

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Expérience de réglage d'un système VAX/VMS

Rahier, Anne-Geneviève

*Award date:*  
1983

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

ANNEE ACADEMIQUE 1982-1983

EXPERIENCE DE REGLAGE

D'UN SYSTEME VAX/VMS

Promoteur :  
Monsieur J. RAMAEKERS

Mémoire présenté en vue de  
l'obtention du grade de  
Licencié et Maître en Informatique

Anne-Geneviève RAHIER

Au terme de ce travail, qu'il me soit permis de remercier Monsieur J. RAMAEKERS qui a bien voulu diriger et orienter cette étude.

J'exprime également toute ma reconnaissance à Monsieur DE MARNEFFE, Professeur à la faculté polytechnique de Mons, qui a donné l'élan nécessaire au démarrage de ce travail. Grâce à la S.S.C.I.SGAB, j'ai pu découvrir le VAX et réaliser les tests sur son matériel. Que Monsieur P. ROUSSEL, Directeur de la SGAB, trouve ici l'expression de ma sincère gratitude.

Enfin, ma plus vive reconnaissance va à tous ceux, qui d'une manière ou d'un autre, m'ont aidée au cours de cette année.

## TABLE DES MATIERES

### INTRODUCTION

#### I. PRINCIPES DE GESTION DYNAMIQUE DES SYSTEMES A MEMOIRE VIRTUELLE

I.1	Mémoire virtuelle	4
I.2	Principe de localité	5
I.3	Traduction des adresses	6
I.4	Politiques de transfert de pages	8
I.5	Politiques de chargement	10
I.6	Working-set	11

## II. LE SYSTEME VAX/VMS

II.1	L'organisation de la mémoire	15
II.1.1	Traduction des adresses	16
II.1.2	Contrôle de l'accès à la mémoire	19
II.1.3	Politique de transfert de pages et de chargement	21
II.2	Working-set d'un processus	22
II.3	Pagination	24
II.3.1	Liste des pages libres	24
II.3.2	Liste des pages modifiées	24
II.4	Evolution d'un working-set	26
II.4.1	Progression d'un working-set lors de l'exécution d'un programme	26
II.4.2	Transferts possibles de pages pour un working-set	29

### III. ASPECTS DU CONTROLE GLOBAL DES SYSTEMES A MEMOIRE VIRTUELLE

III.1	Comportement dynamique d'un système multiprogrammé à mémoire virtuelle	33	
III.2	Modèle de WILKES	35	
III.3	Le rôle des paramètres	37	
	III.3.1	Choix des politiques	37
	III.3.2	Flux des processus	40

## IV. APPLICATION DU MODELE DE WILKES AU SYSTEME VAX/VMS

IV.1	L'environnement d'un VAX	43
IV.2	Balance-set et scheduling	45
IV.2.1	Le scheduler	46
IV.2.2	Le swapper	49
IV.3	Description des paramètres	50
IV.3.1	Par processus	50
IV.3.2	Pour le système	55
IV.4	Politiques et flux des processus dans le VAX/ VMS	69
IV.4.1	Les valeurs des paramètres, image d'une politique	69
IV.4.2	Influence sur les performances d'un système	71

## V. REGLAGE DU SYSTEME

V.1	Objectifs et méthodes	78
V.1.1	Objectifs des mesures	78
V.1.2	Environnement expérimental	79
V.1.3	Méthodes de prise des mesures	81
V.1.4	Connaissance de la charge	82
V.2	Comportement du système vis-à-vis de la charge	84
V.2.1	Description des paramètres initiaux	85
V.2.2	Résultats des mesures	86
V.3	Degré d'ajustement du système à la charge	92
V.3.1	Variation des paramètres	92
V.3.2	Analyse des résultats	93
V.3.3	Evaluation des performances	113

## CONCLUSION

## BIBLIOGRAPHIE

## ANNEXES

1. Les paramètres du système	119
a. Tableaux des paramètres par catégorie	120
b. Liste des paramètres par ordre alphabétique	131
2. Le logiciel de mesure : "MONITOR"	147
3. Les mesures	154
a. Paramètres initiaux	154
b. Résultats par "MONITOR"	157
c. Tableaux synthétiques des résultats	168
1. Variations des paramètres	168
2. Résultats des tests	173
3. Variations des résultats de mesures	175
4. Répartition des défauts de page	176

## INTRODUCTION

---

Depuis quelques années, on constate un intérêt croissant pour l'évaluation des performances des systèmes d'exploitation. Le coût et la complexité du matériel et des programmes ont tendance à augmenter ; l'intérêt économique d'une bonne utilisation des ressources est donc certain.

La clé du problème de l'évaluation des performances d'un système est dans la compréhension de son fonctionnement. C'est dans cette optique que cette expérience de réglage d'un système VAX/VMS a été réalisée.

La complexité d'un système d'exploitation et la diversité des fonctions qu'il remplit nécessitent l'élaboration d'une synthèse de celui-ci. C'est ce que nous présentons dans le chapitre I, tandis que le chapitre II détaille la manière dont le système VAX/VMS, en particulier, a implémenté ces fonctions.

Le chapitre III présente les problèmes de contrôle d'un système multiprogrammé à mémoire virtuelle et les interactions entre les processus. A l'aide du modèle de Wilkes de flux des processus dans un système à mémoire virtuelle, nous avons tenté de déterminer le rôle des paramètres. En effet, bien que la politique générale d'un système d'exploitation soit déterminée par le constructeur, un système doit être, si possible, adaptable aux exigences particulières de chaque environnement d'utilisation.

Sur base du modèle de Wilkes, le chapitre IV explicite l'implémentation concrète dans le système VAX/VMS des politiques choisies et décrit l'influence des principaux paramètres.

Le processus de réglage de ces paramètres est essentiellement cyclique, ce fut la base de la démarche des expériences .

Par un processus itératif de mesure de l'utilisation de la mémoire, modifications des paramètres et nouvelles mesures pour évaluer les effets des modifications, la performance du mécanisme de gestion de la mémoire virtuelle fut étudiée. C'est l'analyse des résultats de ces mesures que relate le chapitre V.

## CHAPITRE I

## PRINCIPES DE GESTION DYNAMIQUE DES SYSTEMES A MEMOIRE VIRTUELLE

1. Mémoire virtuelle
2. Principe de localité
3. Traduction des adresses
4. Politiques de transfert des pages
5. Politiques de chargement
6. Working-set

Depuis quelques années, de nombreux systèmes d'exploitation d'ordinateurs disposent d'une mémoire virtuelle. Elle permet de réaliser l'indépendance des programmes des utilisateurs vis-à-vis de la mémoire physique disponible. Pour fonctionner avec efficacité, un tel système doit avoir une gestion des mémoires centrales et auxiliaires et des transferts qui ne crée pas une charge trop lourde pour le processeur central.

### I.1 MEMOIRE VIRTUELLE

Un système informatique peut être décrit comme un ensemble de ressources. L'exécution d'un programme est un processus demandeur de ressources, elle est constituée d'une séquence d'opérations élémentaires, chacune d'elles requérant l'utilisation de certaines ressources pendant un certain temps. Parmi ces ressources, les processeurs sont les unités qui "traitent" les informations. Chaque traitement élémentaire nécessite l'accès à certaines informations. Celles-ci sont stockées sur des supports physiques appelés "mémoires".

Il existe plusieurs types ou niveaux de mémoires. Les informations directement accessibles du processeur sont celles de la mémoire dite centrale. Celle-ci est très rapide et donc coûteuse et par conséquent limitée. Elle est alors complétée par des mémoires secondaires ou auxiliaires beaucoup plus vastes et moins coûteuses.

Afin de permettre une utilisation rationnelle et efficace des différentes ressources, les systèmes travaillent généralement en multiprogrammation. Ainsi plusieurs programmes peuvent être pris en charge simultanément par le système. Une ressource momentanément non utilisée par un programme peut alors être allouée à un autre programme. Cette utilisation en parallèle des ressources améliore leurs taux d'activité individuels et, par conséquent, les performances du système tout entier.

La gestion d'un système de multiprogrammation pose de nombreux problèmes. Il s'agit d'allouer les ressources aux programmeurs de manière optimale. Un des problèmes clé est l'allocation de la mémoire centrale.

- D'une part, étant elle-même limitée, la mémoire centrale limite la taille et le nombre de programme pouvant être exécutés simultanément.
- D'autre part, les performances propres à l'allocation de la mémoire conditionnent étroitement les performances globales du système.

Parmi les techniques permettant un usage partagé de la mémoire centrale, celle de la mémoire virtuelle est de plus en plus utilisée dans les systèmes actuels. Elle a notamment pour but de reculer "virtuellement" les limites physiques de la mémoire centrale.

## I.2 PRINCIPE DE LOCALITE

Au cours de l'exécution d'un programme, les informations accédées sont généralement concentrées dans certaines zones spécifiques du code objet. C'est sur cette constatation que se base la mémoire virtuelle. Le principe de localité a été énoncé par DENNINE. Il revient à dire que les adresses virtuelles utilisées par un processus en exécution tendent à être groupées dans de petites régions de l'espace d'adresses virtuelles et les changements de ces groupes sont intermittents.

Du fait de cette localisation des références, les besoins instantanés des processus en espace sont réduits comparativement à leur taille totale. Pour chacun des programmes, il est donc raisonnable de ne garder en mémoire centrale que les parties momentanément requises. Les parties restantes sont mises sur une mémoire secondaire et chargées en mémoire centrale en fonction des besoins.

Le mécanisme de la mémoire virtuelle prend en charge cette allocation partielle et dynamique de la mémoire. Les problèmes posés sont :

- d'une part, la découpe des programmes en morceaux de même dimension appelé "pages" et l'identification des informations de chacune des pages par une adresse virtuelle,
- d'autre part, le transfert d'une information entre la mémoire centrale et la mémoire secondaire.

Tout au long de l'exécution des programmes, le système devra décider de la répartition dynamique des informations et par conséquent, de leurs mouvements entre les différents niveaux de la mémoire. Les informations requises par le processeur doivent être présentes en mémoire centrale. Par conséquent, lorsqu'un programme encourt un défaut de page (référence à une information absente de la mémoire centrale), il est interrompu au moins jusqu'à ce que la page manquante soit chargée en mémoire centrale.

L'allocation de la mémoire joue donc un rôle important. Chaque programme devrait être interrompu le moins souvent possible par ces attentes de transferts.

### I.3 TRADUCTION DES ADRESSES

Le problème est de concevoir une méthode d'organisation de la mémoire telle qu'il soit possible de traduire une adresse mémoire virtuelle en une adresse mémoire physique réelle.

Une solution à ce problème est la pagination. Son principe est le suivant :

- l'espace d'adresses virtuelles de chaque processus est divisé en "pages" de dimension fixe,

- la mémoire physique est divisée en "pages frames" ou cadres de même dimension qu'une page.

Chaque processus possède un ensemble de pages qui peut, à tout instant, être subdivisé en deux sous-ensembles :

- les pages résidentes en mémoire centrale,
- les pages conservées en mémoire secondaire.

Le mécanisme de pagination a pour tâche :

- de faire la transformation d'une adresse virtuelle en une adresse physique,
- de gérer les transferts de pages entre la mémoire centrale et la mémoire secondaire.

Une adresse virtuelle est un doublet formé du numéro de la page, suivi du déplacement par rapport au début de la page. A chaque processus est rattaché une table de pages dont chaque entrée correspond à une page de l'espace virtuel du processus. Une entrée contient :

- un indicateur de page en mémoire centrale ou secondaire,
- l'adresse de la page en mémoire centrale ou la position de la page en mémoire secondaire.

En examinant une entrée de la table des pages sur base du numéro de la page, il se peut qu'on constate que la page n'est pas en mémoire centrale. C'est ce qu'on appelle un défaut de page. Dans ce cas, le processus devient non exécutable jusqu'à ce que la page soit amenée en mémoire centrale. Ce transfert est organisé par le système d'exploitation.

L'avantage primordial de la mémoire virtuelle est de libérer le programmeur de la contrainte imposée par la taille de la mémoire physique sur la taille de ses programmes.

Cependant, la mémoire virtuelle a pour désavantage que tout accès à la mémoire est indirect, via la table des pages. Mais cet inconvénient peut être compensé par l'utilisation d'un mécanisme hardware basé sur une mémoire associative pour effectuer la transformation d'adresse. (1)

#### I.4 POLITIQUES DE TRANSFERT DES PAGES

Au fur et à mesure de son exécution, les pages dont le processus a besoin changent. Il faut donc remplacer les pages en mémoire centrale.

La question est de savoir : "Quelle page va être choisie pour fournir la "page frame" (ou cadre en mémoire centrale) nécessaire ?"

La solution idéale serait de pouvoir déterminer la page qui ne sera plus référencée avant le laps de temps le plus long. Mais les références des programmes ne sont jamais connues à l'avance. Toute stratégie d'allocation de la mémoire repose donc sur une méthode de prévision : au fur et à mesure du déroulement d'un programme, son comportement passé est mémorisé pour estimer celui à venir.

Les politiques de remplacement les plus utilisées sont :

a. LRU : least recently used

L'idée est de sélectionner la page qui n'a plus été référencée depuis un laps de temps le plus grand. Cela nécessite la mise en oeuvre d'un mécanisme hardware spécial. A chaque "page frame" (ou cadre en mémoire centrale) est associé un bit de référence qui est positionné dès qu'un accès à la page est exécuté et initialisé périodiquement.

b. LFU : least frequently used

L'idée est de conserver pour chaque page un compteur du nombre d'accès effectués dans le dernier intervalle de temps.

Le cadre en mémoire centrale choisi est celui qui est occupé par la page dont le compteur d'utilisation a la valeur la plus faible.

En plus de l'adaptation d'un mécanisme hardware particulier cette politique a le défaut qu'une page récemment chargée risque d'avoir la valeur la plus faible pour le compteur du nombre d'accès. Aussi faut-il garantir un temps de séjour minimum en mémoire centrale pour chaque page.

c. FIFO : first-in, first-out

La page choisie pour l'expulsion est celle qui réside en mémoire centrale depuis le plus long temps. Ce mécanisme est particulièrement simple, mais risque fort d'engendrer une politique fort éloignée de la solution idéale.

Il reste à savoir :

- si la politique choisie de transfert de pages s'applique au niveau du système complet en considérant toutes les pages en mémoire centrale indépendamment des processus auxquels elles appartiennent,
- ou bien, si chaque processus est considéré indépendamment.

Ce qui reviendra :

- soit à une politique de pagination d'un processus contre les autres processus,
- soit à une politique de pagination du processus contre lui-même,
- soit à un compromis entre ces deux politiques.

## I.5 POLITIQUES DE CHARGEMENT

Ainsi les modèles de prévision du comportement sont formulés comme des règles d'estimation des références futures sur la base de celles passées. Les pages estimées comme "probablement" requises dans un avenir proche sont maintenues (ou chargées) en mémoire centrale, tandis que celles "peu probables" sont laissées (ou déchargées) sur la mémoire secondaire. Il reste à savoir : quand active-t-on le processus des demandes de pages ?

Les demandes de pages sont générées par un programme au cours de son exécution. Mais plus que le transfert d'une page depuis un disque vers un cadre de la mémoire centrale, c'est la recherche de la page sur le disque et le positionnement du bras de lecture qui prennent du temps. Aussi différentes politiques de chargement sont possibles pour répondre au plus vite à ces défauts de page.

### a. Politique "à la demande"

Un transfert de page est activé à chaque défaut de page. Pendant toute la durée du transfert, le processus qui subit le défaut de page est non exécutable.

### b. Politique "anticipatoire"

Cette politique se base sur une hypothèse de comportement futur du processus. Ainsi il est possible d'anticiper les défauts de page en activant préventivement les transferts de pages par chargement de plusieurs pages d'un coup. De cette manière, on peut supprimer les changements d'état du processus et les changements de contexte qui y sont associés.

## I.6 WORKING-SET

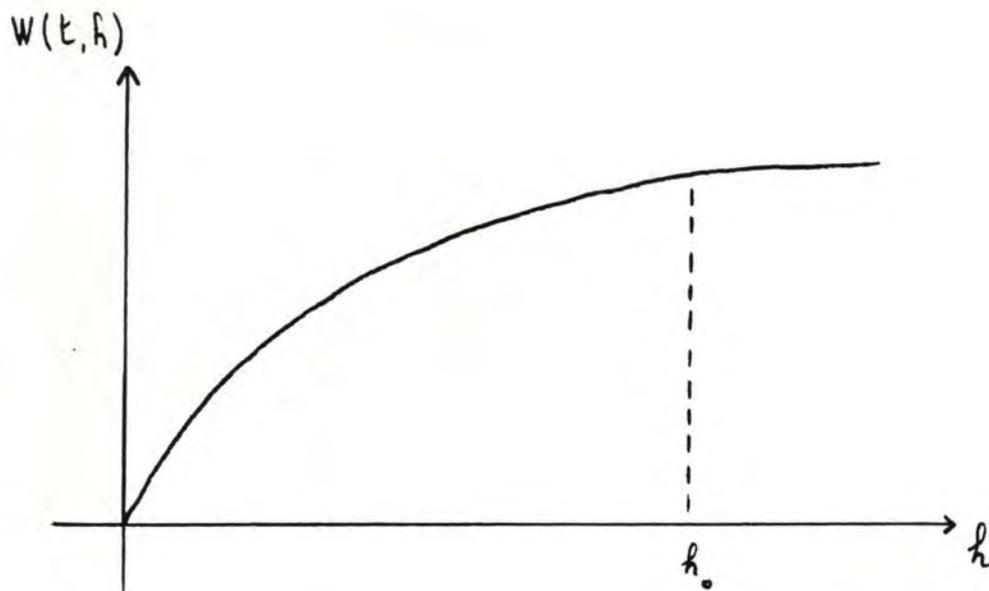
Il est à remarquer que jusqu'à présent, nous avons considéré l'allocation de l'espace en mémoire centrale à chacun des programmes considéré individuellement. Par conséquent, à côté du coût temporel qu'un programme encourt suite aux attentes de transferts de pages, il faudra lui imputer un coût spatial mesurant la quantité d'espace alloué au programme. En effet, la mémoire centrale étant limitée, l'espace alloué à un programme l'est toujours au détriment des autres.

Une allocation "efficace" devrait donc réaliser un certain compromis entre ces deux objectifs contradictoires. De là vient la notion de "working-set" ou ensemble d'activité. Conceptuellement, c'est un principe de l'offre selon les besoins. A tout moment de l'exécution d'un programme, on doit allouer à ce dernier toutes ses pages qui sont très vraisemblablement requises dans un futur proche et ces pages-là seulement.

Cet ensemble de pages requises forment le "working-set". C'est le nombre de pages nécessaires en mémoire centrale, pour que le processus puisse utiliser efficacement le processeur, sans quoi il subirait une série de défauts de page et ne pourrait pas progresser.

D'un point de vue pratique, une telle allocation exige que le système d'exploitation dispose d'une certaine méthode pour estimer les besoins instantanés des programmes sur la base de leur comportement passé. La plupart de ces méthodes se basent sur les références récentes des programmes pour estimer leur besoin actuel, tenant compte ainsi du principe de localité. Ainsi le working-set d'un processus est constitué, à chaque instant, de ses pages qui ont été référencées au moins une fois depuis une certaine durée fixe de temps. Il varie au fur et à mesure de la progression du processus.

Si  $t$  est le temps et  $h$  le nombre de dernières références considérées, l'ensemble d'activité varie en fonction de  $h$  selon l'allure suivante : (2)



En effet :  $W(t, h) = \left\{ \text{page } i \mid \text{page } i \text{ apparaît dans les } h \text{ dernières références} \right\}$

Si  $h$  croît, l'ensemble des pages  $i$ , qui contiennent les  $h$  dernières références, croît. Mais à partir d'une certaine valeur  $h_0$ , vu le principe de localité, tout accroissement de  $h$  n'entraîne plus qu'un très faible accroissement du nombre de pages contenues dans le working-set.

Une autre stratégie pour estimer le working-set dont un processus a besoin se base sur la fréquence des défauts de pages passés. Ainsi les programmes ayant subi de fréquents défauts de page peuvent acquérir des pages supplémentaires au détriment des autres. Inversement, on décroît le nombre de pages résidentes pour le processus dont la fréquence des défauts de page est faible.

L'importance accordée à la réduction des défauts de page provient du rapport des temps d'exécution d'une instruction machine ( $\pm 2 \mu\text{sec}$  de temps CPU) et des temps de transfert d'une page ( $\pm 40 \text{ msec}$  de temps écoulé depuis sa demande jusqu'à sa présence en mémoire centrale). Aussi si trop de processus subissent trop de défauts de page, le processeur ne peut être utilisé efficacement.

En conclusion, la stratégie de l'ensemble d'activité potentiel représente davantage un "principe" d'allocation plutôt que des règles opératoires bien déterminées. Elle suggère comment le principe de l'offre en fonction de la demande devrait être transposé dans le contexte de la gestion d'une mémoire virtuelle.

Ainsi, à tout instant, on vise :

- à remplir la mémoire centrale d'informations les plus utiles actuellement, en prévision des demandes futures,
- à permettre d'utiliser toute cette mémoire comme un pot commun dans lequel chaque programme peut puiser selon ses besoins du moment.

## CHAPITRE II

### LE SYSTEME VAX / VMS

1. L'organisation de la mémoire
  - 1.1 Traduction des adresses
  - 1.2 Contrôle de l'accès à la mémoire
  - 1.3 Politique de transfert de pages et de chargement
2. Working-set d'un processus
3. Pagination
  - 3.1 Liste des pages libres
  - 3.2 Liste des pages modifiées
4. Evolution d'un working-set
  - 4.1 Progression d'un working-set lors de l'exécution d'un programme
  - 4.2 Transferts possibles de pages pour un working-set

## LE SYSTEME VAX / VMS

---

D'un point de vue architecture, la principale caractéristique qui distingue les mini-ordinateurs des gros ordinateurs est la taille de l'espace d'adresses virtuelles c'est-à-dire l'espace d'adresses disponibles individuellement par processus.

Le système d'exploitation à mémoire virtuelle VAX / VMS fait sauter cette distinction. En effet, il s'agit d'un système multiprogrammé qui combine une architecture de 32 bits avec un système d'exploitation à mémoire virtuelle qui fournit un espace d'adressage potentiel quasi illimité ( $2^{32} = 4.3 \text{ G} = 4.3 \times 10^9$  adresses). De là vient la dénomination VAX-11 pour "Virtual Address Extension of PDP-11 architecture" (C.G.Bell).

### II.1 L'ORGANISATION DE LA MEMOIRE

Nous allons à présent découvrir comment le système d'exploitation VAX / VMS procède pour résoudre les nombreux problèmes posés par la gestion d'une mémoire virtuelle :

- Traduction des adresses virtuelles générées par les processus en adresses mémoires physiques.
- Contrôle de l'accès à la mémoire.
- Gestion des mémoires qui permettent à un processus de s'exécuter même si tout son espace d'adresses virtuelles n'est pas simultanément transformé en adresses physiques dans la mémoire réelle.

La méthode choisie par le système d'exploitation VAX / VMS est la gestion de la mémoire par découpe en page, avec un contrôle d'accès au niveau de la page, ainsi qu'un petit nombre de mode d'accès hiérarchique, pour la gestion de problèmes spécifiques.



où les bits 08 : 00 donne un byte dans une page de 512 bytes  
 qui est l'unité de base d'adressage,  
 les bits 29 : 9 donne un numéro de page virtuelle  
 les bits 31 : 30 donne la région d'adresses virtuelles  
 0 0 la région programme  
 0 1 la région contrôle  
 1 0 la région système  
 1 1 la région réservée.

Le mécanisme de traduction d'adresse consiste à sélectionner la table des pages de la région concernée sur base des bits 31 : 30 déterminant la région.

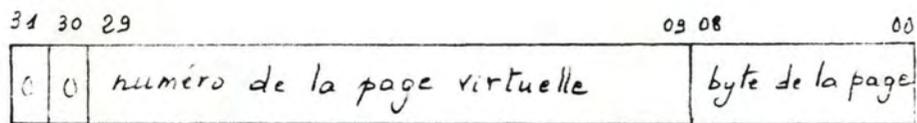
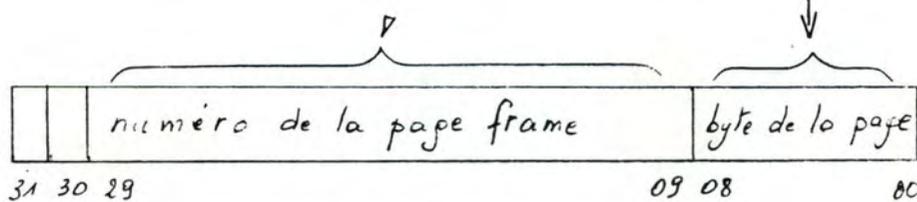
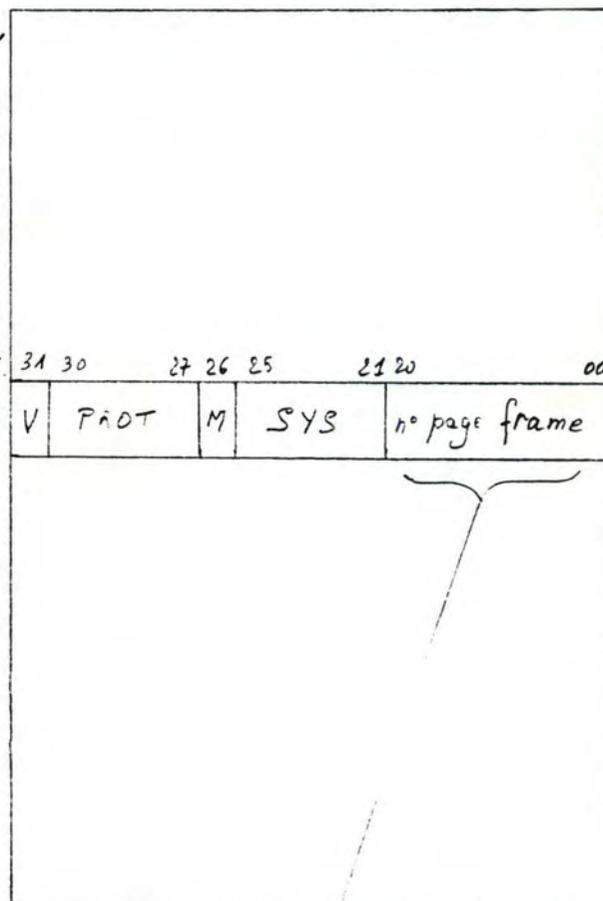
Après avoir vérifié que le numéro de page virtuelle est un de ceux permis pour le processus, le numéro de page virtuelle sert à indexer la table des pages pour sélectionner l'entrée de la table des pages correspondantes.

Cette entrée contient :

- soit le numéro de la "page frame" (ou cadre de la mémoire centrale) correspondante sur 21 bits qui concaténés avec les 9 bits de rang inférieur de l'adresse virtuelle initiale forme l'adresse physique de 30 bits
- soit un indicateur signifiant que la page virtuelle accédée n'est pas en mémoire centrale c'est-à-dire qu'il y a défaut de page. L'exécution de l'instruction du processus courant est alors suspendue et le contrôle est transféré au système d'exploitation qui se chargera du transfert de la page manquante en mémoire centrale. A ce moment, l'exécution de l'instruction du processus courant pourra se poursuivre.

Le schéma suivant décrit la transformation d'une adresse virtuelle en une adresse physique dans le cas où la page est en mémoire centrale.

adresse virtuelle

Table des pages  
de la région programme

adresse physique

- le "valid bit" :  $V = 1$  la page est en mémoire centrale  
 $V = 0$  la page n'est pas en mémoire centrale, alors les bits 25 : 00 sont utilisés pour repérer la position de la page sur le disque.
- les "protect bits" : (PROT) indiquent les modes d'accès possibles pour la page (read-only, read-write) selon les différents modes de fonctionnement du CPU (Kernel, Executive, Supervisor, User);
- le "modify bit" : M est positionné par le hardware dès qu'une instruction a modifié une information dans la page.

La table des pages de la région système est définie par le registre système de base qui contient l'adresse physique de début de la table des pages de la région système et le second registre système qui contient la longueur de la table. Donc la table des pages de la région système est contigue en mémoire centrale.

Les tables de pages de chaque processus sont définies similairement pour les régions contrôle et programme par des registres de base et des registres pour les longueurs des tables. Cependant, les registres de base ne contiennent pas les adresses physiques, mais les adresses virtuelles de la région système. Ainsi, les tables des pages de chaque processus sont contigues dans l'espace d'adresses virtuelles de la région système, mais ne le sont pas nécessairement dans la mémoire physique. Cette méthode permet de paginer les tables des pages de chaque processus et évite ainsi des problèmes d'allocation de mémoire physique.

### II.1.2 Contrôle de l'accès à la mémoire

A tout moment, un processus s'exécute dans un des 4 modes d'accès possibles qui sont par ordre décroissant de privilège :

- le mode Kernel pour les traitements d'interruption, la pagination, les I/O physiques, le scheduling, ...
- le mode Executive pour le I/O logiques provenant de fonctions RMS, ...
- le mode Supervisor pour l'interpréteur des commandes.
- le mode User pour les procédures des utilisateurs et les données.

L'accessibilité de chaque page (lecture, écriture ou aucun) pour chaque mode d'accès est spécifié dans l'entrée de la table des pages pour la page. L'accessibilité est ordonnée hiérarchiquement : si une page peut être écrite dans un mode donné, elle peut aussi y être lue et si une page est accessible dans un mode moins privilégié, elle l'est également dans un mode plus privilégié.

Le mode d'accès courant est stocké pour chaque processus dans son "processor status longword". Les procédures utilisent à chaque mode d'accès une pile séparée avec l'accessibilité correspondante. Aussi à chaque processus sera associé 4 pointeurs de pile dont on sélectionnera celui approprié au mode d'accès courant pour le "processor status longword".

Ainsi donc le contexte d'un processus dans l'architecture VAX est déterminé par

- les 15 registres généraux R0 à R13 et R15
- les 4 copies de R14 (pointeurs de pile) :  
une par mode d'accès
- le "processor status longword"
- les 2 registres de base et de longueur  
pour les tables de pages des régions  
programme et contrôle.

Ce contexte est conservé pour chaque processus dans une structure de donnée appelée "process control block".

### II.1.3 Politique de transfert de pages et de chargement

Au fur et à mesure de la progression des processus, la politique de remplacement des pages en mémoire centrale suivie par le VAX est la politique FIFO au niveau de chaque processus. Le choix provient probablement de la simplification hardware qui en résulte.

D'autre part, l'introduction de zones tampons pour les pages écartées permet d'atténuer le défaut du FIFO. Pour répondre aux demandes de pages générées par un processus au cours de son exécution, le VAX/VMS suit une politique de chargement à la demande.

Mais le VAX / VMS permet de modifier la politique suivie

- soit en gelant une page en mémoire centrale, celle-ci n'est alors jamais choisie pour un remplacement,
- soit en changeant les pages anticipativement, il définit alors le nombre (cluster) de pages qui sera automatiquement chargé en mémoire centrale lors d'un défaut de page.



- . WSEXTENT : la limite absolue du nombre de pages que peut accorder le système au working-set du processus utilisateur, ce nombre dépasse WSQUOTA et les pages ne seront disponibles pour le processus que pour autant que le système ait un excédent de pages libres.

Pour chaque working-set, la condition suivante est invariante :  
 $MINWSCNT < WSLIMIT \ll WSMAX$ , où WSLIMIT représente la taille limite du working-set courant.

Les paramètres globaux sont fixés à l'installation du système tandis que les paramètres par processus sont déterminés à la création de l'utilisateur mais le paramètre WSDEFAULT est modifiable au lancement de chaque processus par l'utilisateur.

La taille limite du working-set courant, WSLIMIT, pour un processus donné, varie en fonction des défauts de page qu'il encourt. La valeur initiale qui lui est donnée est celle du paramètre WSDEFAULT. Ultérieurement, elle est ajustée en fonction de la fréquence des défauts de page. Si la fréquence dépasse la valeur seuil PFRATH, WSLIMIT est augmenté d'une valeur paramètre WSINC, tandis que si la fréquence descend en-dessous d'une autre valeur seuil PFRATL, WSLIMIT est diminué d'une autre valeur paramétrable WSDEC. Cette opération a lieu à l'expiration de chaque quantum de temps CPU.

