



**HAL**  
open science

## Evaluation d'algorithmes de controle de concurrence

J. Madelaine

► **To cite this version:**

J. Madelaine. Evaluation d'algorithmes de controle de concurrence. RR-0164, INRIA. 1982. inria-00076395

**HAL Id: inria-00076395**

**<https://inria.hal.science/inria-00076395>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**IRIA**

**CENTRE DE ROCQUENCOURT**

Rapports de Recherche

N° 164

**ÉVALUATION D'ALGORITHMES  
DE CONTRÔLE DE CONCURRENCE**

*8/80*  
*Reçu le 20.10.82*  
*Kerf 3/6*

Institut National  
de Recherche  
en Informatique  
et en Automatique

**Jacques MADELAINE**

Domaine de Voluceau  
Rocquencourt  
BP 105

78153 Le Chesnay Cedex  
France

Tel 954 90 20

Octobre 1982

# IRIA

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° 164

### ÉVALUATION D'ALGORITHMES DE CONTRÔLE DE CONCURRENCE

Jacques MADELAINE

Octobre 1982

Evaluation d'algorithmes de Contrôle de Concurrence

J. Madelaine

Projet SABRE

INRIA et Université de Paris VI

INRIA, BP 105, 78153 Le Chesnay Cedex, France

AOUT 1982



PAPIER RECUPERE ET RECYCLE

# Evaluation d'algorithmes de Contrôle de Concurrence

J. Madelaine

Projet SABRE

INRIA et Université de Paris VI

INRIA, BP 105, 78153 Le Chesnay Cedex, France

## Résumé

Cette communication compare les performances de deux algorithmes de contrôle de concurrence: verrouillage deux phases avec prévention des verrous mortels et ordonnancement par estampillage. La méthode employée est une résolution analytique d'un réseau de files d'attente permettant de modéliser la machine Base de Données SABRE afin de comparer les temps de réponse. Il y est montré que le verrouillage est meilleur que l'estampillage s'il y a une forte probabilité de conflit entre transactions. Par contre, si le nombre moyen de requêtes par transactions est grand alors l'estampillage devient le meilleur, ceci est renforcé lorsque la fréquence des petites transactions s'élève (nombre de requêtes suivant une loi géométrique).

Mots clefs: Contrôle de concurrence, verrouillage, estampillage, évaluation de performances, réseaux de files d'attente.

AOUT 1982

# Performance evaluation of concurrency control algorithms

J. Madelaine

Projet SABRE

INRIA and Université de Paris VI

INRIA, BP 105, 78153 Le Chesnay Cedex, France

## Abstract

This communication compares the performances of two concurrency control algorithms: two-phases locking and timestamp ordering. This is achieved by solving analytically a queuing network which gives response times of the SABRE data base machine. It is shown that locking is better than timestamp ordering in the case of a high probability of conflict between transactions, and, if the mean number of queries per transactions is high then timestamp ordering is the best, this is improved when small transactions frequency increases (i.e. queries number having a geometrical distribution law).

Key words: Concurrency control, two-phases locking, timestamp ordering, performance evaluation, queuing network.

## 2- Position du problème

Dans un SGBD, le contrôle de concurrence doit permettre à plusieurs utilisateurs de manipuler une même base de données tout en garantissant l'intégrité de celle-ci. De nombreuses techniques permettent d'arriver à ce but. Parmi elles, deux classes, regroupant les principales, ont été dégagées [BERN81] [GARD81]:

- algorithmes basés sur l'ordonnancement par estampillage
- algorithmes basés sur le verrouillage deux phases.

La validité de ces algorithmes étant prouvée, le problème est de faire une comparaison quantitative des performances du système, suivant que la méthode de contrôle de concurrence choisie est du type verrouillage deux phases ou ordonnancement par estampillage. Cette section décrit les origines du problème, puis les deux algorithmes évalués.

### 2.1- Origines du problème

Parmi les nombreuses études d'évaluation qualitatives et quantitatives d'algorithmes de contrôle de concurrence, retenons d'abord l'étude de Bernstein et Goodman : "Fundamental Algorithms for Concurrency Control in DDBS" [BERN81]. Ce travail est en deux parties. La première est une classification des différentes méthodes de contrôle de concurrence et aboutit à en identifier 48 différentes appartenant aux deux classes définies ci-dessus. La deuxième partie est une comparaison qualitative de ces méthodes par rapport au coût en communication, au coût de gestion, au

nombre de reprise et aux blocages éventuels. Il ressort de cette comparaison 12 méthodes optimales par rapport à ces critères dont la méthode d'ordonnement par estampillage et le verrouillage deux phases avec prévention des verrous mortels. Notons que l'utilisation du verrouillage induit la possibilité de verrous mortels qu'il faut supprimer par une technique appropriée (détection, prévention); la prévention se révèle être meilleure que la détection par rapport au nombre de messages utilisés. De plus, Bernstein et Goodman notent, à juste titre, qu'une étude quantitative doit être menée dans un environnement précis. L'environnement de notre étude sera la machine BD SABRE (cf section 3).

Garcia Molina est l'auteur de la première évaluation quantitative d'algorithmes de contrôle de concurrence [GARC79]. Il modélise le contrôle de concurrence d'une BD répartie complètement dupliquée sur plusieurs sites à l'aide de réseaux de files d'attente. En résolvant ces réseaux par simulation et par une méthode analytique, il conclut que le verrouillage deux phases avec détection centralisée des verrous mortels est meilleur que l'algorithme distribué de Thomas [THOM79]. Ces résultats sont obtenus pour des transactions n'effectuant qu'un nombre constant de mises à jour et lorsqu'il y a peu de conflits.

D'autre part, Ries et Stonebraker ont étudié, dans deux papiers "Effects of locking granularity in a DBMS" [RIES77] et "Locking granularity revised" [RIES79] la technique du

## Evaluation d'algorithme de Contrôle de Concurrence

### 1- Introduction

### 2- Position du problème

#### 2.1- Origines du problème

#### 2.2- Description des deux algorithmes évalués

### 3- Modélisation par un réseau de files d'attente

#### 3.1- Description de l'environnement

#### 3.2- Description du traitement d'une transaction

#### 3.3- Réseau de files d'attente modélisant le traitement

### 4- Méthode de résolution

### 5- Résultats et conclusions

#### Annexe A: Résolution du système d'équations

donnant les nombres de visites par classe.

#### Annexe B: Calcul des nombres moyens de visites par files

## 1- Introduction

Les nombreuses techniques de contrôle de concurrence appartiennent, pour la plupart, à deux classes [BERN81] [GARD81]:

- (1) algorithmes basés sur le verrouillage deux phases
- (2) algorithmes basés sur l'ordonnancement par estampillage.

Les méthodes de la première classe évitent les conflits en posant des verrous sur les données utilisées, et, de façon à garantir la cohérence, le verrouillage doit être en deux phases, c'est à dire que les verrous ne peuvent être relâchés qu'en fin de transaction, ce qui implique une première phase de verrouillage et une deuxième de déverrouillage. Cette méthode, pouvant amener une situation de verrous mortels, doit être complétée par une technique éliminant cette possibilité. Nous choisirons d'étudier la technique de prévention par ordonnancement des transactions qui est moins couteuse en messages que la technique de détection. Ceci rend cette méthode de contrôle de concurrence beaucoup plus comparable sur ce critère avec la deuxième méthode étudiée : l'ordonnancement par estampillage. Ce dernier vérifie que l'accès aux données par les transactions suit l'ordre des estampilles. Si cet ordre n'est pas respecté par une transaction alors elle est avortée pour être reprise ultérieurement. Il est démontré que ces deux méthodes sont correctes, reste à trouver quelle est la meilleure au point de vue performances.

Cette communication analyse et compare les performances des deux algorithmes représentatifs des deux classes par résolution analytique d'un réseau de files d'attente modélisant

l'implémentation possible de ces méthodes de contrôle de concurrence dans la machine base de données SABRE. Le temps de réponse moyen pour une transaction est obtenu en fonction des paramètres de l'ensemble des transactions soumises au système. Nous intéressant à un système fermé, le premier paramètre est le nombre total de transactions. Un paramètre important pour notre étude est la probabilité pour une transaction d'entrer en conflit avec une autre transaction lors d'un accès à une donnée. Cette probabilité est fonction non seulement de la taille de la base de données et du nombre de transaction mais aussi de la localisation dans la base des données accédées par les transactions. C'est pourquoi nous retenons ce paramètre plutôt que la taille de la base. Restent deux paramètres: le nombre de requêtes par transaction et le nombre d'accès par requêtes. Pour tenir compte de la loi de distribution de ces nombres et non seulement de la moyenne de ceux-ci, le modèle est affiné par la définition de classes de clients dans le réseau de files d'attente.

