



Amélioration fonctionnelles dans un système d'opérations à trois niveaux de hiérarchie

Jean-Pierre Dupuy

► To cite this version:

Jean-Pierre Dupuy. Amélioration fonctionnelles dans un système d'opérations à trois niveaux de hiérarchie. Modélisation et simulation. 1974. dumas-00295428

HAL Id: dumas-00295428

<https://dumas.ccsd.cnrs.fr/dumas-00295428>

Submitted on 11 Jul 2008

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

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

N° d'ordre

TV 503

THESE

présentée au

CENTRE UNIVERSITAIRE

D'EDUCATION ET DE FORMATION DES ADULTES DE GRENOBLE

pour obtenir le titre

d'INGENIEUR du CONSERVATOIRE NATIONAL des ARTS et METIERS

par

Jean-Pierre DUPUY

— o —

**Améliorations fonctionnelles dans
un système d'opérations
à trois niveaux de hiérarchie**

— o —

Thèse soutenue le 22 mars 1974 devant la commission d'examen

Président

Monsieur L. Bolliet

Président adjoint

Monsieur P. Namian

Examineurs

Messieurs C. Hans

M. Bellot

Président : Monsieur Michel SOUTIF
Vice-Président : Monsieur Gabriel CAU

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Georges	Clinique des maladies infectieuses
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BENOIT Jean	Radioélectricité
	BERNARD Alain	Mathématiques Pures
	BESSON Jean	Electrochimie
	BEZES Henri	Chirurgie générale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BONNIER Etienne	Electrochimie Electrometallurgie
	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques Appliquées
	BRAVARD Yves	Géographie
	BRISSENEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	CABANAC Jean	Pathologie chirurgicale
	CABANEL Jean	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et Toxicologie
	CAQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Oto-Rhino-Laryngologie
	CHATEAU Robert	Thérapeutique
	CHENE Marcel	Chimie papetière
	COEUR André	Pharmacie chimique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique

Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	FELICI Noël	Electrostatique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques Pures
	GALVANI Octave	Mathématiques Pures
	GASTINEL Noël	Analyse numérique
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques Pures
	GIRAUD Pierre	Géologie
	KLEIN Joseph	Mathématiques Pures
Mme	KOFLER Lucie	Botanique et Physiologie végétale
MM.	KOSZUL Jean-Louis	Mathématiques Pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LOUP Jean	Géographie
Mlle	LUTZ Elisabeth	Mathématiques Pures
MM.	MALGRANGE Bernard	Mathématiques Pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MASSEPORT Jean	Géographie
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et Pétrographie
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAUTHENET René	Electrotechnique
	PAYAN Jean-Jacques	Mathématiques Pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET René	Servomécanismes
	PILLET Emile	Physique industrielle
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REULOS René	Physique industrielle
	RINALDI Renaud	Physique
	ROGET Jean	Clinique de pédiatrie et de puériculture
	SANTON Lucien	Mécanique
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SILBERT Robert	Mécanique des fluides
	SOUTIF Michel	Physique générale

MM.	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLAND François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
M.	VERAIN André	Physique
Mme	VEYRET Germaine	Géographie
MM.	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	BULLEMER Bernhard	Physique
	HANO JUN-ICHI	Mathématiques Pures
	STEPHENS Michaël	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

MM.	BEAUDOING André	Pédiatrie
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BONNETAIN Lucien	Chimie minérale
Mme	BONNIER Jane	Chimie générale
MM.	CARLIER Georges	Biologie végétale
	COHEN Joseph	Electrotechnique
	COUMES André	Radioélectricité
	DEPASSEL Roger	Mécanique des fluides
	DEPORTES Charles	Chimie minérale
	GAUTHIER Yves	Sciences biologiques
	GAVEND Michel	Pharmacologie
	GERMAIN Jean-Pierre	Mécanique
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	HACQUES Gérard	Calcul numérique
	JANIN Bernard	Géographie
Mme	KAHANE Josette	Physique
MM.	MULLER Jean-Michel	Thérapeutique
	PERRIAUX Jean-Jacques	Géologie et Minéralogie
	POULOUJADOFF Michel	Electrotechnique
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	ROBERT André	Chimie papetière
	DE ROUGEMONT Jacques	Neurochirurgie
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIBILLE Robert	Construction mécanique
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Dermatologie
	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Yves	Chimie
	BEGUIN Claude	Chimie organique
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BILLET Jean	Géographie
	BLIMAN Samuel	Electronique (EIE)
	BLOCH Daniel	Electrotechnique
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BOUCHET Yves	Anatomie
	BOUVARD Maurice	Mécanique des fluides
	BRODEAU François	Mathématiques (IUT B)
	BRUGEL Lucien	Energétique
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHIAVERINA Jean	Biologie appliquée (EFP)
	CHIBON Pierre	Biologie animale
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	DURAND Francis	Métallurgie
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROULADE Joseph	Biochimie médicale
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Médecine préventive
	IDELMAN Simon	Physiologie animale
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	JOLY Jean-René	Mathématiques Pures
	JOUBERT Jean-Claude	Physique du solide
	JULLIEN Pierre	Mathématiques Pures
	KAHANE André	Physique générale
	KUHN Gérard	Physique
	LACOUME Jean-Louis	Physique
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LANCIA Roland	Physique atomique
	LE JUNTER Noël	Electronique
	LEROY Philippe	Mathématiques
	LOISEAUX Jean-Marie	Physique nucléaire
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LJU DUC Cuong	Chimie organique
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et Médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)