## II.3 PAGINATION

La pagination est l'action d'amener des pages d'un processus en exécution en mémoire centrale au moment où elles sont référencées. En effet, quand un processus s'exécute, ses pages sont dites résider en mémoire virtuelle. Seules les pages actives sont nécessaires en mémoire centrale. Les autres peuvent rester sur disque jusqu'à ce qu'elles deviennent actives. Dans un tel système, un processus pagine quand il référence des pages qu'il ne possède pas dans son working-set ou lorsqu'il démarre son exécution en mémoire. Quand un processus référence une page hors de son working-set, un défaut de page apparaît. Le "pager" lit la page manquante et la rend disponible en mémoire centrale (ainsi que d'autres pages suivant le facteur de groupage (cluster)) en remplaçant la page la moins récemment référencée, si nécessaire. Ainsi le système ne pagine un processus que contre lui-même.

### II.3.1 Liste des pages libres

Si l'on veut augmenter le nombre de pages d'un working-set, il faut acquérir des "pages frames" (ou cadres en mémoire centrale) dans lesquelles on pourra copier les nouvelles pages. Ces pages frames sont extraites d'une liste des pages libres. La liste des pages libres est donc une réserve de pages. Un paramètre du système FREELIM permet de fixer une valeur plancher minimum pour le nombre de pages de cette liste. La liste est organisée selon la méthode FIFO.

### II.3.2 Liste des pages modifiées

De même, lorsqu'un working-set abandonne une de ses pages, cette page est ajoutée soit à la liste des pages libres, soit à la liste des pages modifiées, selon que le contenu de la page n'a pas été modifié ou a été modifié pendant le séjour de la page dans le working-set.

La liste des pages modifiées est une zone tampon pour les pages qui doivent être copiées sur disque parce que leur contenu a été modifié et que la version de la page qui est sur disque n'est plus à jour. Deux paramètres du système fixent la taille maximum (MPW-HILIMIT) et la taille minimum (MPW-LOLIMIT) de cette liste. Si la taille de la liste dépasse MPW-HILIMIT, des pages sont copiées sur disque par groupe (ou cluster) de MPW-WRTCLUSTER jusqu'à ce que la taille de la liste atteigne la valeur MPW-LOLIMIT. L'organisation de cette liste est également du type FIFO. Les "pages frames" (ou cadres en mémoire centrale) libérées après leur réécriture retournent à la liste des pages libres.

## II.4 EVOLUTION D'UN WORKING-SET

Considérons un programme dont l'image, ou version exécutable, est en mémoire secondaire sur disque. L'image contient essentiellement les instructions-machine correspondantes au programme source et les liens éventuels entre modules. Cela détermine l'espace d'adresses du programme. Rien n'est encore réservé au niveau mémoire pour les divers objets à manipuler dans l'image. C'est à l'exécution du programme que les pages destinées à les contenir seront créées au fur et à mesure des besoins. C'est ce qu'on appelle des "demandes-zéro" de pages.

### II.4.1 Progression du working-set lors de l'exécution d'un programme

Suivons schématiquement l'évolution du working-set d'un processus chargé d'exécuter l'image de ce programme. Quand le système d'exploitation active l'exécution d'une image, un certain nombre de pages sont lues en mémoire secondaire depuis le fichier image situé sur disque et transférées en mémoire principale pour constituer le working-set initial du processus. Le nombre de pages lues d'un disque vers la mémoire centrale en une seule opération dépend d'un paramètre système (PFCDEFAULT) fournit à l'initialisation du système. Cette possibilité de lire un bouquet de pages (ou cluster) en une fois permet au processus de progresser dans son exécution sans encourir trop vite des défauts de pages.

Au fur et à mesure que l'exécution de l'image progresse, des défauts de page peuvent se produire.

- Si la page manquante est une page de l'image, un "cluster" de pages du fichier image sur disque est ajouté aux pages déjà en mémoire centrale dans le working-set.
- Si la page manquante est celle d'une zone de données du programme, une page "demande-zéro" est ajoutée au working-set.

Ainsi progressivement, la taille du working-set risque d'atteindre la valeur limite du working-set courant WSLIMIT. Celle-ci ayant cependant été réajustée à chaque quantum de temps CPU suivant le nombre de défauts de page encouru par augmentation de WSINC ou diminution de WSDEC si la fréquence des défauts de page était inférieure à une borne fixée.

Dès qu'un working-set a atteint sa taille limite WSLIMIT, le processus devra, au prochain défaut de page, céder une ou plusieurs de ses pages afin d'inclure dans son working-set la page ou l'ensemble de pages. Les pages cédées quittent le working-set pour appartenir à la liste des pages libres ou la liste des pages modifiées, suivant qu'elles ont été modifiées durant leur séjour dans le working-set.

La politique de pagination du VAX / VMS est donc une politique de pagination contre le processus. Si un processus pagine, il ne pagine que contre lui-même c'est-à-dire qu'en aucun cas, il ne peut retirer des pages appartenant aux working-sets des autres processus. Par conséquent, un processus exécutant un programme dont la localité est faible encourt une foule de défauts de page, mais il ne déplacera que les pages de la liste des pages libres et celles de la liste des pages modifiées et ne touchera pas aux working-sets des autres processus.

En effet, lorsqu'un processus encourt un défaut de page, la page qui fait défaut appartient :

- soit au fichier image (image file)

La page n'a jamais été lue en mémoire, ou bien elle l'a été mais n'a pas été modifiée. Dans ces cas, dans la table des pages, on possède des informations sur sa situation sur les disques.

- soit au fichier pagination (paging file)

Lorsque la liste des pages modifiées atteint sa taille maximum (MPW-HILIMIT), les pages qui séjournent dans cette liste

depuis le plus long temps sont copiées sur disque dans un fichier spécial (paging file).

- soit aux pages en transition.  
Ce sont les pages de la liste des pages libres et celles de la liste des pages modifiées.
- soit à la zone des données.  
C'est une "demande-zéro" de page. Cette page n'a jamais été accédée précédemment. Alors une page venant de la liste des pages libres sera fournie au processus après avoir été réinitialisée (tous les bits de la page sont mis à zéro).
- soit son transfert est en cours depuis ou vers le disque.

Par conséquent, on voit que les 2 listes contenant les pages en transition agissent en fait comme une extension du working-set des processus. Puisque si la page qui fait défaut appartient à l'une de ces listes, le défaut de page est résolu en  $\pm 200$   $\mu$ sec au lieu de  $\pm 40$  msec pour un défaut de page nécessitant le transfert d'une page depuis un disque.

Si la rotation des pages dans ces 2 listes n'est pas trop rapide, on voit que l'existence de ces listes peut compenser le défaut de l'algorithme de remplacement FIFO. En effet, si la page écartée par FIFO est une page fortement utilisée, une demande de page pour cette page surviendra alors qu'elle se trouve dans une des 2 listes en transition. Cette page sera donc récupérée à faible coût. Cependant, on voit que l'effet d'un programme à faible localité pour les autres processus est de provoquer une rotation anormalement rapide des listes des pages en transition.

Il en résulte que, dans ce cas, les listes de pages en transition ne sont plus de véritables extensions des working-sets étant donné aussi que l'évolution d'un working-set est affectée par la variation de WSLIMIT en fonction de la fréquence des défauts de page.

#### II.4.2 Transferts possibles de pages pour un working-set

Résumons à présent les transferts de pages possibles pour un working-set.

Le transfert d'une page est représenté par un trait fin.

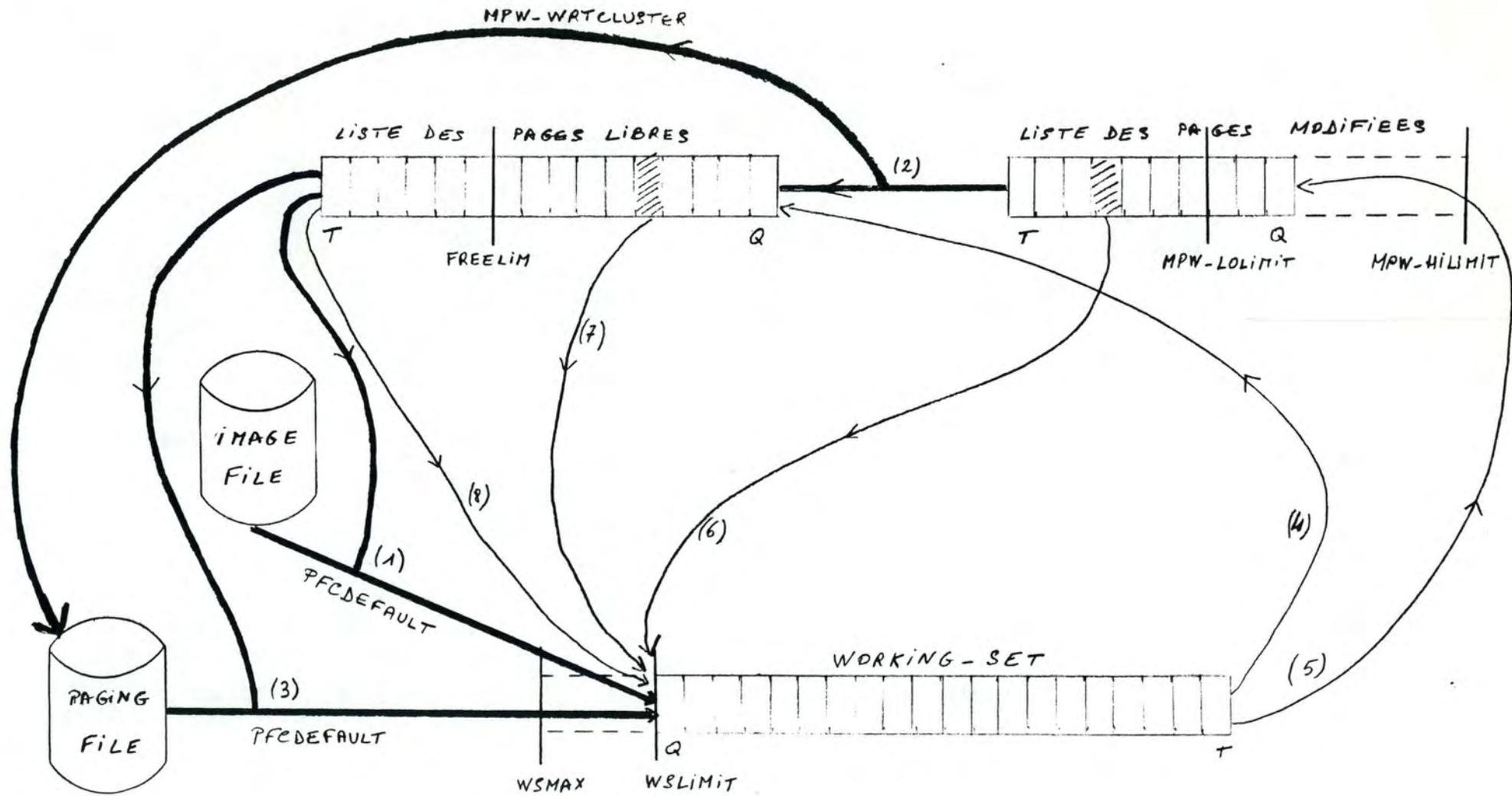
Le transfert d'un groupe de pages est représenté par un trait épais.

Les symboles T et Q désignent respectivement la tête et la queue des listes et du working-set. L'organisation est toujours du type FIFO.

Lorsque le transfert a lieu entre la mémoire et les disques, il est constitué de deux transferts différents :

- a. les pages elles-mêmes du disque vers la mémoire
- b. les "pages frames" (ou cadres de la mémoire centrale) requises pour contenir les pages lues en mémoire secondaire.

- (1) Un cluster de pages passe du fichier image au working-set ; les "pages frames" nécessaires sont extraites de la liste des pages libres.
- (2) Un certain nombre de pages sont transférées de la liste des pages modifiées au fichier de pagination ; les pages copiées rejoignent la liste des pages libres. Ce transfert a lieu dès que la taille de la liste des pages modifiées dépasse MPW-HILIMIT ; les pages sont transférées par cluster jusqu'à ce que la taille de la liste soit inférieure à MPW-LOLIMIT.
- (3) Un cluster passe du fichier de pagination au working-set ; les "pages frames" nécessaires sont extraites de la liste des pages libres.
- (4) Une page passe du working-set à la liste des pages libres.



REPRESENTATION SCHEMATIQUE DE L'EVOLUTION D'UN WORKING-SET (2)

- (5) Une page passe du working-set à la liste des pages modifiées.
- (6) Un défaut de page a lieu pour une page qui appartient à la liste des pages modifiées, la page retourne au working-set.
- (7) Un défaut de page a lieu pour une page qui appartient à la liste des pages libres ; la page retourne au working-set.
- (8) Suite à un défaut de page pour les données, une page "demande-zéro" est ajoutée au working-set en provenance de la liste des pages libres.

Il est à remarquer que seules les pages en transfert entre disque et mémoire centrale sont déplacées physiquement. Les transferts internes à la mémoire centrale reviennent en fin de compte à la mise à jour de la table des "pages frames" qui forment les listes des pages en transition et de chacun des working-sets.

## CHAPITRE III

### ASPECTS DU CONTROLE GLOBAL DES SYSTEMES A MEMOIRE VIRTUELLE

1. Comportement dynamique d'un système multiprogrammé à mémoire virtuelle
2. Modèle de WILKES
3. Le rôle des paramètres
  - 3.1 Choix des politiques
  - 3.2 Flux des processus

Considérons à présent l'allocation des ressources du réseau à un ensemble de programmes, c'est-à-dire les interactions entre les différents programmes au niveau global du système.

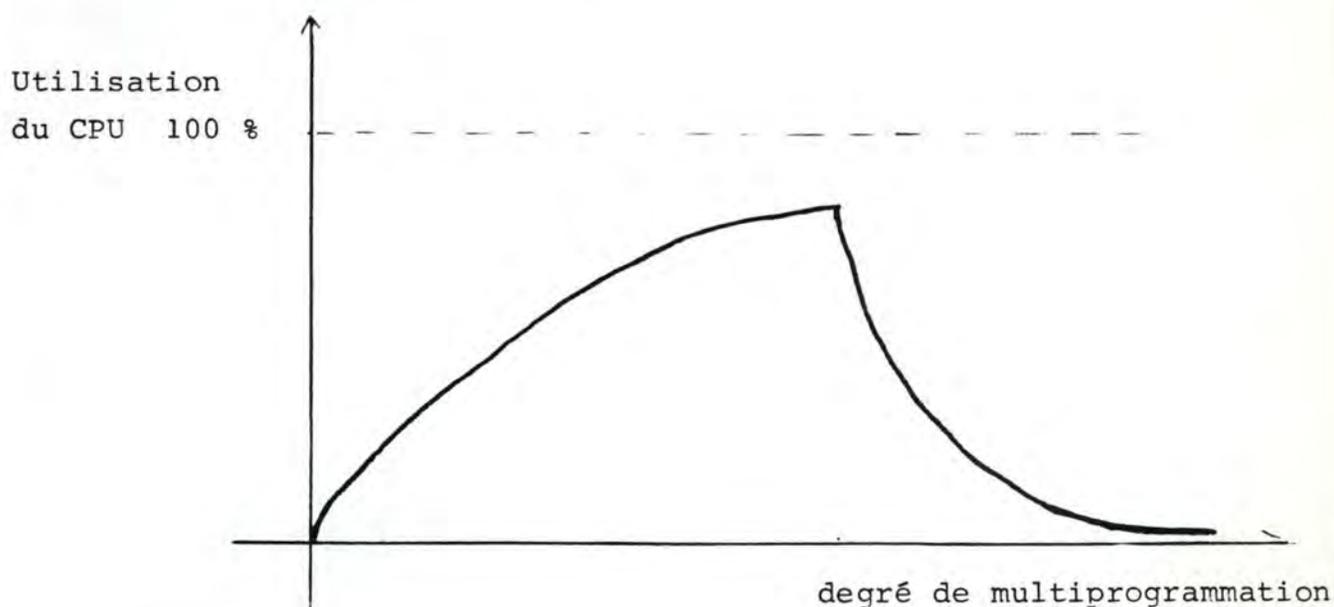
Etant donné un certain ensemble de programmes à exécuter, il s'agit de déterminer les règles de partage de la mémoire centrale et les règles de priorité à chacune des unités du système, cela de façon à ce que certaines unités fondamentales du système (processuer, E/S, par exemple) soient utilisées le plus possible.

La question se posera alors de la caractérisation d'une charge optimale à imposer au système en fonction de critères mesurables.

### III.1 COMPORTEMENT DYNAMIQUE D'UN SYSTEME MULTIPROGRAMME A MEMOIRE VIRTUELLE

L'analyse du comportement dynamique d'un système multiprogrammé à mémoire virtuelle permet de mettre en évidence l'existence d'un degré de multiprogrammation optimal. Autrement dit, il apparaît un nombre maximum de processus possédant des pages en mémoire centrale au delà duquel le taux d'utilisation du CPU cesse de croître pour se détériorer très rapidement.

Pour une taille de mémoire donnée, le pourcentage d'utilisation du CPU en fonction du degré de multiprogrammation fournit une courbe d'allure suivante : (2)

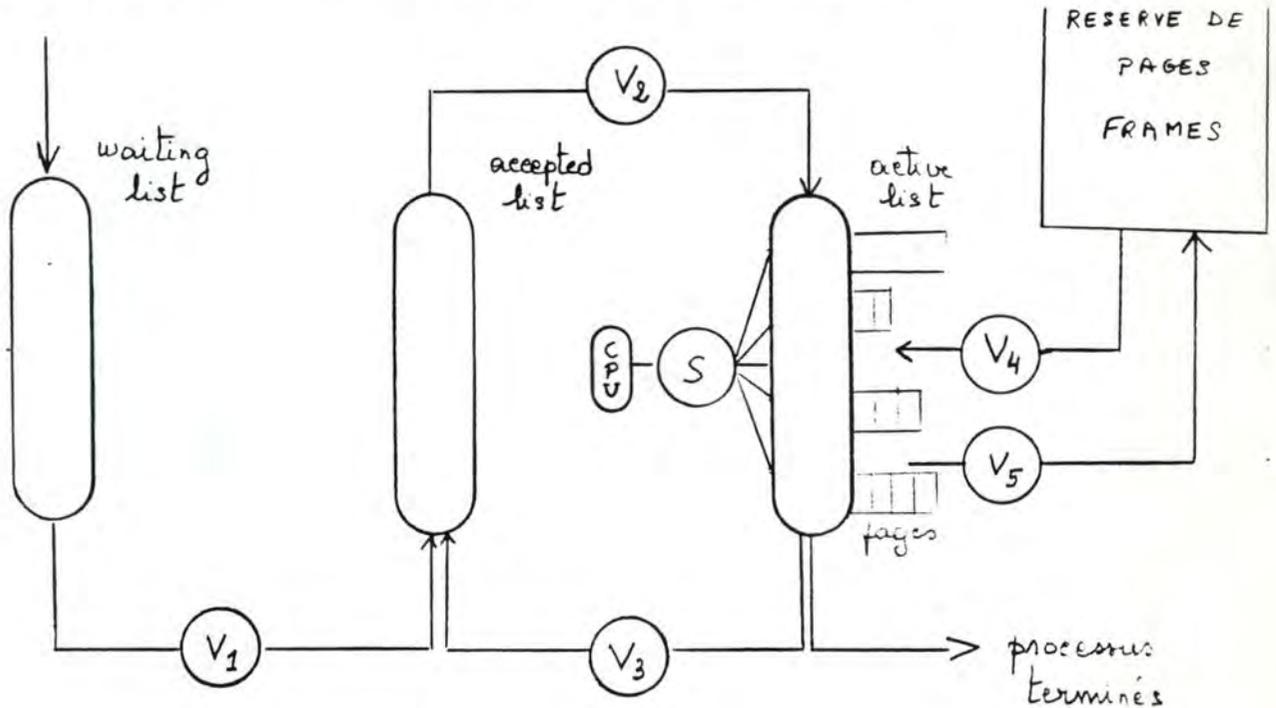


où le pourcentage d'utilisation du CPU signifie le rapport du temps utile consommé par le CPU au temps total écoulé. La partie ascendante de la courbe s'explique par le fait que plus le nombre de processus croît, plus la probabilité de trouver un processus exécutable croît.

Par contre, la partie descendante signifie que passé une valeur seuil, le nombre de processus qui subissent des défauts de pages croît. En fait, aucun processus ne peut conserver en mémoire suffisamment de pages pour progresser à une allure suffisante. Dès qu'un processus exécutable acquiert le processeur, il subit un défaut de page. Ainsi, tous les processus s'étranglent au niveau de l'unité de pagination, il y a saturation du canal de transfert entre la mémoire centrale et les disques. C'est ce qu'on appelle l'écroulement du système (ou trashing).

### III.2 MODELE DE WILKES

Un modèle général du flux des processus dans un système à mémoire virtuelle a été proposé par M.V. WILKES (1973). (17)



Le comportement d'un système informatique est modélisé par un réseau de files d'attente dans lequel circulent des processus demandeurs de ressources.

Les processus sont répartis en 3 listes :

- la liste des processus en attente (waiting list)  
Un nouveau processus qui arrive réside d'abord dans cette file d'attente externe jusqu'à ce qu'il soit admis par le système.
- la liste des processus acceptés (accepted list)  
Ce sont les processus pris en charge par le système, mais qui n'ont pas de pages en mémoire centrale pour l'instant.
- la liste des processus actifs (active list)  
Ce sont l'ensemble des processus pris en charge par le système et qui ont des pages en mémoire centrale.

La réserve de "pages frames" rassemble les cadres de la mémoire centrale momentanément inemployés.

En résumé, le modèle est un réseau de files d'attente dans lequel les caractéristiques opérationnelles des processus (notamment leur taux de défauts de page) varient avec le nombre total de processus en mémoire. Le flux de processus entre les différentes listes est commandé par un ensemble de vannes. Chacune d'elle représente un point de commande du système. C'est en ces points que doivent se prendre les décisions relatives au partage des ressources et aux règles de priorité aux différentes unités du système.

- . Ainsi la vanne  $V_3$  représente l'implémentation d'une politique de transfert de certains processus inachevés de la mémoire centrale vers une mémoire secondaire. C'est le swapping.
- . Les vannes  $V_4$  et  $V_5$  gèrent la réserve de "pages frames" (ou cadres de la mémoire centrale) et sont donc fonction de la politique de pagination.
- . La vanne S correspond à la politique d'attribution du processeur.
- . Quant à  $V_2$ , elle implémente la politique de passage de la liste des acceptés à celle des actifs.
- . Et la vanne  $V_1$ , détermine le nombre de processus pris en charge par le système.

Il apparaît donc que selon le type de politique que ces vannes implémentent, le flux des processus dans le système sera différent.

### III.3 LE ROLE DES PARAMETRES

Il est clair que le système d'exploitation implémenté par le constructeur contient déjà les choix des politiques. Cette politique implémentée n'est pas modifiable par l'utilisateur du système.

Par contre, étant donné qu'un système doit être d'usage général, l'application de la politique doit être adaptable aux exigences de chaque environnement d'utilisation. Cette adaptation se fait en fixant les valeurs des paramètres du système. Ceux-ci déterminent l'ouverture des diverses vannes afin d'assurer un certain flux des processus.

#### III.3.1 Choix des politiques

=====

##### A. Politique de scheduling

C'est la politique de transfert des processus de la liste des processus acceptés à la liste des processus actifs et vice-versa ainsi que les transferts des processus de la liste des processus en attente à la liste des processus acceptés.

Le point de décision crucial est dans ce cas le transfert des processus entre la liste des acceptés et la liste des actifs.

Dans un système time-sharing, il est souhaité que les processus après un temps suffisant passé dans la liste des actifs, retournent dans la liste des acceptés, afin d'assurer une rotation. Par contre, dans le système batch, un processus passé dans la liste des actifs devrait pouvoir y rester jusqu'à son achèvement. Aussi les facteurs à prendre en compte pour ce transfert de la liste des actifs à la liste des acceptés sont :

- le total du temps processeur reçu durant son séjour dans la liste des actifs.
- si le processus attend une entrée/sortie, faire la différence entre une attente de page ou de sémaphore.
- l'exigence du processus en "page frames" (ou cadres en mémoire centrale).

Les facteurs dont il faut tenir compte pour le passage d'un processus de la liste des attentes à la liste des acceptés sont :

- si le processus sera libre de s'exécuter
- le nombre de processus dans la liste des acceptés
- le nombre de processus dans la liste des actifs
- le nombre de processus qui se sont récemment achevés
- la priorité attachée au processus

#### B. Politique d'allocation du processeur

Elle gouverne l'allocation du temps processeur aux processus de la liste des processus actifs.

Le processeur peut devenir libre suite à une fin de processus, un transfert d'un processus hors de la liste des actifs ou une attente de page. Suivant la politique adoptée, il peut également être libéré si un processus y a terminé une tranche de temps peut-être inférieure à sa consommation totale permise pendant son séjour dans la liste des actifs.

A ce moment, les facteurs suivants doivent être pris en compte pour déterminer le prochain utilisateur du processeur

- soit le choix arbitraire d'un processus
- soit l'ancienneté des processus dans la liste des actifs
- soit la priorité des processus.

### C. Politique de pagination

Elle se subdivise en deux routines d'allocation et de dés-allocation des pages.

Il est nécessaire de déterminer :

- quand elles sont déclenchées, dans quelles circonstances
- quelles actions prendre à leur déclenchement
- les critères sur lesquels se basent cette action.

\* La routine d'allocation peut être déclenchée :

- quand un nouveau processus entre dans la liste des actifs
- quand un processus demande une page.

L'action prise à son déclenchement est :

- soit l'allocation d'une page unique
- soit l'allocation d'un nombre spécifique de pages.

Les facteurs pris en compte pour décider si l'allocation peut se faire sont :

- le nombre de pages déjà en mémoire centrale pour le processus
- l'ancienneté du processus
- la priorité du processus

\* La routine de désallocation peut être déclenchée

- quand un processus quitte la liste des actifs,
- quand la réserve des "pages frames" libres passe sous une borne fixée à priori,
- à intervalles réguliers.

L'action prise à son déclenchement est :

- soit l'ajoute d'une simple "page frame" à la réserve des pages frames libres,
- soit l'addition d'un nombre déterminé de pages frames,
- soit l'ajoute à la réserve des "pages frames" de toutes les pages satisfaisant un certain critère.

Les facteurs à prendre en compte pour décider des pages à désallouer sont :

- le temps écoulé depuis le dernier accès à la page
- si le processus propriétaire est dans la liste des actifs
- le nombre total de pages déjà possédées par le processus
- l'ancienneté du processus propriétaire
- si la page est partagée par plusieurs processus.

### III.3.2 Flux des processus

=====

Les politiques adoptées sur les systèmes particuliers doivent répondre à 2 objectifs :

- donner au système un degré de performance constant
- maximiser la proportion de temps processeur dépensée au travail utile et donc maximiser le flux du système.

Ce problème peut se décomposer par le fait que les événements concernés (changements d'état du système) se décomposent naturellement en 2 classes suivant leurs fréquences relatives :

- d'une part, ceux provoquant un changement du nombre de processus dans le système, c'est-à-dire les entrées de nouveaux processus ou les départs de processus arrivés à leur fin ; ces événements sont relativement espacés. Le contrôle de ces événements est assuré par la vanne  $V_1$ .
- d'autre part, les transitions de processus à l'intérieur du système, c'est-à-dire les passages des processus d'une file interne à une autre ; ces événements sont consécutifs à des fins de traitements élémentaires à une unité du système. Le contrôle de ces événements est assuré par la vanne  $V_2$ .

En ce qui concerne  $V_2$  :

Si trop peu de processus entrent dans la liste des actifs, il y aura du temps processuer inoccupé. Par contre, si le nombre de processus dans la liste des actifs croît trop, il y aura finalement insuffisance de pages pour satisfaire leurs besoins et cela provoquera l'écroulement du système.

(cf. figure p. 41).

Si les processus sont uniformes dans leurs exigences en pages, le nombre de processus de la liste des actifs pourra être un nombre prédéterminé, sinon, la méthode de contrôle de la vanne  $V_2$  devra se baser sur la connaissance du comportement passé des processus pour déterminer leur comportement futur. Cela entraîne donc l'utilisation d'un modèle basé sur la notion de working-set.

En ce qui concerne  $V_1$  :

Si trop de processus sont tolérés dans la liste des acceptés, la conséquence sera la dégradation du système. Le contrôle de  $V_1$  consiste donc à garder la charge totale du système dans les capacités de celui-ci.

Ainsi donc, nombre de politiques différentes peuvent être appliquées, dans le cadre d'une politique générale implémentée par le système d'exploitation, par adaptation des paramètres d'un système.

## CHAPITRE IV

### APPLICATION DU MODELE DE WILKES AU SYSTEME VAX / VMS

1. L'environnement d'un VAX
2. Balance-set et scheduling
  - 2.1 Le scheduler
  - 2.2 Le swapper
3. Description des paramètres
  - 3.1 Par processus
  - 3.2 Pour le système
4. Politiques et flux des processus dans le VAX / VMS
  - 4.1 Les valeurs des paramètres image d'une politique
  - 4.2 Influence sur les performances d'un système

Etudions à présent le fonctionnement global du système VAX / VMS et plus particulièrement l'allocation de ses ressources entre un ensemble de processus concurrents au processeur.

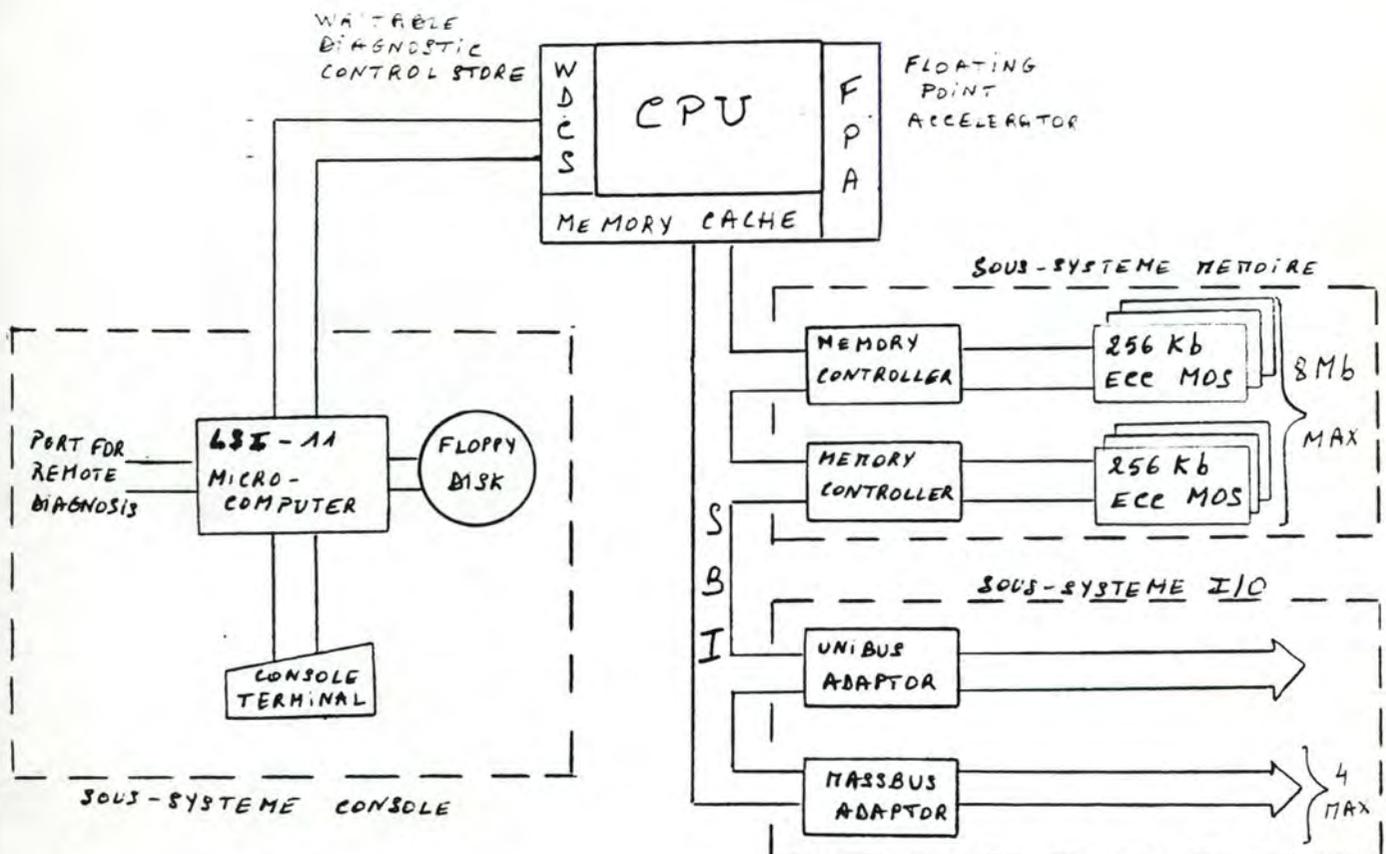
#### IV.1 L'ENVIRONNEMENT D'UN VAX

Le système consiste en une unité centrale et en 3 sous-systèmes :

- le sous-système de la console
- le sous-système de la mémoire
- le sous-système des I/O

Le CPU, les sous-systèmes de la mémoire et des I/O sont reliés par un bus de synchronisation, le SBI (synchronous backplane interconnect).