La section 2 passe en revue quelques travaux d'évaluation d'algorithmes de contrôle de concurrence et présente les deux algorithmes évalués. Le traitement d'une transaction dans la machine BD SABRE est décrit dans la section 3 ainsi que le réseau de files d'attente le modélisant. La section 4 est constitué par les définitions des classes de clients et le calcul des nombres moyens de visites par classe et par file. Les résultats sont analysés dans la section 5 et il ressort que, par rapport au temps de réponse, le verrouillage est meilleur que l'estampillage

s'il y a une forte probabilité de conflit entre transactions et que ces transactions sont "petites", cet avantage s'amenuisant dès que les transactions "grossissent". Si, de plus, il y a une importante proportion de "petites" transactions dans l'ensemble des transactions (nombre de requêtes et d'accès suivant une loi géométrique) alors l'estampillage devient meilleur quelque soit la probabilité de conflit.

verrouillage par simulation d'un SGBD mono-site. Ils s'intéressent à la taille du granule de verrouillage et concluent que pour des transactions comportant beaucoup de requêtes de "gros" granules suffisent (environ 10 pour toute la base), par contre, pour de petites transactions, de "petits" granules deviennent intéressants. Dans cette étude le nombre de requêtes pour chaque type de transactions (grosses ou petites) est constant.

Une étude similaire qui aboutit aux mêmes résultats a été réalisée par Irani et Lin [IRAN79], la méthode employée étant ici une résolution analytique d'un réseau de files d'attente. Potier et Leblanc dans "Analysis of Locking Policies in Database Management System" [POTI80] ont également travaillé à ce problème. Ils ont affiné l'étude en ce qui concerne le calcul de la probabilité de conflit entre transactions. Cette probabilité est obtenue en fonction du nombre de granules de la base, du nombre d'accès par transactions et du nombre de transactions dans le système et repose sur l'hypothèse que les accès aux données sont répartis uniformément sur toute la base. Les résultats obtenus se trouvent donc en conséquence légèrement modifiés par rapport à ceux des deux précédents papiers.

Nous sommes donc amenés à évaluer quantitativement deux algorithmes trouvés optimaux par Bernstein et Goodman, ceci avec une méthode analytique inspirée par les deux dernières références.

## 2.2- Description des deux algorithmes évalués

### 2.2.a- Verrouillage deux phases avec prévention des verrous mortels

Le verrouillage consiste à éviter la génération d'exécution incorrecte de transactions en faisant attendre les transactions voulant effectuer des opérations conflictuelles sur une même donnée. Cette méthode est comparable à la méthode classique d'allocation de ressources à des processeurs. Mais, pour avoir un contrôle de concurrence correct, il faut restreindre l'utilisation des actions "verrouiller" et "déverrouiller". Deux phases sont ainsi obtenues: une première phase de verrouillage suivie d'une deuxième de déverrouillage. D'autre part, cette méthode pose le problème des verrous mortels. Pour prévenir leurs formations, il est possible de définir un ordre initial des transactions à l'aide d'estampilles et d'interdire les attentes entre transactions incompatibles avec cet ordre. C'est selon ce principe que fonctionne la stratégie DIE-WAIT décrite par Rosenkrantz [ROSE78]. Elle autorise seulement une transaction à en attendre une plus jeune; si une trans est bloquée par une trans plus vieille qu'elle même alors la trans bloquée "meurt" (est reprise).

### 2.2.a- Ordonnancement par estampillages

Cette méthode consiste d'abord à définir un ordre initial des transactions en leur attribuant une estampille (un simple

nombre), et ensuite à vérifier à l'aide des estampilles des données, que les accès conflictuels sont effectués dans l'ordre donné; si deux trans ne respectent pas cet ordre alors l'une d'elle est reprise [BERN80]. La méthode la plus performante utilise deux estampilles de données: une estampille de lecture (EL) et une d'écriture (EE). On obtient les algorithmes suivants [GARD81] où  $T_i$  est une transaction d'estampille  $i$  et  $d$  une donnée qu'elle désire lire ou écrire:

Procédure LIRE ( $T_i, d$ )

Si (  $EE(d) \leq i$  ) alors

    "exécuter la lecture"

$EL(d) := \max( EL(d), i )$

Sinon AVORTER

finsi

finLIRE

Procédure ECRIRE ( $T_i, d$ )

si (  $EL(d) \leq i$  ) alors

    si (  $EE(d) \leq i$  ) alors

        "exécuter l'écriture"

$EE(d) := i$

    finsi

    sinon AVORTER

    finsi

finWRITE

### 3- Modélisation par un réseau de files d'attente

#### 3.1- Description de l'environnement

Nous modélisons un système (cf figure 3.1) composé de trois entités:

- une Machine BD SABRE
- un réseau de communication
- un ensemble de terminaux Hôtes.

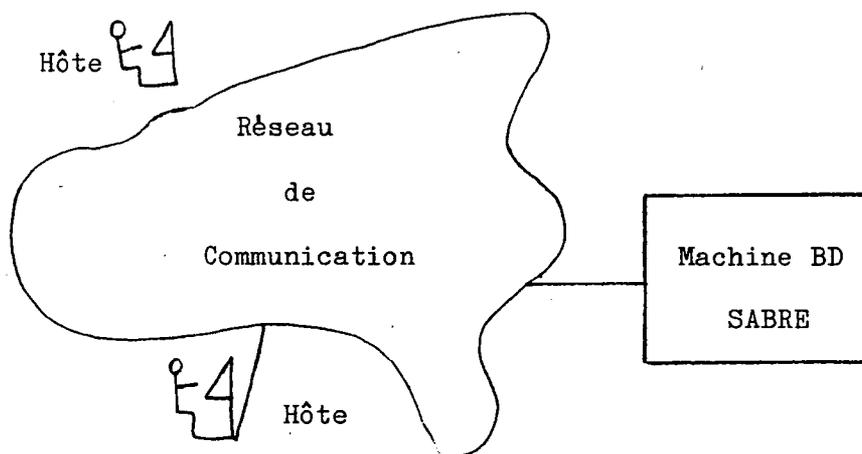


Figure 3.1: Architecture du système.

Des transactions émises par les utilisateurs à partir des terminaux, transitent par le réseau de communication et sont ensuite traitées par la machine BD qui retourne les résultats à travers le réseau de communication.

La machine BD SABRE a une architecture multi-processeurs possédants chacun une mémoire locale et communiquants à travers un bus (cf figure 3.2). Nous distinguons les processeurs suivants:

