplissage est en moyenne meilleur pour des nuplets plus courts. En ce qui concerne la relation R3, il faut également noter comme précédemment que 100 octets sont systèmatiquement perdus, ce qui explique cette plus grande différence.

L'influence de la taille d'attribut de classement n'est pas significative, en effet, le plus souvent la comparaison de l'attribut à la constante de la condition s'arrête aux premiers octets.

Durée de Recherche

Les résultats des mesures des durées de recherche avec une condition élémentaire sont représentée sur les figures 13 et 14.

Taille de la base \ Relation	R1	R2	R3
10 K	1.5	1.7	1.7
50 K	1.9	1.8.	1.8
100 K	2.2	2.	2.

Figure 13. Tableau des durées de recherche avec une condition élémentaire dans les relations classées R1, R2 et R3 en fonction de la taille de la base.

Le nombre de paramètres intervenant dans une recherche est assez important : nombre exact de nuplets vérifiant la condition, situation de ces nuplets (dans une même page ou chevauchant deux pages)... Il est donc difficile d'évaluer l'influence de la longueur du nuplet sur le temps de recherche. On peut seulement constater que la variation en fonction de la taille de la base est assez faible et est assez proche d'une variation logarithmique.

Ces résultats doivent surtout être pris pour des ordres de grandeur à comparer par exemple avec ceux obtenus sur une relation non classée.

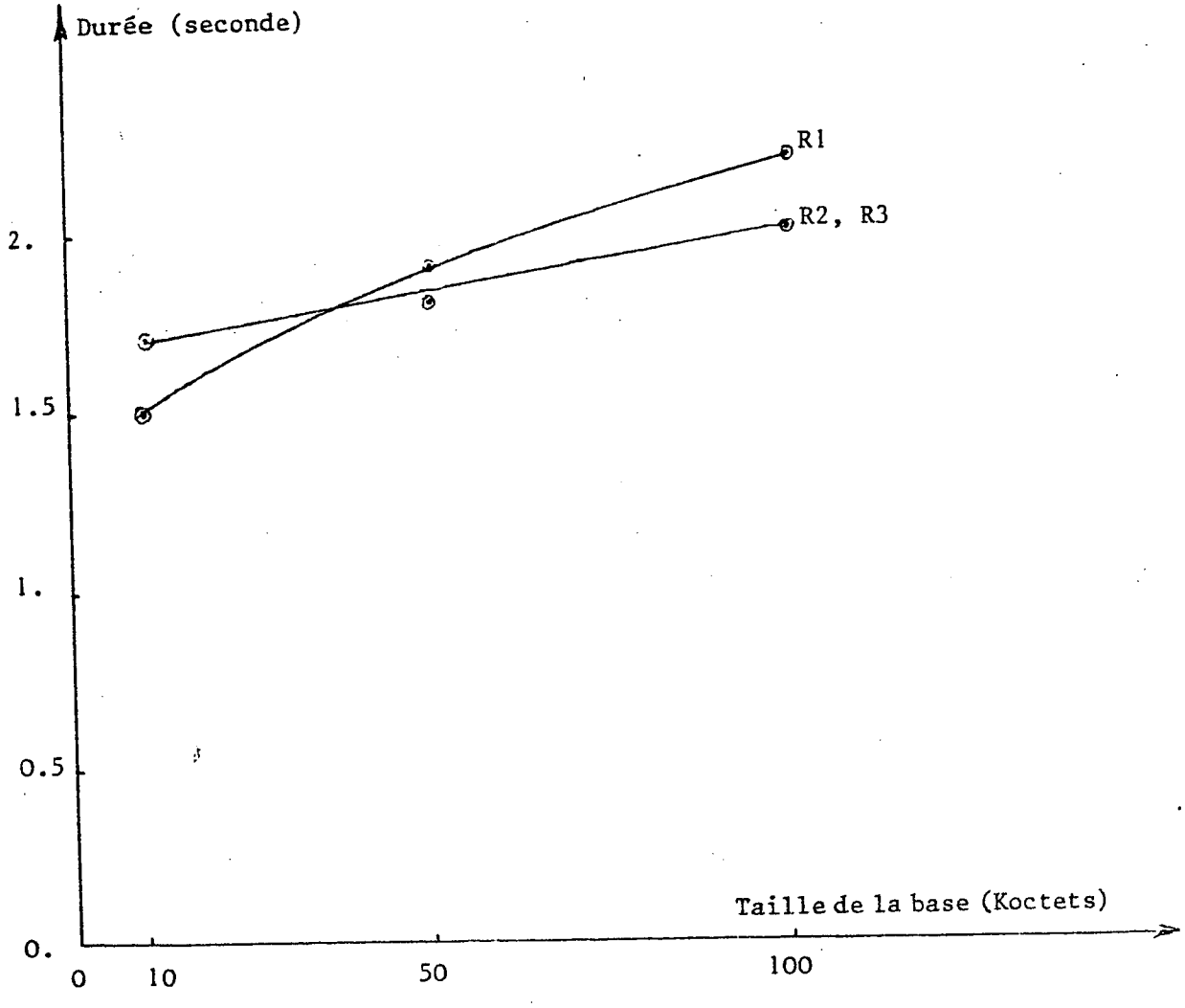

Figure 13 : Durée de recherche dans les relations classées R1, R2 et R3 en fonction de la taille de la base.