Schématiquement, le système revient à la figure suivante :



a. Le CPU.

Le CPU contient une mémoire cache pour réduire les temps d'accès mémoire. De plus, il contient un buffer de mémoire associative de 128 entrées pour accélérer le mécanisme de transformation des adresses virtuelles en adresses physiques. (address translation buffer) (16)

b. Le sous-système de la console.

Il est composé d'un micro LSI-11 avec 16 K de RAM et 8 K de ROM pour stocker le bootstrap du LSI-11, les routines de la console et les routines de diagnostics. Il comporte également un floppy disk et un interface avec le terminal console.

c. Le bus de synchronisation (SBI : synchronous backplane interconnect).

C'est le chemin de contrôle et transfert des données. Il a un cycle de 200 nanosecondes et la largeur du chemin des données est de 32 bits. Ce bus est utilisé pour l'envoi des données, mais est libéré durant le temps d'accès à la mémoire. Chaque interface a une priorité spécifique d'accès au SBI. Chaque transfert subit un contrôle de parité et est confirmé par le receveur.

d. Le sous-système de la mémoire.

Il est constitué d'un maximum de 8 contrôleurs de mémoires supportant chacun un maximum d'un Mégabytes de mémoire. La mémoire est construite en composants de 4 K bits MOS de mémoire RAM.

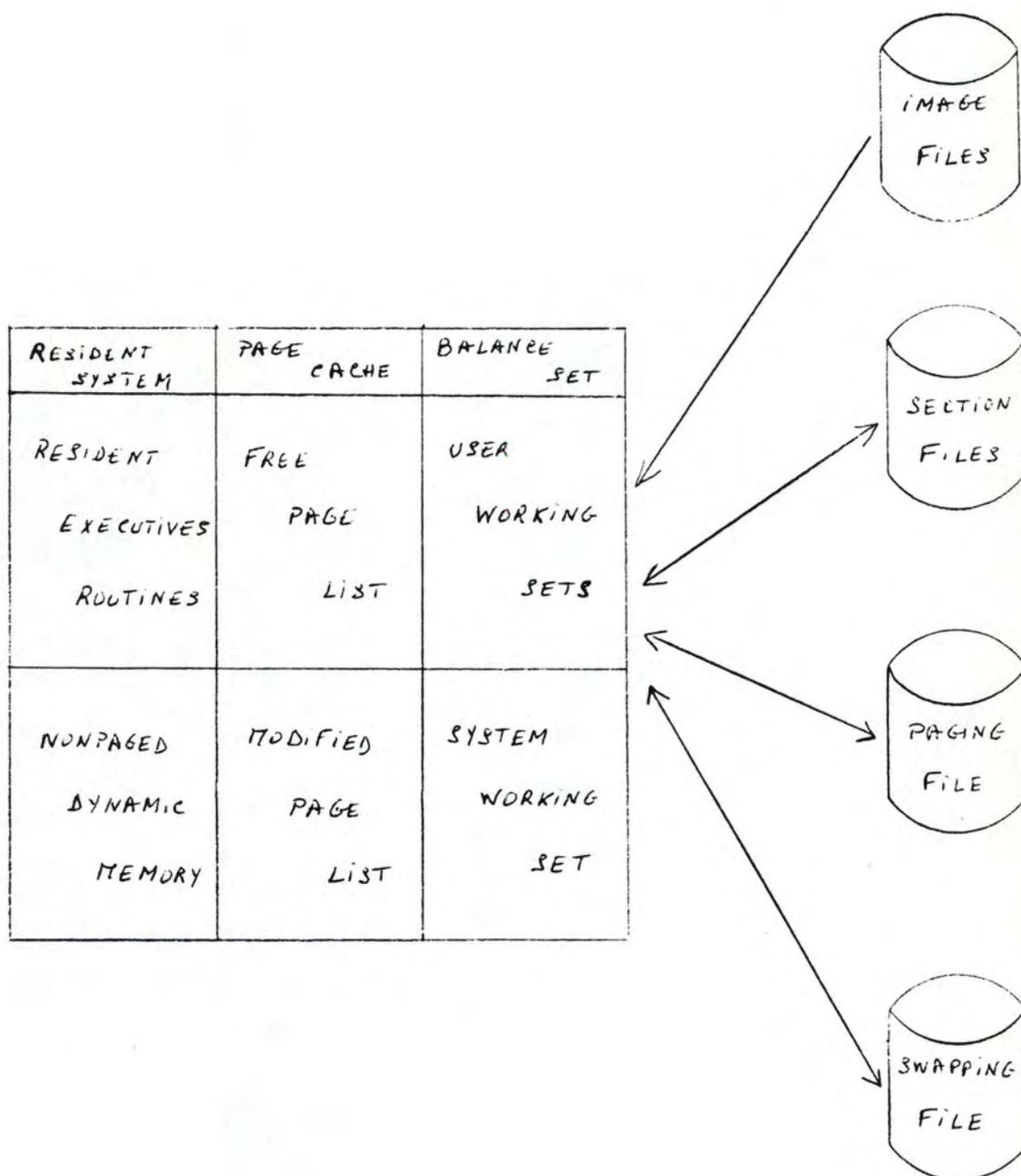
e. Le sous-système des entrées/sorties.

Il est formé d'adaptateurs entre le SBI et 2 types de bus pour périphériques utilisés sur les systèmes PDP-11 : l'unibus et le massbus. Le VAX-11/780 supporte un unibus et de 1 à 4 massbus. En plus des fonctions habituelles, ces adaptateurs effectuent la transformation des adresses de 18 bits de l'unibus ou du massbus en adresses physiques de 30 bits pour le SBI.

IV.2 BALANCE-SET ET SCHEDULING

Décrivons à présent l'organisation de l'ensemble des working-sets en mémoire centrale.

L'ensemble des mémoires dont dispose un système VAX / VMS peut être schématisé par :



Le balance-set est l'ensemble des working-sets de tous les processus résidents en mémoire centrale. Le balance-set est maintenu à jour par le processus système de swapping (swapper).

Le scheduler est la routine d'interruption qui provoquera le scheduling entre les processus exécutables. Il lui est associé des files d'état (state queue) qui sont des listes de processus par état. Celles-ci permettent au scheduler de connaître les processus éligibles au CPU.

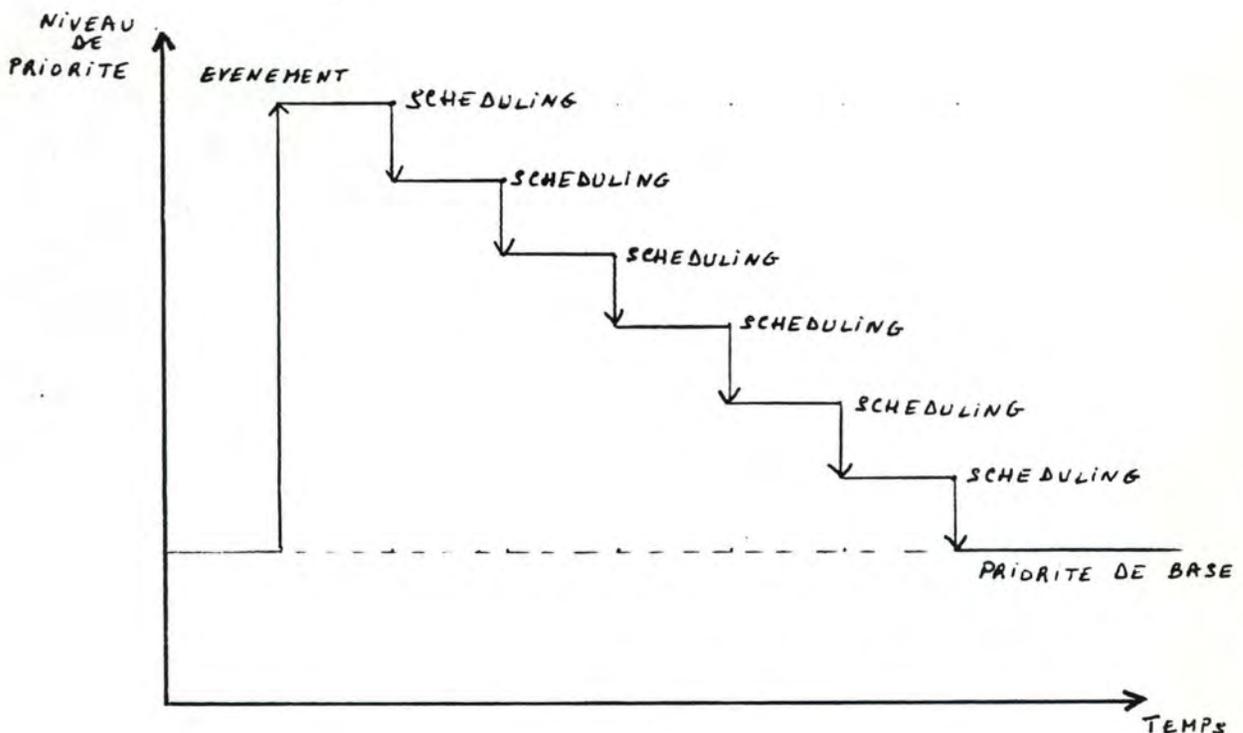
Le swapping est la méthode utilisée pour partager les ressources mémoire entre plusieurs processus en transférant l'entièreté du working-set d'un processus sur mémoire secondaire (swap-out) ou inversement en mémoire centrale (swap-in).

#### IV.2.1 Le scheduler

Le scheduling dans le système VAX / VMS est basé sur la notion de priorité associée au processus. A la création d'un processus, une priorité de base lui est associée. Le système d'exploitation reconnaît 32 niveaux de priorité de processus (du niveau le plus bas 0 au niveau le plus élevé 31). Les niveaux 16 à 31 sont réservés aux processus en temps réel tandis que les niveaux 0 à 15 sont associés aux processus normaux. Les priorités des processus en temps réel ne seront pas modifiées par le système, seule une intervention manuelle pourra le faire.

Par contre, le système pourra temporairement accroître la priorité de base d'un processus normal, puis diminuer la priorité courante du processus en fonction de l'activité du processus. Mais la priorité courante ne pourra jamais descendre sous le niveau de la priorité de base.

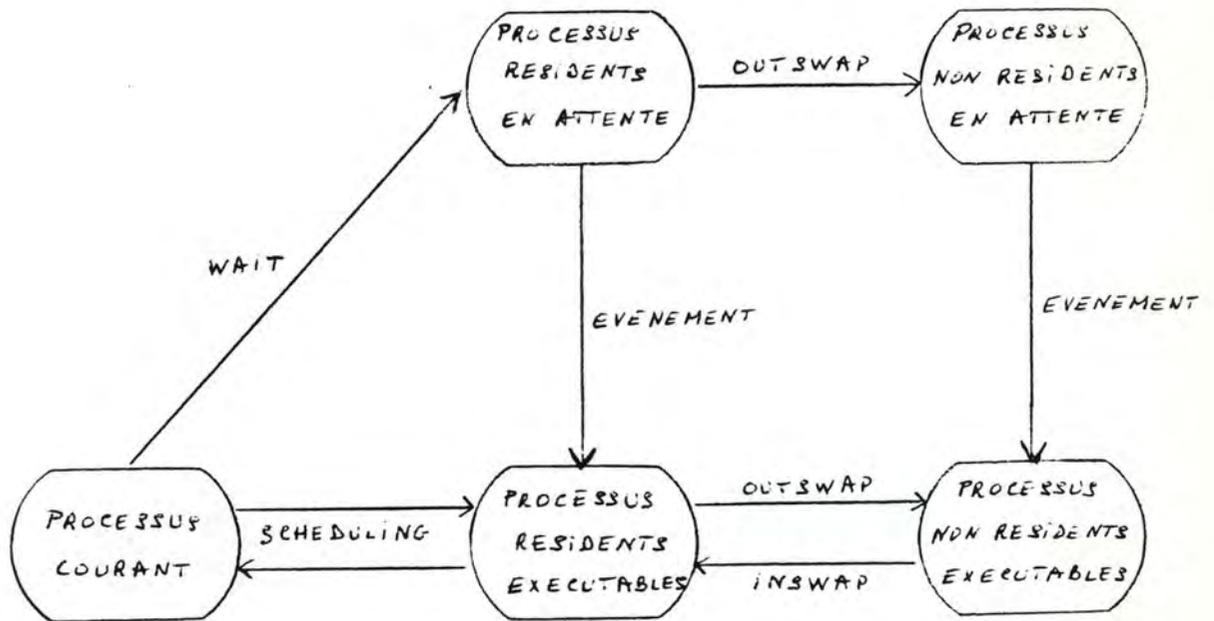
Pour un processus donné, cette variation peut se schématiser ainsi :



Cette méthode permet à la fois de gérer efficacement en parallèle des processus en temps réel et des processus normaux. D'autre part, elle permet de rectifier l'allocation du processeur en fonction de l'activité du processus.

Le rôle du scheduler et du swapper est d'assurer que les processus exécutables reçoivent du temps processeur en rapport avec leur niveau de priorité.

Leur fonctionnement peut se représenter par le schéma suivant :



Il s'en suit que le scheduler a peu de choix à faire. Il sélectionnera toujours le processus exécutable de plus haut niveau de priorité. Le scheduler est déclenché par les événements qui affectent la capacité d'un processus d'être exécutable. Ce sont :

- soit des événements externes au processus actuellement en exécution tels que des fins d'E/S, des fins de tranche de temps,...
- soit des événements internes au processus en exécution tels qu'une demande d'E/S, une demande de ressource (exemple : défaut de page),...

Chaque processus est inscrit dans une des listes par état qui identifie si le processus est exécutable, ou non et, dans ce cas, l'événement attendu ou la ressource manquante.

Lorsqu'un événement se produit, le scheduler met à jour ces listes. Il modifie la priorité des processus (en restant toujours entre la priorité de base et 15),

- en augmentant la priorité du processus pour lequel un événement s'est produit (fin d'E/S,...)
- en diminuant la priorité du processus courant en fin de tranche de temps.

Il replace ensuite le processus concerné en queue de la file de sa priorité. Ces files, suivant les priorités, doivent faciliter le choix du processus, candidat au processeur par le scheduler.

#### IV.2.2 Le\_swapper

Le swapper, quant à lui, assure que la balance-set contient toujours les processus de plus haute priorité au cas où le système n'a pas assez de mémoire centrale pour garder tous les working-sets des processus résidents en mémoire centrale.

Le swapper est activé quand un événement rend un processus non résident exécutable ou un processus résident non exécutable. Il utilise alors 3 conditions pour déterminer si un processus doit être swap-in ou swap-out :

- les états des processus
- les priorités des processus
- si le quantum de temps d'un processus a expiré.

Le quantum est à la fois le temps minimum de séjour d'un working-set dans le balance-set et la tranche de temps CPU maximum qui lui est accordé en une fois. Ainsi le quantum provoque une rotation des processus dans le balance-set tout en garantissant un temps minimum de travail à chaque processus avant d'être swappé. Un processus aura donc pu être schedulé plusieurs fois avant d'avoir eu droit à un quantum complet, mais il sera resté au minimum un quantum de temps dans le banbalance-set.

### IV.3 DESCRIPTION DES PARAMETRES

Les paramètres du système VAX / VMS se rapportent

- au processus :
  - . protections
  - . limites
  - . priorités
  - . privilèges
  
- au système :
  - . les paramètres qui affectent les opérations du système
  - . les paramètres de contrôle des jobs
  - . les paramètres associés au processus de contrôle
  - . les paramètres associés au terminal
  - . les paramètres associés aux fonctions RMS
  - . les paramètres qui contrôlent les "system communication services"
  - . les paramètres associés aux ressources et quotas à la création de processus.

#### IV.3.1 Par processus

L'ensemble des protections, limites, priorités et privilèges sont rassemblés par utilisateur dans le "User Authorization File" (UAF) qui contient entre autres :

- le nom et le mot de passe de l'utilisateur lui permettant d'entrer en contact avec le système
- le "User Identification Code" (UIC) qui identifie l'utilisateur par un numéro de groupe et un numéro de membre :
  - (xxx,yyy) où xxx : numéro de groupe
  - yyy : numéro de membre
- les limites d'utilisation des ressources du système par l'utilisateur
- la priorité de base des processus créés par l'utilisateur
- les privilèges ou limites d'action de l'utilisateur.

- a. La protection contrôle l'accès aux programmes et fichiers. Elle est basée sur le code UIC de l'utilisateur qui crée le fichier.

Les actions possibles sont

- R pour lire le fichier
- W pour écrire dans le fichier
- E pour exécuter le fichier
- D pour supprimer le fichier

Les classes possibles sont

- SYS pour l'utilisateur avec un numéro de groupe entre 0 et 7 ou avec le privilège SYSPRV
- GR pour l'utilisateur avec même numéro de groupe
- OWNER pour l'utilisateur avec même UIC
- WORLD pour n'importe qui.

La protection est alors donnée sous la forme :

PROTECTION = (SYS : RWED, GR : RWED, OWNER : RWED, WORLD : RE)  
(par défaut).

- b. Les limites aux ressources contrôlent la quantité des ressources du système et des caractères sur disque.

Par exemple : CPUTIME : temps CPU maximum qu'un processus utilisateur peut utiliser par session

MAXJOBS : nombre maximum de processus actifs en même temps, permis à l'utilisateur

PGFLQUOTA : nombre maximum de pages qu'un processus de l'utilisateur peut prendre dans le paging file

WSDEFAULT : nombre de pages d'un working-set courant par défaut



Par exemple : . SHOW PROCESS (process-name) / ACCOUNTING  
/ PRIVILEGES  
/ QUOTAS

- . SHOW PROTECTION
- . SHOW QUOTA
- . SHOW STATUS
- . SHOW SYSTEM
- . SHOW WORKING-SET

## \* SH PROCESS/ACCOUNTING

13-AUG-1983 09:30:12.68

TTA2:

User : RAHIER

## Accounting information:

Buffered I/O count :	1620	Peak working set size :	142
Direct I/O count :	269	Peak virtual size :	1299
Page faults :	2506	Mounted volumes :	0
Elapsed CPU time :	0 00:00:14.19		
Connect time :	0 00:23:33.96		

## \* SH SYS

Pid	Process Name	UIC	State	Pri	Dir.	I/O	CPU	Page flts	Ph.Mem.
00010000	NULL	000,000	COM	0		0	02:10:35.78	0	0
00010001	SWAPPER	000,000	HIB	16		0	00:00:00.41	0	0
0004003D	_TTB4:	007,004	LEF	13		19	00:00:03.73	1270	75
0005003E	RAHIER	001,004	CUK	8		505	00:00:45.54	6115	109
0001003F	PRTSYMB2	001,004	HIB	8		2	00:00:00.07	26	16 S
00010040	PRTSYMB1	001,004	HIB	8		2	00:00:00.07	26	16 S
00010041	OPCOM	001,004	LEF	5		3	00:00:00.15	46	78
00010042	JOB_CONTROL	001,004	HIB	9		21	00:00:00.76	109	121
00030043	SGAB	007,004	LEF	10		68	00:00:03.36	1020	105
00010044	DRAOBACP	001,003	HIB	10		361	00:00:36.15	2182	107
00020045	ERRFMT	001,006	HIB	4		29	00:00:00.41	28	62

## \* SH STATUS

Status on 13-AUG-1983 17:47:40.21 Elapsed CPU : 0 00:00:47.63  
 Buff. I/O : 5116 Cur. ws. : 150 Open files : 0  
 Dir. I/O : 510 Phys. Mem. : 68 Page Faults : 6386

## \* SH WORKING\_SET

working set Limit= 150 /Quota= 350 /Extent= 1024  
 Adjustment enabled Authorized Quota= 350 Authorized Extent= 1024

## \* LOGOUT/FULL

RAHIER logged out at 13-AUG-1983 17:48:03.63

## Accounting information:

Buffered I/O count:	5129	Peak working set size:	175
Direct I/O count:	510	Peak page file size:	1299
Page faults:	6446	Mounted volumes:	0
Elapsed CPU time:	0 00:00:47.88	Elapsed time:	0 01:19:17.00

#### IV.3.2 Pour le système

C'est ici qu'interviennent les paramètres qui permettent de contrôler l'ensemble du système et de l'ajuster suivant la performance voulue et la configuration disponible.

L'ensemble des 160 paramètres du système VAX / VMS (V.3.0) se subdivise en 11 catégories :

- . MAJOR les paramètres majeurs, les plus sujets à des réajus-
- . SYS les paramètres systèmes, qui affectent toute opération du système
- . JOB les paramètres de contrôle des jobs
- . ACP les paramètres associés aux processus de contrôle des I/O
- . TTY les paramètres associés au terminal
- . SCS les paramètres qui contrôlent le "System Communication Services"
- . RMS les paramètres associés aux fonctions RMS
- . PQL les paramètres associés aux limites et quotas lors de la création des processus
- . GEN les paramètres qui affectent la création et l'initialisation des structures de données au lancement du système
- . SPECIAL les paramètres spéciaux utilisés par Digital
- . DYNAMIC les paramètres dont les valeurs peuvent être réajustées pendant le fonctionnement du système, sans devoir réinitialiser le système.

Aussi, nous limiterons nous à l'analyse des 24 paramètres majeurs les plus couramment soumis à des réajustements.

NOM	DESCRIPTION	Valeur par défaut	Minimum	maximum	Unité	Dyna- mique
PFCDEFAULT	taille du cluster lu lors d'un défaut de page	32	0	127	pages	D
GBLSECTIONS	nombre de descripteurs de section globale	80	20	- 1	sections	
GBLPAGES	nombre d'entrées dans la table des pages globales	3072	512	- 1	pages	
MAXPROCESSCNT	nombre maximum de processus	72	12	8192	processus	
SYSMWCNT	taille maximale du working-set du système	160	20	16384	pages	
BALSETCNT	nombre maximum de working-sets résidents	36	4	1024	entrées	
IRPCOUNT	nombre de packets intermédiaires préalloués	60	0	32768	packets	
WSMAX	taille maximale de tout working-set	1024	60	16384	pages	
NPAGEDYN	taille du pool dynamique non paginé	64000	16384	- 1	bytes	
PAGEDYN	taille du pool dynamique paginé	80000	8192	- 1	bytes	

NOM	DESCRIPTION	Valeur par défaut	Minimum	Maximum	Unité	Dyna- mique
VIRTUALPAGECNT	espace virtuel maximum par processus	8192	512	262144	pages	
LRPCOUNT	nombre de grands packets préalloués	4	0	4096	packets	
SRPCOUNT	nombre de petits packets préalloués	120	0	4096	packets	
QUANTUM	temps maximum qu'un processus peut utiliser en une fois et temps minimum qu'un processus doit recevoir avant d'être swappé	20	2	32767	10 milli- sec	D
PFRATL	limite inférieur du taux de défauts de page	1	0	- 1	défauts/ 10 sec.	D
PFRATH	limite supérieure du taux de défauts de page	120	0	- 1	défaut/ 10 sec.	D
WSINC	incrémentation du working-set	150	0	- 1	pages	D
WSDEC	décrémentation du working-set	35	0	- 1	pages	D
FREELIM	taille minimum de la liste des pages libres	32	16	- 1	pages	

NOM	DESCRIPTION	Valeur par défaut	Minimum	Maximum	Unité	Dyna- mique
FREEGOAL	nombre de pages exigées dans la liste des pages libres après une pénurie	200	16	- 1	pages	
GROWLIM	nombre de pages exigées dans la liste des pages libres pour permettre à un processus de dépasser son quota	63	0	- 1	pages	D
BORROWLIM	taille minimum de la liste des pages libres avant qu'un processus puisse s'étendre au-delà de son quota	300	0	- 1	pages	D
LOCKIDTBL	nombre d'entrées dans la table des verrous du système	128	16	16000	entrées	
RESHASHTBL	nombre d'entrées dans la table des noms des ressources verrouillables	32	1	8192	entrées	

#### 1. PFCDEFAULT

Pendant l'exécution des programmes, PFCDEFAULT contrôle le nombre de pages lues sur disque, depuis le fichier image (image file) ou depuis le fichier de pagination (paging file), par opération I/O, lorsqu'un défaut de page se produit.

La valeur de ce paramètre doit être suffisamment grande pour assurer la performance des I/O, mais ne doit pas être trop élevée par rapport au working-set moyen sans quoi elle risque de forcer le déplacement d'une partie importante du working-set du processus.

#### 2. GBLSECTIONS

Ce paramètre indique le nombre de descripteurs de sections globales allouées au système au moment du lancement. Il sera fonction du nombre potentiel d'images partagées que les processus voudront utiliser.

#### 3. GBLPAGES

Ce paramètre indique le nombre d'entrées de la table des pages globales allouées au moment du lancement. Chaque section globale requiert une entrée dans la table des pages globales par page dans la section.

#### 4. MAXPROCESSCNT

MAXPROCESSCNT détermine le nombre maximal de processus concurrents admis dans le système.

## 5. SYSMWCNT

SYSMWCNT est la taille maximale du working-set du système (qui contient la partie paginable du système). Une valeur trop élevée réserve de la place mémoire au détriment des working-sets des processus utilisateurs tandis qu'une valeur trop faible peut sérieusement ralentir le système en le forçant à paginer excessivement.

## 6. BALSETCNT

Ce paramètre détermine le nombre maximum de processus dans le balance-set, c'est-à-dire de working-sets résidents en mémoire centrale. Ce nombre sera donc toujours inférieur à MAXPROCESSCNT. Diminuer ce paramètre a pour effet d'augmenter la taille de la liste des pages libres, mais à ce moment, il se peut que le taux de swapping croisse, ce qui peut affecter la performance du système.

## 7. IRPCOUNT

Ce paramètre fixe le nombre de "packets" moyens (160 bytes) préalloués pour les E/S. S'il est surdimensionné, une partie de la mémoire centrale est gaspillée.

## 8. WSMAX

WSMAX est le nombre maximum de pages pour un working-set. Ce paramètre affecte la taille du swapping file ainsi que la taille de mémoire allouée à la partie du système résident en permanence en mémoire centrale.

## 9. NPAGEDYN

Ce paramètre détermine la taille du pool dynamique non paginé. Il fixe en fait la valeur initiale qui peut être augmentée dynamiquement, si nécessaire, mais au prix d'une perte de mémoire centrale supplémentaire. S'il est surdimensionné, une partie de la mémoire centrale est gaspillée.

## 10. PAGEDYN

Ce paramètre détermine la taille du pool dynamique paginé. Ce pool de pages sert à stocker les noms logiques, les entêtes des images résidentes, les structures partagées pour les fichiers RMS.

## 11. VIRTUALPAGECNT

VIRTUALPAGECNT est le nombre maximum de pages virtuelles d'un processus.

## 12. LRPCOUNT

Ce paramètre fixe le nombre de grands "packets" (640 bytes) préalloués pour les E/S. S'il est surdimensionné, une partie de la mémoire centrale est gaspillée. S'il est trop petit, le système accroît automatiquement sa valeur, suivant ses besoins, mais au prix d'une perte de mémoire centrale supplémentaire.

## 13. SRPCOUNT

Ce paramètre fixe le nombre de petits "packets" (96 bytes) préalloués pour les E/S. S'il est surdimensionné, une partie de la mémoire centrale est gaspillée. S'il est trop petit, le système accroît automatiquement sa valeur, suivant

ses besoins, mais au prix d'une perte de mémoire centrale supplémentaire.

#### 14. QUANTUM

Ce paramètre définit :

a. Au point de vue du processeur

C'est le temps processeur maximum qu'un processeur peut recevoir en une fois avant de passer le contrôle à un processus d'égale priorité qui est prêt à s'exécuter.

b. Au point de vue du balance-set

C'est le temps processeur minimum qu'un processus exécutable doit recevoir avant d'être éjecté du balance-set.

Son unité est la tranche de 10 millisecondes.

#### 15. PFRATL

PFRATL spécifie le nombre de défauts de pages sous lequel un working-set est automatiquement diminué. L'unité de mesure est la tranche de 10 secondes de temps processeur. Par exemple, s'il est positionné à 1, cela signifie que le système diminue automatiquement le working-set, si le processus a exécuté moins d'un défaut de page par 10 secondes. Par conséquent, augmenter la valeur de PFRATL tend à diminuer les working-sets des processus tandis que la diminuer aura tendance à accroître la taille des working-sets.

## 16. PFRATH

Ce paramètre est le pendant du précédent. Il spécifie le nombre de défauts de pages au-dessus duquel un working-set est automatiquement agrandi. L'unité de mesure est la tranche de 10 secondes de temps processeur. Par exemple, s'il est positionné à 120, cela signifie que le système agrandit automatiquement le working-set si le processus a exécuté plus de 120 défauts de pages par 10 secondes. Par conséquent, diminuer la valeur de PFRATH tend à agrandir les working-sets des processus tandis que l'augmenter aura tendance à diminuer la taille des working-sets.

## 17. WSINC

Ce paramètre spécifie le nombre de pages dont est automatiquement augmenté un working-set à chaque fois qu'un réajustement croissant est requis (en fin de quantum). Diminuer la valeur de ce paramètre tend à ralentir la croissance des working-sets à fort taux de défauts de page.

## 18. WSDEC

Ce paramètre est le pendant du précédent. Il spécifie le nombre de pages dont est automatiquement diminué un working-set à chaque fois qu'un réajustement décroissant est requis (en fin de quantum). Augmenter la valeur de ce paramètre tend à accélérer la décroissance des working-sets à faible taux de défauts de page.

## 19. FREELIM

FREELIM fixe le nombre minimum de pages que doit comporter la liste des pages libres. Pour maintenir ce nombre le sys-

tème devra recopier les pages de la liste des pages modifiées, éjecter des working-sets de la mémoire par swapping ou réduire la taille des working-sets.

Une liste des pages libres importante signifie moins de pagination avec opération I/O, mais cela implique moins d'espace pour le balance-set et donc une tendance à plus de swapping.

#### 20. FREEGOAL

Ce paramètre spécifie le nombre de pages souhaitées dans la liste des pages libres suite à une pénurie (passage sous le nombre minimum FREELIM). Il sera donc toujours supérieur ou égal à FREELIM.

#### 21. GROWLIM

Ce paramètre détermine le nombre minimum de pages de la liste des pages libres pour qu'un processus puisse ajouter une page à son working-set lorsqu'il a déjà atteint son quota. GROWLIM n'a donc aucun effet si le working-set du processus est sous son quota.

Ce paramètre a donc pour effet de n'autoriser l'extension d'un working-set que pour autant qu'il y ait suffisamment de mémoire disponible.

#### 22. BORROWLIM

Ce paramètre détermine le nombre minimum de pages de la liste des pages libres nécessaire pour que le working-set d'un processus puisse dépasser son quota. Ce paramètre devra donc toujours être supérieur à FREELIM.

Ce paramètre fixe donc l'excédent de pages libres nécessaire avant de permettre à un working-set de s'étendre, ce qui lui permettra d'alléger la charge des défauts de page.

