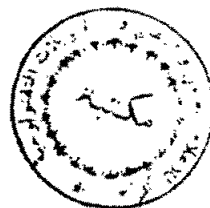


laboratoire de recherche
en informatique
et automatique

Analyse d'un algorithme
de gestion simultanée.
Mémoire centrale,
disque de pagination.

E. Gelenbe, J. Lenfant,
A. Brandwajn, D. Potier.



Rapport de Recherche n°19

mai 1973

C. 535

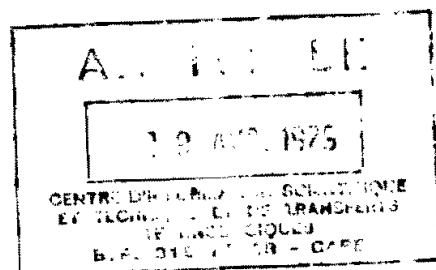
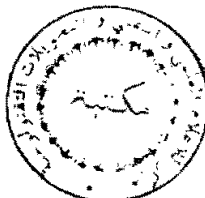
**ANALYSE D'UN ALGORITHME DE GESTION SIMULTANÉE
MÉMOIRE CENTRALE - DISQUE DE PAGINATION**

**E. Gelenbe - J. Lenfant*
A. Brandwajn - D. Potier**

RÉSUMÉ

L'optimisation de la gestion d'un disque sur lequel les pages ont des images fixes et l'optimisation du remplacement à la demande de pages de la mémoire centrale mettent en évidence des exigences contradictoires. Dans cette étude une nouvelle politique-globale-de gestion d'un tel système de pagination est analysée sur un modèle mathématique.

Pour cela, une classe à un paramètre d'algorithmes de remplacement qui choisissent la page à remplacer non seulement en fonction de l'état de la mémoire centrale et du comportement du programme, mais aussi en fonction de la position instantanée du disque est introduite. Ses performances sont évaluées en monoprogrammation. Les résultats théoriques sont illustrés d'exemples numériques.



* *Département de Mathématiques et Informatique
Université de RENNES - 35031 RENNES-CEDEX.*

BIBLIOTHEQUE DU CERIST

TABLE DES MATIERES

I - INTRODUCTION

I.1. La gestion des mémoires paginées	1
I.2. La gestion simultanée des disques et de la mémoire centrale	3
I.3. Plan de l'étude	4

II - LES ALGORITHMES DE REMPLACEMENT DE PAGE ALÉATOIRES A PRECHARGEMENT PARTIEL

II.1. Le modèle de programme à références indépendantes	5
II.2. Définition formelle des ARPAPP	6
II.3. Taux de défauts de page des ARPAPP	9
II.4. Les ARPAPP optimaux	11

III - GESTION DU DISQUE DE PAGINATION

III.1. Introduction	12
III.2. Calcul du temps moyen de service sans tampon	14
III.3. Calcul du temps moyen de service avec tampon	19

IV - RESULTATS NUMERIQUES

IV.1. Introduction	25
IV.2. Détermination de la valeur optimale de ρ	25
IV.3. Utilité du tampon	27

<u>APPENDICE</u>	29
------------------	----

<u>REFERENCES</u>	32
-------------------	----

<u>FIGURES</u>	
----------------	--

I - INTRODUCTION

I.1. - LA GESTION DES MEMOIRES PAGINEES

Dès les débuts de l'informatique, les programmeurs ont essayé de disposer d'un espace adressable considérablement plus vaste que ne l'est la mémoire centrale. L'information ne doit se trouver en mémoire centrale qu'au moment d'être traitée par les processeurs ; sinon elle peut résider dans des mémoires auxiliaires à accès plus lent mais moins coûteuses. Dans de nombreux systèmes d'exploitation récents - ceux qui utilisent le concept de mémoire virtuelle - c'est le système lui-même qui contrôle la répartition de l'information entre les différents supports disponibles et qui décide de l'opportunité des transferts. Le choix de cette stratégie d'allocation est crucial pour l'efficacité du calculateur.

Le procédé le plus fréquemment utilisé pour réaliser une mémoire virtuelle est la pagination. L'espace adressable de chaque programme est divisé en blocs de même taille, les "pages". La mémoire centrale est elle-même divisée en "pages physiques" pouvant chacune être le support d'une page du programme. Un dispositif câblé permet de traduire toute adresse logique en une adresse physique.

La partie de l'information d'un programme qui ne se trouve pas en mémoire centrale est souvent rangée sur un disque à secteurs ayant une tête de lecture/écriture par piste (ou sur un tambour, ce qui est équivalent du point de vue de l'adressage). La portion d'une piste située à l'intérieur d'un secteur constitue l'unité de mémoire adressable et transférable sur ce type de dispositifs. Pour de plus amples détails on pourra se reporter, à titre d'exemple, à [1] .

C'est dans cet environnement "hardware" que sont définies les stratégies que cette étude se propose d'analyser. Nous considérerons l'espace adressable N d'un programme comme un ensemble de n pages : $N = \{p_1, \dots, p_n\}$

A ce programme sont attribuées m ($m < n$) pages physiques en mémoire centrale. Pendant l'exécution, un "défaut de page" se produit lorsqu'une page non présente en mémoire centrale est référencée par le programme. Si les m pages physiques ne sont pas toutes occupées, la page demandée est alors transférée vers l'une d'entre elles, choisie de manière quelconque ; sinon un "algorithme de remplacement de page" (ARP) détermine celle des m pages physiques qui recevra la page demandée. Ce transfert doit être précédé d'un transfert - en sens inverse - de la page qui occupait l'emplacement physique sélectionné, au moins si cette page a été modifiée. Lorsque les seuls instants où une décision de remplacement de page peut être prise sont ceux des défauts de page, l'ARP est dit "à la demande".