c. Validation et reprise

Comme cela a été décrit dans la première partie, l'opération de reprise se limite à la 2eme phase de la validation. Après quelques mesures il s'est avéré que la phase la plus longue de la validation est cette 2eme phase,ce qui conduit à étudier conjointement ces deux opérations.

Les variations de la durée de validation sur diverses transactions sont assez faibles (moins de 10 %) par rapport à la durée totale. Les paramètres intervenant sur la durée de validation (nombre de pages en mémoire centrale à réécrire sur disque, nombre de pages modifiées pour la transaction) sont trop aléatoires pour qu'il soit possible d'étudier précisement cette durée. Seule une valeur moyenne, observée lors des expériences précédentes peut-être donnée : 2,5 s.

Cette durée est assez importante par rapport au temps d'exécution d'une petite transaction (quelques insertions par exemple) mais devient assez négligeable quand la transaction comporte un plus grand nombre d'opérations.

Les mesures qui ont été présentées dans cette seconde partie permettent de déduire le comportement de transactions effectuant des opérations sur les nuplets d'une base. Elles seront complétées, au fur et à mesure de leur mise au point par des mesures sur les opérations relationnelles, à commencer par la JONCTION et le TRI dont on sait la grosse incidence sur les performances des SGBD.

PERSPECTIVES ET CONCLUSION

Dans une phase ultérieure de notre travail l'utilisation de PEPIN pour l'évaluation de performances va être poursuivie, en lien avec des études théoriques sur la modélisation. Ceci nous permettra de faire "les meilleurs choix" pour PEPIN, en tenant compte de l'environnement particulier que crée une implantation sur microordinateurs.

Le fait que PEPIN ait été conçu pour une implantation multimachine permet de développer ces travaux dans deux directions : l'une concerne l'aspect base de données et l'autre l'aspect répartition. Sur certains points ayant déjà fait l'objet d'études théoriques notre travail apportera l'élément de comparaison pratique et réel qui manque le plus souvent.

Pour l'aspect base de données les points qui nous semblent devoir être abordés pour commencer concernent :

- la taille des pages
- La granularité des verrous
- les problèmes de gestion de la mémoire liés à la concurrence
- la taille des résultats d'opérations relationnelles, en lien avec le problème de l'optimisation des requêtes.

. la comparaison d'algorithmes de maintien de la cohérence des données.

L'aspect répartition nous intéresse particulièrement car beaucoup de problèmes sont ouverts : en effet il a d'abord fallu trouver des algorithmes montrant la faisabilité, après quoi il est nécessaire d'évaluer ces algorithmes pour en apprécier le comportement et l'impact sur le fonctionnement du système. Ces algorithmes du maintien de la cohérence dans les bases de données réparties illustrent parfaitement cette situation où de très nombreux algorithmes ont été proposés mais, faute de réalisations, leur impact sur le fonctionnement du système ne peut pas être apprécié. Il est aussi intéressant par rapport à une application, ici SGBD, d'évaluer l'impact de la répartition sur ces performances, par la surcharge induite en ressource utilisées (calcul et mémoire) et en oisiveté forcée, à l'occasion par exemple de synchronisation. Un autre vaste domaine d'investigation pour l'évaluation de performance concerne tout ce qui a trait au parallélisme interne aux transactions et au parallèlisme entre transactions.

L'existence de notre maquette PEPIN nous permet donc d'envisager de nombreuses études d'évaluation des performances (certaines sont en cours actuellement), dont celle présentée dans la seconde partie de cet article n'est que la première.

BIBLIOGRAPHIE

[AsKs 80] M.M. ASTRAHAN, W.KIM, M.SCHKOLNICK
"Evaluation of the System R Access Path Selection Mechanism"
IBM Research Laboratory, San José (Cal) U.S.A.
RJ 27 97 - 4/10/80 - 18 pages.

[BCFJK 81a] P.BOUCHET, A.CHESNAIS, J.M.FEUVRE, G.JOMIER, A.KURINCK
"PEPIN : An Experimental Multi-microcomputer Data Base Management System"
2nd Conference on Distributed Computing Systems - Versailles - Avril 1981 - p 211-217.

[BCFJK 81b] P.BOUCHET, A.CHESNAIS, J.M. FEUVRE, G.JOMIER, A.KURINCK
"Data base for Microcomputers : the Pepin Approach"
ACM SIGMOD / SIGSMALLS - Orlando, October 81
Sigsmall Newsletter Vol 7, n°2, pp 145-151.