MM.	MAYNARD Roger	Physique du solide
	MICHOULIER Jean	Physique (IUT A)
	MICOUD Max	Maladies infectieuses
	MOREAU René	Hydraulique (INP)
	NEGRE Robert	Mécanique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PELMONT Jean	Physiologie animale
	PERRET Jean	Neurologie
	PERRIN Louis	Pathologie expérimentale
	PFISTER Jean-Claude	Physique du solide
	PHÉLIP Xavier	Rhumatologie
Mlle	RIERY Yvette	Biologie animale
MM.	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RENAUD Maurice	Chimie
	RICHARD Lucien	Botanique
Mme	RINAUDO Marquerite	Chimie macromoléculaire
MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VAN CUTSEM Bernard	Mathématiques appliquées
	VEILLON Gérard	Mathématiques appliquées (INP)
	VIALON Pierre	Géologie
	VOOG Robert	Médecine interne
	VROUSSOS Constantin	Radiologie
	ZADWORNÝ François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM.	BOUDOURIS Georges	Radioélectricité
	CHEEKE John	Thermodynamique
	GOLDSCHMIDT Hubert	Mathématiques
	SIDNEY STUARD	Mathématiques Pures
	YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

Mme	BERIEL Hélène	Physiologie
Mme	RENAUDET Jacqueline	Microbiologie

Je tiens à remercier

Monsieur le Professeur L. BOLLIET qui a bien voulu me faire l'honneur de présider le Jury,

Monsieur le Professeur P. NAMIAN qui a accepté d'être Président Adjoint du Jury,

Monsieur C. HANS qui fut à l'origine de cette thèse et auprès de qui j'ai toujours trouvé conseils et encouragements,

Monsieur M. BELLOT qui a rendu possible cette thèse et qui a suivi avec le plus grand intérêt ce travail.

Je voudrais aussi remercier

MM. M. PELTIER et A. AUROUX qui, en tant que Directeurs du Centre Scientifique IBM de Grenoble m'ont procuré l'environnement idéal pour mener à bien ces travaux,

Monsieur J. RODRIGUEZ-ROSELL qui a orienté les diverses phases de cette étude, et qui a participé à sa réalisation,

Tous mes collègues du Centre Scientifique IBM et du CICG, notamment Madame L. SIRET, Messieurs J. BELLINO, M. BERTHAUD, J. GUILLOU, M. REY, J.P. LE HEIGET, X. DE LAMBERTERIE qui n'ont pas ménagé leurs suggestions et leurs critiques,

Madame G. EYRAUD qui a eu la lourde charge de frapper ce texte,

Je voudrais enfin remercier le Service Tirage du CICG qui en a assuré l'impression.

PRESENTATION

INTRODUCTION	1
--------------	---

<u>CHAPITRE I</u> : EXPLORATION ET EVALUATION DE L'HYPERVEISEUR	6
---	---

1. INTRODUCTION	7
-----------------	---

2. LE SYSTEME A MESURER : CP67	9
--------------------------------	---

2.1. Gestion de la mémoire	10
----------------------------	----

2.1.1. La mémoire virtuelle	11
-----------------------------	----

2.1.2. Traduction dynamique des adresses	11
--	----

2.1.3. Mécanismes de pagination dans CP	13
---	----

2.2. Gestion de l'unité centrale	15
----------------------------------	----

2.2.1. Classification des machines virtuelles	16
---	----

2.2.2. Principaux paramètres du contrôleur	18
--	----

2.2.3. Choix d'une machine virtuelle	19
--------------------------------------	----

2.3. Premières variables à mesurer	21
------------------------------------	----

2.3.1. Variables venant du contrôleur	22
---------------------------------------	----

2.3.2. Variables venant de la pagination	23
--	----

2.4. Configuration de l'ordinateur	24
------------------------------------	----

3. LE SYSTEME DE MESURE	26
-------------------------	----

3.1. Objectifs du système de mesure	27
-------------------------------------	----

3.1.1. Choix des variables	29
----------------------------	----

3.1.2. Classification des variables	29
-------------------------------------	----

3.2. Relevé des mesures	32
-------------------------	----

3.2.1. L'instruction "DIAGNOSE"	33
---------------------------------	----

3.2.2. La machine virtuelle MESURE	34
------------------------------------	----

3.3. Exploitation des mesures	35
-------------------------------	----

3.3.1. Niveau moyen de multiprogrammation	36
---	----

3.3.2. Présentation et discussion de certaines courbes	37
--	----

<u>CHAPITRE II</u> : LE MODELE THEORIQUE DES WORKING SET	41
1. INTRODUCTION	42
2. COMPORTEMENT DES PROGRAMMES DANS UN CONTEXTE DE PAGINATION	43
3. DEFINITION DU WORKING SET	45
4. ADAPTATION DU MODELE DE WORKING SET A L'HYPERVISEUR	46
<u>CHAPITRE III</u> : DESCRIPTION ET EVALUATION DES CHANGEMENTS APPORTES A L'HYPERVISEUR	48
1. PRINCIPES DE FONCTIONNEMENT DE LA PAGINATION	49
2. PRINCIPES DE FONCTIONNEMENT DU CONTROLEUR	50
2.1. Organisation des ensembles de machines virtuelles	53
2.2. Principes de migration des machines virtuelles	54
2.3. Critères de choix du contrôleur	57
3. DESCRIPTION DE LA MISE EN OEUVRE DES NOUVEAUX ALGORITHMES	59
3.1. Calcul d'un working set, organisation de la mémoire centrale	59
3.2. Traitement d'une interruption pour page manquante	61
3.3. Le contrôleur	63
3.3.1. Détermination de l'état d'une machine virtuelle	64
3.3.2. Choix d'une machine virtuelle	64
3.3.3. Remarques concernant le contrôleur	66
4. EVALUATION DES MODIFICATIONS	68
4.1. Etude des files d'attente	69
4.2. Etude de la pagination	73
4.3. Etude de l'utilisation de l'unité centrale	75
CONCLUSION	77
BIBLIOGRAPHIE	79

PRESENTATION

Dans ce mémoire, nous nous proposons de décrire les travaux que nous avons réalisés dans le but d'améliorer les performances d'un système générateur de machines virtuelles: CP-67.