- Processeur Station de Transport qui gère les communications

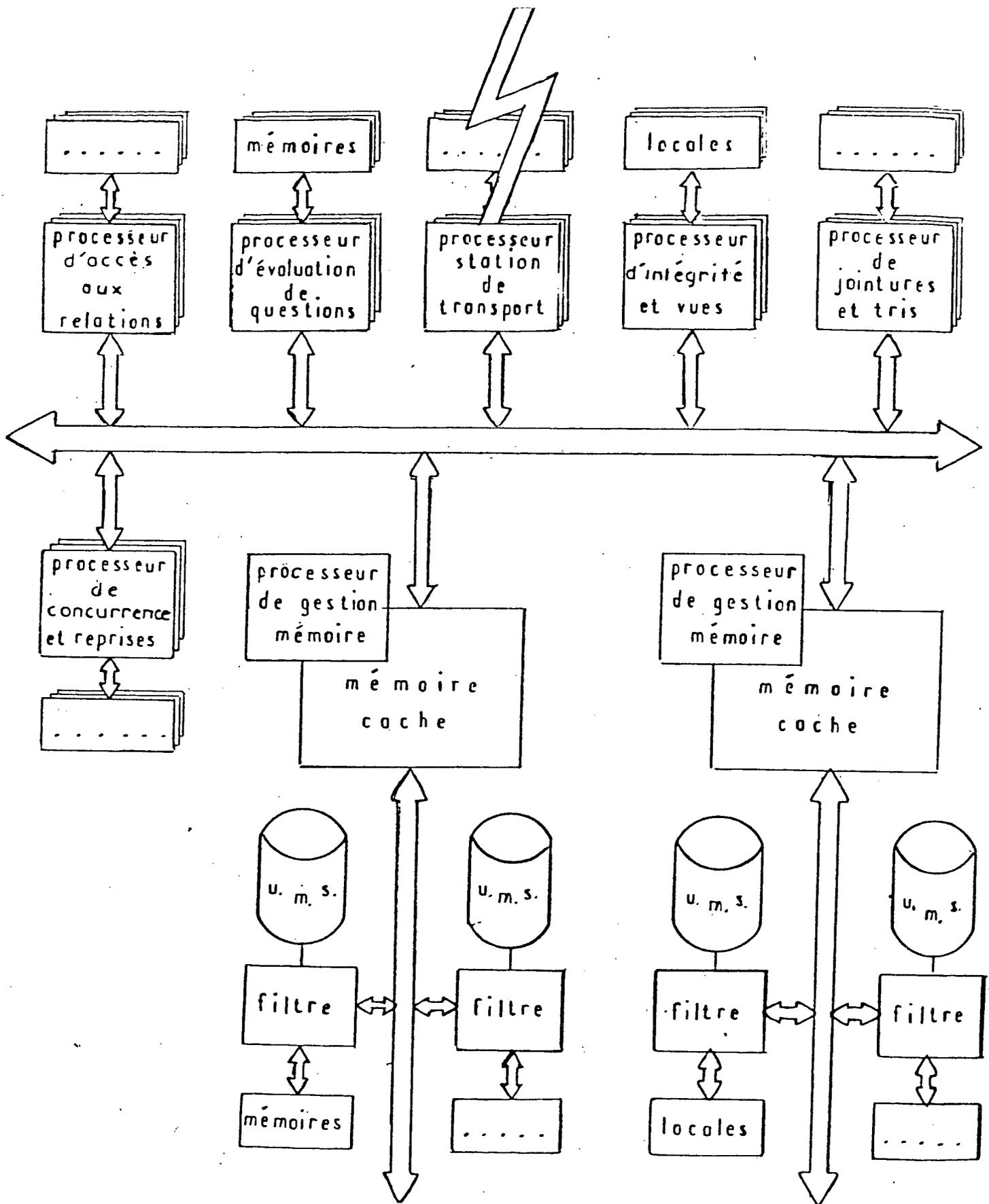


Figure 3.2: Architecture de la machine base de données SABRE

- Processeur d'Intégrité et de Vues qui permet d'utiliser des vues externes
- Processeur d'Evaluations de Questions (PEQ) qui décompose et optimise l'exécution des requêtes
- Processeur d'Accès aux Relations (PAR) qui gère les chemins d'accès aux données
- Processeurs de Jointures et de Tris (PJT) qui effectuent les opérations binaires sur les relations (jointures, unions, intersections, etc ) et les tris
- Processeur de concurrence et de journalisation (PCJ) qui s'occupe du contrôle de concurrence et des journaux permettant les reprises
- Processeur de gestion de la Mémoire Cache (PMC) qui gère les relations intermédiaires et finales, et permet l'accès aux données stockées sur mémoires secondaires à travers des filtres; c'est ici que sont effectuées les restrictions et projections de relations (sans élimination des doubles) .

Pour plus détails nous renvoyons le lecteur à la présentation générale de la machine SABRE [GARD82].

### 3.2- Description du traitement d'une transaction

Une transaction est formée de requêtes donnant lieu à des accès aux données. Le traitement d'une transaction est représenté schématiquement sur la figure 3.3.

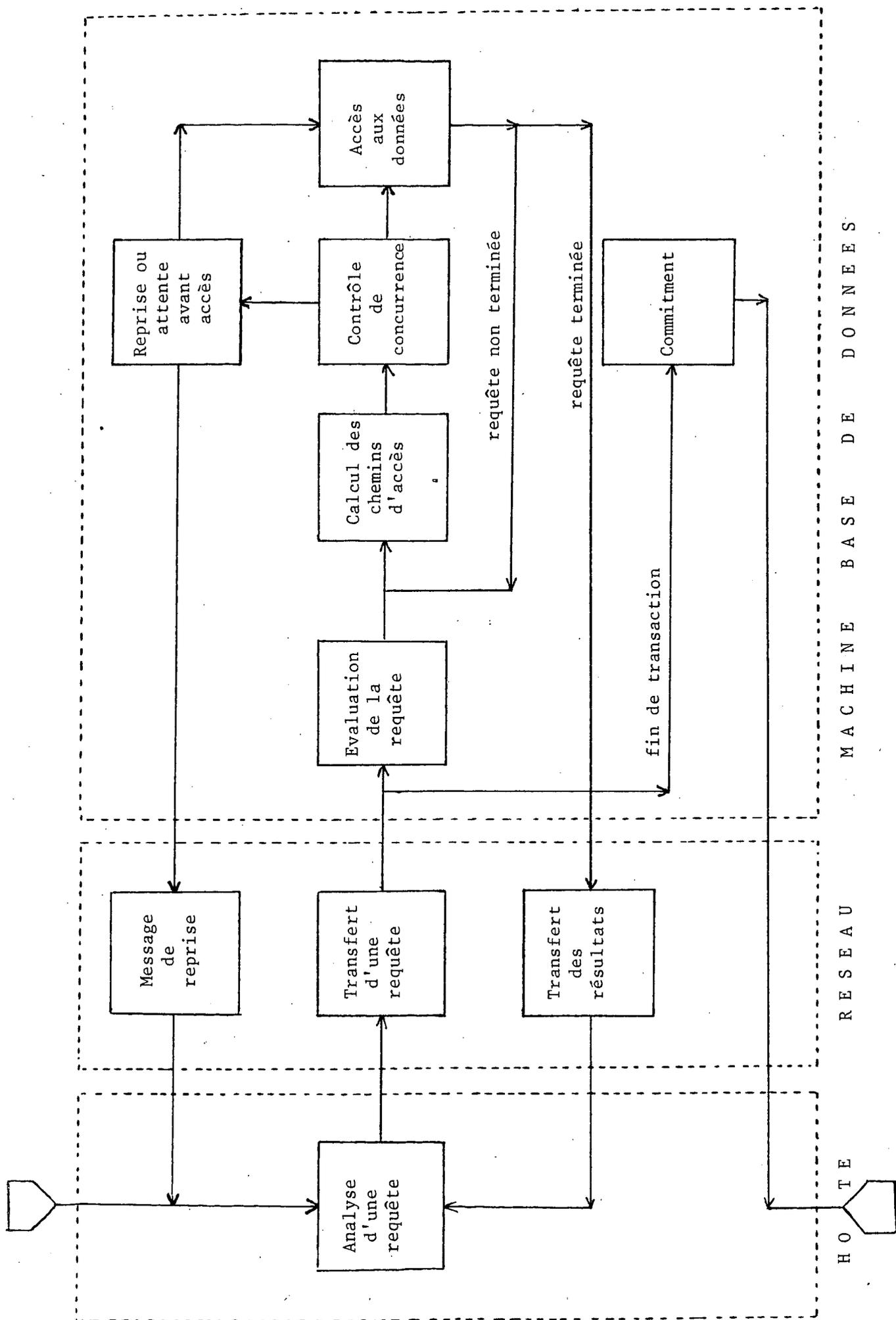


Figure 3.3. : Description du traitement d'une transaction

Le traitement d'une transaction comprend pour chaque requête la constituant:

- l'analyse de la requête dans le calculateur Hôte
- son transfert par le réseau
- l'évaluation de la requête et sa décomposition en accès par le PEQ

et pour chaque accès:

- le calcul des chemins d'accès par le PAR
- le contrôle de concurrence par le PCJ qui entraîne, soit la reprise de la transaction suivie d'un message de reprise, soit éventuellement une attente de l'exécution jusqu'au déblocage d'un verrou
- l'accès aux données qui est réalisé par le PMC et si nécessaire par le PJT.

Une fin de transaction provoque le commitment de cette transaction.