Il est clair que l'influence des défauts de page sur l'efficacité du système est particulièrement néfaste. Pendant les transferts évoqués ci-dessus, l'exécution du programme ayant subi un défaut de page doit être suspendue. Du point de vue de l'utilisateur la durée totale de ces suspensions est un élément primordial du temps de réponse. Le nombre de programmes traités par le calculateur dans un temps donné est lui aussi très sensible à une augmentation du taux de défauts de page, même si le système d'exploitation est multiprogrammé. Dans ce cas, en effet, si le taux de défauts de page de chacun des programmes qui se partagent l'unité centrale est élevé, il en est de même de la probabilité que tous ces programmes soient suspendus et donc que l'unité centrale soit inactive. En outre, le trafic sur les canaux a pour conséquence des vols de cycles-mémoire qui affaiblissent la vitesse d'exécution du calculateur. Enfin, le temps de calcul nécessaire au traitement de ces défauts de page par le moniteur peut devenir important. Pour limiter ces inconvénients, des travaux ont été menés dans deux directions :

- ~~minim~~isation du taux de défauts de page. Les ARP conçus dans ce but s'efforcent de remplacer la page dont la présence en mémoire centrale

BIBLIOTHEQUE DU CERIST



sera inutile jusqu'à la date la plus éloignée [2] .

- minimisation du temps de réponse du disque. Ce délai peut être réduit par l'utilisation de disques à têtes fixes, à secteurs [3] [4] .

Le but de la présente étude est d'analyser des stratégies globales d'allocation de la mémoire hiérarchisée qui concilient les exigences parfois contradictoires de la gestion de mémoire centrale et de celle des mémoires secondaires rotatives.

I.2. - LA GESTION SIMULTANEE DES DISQUES ET DE LA MEMOIRE CENTRALE

Dans un système à mémoire virtuelle, il est important d'améliorer le temps de réponse, à temps de calcul donné. Il est donc naturel de choisir pour critère d'optimisation :

$$\eta = \frac{e}{e + T}$$

où e est le temps moyen entre deux défauts de page consécutifs et T le temps moyen de suspension d'un programme à la suite d'un défaut de page. Cette grandeur peut encore s'écrire :

$$\eta = \frac{1}{1 + FT}$$

où $F = \frac{1}{e}$ désigne le taux de défauts de page. La maximisation de η équivaut à la minimisation de FT . Les études de gestion de disques que nous avons citées ont pour but la minimisation de T, les études sur les ARP celle de F. Or ces optimisations peuvent reposer sur des exigences contradictoires. Il suffit pour s'en convaincre de considérer le cas suivant : supposons que l'image de la page sélectionnée par un ARP optimisé vienne juste de passer sous la tête de lecture-écriture du disque ; l'attente alors nécessaire contrebalance, et au delà, le bénéfice de l'optimisation du remplacement. Nous analyserons donc une classe d'ARP qui prennent leurs décisions non seulement en fonction de l'état de la mémoire et du comportement du programme mais aussi, ce qui est nouveau, en fonction d'une prévision du temps de

BIBLIOTHEQUE DU CERIST

réponse du disque. L'effet de ces ARP sera évalué selon le critère n précédemment défini et comparé avec les méthodes classiques.

I.3. - PLAN DE L'ETUDE

Considérons un système paginé dans lequel chaque page de l'espace d'adressage N d'un programme a une image fixe sur disque. Cette image correspond à un secteur d'une piste du disque à têtes fixes (voir fig. 1). Considérons aussi un ARP qui détermine à chaque instant de défaut de page un ensemble Π_k de k pages susceptibles d'être remplacées. Supposons qu'à chaque défaut de page deux transferts entre mémoire centrale et mémoire secondaire doivent être effectués : le chargement de la page référencée et le rangement de la page remplacée. Les ARP de la classe que nous considérons choisissent un élément p_i de Π_k qui minimise le temps de réponse du disque, ce temps étant quasiment identique au temps de suspension du programme. S'il n'existe aucune corrélation entre la position de l'image de la page à charger et celle de la page à ranger, il est alors raisonnable de supposer que toutes les pages de Π_k ont la même chance d'être sélectionnées par l'ARP; c'est pourquoi nous considérons dans la deuxième partie le modèle des ARP avec sélection aléatoire d'un élément de Π_k .

Dans la troisième partie nous calculons la valeur moyenne du temps de réponse du disque sous deux politiques. Dans l'une d'elles aucune page physique n'est réservée comme tampon ce qui contraint à effectuer le rangement d'une page avant le chargement de la page référencée. Dans l'autre politique une telle réservation est effectuée, ce qui lève cette contrainte mais à l'inconvénient de diminuer l'espace physique dans lequel le programme peut s'exécuter.

Le modèle considéré ignore les files d'attente au niveau du disque. Il sera donc apte à décrire les systèmes monoprogrammés à mémoire virtuelle ainsi que les systèmes dans lesquels la charge des disques est légère.