Ce système crée des machines virtuelles qui sont la réplique d'ordinateurs existants , et répartit les ressources réelles entre ces différentes machines virtuelles. La ressource "unité de traitement" est gérée en utilisant une méthode de multiprogrammation et de temps partagé ; la ressource mémoire centrale est gérée en utilisant le dispositif de traduction dynamique des adresses de l'ordinateur IBM 360/67 qui conduit à la mise en oeuvre d'une technique de pagination.

Dans le premier chapitre, nous décrivons ces deux parties du système CP-67 puis nous présentons un outil de mesure par programme que nous avons réalisé pour étudier le comportement du système pendant son exploitation normale. Nous montrerons les principaux résultats, obtenus à l'aide de cet outil, qui mettent en évidence une faiblesse de l'hyperviseur due à l'inexistence d'un couplage entre la gestion de la mémoire centrale et la gestion de l'unité de traitement.

Le second chapitre présente les notions essentielles du modèle théorique des "working set", et indique comment nous avons décidé d'utiliser ces notions pour combattre la faiblesse citée précédemment.

Dans le troisième chapitre, nous décrivons les améliorations que nous avons apportées à CP-67 (dans les parties qui gèrent la mémoire centrale et l'unité de traitement). Nous présentons enfin une évaluation chiffrée des résultats obtenus grâce à ce travail.

INTRODUCTION

Au cours de l'année 1969, nous avons pu observer que le système CP-67 qui était le système de contrôle de l'ordinateur du Centre Interuniversitaire de Calcul de Grenoble, pendant une partie de l'exploitation, présentait un fonctionnement dégradé. De façon aléatoire, au cours d'une session, les utilisateurs constataient une baisse importante du rendement de l'ordinateur.

A cette période, nous nous intéressions au comportement des systèmes générateurs de machines virtuelles, et il nous a semblé très important d'étudier ces dégradations pour en connaître les causes et pour tenter d'y remédier ; sous la direction de Claude Hans, Juan Rodriguez-Rosell et moi-même avons donc entrepris cette étude.

INTRODUCTION

Ce travail comporte deux aspects distincts mais très liés :

- observation du système pendant son fonctionnement normal afin de trouver les causes de la baisse de performance
- recherche et réalisation d'une solution afin d'éliminer les causes de la dégradation.

La notion de performance est ambiguë car les critères qui permettent de l'évaluer sont souvent différents, et parfois même contradictoires. Par exemple, l'utilisateur d'un système informatique évaluera cette performance en mesurant le temps qui s'est écoulé entre l'instant où il a soumis son travail et l'instant où il en recueille les résultats ; le chef de l'exploitation jugera, lui, de la performance en comptant le nombre de travaux exécutés pendant une journée. Il existe différentes techniques qui permettent d'évaluer le comportement d'un système, nous en citerons trois qui sont les plus fréquemment utilisées.

- Evaluation par unité câblée

A l'aide d'une telle unité, indépendante de l'ordinateur lui-même, on peut prélever des signaux qui indiquent le changement d'état d'un composant électronique, ensuite ces signaux sont interprétés et permettent soit de compter le

INTRODUCTION

nombre d'occurrences d'un évènement, soit de mesurer le temps séparant deux évènements.

- Evaluation par modèle

Pour utiliser cette méthode, il faut construire un modèle qui simule à la fois l'ordinateur et le système qui le contrôle, ensuite en lui fournissant un lot de transactions à traiter on peut évaluer les performances.

- Evaluation par examen programmé

Cette technique consiste à modifier le système de contrôle de l'ordinateur pour y introduire des variables qui serviront à enregistrer des valeurs attachées à des évènements, par la suite pendant le fonctionnement du système, il faudra prélever ces valeurs et les analyser.

Après avoir expérimenté une observation par unité câblée, nous avons entrepris la réalisation d'un système de mesure par programme.

Les résultats des mesures ont fait apparaître que la cause principale des baisses de performance était une absence de liaison entre la gestion de l'unité de traitement : le contrôleur ("Dispatcher"), et la gestion de la mémoire réelle : la pagination.

INTRODUCTION

Nous avons alors cherché comment nous pouvions introduire cette liaison entre les deux parties de CP-67 ; après avoir étudié différentes solutions , nous avons décidé de nous inspirer du modèle théorique des "working set" proposé par P.J. Denning (8).

Juan Rodriguez-Rosell a étudié alors ce modèle, et le comportement des programmes dans un tel environnement ; les conclusions de cette étude nous ont permis, entre autres, de fixer les valeurs de certains paramètres des algorithmes correspondants (15).

Pour la réalisation des nouveaux programmes nous avons adopté des règles précises pour définir les liaisons entre le contrôleur et la pagination. La plus importante de ces règles est ce que nous avons appelé le "principe de working set": lorsqu'il y a saturation de la mémoire réelle, la pagination ne doit pas prendre des pages aux working set actifs. Pour traiter un défaut de page dans des conditions de saturation, la pagination enlève une machine virtuelle de l'ensemble de multiprogrammation, ce qui a pour effet de libérer les blocs de la mémoire réelle occupés par le working set de cette machine.

L'utilisation du modèle des working set laisse le choix de l'algorithme de sélection d'un bloc libre ; pour le contrôleur, le seul critère imposé par le modèle est celui de la mémoire: une

INTRODUCTION

machine virtuelle ne peut accéder à l'ensemble de multiprogrammation que s'il y a suffisamment de blocs libres en mémoire réelle pour contenir son working set.

Lorsque nous avons intégré ces nouveaux programmes dans l'hyperviseur, le système de mesure nous a permis d'évaluer le fonctionnement de CP-67. Nous avons pu notamment étudier la répartition des machines virtuelles dans les différentes files d'attente.

CHAPITRE I

EXPLORATION ET EVALUATION DE L'HYPERVERSEUR CP-67

CHAPITRE I

1. INTRODUCTION

La plupart des systèmes de contrôle sont devenus volumineux et complexes à un point tel qu'il est parfois difficile d'expliquer leur comportement : en particulier de déterminer pourquoi tel système a un fonctionnement dégradé ou anormal dans certaines circonstances.