### 3.3- Réseau de files d'attente modélisant le traitement de transactions

A partir de la description précédente et en supposant que les processeurs PEQ, PAR et PCJ partagent le même processeur physique, nous obtenons le réseau de files d'attente présenté sur la figure 3.4. Nous allons décrire ce réseau en précisant:

- les types et disciplines de service des différentes stations
- la signification des files et leurs temps de service
- les transitions dans le réseau.

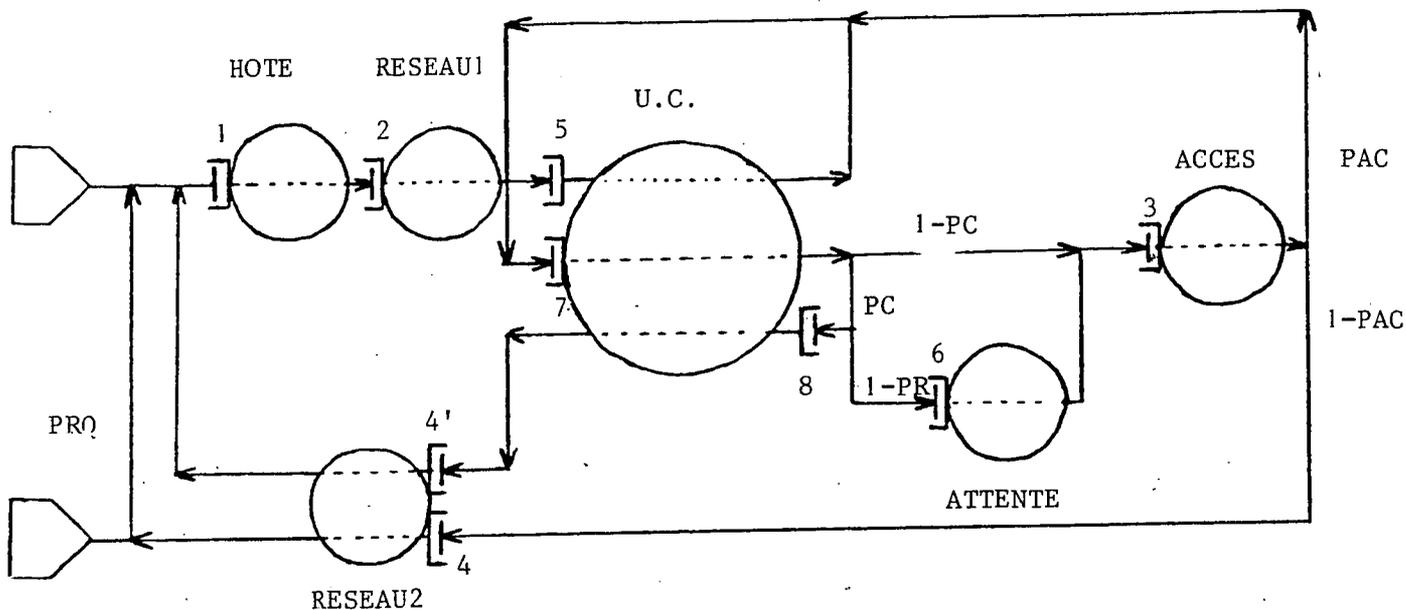


Figure 3.4: Réseau de files d'attente.

Le réseau de files d'attente est formé des 6 stations suivantes:

- HOTE qui modélise l'ensemble des terminaux hôtes qui est du type Serveur Infini (SI) ou délai
- RESEAU1 et RESEAU2 qui modélisent le réseau de communication, sont également du type SI (cette station est dédoublée simplement pour la clarté de la figure)
- U.C. qui modélise les processeurs PEQ, PAR et PCJ, est du type serveur unique et a Processor Sharing (PS) pour discipline de service. Rappelons que PS est le cas limite du Round Robin avec un overhead nul et un quantum tendant vers zéro.
- ACCES qui modélise l'accès aux données (PJT, PMC et mémoires secondaires) est du type Serveur Unique (SU) et a PS pour discipline de service, les processeurs PJT et PMC travaillant en temps partagé

- ATTENTE qui ne modélise pas un processeur, mais les files d'attente sur les verrous, est du type SI (il existe toujours une file pour accueillir une transaction en attente sur un verrou). Avant de préciser les temps de service des files, remarquons que la moyenne du temps de réponse d'un réseau formé de stations de types SI ou SU avec Processor Sharing, dépend uniquement des moyennes des temps de service et est indépendante de leurs lois. Le tableau suivant récapitule l'ensemble des caractéristiques des différentes files et stations du réseau.

No file	Signification	Station	Type	Service moyen
1	Réflexion de l'utilisateur et Analyse de la requête	HOTE	SI	30 s
2	Transfert d'une requête	RESEAU1	SI	0.2 s
3	Accès aux données	ACCES	SU	0.1 à 1 s
4	Transfert des résultats	RESEAU2	SI	2 s
4'	Transfert du message de reprise	RESEAU2	SI	0.2 s
5	Evaluation de la requête	U.C.	SU	0.1 s
6	Attente sur un verrou	ATTENTE	SI	?
7	Chemin d'accès et Concurrence	U.C.	SU	0.01s
8	Avortement de la transaction	U.C.	SU	0.1 s

Le temps d'attente moyen sur un verrou est égal au temps moyen d'exécution d'une transaction, soit encore le temps moyen de réponse divisé par le nombre moyen d'exécution (nombre de reprise + 1); une méthode de résolution par itération successive sera donc employée pour trouver le temps de réponse.

Les transitions dans le réseau sont données à l'aide des

probabilités suivantes:

PRQ : Probabilité de lancer une nouvelle requête

PAC : Probabilité d'effectuer un nouvel accès

PC : Probabilité d'avoir un conflit lors de l'accès

PR : Probabilité de reprise lors d'un conflit.

Les deux premières probabilités déterminent respectivement les boucles sur chaque requêtes et sur chaque accès. Après le calcul des chemins d'accès et du contrôle de concurrence (file 7), trois transitions sont possibles:

- s'il n'y a pas conflit (probabilité  $1-PC$ ), vers la file 3
- s'il y a conflit et reprise (probabilité  $PC.PR$ ), vers la file 8
- s'il y a conflit et attente sur un verrou (probabilité  $PC(1-PR)$ ), vers la file 6.

La probabilité de conflit  $PC$  est une caractéristique de l'ensemble des transactions soumises au système. La probabilité de reprise  $PR$  est fonction de la technique de contrôle de concurrence utilisée. Dans le cas de l'Estampillage, dès qu'il y a conflit, il y a reprise et  $PR=1$ . Par contre dans le cas du Verrouillage avec prévention des verrous mortels, s'il y a conflit la transaction est seulement autorisée à attendre une transaction plus jeune qu'elle, et ceci se produit avec la probabilité  $1/2$ , et donc  $PR=1/2$ .

C'est un réseau fermé qui sera utilisé pour comparer les deux techniques de contrôle de concurrence. En effet, la technique de Verrouillage, bloquant des transactions, a tendance

a faire croître le nombre de clients d'un réseau ouvert, ce qui rend la comparaison avec l'Estampillage illusoire. Une autre justification de l'étude d'un réseau fermé est que le contrôle de concurrence prend d'autant plus d'importance que le degré de concurrence est élevé, c'est à dire que la charge est élevée; tous les systèmes ayant un nombre de portes d'entrée limité, nous étudierons le système a pleine charge: une fin de transaction sera aussitôt remplacée par le début d'une nouvelle transaction émise par le même usager ou par un autre qui l'aura immédiatement remplacé.

En ce qui concerne l'obtention du temps de réponse moyen du réseau, nous pouvons remplacer le réseau par un réseau équivalent composé des mêmes stations où chaque file a pour temps de service celui de l'ancienne file correspondante multiplié par le nombre moyen de visites. Ceci est justifié par le fait que les stations du réseau vérifient les hypothèses du théorème de B.C.M.P. [BASK75] [SAUES1].

#### 4- Méthode de résolution

Le temps de réponse moyen pour une transaction est calculé en connaissant:

- la probabilité de conflit PC,
- la probabilité de reprise PR,
- la loi de distribution du nombre de requêtes par transaction,
- la loi de distribution du nombre d'accès par requête.