### 23. LOCKIDTBL

Ce paramètre établit le nombre d'entrées dans la table des verrous du système, ce qui limite le nombre de verrous du système.

### 24. RESHASHTBL

Ce paramètre définit le nombre d'entrées dans la table des noms des ressources verrouillables.

En dehors de ce paramètres dits "majeurs" nous en avons relevé quelques autres dont l'influence sur le système pourrait être significative.

#### . MPW-WRTCLUSTER

Ce paramètre fixe le nombre de pages qui seront réécrites par cluster à partir de la liste des pages modifiées vers le paging file ou un fichier ordinaire, en une seule opération I/O. Si MPW-WRTCLUSTER est trop petit, il faudra beaucoup d'opérations I/O pour vider la liste des pages modifiées.

#### . MPW-HILIMIT

Ce paramètre fixe la limite supérieure de la liste des pages modifiées. Dès que le nombre de pages de la liste des pages modifiées devient supérieur à cette limite, il y a réécriture des pages dans le paging file et transfert des "pages frame" (ou cadres de la mémoire centrale) vers la liste des pages libres.

Si MPW-HILIMIT est trop bas, il y aura un nombre excessif de défauts de page résolus par le paging file. Par contre, s'il est trop élevé, un nombre de pages mémoire trop important sera consommé.

. MPW-LOLIMIT

Ce paramètre fixe la limite inférieure de la liste des pages modifiées. Lorsque le nombre de pages de la liste des pages modifiées tombe sous cette limite, la réécriture des pages est stoppée.

MPW-LOLIMIT assure qu'un certain nombre de pages seront disponibles dans la liste des pages modifiées pour la résolution des défauts de page. Si ce paramètre est trop petit, cette liste ne parvient plus à jouer son rôle d'extension des working-sets. S'il est positionné trop haut, moins de mémoire est disponible pour les processus, aussi le swapping risque-t-il d'augmenter.

La différence entre MPW-HILIMIT et MPW-LOLIMIT indiquera le nombre d'opérations I/O à exécuter lors de la réécriture de pages de la liste des pages modifiées, et cela, en fonction de MPW-WRTCLUSTER. Si on augmente MPW-LOLIMIT ou si on diminue MPW-HILIMIT, on réduit le nombre d'opérations I/O lors de la réécriture, mais on risque d'accroître le taux de swapping.

Toujours dans le cadre des paramètres gérant la taille des working-sets :

- . AWSMIN détermine le nombre minimum de pages qu'un working-set peut atteindre par décroissance lors des réajustements automatiques.

- . AWSTIME spécifie le temps processeur minimum pour calculer le taux de défauts de page d'un working-set. Ce temps est exprimé en tranche de 10 millisecondes.

Nous reviendrons sur leur influence dans le système, dans le chapitre suivant concernant le réglage du système.

#### IV.4 POLITIQUES ET FLUX DES PROCESSUS DANS LE VAX / VMS

Dans la suite, nous allons examiner le rôle des paramètres du système VAX / VMS à la lumière du modèle de WILKES.

##### IV.4.1 Les valeurs des paramètres, image d'une politique (2)

"Les vannes (fig. p. 43) entre la liste des processus acceptés et la liste des processus actifs (swap-in et swap-out) sont réalisées par un processus : le "swapper". Il gère l'introduction des processus en mémoire centrale et le retrait de ceux-ci. Son action dépend de la politique implémentée dans le système et de valeurs données à certains paramètres du système. La politique est inchangeable, mais son application peut être adaptée par le choix des paramètres.

Les paramètres du système MAXPROCESSCNT et BALSETCNT fixent respectivement le nombre maximum des processus présents dans le système et le nombre maximum de processus résidents en mémoire centrale. Le "swapper" tente d'assurer la valeur de BALSETCNT, c'est-à-dire le nombre de processus dans la liste des processus actifs.

D'autre part, le paramètre FREELIM fixe la taille de la liste des pages libres. Si sa taille descend en-dessous de cette valeur, le swapper recopie des pages de la liste des pages modifiées, s'il y en a plus que le minimum défini par le paramètre MPW-LOLIMIT. Si cette opération ne suffit pas, le "swapper" choisit un processus de la liste des actifs et le transfère dans la liste des processus acceptés. Ce transfert implique la copie sur disque de toutes les pages appartenant au working-set du processus choisi, libérant ainsi des pages qui rejoignent la liste des pages libres. Ce transfert d'un working-set s'appelle un "outswap" et correspond à la vanne  $V_3$  du modèle de Wilkes.

Pour choisir le processus qui sera transféré, le "swapper" essaye de déterminer le processus qui a la possibilité la plus faible d'être exécutable à bref délai (par exemple : parmi les processus non exécutables, on choisira celui en attente d'une réponse d'un terminal plutôt qu'un processus attendant la fin d'une E/S sur disque). S'il n'y a que des processus exécutables, le swapper en choisit un en fonction de sa priorité.

La liste des processus acceptés contient donc des processus non exécutables qui y ont été amenés par le swapper. Ces processus sont non-exécutables tant qu'un événement n'a pas eu lieu (fin d'une E/S, par exemple). Dès qu'un de ces processus devient exécutable, le "swapper" va essayer de récupérer des pages libres afin de ramener le working-set de ce processus en mémoire centrale. Pour libérer des pages, il peut être amené à transférer certains processus de la liste des actifs à la liste des acceptés. Le transfert d'un working-set de la liste des acceptés à la liste des actifs s'appelle un "inswap" et correspond à la vanne  $V_2$  de Wilkes. Le nombre de processus dans la liste des processus actifs correspond au degré de multiprogrammation du système. Le nombre maximum en est fixé dans le VAX / VMS par le paramètre BALSETCNT. Sa valeur, autrement dit le réglage de la vanne  $V_2$ , est fondamental pour l'utilisation efficace du système.

En effet, si on n'admet trop peu de processus, le CPU risque de rester inactif, si on n'admet trop de processus, la mémoire risque d'être engorgée, et le système s'écroule sous le nombre de défauts de page.

Par un choix adéquat de MAXPROCESSCNT et BALSETCNT, on peut adapter le rôle joué par le swapping aux objectifs de chaque installation. On peut soit, considérer que le swapping fait partie de la routine du système c'est-à-dire qu'il n'est pas anormal qu'un processus soit transféré plusieurs fois d'une liste à l'autre (le système garantit de toute façon qu'un processus de la liste des actifs ne peut être candidat au transfert que

s'il a déjà progressé d'au moins un quantum depuis son entrée dans le balance-set), soit se poser sur le swapping comme sur une soupape de sécurité intervenant en cas de surcharge quant à l'utilisation de la mémoire.

#### IV.4.2 Influence sur les performances d'un système

L'ingénieur système a la responsabilité d'optimiser à la fois la performance et l'efficacité du système. Pour prendre ses décisions, il devra à la fois avoir une bonne connaissance des besoins des utilisateurs et des capacités du système d'exploitation du VAX / VMS. Ce sont ces dernières que nous développerons dans ce paragraphe.

Dans un système à mémoire virtuelle, il est certain que les capacités mémoire physique primaire et secondaire jouent un rôle primordial. Mais d'autre part, un système VAX / VMS à la limite de sa charge devrait pouvoir être sensiblement amélioré du point de vue de ses performances par un meilleur ajustement de ses paramètres à sa propre charge. C'est ce que nous allons essayer de justifier dans ce paragraphe.

Nous partirons des hypothèses :

1. Les ressources hardware sont adaptées à la charge
2. La charge est distribuée de la façon la plus équilibrée possible.
3. Les processus travaillent en code partagé le plus fréquemment possible.

Bien que les facteurs influençant la performance d'un système VAX / VMS apparaissent être très imbriqués, nous en étudierons quatre principaux :

- la taille des working-sets
- les caractéristiques des listes de pages libres et modifiées
- la priorité des processus
- le quantum.

#### A. La taille des working-sets

Diminuer la taille limite des working-sets revient à augmenter le nombre de working-sets pouvant résider en mémoire centrale et donc à réduire le besoin en swapping, mais aussi à accroître le besoin en pagination. Cependant, il semble qu'une pagination importante soit de loin préférable à un swapping important car :

- le swapping correspond à un transfert aller et retour d'un working-set complet, tandis que la pagination ne concerne habituellement qu'un sous-ensemble du working-set,
- la pagination n'implique pas nécessairement des opérations physiques d'I/O sur disque grâce aux listes des pages libres et modifiées.

Les limites initiales des working-sets peuvent être fixées par le paramètre propre à chaque utilisateur WSDEFAULT.

D'autre part, le système utilise un ajustement automatique des limites du working-set, basé sur le comportement observé de celui-ci, grâce aux paramètres WSINC (WSDEC), PFRATH (PFRATL) et AWSTIME. Ceux-ci permettront même à un processus de s'étendre au-delà de la valeur WSQUOTA jusqu'à WSEXTENT

pour autant que le système ait une réserve suffisante de pages libres. Cette augmentation atténuera le taux de pagination.

. Ainsi un taux de pagination pourra être diminué :

- soit en réduisant la valeur de PFRATH
- soit en réduisant la valeur de PFRATL  
en accroissant la valeur de WSINC  
et en décroissant la valeur de WSDEC.

. De même, un taux de swapping pourra être diminué :

- soit en accroissant la valeur de PFRATH
- soit en augmentant la valeur de PFRATL,  
en diminuant celle de WSINC  
et en accroissant celle de WSDEC, pour autant que les working-sets soient de petites tailles et en espérant accroître le nombre de working-sets dans le balance-set.

#### B. Les caractéristiques des listes de pages libres et modifiées

Quoique les working-sets et les listes de pages libres et modifiées soient gérés par la méthode FIFO, le retour dans les working-sets de pages à nouveau référencées des listes de pages libres ou modifiées a pour effet d'éjecter de la mémoire les pages les moins fréquemment utilisées et retenir celles le plus fréquemment utilisées.

L'efficacité de ce système dépendra à la fois de la taille de ces listes en fonction de leur utilisation et de la taille des

clusters (ou bouquets) de pages lues sur disque pour les working-sets ou écrites sur disque à partir de la liste des pages modifiées.

- Au point de vue de la taille des listes

Le paramètre système FREELIM donne la taille minimum de la liste des pages lues. Mais la liste pourra s'étendre tant qu'il y a des pages disponibles.

En plus de l'ajustement des working-sets, celui de FREELIM peut diminuer le pourcentage de défauts de pages résultant en une lecture sur disque en accroissant la valeur de FREELIM. De même un haut taux de swapping peut indiquer que la valeur de FREELIM doit être réduite pour donner plus de place au balance-set.

La liste des pages modifiées est réglée par 3 paramètres : MPW-HILIMIT, MPW-LOLIMIT, et MPW-WRTCLUSTER. Le réglage de ces paramètres peut-être déterminé par le nombre d'opérations d'écriture sur disque et le taux de swapping. Augmenter les limites inférieures et supérieures de la liste des pages modifiées normalement diminue le taux d'opération I/O en écriture. Mais le taux de swapping pourrait s'accroître car des valeurs arbitrairement grandes de FREELIM et MPW-LOLIMIT réduisent le nombre de pages restantes dans le balance-set en mémoire centrale.

- Au point de vue de la taille des clusters

Le paramètre système PFCDEFAULT contrôle le nombre de pages lues du disque vers les working-sets en une opération I/O. Une taille de cluster trop importante par rapport au working-set provoque une rotation importante de pages

dans le working-set, car les pages transférées éjectent des pages du working-set dès que celui-ci atteint sa limite.

#### C. La priorité des processus

=====

La priorité des processus permet de refléter les temps de réponse relatifs tolérés par les différents processus. Ainsi les processus batchs seront munis d'une priorité de base la plus basse, si l'on désire privilégier les temps de réponses des travaux interactifs.

Cependant, en diminuant la priorité de base des processus des utilisateurs, on obtient un effet secondaire au but poursuivi (influencer la stratégie d'allocation du CPU). En effet, les processus les plus actifs auront le plus de pages sorties de leur working-set. Ainsi on augmente le coût moyen en temps du service d'un défaut de page pour ces processus.

#### D. Le quantum

=====

Un quantum important assure en général un meilleur service dans la mesure où il signifie des swappings moins fréquents, pour autant que les tailles des working-sets des utilisateurs actifs dépassent le nombre de "pages frames" (ou cadres en mémoire centrale) disponibles. Sans quoi un quantum légèrement inférieur au temps moyen d'exécution d'une demande I/O des processus utilisateurs, permet une meilleure rotation des processus au processeur, et cela, sans désavantager les processus effectuant beaucoup d'I/O par rapport aux autres processus.

De plus, dans ce cas, les working-sets seront réajustés plus souvent (à la fin de chaque quantum), c'est-à-dire que les tailles des working-sets seront mieux adaptées et le taux de pagination devrait diminuer.

## CHAPITRE V

### REGLAGE DU SYSTEME

#### 1. Objectifs et méthodes

1.1 Objectifs des mesures

1.2 Environnement expérimental

1.3 Méthodes de prise des mesures

1.4 Connaissance de la charge

#### 2. Comportement du système vis-à-vis de la charge

2.1 Description des paramètres initiaux

2.2 Résultats des mesures

#### 3. Degré d'ajustement du système à la charge

3.1 Variations des paramètres

3.2 Analyse des résultats

3.3 Evaluation des performances

Ce chapitre se propose de relater les différents tests réalisés sur un VAX-11/780 afin de constater l'influence réelle des paramètres sur les performances d'un système.

## V.1 OBJECTIFS ET METHODES

### V.1.1 Objectifs des mesures

Dans le cadre d'un centre informatique, le gestionnaire a la responsabilité d'exploiter la charge qui lui est fournie avec les moyens dont il dispose dans les meilleures conditions d'exécution possibles. Il doit donc déterminer quelles sont les modifications souhaitables à imposer à la charge ou au système afin d'améliorer les conditions d'exécution.

Dans le cadre de ces tests, nous sommes partis de l'hypothèse que la charge était fixée et il n'était pas possible au gestionnaire de la modifier. Le but de ce réglage se réduit alors à la réduction des pertes d'efficacité dues à la mauvaise adaptation du système à la charge.

Suivant les conclusions théoriques du chapitre précédent, nous avons envisagé de faire varier certains paramètres du système afin d'étudier l'évolution de son comportement en vue d'une optimisation du système par l'observation des files d'attente et la réduction des opérations non productives.

Dans un premier temps, nous avons envisagé de créer un benchmark complet afin de déterminer le meilleur ajustement des paramètres du système à la charge. Malheureusement, des contraintes temporelles nous empêché de développer ce benchmark et nous avons dû nous résoudre à réaliser certains tests élémentaires.

Aussi, nous nous sommes finalement limités à l'étude de l'influence des variations des paramètres du système relatifs à la gestion de la mémoire virtuelle. Ceux-ci sont, en effet, une des particularités les plus marquantes du système VAX / VMS. Leur étude permet d'analyser le comportement réel de cette gestion de mémoire virtuelle au niveau des défauts de page, du swapping, des opérations I/O nécessitées par cette gestion et de l'utilisation effective de la mémoire. Cette étude a été réalisée sur base d'un benchmark fourni par Digital (UETP).

Finalement, étant donné le temps machine qui nous était imparti, nous n'avons pas pu réaliser ces tests un nombre de fois suffisamment grand que pour en tirer de réelles statistiques. Malgré cela, leurs résultats peuvent quand même nous fournir certaines indications sur l'orientation que prendrait une analyse plus précise.

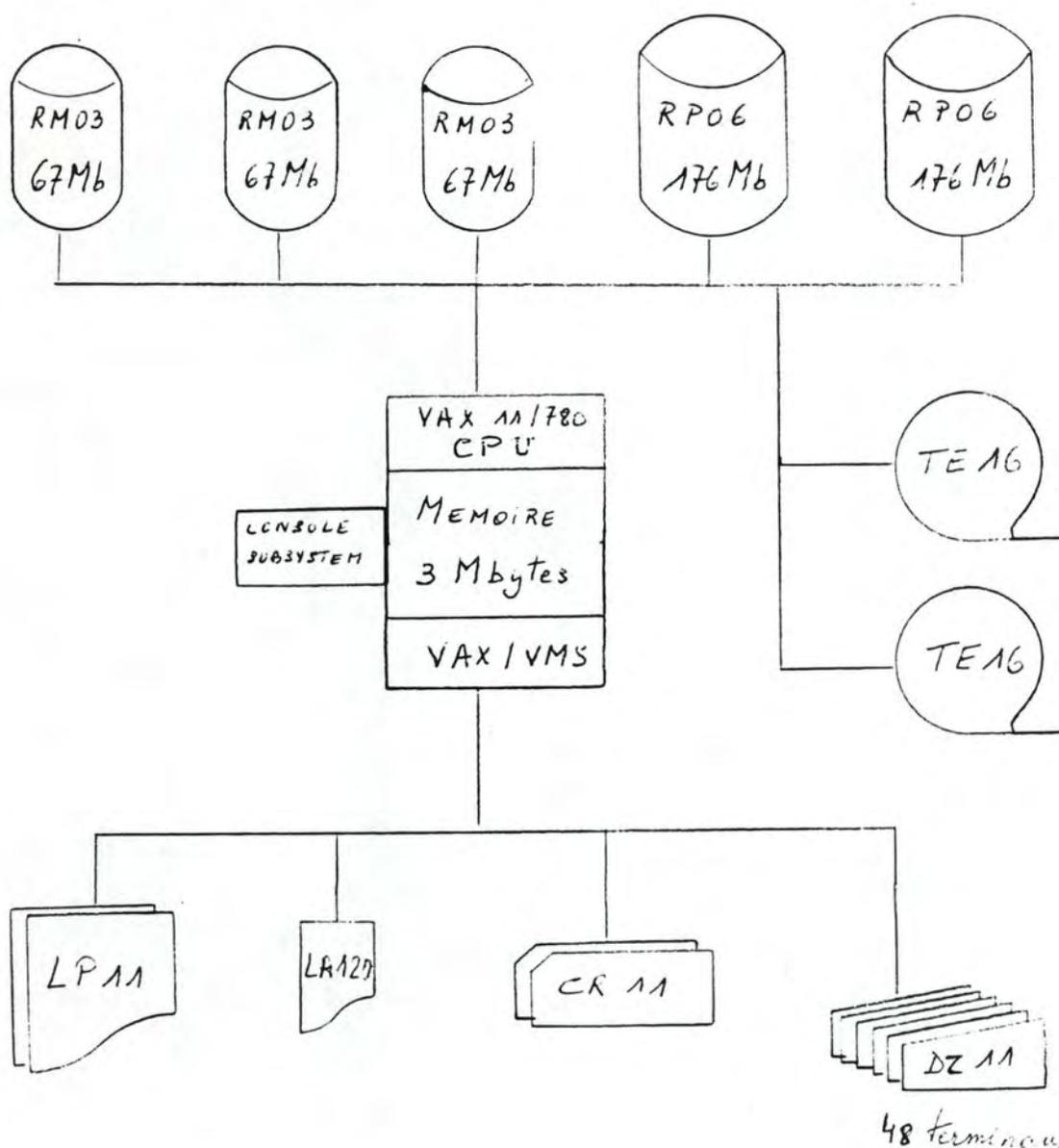
#### V.1.2 Environnement expérimental

La configuration sur laquelle se sont déroulés les tests est un VAX 11/780 avec 3 mégabytes de mémoire centrale (6144 pages) dont 1292 pages sont allouées en permanence au système d'exploitation VMS. En outre, il est muni de 3 disques RMO3 et 2 disques RPO6. Chaque disque RMO3 a une capacité de 67 mégabytes, un taux de transfert de 1200 Kbytes/sec et un temps d'accès moyen de 0.0383 secondes. Chaque disque RP06 a une capacité de 176 Mégabytes, un taux de transfert de 806 Kbytes/sec et un temps d'accès moyen de 0.0363 secondes. Le système possède également 2 armoires à bande magnétique TE16. Il possède pour périphériques lents : une imprimante LA120 (120 caractères/sec), 2 imprimantes LP11 (1000 lignes/min), 48 terminaux et 2 lecteurs de carte.

Le système d'exploitation est le système à mémoire virtuelle de DEC (VAX / VMS). Il supporte en concurrence le développement et l'exécution de programmes time-sharing, batch ou temps réel

écrit en BASIC-11 ou FORTRAN. La version est actuellement la V3.2. VAX supporte 2 langages de commandes (CLI : command language interpreter et DCL : digital command language), un éditeur et divers utilitaires.

Lors des tests, un seul disque était accessible. Un seul canal cumulait donc les I/O des utilisateurs, les I/O venant de la pagination et du swapping.



### V.1.3 Méthodes de prise des mesures

L'outil de mesure employé durant les tests est un logiciel de mesure fournit par Digital : "MONITOR".

Ce logiciel de mesure a l'avantage d'être indépendant vis-à-vis des éléments observés. Il procède principalement par lecture des tables du système. Ce mesureur est introduit comme une tâche indépendante parmi l'ensemble des tâches concurrentes. Quand le processeur lui sera alloué, elle pourra effectuer les mesures. Une fois effectuées, elle se suspend pour un temps déterminé. Il s'en suit une certaine imprécision quant à la périodicité exacte de l'échantillonnage qui dépend de l'environnement. D'autre part, la fréquence d'échantillonnage doit être suffisante pour observer avec une bonne résolution les modifications de l'état du système, mais elle augmente la surcharge due à la prise de mesure. Puisque le mesureur est une tâche concurrente, il introduit une perturbation dans le système. Les temps et ressources nécessaires aux mesures seront mesurés eux-mêmes. Ces quantités sont approximativement équivalentes à celles utilisées par un utilisateur simulé dans le benchmark.

"MONITOR" permet d'obtenir des mesures de différents types sur les performances du système d'exploitation vis-à-vis de la charge qui lui est soumise :

- DECNET : Statistiques sur le réseau DECNET-VAX
- FCP : Statistiques sur le fichiers
- IO : Statistiques sur les opérations I/O
- LOCK : Statistiques sur la gestion des verrous
- MODES : Temps passé dans chaque mode par les processus
- PAGE : Statistiques sur la gestion des pages mémoires
- POOL : Statistiques sur l'espace alloué au pool dynamique non paginé
- PROCESSES : Informations sur tous les processus dans le système
- STATES : Nombre de processus dans chacun des états.

Il permet de spécifier l'intervalle d'activation du mesureur et la présentation des résultats (voir annexe). Ceux-ci seront, soit stockés de façon brute en code binaire, soit synthétisés afin de représenter le comportement globalisé des variables observées pour la période, ou soit encore divers tableaux donneront la chronologie du comportement des variables observées.

En ce qui concerne l'observation de la gestion de la mémoire virtuelle, nous nous sommes basés sur les statistiques concernant la gestion des pages de la mémoire centrale (PAGE). L'intervalle d'activation considéré fut de 3 secondes.

#### V.1.4 Connaissance de la charge

La charge fut simulée grâce à un logiciel fournit par Digital : l'User Environment Test Package (UETP).

l'UETP simule un environnement typiquement time-sharing. En fait, il crée une série de processus détachés qui exécutent une procédure de commande. Chacun des processus simule un utilisateur à son terminal. Chacune des procédures est formée de différentes commandes du langage DCL. Celles-ci se déroulent en parallèle et ont, sur le système, un effet similaire à un ensemble d'utilisateurs interactifs.

Cette charge a été choisie car elle met particulièrement à l'épreuve la capacité multi-utilisateur du VAX. Par conséquent, avec un nombre important d'utilisateurs, la place mémoire devient critique. Dès lors, cela permet de mettre en évidence le mécanisme de gestion de la mémoire virtuelle.

Le nombre d'utilisateurs considérés pour les tests fut de 30. Ce chiffre fut déterminé suite à quelques essais au cours dequels on constata :

- un nombre inférieur n'apparaissait pas être un nombre critique pour ce qui est de l'occupation mémoire,
- un nombre supérieur provoquait un taux relativement élevé de swapping.

D'autre part, habituellement, l'installation tourne avec un nombre moyen de 35 processus.

Les conditions de réalisation des tests étant à présent posées, dans la suite nous décrirons et analyserons les phénomènes observés.

## V.2 COMPORTEMENT DU SYSTEME VIS-A-VIS DE LA CHARGE

Rappelons brièvement les conditions de tests :

1. La configuration est un VAX 11/780 de 3 mégabytes de mémoire centrale muni d'un seul disque RM03.
2. Les mesures sont prises par le logiciel de mesure "MONITOR" dont l'intervalle d'activation est de 3 secondes.
3. La charge est simulée par le logiciel "UETP". Elle est constituée de 30 utilisateurs interactifs, des 8 jobs systèmes permanents (NULL, SWAPPER, PRYSYMB2, PRYSYMB1, OPCOM, JOB-CONTROL, DRAOBACP, ERRFMT), du batch qui exécute le programme "MONITOR" et d'un utilisateur interactif supplémentaire, observateur du comportement du système (SHOW MEMORY, SHOW SYSTEM).

### V.2.1 Description des paramètres initiaux

Dans une première étape, le benchmark a tourné avec les paramètres du système tels qu'ils étaient réglés pour la configuration. Cela permet de déterminer les valeurs "étalons" de référence pour les tests suivants.

Voici les paramètres majeurs initiaux :

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
PFCDEFAULT	127	32	0	127	Pages	Dynamic
GRLSECTIONS	100	80	20	-1	Sections	
GRLPAGES	3326	3072	512	-1	Pages	
MAXPROCESSCNT	70	72	12	8192	Processes	
SYSMWCNT	247	160	20	16384	Pages	
BALSETCNT	52	36	4	1024	Slots	
IRPCOUNT	750	60	0	32768	Packets	
WSMAX	3072	1024	60	16384	Pages	
NPAGEDYN	179712	64000	16384	-1	Bytes	
PAGEDYN	84992	80000	8192	-1	Bytes	
VIRTUALPAGECNT	16384	8192	512	262144	Pages	
LRPCOUNT	8	4	0	4096	Packets	
SRPCOUNT	380	120	0	4096	Packets	
QUANTUM	20	20	2	32767	10Ms	Dynamic
FFRATL	1	1	0	-1	Flts/10Sec	Dynamic
FFRATH	120	120	0	-1	Flts/10Sec	Dynamic
WSINC	21	150	0	-1	Pages	Dynamic
WSDEC	35	35	0	-1	Pages	Dynamic
FREELIM	16	32	16	-1	Pages	
FREEGOAL	156	200	16	-1	Pages	
GROWLIM	51	63	0	-1	Pages	Dynamic
BORROWLIM	300	300	0	-1	Pages	Dynamic
LOCKIDTBL	1000	128	16	16000	Entries	
RESHASHTBL	256	32	1	8192	Entries	

Dans les tests, nous parlerons également des paramètres suivants, bien qu'il ne soient pas considérés comme "majeurs" par Digital :

Nom du paramètre	Courant	Par Défaut	Min	Max	Unité	Dynamique
MPW-WRTCLUSTER	120	96	16	120	pages	
MPW-HILIMIT	500	500	0	16384	pages	
MPW-LOLIMIT	120	32	0	16384	pages	
AWSMIN	50	50	0	- 1	pages	D
AWSTIME	20	20	1	- 1	10milli sec.	D

### V.2.2 Résultats des mesures

Les tests ont été analysés sur base du tableau fourni par le monitor, synthétisant la période mesurée : PAGE MANAGEMENT STATISTICS. Celui-ci nous a semblé suffisamment représentatif de la réalité. En effet, un test consiste à soumettre le système à 30 utilisateurs interactifs pendant une dizaine de minutes. Cette charge étant intensive, il ne nous a pas paru nécessaire d'étudier la chronologie du comportement des variables observées, ce qui aurait considérablement alourdi le travail de dépouillement. Dans ce cas-ci, la moyenne et le maximum fournissent une première indication sur l'orientation que prendrait une analyse plus poussée.

Lors de la première étape, le benchmark a tourné avec les paramètres initiaux.

Voici les observations réalisées :

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 09:26:46  
To: 14-AUG-1983 09:37:49

	CUR	AVE	MIN	MAX
Page Fault Rate	24.26	114.24	0.00	836.63
Page Read Rate	1.96	7.99	0.00	57.51
Page Read I/O Rate	1.31	2.88	0.00	15.68
Page Write Rate	0.00	9.94	0.00	394.73
Page Write I/O Rate	0.00	0.08	0.00	3.28
Free List Fault Rate	9.83	17.72	0.00	317.98
Modified List Fault Rate	7.54	67.40	0.00	758.41
Demand Zero Fault Rate	0.00	5.80	0.00	252.47
Global Valid Fault Rate	5.57	19.19	0.00	83.12
Wrt In Progress Fault Rate	0.00	0.79	0.00	60.85
System Fault Rate	0.00	0.00	0.00	0.92
Free List Size	4111.00	2623.41	146.00	4120.00
Modified List Size	127.00	280.48	27.00	495.00

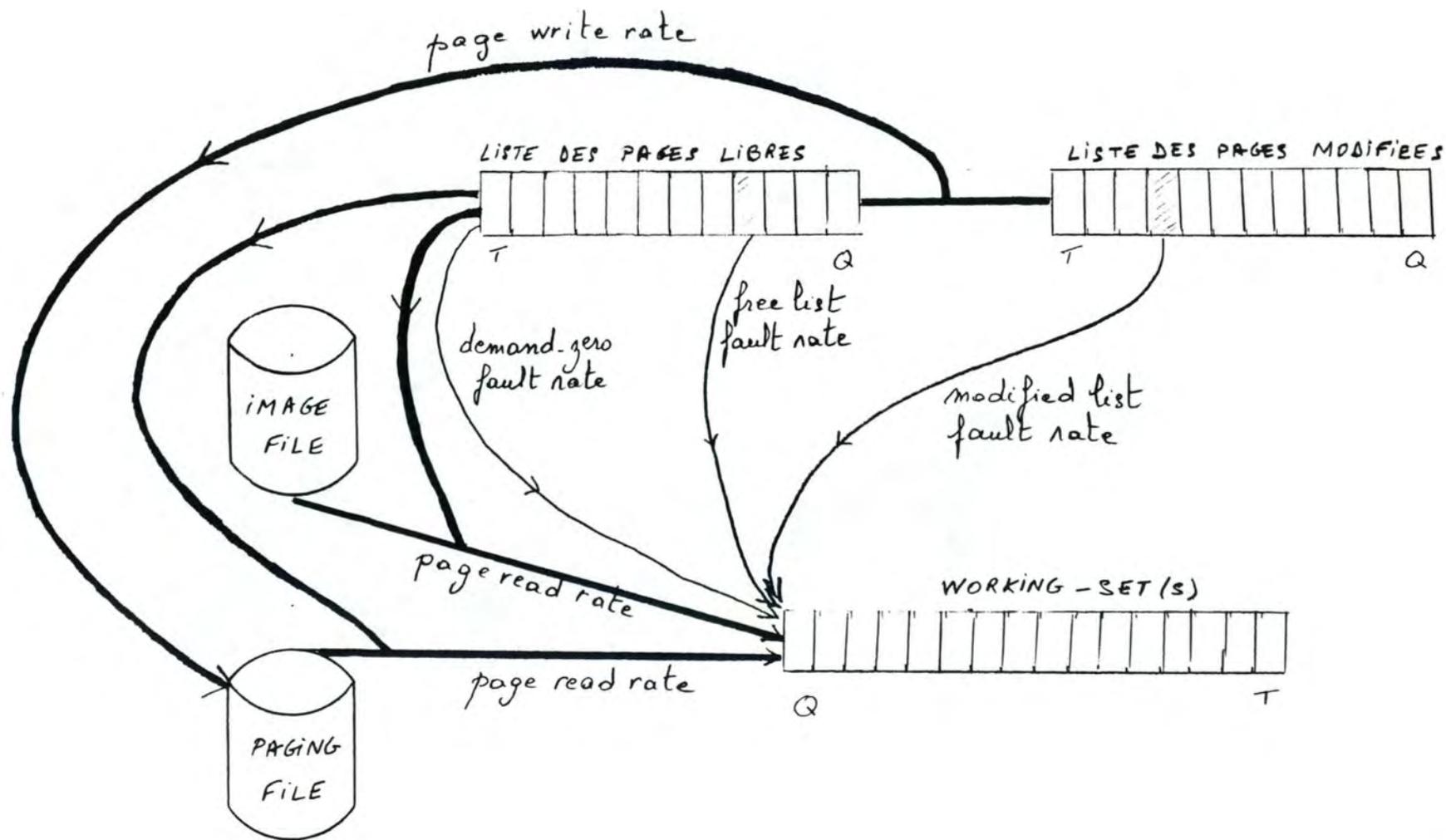
PLAYBACK

SUMMARIZING

Cette statistique de la gestion des pages mémoire fournit :

- Page fault rate :  
le taux de défaut de page pour tous les working-sets  
(nombre de défauts de page/sec)
- Page read rate :  
le taux des pages lues sur disque suite à un défaut de page  
(nombre de pages lues/sec)
- Page read I/O rate :  
le taux d'opérations I/O de lecture sur disque suite à un  
défaut de page (nombre d'opérations de lecture/sec)
- Page write rate :  
le taux de réécriture des pages de la liste de pages modifiées  
vers le paging file (nombre de pages écrites/sec)
- Page write I/O rate :  
le taux d'opérations I/O d'écriture sur disque de la liste  
des pages modifiées vers le paging file (nombre d'opérations  
d'écriture/sec)
- Free list fault rate :  
le taux de pages lues dans la liste des pages libres suite à  
un défaut de page (nombre de pages lues/sec)
- Modified list fault rate :  
le taux de pages lues dans la liste des pages modifiées  
suite à un défaut de page (nombre de pages lues/sec)
- Demand zero fault rate :  
le taux de pages réinitialisées allouées aux working-sets  
suite à un défaut de page (nombre de pages allouées/sec)

- Global valid fault rate :  
le taux de défaut de page pour des pages qui ne sont pas dans le working-set du processus mais en mémoire centrale renseignées dans la table des pages globales du système (nombre de défauts de pages/sec)
  
- WRT in progress fault rate :  
le taux de pages demandées lors d'un défaut de page qui sont déjà dans le processus de réécriture sur disque (nombre de pages/seconde)
  
- System fault rate :  
le taux de défaut de page pour des pages du système (nombre de défauts de page/seconde)
  
- Free list size :  
nombre de pages dans la liste des pages libres
  
- Modified list size :  
nombre de pages dans la liste des pages modifiées



A partir de ces statistiques, il est possible de calculer :

- le nombre moyen de pages lues par opération I/O de lecture :  

$$\frac{\text{page read rate}}{\text{page read I/O rate}}$$
- le nombre moyen de pages réécrites dans le paging file par opération I/O d'écriture : 
$$\frac{\text{page write rate}}{\text{page write I/O rate}}$$
- le pourcentage de défauts de page provoquant une opération I/O :  

$$\frac{\text{page read I/O rate}}{\text{page fault rate}}$$

De plus, on peut apprécier les pourcentages relatifs des différents types de défauts de pages à l'aide du tableau suivant :

	Moyenne/seconde	Pourcentage de "page fault rate"
Page fault rate	114.21	100 %
page read rate	7.99	7.0
Free list fault rate	17.72	15.5
Modified list fault rate	67.40	59.0
Demand zero fault rate	5.80	5.1
Global valid fault rate	19.19	16.8
WRT in progress fault rate	0.79	0.7
System fault rate	0.00	0

Ainsi, on observe que la plus grande partie des défauts de page est résolue grâce à la liste des pages modifiées.