Il existe de nombreuses raisons de vouloir analyser le comportement et mesurer les performances de la configuration d'une installation. Dans notre cas, nous voulions disposer d'un "appareillage de mesure" permanent pour pouvoir, d'une part évaluer les changements apportés, aussi bien à la configuration de l'ordinateur, qu'aux programmes et algorithmes du système, et d'autre part contrôler l'évolution des paramètres fondamentaux de l'hyperviseur (11).

Pour réaliser ces différentes opérations, nous avons utilisé un appareil spécialisé connu sous le nom de BCU (10). Cette unité, indépendante de l'ordinateur, permet d'effectuer des mesures sans perturber le fonctionnement du système. Cet appareil, bien qu'il permette d'obtenir des mesures intéressantes, présente quelques inconvénients majeurs.

L'emploi du BCU amène une restriction importante : cette unité ne peut mesurer que des événements "électroniques", c'est à dire des événements qui changent l'état d'un composant de l'ordinateur, par

CHAPITRE I

exemple le passage de l'état "prêt" à l'état "attente". Il est donc impossible avec une telle unité de suivre le déroulement d'un processus particulier.

C'est à cause de cette restriction que nous avons abandonné l'emploi du BCU et que nous avons envisagé la réalisation d'un système de mesure par programme. L'élaboration d'un tel outil doit être basée sur une étude approfondie du système à mesurer pour définir le plus clairement possible les variables à mesurer, et pour déterminer quelles sont les modifications à apporter à ce système pour prélever les mesures.

Nous considérons que l'hyperviseur CP-67 est à chaque instant dans un certain état caractérisé par les ressources dont il dispose et par les processus qui utilisent, ou qui voudraient utiliser, ces ressources. Le changement d'état du système est dû à un évènement et c'est cet évènement, accompagné des modifications, qu'il fait apparaître qui doivent être enregistrés.

La réalisation de l'outil de mesure fut facilitée par deux facteurs :

- l'hyperviseur CP-67 contient déjà des variables qui enregistrent l'occurrence de certains phénomènes

CHAPITRE I

- l'emploi d'une machine virtuelle pour le relevé, l'analyse et l'exploitation des données, limite au minimum le temps que l'hyperviseur consacre aux mesures.

2. LE SYSTEME A MESURER : L'HYPERVISEUR CP-67 (4, 22)

C'est un système de multiprogrammation à temps partagé qui utilise le dispositif de traduction dynamique des adresses de l'ordinateur.

Les ressources réelles de la configuration (unité centrale, mémoire centrale, canaux d'entrée-sortie, ...) sont partagées entre les différentes machines virtuelles, et CP-67 a la charge de répartir au mieux ces différentes ressources, en fonction des demandes provenant de ces machines.

Lorsqu'un utilisateur se connecte au système, CP-67 le reconnaît et génère la machine virtuelle qui est pré-définie pour cet utilisateur, ce dernier peut alors charger dans la mémoire de sa machine virtuelle le système de programmation de son choix, par exemple CMS, OS/360, ou bien CP-67 lui-même, etc...

Lors de la conception de l'outil de mesure, nous avons décidé d'examiner plus particulièrement deux parties de CP-67 : la gestion de la mémoire centrale et la gestion de l'unité centrale ; ceci parce que nos observations préliminaires nous

CHAPITRE I

avaient permis de penser que c'était dans la gestion de ces deux ressources qu'il y avait un problème de performance. Ainsi nous avons effectué une étude détaillée des algorithmes liés à la distribution de ces ressources entre les différentes machines virtuelles.

2.1 Gestion de la mémoire centrale

Elle est conditionnée par la mise en oeuvre des mémoires virtuelles et par l'utilisation du mécanisme de traduction dynamique des adresses. Nous rappelons ici les grands principes de ces notions qui sont décrites plus en détail dans plusieurs ouvrages (4, 22, 23).

2.1.1 La mémoire virtuelle

Un programme qui s'exécute dans une machine virtuelle utilise la mémoire dont il dispose de la même manière que s'il s'exécutait dans un ordinateur réel, c'est à dire qu'il peut référencer toutes les adresses de 0 à $N-1$, N étant la taille de la mémoire de sa machine virtuelle, ou encore son espace d'adressage ; on appelle adresse virtuelle, une adresse quelconque de cet espace.

CHAPITRE I

2.1.2 Traduction dynamique des adresses

C'est la fonction qui associe à toute adresse virtuelle l'adresse réelle, si elle existe, qui lui correspond en mémoire centrale.

Une mémoire virtuelle est définie par un ensemble de tables :