Le calcul passe d'abord par la définition de classes de clients dans les files pour prendre en compte les lois de distribution des nombres de requêtes par transaction et d'accès par requêtes. Le temps de réponse moyen est obtenu en appliquant de façon itérative l'algorithme de Reiser "Mean Value Analysis" (les itérations sont nécessaires pour connaître le temps d'attente moyen, lui-même fonction du temps de réponse total). Pour appliquer cet algorithme, les nombres moyens de visites par files doivent être calculés; ces nombres sont obtenus par sommation des nombres de visites par classes solutions d'un système infini d'équations linéaires, ces équations se déduisant des transitions de classes.

Si PRQ et PAC sont indépendants du numéro de la requête et de l'accès, comme nous avons laissé entendre dans la section 3.3, les nombres de requêtes et d'accès suivent des lois géométriques de distribution. Pour pouvoir accepter d'autres distributions, PRQ et PAC doivent dépendre du numéro de, respectivement, la requête et l'accès. Il faut alors pouvoir repérer une

transaction à un instant donné dans le réseau. La position d'une transaction sera définie par le quadruplet  $(n,p,q,a)$  où:

- $n$  est le numéro de la file d'attente,
- $p$  est le nombre de reprise,
- $q$  est le numéro de la requête,
- $a$  est le numéro de l'accès (ce numéro est égal à 0 pour les files 1, 2, 4 et 5).

Nous avons ainsi des classes de clients dans les files. Les transitions de classes sont les suivantes:

- $(1,p,q,0) \text{ ---> } (2,p,q,0)$
- $(2,p,q,0) \text{ ---> } (5,p,q,0)$
- $(5,p,q,0) \text{ ---> } (7,p,q,1)$
- $(7,p,q,a) \text{ ---> } (8,p,q,a)$  avec la probabilité:  $PC \cdot PR$
- $\text{---> } (6,p,q,a)$  avec la probabilité:  $PC(1-PR)$
- $\text{---> } (3,p,q,a)$  avec la probabilité:  $1-PC$
- $(8,p,q,a) \text{ ---> } (4',p,q,0)$
- $(6,p,q,a) \text{ ---> } (3,p,q,a)$
- $(3,p,q,a) \text{ ---> } (7,p,q,a+1)$  avec la probabilité:  $PAC$
- $\text{---> } (4,p,q,0)$  avec la probabilité:  $1-PAC^a$
- $(4,p,q,0) \text{ ---> } (1,p,q+1,0)$  avec la probabilité:  $PRQ^a$
- $\text{---> } (1,1,1,0)$  avec la probabilité:  $1-PRQ^q$
- $(4',p,q,0) \text{ ---> } (1,p+1,1,0)$   $^q$

Posons  $e(n,p,q,a)$  le nombre de visites de la classe  $(n,p,q,a)$ . Les transitions précédentes permettent d'obtenir des équations vérifiées par les nombres de visites. Par exemple, la transition:

$(4,p,q,0) \text{ ---> } (1,p,q+1,0)$  avec la probabilité:  $\frac{PRQ}{q}$   
peut s'écrire si  $q > 1$  :

$(4,p,q+1,0) \text{ ---> } (1,p,q,0)$  avec la probabilité:  $\frac{PRQ}{q}$

on a donc :  $e(1,p,q,0) = \frac{PRQ}{q-1} e(4,p,q-1,0)$

Un deuxième exemple peut être donné par la transition :

$(8,p,q,a) \text{ ---> } (4',p,q,0)$

Cette transition étant vraie pour tout  $a$ , on obtient :

$$e(4',p,q,0) = \sum_{a>0} e(8,p,q,a)$$

En utilisant toutes les transitions, le système d'équations linéaires suivant est obtenu :

(1) pour  $q > 1$

$$e(1,p,q,0) = PRQ \sum_{q-1} e(4,p,q-1,0)$$

(1b) pour  $p > 1$

$$e(1,p,1,0) = \sum_{q>0} \sum_{a>0} e(4',p-1,q,a)$$

$$(1t) \quad e(1,1,1,0) = \sum_{p>0} \sum_{q>0} (1 - PRQ) \sum_q e(4,p,q,0)$$

$$(2) \quad e(2,p,q,0) = e(1,p,q,0)$$

$$(3) \quad e(3,p,q,a) = (1 - PC) e(7,p,q,a) + e(6,p,q,a)$$

$$(4) \quad e(4,p,q,0) = \sum_{a>0} (1 - PAC) \sum_a e(3,p,q,a)$$

$$(4') \quad e(4',p,q,0) = \sum_{a>0} e(8,p,q,a)$$

$$(5) \quad e(5,p,q,0) = e(2,p,q,0)$$

$$(6) \quad e(6,p,q,a) = PC (1 - PR) e(7,p,q,a)$$

(7) pour  $a > 1$

$$e(7,p,q,a) = PAC \sum_{a-1} e(3,p,q,a-1)$$

$$(7b) \quad e(7,p,q,1) = e(5,p,q,0)$$

$$(8) \quad e(8,p,q,a) = PC PR e(7,p,q,a)$$

Remarquons que l'équation (1t) est redondante, elle exprime simplement que le réseau est fermé et que le flux entrant est égal au flux sortant. Il faut donc fixer la valeur d'une inconnue pour obtenir une solution unique du système. Il convient de fixer  $e(1,1,1,0)$  à 1 pour trouver le nombre de

visites de chaque classe au cours du traitement complet d'une transaction.

On obtient, par la méthode de substitution, les solutions suivantes (cf Annexe A pour la résolution complète):

$$e(1,p,q,0) = (1 - T)^{p-1} S^{q-1} \left( \sum_{k=1}^{q-1} PRQ_k \right)$$

$$e(2,p,q,0) = e(5,p,q,0) = e(1,p,q,0)$$

$$e(7,p,q,a) = (1 - PC.PR)^{a-1} \left( \sum_{i=1}^{a-1} PAC_i \right) e(1,p,q,0)$$

$$e(3,p,q,a) = (1 - PC.PR) e(7,p,q,a)$$

$$e(8,p,q,a) = PC.PR e(7,p,q,a)$$

$$e(6,p,q,a) = PC (1 - PR) e(7,p,q,a)$$

$$e(4',p,q,0) = \sum_{a>0} PC.PR e(7,p,q,a)$$

$$e(4,p,q,0) = S e(1,p,q,0)$$

$$\text{Où } S = G_{AC} (1 - PC.PR) \text{ et } T = G_{RQ} (S)$$

avec  $G_{AC}$  la fonction génératrice de la loi du nombre d'accès

et  $G_{RQ}$  la fonction génératrice de la loi du nombre de requêtes

A partir de ces nombres, on trouve les nombres de visites par files. Si  $E(n)$  est le nombre de visites de la file  $n$ , alors:

$$E(n) = \sum_{p>0} \sum_{q>0} \sum_{a>0} e(n,p,q,a)$$

On obtient tous calculs faits (cf Annexe B):

$$E(1) = \frac{1}{T} \cdot \frac{1-T}{1-S} \quad \text{ou } \overline{RQ} \quad \text{si } PC.PR=0$$

$$E(2) = E(4) = E(5) = E(1)$$

$$E(3) = \frac{1 - PC.PR}{T} \cdot \frac{1 - T}{PC.PR} \quad \text{ou } \overline{RQ.AC} \quad \text{si } PC.PR=0$$

$$E(7) = \frac{1}{T} \cdot \frac{1 - T}{PC.PR} \quad \text{ou } PC.\overline{RQ.AC} \quad \text{si } PC.PR=0$$

$$E(8) = \frac{1 - T}{T} \quad \text{ou } 0 \quad \text{si } PC.PR=0$$

Les valeurs de ces nombres pour  $PR=0$  sont dépendantes seulement des moyennes des nombres de requêtes et d'accès. Ceci veut dire qu'une technique de contrôle de concurrence n'occasionnant pas de reprises, induit un temps de réponse moyen indépendant de la loi de distribution de ces nombres. Ainsi, l'Estampillage sera beaucoup plus sensible à ces lois que le Verrouillage avec prévention des verrous mortels.

D'autre part,  $E(8)$  est le nombre moyen de reprises, et  $E(8)+1$  sera le nombre moyen d'exécutions; donc, le temps moyen d'attente qui est égal au temps moyen d'une exécution est égal au temps de réponse moyen divisé par  $E(8)+1$ .