### V.3 DEGRE D'AJUSTEMENT DU SYSTEME A LA CHARGE

Il nous semble utile, à ce stade de notre étude, de faire une mise au point. En effet, malgré les justifications avancées au chapitre précédent, il se peut que les améliorations attendues ne se réalisent pas. Dans le cas où elles ne se vérifieraient pas, l'un des attrait de ces tests sera alors de nous amener à réfléchir aux divergences entre les mesures et les estimations ainsi qu'à leurs causes.

#### V.3.1 Variations des paramètres

Comme annoncé précédemment, l'objectif de ce chapitre est l'étude de l'influence des variations des paramètres du système sur le mécanisme de gestion de la mémoire virtuelle.

Les paramètres concernés sont :

- PFCDEFAULT
- MAX PROCESSCNT
- SYSMWCNT
- BALSETCNT
- WSMAX
- QUANTUM
- MPW-WRTCLUSTER
- MPW-HILIMIT
- MPW-LOLIMIT
- MPW-THRESH
- MPW-WAITLIMIT
- PFRATL
- PFRATH
- WSINC
- WSDEC
- AWSMIN
- AWSTIME
- FREELIM
- FREEGOAL
- GROWLIM
- BORROWLIM

Mais seul la variation des paramètres soulignés a été étudiée. Les raisons en sont multiples :

- . D'après l'étude théorique des chapitres précédents, ce sont ces paramètres qui devraient avoir une influence déterminante sur le comportement du système.
- . Il a été supposé que les paramètres concernant la taille des tables, des zones tampons, ainsi que ceux relatifs au nombre de processus admissibles (MAXPROCESSCNT, BALSETCNT) et aux pages réservées au système (SYSMWCNT) étaient adaptés à la configuration du VAX 11/780 en présence.
- . L'accent a été mis sur les paramètres dynamiques, qui peuvent être réajustés à tout moment, sans devoir relancer le système.

### V.3.2 Analyse des résultats

Quelques expériences initiales nous permirent d'évaluer la reproductibilité des mesures. Par la même occasion, celles-ci nous permirent de découvrir une façon de gagner de la place mémoire en période d'intense activité des utilisateurs. En effet, un certain nombre de processus système (ERRFMT, JOB-CONTROL, OPCOM, PRTSYMB1 et PRTSYMB2) sont invoqués peu fréquemment. De plus, ils ont une priorité de base élevée et, à part OPCOM, ils hibernent la plupart du temps. Aussi en lançant tous les processus du benchmark avec une priorité 8, en période de surcharge de la mémoire, ces processus système seront swappés avant les processus utilisateurs.

D'autre part, étant donné le mécanisme de réajustement automatique des working-sets, nous ne nous sommes pas attardés au réglage des working-sets par défaut des utilisateurs.

Enfin, le processus de prise de mesure "MONITOR" tournait dans une queue batch "locked" afin d'éviter son swapping. Les ressources utilisées par ce logiciel étaient approximativement équivalentes à celles utilisées par un utilisateur simulé dans le benchmark.

1. Variation des paramètres relatifs à la liste des pages  
modifiées-----

A présent, analysons plus en détail les tests 2 et 3 portant sur la variation des paramètres relatifs à la liste des pages modifiées, synthétisée dans le tableau suivant :

	Situation initiale	Test 2	Test 3
MPW-WRTCLUSTER	120	120	96
MPW-HILIMIT	500	500	500
MPW-LOLIMIT	120	380	32

Le test 2 a pour but, en augmentant la valeur de MPW-LOLIMIT, d'accroître la taille de la liste des pages modifiées et par la même occasion de diminuer le nombre d'opérations nécessaires aux réécritures des pages modifiées dans le paging file. En effet, le nombre de pages à recopier passe d'environ 380 (500-120) à plus ou moins 120 (500 - 380) pour un même cluster (MPW-WRTCLUSTER = 120). De plus, cette extension du working-set que constitue la liste des pages modifiées, devrait contribuer à un plus grand nombre de résolutions des défauts de page et donc réduire le nombre d'opérations I/O de lecture lors de défauts de page.

Par contre, le test 3 devrait vérifier exactement les conclusions inverses. En effet, en diminuant simultanément le cluster (MPW-WRTCLUSTER : 120 ↘ 96) et la taille minimum de la liste des pages modifiées (MPW-LOLIMIT : 120 ↘ 32), le nombre d'opérations I/O de réécriture vers le paging file devrait s'accroître.

Les résultats des mesures fournies par Monitor donnent :

	Situation initiale		Test 2		Test 3	
	moyenne	max	moyenne	max	moyenne	max
Page fault rate	114.24	836.63	98.85	262.82	108.52	610.56
Page read rate	7.99	57.51	7.29	39.80	8.06	43.17
Page read I/O rate	2.88	15.68	2.57	12.13	2.94	13.15
Page write rate	<u>9.94</u>	394.73	<u>1.33</u>	118.03	<u>13.58</u>	405.61
Page write I/O rate	<u>0.08</u>	3.28	<u>0.01</u>	0.98	<u>0.14</u>	4.29
Free list fault rate	<u>17.72</u>	317.98	<u>10.58</u>	124.53	20.10	316.50
Modified list fault rate	67.40	758.41	62.25	830.26	58.56	401.31
Demand zero fault rate	5.80	252.47	4.17	22.69	5.87	131.90
Global valid fault rate	19.19	83.12	18.75	76.31	19.36	63.96
Wrt in progress fault rate	<u>0.79</u>	60.85	<u>0.06</u>	11.47	<u>1.24</u>	44.82
System fault rate	0.00	0.92	0.00	0.33	0.01	1.97
Free list size	2623.41	4120	2630.46	4148	2627.68	4139
Modified list size	<u>280.48</u>	495	<u>329.78</u>	491	306.08	495

Ce qui traduit en termes de pourcentage des variations des valeurs moyennes par rapport à la situation initiale se résume par le tableau suivant :

	Test 2	Test 3
Page fault rate	- 13.4 %	- 5 %
Page read rate	- 8.8 %	+ 0.9 %
Page read I/O rate	- 10.8 %	+ 2.1 %
Page write rate	<u>- 86.6 %</u>	<u>+ 36.6 %</u>
Page write I/O rate	<u>- 87.5 %</u>	<u>+ 75 %</u>
Free list fault rate	<u>- 40.3 %</u>	<u>+ 13.4 %</u>
Modified list fault rate	- 7.6 %	<u>- 13.1 %</u>
Demand zero fault rate	- 28.1 %	+ 1.2 %
Global valid fault rate	- 23 %	+ 0.9 %
Wrt in progress fault rate	<u>- 92.4 %</u>	<u>+ 57 %</u>
System fault rate (nb moyen/seconde)	0.00	0.01
Free list size	+ 0.3 %	+ 0.2 %
Modified list size	<u>+ 17.9 %</u>	+ 9.3 %

On constate donc :

- a. En ce qui concerne le test 2 (augmentation de MPW-LOLIMIT : 120 ↗ 380) le taux de réécriture des pages dans le paging file ainsi que le nombre d'opérations I/O correspondantes décroissent de plus de 80 %. En corollaire, il y a une augmentation de la liste des pages modifiées de 17.9 %. Du même coup, la résolution des défauts de page par la liste des pages

libres a diminué de 40.3 %, car celle-ci ne contient plus, comme dans la situation initiale, les pages recopiées plus tôt dans le paging file. De là vient aussi la décroissance de 92 % des défauts de page pour des pages déjà dans le processus de réécriture sur disque. Ce pourcentage doit cependant être atténué par le fait que tous les I/O se concentrent sur un seul canal. Vu l'allongement de la liste des pages modifiées le nombre de défauts de page demandant une page du paging file est aussi moins important (-8.8 % et - 10.8 % en nombre d'opérations I/O). Globalement, la diminution des 13.4 % du taux global de défauts de page révèle que la rotation des pages est moins importante. En effet, le nombre de pages écrites dans le paging file diminuant, le nombre de clusters revenant du paging file décroît et par conséquent les éjections de pages du working-set nécessaires pour les accueillir sont moins importantes, ce qui ralentit la croissance de la liste des pages modifiées. L'accroissement de MPW-LOLIMIT provoque ainsi un effet boule de neige non négligeable.

- b. Dans le test 3 (MPW-LOLIMIT : 120 ↘ 32 et MPW-WRTCLUSTER : 120 ↘ 96), les effets sont inversés. Il y a une augmentation de 36.6 % du taux de copie des pages modifiées dans le paging file alors que le nombre d'opérations I/O correspondantes croît de 75 %. Cette forte différence de croissance est due à la diminution simultanée du cluster et du nombre minimum de pages dans la liste des pages modifiées. D'autre part, on constate que les défauts de page précédemment résolus par la liste des pages modifiées (- 13.1 %) sont désormais résolus par la liste des pages libres (+ 13.4 %), puisqu'une fois recopiées, les pages de la liste des pages modifiées sont transférées dans la liste des pages libres. De plus, l'utilisation d'un cluster de réécriture plus petit provoque apparemment une surcharge au niveau de l'unique canal disponible, d'où l'augmentation de 57 % des défauts de page pour des pages dans le processus de réécriture sur disque.

La répartition des différents types de défaut de page reflète exactement les mêmes conclusions :

	Situation initiale	Test 2	Test 3
Page fault rate (nombre moyen/sec)	114.24	98.85	108.52
Page read rate	7 %	7.4 %	7.4 %
Free list fault rate	15.5 %	10.7 %	18.5 %
Modified list fault rate	59 %	63 %	54 %
Demand zero fault rate	5.1 %	4.2 %	5.4 %
Global valid fault rate	16.8 %	19 %	17.8 %
Wrt in progress fault rate	0.7 %	0.1 %	1.1 %
System fault rate (nombre moyen/sec)	0.00	0.00	0.01

De ces premiers tests, il résulte que la taille de la liste des pages modifiées est critique,

- d'un part, la diminution de MPW-WRTCLUSTER ne fait qu'accroître inutilement le nombre d'opérations I/O d'écriture sur le paging file
- d'autre part, l'augmentation de MPW-LOLIMIT est un gain considérable, tant au niveau du nombre total de défauts de page, qu'au niveau des opérations I/O provoquées par les transferts de pages.

Il reste à voir si la perte de place mémoire correspondante (260 pages) est supportable pour la configuration. Ce qui semblait être notre cas sur le VAX 11/780 avec 3 mégabytes de mémoire centrale.

## 2. Variation du paramètre PFCDEFAULT

Dans un deuxième temps, nous avons étudié les résultats donnés par les tests 4 et 5 portant sur la variation du cluster de pages lues sur disque (PFCDEFAULT), lors d'un défaut de page nécessitant un I/O.

Ces tests se basaient sur la constatation suivante : quand un défaut de page se produit pour une page extérieure à la mémoire, VMS n'amène pas seulement la page manquante mais aussi un certain nombre de pages contigues. Ce chargement anticipatif a pour but de diminuer le nombre d'opérations I/O de lecture.

Malheureusement, l'arrivée de ces pages dans le working-set force bien souvent l'éjection d'autres pages, causant de nouveaux défauts de page et peut-être des transferts de pages supplémentaires. Bien entendu, ce sont les transferts de pages, plus que les défauts de pages supplémentaires, qui sont déterminants pour la performance du système.

Dès lors, des mesures ont été réalisées avec les valeurs successives : PFCDEFAULT = 32 et PFCDEFAULT = 64. Les résultats de celles-ci sont regroupés dans le tableau suivant :

	Situation initiale		Test 4		Test 5	
	PFCDEFAULT = 127		PFCDEFAULT = 32		PFCDEFAULT = 64	
	moyenne	max	moyenne	max	moyenne	max
Page fault rate	114.24	826.63	112.87	709.29	208.82	799.34
Page read rate	<u>7.99</u>	57.51	<u>7.55</u>	43.23	<u>7.31</u>	45.72
Page read I/O rate	<u>2.88</u>	15.68	<u>2.75</u>	15.84	<u>2.68</u>	17.10
Page write rate	9.94	394.73	11.15	354.09	9.50	393.44
Page write I/O rate	0.08	3.28	0.09	2.95	0.07	3.27
Free list fault rate	17.72	317.98	18.61	341.11	16.89	367.62
Modified list fault rate	67.40	758.41	65.75	542.12	64.09	781.90
Demand zero fault rate	5.80	252.47	5.85	176.92	5.73	226.79
Global valid fault rate	19.19	83.12	18.67	74.67	18.35	72.03
Wrt in progress fault rate	0.79	60.85	0.76	29.93	0.65	33.33
System fault rate	0.00	0.92	0.00	0.98	0.01	1.78
Free list size	2623.41	4120	2617.59	4095	1653.50	4139
Modified list size	280.48	495	261.29	493	269.08	489

Ces résultats paraissent difficilement interprétables.  
Seuls le taux de défauts de page provoquant une lecture sur disque semble varier quelque peu :

Situation initiale	PFCDEFAULT = 32	PFCDEFAULT = 64
PFCDEFAULT = 127		
Page read rate	- 5.5 %	- 8.5 %
Page read I/O rate	- 4.5 %	- 6.9 %

Si l'on interprète ces résultats, il s'en dégage :

- Le paramètre PFCDEFAULT = 32 est un cluster mieux approprié car il provoque 5.5 % de défauts de page de lecture sur disque en moins, autrement dit il a réalisé moins d'éjections par son arrivée de 32 pages que par celle de 127.
- Le paramètre PFCDEFAULT = 64 semble mieux indiqué encore que le précédent car il nécessite 8.5 % en moins de défauts de page de lecture sur disque, ce qui correspond à un gain de 6.9 % en opérations I/O de lecture.

Dans un premier temps, on concluerait donc que le cluster de 32 pages est trop petit et nécessite des opérations I/O de lecture supplémentaires par rapport au cluster de 64, malgré que celui-ci éjecte plus de pages du working-set par son arrivée.

Cependant, si l'on calcule le rapport du nombre de pages lues sur disque sur le nombre d'opérations I/O correspondantes, on obtient :

	PFCDEFAULT=127	PFCDEFAULT=32	PFCDEFAULT=64
<u>page read rate</u>	2.77	2.75	2.73
page read I/O rate			

Cela signifierait que lors d'une opération I/O de lecture sur disque, un cluster moyen de 2 à 3 pages est transféré en mémoire. Ce qui contredit les conclusions du paragraphe précédent.

Il est certain que le paramètre PFCDEFAULT ne semble avoir guère d'influence sur le mécanisme de la gestion de la mémoire virtuelle. Il n'en reste pas moins que les raisons en demeurent inconnues à ce stade de l'étude.

### 3. Variation du Quantum

Le troisième type de tests se rapporte à la variation de la longueur du quantum.

L'effet de la longueur du quantum sur le scheduling au processeur est peu significatif puisqu'il force la rotation parmi des processus d'égale priorité. Par contre, un long quantum est nécessaire pour régler le taux de swapping qui est déterminant pour la performance du système si la taille des working-sets des utilisateurs actifs dépasse le nombre de "pages frames" (ou cadre en mémoire centrale) disponibles pour les processus utilisateurs. Si le swapping est négligeable, un long quantum n'est plus nécessaire. Une valeur plus petite provoquera le réajustement automatique des limites des working-sets plus fréquemment. Ce qui devrait diminuer la pagination.

Partant d'un quantum initial de 20, les tests ont été réalisés avec les valeurs successives de 40 et 60 (l'unité est 10 milli-secondes).

Les résultats en sont donnés par le tableau suivant, en termes de pourcentage de variation des valeurs moyennes :

Situation initiale QUANTUM = 20	QUANTUM = 40	QUANTUM = 60
Page fault rate	<u>+ 11.7</u>	<u>+ 14.6</u>
Page read rate	<u>- 15.6</u>	<u>- 9.5</u>
Page read I/O rate	- 15.3	+ 0.4
Page write rate	<u>+ 79.4</u>	<u>+ 142</u>
Page write I/O rate	+ 75.0	+ 150
Free list fault rate	+ 40.5	+ 66.8
Modified list fault rate	+ 10.8	+ 6.6
Demand zero fault rate	- 0.5	- 28.5
Global valid fault rate	- 6.6	- 3.4
Wrt in progress fault rate	+ 79.8	+ 31.9
System fault rate (nb moyen/seconde)	0.00	0.00
Free list size	- 1.0	- 0.2
Modified list size	+ 7.5	+ 14.6

Dans les deux cas, les mêmes phénomènes sont observés, mais ils sont plus marqués pour le quantum de 60.

- . une augmentation du nombre total des défauts de page de plus de 10 %
- . une diminution des défauts de page demandant la lecture d'une page sur disque
- . une augmentation impressionnante des recopies de pages modifiées sur disque et des I/O correspondants (75 % et 150 %)
- . un fort accroissement du nombre de défauts de page résolus par la liste des pages libres et un faible accroissement de ceux résolus par la liste des pages modifiées.

On constate ainsi qu'un quantum plus grand, et par conséquent un réajustement moins fréquent des working-sets, provoque une augmentation du taux de pagination et un plus grand transfert de pages vers les listes des pages libres et modifiées, ce qui accroît considérablement le nombre d'I/O en écriture vers le paging file. Mais étant donné que la rotation des processus au processeur est moins rapide, on observe aussi une diminution du taux de défauts de page résolu par lecture sur disque au profit des pages libres, non encore réutilisées par un autre processus.

La répartition des différents types de défauts de page traduit ce glissement :

	Sit. initiale	QUANTUM=40	QUANTUM=60
	QUANTUM = 20		
Page fault rate (nombre moyen/sec)	114.24	127.56	130.89
Page read rate	7 %	5.3 %	5.5 %
Freelist fault rate	15.5 %	19.5 %	22.6 %
Modified list fault rate	59 %	58.5 %	54.9 %
Demand zero fault rate	5.1 %	4.5 %	3.2 %
Global valid fault rate	16.8 %	14.1 %	14.2 %
Wrt in progress fault rate	0.7 %	1.1 %	2.5 %
System fault rate (nombre moyen/sec)	0.00	0.00	0.00

#### 4. Interaction des paramètres

Comme il était apparu dans les chapitres précédents, nos multiples expériences de réglage révélèrent un nombre important d'interactions entre les paramètres. Il est certain que celles-ci étaient accentuées par la sévère contrainte de l'unique canal cumulant tous les types d'I/O.

Le test 8 espérait cumuler les avantages :

- d'un petit quantum (20), c'est-à-dire un réajustement plus fréquent des working-sets,
- d'une augmentation de la liste des pages modifiées (MPW-LOLIMIT = 380) afin de gagner des opérations I/O de lecture et principalement d'écriture,
- d'une diminution du cluster de pages lues (PFCDEFAULT = 64).

Chacune de ces opérations, individuellement, provoque une diminution du nombre de défauts de page global et un nombre moins important d'opérations I/O pour la gestion de ces défauts de page.

Les mesures donnèrent pour résultats :

	Situation initiale		Test 8	
	moyenne	max	moyenne	max
Page fault rate	117.24	836.63	115.72	588.38
Page read rate	7.99	57.51	8.69	54.40
Page read I/O rate	2.88	15.68	3.09	15.14
Page write rate	9.94	394.73	8.46	237.62
Page write I/O rate	0.08	3.28	0.07	2.23
Free list fault rate	17.72	317.98	16.83	204.83
Modified list fault rate	67.40	758.41	69.00	468.42
Demand zero fault rate	5.80	252.47	6.24	162.17
Global valid fault rate	19.19	83.12	19.75	71.06
Wrt in progress fault rate	0.79	60.85	0.32	26.40
System fault rate	0.00	0.92	0.001	1.23
Free list size	2623.41	4120	2527.35	4106
Modified list size	280.48	495	341.64	486

Dans l'ensemble, on observe une variation moins marquée que dans les variations individuelles des paramètres :

- . Le gain d'opérations I/O en écriture des pages modifiées est moindre qu'attendu.
- . La liste des pages modifiées s'allonge effectivement, ce qui provoque un gain marqué sur le nombre de défauts de page portant sur des pages dans le processus de réécriture et moins de recopies de pages modifiées.
- . Par contre, le taux de défauts de page du système s'accroît, ce qui indique que la mémoire est chargée, et a pour conséquence d'augmenter le nombre de défauts de page demandant une lecture sur disque.

Ceci met en évidence les interactions complexes entre le swapping, la pagination et les listes des pages libres et modifiées.

## 5. Conclusions des différents tests

- Les tests suivants sont partis de la situation du test 8 afin d'étudier les variations combinées de paramètres.

TEST 9  
=====

Le nombre maximal de défauts de page par 10 secondes forçant la décroissance automatique d'un working-set, PFRATL, passe de 1 à 0. Autrement dit, la décroissance automatique des working-sets est annulée. Bien que le nombre total de défauts de page décroît, le test montre un pourcentage de défaut de page provoquant des I/O plus important.

	Situation initiale	Test 9
<u>page read I/O rate</u>	2.52 %	3.88 %
page fault rate		

Ce résultat s'explique aisément. Dans l'ensemble, les défauts de page sont moins importants puisque les working-sets sont plus grands. Mais les working-sets s'étant toujours étendus, ils ont considérablement restreint les listes des pages libres et modifiées. Aussi la résolution des défauts de page se fait-elle désormais par lecture sur disque.

TEST 10  
=====

Le nombre maximal de défauts de page par 10 secondes forçant la décroissance automatique d'un working-set, PFRATL, passe de 1 à 2. Seul un gain en recopie de pages modifiées est constaté.

TESTS 11 et 12  
=====

Le nombre minimum de défauts de page par 10 secondes provoquant une croissance automatique du working-set PFRATH influence la vitesse de croissance d'un working-set.

Situation initiale	PFRATH = 60	PFRATH = 80
PRFATH = 20		
Nombre de défauts de page avec lecture sur disque	- 8 %	-
Nombre de pages modifiées recopiées sur disque	- 29 %	- 35 %

## TESTS 13 et 14

=====

Le nombre de pages ajoutées à un working-set lors de la croissance automatique, WSINC, passe respectivement de 21 à 10 et de 21 à 150.

Ces 2 tests donnent des résultats presque identiques. Cette variation provoque un gain considérable sur le nombre total des défauts de page et un gain de plus de 80 % dans la copie des pages modifiées.

## TESTS 15 et 16

=====

Variation du nombre de pages retirées à un working-set lors de la décroissance automatique, WSDEC.

Situation initiale	WSDEC = 10	WSDEC = 100
WSDEC = 35		
Page fault rate	- 8 %	+ 16 %
Page read rate	+ 4 %	- 1 %
Page write rate	- 31 %	- 28 %
Free list fault rate	- 33 %	+ 18 %
Modified list fault rate	- 6 %	+ 22 %

La plus grande partie de ce nombre supplémentaire de défauts de page dans le cas WSDEC = 100 est résolu par les listes des pages libres et modifiées, ce qui n'engendre aucune opération I/O.

## TESTS 17 et 18

=====

Variation de la taille minimum du working-set tolérée lors des réajustements automatiques.

Situation initiale AWSMIN = 50	AWSMIN = 20	AWSMIN = 100
Page fault rate	+ 12 %	- 7 %
Page read rate	+ 2 %	+ 1 %
Page write rate	- 29 %	- 36 %
Free list fault rate	- 5 %	- 35 %
Modified list fault rate	+ 13 %	- 2 %

Dans le premier cas, on constate une augmentation significative du nombre total de défauts de page. Cela est dû à un working-set inutilement petit. Tandis que dans le second cas, le nombre total de défauts de page décroît mais les tailles des working-sets sont plus importantes.

## TESTS 19 et 20

=====

Variation du temps d'estimation du taux de défauts de page pour pratiquer le réajustement automatique des working-sets.

Situation initiale AWSTIME = 20	AWSTIME = 40	AWSTIME = 60
Page fault rate	+ 10 %	+ 28 %
Page read rate	+ 9 %	+ 12 %
Page write rate	+ 51 %	+ 216 %
Free list fault rate	+ 16 %	+ 93 %
Modified list fault rate	+ 12 %	+ 18 %

Cette forte augmentation du taux de défauts de page dans les 2 cas prouve le mauvais ajustement des working-sets des processus. La tranche de temps pour permettre de déterminer ces ajustements est trop longue et n'est pas en rapport avec le quantum (20).

Cette dégradation brutale suite à une variation d'un paramètre non "majeur" (pour Digital) met l'accent sur l'influence que peut avoir le réglage des paramètres sur le comportement du système.

En conséquence, le mécanisme de gestion de la mémoire virtuelle apparaît être une des clés de la performance du système VAX / VMS. Ses paramètres principaux de réglage peuvent se regrouper dans les objectifs :

1. Les priorités des divers processus
2. Les tailles des working-sets des processus
3. La longueur du quantum
4. La gestion des pages libres et modifiées.

### V.3.3 Evaluation des performances

Comme déjà signalé précédemment, vu les contraintes temporelles, seul un petit nombre d'expériences ont pu être réalisées. Aussi les chiffres présentés ne sont-ils pas le reflet d'un échantillonnage statistique complet. C'est pourquoi l'étude des résultats a toujours été faite en termes de variations par rapport à la mesure "étalon" choisie, les résultats du benchmark avec les paramètres initiaux de la configuration.

Il n'en est pas moins vrai que cet ensemble de tests est indicatif sur l'orientation que prendrait une analyse plus précise. D'autre part, leur examen confirme la tendance découverte dans les mécanismes du fonctionnement de la mémoire virtuelle dans le VAX / VMS. De plus, il souligne l'importance relative de la variation de l'un ou l'autre paramètre. En ce sens, ces résultats sont exploitables puisqu'ils mettent en évidence les principaux paramètres de réglage du VAX / VMS à ajuster suivant la configuration et la politique du gestionnaire.

Certes, de nombreuses mesures sont encore nécessaires pour étendre l'étude au-delà du mécanisme de gestion de la mémoire virtuelle au système tout entier afin d'analyser d'autres grandeurs telles que l'utilisation du CPU, l'utilisation des disques, le temps de réponse, ...

Mais dès à présent, ces considérations suggèrent un ensemble de point dont un gestionnaire doit tenir compte pour l'optimisation de son système. Dans le système VAX / VMS, le responsable du système a à sa disposition un éventail considérable de variables à adapter à sa configuration. En supposant que la charge soumise au système n'est pas disproportionnée par rapport aux ressources hardware disponibles, les paramètres reflèteront ses choix. Les principales caractéristiques de sa politique se traduiront à travers les éléments suivants :

- La répartition de l'utilisation de la mémoire grâce
  - . aux paramètres propres à chaque utilisateur concernant la taille de son working-set (WSDEFAULT, WSQUOTA, WSEXTENT)
  - . aux paramètres limitant le nombre de processus admis dans le système et le nombre de réellement actifs (MAXPROCESSCNT, BALSETCNT), ce qui sera déterminant pour le taux de swapping
  - . aux paramètres déterminant les caractéristiques des pages libres et modifiées
  - . aux paramètres concernant l'ajustement automatique des working-sets
  - . aux paramètres relatifs aux espaces mémoires dont le bon réglage peut être vérifié par la commande "SHOW MEMORY".

**System Memory Resources on 13-AUG-1983 09:22:18.90**

Physical Memory Usage (pages):				
	Total	Free	In Use	Modified
Main Memory (3.00Mb)	6144	4018	1837	289
Slot Usage (slots):				
	Total	Free	Resident	Swapped
Process Entry Slots	70	60	10	0
Balance Set Slots	52	44	8	0
Fixed-Size Pool Areas (packets):				
	Total	Free	In Use	Size
Small Packet (SRP) List	380	196	184	96
I/O Request Packet (IRP) List	750	707	43	160
Large Packet (LRP) List	8	3	5	640
Dynamic Memory Usage (bytes):				
	Total	Free	In Use	Largest
Nonpaged Dynamic Memory	179712	100512	79200	98816
Paged Dynamic Memory	84992	29984	55008	29984
Paging File Usage (pages):				
		Free	In Use	Total
DISK\$VAX100:[SYSO.SYSEXEX]SWAPFILE.SYS		21808	1200	23008
DISK\$VAX100:[SYSO.SYSEXEX]PAGEFILE.SYS		39762	246	40008

Of the physical pages in use, 1292 pages are permanently allocated to VMS.

- L'ordonnancement des tâches grâce :
  - . à l'influence sur le scheduling par le jeu de priorités
  - . à la variation du quantum pour la rotation des processus au processeur.

Mais l'interaction importante entre ces variables de réglage est un élément à ne jamais perdre de vue. Cependant, le système VAX / VMS offre un outil majeur aux néophytes (MONITOR) pour l'observation du comportement du système et donc de l'impact des réglages successifs. De plus, il offre la souplesse d'un réajustement dynamique pour un certain nombre de paramètres. Cela permet au gestionnaire de rectifier sa politique en fonction de la charge du moment sans devoir interrompre le système.

## CONCLUSION

Ce travail a permis d'explorer, dans certains de ses détails, le système d'exploitation VAX / VMS. Nous avons eu l'occasion de travailler sur différentes versions et avons dû nous adapter à ces versions successives, ce qui n'a pas facilité l'avancement du travail.

L'objectif de cette étude était de tenter d'évaluer l'impact des très nombreux paramètres du système VAX / VMS sur les performances de celui-ci.