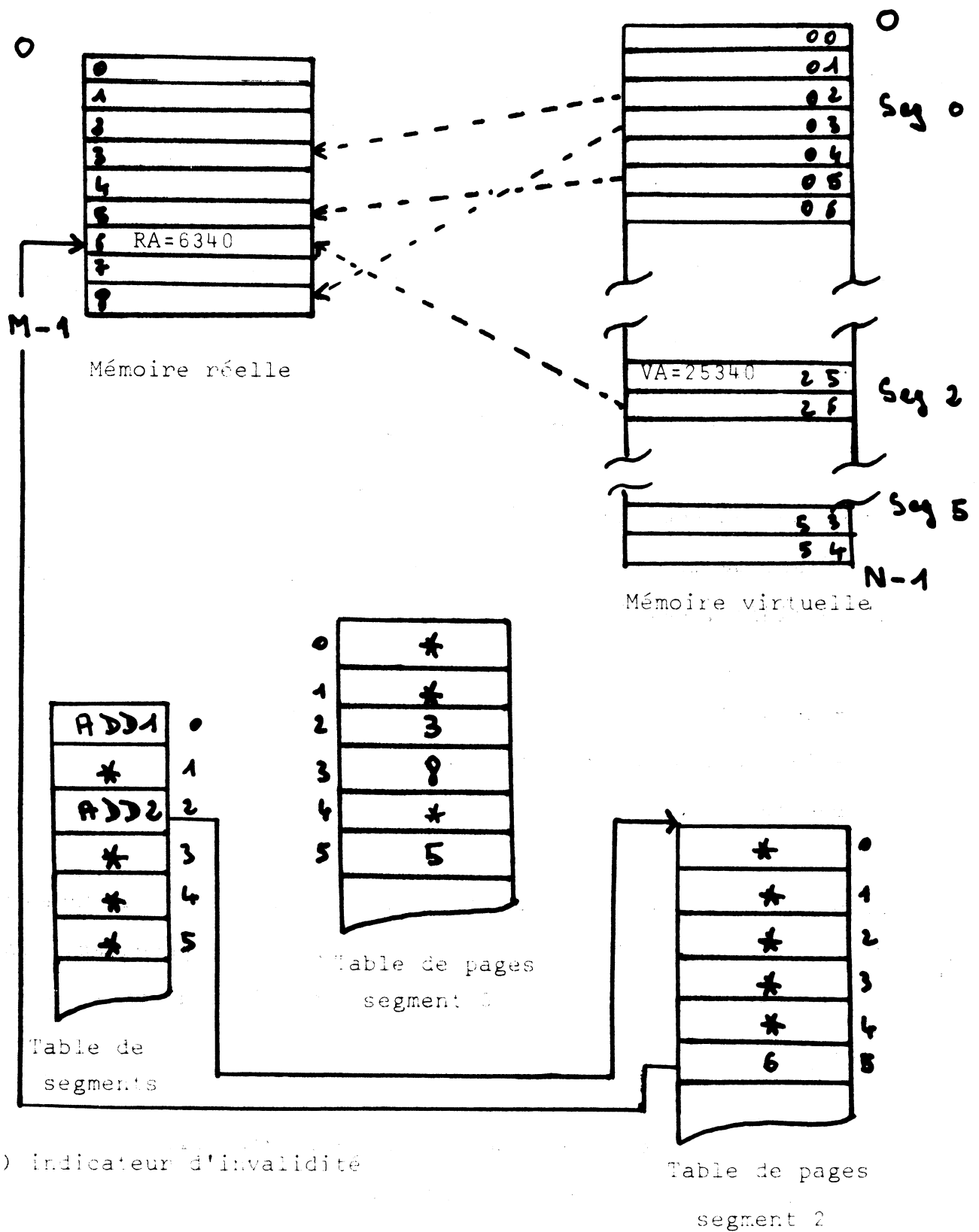
- une table de segments
- plusieurs tables de pages (1 par segment)

Dans une entrée de la table de segment, on trouve soit l'adresse de la table de pages du segment, soit un indicateur d'invalidité. Cet indicateur a deux sens :

- le segment n'existe pas dans la mémoire virtuelle
- la table de page du segment ne se trouve pas en mémoire réelle.

Dans une entrée d'une table de page, on trouve soit le numéro du bloc de la mémoire réelle qui contient cette page, soit un indicateur d'invalidité. De nouveau cet indicateur a deux sens :

- la page n'existe pas dans la mémoire virtuelle
- la page ne réside pas en mémoire réelle



TRANSFORMATION DES DRS ADRESSES

FIGURE 1.1

CHAPITRE I

Lorsque l'unité centrale reconnaît une adresse virtuelle, elle la transmet au mécanisme de traduction qui retourne alors une des trois réponses :

- adresse correspondante de la mémoire réelle
- invalidité de segment
- invalidité de page

Examinons cette consultation de tables (fig I.1).

Soit l'adresse VA = 25340 à traduire. Cette adresse est décomposée en 3 parties :

- numéro de segment : 2
- numéro de page : 5
- déplacement à l'intérieur de la page : 340

L'entrée 2 de la table de segments donne l'adresse de la table de page ; l'entrée 5 de cette table de pages indique que la page de la mémoire virtuelle est dans le bloc 6 de la mémoire réelle.

Ainsi à l'adresse VA = 25340 correspond l'adresse RA = 6340.

Pour l'ordinateur IBM 360/67, ces différents éléments ont une taille définie :

CHAPITRE I

- la taille d'une page est de 4096 octets
- la taille d'un segment est de 256 pages
- une mémoire virtuelle a au maximum 16 segments

Ainsi la capacité maximale d'adressage dans une mémoire virtuelle est de 16 mégaoctets, ce qui correspond à la plus grande adresse que l'on peut écrire sur 24 positions binaires.

2.1.3 Mécanisme de pagination dans l'hyperviseur CP-67

Avant d'étudier l'algorithme de pagination propre à CP-67, ajoutons quelques remarques sur la façon dont l'hyperviseur utilise le dispositif de traduction dynamique des adresses. Par construction, toutes les tables de pages des mémoires virtuelles sont résidentes en mémoire réelle, donc l'indicateur d'absence dans une entrée de la table de segment signifie que le segment n'existe pas.

D'autre part, lorsqu'une page ne réside pas en mémoire réelle, elle se trouve sur un support externe à accès sélectif : tambour ou disque. L'hyperviseur doit donc être capable d'associer à une page d'une mémoire virtuelle une adresse sur un support externe ; pour cela, à chaque table de pages est associée une table équivalente : c'est la table de pages externes, qui entre autres renseignements, contient l'adresse des pages sur le support externe.

CHAPITRE I

Enfin l'hyperviseur doit connaître l'état de l'occupation de la mémoire réelle ; pour cela il utilise une table de mémoire réelle qui indique pour chaque bloc de cette mémoire :

- si le bloc est occupé par l'hyperviseur
- si le bloc est libre
- si le bloc est occupé par une page de la mémoire virtuelle MVi
- si le bloc est occupé par une page de la mémoire virtuelle MVi, qui doit rester en mémoire réelle (page verrouillée).