[ChGM 81] A.CHESNAIS, E.GELENBE, I.MITRANI
"On the Modelling of Concurrent Access to Shared Data"
Applied Probability / Computer Science : the interface - Bocaraton - Florida - Jan 81.

[CoGP 80] E.G. COFFMAN, E.GELENBE, B.PLATEAU
"Optimization of the Number of Copies in a Distributed Data Base System"
Performance 80 - Toronto - ACM SIGMETRICS vol 9, 2 - Summer 80, pp 93-104.

[Ever 81] G.C. EVEREST
"Comparative Study of Data Base Management Systems on Microcomputers"
ACM SIGMOD / SIGSMALLS - Orlando, Oct 1981 - Sigsmall Newsletter n°2 Vol 7 - p 77-89.

[Garc 79] H. GARCIA MOLINA
"Performance of Update Algorithms for Related Data in a Distributed Data Base" Stanford University, USA, 1979, Ph.D. - Thesis.

[GeGa 82] E.GELENBE, D.GARDY
"On the Size of Projections I"
"On the Size of Projections II"
Rapport de Recherche - LRI - Université Paris Sud-Orsay , France 1982 - proposé pour publication .

[Giff 80] DK. GIFFORD
"Violet, an Experimental Decentralized System"
Integrated office Systems - Burotics
N.Naffah Editor - North Holland - 1980.

[GoVe 80] V.GOPALAKRISHNA, C.E. VENI MADHAVAN
"Performance Evaluation of Attribute Based Tree Organization"
ACM Trans. on Data Base Systems Vol 6,
n°1, March 80, pp 69-87.

[HaSt 79] P.HAWTHORN and M.STONEBRAKER
"Performance Analysis of a Relational Data Base Management System"
Proc. ACM SIGMOD Conf. Boston 1979 - pp 1-14.

[HeWY 81] P.HEIDELBERGER, P.D. WELCH and P.L.YUE
"Statistical Analysis of Data Base Systems Measurements"
Performance 81 - Amsterdam - Nov 81 - pp 325-344.

[KIM 81] W.KIM
"Query Optimization for Relationnal Data Base Systems"
IBM Research Report - San José (Cal) - USA - RJ 3081 -
3/10/81.

[LaSt 76] B.W. LAMPSON, H.E. STURGIS
"Crash Recovery in a Distributed System"
XEROX Palo Alto Research Center - 1976.

[Lori 77] R.A. LORIE
"Physical Integrity in a Large Segmented Data Base"
ACM Transactions on Data Base Systems
March 1977 - Vol 2 n°1 - 1977.

[LoMa 81] B.J. LOWNDES and J.W. MARTIN
"A Comparative Study of Join Data Base Management Systems"
Data bases - Proc 1st. British Nat. Conf. on Data Bases -
Cambridge July 81 - pp 187-205.

[Mira 82] S.MIRANDA
"SGBD Relationnels : un survol et une proposition"
Rapport CERISS 82009 Janvier 1982
Université des Sciences Sociales de Toulouse.

[PoLe 80] D.POTIER and Ph.LEBLANC
"Analysis of Locking Policies in Data Bases Management Systems"
CACM 23, 10, Oct 1980 - pp 584-593

[RiSt 77] D.R. RIES, M.STONEBRAKER
"Effects of Locking Granularity in Data Base Management System"
ACM Trans. on Data Base Systems. Vol 2, n°3 - Sept. 1977 -
pp 233-246.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

[Ries 81] D.R. RIES
"The Effects of Concurrency Control on the Performance of Data Base Management Systems"
Rapport Univ. California, Berkeley, CA, USA
Electronics Research Laboratory - 1981 - 222p.

[Rich 81] Ph. RICHARD
"Evaluation of the Size of a Query Expressed in Relational Algebra"
Proc ACM Sigmod Conf. 1981 - p 155-163.

[Sevc 81] K.SEVCIK
"Data Base System Performance Prediction Using an Analystical Model"
Proc. 7th VLDB - Ed. Zaniolo - Delobel - PP 182-198 -
Cannes - 1981.

[ShSp 81] A.W.SHUM and P.G. SPIRAKIS
"Performance Analysis of Concurrency Control Methods in Data Base Systems"
8th International Symposium on Computer Performance Modelling Measurement and Evaluation.
Nov 1981 - Amsterdam - pp 1-20.

[Tsic 81] D.TSICHRITZIS
"Integrating Data Bases and Message Systems"
Proc 7th VLDB. Cannes Sept 1981 - pp 356-362.

[Wilm 79] P.WILMS
"Etude et comparaison d'algorithmes de maintien de la cohérence dans les bases de données réparties"
Thèse - INP Grenoble, France - 23/11/79.

ISSN 0249-6399