Le mécanisme de la gestion de la mémoire virtuelle est le noeud de ce système d'exploitation. Aussi nous sommes nous limité à l'influence des principaux paramètres relatifs à la gestion de la mémoire. Vu les contraintes temporelles, nous n'avons pas pu réaliser un benchmark complet et toutes les mesures nécessaires. Par le fait même, nous avons pu apprécier les outils mis à la disposition des néophytes par Digital sur le VAX. Ceux-ci ont permis la réalisation d'un ensemble élémentaire de mesures qui ont permis d'approcher les réalités du comportement du système VAX / VMS. Leurs résultats peuvent nous fournir certaines indications de l'impact de la variation de certains paramètres sur le mécanisme d'allocation de la mémoire.

Au-delà de cette étude, le problème de l'évaluation du système implique la définition de grandeurs caractérisant la performance. Pour des systèmes interactifs, une voie intéressante de recherche pourrait être orientée vers des critères tels que le temps de réponse, l'utilisation du CPU, ...

Au terme de cette première approche, il apparaît déjà qu'un ajustement des paramètres du système VAX / VMS, à sa charge dans le cadre d'une configuration donnée, est un élément non négligeable pour la performance du système au niveau de l'utilisation des unités E/S. Aussi serait-il intéressant de perfectionner une

méthode de simulation ou un modèle analytique afin de déterminer les paramètres optimaux pour des charges précises, dans le cadre d'une configuration donnée, en tenant compte de la politique souhaitée par le gestionnaire. Cette assistance au réglage des paramètres constituerait un instrument précieux pour l'adaptation des systèmes VAX / VMS aux besoins actuels et futurs des utilisateurs.

## ANNEXES

1. Les paramètres du système
  - a. Tableaux des paramètres par catégorie
  - b. Liste des paramètres par ordre alphabétique
  
2. Le logiciel de mesure "MONITOR"
  
3. Les mesures
  - a. Paramètres initiaux
  - b. Résultats par MONITOR
  - c. Tableaux synthétiques des résultats
    - 1) Variations des paramètres
    - 2) Résultats des tests
    - 3) Variations des résultats de mesures
    - 4) Répartition des défauts de page

1. LES PARAMETRES DU SYSTEME  
=====

Les catégories des paramètres sont :

- MAJOR : paramètres majeurs les plus sujets à des réajustements
- SYS : paramètres systèmes qui affectent l'ensemble des opérations
- JOB : paramètres de contrôle des jobs
- ACP : paramètres associés aux processus de contrôle des I/O
- TTY : paramètres associés au comportement du terminal
- SCS : paramètres qui contrôlent les "system communication services"
- RMS : paramètres associés aux fonctions RMS
- PQL : paramètres associés aux limites et quotas lors de la création des processus
- GEN : paramètres qui affectent la création et l'initialisation des structures de données au lancement du système
- SPECIAL : paramètres spéciaux utilisés par Digital
- DYNAMI : paramètres dont les valeurs peuvent être réajustées pendant le fonctionnement du système, sans devoir le relancer

A. TABLEAUX DES PARAMETRES PAR CATEGORIE

## Paramètres majeurs (MAJOR)

Parameter Name	Description	Dynamic
PFCDEFAULT	Default page fault cluster size (in pages)	D
GBLSECTIONS	Number of global section descriptors	
GBLPAGES	Number of global page table entries	
MAXPROCESSCNT	Maximum number of processes	
SYSMWCNT	Maximum size of system working set (in pages)	
BALSETCNT	Maximum number of resident working sets	
IRPCOUNT	Number of preallocated intermediate request packets	

Parameter Name	Description	Dynamic
WSMAX	Maximum size of any working set (in pages)	
NPAGEDYN	Size of nonpaged dynamic pool (in bytes, but rounded down to an integral number of pages by the system)	
PAGEDYN	Size of paged dynamic pool (in bytes, but rounded down to an integral number of pages by the system)	
VIRTUALPAGECNT	Maximum virtual space per process (in pages)	
LRPCOUNT	Number of preallocated large request packets	
SRPCOUNT	Number of preallocated small request packets	
QUANTUM	Maximum time a process can use at once and minimum service a process must receive before being outswapped, (in 10ms units)	D
PFRATL	Page fault rate low limit (in faults per 10 seconds of processor time)	D
PFRATH	Page fault rate high limit (in faults per 10 seconds of processor time)	D
WSINC	Working set increment (in pages)	D
WSDEC	Working set decrement (in pages)	D
FREELIM	Minimum size of the free page list	
FREEGOAL	Number of pages required on the free page list after a memory shortage	
GROWLIM	Number of pages required on the free page list to allow process to exceed working set quota	D
BORROWLIM	Minimum size of the free page list before process can grow beyond process working set quota (in pages)	D
LOCKIDTBL	Number of entries in the system lock id table	
RESHASHTBL	Number of entries in the lock management resource name hash table	

## . PARAMETRES SYSTEMES (SYS)

Parameter Name	Description	Dynamic	MAJOR
PFCDEFAULT	Default page fault cluster size (in pages)	D	M
KFILSTCNT	Number of known file list heads		
GBLSECTIONS	Number of global section descriptors		M
GBLPAGES	Number of global page table entries		M
GBLPAGFIL	Maximum number of system-wide pages allowed for global page-file sections		
MAXPROCESSCNT	Maximum number of processes		M
PROCSECTCNT	Number of process sections		
MINWSCNT	Minimum number of fluid pages in any working set		
PAGFILCNT	Maximum number of paging files that can be installed		
SWPFILCNT	Maximum number of swapping files that can be installed		
SYSMWCNT	Maximum size of system working set (in pages)		M
INTSTKPAGES	Size of interrupt stack (in pages)		
BALSETCNT	Maximum number of resident working sets		M
IRPCOUNT	Number of preallocated intermediate request packets		M
IRPCOUNTV	Maximum size to which IRPCOUNT can be increased		
WSMAX	Maximum size of any working set (in pages)		M
NPAGEDYN	Size of nonpaged dynamic pool (in bytes, but rounded down to an integral number of pages by the system)		M
NPAGEVIR	Maximum size to which NPAGEDYN can be increased		
PAGEDYN	Size of paged dynamic pool (in bytes, but rounded down to an integral number of pages by the system)		M
VIRTUALPAGECNT	Maximum virtual space per process (in pages)		M

Parameter Name	Description	Dynamic	MAJOR
SPTREQ	Number of additional system page table entries		
LRPCOUNT	Number of preallocated large request packets		M
LRPCOUNTV	Maximum value to which LRPCOUNT can be increased		
LRPSIZE	Size of the large request packets (in bytes)		
SRPCOUNT	Number of preallocated small request packets		M
SRPCOUNTV	Maximum value to which SRPCOUNT can be increased		
QUANTUM	Maximum time a process can use at once and minimum service a process must receive before being outswapped, (in 10ms units)	D	M
• MPW_WRTCLUSTER	Number of pages written per I/O from the modified page list		
• MPW_HILIMIT	Maximum size of modified page list (in pages)		
• MPW_LOLIMIT	Minimum size of modified page list (in pages)		
MPW_THRESH	Minimum size of the modified page list requiring swapper action (in pages)	D	
MPW_WAITLIMIT	Number of pages on the modified page list that forces a process to wait until the next time the modified page writer writes the modified page list	D	
PFRATL	Page fault rate low limit (in faults per 10 seconds of processor time)	D	M
PFRATH	Page fault rate high limit (in faults per 10 seconds of processor time)	D	M
WSINC	Working set increment (in pages)	D	M
WSDEC	Working set decrement (in pages)	D	M
• AWSMIN	Automatic working set minimum (in pages)	D	
• AWSTIME	Automatic working set time for collecting sample (in 10ms units)	D	

Parameter Name	Description	Dynamic	MAJOR
SWPOUTPGCNT	Process size before outswapping occurs (in pages)	D	
LONGWAIT	Time to elapse before a process is judged idle by the swapper (in 6.6ms units)	D	
EXTRACPU	Extra CPU time added to process after CPU time is expired (in 10ms units)	D	
MAXSYSGROUP	Highest system UIC	D	
MVTIMEOUT	Amount of time allowed for a mount verification attempt to succeed before it is cancelled	D	
MAXBUF	Maximum number of bytes that can be transferred in one buffered I/O	D	
DEFMBXBUFQUO	Default mailbox buffer quota (in bytes)	D	
DEFMBXMXMSG	Default mailbox maximum message size (in bytes)	D	
DEFMBXNUMMSG	Not implemented	D	
FREELIM	Lower limit of free page list (in bytes)		M
FREEGOAL	Number of pages required on the free page list after a memory shortage		M
GROWLIM	Number of pages required on the free page list to allow process to exceed working set quota	D	M
BORROWLIM	Minimum size of the free page list before process can grow beyond process working set quota (in pages)	D	M
XFMAXRATE	Maximum rate of transfer for DR32 devices	D	
LAMAPREGS	Number of map registers allocated to an LPA11 device driver		
REALTIME_SPTS	Number of system page table entries reserved for connect-to-interrupt processes		
CLISYMTBL	Size of command interpreter symbol table (in pages)	D	
LOCKIDTBL	Number of entries in the system lock id table		M

Parameter Name	Description	Dynamic	MAJOR
RESHASHTBL	Number of entries in the lock management resource name hash table		M
DEADLOCK_WAIT	Time that a lock request waits before deadlock search is initiated (in seconds)	D	
TIMEPROMPTWAIT	Time allowed for entry of the system time during a boot (in micro- fortnights)		
LOGSHASHTBL	Number of entries in the system logical name hash table		
LOGGHASHTBL	Number of entries in the group logical name hash table		
LOGPHASHTBL	Number of entries in the process logical name hash table		
BUGREBOOT	Automatic reboot on fatal bugcheck; switch	D	
CRDENABLE	Detection and logging of memory-corrected read errors; switch		
DUMPBUG	Writing of dump file on fatal bugcheck; switch		
BUGCHECKFATAL	All bugchecks fatal; switch	D	
SETTIME	Time-of-day prompt at boot time; switch		
UAFALTERNATE	Use of alternate UAF; switch		
MOUNTMSG	Controls OPCOM handling of volume mount messages; switch	D	
DISMOUNMSG	Controls OPCOM handling of volume dismount messages; switch	D	
DEFPRI	Sets the base default priority for processes	D	

• PARAMETRES TERMINAUX (TTY)

Parameter Name	Description	Dynamic
TTY_SCANDELTA	Terminal dial-up/hang-up scan interval (in increments of 100ns)	
TTY_DIALTYPE	Dialup flag bits	
TTY_SPEED	Default speed for terminals; code	
TTY_RSPEED	Receive speed for terminals	
TTY_PARITY	Not implemented	
TTY_BUF	Default line width for terminal	
TTY_DEFCHAR	Default terminal characteristics, longword 1	
TTY_DEFCHAR2	Default terminal characteristics, longword 2	
TTY_TYPAHDSZ	Size of terminal type-ahead buffer (in bytes)	
TTY_ALTYPAMD	Size of alternate type-ahead buffer (in bytes)	
TTY_ALTALARM	Size of alternate type-ahead buffer alarm (in bytes)	
TTY_DMASIZE	Minimum number of output buffer characters to invoke DMA transfers	D
TTY_PROT	Terminal protection against allocation by another process; mask	
TTY_OWNER	Owner UIC for TTY_PROT specification	
TTY_CLASSNAME	Terminal class driver name prefix for booting	
TTY_SILOTIME	Input silo polling interval for DMF-32 hardware (in milliseconds)	

## . PARAMETRES ACP (ACP)

Parameter Name	Description	Dynamic
ACP_MULTIPLE	One ACP per disk volume mounted on different device types; switch	D
ACP_SHARE	Sharing of ACP code; switch	
ACP_MAPCACHE	Size of bit map cache (in pages)	D
ACP_HDRCACHE	Size of file header cache (in pages)	D
ACP_DIRCACHE	Size of directory cache (in pages)	D
ACP_WORKSET	Working set default for ACP	D
ACP_FIDCACHE	Size of file identification cache (in pages)	D
ACP_EXTCACHE	Size of extent cache (in pages)	D
ACP_EXTLIMIT	Maximum amount of free space in extent cache (in tenths of a percent of available free space)	D
ACP_QUOCACHE	Number of entries in quota cache	D
ACP_SYSACC	Size of directory access cache (in pages)	D
ACP_MAXREAD	Maximum directory blocks to read (in blocks)	D
ACP_WINDOW	Default number of window pointers	D
ACP_WRITEBACK	Caching of file headers; switch	D
ACP_DATACHECK	Data verification on ACP I/O	D
ACP_BASEPRIO	Base priority for ACP processes	D
ACP_SWAPFLGS	Swapping of ACP working sets; code	D

• PARAMETRES JOBS (JOB)

Parameter Name	Description	Dynamic
JOBQUEUES	Print or batch queue utilization; switch	D
REINITQUE	Reinitialization of queue file; switch	D
MAXPRINTSYMB	Maximum number of print symbionts	D
DEFPRI	Default priority	D
IJOBLIM	Not implemented	D
BJOBLIM	Not implemented	D
NJOBLIM	Not implemented	D
RJOBLIM	Maximum number of remote terminals	D

• PARAMETRES RMS (RMS)

Parameter Name	Description	Dynamic
RMS_DFMBBC	Default multiblock count	D
RMS_DFMBFSDK	Default multibuffer count for sequential disk operations	D
RMS_DFMBFSMT	Default multibuffer count for magnetic tape operations	D
RMS_DFMBFSUR	Not implemented	D
RMS_DFMBFREL	Default multibuffer count for relative disk operations	D
RMS_DFMBFIDX	Default multibuffer count for indexed sequential disk operations	D
RMS_DFMBFHS	Not implemented	D
RMS_PROLOGUE	Default file structure level for VAX-11 RMS files; code	D
RMS_EXTEND_SIZE	Default file extend size (in blocks)	D
RMS_FILEPROT	Default file protection; mask	

## . PARAMETERS PQL (PQL)

Parameter Name	Description	Dynamic
PQL_DASTLM	Default number of pending ASTs	D
PQL_MASTLM	Minimum number of pending ASTs	D
PQL_DBIOLM	Default buffered I/O limit	D
PQL_MBIOLM	Minimum buffered I/O limit	D
PQL_DBYTLM	Default buffered I/O byte limit	D
PQL_MBYTLM	Minimum buffered I/O byte limit	D
PQL_DCPULM	Default CPU time limit (in increments of 10ms)	D
PQL_MCPULM	Minimum CPU time limit (in increments of 10ms)	D
PQL_DDIOLM	Default direct I/O limit	D
PQL_MDIOLM	Minimum direct I/O limit	D
PQL_DFILLM	Default open file limit	D
PQL_MFILLM	Minimum open file limit	D
PQL_DPGFLQUOTA	Default paging file quota	D
PQL_MPGFLQUOTA	Minimum paging file quota	D
PQL_DPRCLM	Default subprocess limit	D
PQL_MPRCLM	Minimum subprocess limit	D
PQL_DTQELM	Default timer queue entries	D
PQL_MTQELM	Minimum timer queue entries	D
PQL_DWSDEFAULT	Default working set sizes	
PQL_MWSDEFAULT	Minimum default working set size	
PQL_DWSQUOTA	Default working set quota	D
PQL_MWSQUOTA	Minimum working set quota	D
PQL_DWSEXTENT	Default working set extent	D
PQL_MWSEXTENT	Minimum working set extent	D
PQL_DENQLM	Default number of locks queued at one time	D
PQL_MENQLM	Minimum number of locks queued at one time	D

• PARAMETRES SCS (SCS)

Parameter Name	Description	Dynamic
SCSBUFFCNT	Not implemented	
SCSCONNCNT	Maximum number of SCS connections for System Applications	
SCSRESPCNT	Maximum number of response descriptor table entries for System Applications	
SCSMAXDG	Maximum amount of application data in one datagram (in bytes)	
SCSMAXMSG	Maximum amount of application data in one SCS message (in bytes)	
SCSFLOWCUSH	Threshold value for notifying the remote SCS of new receive buffers	D
SCSSYSTEMID	Identifier of the system within a cluster	
PASTRETRY	Maximum number of times CI port driver attempts start datagram exchange	D
PASTIMOUT	Time interval between CI port driver wakeups for time-based operations (in seconds)	D
PASTDGBUF	Maximum number of CI port driver start handshakes in progress simultaneously	
PAPOLLINTERVAL	Time between CI port driver polling activations (in seconds)	D
PAPOOLINTERVAL	Time between wakeups for CI port driver message buffer allocation requests (in seconds)	D
UDABURSTRATE	Not implemented	

B. LISTE DES PARAMETRES PAR ORDRE ALPHABETIQUE

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
ACP-BASEPRIO	Priorité de base des processus ACP	8	4	31	priorité	D
ACP-DATACHECK	vérification des données sur les ACP	2	0	3	booléen	D
ACP-DIRCACHE	taille de la directory du cache	80	2	- 1	pages	D
ACP-EXTCACHE	nombre d'extensions du cache	64	0	- 1	extension	D
ACP-EXTLIMIT	grandeur maximum de l'espace libre dans l'extension du cache	300	0	1000	pourcent/10	D
ACP-FIDCACHE	nombre d'identification des fichiers du cache	64	0	- 1	id.fichiers	D
ACP-HDRCACHE	taille de l'entête des fichiers du cache	128	2	- 1	pages	D
ACP-MAPCACHE	taille du map de bits du cache	8	1	- 1	pages	D
ACP-MAXREAD	nombre maximum de blocks de la directory à lire	32	1	64	blocks	D
ACP-MULTIPLE	un ACP par volume disque monté	1	0	1	booléen	D
ACP-QUOCACHE	nombre d'entrées dans le quota du cache	64	0	- 1	utilisateurs	D
ACP-SHARE	code ACP en partagé	1	0	1	booléen	
ACP-SWAPFLGS	swapping des working-sets ACP	15	0	15	booléen	D
ACP-SYSACC	nombre de directory d'accès au cache	8	0	- 1	directory	D

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
ACP-WINDOW	nombre, par défaut, de pointeurs de fenêtre	7	1	- 1	pointeur	D
ACP-WORKSET	working-set par défaut pour ACP	0	0	- 1	pages	D
ACP-WRITEBACK	cache pour les entêtes de fichiers	1	0	1	booléen	D
AWSMIN	working-set réajusté minimum	50	0	- 1	pages	D
AWSTIME	temps de collecte d'échantillon pour le réajustement automatique d'un working-set	20	1	- 1	10 millisecc	D
BALSETCNT	nombre maximum de working-sets résidents	36	4	1024	entrées	
BJOBLIM	non utilisé					
BORROWLIM	taille minimum de la liste des pages libres avant qu'un processus puisse s'étendre au-delà de son quota	300	0	- 1	pages	D
BUGCHECKFATAL	tous les bugs sont fatals	0	0	1	booléen	D
BOGREBOOT	relancement automatique lors d'un bug fatal	1	0	1	booléen	D
CLISYMTBL	taille de la table des symbols de l'interpreteur de commandes	40	10	128	pages	D
CRDENABLE	détection et enregistrement des erreurs de lecture corrigés en mémoire	1	0	1	booléen	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
DEADLOCK-WAIT	temps d'attente d'une demande de verrou avant qu'une recherche de l'interblocage soit amorcée	10	0	- 1	secondes	D
DEFMBXBUFQUO	quota du buffer de boîte aux lettres par défaut	1056	256	64000	bytes	D
DEFMBXMXMSG	taille maximale des messages par boîte aux lettres par défaut	256	64	64000	bytes	D
DEFMBXNUMMSG	non utilisé					
DEFPRI	priorité de base par défaut des processus	4	1	31	priorité	D
DISMOUMSG	contrôle du traitement par OPCOM des messages de démontage de volume	0	0	1	booléen	D
DUMPCPU	écriture du fichier dump sur bug fatal	1	0	1	booléen	
EXTRACPU	temps CPU supplémentaire accordé aux processus quand leur temps CPU est écoulé	1000	0	- 1	10 millisecc	D
FREEGOAL	nombre de pages exigées dans la liste des pages libres après une pénurie	200	16	- 1	pages	
FREELIM	limite inférieure de la liste des pages libres	32	16	- 1	pages	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
GBLPAGES	nombre d'entrées dans la table des pages globales	3072	512	- 1	pages	
GBLPAGFIL	nombre maximum de pages systèmes accordées aux pages de fichier globales	1024	128	- 1	pages	
GBLSECTIONS	nombre de descripteurs de section globale	80	20	- 1	sections	
GROWLIM	nombre de pages exigées dans la liste des pages libres pour autoriser un processus à dépasser son quota	63	0	- 1	pages	D
IJOB LIM	non utilisé					
INTSTKPAGES	taille de pile d'interruption	2	1	- 1	pages	
IRPCOUNT	nombre de packets intermédiaires alloués	60	0	32768	packets	
IRPCOUNTV	taille maximale jusqu'où IRPCOUNT peut accroître	1000	0	32768	packets	
JOBQUEUES	utilisation des queues batch ou print	1	0	1	booléen	D
KFILSTCNT	nombre d'entête de listes de fichiers connus	4	2	255	entrées	
LAMAPREGS	nombre de registres map alloués à un driver pour LPA 11	0	0	255	mapregs	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unite	Dynamique
LOCKIDTBL	nombre d'entrées dans la table des verrous du système	128	16	16000	entrées	
LOGGHASHTBL	nombre d'entrées dans la table hash des nom logiques par groupe	32	1	16383	entrées	
LOGPHASHTBL	nombre d'entrées dans la table hash des noms logiques par processus	128	1	16383	entrées	
LOGSHASHTBL	nombre d'entrées dans la table hash des noms logiques pour le système	128	1	16383	entrées	
LONGWAIT	temps écoulé avant que le swapper juge un processus oisif	700	40	32767	6.6 mil.sec	D
LRPCOUNT	nombre de grands packets préalloués	4	0	4096	packets	
LRPCOUNTV	valeur maximale jusqu'où LRPCOUNT peut croître	80	0	4096	packets	
LRPSIZE	taille des grands packets	576	256	16384	bytes	
MAXBUF	nombre maximum de bytes qui peuvent être transférés en une opération I/O par buffer	1056	512	64000	bytes	D
MAXPRINTSYMB	nombre maximum d'imprimante	3	1	32	processus	D
MAXPROCESSCNT	nombre maximum de processus	72	12	8192	processus	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
MAXSYSGROUP	le plus grand UIC system	8	1	32768	groupe UIC	D
MINWSCNT	nombre minimum des pages fluides dans un working-set	20	10	- 1	pages	
MOUNTMSG	contrôle du traitement par OPCOM des messages de montage de volume	0	0	1	booléen	D
MPW-HILIMIT	taille maximale de la liste des pages modifiées	500	0	16384	pages	
MPW-LOLIMIT	taille minimale de la liste des pages modifiées	32	0	16384	pages	
MPW-THRESH	taille minimale de la liste des pages modifiées qui requiert l'action du swapper	200	0	16384		D
MPW-WAITLIMIT	nombre de pages de la liste des pages modifiées qui force un processus à attendre jusqu'à la prochaine recopie de pages de la liste des pages modifiées	500	0	16383		D
MPW-WRTCLUSTER	nombre de pages de la liste des pages modifiées recopiées par opération I/O	96	16	120	pages	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
MVTIMEOUT	temps autorisé pour la vérification du succès d'une opération de montage avant d'être annulée	600	1	64000	secondes	D
NJOBLIM	non utilisé					
NPAGEDYN	taille du pool dynamique non paginé	64000	16384	- 1	bytes	
NPAGEVIR	taille maximale jusqu'où NPAGEDYN peut croître	400000	16384	- 1	bytes	
PAGEDYN	taille du pool dynamique paginé	80000	8192	- 1	bytes	
PAGFILCNT	nombre maximum de fichiers de pagination	2	1	63	fichiers	
PAPOLLINTERVAL	temps entre les activations du CI port driver	15	2	32767	secondes	
PAPOOLINTERVAL	temps entre les réveils pour demandes d'allocation de buffer pour message par CI port driver	15	2	32767	secondes	D
PASTDGBUF	nombre maximum de CI port driver commençant simultanément	4	1	16	buffers	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
PASTIMOUT	intervalle de temps entre les réveils de CI port driver pour des opérations synchronisés	5	2	100	secondes	D
PASTRETRY	nombre maximum de fois q'un CI port driver tente de commencer un envoi de données	4	0	32767	essais	D
PFCDEFAULT	taille du cluster lu lors d'un défaut de page	32	0	127	pages	D
PFRATH	limite supérieure du taux de défaut de page	120	0	- 1	défaut/ 10 sec	D
PQL-DASTLM	nombre, par défaut, d'AST en attente	6	- 1	- 1	AST	D
PQL-DBIOLM	limite, par défaut, des I/O bufferisés	6	- 1	- 1	I/O	D
PQL-DBYTLM	limite, par défaut, en byte des I/O bufferisés	8192	- 1	- 1	bytes	D
PQL-DCPVLM	limite, par défaut, du temps CPU	0	- 1	- 1	10 milli sec.	D
PQL-DDIOLM	limite, par défaut, des I/O directs	6	- 1	- 1	I/O	D

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
PQL-DENQLM	nombre de verrous, par défauts, en file en un temps donné	20	- 1	- 1	verrous	D
PQL-DFILLM	limite, par défaut, du nombre de fichiers ouverts	16	- 1	- 1	fichiers	D
PQL-DPGFLQUOTA	quota, par défaut, dans le fichier de pagination	2048	- 1	- 1	pages	D
PQL-DPRCLM	limite, par défaut, de sous-processus	8	- 1	- 1	processus	D
PQL-DTQELM	nombre d'entrées en file en même temps, par défaut	8	- 1	- 1		D
PQL-DWSDEFAULT	taille des working-sets, par défaut	100	- 1	- 1	pages	
PQL-DWSEXTENT	extension des working-sets, par défaut	200	- 1	- 1	pages	D
PQL-DWSQUOTA	quota des working-sets, par défaut	200	- 1	- 1	pages	D
PQL-MASTLM	nombre minimum d'AST en attente	2	- 1	- 1	AST	D
PQL-MBIOLM	limite inférieure des I/O bufferisés	2	- 1	- 1	I/O	D
PQL-MBYTLM	limite inférieure en byte des I/O des bufferisés	1024	- 1	- 1	bytes	D

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
PQL-MCPULM	limite inférieure du temps CPU	0	- 1	- 1	10 milli sec.	D
PQL-MDIOLM	limite inférieure des I/O directs	2	- 1	- 1	I/O	D
PQL-MENQLM	nombre minimum de verrous en file en un temps donné	2	- 1	- 1	verrous	D
PQL-MFILLM	limite inférieure du nombre de fichiers ouverts	2	- 1	- 1	fichiers	D
PQL-MPGFLQUOTA	quota minimum dans le fichier de pagination	256	- 1	- 1	pages	D
PQL-MPRCLM	limite inférieure des sous-processus	0	- 1	- 1	processus	D
PQL-MTQELM	nombre minimum d'entrées en file en même temps	0	- 1	- 1		D
PQL-MWSDEFAULT	taille minimale des working-sets	10	- 1	- 1	pages	
PQL-MWSEXTENT	extension minimale des working-sets	10	- 1	- 1	pages	D
PQL-MWSQUOTA	quota minimum des working-sets	10	- 1	- 1	pages	D
PROCSECTCNT	nombre de sections d'un processus	32	5	1024	sections	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
QUANTUM	temps maximum qu'un processus peut utiliser en une fois et temps minimum qu'un processus doit recevoir avant d'être swappé	20	2	32767	10 milli sec.	D
REALTIME_SPTS	nombre d'entrées dans la table des pages système réservées pour les processus de connection d'interruption	0	0	- 1	pages	
REINITIQUE	réinitialisation des queues	0	0	1	booléen	D
RESHASH TBL	nombre d'entrées dans la table hash des noms de ressources verrouillables	32	1	8192	entrées	
RJOB LIM	nombre maximum de terminaux à distance	16	0	254	jobs	D
RMS-DFMBC	nombre de blocks, par défaut	16	1	127	blocks	D
RMS-DFMBF HSH	non utilisé					
RMS-DFMBF IDX	nombre de buffers, par défaut, pour les opérations disque avec fichier séquentiel indexé	0	0	127	buffers	D

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
RMS-DFMBFREL	nombre de buffers, par défaut, pour les opérations disque avec fichier relatif	0	0	127	buffers	D
RMS-DFMBFSDK	nombre de buffers, par défaut, pour les opérations disque avec fichier séquentiel	0	0	127	blocks	D
RMS-DFMBFSMT	nombre de buffers, par défaut, pour les opérations avec bande magnétique	0	0	127	blocks	D
RMS-DFMBFSUR	non utilisé					
RMS-EXTENT-SIZE	taille d'extension de fichier, par défaut	80	0	65535	blocks	D
RMS-FILEPROT	protection de fichier, par défaut	64000	0	65535	prot-mark	D
RMS-PROLOGUE	niveau de structure de fichier, par défaut pour les fichiers VAX-II RMS	0	0	3	niv. prologue	D
SCSBUFFCNT	non utilisé					
SCSCONNCNT	nombre maximum de connections SCS pour les applications système	18	2	32767	entrées	
SCSFLOWCUSH	valeur initiale pour notifier les SCS de nouveaux buffers reçus	0	0	16	crédits	D

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
SCSMAXDG	nombre maximum de données d'application dans un envoi	576	28	985	bytes	
SCSMAXMSG	nombre maximum de données d'application dans un message SCS	96	52	985	bytes	
SCSRESPCNT	nombre maximum d'entrées dans la table des descripteurs de réponse pour les applications système	20	0	32767	entrées	
SCSSYSTEMID	identifieur du système dans un cluster	1	1	255	nombre pur	
SETTIME	heure du jour donnée au lancement du système	0	0	1	booléen	
SPTREQ	nombre d'entrées dans la table des pages du système supplémentaires	720	- 1	- 1	pages	
SRPCOUNT	nombre de petits packets préalloués	120	0	4096	packets	
SRPCOUNTV	valeur maximale jusqu'où SRPCOUNT peut croître	1000	0	131072	packets	
SWPFILCNT	nombre maximum de swapping files	2	0	63	fichiers	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
SWPOUTPGCNT	taille du processus avant d'être swappé	60	0	- 1	pages	D
SYSMWCNT	taille maximale du working set du système	160	20	16384	pages	
TIMEPROMPTWAIT	temps autorisé pour l'entrée de l'heure système pendant le lancement	- 1	0	- 1		
TTY-ALTALARM	taille de l'alarme de buffer secondaire	64	0	- 1	bytes	
TTY-ALTYPAHD	taille du buffer secondaire	200	0	- 1	bytes	
TTY-BUF	longueur des lignes, par défaut, pour les terminaux	80	0	65535	caractères	
TTY-CLASSNAME	préfixe du nom de driver de la classe des terminaux	21588	16705	23130	ASCII	
TTY-DEFCHAR	caractéristiques des terminaux, par défaut	402657952	0	- 1	bit encodé	
TTY-DEFCHARZ	caractéristiques des terminaux, par défaut	2	0	- 1	bit encodé	
TTY-DIALTYPE	dialup flag bits	0	0	255	bit encodé	

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
TTY-DMASIZE	nombre minimum de caractères en buffer pour invoquer le transfert DMA	64	0	- 1	bytes	D
TTY-OWNER	UIC pour la spécification TTY-PROT	65540	0	- 1	UIC	
TTY-PARITY	non utilisé					
TTY-PROT	protection du terminal contre l'allocation par un autre processus	65520	0	- 1	protection	
TTY-RSPEED	vitesse de réception des terminaux	0	0	16	spécial	
TTY-SCANDELTA	intervalle pour l'élection des terminaux aux événements dial-up/hang-up	10000000	100000	- 1	100 nano-sec	
TTY-SILOTIME	intervalle d'électron de silo input pour l'hardware DMF-32	8	0	255	millisec	
TTY-SPEED	vitesse, par défaut, des terminaux	15	1	16	special	
TTY-TYPAHDSZ	taille du buffer terminal	78	0	- 1	bytes	
UAFALTERNATE	utilisation des UAF secondaires	0	0	1	booléen	
UDABURSTRATE	non utilisé					