Lorsque CP-67 est averti par le mécanisme de traduction dynamique des adresses qu'une page référencée n'est pas en mémoire réelle (défaut de page), l'algorithme de sélection recherche un bloc en consultant la table de la mémoire réelle, pour y amener la page manquante.

Cette sélection se fera ainsi : CP choisit soit un bloc libre, soit un bloc contenant une page non verrouillée d'une mémoire virtuelle ne participant pas à la multiprogrammation, soit enfin le premier bloc contenant une page, non verrouillée, appartenant à une mémoire virtuelle participant à la multiprogrammation ; dans ce dernier cas nous dirons qu'il y a "vol de page". Lorsque l'emplacement est choisi, le superviseur de pagination doit peut être réécrire le contenu du bloc sur support externe : en effet, si ce contenu a été

CHAPITRE I

modifié pendant son séjour en mémoire réelle, la copie qui se trouve sur le support externe n'est plus valide.

Après cette éventuelle réécriture, le superviseur de pagination peut demander la lecture de la page manquante à partir du support externe, grâce à la table des pages externes.

2.2 Gestion de l'unité centrale

Elle est assurée par une partie de l'hyperviseur CP-67 que nous appellerons contrôleur ; celui-ci a plusieurs fonctions, mais son rôle principal est d'assurer la meilleure utilisation de la ressource unité centrale afin que l'efficacité mesurée (temps d'exécution des machines virtuelles par rapport au temps total) soit la plus grande possible. Toutefois, dans un système à partage de temps, le fait d'avoir une grande efficacité ne doit pas être le seul critère de jugement ; en effet, il est facile d'obtenir une grande efficacité en choisissant des processus fort consommateurs de temps d'unité centrale, et cela au détriment des processus qui attendent une réponse rapide du système. Ainsi, un autre critère de jugement, le "temps de réponse", doit être pris en compte ; malheureusement ce temps de réponse est très difficile à définir et à mesurer au niveau du système.

CHAPITRE I

Le but du contrôleur est donc d'assurer à la fois une bonne efficacité de l'unité centrale, et un bon temps de réponse aux utilisateurs. Pour cela le contrôleur doit choisir une machine virtuelle et lui donner le contrôle.

Ce choix est conditionné par la valeur instantanée de certains paramètres de fonctionnement qui déterminent l'état des différentes machines virtuelles.

2.2.1 Classification des machines virtuelles

Les différentes machines virtuelles présentes dans le système sont constamment en évolution et le contrôleur sait déterminer leur état et le genre de travail qu'elles effectuent ; cette classification est propre au contrôleur de CP-67.

- Machines virtuelles dans l'état arrêté : ce sont les machines en attente pour une durée indéterminée. Par exemple, attente de la fin d'une entrée d'informations depuis le terminal.

- Machines virtuelles dans l'état non-prêt : ce sont les machines en attente pour un court instant ; par exemple, attente de la disponibilité d'une page manquante.

CHAPITRE I

- machines virtuelles dans l'état prêt : ce sont les machines virtuelles qui ne sont en attente d'aucun évènement
- machines virtuelles dans l'état éligible : ce sont les machines virtuelles qui sont prêtes, mais qui ne participent pas à la multiprogrammation
- machines virtuelles dans l'état élu : ce sont les machines virtuelles qui ont été choisies par le contrôleur, parmi les machines dans l'état éligible pour participer à la multiprogrammation.

Remarque : seules les machines virtuelles dans l'état élu peuvent se trouver aussi dans l'état non-prêt. En plus de leur état, le contrôleur attribue aux machines virtuelles une propriété suivant le genre de travail qu'elles effectuent.

- machines virtuelles interactives : ce sont les machines virtuelles pour lesquelles une opération d'entrée d'informations depuis le terminal, vient de s'achever ; plus généralement ce sont les machines qui quittent l'état arrêté.
- machines virtuelles non interactives : ce sont les machines qui n'ont pas effectué d'opération d'entrée d'informations depuis le terminal pendant qu'elles étaient dans l'état élu.

CHAPITRE I

2.2.2 Principaux paramètres du contrôleur

Lorsque le contrôleur attribue l'unité centrale à une machine virtuelle, il doit s'assurer qu'elle ne gardera pas cette ressource pendant un temps trop grand ; pour cela il utilisera l'horloge interne de l'unité centrale afin de limiter le temps pendant lequel une machine virtuelle dispose de la ressource unité centrale.

On appelle POTENTIEL le temps d'unité centrale pendant lequel une machine virtuelle pourra rester dans l'état élu. Ce potentiel est différent suivant que la machine virtuelle est interactive ou non ; pour une machine interactive il est de 400 millisecondes, pour une machine non interactive, il est de 5 secondes. Ce potentiel n'est pas attribué en une fois à une machine virtuelle, il est divisé en tranches de temps appelées QUANTUM.

Le quantum est le temps maximum pendant lequel une machine virtuelle peut garder le contrôle de l'unité centrale. Ce quantum est de 50 millisecondes, quelle que soit la valeur du potentiel. Il faut noter que le temps consommé par une machine virtuelle est calculé par le contrôleur en additionnant le temps pendant lequel cette machine a disposé de l'unité centrale, et le temps passé par l'hyperviseur à effectuer un travail pour cette machine.

CHAPITRE I

Il existe deux autres paramètres utilisés par le contrôleur :

- le nombre maximum de machines virtuelles interactives dans l'état élu : NBQ1

- le nombre maximum de machines virtuelles non interactives dans l'état élu : NBQ2.

Les valeurs de ces deux paramètres ne dépendent que de la taille de la mémoire centrale.

Bien que dans l'organisation de ce contrôleur il n'existe pas de file d'attente, pour faciliter l'exposé du choix d'une machine virtuelle nous dirons que les machines virtuelles interactives dans l'état élu sont membres de la liste Q1, les machines virtuelles non interactives dans l'état élu sont membres de la liste Q2.