## 5- Résultats et conclusions

Rappelons que nous trouvons le temps de réponse moyen en fonction de:

- la méthode de contrôle de concurrence (Verrouillage ou Estampillage)
- la probabilité de conflit (PC)
- le nombre de transactions dans le système (n)
- la moyenne et la loi de distribution du nombre de requêtes par transaction
- la moyenne et la loi de distribution du nombre d'accès par requête.

Nous présentons des courbes du temps de réponse moyen en fonction de la probabilité de conflit, les quatre derniers paramètres ayant été fixés. Après avoir décrit l'allure générale des courbes, les résultats sont discutés en fonction des moyennes et des lois de distribution des nombres de requêtes et d'accès.

L'allure générale des courbes est la même pour les différentes valeurs du nombre de transactions (n), comme l'on peut s'en rendre compte sur les courbes de la figure 5.1 qui ont été obtenues pour des nombres de requêtes et d'accès constants et égaux à 1 (Ces courbes sont tracées à partir des valeurs regroupées dans le tableau de la figure 5.2). En ce qui concerne la comparaison estampillage/verrouillage, il n'y a, bien sur, pas de différence pour de faibles probabilités de conflit, le contrôle de concurrence n'intervenant que très peu. Par contre,

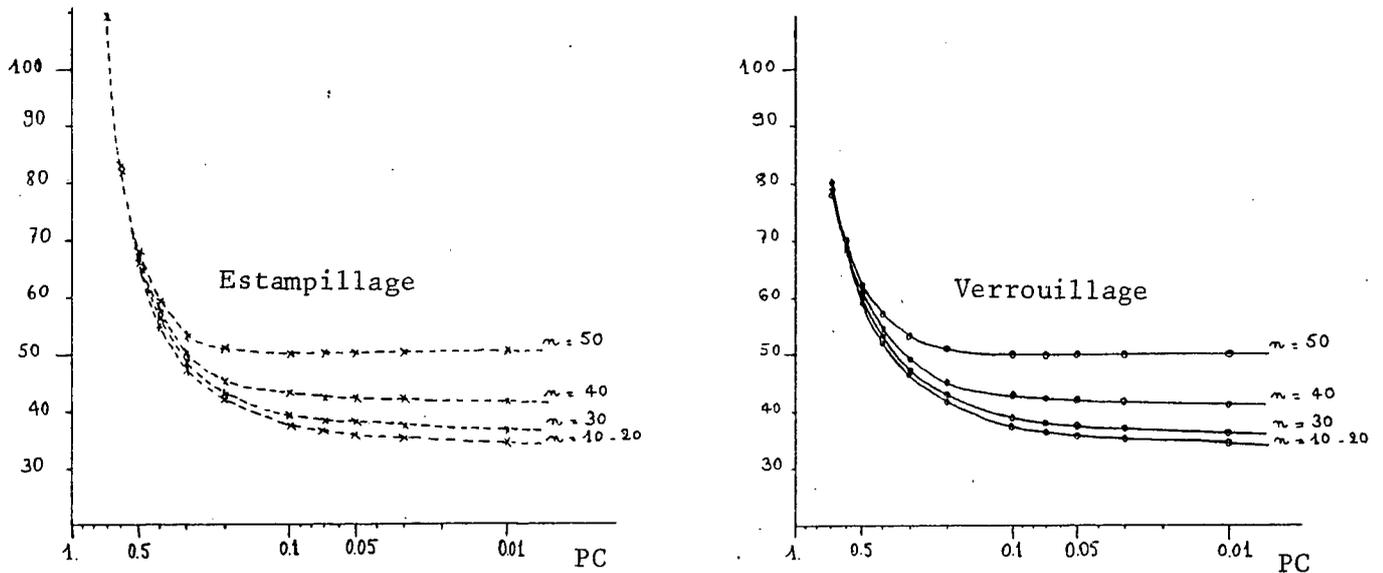


Figure 5.1: Temps moyen de réponse en secondes en fonction de la probabilité de conflit (PC) pour une requête par transaction et un accès par requête.

n \ PC	10		20		30		40		50	
	Est	Ver								
0.00	34	34	34	34	36	36	41	41	50	50
0.01	34	34	35	35	37	36	41	41	50	50
0.03	35	35	35	35	37	37	42	42	50	50
0.05	35	35	36	36	38	38	42	42	50	50
0.07	36	36	37	37	38	38	42	42	50	50
0.10	37	37	38	38	39	39	43	43	50	50
0.20	42	41	42	42	43	43	45	45	51	51
0.30	47	46	48	47	48	47	50	49	53	53
0.40	55	52	55	52	56	53	57	54	59	57
0.50	66	59	66	59	66	60	67	61	68	62
0.60	82	67	82	68	82	68	83	69	83	70
0.70	109	78	109	78	109	79	109	79	109	80

Figure 5.2: Temps de réponse moyen en fonction de la probabilité de conflit (PC), du nombre de transactions (n), avec des nombre de requêtes et d'accès égaux à 1.

pour de plus grande valeur de cette probabilité, la courbe obtenue avec l'estampillage se caractérise par une très forte pente. Ceci s'explique par le fait que cette méthode conduit, en cas de forte probabilité de conflit, à un nombre de reprises de transactions tel qu'il implique une très forte croissance du temps de réponse moyen. Pour ces grandes valeurs de la probabilité de conflit, le verrouillage se comporte beaucoup mieux.

Pour des nombres de requêtes et d'accès égaux à 1, le verrouillage est meilleur que l'estampillage quelques soient les valeurs des autres paramètres (cf figure 5.2). Si le nombre moyen de requêtes est supérieur à 1 alors les courbes obtenues pour les deux méthodes se croisent de façon telle qu'en cas de probabilité de conflit faible, l'estampillage devient meilleur que le verrouillage (cf figure 5.3). Ceci s'explique par le fait que l'estampillage est, d'une certaine manière, plus optimiste que le verrouillage, et que par conséquent, il doit donner de meilleurs résultats en cas de conflits peu fréquents. Mais il faut remarquer que les différences entre les deux méthodes restent faibles si les lois de distribution des nombres de requêtes et d'accès sont constantes.

Lorsque les lois de distribution des nombres d'accès et de requêtes suivent une loi géométrique, l'estampillage devient, de façon beaucoup plus significative, meilleur que le verrouillage, et ceci reste vrai même si le nombre moyen de requêtes est égal à

Estampillage: -x-x-x-

Verrouillage: -o-o-o-

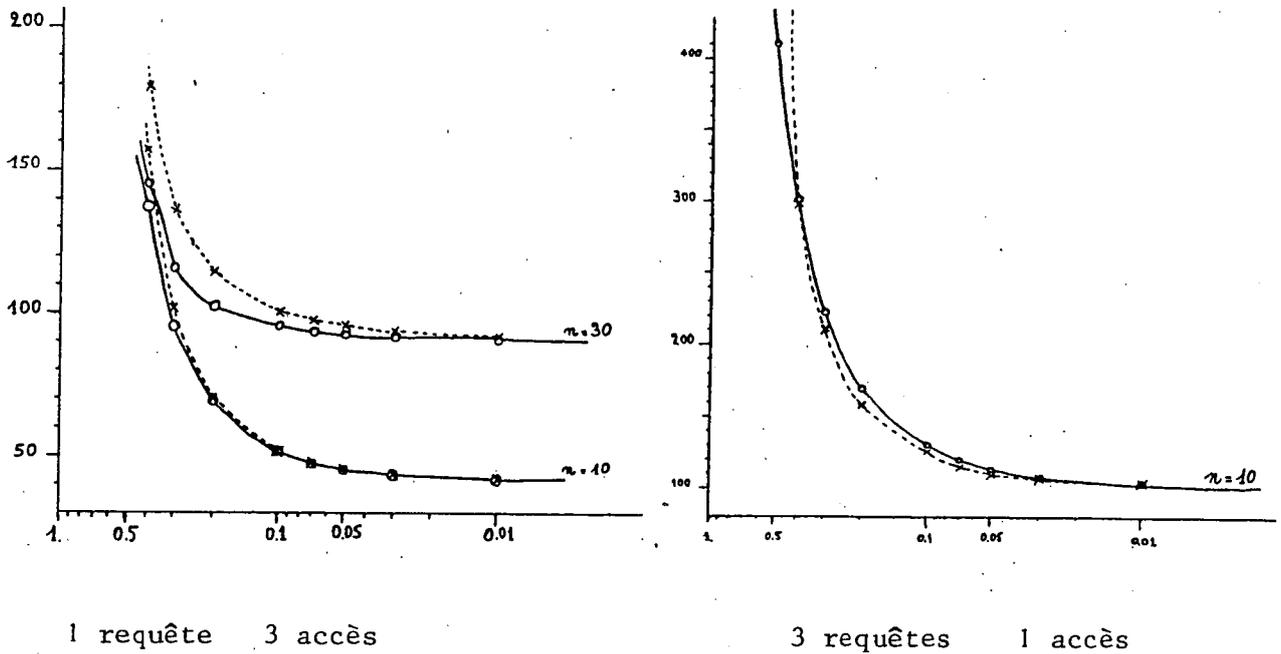


Figure 5.3: Temps de réponse moyen avec des lois de distribution des nombres de requêtes et d'accès constantes.