NOM	DESCRIPTION	Valeur par défaut	Min	Maximum	Unité	Dynamique
VIRTUALPAGECNT	espace virtuel maximum par processus	8192	512	262144	pages	
WSDEC	décrémentation d'un working-set	35	0	- 1	pages	D
WSINC	incrémentation d'un working-set	150	0	- 1	pages	D
WSMAX	taille maximale de tout working-set	1024	60	16384	pages	
XFMAXRATE	taux maximum de transfert pour les devices DR 32	236	0	255	special	D

2. LE LOGICIEL DE MESURE "MONITOR"  
=====

Les états de sortie sont :

1. Processes
2. Process states
3. Time in processor modes
4. Page management statistics
5. I/O system statistics
6. File primitive statistics
7. Nonpaged pool statistics
8. Lock management statistics
9. Decnet statistics

Process Count: 18

VAX/VMS Monitor Utility  
PROCESSES  
SUMMARY

Uptime: 0 22:54:00

From: 5-AUG-1983 08:00:00  
To: 5-AUG-1983 18:00:57

PID	UIC	STATE	PRI	NAME	SIZE	DIOCNT	FAULTS	CPU TIME
00010000	[000,000]	COM	0	NULL	0/0	0	0	15:46:35.7
00010001	[000,000]	HIB	16	SWAPPER	0/0	0	0	00:00:50.5
0006002B	[001,004]	HIB	8	PRTSYMB1	0/16	168	2902	00:01:05.3
0004002D	[001,004]	HIB	8	PRTSYMB4	0/16	461	6034	00:01:58.4
00060037	[001,004]	HIB	8	PRTSYMB3	0/16	311	4936	00:01:45.6
00010039	[001,003]	HIB	8	REMACP	0/21	1	33	00:00:00.0
0003003A	[001,004]	HIB	5	EVL	0/19	2	4967	00:00:03.5
0006003B	[001,004]	HIB	10	NETACP	0/92	69	407	00:00:02.6
000E003C	[001,005]	CUR	4	_JOB32	0/90	412	1804	00:00:11.9
0003003D	[007,004]	LEF	4	_TTB0:	40/100	16	641	00:00:03.0
0001003E	[007,004]	LEF	4	_TTB1:	17/74	91770	243034	00:15:36.8
0001003F	[001,004]	HIB	8	PRTSYMB2	0/16	581	8391	00:03:00.1
000B0040	[113,101]	LEF	4	DOUC01	27/76	24	4802	00:00:17.7
00010041	[001,004]	LEF	5	DPCOM	0/44	101	602	00:00:02.8
00010042	[001,004]	HIB	9	JOB_CONTROL	23/76	1303	7801	00:00:43.3
00210043	[113,101]	LEF	4	_CRA0:	4/106	7	75	00:00:00.7
00010044	[001,003]	HIB	10	DRAOBACP	58/174	47371	69040	00:19:29.2
00020045	[001,006]	HIB	3	ERRFMT	0/46	431	118	00:00:06.0

VAX/VMS Monitor Utility  
 PROCESS STATES  
 SUMMARY

From: 5-AUG-1983 08:00:00  
 To: 5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
Collided Page Wait	0.00	0.00	0.00	0.00
Mutex & Misc Resource Wait	0.00	0.03	0.00	1.00
Common Event Flag Wait	0.00	0.00	0.00	0.00
Page Fault Wait	0.00	0.08	0.00	4.00
Local Event Flag Wait	5.00	17.53	4.00	27.00
Local Evt Flag (Outswapped)	0.00	0.00	0.00	0.00
Hibernate	11.00	10.84	8.00	14.00
Hibernate (Outswapped)	0.00	0.00	0.00	0.00
Suspended	0.00	0.00	0.00	0.00
Suspended (Outswapped)	0.00	0.00	0.00	0.00
Free Page Wait	0.00	0.00	0.00	0.00
Compute	1.00	1.59	1.00	11.00
Compute (Outswapped)	0.00	0.00	0.00	0.00
Current Process	1.00	1.00	1.00	1.00

SUMMARIZING

RECORDING

+-----+  
| CUR |  
+-----+

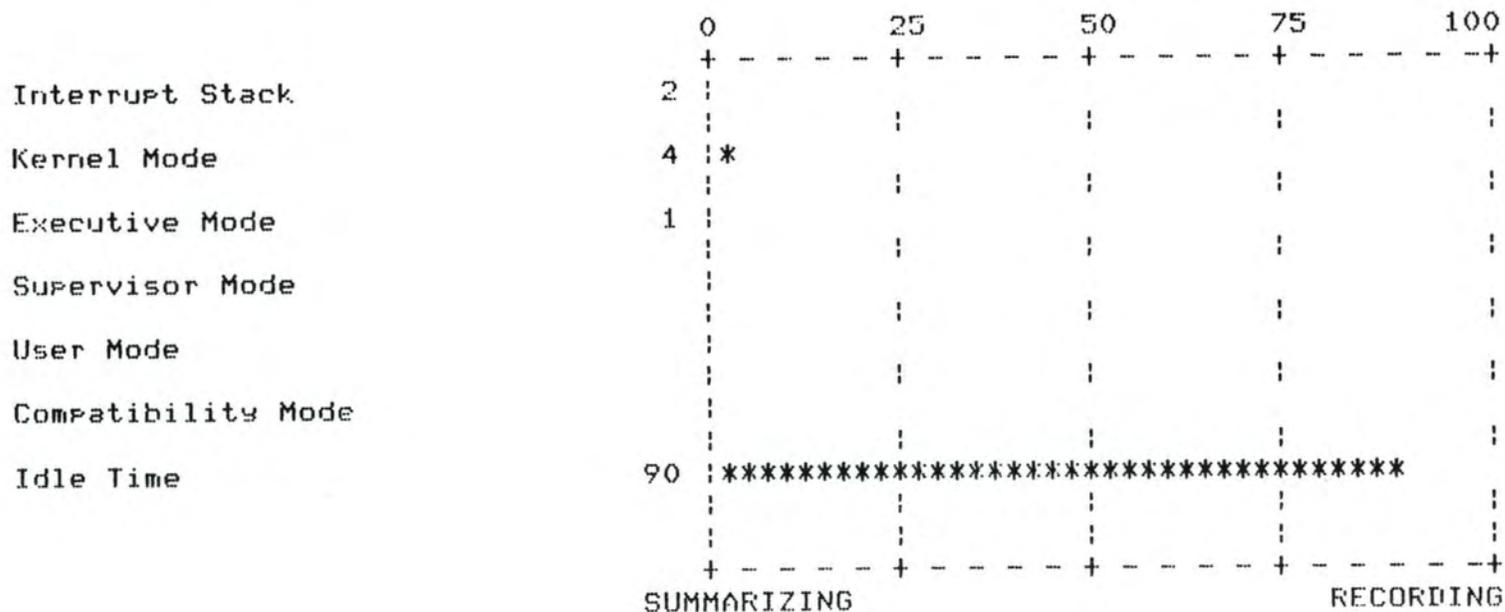
VAX/VMS Monitor Utility

TIME IN PROCESSOR MODES

SUMMARY

From: 5-AUG-1983 08:00:00

To: 5-AUG-1983 18:00:57



VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 5-AUG-1983 08:00:00  
To: 5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
Page Fault Rate	6.17	78.03	0.18	301.83
Page Read Rate	0.74	8.35	0.00	27.55
Page Read I/O Rate	0.31	1.17	0.00	4.52
Page Write Rate	0.31	3.89	0.00	18.05
Page Write I/O Rate	0.02	0.06	0.00	0.32
Free List Fault Rate	2.38	20.90	0.08	75.61
Modified List Fault Rate	2.09	33.56	0.02	164.99
Demand Zero Fault Rate	0.84	4.85	0.00	14.11
Global Valid Fault Rate	0.52	17.33	0.00	73.61
Wrt In Progress Fault Rate	0.00	0.03	0.00	0.54
System Fault Rate	0.00	3.04	0.00	11.97
Free List Size	3589.00	2086.54	948.00	3712.00
Modified List Size	202.00	283.84	14.00	491.00

SUMMARIZING

RECORDING

VAX/VMS Monitor Utility  
I/O SYSTEM STATISTICS  
SUMMARY

From: 5-AUG-1983 08:00:00  
To: 5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
Direct I/O Rate	0.57	9.22	0.01	40.71
Buffered I/O Rate	10.14	20.78	0.00	58.80
Mailbox Write Rate	0.06	0.08	0.00	0.63
Window Turn Rate	0.00	0.01	0.00	0.57
Los Name Translation Rate	1.81	5.88	0.00	26.90
File Open Rate	0.09	0.56	0.00	3.83
Page Fault Rate	6.17	78.03	0.18	301.86
Page Read Rate	0.74	8.35	0.00	27.55
Page Read I/O Rate	0.31	1.17	0.00	4.52
Page Write Rate	0.31	3.89	0.00	18.05
Page Write I/O Rate	0.02	0.06	0.00	0.32
Inswap Rate	0.00	0.00	0.00	0.00
Free List Size	3589.00	2086.40	948.00	3712.00
Modified List Size	202.00	283.65	14.00	491.00

SUMMARIZING

RECORDING

VAX/VMS Monitor Utility  
FILE PRIMITIVE STATISTICS

SUMMARY From: 5-AUG-1983 08:00:00  
To: 5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
FCP Call Rate	0.72	1.95	0.00	15.28
Allocation Rate	0.01	0.10	0.00	0.41
Create Rate	0.01	0.09	0.00	0.68
Disk Read Rate	0.12	0.48	0.00	7.97
Disk Write Rate	0.20	0.40	0.00	1.56
Cache Hit Rate	2.81	4.13	0.00	113.44
CPU Tick Rate	1.03	2.29	0.00	29.01
Window Turn Rate	0.00	0.01	0.00	0.57
File Lookup Rate	0.29	0.66	0.00	4.06
File Open Rate	0.09	0.56	0.00	3.83

SUMMARIZING

RECORDING

VAX/VMS Monitor Utility  
NONPAGED POOL STATISTICS

SUMMARY From: 5-AUG-1983 08:00:00  
To: 5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
SRPs Left	739.00	258.99	7.00	742.00
IRPs Left	649.00	603.41	567.00	656.00
LRPs Left	5.00	4.01	1.00	5.00
Total Space Left	88368.00	79714.26	73264.00	88368.00
Holes In Pool	3.00	15.46	2.00	28.00
Largest Block	86176.00	76963.47	71536.00	86176.00
Smallest Block	496.00	86.70	16.00	496.00
Blocks Less or Eq 32 Bytes	0.00	2.14	0.00	7.00

SUMMARIZING

RECORDING

VAX/VMS Monitor Utility  
LOCK MANAGEMENT STATISTICS

SUMMARY

From:

5-AUG-1983 08:00:00

To:

5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
New ENQ Rate	0.03	7.88	0.00	75.89
Converted ENQ Rate	0.02	15.61	0.00	147.85
DEQ Rate	0.03	7.87	0.00	75.68
ENQs Forced To Wait Rate	0.00	0.00	0.00	0.14
ENQs Not Queued Rate	0.00	0.00	0.00	0.44
Deadlock Search Rate	0.00	0.00	0.00	0.00
Deadlock Find Rate	0.00	0.00	0.00	0.00
Total Locks	0.00	127.92	0.00	255.00
Total Resources	0.00	87.31	0.00	193.00

SUMMARIZING

RECORDING

VAX/VMS Monitor Utility  
DECNET STATISTICS

SUMMARY

From:

5-AUG-1983 08:00:00

To:

5-AUG-1983 18:00:57

	CUR	AVE	MIN	MAX
Arriving Local Packet Rate	0.00	0.18	0.00	13.39
Departing Local Packet Rate	0.00	0.18	0.00	13.27
Arriving Trans Packet Rate	0.00	0.00	0.00	0.00
Trans Congestion Loss Rate	0.00	0.00	0.00	0.00
Receiver Buff Failure Rate	0.00	0.00	0.00	0.00
LRPs Left	5.00	4.01	1.00	5.00

SUMMARIZING

RECORDING

### 3. LES MESURES.

#### A. PARAMETRES INITIAUX.

Parameters in use:	Active					
Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
PFCDEFAULT	127	32	0	127	Pages	Dynamic
KFILSTCNT	4	4	2	255	Slots	
GBLSECTIONS	100	80	20	-1	Sections	
GBLPAGES	3326	3072	512	-1	Pages	
GBLPAGFIL	1024	1024	128	-1	Pages	
MAXPROCESSCNT	70	72	12	8192	Processes	
PROSECTCNT	32	32	5	1024	Sections	
MINWSCNT	20	20	10	-1	Pages	
PAGFILCNT	2	2	1	63	Files	
SWPFILCNT	2	2	0	63	Files	
SYSMWCNT	247	160	20	16384	Pages	
INTSTKPAGES	2	2	1	-1	Pages	
BALSETCNT	52	36	4	1024	Slots	
IRPCOUNT	750	60	0	32768	Packets	
IRPCOUNTV	780	1000	0	32768	Packets	
WSMAX	3072	1024	60	16384	Pages	
NPAGEDYN	179712	64000	16384	-1	Bytes	
NPAGEVIR	281600	400000	16384	-1	Bytes	
PAGEDYN	84992	80000	8192	-1	Bytes	
VIRTUALPAGECNT	16384	8192	512	262144	Pages	
SPTREQ	720	720	-1	-1	Pages	
LRPCOUNT	8	4	0	4096	Packets	
LRPCOUNTV	60	80	0	4096	Packets	
LRPSIZE	576	576	256	16384	Bytes	
SRPCOUNT	380	120	0	4096	Packets	
SRPCOUNTV	1140	1000	0	131072	Packets	
QUANTUM	20	20	2	32767	10Ms	Dynamic
MPW_WRTCLUSTER	120	96	16	120	Pages	
MPW_HILIMIT	500	500	0	16384	Pages	
MPW_LOLIMIT	120	32	0	16384	Pages	
KPW_THRESH	200	200	0	16384		Dynamic
MPW_WAITLIMIT	500	500	0	16384		Dynamic
PFRATL	1	1	0	-1	FIts/10Sec	Dynamic
PFRATH	120	120	0	-1	FIts/10Sec	Dynamic
WSINC	21	150	0	-1	Pages	Dynamic
WSDEC	35	35	0	-1	Pages	Dynamic
AWSMIN	50	50	0	-1	Pages	Dynamic
AWSTIME	20	20	1	-1	10Ms	Dynamic
SWPOUTPGCNT	60	60	0	-1	Pages	Dynamic
LONGWAIT	700	700	40	32767	6.6Ms	Dynamic
EXTRACPU	1000	1000	0	-1	10Ms	Dynamic
MAXSYSGROUP	8	8	1	32768	UIC Group	Dynamic
KVTIMEDOUT	600	600	1	64000	Seconds	Dynamic
MAXBUF	1056	1056	512	64000	Bytes	Dynamic
DEFMBXBUFQUO	1056	1056	256	64000	Bytes	Dynamic
DEFMBXMXMSG	256	256	64	64000	Bytes	Dynamic
DEFMBXNUMMSG	16	16	1	-1	Messages	Dynamic

## Parameters in use: Active

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
FREELIM	16	32	16	-1	Pages	
FREEGOAL	156	200	16	-1	Pages	
GROWLIM	51	63	0	-1	Pages	Dynamic
BORROWLIM	300	300	0	-1	Pages	Dynamic
XFMAXRATE	236	236	0	255	Special	Dynamic
LAMAPREGS	0	0	0	255	Mapress	
REALTIME_SPTS	0	0	0	-1	Pages	
CLISYMTBL	40	40	10	128	Pages	Dynamic
LOCKIDTBL	1000	128	16	16000	Entries	
RESHASHTBL	256	32	1	8192	Entries	
DEADLOCK_WAIT	10	10	0	-1	Seconds	Dynamic
SCSBUFFCNT	10	10	0	32767	Entries	
SCSCONNCNT	18	18	2	32767	Entries	
SCSRESPCNT	20	20	0	32767	Entries	
SCSMAXDG	576	576	28	985	Bytes	
SCSMAXMSG	96	96	52	985	Bytes	
SCSFLOWCUSH	0	0	0	16	Credits	Dynamic
SCSSYSTEMID	1	1	1	255	Pure-number	
PASTRETRY	4	4	0	32767	Retries	Dynamic
PASTIMOUT	5	5	2	100	Seconds	Dynamic
PASTDGBUF	4	4	1	16	Buffers	
PAPOLLINTERVAL	15	15	2	32767	Seconds	Dynamic
PAPOLINTERVAL	15	15	2	32767	Seconds	Dynamic
TIMEPROMPTWAIT	65535	-1	0	-1	uFortnights	
UDABURSTRATE	0	0	0	31	Longwords	
LOGSHASHTBL	128	128	1	16383	Entries	
LOGGHASHTBL	32	32	1	16383	Entries	
LOGPHASHTBL	128	128	1	16383	Entries	
BUGREBOOT	1	1	0	1	Boolean	Dynamic
CRDENABLE	1	1	0	1	Boolean	
DUMFBUG	1	1	0	1	Boolean	
BUGCHECKFATAL	0	0	0	1	Boolean	Dynamic
ACP_MULTIPLE	1	1	0	1	Boolean	Dynamic
SETTIME	0	0	0	1	Boolean	
ACP_SHARE	1	1	0	1	Boolean	
UAFALTERNATE	0	0	0	1	Boolean	
JOBQUEUES	1	1	0	1	Boolean	Dynamic
REINITQUE	0	0	0	1	Boolean	Dynamic
MOUNTMSG	0	0	0	1	Boolean	Dynamic
DISMOUNMSG	0	0	0	1	Boolean	Dynamic
TTY_SCANDELTA	10000000	10000000	100000	-1	100Ns	
TTY_DIALTYPE	0	0	0	255	Bit-Encoded	
TTY_SPEED	15	15	1	16	Special	
TTY_RSPEED	0	0	0	16	Special	
TTY_PARITY	24	24	0	-1	Special	
TTY_BUF	80	80	0	65535	Characters	
TTY_DEFCHAR	402657952	402657952	0	-1	Bit-Encoded	
TTY_DEFCHAR2	2	2	0	-1	Bit-Encoded	
TTY_TYPAHDSZ	78	78	0	-1	Bytes	
TTY_ALTYPAMD	200	200	0	-1	Bytes	
TTY_ALTALARM	64	64	0	-1	Bytes	
TTY_DMASIZE	64	64	0	-1	Bytes	Dynamic
TTY_PROT	65520	65520	0	-1	Protection	
TTY_OWNER	65540	65540	0	-1	UIC	
TTY_CLASSNAME	21588	21588	16705	23130	Ascii	
TTY_SILOTIME	8	8	0	255	Ms	

## Parameters in use: Active

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
RMS_DFMBBC	16	16	1	127	Blocks	Dynamic
RMS_DFMBFSIK	0	0	0	127	Blocks	Dynamic
RMS_DFMBFSMT	0	0	0	127	Blocks	Dynamic
RMS_DFMBFSUR	0	0	0	127	Buffers	Dynamic
RMS_DFMBFREL	0	0	0	127	Buffers	Dynamic
RMS_DFMBFIDX	4	0	0	127	Buffers	Dynamic
RMS_DFMBFHSH	0	0	0	127	Buffers	Dynamic
RMS_PROLOGUE	0	0	0	3	Prolog-Lvl	Dynamic
RMS_EXTEND_SIZE	80	80	0	65535	Blocks	Dynamic
RMS_FILEPROT	64000	64000	0	65535	Prot-mask	
PQL_DASTLM	6	6	-1	-1	Ast	Dynamic
PQL_MASTLM	2	2	-1	-1	Ast	Dynamic
PQL_DBIOLM	6	6	-1	-1	I/O	Dynamic
PQL_MBIOLM	2	2	-1	-1	I/O	Dynamic
PQL_DBYTLM	8192	8192	-1	-1	Bytes	Dynamic
PQL_MBYTLM	1024	1024	-1	-1	Bytes	Dynamic
PQL_DCPULM	0	0	-1	-1	IOBs	Dynamic
PQL_MCPULM	0	0	-1	-1	IOBs	Dynamic
PQL_DDIOLM	6	6	-1	-1	I/O	Dynamic
PQL_MDIOLM	2	2	-1	-1	I/O	Dynamic
PQL_DFILLM	16	16	-1	-1	Files	Dynamic
PQL_MFILLM	2	2	-1	-1	Files	Dynamic
PQL_DPGFLQUOTA	2048	2048	-1	-1	Pages	Dynamic
PQL_MPGFLQUOTA	256	256	-1	-1	Pages	Dynamic
PQL_DPRCLM	8	8	-1	-1	Processes	Dynamic
PQL_MPRCLM	0	0	-1	-1	Processes	Dynamic
PQL_DTQELM	8	8	-1	-1	Timers	Dynamic
PQL_MTQELM	0	0	-1	-1	Timers	Dynamic
PQL_DWSDEFAULT	100	100	-1	-1	Pages	
PQL_MWSDEFAULT	10	10	-1	-1	Pages	
PQL_DWSQUOTA	200	200	-1	-1	Pages	Dynamic
PQL_MWSQUOTA	10	10	-1	-1	Pages	Dynamic
PQL_DWSEXTENT	200	200	-1	-1	Pages	Dynamic
PQL_MWSEXTENT	10	10	-1	-1	Pages	Dynamic
PQL_DENQLM	50	20	-1	-1	Locks	Dynamic
PQL_MENQLM	5	2	-1	-1	Locks	Dynamic
ACP_MAPCACHE	20	8	1	-1	Pages	Dynamic
ACP_HIRCACHE	120	128	2	-1	Pages	Dynamic
ACP_DIRCACHE	104	80	2	-1	Pages	Dynamic
ACP_WORKSET	0	0	0	-1	Pages	Dynamic
ACP_FIDCACHE	64	64	0	-1	File-Ids	Dynamic
ACP_EXTCACHE	32	64	0	-1	Extents	Dynamic
ACP_EXTLIMIT	300	300	0	1000	Percent/10	Dynamic
ACP_QUOCACHE	70	64	0	-1	Users	Dynamic
ACP_SYSACC	10	8	0	-1	Directories	Dynamic
ACP_MAXREAD	32	32	1	64	Blocks	Dynamic
ACP_WINDOW	7	7	1	-1	Pointers	Dynamic
ACP_WRITEBACK	1	1	0	1	Boolean	Dynamic
ACP_DATACHECK	2	2	0	3	Boolean	Dynamic
ACP_BASEPRIO	8	8	4	31	Priority	Dynamic
ACP_SWAPFLGS	14	15	0	15	Boolean	Dynamic
MAXPRINTSYMB	4	3	1	32	Processes	Dynamic
DEFPRI	3	4	1	31	Priority	Dynamic
IJOBLIM	64	64	1	1024	Jobs	Dynamic
BJOBLIM	16	16	0	1024	Jobs	Dynamic
NJOBLIM	16	16	0	1024	Jobs	Dynamic
RJOBLIM	16	16	0	254	Jobs	Dynamic

TEST 1: Situation Initiale.

B. RESULTS PAR "MONITOR"

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS

SUMMARY

From: 14-AUG-1983 09:26:46

To: 14-AUG-1983 09:37:49

	CUR	AVE	MIN	MAX
Page Fault Rate	24.26	114.24	0.00	836.63
Page Read Rate	1.96	7.99	0.00	57.51
Page Read I/O Rate	1.31	2.88	0.00	15.68
Page Write Rate	0.00	9.94	0.00	394.73
Page Write I/O Rate	0.00	0.08	0.00	3.28
Free List Fault Rate	9.83	17.72	0.00	317.98
Modified List Fault Rate	7.54	67.40	0.00	758.41
Demand Zero Fault Rate	0.00	5.80	0.00	252.47
Global Valid Fault Rate	5.57	19.19	0.00	83.12
Wrt In Progress Fault Rate	0.00	0.79	0.00	60.85
System Fault Rate	0.00	0.00	0.00	0.92
Free List Size	4111.00	2623.41	146.00	4120.00
Modified List Size	127.00	280.48	27.00	495.00

PLAYBACK

SUMMARIZING

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS

SUMMARY

From: 13-AUG-1983 22:10:32  
To: 13-AUG-1983 22:21:03

	CUR	AVE	MIN	MAX
Page Fault Rate	19.47	98.85	0.00	862.82
Page Read Rate	0.00	7.29	0.00	39.80
Page Read I/O Rate	0.00	2.57	0.00	12.13
Page Write Rate	0.00	1.33	0.00	118.03
Page Write I/O Rate	0.00	0.01	0.00	0.98
Free List Fault Rate	6.27	10.58	0.00	124.53
Modified List Fault Rate	9.24	62.25	0.00	830.26
Demand Zero Fault Rate	0.00	4.17	0.00	22.69
Global Valid Fault Rate	3.96	18.75	0.00	76.31
Wrt In Progress Fault Rate	0.00	0.06	0.00	11.47
System Fault Rate	0.00	0.00	0.00	0.33
Free List Size	4148.00	2630.46	148.00	4148.00
Modified List Size	123.00	329.78	123.00	491.00

PLAYBACK

SUMMARIZING

TEST 3 : MPW - LOLIMIT 120 ↓ 32 ; MPW - WRCLUSTER 120 ↓ 96

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS

SUMMARY

From: 14-AUG-1983 08:37:59  
To: 14-AUG-1983 08:48:49

	CUR	AVE	MIN	MAX
Page Fault Rate	27.39	108.52	0.00	610.56
Page Read Rate	0.66	8.06	0.00	43.17
Page Read I/O Rate	0.33	2.94	0.00	13.15
Page Write Rate	0.00	13.58	0.00	405.61
Page Write I/O Rate	0.00	0.14	0.00	4.29
Free List Fault Rate	11.88	20.10	0.00	316.50
Modified List Fault Rate	9.57	58.56	0.00	401.31
Demand Zero Fault Rate	0.00	5.87	0.00	131.90
Global Valid Fault Rate	5.61	19.36	0.00	63.96
Wrt In Progress Fault Rate	0.00	1.24	0.00	44.82
System Fault Rate	0.00	0.01	0.00	1.97
Free List Size	4139.00	2627.68	157.00	4139.00
Modified List Size	124.00	306.08	13.00	495.00

PLAYBACK

SUMMARIZING

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 10:14:00  
To: 14-AUG-1983 10:24:46

	CUR	AVE	MIN	MAX
Page Fault Rate	32.67	112.87	0.00	709.29
Page Read Rate	3.63	7.55	0.00	43.23
Page Read I/O Rate	2.31	2.75	0.00	15.84
Page Write Rate	0.00	11.15	0.00	354.09
Page Write I/O Rate	0.00	0.09	0.00	2.95
Free List Fault Rate	10.89	18.61	0.00	341.11
Modified List Fault Rate	9.90	65.75	0.00	542.12
Demand Zero Fault Rate	1.65	5.85	0.00	176.92
Global Valid Fault Rate	7.92	18.67	0.00	74.67
Wrt In Progress Fault Rate	0.00	0.76	0.00	29.93
System Fault Rate	0.00	0.00	0.00	0.98
Free List Size	4095.00	2617.59	96.00	4095.00
Modified List Size	91.00	261.29	32.00	493.00

PLAYBACK

SUMMARIZING

TEST 5 PFCDEFAULT 127 ↘ 64.

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 10:39:05  
To: 14-AUG-1983 10:49:48

	CUR	AVE	MIN	MAX
Page Fault Rate	35.60	108.82	0.00	799.34
Page Read Rate	8.90	7.31	0.00	45.72
Page Read I/O Rate	5.63	2.68	0.00	17.10
Page Write Rate	0.00	9.50	0.00	393.44
Page Write I/O Rate	0.00	0.07	0.00	3.27
Free List Fault Rate	13.35	16.89	0.00	367.62
Modified List Fault Rate	6.23	64.09	0.00	781.90
Demand Zero Fault Rate	2.67	5.73	0.00	226.79
Global Valid Fault Rate	7.71	18.35	0.00	72.03
Wrt In Progress Fault Rate	0.00	0.65	0.00	33.33
System Fault Rate	1.78	0.01	0.00	1.78
Free List Size	4078.00	2653.50	63.00	4139.00
Modified List Size	108.00	269.08	35.00	489.00

PLAYBACK

SUMMARIZING

VAX/VMS Monitor Utilities  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 10:58:27  
To: 14-AUG-1983 11:09:20

	CUR	AVE	MIN	MAX
Page Fault Rate	21.45	127.56	0.00	780.85
Page Read Rate	1.98	6.74	0.00	43.56
Page Read I/O Rate	1.32	2.44	0.00	12.69
Page Write Rate	0.00	17.83	0.00	458.59
Page Write I/O Rate	0.00	0.14	0.00	3.82
Free List Fault Rate	12.54	24.90	0.00	371.94
Modified List Fault Rate	5.28	74.67	0.00	692.30
Demand Zero Fault Rate	0.00	5.77	0.00	161.09
Global Valid Fault Rate	2.31	17.92	0.00	86.51
Wrt In Progress Fault Rate	0.00	1.42	0.00	42.01
System Fault Rate	0.00	0.00	0.00	0.32
Free List Size	4088.00	2597.93	294.00	4089.00
Modified List Size	139.00	300.71	10.00	485.00

PLAYBACK

SUMMARIZING

VAX/VMS Monitor Utilities  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 11:16:53  
To: 14-AUG-1983 11:27:21

	CUR	AVE	MIN	MAX
Page Fault Rate	20.46	130.89	0.00	1265.67
Page Read Rate	1.98	7.23	0.00	53.13
Page Read I/O Rate	1.32	2.89	0.00	17.49
Page Write Rate	0.00	24.05	0.00	554.45
Page Write I/O Rate	0.00	0.20	0.00	4.62
Free List Fault Rate	9.57	29.56	0.00	467.98
Modified List Fault Rate	7.26	71.85	0.00	1064.68
Demand Zero Fault Rate	0.00	4.15	0.00	28.94
Global Valid Fault Rate	2.31	18.53	0.00	86.18
Wrt In Progress Fault Rate	0.00	3.31	0.00	115.78
System Fault Rate	0.00	0.00	0.00	0.33
Free List Size	4071.00	2617.58	649.00	4073.00
Modified List Size	147.00	320.64	20.00	499.00

PLAYBACK

SUMMARIZING

TEST B · MPW-LOLIMIT ARO ↑ 330 ; PFCDEFAULT 127 ↘ 64.

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 19:24:13  
To: 14-AUG-1983 19:35:20

	CUR	AVE	MIN	MAX
Page Fault Rate	29.41	115.72	0.00	588.38
Page Read Rate	4.95	8.69	0.00	54.40
Page Read I/O Rate	3.71	3.09	0.00	15.14
Page Write Rate	0.00	8.46	0.00	237.62
Page Write I/O Rate	0.00	0.07	0.00	2.23
Free List Fault Rate	11.76	16.83	0.00	204.83
Modified List Fault Rate	6.81	69.00	0.00	468.42
Demand Zero Fault Rate	2.16	6.24	0.00	162.17
Global Valid Fault Rate	4.64	19.75	0.00	71.06
Wrt In Progress Fault Rate	0.00	0.32	0.00	26.40
System Fault Rate	1.23	0.01	0.00	1.23
Free List Size	4090.00	2527.35	111.00	4106.00
Modified List Size	128.00	341.64	116.00	486.00

PLAYBACK

SUMMARIZING

TEST 9: MPW-LOLIMIT 120 ↑ 320 ; PFCDEFAULT 127 ↓ 64 , 162.  
PFRATL 1 ↓ 0.

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 15:59:51  
To: 14-AUG-1983 16:10:38

	CUR	AVE	MIN	MAX
Page Fault Rate	21.45	96.35	0.00	669.77
Page Read Rate	3.96	9.27	0.00	67.30
Page Read I/O Rate	1.32	3.74	0.00	15.82
Page Write Rate	0.00	8.72	0.00	277.22
Page Write I/O Rate	0.00	0.07	0.00	2.31
Free List Fault Rate	2.97	10.52	0.00	247.90
Modified List Fault Rate	5.61	55.69	0.00	447.52
Demand Zero Fault Rate	0.00	6.06	0.00	153.94
Global Valid Fault Rate	11.22	19.16	0.00	64.24
Wrt In Progress Fault Rate	0.00	0.67	0.00	24.09
System Fault Rate	0.00	0.00	0.00	0.99
Free List Size	3900.00	2240.94	175.00	3900.00
Modified List Size	16.00	238.68	16.00	496.00

PLAYBACK

SUMMARIZING

TEST 10: MPW-LOLIMIT 120 ↑ 320 ; PFCDEFAULT 127 ↓ 64 ,  
PFRATL 1 ↑ 2.