2.2.3 Choix d'une machine virtuelle

Avant que le contrôleur puisse choisir une nouvelle machine virtuelle, il doit calculer le temps consommé par la machine qui disposait de l'unité centrale. Cette machine peut alors changer de classification : si elle a épuisé son potentiel, elle passera de l'état élu à l'état éligible et de plus, si elle était interactive, elle perdra cette propriété.

CHAPITRE I

Le contrôleur procédera au choix de la nouvelle machine virtuelle qui contrôlera l'unité centrale, en appliquant l'un des critères suivants dans l'ordre où ils sont énumérés :

1 - la machine virtuelle choisie est la machine virtuelle interrompue précédemment si elle n'a pas épuisé son quantum, si elle est toujours dans les états prêt et élu

2 - la machine virtuelle choisie est la première machine dans l'état prêt trouvée dans Q1

3 - Si la valeur NBQ1 n'est pas atteinte, la machine virtuelle choisie est la première machine interactive dans l'état éligible

4 - Si la valeur NBQ2 n'est pas atteinte la machine virtuelle choisie est la première machine virtuelle non-interactive dans l'état éligible

5 - La machine virtuelle choisie est la première machine dans l'état prêt trouvée dans Q2.

Lorsque ces 5 critères ne sélectionnent aucune machine virtuelle, le contrôleur mettra l'unité centrale dans l'état attente.

Si les critères 1, 2 permettent le choix d'une machine

virtuelle, on dira qu'il y a une relance rapide ; dans le cas contraire, il faut examiner toutes les machines virtuelles.

Il faut faire aussi deux remarques sur l'ordre dans lequel sont appliqués les critères :

- dans la mesure du possible, le contrôleur essaie de choisir une machine virtuelle interactive, ceci afin de réduire le plus possible le temps de réponse de cette classe de machines virtuelles.

- le fait de placer le critère 4 avant le critère 5 montre que le contrôleur essaie d'avoir le plus possible de machines virtuelles dans l'état élu, ceci afin de remplir au maximum la mémoire centrale.

2.3 Premières variables à mesurer

Les algorithmes que nous venons de décrire sont ceux de la version 3.0 de l'hyperviseur CP-67. Cette version était en fonction sur l'ordinateur IBM 360/67 durant la période de mesures que nous présenterons ultérieurement.

De cette étude préliminaire, nous avons établi une première liste de variables qu'il nous semblait indispensable de mieux

CHAPITRE I

connaître pour évaluer le comportement de CP-67 et intervenir en retour sur la valeur de certaines d'entre elles afin d'améliorer les performances du système.

2.3.1 Variables venant du contrôleur

Comme nous l'avons souligné dans la présentation du contrôleur, CP-67 attribue l'unité centrale aux machines virtuelles ou utilise lui-même cette ressource pour satisfaire les besoins de ces machines. Nous devons souligner ici une particularité importante de CP-67 ; lorsque l'unité centrale est utilisée par l'hyperviseur, elle est placée dans un état privilégié : elle fonctionne en mode superviseur, n'utilise pas la traduction des adresses et n'est pas interruptible par les canaux d'entrée-sortie. Par contre, lorsqu'une machine virtuelle a le contrôle de l'unité centrale, le mode de fonctionnement est le mode problème, la traduction des adresses est en fonction et toutes les interruptions sont permises. Cette particularité fait apparaître 3 variables à mesurer :

- le temps d'unité centrale passé en mode superviseur
- celui consommé en mode problème
- déduit des 2 précédents, le temps que l'unité centrale a passé en état "attente".

CHAPITRE I

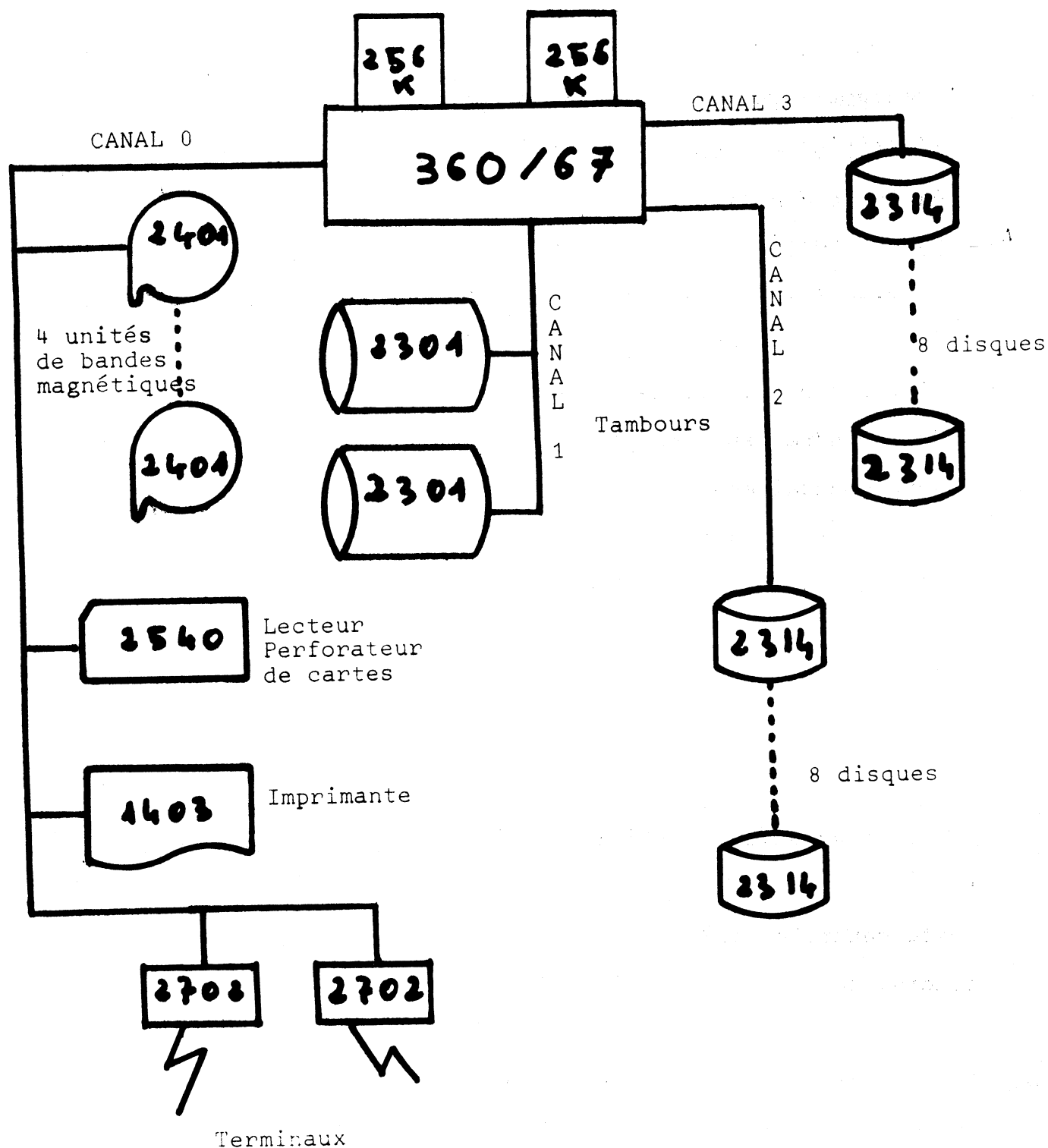
La répartition du temps est nettement distinguée entre l'hyperviseur et les machines virtuelles, aussi nous pouvons définir une des variables fondamentales du système : l'EFFICACITE ; nous appelons efficacité le rapport entre le temps en mode problème et le temps total, ce rapport indique la fraction du temps qui est accordé aux programmes qui s'exécutent dans les différentes machines virtuelles.