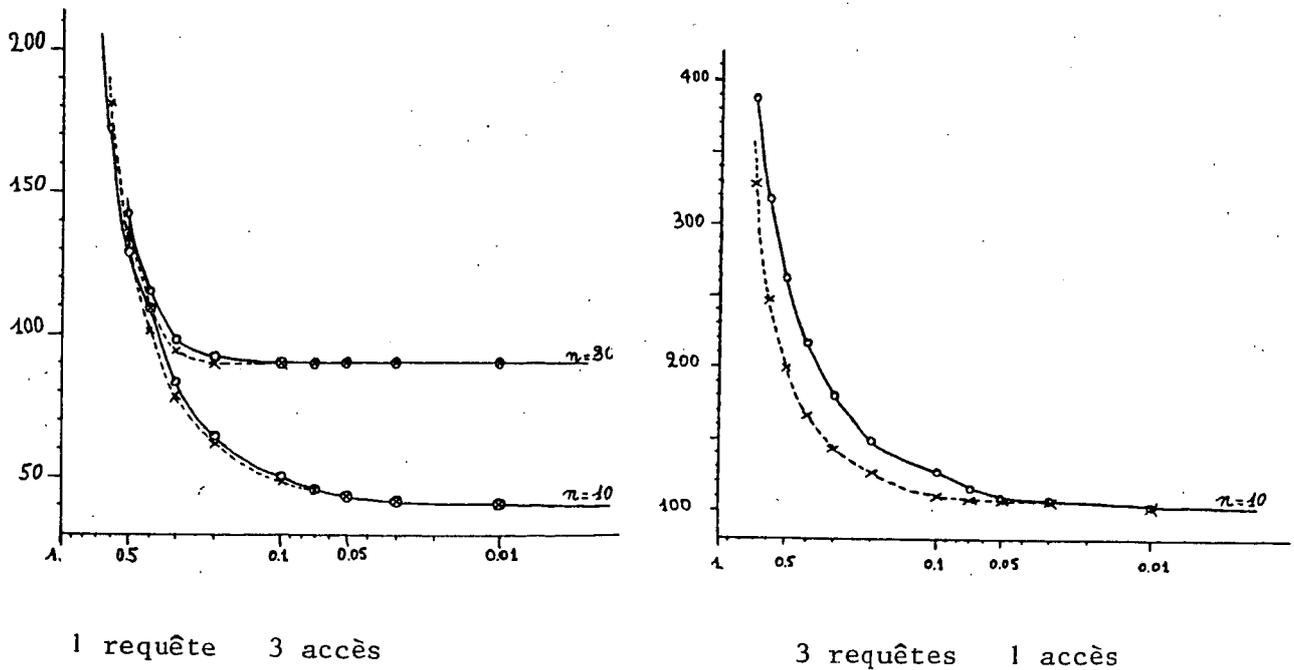


Figure 5.4: Temps de réponse moyen avec des lois de distribution des nombres de requêtes et d'accès géométriques.

1 (cf figure 5.4). Cette dernière observation s'explique par le fait qu'en cas de distribution géométrique, il y a une forte proportion de petites transactions et que les reprises deviennent moins nombreuses et moins coûteuses, ce qui avantage l'estampillage; d'autre part, le verrouillage "pur" (sans prévention des verrous mortels) est insensible aux lois de distribution des nombres de requêtes et d'accès et ne dépend que de la moyenne (cf section 4).

En résumé, la technique de verrouillage deux phases avec prévention des verrous mortels donne un meilleur temps de réponse moyen que l'estampillage si les nombres moyens de requêtes par transaction et d'accès par requêtes sont petits, ou si la probabilité de conflit est très forte. L'estampillage devient meilleur si les nombres moyens de requêtes et d'accès augmentent. Si, de plus, leurs lois de distribution sont géométriques alors l'estampillage est très significativement meilleur que le verrouillage.

Remerciements: Je tiens à remercier Georges Gardarin, Philippe Bernadat, Patrick Valduriez, Yann Viémont ainsi que les autres membres de l'équipe SABRE pour les discussions fructueuses que nous avons eues.

- [IRAN79] Irani K.B. and Lin H., "Queing Network models for Concurrent Transaction Processing in a Database System", ACM-SIGMOD, May 1979.
- [POTI80] Potier D. and Leblanc Ph., "Analysis of Locking Policies in Database Management Systems", Communications of the ACM, Vol 23, No 10, Oct 80.
- [RIES77] Ries D.R. and Stonebraker M., "Effect of Locking Granularity in a Database Management System", ACM Transactions on DBS, Vol 2, No 3, p 233-246, Sept 1977.
- [RIES79] Ries D.R. and Stonebraker M., "Locking Granularity Revised", ACM Transactions on DBS, Vol 4, No 2, p 210-227, June 1979.
- [ROSE78] Rosenkrantz D.J., Stearns R.E., Lewis P.W., "System Level Concurrency Control for Distributed Database Systems", ACM TODS, Vol 2, No 3, P 233-246, Sept 1978.
- [SAUE81] Sauer C.H. and Chandy K.M., "Computer systems performance modeling", Prentice-Hall (1981).
- [THOM79] Thomas R.H., "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases", ACM TODS, Vol 4, No 2, P 180-209, June 1979.

## Références bibliographiques

- [BASK75] Baskett, Chandy, Muntz, Palacios, "Open, closed and mixed networks with different classes of customers", JACM, Vol 22, No 2, P 247-260, April 1975.
- [BERN80] Bernstein P.A., Goodman N., "Timestamp Based Algorithms for Concurrency Control in DDBMS", 6th Very Large Data Bases, Montréal, Oct 1980.
- [ESWA76] Eswaran K.P., Gray J.N., Lorie R.A., Traiger L.L., "The Notion of Consistency and Predicate Locks in a Database System", Comm. of the ACM, Vol 19, No 11, P 624-633, Nov 1976.
- [BERN81] Bernstein P.A., Goodman N., "Concurrency Control in Distributed Database Systems", ACM Computing surveys, Vol 13, No 2, June 1981, P 185-222.
- [GARC79] Garcia Molina H., "Performance of Update Algorithms for Distributed Database", Ph.D. Thesis, Stan.Cs.79.744 Stanford University, June 1979.
- [GARD81] Gardarin G., Melkanoff M., "Concurrency Control Principles in Distributed and Centralized Databases", Rapport de Recherche INRIA No 113, Jan 1981.
- [GARD82] Gardarin G., Bernadat P., Temmerman N., Valduriez P., Viémont Y., "SABRE: A Relational database system for a multi-microprocessor machine", Proc. of the int. workshop on database machine, San Diego, California, Aug 1982.

Annexe A: Résolution du système d'équations

donnant les nombres de visites par classe.