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS  
SUMMARY

From: 14-AUG-1983 16:20:30  
To: 14-AUG-1983 16:31:14

	CUR	AVE	MIN	MAX
Page Fault Rate	11.88	111.47	0.00	799.00
Page Read Rate	0.66	7.81	0.00	42.90
Page Read I/O Rate	0.33	2.77	0.00	11.22
Page Write Rate	0.00	6.89	0.00	197.36
Page Write I/O Rate	0.00	0.05	0.00	1.64
Free List Fault Rate	7.26	15.71	0.00	230.26
Modified List Fault Rate	3.30	67.26	0.00	754.78
Demand Zero Fault Rate	0.00	5.96	0.00	111.84
Global Valid Fault Rate	0.99	19.09	0.00	68.57
Wrt In Progress Fault Rate	0.00	0.20	0.00	17.10
System Fault Rate	0.00	0.04	0.00	6.90
Free List Size	4117.00	2646.81	177.00	4117.00
Modified List Size	97.00	307.13	97.00	490.00

PLAYBACK

SUMMARIZING

TEST 11 . PPW-LOLIMIT 120 ↑ 320 ; PFCDEFAULT 127 ↘ 64 . 163.  
PFRATH 120 ↘ 60 .

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 16:43:48  
To: 14-AUG-1983 16:54:29

	CUR	AVE	MIN	MAX
Page Fault Rate	5.28	112.95	0.00	681.25
Page Read Rate	0.00	7.32	0.00	46.20
Page Read I/O Rate	0.00	2.60	0.00	12.50
Page Write Rate	0.00	7.10	0.00	234.52
Page Write I/O Rate	0.00	0.05	0.00	1.95
Free List Fault Rate	2.31	15.43	0.00	246.38
Modified List Fault Rate	2.97	69.52	0.00	631.25
Demand Zero Fault Rate	0.00	5.91	0.00	162.84
Global Valid Fault Rate	0.00	18.64	0.00	71.47
Wrt In Progress Fault Rate	0.00	0.39	0.00	24.42
System Fault Rate	0.00	0.00	0.00	0.30
Free List Size	4127.00	2636.42	154.00	4132.00
Modified List Size	111.00	316.03	111.00	494.00

PLAYBACK

SUMMARIZING

TEST 12 : PPW-LOLIMIT 120 ↑ 320 ; PFCDEFAULT 127 ↘ 64 ;  
PFRATH 120 ↑ 120

VAX/VMS Monitor Utility  
PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 17:20:21  
To: 14-AUG-1983 17:31:26

	CUR	AVE	MIN	MAX
Page Fault Rate	16.77	114.29	0.00	829.60
Page Read Rate	5.59	8.08	0.00	44.69
Page Read I/O Rate	4.60	2.85	0.00	10.80
Page Write Rate	0.00	6.49	0.00	198.01
Page Write I/O Rate	0.00	0.05	0.00	1.65
Free List Fault Rate	4.93	15.31	0.00	173.68
Modified List Fault Rate	1.64	69.90	0.00	635.19
Demand Zero Fault Rate	2.63	5.98	0.00	165.67
Global Valid Fault Rate	2.96	19.48	0.00	69.40
Wrt In Progress Fault Rate	0.00	0.28	0.00	14.47
System Fault Rate	1.97	0.01	0.00	1.97
Free List Size	4140.00	2596.53	128.00	4143.00
Modified List Size	141.00	319.00	141.00	499.00

PLAYBACK

SUMMARIZING

TEST 13. MPW - LOLIMIT 120 ↑ 320 ; PFCDEFAULT 127 ↘ 64 , 164.  
WSINC 21 ↑ 100.

VAX/VMS Monitor Utility  
 PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 17:38:46  
 To: 14-AUG-1983 17:49:37

	CUR	AVE	MIN	MAX
Page Fault Rate	15.24	93.38	0.00	499.01
Page Read Rate	0.60	7.82	0.00	34.42
Page Read I/O Rate	0.30	2.75	0.00	11.88
Page Write Rate	0.00	1.85	0.00	157.89
Page Write I/O Rate	0.00	0.03	0.00	1.98
Free List Fault Rate	7.62	7.18	0.00	49.17
Modified List Fault Rate	3.96	58.09	0.00	403.30
Demand Zero Fault Rate	0.00	6.04	0.00	249.01
Global Valid Fault Rate	3.35	18.75	0.00	76.79
Wrt In Progress Fault Rate	0.00	0.21	0.00	11.92
System Fault Rate	0.00	0.00	0.00	0.33
Free List Size	4067.00	2459.21	120.00	4067.00
Modified List Size	107.00	299.23	76.00	473.00

PLAYBACK

SUMMARIZING

TEST 14: MPW - LOLIMIT 120 ↑ 320 ; PFCDEFAULT 127 ↘ 64 ,  
WSINC 21 ↑ 150.

VAX/VMS Monitor Utility  
 PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 17:55:38  
 To: 14-AUG-1983 18:06:56

	CUR	AVE	MIN	MAX
Page Fault Rate	21.05	94.29	0.00	486.46
Page Read Rate	1.31	8.13	0.00	39.14
Page Read I/O Rate	0.98	2.76	0.00	10.19
Page Write Rate	0.00	1.22	0.00	99.01
Page Write I/O Rate	0.00	0.01	0.00	1.64
Free List Fault Rate	6.90	5.81	0.00	41.58
Modified List Fault Rate	6.25	60.85	0.00	372.36
Demand Zero Fault Rate	0.00	6.11	0.00	243.56
Global Valid Fault Rate	6.90	18.34	0.00	68.64
Wrt In Progress Fault Rate	0.00	0.08	0.00	14.33
System Fault Rate	0.00	0.00	0.00	0.99
Free List Size	4060.00	2431.28	89.00	4060.00
Modified List Size	97.00	299.44	45.00	475.00

PLAYBACK

SUMMARIZING

TEST 15 MPW-LOLIMIT 120 ↑ 320 , PFCDEFAULT 127 ↘ 64 , 165.  
WSDEC 35 ↘ 10

VAX/VMS Monitor Utilities  
 PAGE MANAGEMENT STATISTICS  
 SUMMARY

From: 14-AUG-1983 18:12:55  
 To: 14-AUG-1983 18:23:50

	CUR	AVE	MIN	MAX
Page Fault Rate	31.47	104.91	0.00	758.74
Page Read Rate	6.22	8.30	0.00	51.64
Page Read I/O Rate	4.91	3.01	0.00	12.87
Page Write Rate	0.00	6.88	0.00	235.29
Page Write I/O Rate	0.00	0.05	0.00	1.96
Free List Fault Rate	11.47	11.89	0.00	211.88
Modified List Fault Rate	6.22	63.52	0.00	498.01
Demand Zero Fault Rate	2.62	6.17	0.00	125.32
Global Valid Fault Rate	6.22	19.50	0.00	66.88
Wrt In Progress Fault Rate	0.00	0.35	0.00	22.44
System Fault Rate	1.96	0.01	0.00	1.96
Free List Size	4094.00	2461.61	145.00	4098.00
Modified List Size	74.00	293.23	73.00	493.00

PLAYBACK

SUMMARIZING

TEST 16 : MPW-LOLIMIT 120 ↑ 320 , PFCDEFAULT 127 ↘ 64 ,  
WSDEC 35 ↑ 100.

VAX/VMS Monitor Utilities  
 PAGE MANAGEMENT STATISTICS  
 SUMMARY

From: 14-AUG-1983 18:42:43  
 To: 14-AUG-1983 18:54:09

	CUR	AVE	MIN	MAX
Page Fault Rate	23.43	132.17	0.00	713.81
Page Read Rate	1.32	7.94	0.00	44.26
Page Read I/O Rate	0.99	2.75	0.00	11.51
Page Write Rate	0.00	7.17	0.00	288.28
Page Write I/O Rate	0.00	0.05	0.00	2.40
Free List Fault Rate	6.93	20.89	0.00	237.23
Modified List Fault Rate	7.59	82.07	0.00	550.88
Demand Zero Fault Rate	0.00	6.12	0.00	215.51
Global Valid Fault Rate	7.92	19.52	0.00	69.44
Wrt In Progress Fault Rate	0.00	0.34	0.00	35.13
System Fault Rate	0.00	0.00	0.00	0.65
Free List Size	4106.00	2626.83	167.00	4113.00
Modified List Size	115.00	321.62	115.00	495.00

PLAYBACK

SUMMARIZING

TEST 17: MPW-LOLIMIT 120 ↑ 380; PFCDEFAULT 127 ↘ 64;  
AWSMIN 50 ↘ 20.

166.

VAX/VMS Monitor Utility  
 PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 21:21:35  
 To: 14-AUG-1983 21:32:53

	CUR	AVE	MIN	MAX
Page Fault Rate	47.03	127.82	0.00	752.84
Page Read Rate	14.80	8.17	0.00	41.83
Page Read I/O Rate	7.56	2.86	0.00	14.70
Page Write Rate	0.00	7.07	0.00	277.22
Page Write I/O Rate	0.00	0.05	0.00	2.31
Free List Fault Rate	14.14	16.88	0.00	247.46
Modified List Fault Rate	12.17	76.06	0.00	517.77
Demand Zero Fault Rate	2.96	6.02	0.00	160.72
Global Valid Fault Rate	9.53	25.06	0.00	78.31
Wrt In Progress Fault Rate	0.00	0.46	0.00	34.65
System Fault Rate	1.97	0.01	0.00	1.97
Free List Size	4102.00	2663.66	190.00	4124.00
Modified List Size	112.00	285.23	104.00	499.00

PLAYBACK

SUMMARIZING

TEST 18: MPW-LOLIMIT 120 ↑ 380; PFCDEFAULT 127 ↘ 64;  
AWSMIN 50 ↑ 100.

VAX/VMS Monitor Utility  
 PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 21:39:09  
 To: 14-AUG-1983 21:50:05

	CUR	AVE	MIN	MAX
Page Fault Rate	41.25	106.02	0.00	828.71
Page Read Rate	6.27	8.08	0.00	38.94
Page Read I/O Rate	4.95	2.79	0.00	12.21
Page Write Rate	0.00	6.40	0.00	197.36
Page Write I/O Rate	0.00	0.05	0.00	1.64
Free List Fault Rate	16.17	11.61	0.00	233.33
Modified List Fault Rate	8.91	66.21	0.00	769.63
Demand Zero Fault Rate	2.64	6.17	0.00	200.66
Global Valid Fault Rate	8.25	18.49	0.00	69.63
Wrt In Progress Fault Rate	0.00	0.30	0.00	18.82
System Fault Rate	1.98	0.01	0.00	1.98
Free List Size	4139.00	2612.85	148.00	4140.00
Modified List Size	102.00	289.77	102.00	495.00

PLAYBACK

SUMMARIZING

TEST 19 MPW - LOLIMIT 120 ↑ 380 ; PFCDEFAULT 127 ↘ 64, 167.  
AWSTIME 20 ↑ 40.

VAX/VMS Monitor Utility  
 PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 21:55:32  
 To: 14-AUG-1983 22:06:27

	CUR	AVE	MIN	MAX
Page Fault Rate	28.71	125.22	0.00	636.18
Page Read Rate	5.28	8.68	0.00	47.52
Page Read I/O Rate	3.96	3.09	0.00	16.11
Page Write Rate	0.00	15.02	0.00	268.37
Page Write I/O Rate	0.00	0.12	0.00	2.23
Free List Fault Rate	8.25	20.63	0.00	267.10
Modified List Fault Rate	7.59	75.19	0.00	553.79
Demand Zero Fault Rate	2.31	6.15	0.00	136.51
Global Valid Fault Rate	6.27	18.59	0.00	84.48
Wrt In Progress Fault Rate	0.00	1.09	0.00	28.71
System Fault Rate	1.32	0.01	0.00	1.32
Free List Size	4101.00	2600.51	507.00	4101.00
Modified List Size	84.00	285.88	70.00	499.00

PLAYBACK

SUMMARIZING

TEST 20 MPW - LOLIMIT 120 ↑ 380 ; PFCDEFAULT 127 ↘ 64,  
AWSTIME 20 ↑ 60.

VAX/VMS Monitor Utility  
 PAGE MANAGEMENT STATISTICS

SUMMARY From: 14-AUG-1983 22:10:54  
 To: 14-AUG-1983 22:22:10

	CUR	AVE	MIN	MAX
Page Fault Rate	9.57	146.00	0.00	887.54
Page Read Rate	0.66	8.95	0.00	52.96
Page Read I/O Rate	0.33	3.43	0.00	15.18
Page Write Rate	0.00	31.45	0.00	423.07
Page Write I/O Rate	0.00	0.26	0.00	3.52
Free List Fault Rate	4.95	34.19	0.00	404.29
Modified List Fault Rate	2.97	79.25	0.00	718.15
Demand Zero Fault Rate	0.00	6.40	0.00	214.14
Global Valid Fault Rate	1.32	19.33	0.00	71.94
Wrt In Progress Fault Rate	0.00	2.92	0.00	57.87
System Fault Rate	0.00	0.00	0.00	0.32
Free List Size	4031.00	2483.07	537.00	4035.00
Modified List Size	78.00	302.89	78.00	499.00

PLAYBACK

SUMMARIZING

C. TABLEAUX SYNTHETIQUES DES RESULTATS

1. Variation des paramètres

=====

La variation des paramètres a porté sur :

MPW-WRTCLUSTER

nombre de pages de la liste des pages modifiées recopiées  
en une opération I/O

MPW-HILIMIT

taille maximale de la liste des pages modifiées

MPW-LOLIMIT

taille minimale de la liste des pages modifiées

PFCDEFAULT

nombre de pages lues sur disque lors d'un défaut de page  
en une seule opération I/O

QUANTUM

- . temps processeur maximum qu'un processus peut utiliser  
en une seule fois
- . temps processeur minimum qu'un processus exécutable  
doit recevoir avant d'être éjecté du balance-set

PFRATL

nombre de défauts de page sous lequel un working-set  
est automatiquement diminué

## PFRATH

Nombre de défauts de page au-dessus duquel un working-set est automatiquement agrandi

## WSINC

nombre de pages dont est automatiquement augmenté un working-set lors d'un réajustement croissant

## WSDEC

nombre de pages dont est automatiquement diminué un working-set lors d'un réajustement décroissant

## AWSMIN

nombre minimum de pages que peut atteindre un working-set par les réajustements automatiques

## AWSTIME

temps processeur minimum pour calculer le taux de défauts de page d'un working-set

## 2. Résultats des tests =====

Les données fournies par les tests sont :

### PAGE FAULT RATE

le taux global de défaut de page pour tous les working-sets  
(nb défauts de page/sec)

### PAGE READ RATE

le taux de pages lues sur disque suite à un défaut de page  
(nb pages lues/sec)

### PAGE READ I/O RATE

le taux d'opérations I/O de lecture sur disque suite  
à un défaut de page  
(nb d'opérations I/O sec)

### PAGE WRITE RATE

le taux de recopies des pages de la liste des pages  
modifiées dans le paging file  
(nb pages écrites/sec)

### PAGE WRITE I/O RATE

le taux d'opérations I/O d'écriture dans le paging file  
des pages de la liste des pages modifiées  
(nb opérations I/O sec)

### FREE LIST FAULT RATE

le taux de pages lues dans la liste des pages libres  
suite à un défaut de page  
(nb pages lues/sec)

**MODIFIED LIST FAULT RATE**

le taux de pages lues dans la liste des pages modifiées  
suite à un défaut de page  
(nb de pages lues/sec)

**DEMAND ZERO FAULT RATE**

le taux de pages réinitialisées allouées aux working-sets  
suite à un défaut de page  
(nb pages allouées/sec)

**GLOBAL VALID FAULT RATE**

le taux de défauts de page pour des pages renseignées  
dans la table des pages globales du système  
(nb défauts de page/sec)

**WRT IN PROGRESS FAULT RATE**

le taux de pages demandées lors d'un défaut de page et  
qui sont déjà dans le processus de réécriture sur disque  
(nb pages demandées/sec)

**SYSTEM FAULT RATE**

le taux de défauts de page pour des pages du système  
(nb défauts de page/sec)

**FREE LIST SIZE**

nombre de pages dans la liste des pages libres

**MODIFIED LIST SIZE**

nombre de pages dans la liste des pages modifiées

PARAMETRES	TESTS																			
	Test 1 (sit. initiale)	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16	Test 17	Test 18	Test 19	Test 20
MPW - WRTCLUSTER	120	120	96	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
MPW - HILIMIT	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
MPW - LOLIMIT	120	380	32	120	120	120	120	380	380	380	380	380	380	380	380	380	380	380	380	380
PFCDEFAULT	127	127	127	32	64	127	127	64	64	64	64	64	64	64	64	64	64	64	64	64
QUANTUM	20	20	20	20	20	40	60	20	20	20	20	20	20	20	20	20	20	20	20	20
PFRATL	1	1	1	1	1	1	1	1	0	2	1	1	1	1	1	1	1	1	1	1
PFRATH	120	120	120	120	120	120	120	120	120	120	60	180	120	120	120	120	120	120	120	120
WSINC	21	21	21	21	21	21	21	21	21	21	21	21	100	150	21	21	21	21	21	21
WSDEC	35	35	35	35	35	35	35	35	35	35	35	35	35	35	10	100	35	35	35	35
AWSMIN	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	20	100	50	50
AWSTIME	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	40	60

- indique une augmentation du paramètre par rapport à la situation initiale  
 ⬡ indique une diminution du paramètre par rapport à la situation initiale

TESTS  MESURES	Test 1 (Sit. initiale)		Test 2		Test 3		Test 4		Test 5		Test 6		Test 7		Test 8		Test 9		Test 10	
	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX
	Page Fault Rate	114.2	836.6	98.9	862.8	108.5	610.6	112.9	709.3	108.8	799.3	127.6	780.9	130.9	1265.7	115.7	588.4	96.4	669.8	111.5
Page Read Rate	7.99	57.51	7.29	39.80	8.06	43.17	7.55	43.23	7.31	45.72	6.74	43.56	7.23	53.13	8.69	54.40	9.27	67.30	7.81	42.90
Page Read I/O Rate	2.88	15.68	2.57	12.13	2.94	13.15	2.75	15.84	2.68	17.10	2.44	12.69	2.89	17.49	3.09	15.14	3.74	15.82	2.77	11.22
Page Write Rate	9.94	394.73	1.33	118.03	13.58	405.61	11.15	354.09	9.50	393.44	17.83	458.59	24.05	554.45	8.46	237.62	8.72	277.22	6.89	197.36
Page Write I/O Rate	0.08	3.28	0.01	0.98	0.14	4.29	0.09	2.95	0.07	3.27	0.14	3.82	0.20	4.62	0.07	2.23	0.07	2.31	0.05	1.64
Free List Fault Rate	17.72	317.98	10.58	124.55	20.10	316.50	18.61	341.11	16.89	367.62	24.90	371.94	29.56	467.98	16.83	204.83	10.52	247.90	15.71	230.26
Modified List Fault Rate	67.40	758.41	62.25	830.26	58.56	401.31	65.75	542.12	64.09	781.90	74.67	692.30	71.85	1064.68	69.00	468.42	55.69	447.52	67.26	754.78
Demand Zero Fault Rate	5.80	252.47	4.17	22.69	5.87	131.90	5.85	176.92	5.73	226.79	5.77	161.09	4.15	28.94	6.24	162.17	6.06	153.94	5.96	111.84
Global Valid Fault Rate	19.19	83.12	18.75	76.31	19.36	63.96	18.67	74.67	18.35	72.03	17.92	86.51	18.53	86.18	19.75	71.06	19.16	64.24	19.09	68.57
Wrt In Progress Fault Rate	0.79	60.85	0.06	11.47	1.24	44.82	0.76	29.93	0.65	33.33	1.42	42.01	3.31	115.78	0.32	26.40	0.67	24.09	0.20	17.10
System Fault Rate	0.00	0.92	0.00	0.33	0.01	1.97	0.00	0.98	0.01	1.78	0.00	0.32	0.00	0.33	0.01	1.23	0.00	0.99	0.04	6.90
Free List Size	2623	4120	2630	4148	2628	4139	2618	4095	2654	4139	2598	4089	2618	4073	2527	4106	2241	3900	2647	4117
Modified List Size	280	495	330	491	306	495	261	493	269	489	301	485	321	499	342	486	239	496	307	490

Tableau synthétique des résultats des différents tests

(nb moyen/sec, nb max/période)

TESTS MESURES	Test 1 (Sit. initiale)		Test 11		Test 12		Test 13		Test 14		Test 15		Test 16		Test 17		Test 18		Test 19		Test 20	
	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX	MOYENNE	MAX
	Page Fault Rate	114.2	836.6	112.9	681.3	114.3	829.6	93.4	499.0	94.3	486.5	104.9	758.7	132.2	713.8	127.8	752.8	106.0	828.7	125.2	636.2	146.0
Page Read Rate	7.99	57.51	732	46.20	8.08	44.69	7.81	34.42	8.13	39.14	8.30	51.64	7.94	44.26	8.17	41.83	8.08	38.94	8.68	47.52	8.95	52.96
Page Read I/O Rate	2.88	15.68	2.60	12.50	2.85	10.80	2.75	11.88	2.76	10.19	3.01	12.87	2.75	11.51	2.86	14.70	2.79	12.21	3.09	16.11	3.43	15.18
Page Write Rate	9.94	394.73	7.10	234.52	6.49	198.01	1.85	157.89	1.22	99.01	6.88	235.29	7.17	288.28	7.07	277.22	6.40	197.36	15.02	268.37	31.45	423.07
Page Write I/O Rate	0.08	3.28	0.05	1.95	0.05	1.65	0.03	1.98	0.01	1.64	0.05	1.96	0.05	2.40	0.05	2.31	0.05	1.64	0.12	2.23	0.26	3.52
Free List Fault Rate	17.72	317.98	15.43	246.38	15.31	173.68	7.18	49.17	5.81	41.58	11.89	211.88	20.89	237.23	16.88	247.46	11.61	233.33	20.63	267.10	34.19	404.29
Modified List Fault Rate	67.40	758.41	69.52	631.25	69.90	635.19	58.09	403.30	60.85	372.36	63.52	498.01	82.07	550.88	76.06	517.77	66.21	762.63	75.19	553.79	79.25	718.15
Demand Zero Fault Rate	5.80	252.47	5.91	162.84	5.98	165.67	6.04	249.01	6.11	243.56	6.17	125.32	6.12	215.51	6.02	160.72	6.17	200.66	6.15	136.51	6.40	214.14
Global Valid Fault Rate	19.19	83.12	18.64	71.47	19.48	69.40	18.75	76.79	18.34	68.64	19.50	66.88	19.52	69.44	25.06	78.31	18.49	69.63	18.59	84.48	19.33	71.94
Wrt In Progress Fault Rate	0.79	60.85	0.39	24.42	0.28	14.47	0.21	11.92	0.08	14.33	0.35	22.44	0.34	35.13	0.46	34.65	0.30	18.82	1.09	28.71	2.92	57.87
System Fault Rate	0.00	0.92	0.00	0.30	0.01	1.97	0.00	0.33	0.00	0.99	0.01	1.96	0.00	0.65	0.01	1.97	0.01	1.98	0.01	1.32	0.00	0.32
Free List Size	2623	4120	2636	4132	2597	4143	2459	4067	2431	4060	2462	4098	2627	4113	2664	4124	2613	4140	2600	4101	2483	4035
Modified List Size	280	495	316	494	319	499	299	473	299	475	293	493	322	495	285	499	290	495	286	499	303	499

Tableau synthétique des résultats des différents tests (suite)

TESTS MESURES	Test 1 (sit. initiale)	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16	Test 17	Test 18	Test 19	Test 20
	Page Fault Rate		-13.4	- 5.0	- 1.1	- 4.7	+11.7	+14.6	+ 1.3	-15.6	- 2.4	- 1.1	+ 0.1	-18.2	-17.4	- 8.1	+15.8	+11.9	- 7.2	+ 9.6
Page Read Rate		- 8.8	+ 0.9	- 5.5	- 8.5	-15.6	- 9.5	+ 8.8	+16.0	- 2.3	- 8.4	+ 1.1	- 2.1	+ 1.8	+ 3.9	- 0.6	+ 2.3	+ 1.1	+ 8.6	+12.0
Page Read I/O Rate		-10.8	+ 2.1	- 4.5	- 6.9	-15.3	+ 0.4	+ 7.3	+29.9	- 3.8	- 9.7	- 1.0	- 4.5	- 4.2	+ 4.5	- 4.5	- 0.7	- 3.1	+ 7.3	+19.1
Page Write Rate		-86.6	+36.6	+12.2	- 4.4	+79.4	+142	-14.9	-12.3	-30.7	-28.6	-34.7	-81.4	-87.7	-30.8	-27.9	-28.9	-35.6	+51.1	+216
Page Write I/O Rate		-87.5	+75.0	+12.5	-12.5	+75.0	+150	-12.5	-12.5	-37.5	-37.5	-37.5	-62.5	-87.5	-37.5	-37.5	-37.5	-37.5	+50.0	+225
Free List Fault Rate		-40.3	+13.4	+ 5.0	- 4.7	+40.5	+66.8	- 5.0	-40.6	-11.3	-12.9	-13.6	-59.5	-67.2	-32.9	+17.9	- 4.7	-34.5	+16.4	+93.0
Modified List Fault Rate		- 7.6	-13.1	- 2.5	- 4.9	+10.8	+ 6.6	+ 2.4	-17.4	- 0.2	+ 3.2	+ 3.7	-13.8	- 9.7	- 5.8	+21.8	+12.9	- 1.8	+11.6	+17.6
Demand Zero Fault Rate		-28.1	+ 1.2	+ 0.9	- 1.2	- 0.5	-28.5	+ 7.6	+ 4.5	+ 2.8	+ 1.9	+ 3.1	+ 4.1	+ 5.3	+ 6.4	+ 5.5	+ 3.8	+ 6.4	+ 6.0	+10.3
Global Valid Fault Rate		- 2.3	+ 0.9	- 2.7	- 4.4	- 6.6	- 3.4	+ 2.9	- 0.2	- 0.5	- 2.9	+ 1.5	- 2.3	- 4.4	+ 1.6	+ 1.7	+30.6	- 3.7	- 3.1	+ 0.7
Wrt In Progress Fault Rate		-92.4	+57.0	- 3.8	-17.7	+79.8	+319	-59.5	-15.2	-74.7	-50.6	-64.6	-73.4	-89.9	-55.7	-57.0	-41.8	-62.0	+38.0	+270
System Fault Rate (nb moyen/ sec)		0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.04	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00
Free List Size		+ 0.3	+ 0.2	- 0.2	+ 1.2	- 1.0	- 0.2	- 3.7	-14.6	+ 0.9	+ 0.5	- 1.0	- 6.3	- 7.3	- 6.1	+ 0.2	+ 1.6	- 0.4	- 0.9	- 5.3
Modified List Size		+17.9	+ 9.3	- 6.8	- 3.9	+ 7.5	+14.6	+21.1	-14.6	+ 9.6	+12.9	+13.9	+ 6.8	+ 6.8	+ 4.6	+15.0	+ 1.8	+ 3.6	+ 2.1	+ 8.2

Tableau synthétique des variations des mesures moyennes (en pourcentage)

DEFAUTS DE PAGE	TESTS																				
	Test 1	sit. initiale	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16	Test 17	Test 18	Test 19	Test 20
Page Fault Rate (nb moyen/ sec)	114.24	100	98.85	108.52	112.87	108.82	127.56	130.89	115.72	96.35	111.47	112.95	114.29	93.38	94.29	104.91	132.17	127.82	106.02	125.22	146.00
Page Read Rate	7.99	7.0	7.4	7.4	6.7	6.7	5.3	5.5	7.5	9.6	7.0	6.5	7.1	8.4	8.6	7.9	6.0	6.4	7.6	6.9	6.1
Free List Fault Rate	17.72	15.5	10.7	18.5	16.5	15.5	19.5	22.6	14.5	10.9	14.1	13.7	13.4	7.7	6.2	11.3	15.8	13.2	11.0	16.5	23.4
Modified List Fault Rate	67.40	59.0	63.0	54.0	58.3	58.9	58.5	54.9	59.6	57.8	60.3	61.6	61.2	62.2	64.5	60.6	62.1	59.5	62.5	60.1	54.3
Demand Zero Fault Rate	5.80	5.1	4.2	5.4	5.2	5.3	4.5	3.2	5.4	6.3	5.4	5.2	5.2	6.5	6.5	5.9	4.6	4.7	5.8	4.9	4.4
Global Valid Fault Rate	19.19	16.8	19.0	17.8	16.5	16.9	14.1	14.2	17.1	19.9	12.1	16.5	17.0	20.1	19.5	18.6	14.8	19.6	17.4	14.9	13.2
Wrt In Progress Fault Rate	0.79	0.7	0.1	1.1	0.7	0.6	1.1	2.5	0.3	0.7	0.2	0.4	0.2	0.2	0.1	0.3	0.3	0.4	0.3	0.9	2.0
System Fault rate (nb moyen/sec)	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.04	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00

Tableau synthétique de la répartition du nombre moyen de défauts de page (en pourcentage)

## BIBLIOGRAPHIE

1. CROCUS  
"Système d'exploitation des ordinateurs : principes de conception"  
Collection DUNOD : Informatique - Bordas, Paris, 1975
  
2. DE MARNEFFE  
"Le système VAX / VMS"  
Notes de cours donné à la faculté polytechnique de Mons
  
3. DIGITAL EQUIPMENT CORPORATION  
"VAX 11/780 : Technical summary"  
Magnard, Massachussetts, 1978
  
4. DIGITAL EQUIPMENT CORPORATION  
"VAX / VMS System Manager and Operations Guide"  
"VAX / VMS UETP User's Guide"  
"VAX / VMS Utilities Reference Manual"  
Software version : VAX / VMS Version 3.0  
Magnard, Massachussetts, May 1982
  
5. BELL C.G.  
"Computer Engeneering"  
Digital Press, Bedford, Massachussetts, 1979
  
6. HODGES L.F. & STEWERT W.J.  
"Workload characterization and performance evaluation in a research environment"  
Performance Evaluation Review - ACM Symetrics 1982, Vol.11 n°4

7. INFOTECH STATE OF THE ART REPORT  
"System Tuning"  
Infotech International, 1977
  
8. LEVY H.M. & ECKHOUSE R.H.  
"Computer Programming and Architecture : the VAX-11"  
Digital Press, Bedford, Massachusetts, 1980
  
9. LOZOWSKA E.D. & KELSEY J.M.  
"Notes on tuning VAX / VMS"  
Department of Computer Science, University of Washington  
December 1978 - Technical Report NO. 78-12-01
  
10. LOZOWSKA E.D.  
"The benchmarking, tuning and analytic modeling of VAX / VMS"  
Conference on Simulation Measurement and Modeling of  
Computer Systems, 1979  
Department of Computer Science, University of Washington  
April 1979 - Technical Report, NO. 79-04-01
  
11. MANSFORD E. & DRUMMOND J.R.  
"Evaluation and Measurement techniques for Digital Computer  
Systems"  
Prentice-Hall, Series in Automatic Computation, 1973
  
12. "Proceedings of the Digital Equipment Computer Users Society"  
DECUS, Europe, 1982.
  
13. RAMAEKERS J.  
"Efficacité des systèmes informatiques"  
Institut d'Informatique, FNNDP, Namur, 1976

14. SABOUREAU J.P.  
"Contrôle des performances de l'exploitation d'un centre informatique"  
Editions Hommes et Techniques, 1981
  
15. SUP'AERO ECOLE NATIONALE SUPERIEURE DE L'AERONAUTIQUE  
DE L'ESPACE  
"Mesures des systèmes informatiques"  
Cépadues éditions, Toulouse 1978
  
16. DIGITAL EQUIPMENT CORPORATION  
VAX Architecture Handbook  
VAX Hardware Handbook  
VAX Software Handbook  
Digital 1982-1983, Maynard, Massachussetts
  
17. WILKES M.V.  
"The Dynamics of Paging"  
Computer Journal, 1973, Vol. 16, p 1
  
18. WILKES M.V.  
"Time-sharing Computer Systems"  
Third edition 1975, Mac Donald and Jane's, London