Les équations (2), (5) et (7b) donnent:

$$(9) \quad e(1,p,q,0) = e(2,p,q,0) = e(5,p,q,0) = e(7,p,q,1)$$

(3) et (6) donnent:

$$(10) \quad e(3,p,q,a) = [ (1-PC) + PC(1-PR) ] e(7,p,q,a) \\ = (1 - PC.PR) e(7,p,q,a)$$

en reportant dans (7), on obtient pour  $a > 1$ :

$$e(7,p,q,a) = (1 - PC.PR) PAC_{a-1} e(7,p,q,a-1)$$

donc, par itération:

$$e(7,p,q,a) = (1 - PC.PR)^{a-1} \left( \sum_{i=1}^{a-1} PAC_i \right) e(7,p,q,1)$$

et en utilisant (9):

$$(11) \quad e(7,p,q,a) = (1 - PC.PR)^{a-1} \left( \sum_{i=1}^{a-1} PAC_i \right) e(1,p,q,0)$$

(remarquons que cette équation est vraie pour tout  $a$ )

en reportant dans (10), on trouve:

$$e(3,p,q,a) = (1 - PC.PR)^a \left( \sum_{i=1}^{a-1} PAC_i \right) e(1,p,q,0)$$

(1) et (4) donnent pour  $q > 1$  :

$$e(1,p,q,0) = PRQ_{q-1} \sum_{a>0} (1-PAC)_a (1-PC.PR)^a \left( \sum_{i=1}^{a-1} PAC_i \right) e(1,p,q-1,0)$$

en posant  $S = \sum_{a>0} (1-PC.PR)^a (1-PAC)_a \left( \sum_{i=1}^{a-1} PAC_i \right)$  il vient:

$$e(1,p,q,0) = S PRQ_{q-1} e(1,p,q-1,0)$$

et, par itération, on obtient l'équation (12) vraie pour tout  $q$ :

$$(12) \quad e(1,p,q,0) = S^{q-1} \left( \sum_{k=1}^{q-1} PRQ_k \right) e(1,p,1,0)$$

D'autre part, (1b), (4') et (8) donnent pour  $p > 1$ :

$$e(1,p,1,0) = \sum_{q>0} \sum_{a>0} PC PR e(7,p-1,q,a)$$

et avec (11) et (12) :

$$(13) \quad e(1,p,1,0) = \sum_{q>0} \sum_{a>0} PC \cdot PR (1-PC \cdot PR)^{a-1} \left( \sum_{i=1}^{a-1} PAC \right) S^{q-1} \left( \sum_{k=1}^{q-1} PRQ \right) e(1,p-1,1,0)$$

Pour réduire cette dernière équation, démontrons le lemme suivant:

$$S = G_{AC} (1 - PC \cdot PR) \quad \text{où } G \text{ est la fonction génératrice de } AC$$

$$\text{la loi du nombre d'accès: } G_{AC}(x) = \sum_{n>0} x^n P[\text{nombre d'accès} = n]$$

$$\text{Il est facile de voir que: } (1-PAC) \left( \sum_{i=1}^{a-1} PAC \right) = P[\text{nombre d'accès} = a]$$

et, de la définition de la fonction génératrice, il vient:

$$S = \sum_{a>0} (1-PC \cdot PR)^a P[\text{nombre d'accès} = a] = G_{AC} (1-PC \cdot PR)$$

Q.E.D

$$\text{D'autre part, calculons: } \sum_{n>0} x^{n-1} P[A>=n]$$

où A est une variable aléatoire entière positive.

$$\begin{aligned} \sum_{n>0} x^{n-1} P[A>=n] &= \sum_{n>0} x^{n-1} \left( \sum_{i>=n} P[A=i] \right) \\ &= \sum_{i>0} \left( \sum_{n=0}^{i-1} x^n \right) P[A=i] \end{aligned}$$

Si  $x=1$

$$\sum_{n>0} x^{n-1} P[A>=n] = \bar{A} = G'_A(1) = \text{moyenne de } A$$

Si  $x \neq 1$

$$\sum_{n>0} x^{n-1} P[A>=n] = \sum_{i>0} \frac{1-x}{1-x} x^i P[A=i] = \frac{1}{1-x} (1 - G_A(x))$$

( notons que cette fonction de x est continue en 1 )

En posant  $G_{RQ}$  la fonction génératrice du nombre de requêtes

$$\text{et } T = G_{RQ}(S) = G_{RQ}(G_{AC}(1-PC \cdot PR)), \text{ l'équation (13) se}$$

simplifie et devient:

$$e(1,p,1,0) = PC.PR \frac{1-T}{1-S} \frac{1-S}{PC.PR} e(1,p-1,1,0)$$

$$e(1,p,1,0) = (1-T)^{p-1} e(1,p-1,1,0)$$

Cette dernière équation est vraie quelque soit x et si S=1 du fait de la continuité de la fonction de x précédemment définie.

Les équations (4), (10) et (11) donnent:

$$e(4,p,q,0) = \sum_{a>0} (1-PC.PR)^a (1-PAC) \left( \sum_{i=1}^{a-1} PAC \right) e(1,p,q,0)$$

$$e(4,p,q,0) = S e(1,p,q,0)$$

Finalement, en posant  $e(1,1,1,0) = 1$ , nous trouvons le nombre de visite de chaque classe pour le traitement complet d'une transaction :

$$e(1,p,q,0) = (1-T)^{p-1} S^{q-1} \left( \sum_{k=1}^{q-1} PRQ \right)$$

$$e(2,p,q,0) = e(5,p,q,0) = e(1,p,q,0)$$

$$e(7,p,q,a) = (1-PC.PR)^{a-1} \left( \sum_{i=1}^{a-1} PAC \right) e(1,p,q,0)$$

$$e(3,p,q,a) = (1-PC.PR) e(7,p,q,a)$$

$$e(8,p,q,a) = PC.PR e(7,p,q,a)$$

$$e(6,p,q,a) = PC(1-PR) e(7,p,q,a)$$

$$e(4',p,q,0) = \sum_{a>0} PC.PR e(7,p,q,a)$$

$$e(4,p,q,0) = S e(1,p,q,0)$$

A titre de vérification, nous pouvons calculer:

$$\sum_{p>0} \sum_{q>0} (1-PRQ) e(4,p,q,0)$$

qui doit être égal à 1 pour vérifier (1t).

Cette expression est égale à :

$$\sum_{p>0} \sum_{q>0} (1 - T)^{p-1} S^q (1 - \frac{PRQ}{q}) \left( \sum_{k=1}^{q-1} \frac{PRQ}{k} \right)$$

soit:  $\sum_{p>0} (1 - T)^{p-1} T = 1$  Q.E.D

Annexe B: Calcul des nombres moyens de visites par files

$E(n)$  est le nombre moyen de visites de la file  $n$ .

$$E(n) = \sum_{p>0} \sum_{q>0} \sum_{a>0} e(n,p,q,a)$$

$$E(1) = \sum_{p>0} \sum_{q>0} (1 - T)^{p-1} S^{q-1} \left( \sum_{k=1}^{q-1} PRQ \right)$$

Si  $S \neq 1$  :

$$E(1) = \sum_{p>0} (1 - T)^{p-1} \frac{1 - T}{1 - S} \quad (\text{cf lemme annexe A})$$

$$E(1) = \frac{1}{T} \cdot \frac{1 - T}{1 - S} \quad (T \neq 0 \text{ car } PC < 1)$$

Si  $S = 1$  :

$$E(1) = \overline{RQ} = \text{nombre moyen de requêtes par transaction}$$

$$E(2) = E(5) = E(4) = E(1)$$

$$E(7) = \sum_{p>0} \sum_{q>0} \sum_{a>0} (1 - PC \cdot PR)^{a-1} \left( \sum_{i=1}^{a-1} PAC \right) e(1,p,q,0)$$

$$= \sum_{p>0} \sum_{q>0} \frac{1 - S}{PC \cdot PR} e(1,p,q,0)$$

$$= \frac{1 - S}{PC \cdot PR} \cdot \frac{1}{T} \cdot \frac{1 - T}{1 - S}$$

$$E(7) = \frac{1}{T} \cdot \frac{1 - T}{PC \cdot PR} \quad \text{ou} \quad \overline{RQ} \cdot \overline{AC} \quad \text{si } PC \cdot PR = 0$$

$$E(8) = PC \cdot PR \cdot E(7) = \frac{1 - T}{T} \quad (\text{vrai } \forall PC \cdot PR)$$

$$E(3) = (1 - PC \cdot PR) \cdot E(7)$$

$$E(3) = \frac{1 - PC \cdot PR}{T} \cdot \frac{1 - T}{PC \cdot PR} \quad \text{ou} \quad \overline{RQ} \cdot \overline{AC} \quad \text{si } PC \cdot PR = 0$$

$$E(6) = \frac{PC(1 - PR)}{T} \cdot \frac{1 - T}{PC \cdot PR} \quad \text{ou} \quad PC \cdot \overline{RQ} \cdot \overline{AC} \quad \text{si } PC \cdot PR = 0$$