De façon similaire nous établissons le rapport entre le temps en mode superviseur et le temps total, ce rapport indique la fraction du temps pris par l'hyperviseur CP-67 lui-même.

2.3.2 Variables venant de la pagination

L'algorithme de pagination et le mécanisme de traduction dynamique des adresses ont mis en évidence certaines variables à mesurer :

- le nombre de "défauts de page" : nombre de réponses page absente provenant du mécanisme de traduction
- le nombre de "pages volées" : nombre de fois où il a été nécessaire de prendre un bloc attribué à une machine virtuelle élue
- le nombre de "pages écrites" : nombre de fois où il a



CONFIGURATION EN JUILLET 1971

FIGURE I.2

CHAPITRE I

fallu écrire sur support externe le contenu d'un bloc de la mémoire réelle.

Il faut remarquer que CP-67 au niveau de la version 3.0 était totalement résident en mémoire centrale et par là-même, toute l'activité de pagination était due uniquement aux différentes machines virtuelles. Cette liste de variables constitue le lot initial des paramètres que nous voulions étudier ; toutefois la connaissance plus approfondie de CP-67, de l'éventail des activités des machines virtuelles et l'évolution naturelle de nos premiers travaux nous ont permis de prendre en compte un certain nombre de variables supplémentaires ; c'est l'ensemble des variables que nous présenterons dans la suite de ce texte. Avant de passer à l'examen des mesures effectuées, donnons la configuration de l'ordinateur sur lequel CP-67 était en fonctionnement au mois de juillet 1971.

2.4 Configuration de l'ordinateur (fig. 1.2)

Unité centrale IBM 360/67 avec une mémoire centrale de 512K octets, soit 128 blocs de 4K-octets.

Par construction de l'hyperviseur et par la définition des unités d'entrée-sortie qui lui sont attribuées, les tambours magnétiques sont réservés à la pagination, c'est à dire qu'ils

CHAPITRE I

sont le support externe des mémoires virtuelles.

Les unités de disques servent de support aux fichiers des utilisateurs qui sont à conserver en permanence ; une fraction de ces disques est réservée à CP-67 lui-même pour y installer une copie de tous les constituants du système, ainsi que des espaces de travail nécessaires à son fonctionnement (pour mémoire on peut citer les zones indispensables à la mise en oeuvre des techniques de spooling, les zones attribuées dynamiquement aux machines virtuelles, ainsi que les zones qui peuvent être utilisées pour une extension de la pagination lorsque les tambours sont saturés).

Quant à la mémoire centrale, elle est divisée en 3 zones ayant des usages différents ; une zone de 25 blocs de 4K reçoit les instructions de l'hyperviseur, une zone de 12 blocs de 4K contient les différentes tables et les informations de contrôle nécessaires à CP-67 pour lui permettre de gérer l'ordinateur et les machines virtuelles (cette zone peut être augmentée par blocs de 4K octets, si nécessaire au détriment de la zone suivante), une zone comportant tous les blocs restants qui recueillent les pages des différentes mémoires virtuelles.

Enfin connectés aux unités de contrôle de transmission, 40 terminaux peuvent accéder au système.

CHAPITRE I

3. LE SYSTEME DE MESURE

Lorsque nous avons commencé à élaborer ce système, nous savions que la plupart des machines virtuelles répertoriées dans le catalogue de CP-67 (200 environ) avaient chacune 64 pages de mémoire virtuelle, quelques unes en ayant 128 ou 256. Nous savions aussi que la charge de l'ordinateur provenait de l'utilisation de composants tels que l'éditeur de fichier, le sous-système d'aide à la mise au point des programmes, les assembleurs ou compilateurs pour les langages FORTRAN, PL/1, ALGOL 60, ALGOL W, PL/360 ...etc..., ainsi que des composants spéciaux développés par les utilisateurs.

Il nous a paru essentiel que notre système de mesure fonctionne d'une façon continue sans perturber le déroulement normal des opérations de l'hyperviseur ; cela fut facilité par l'existence d'un moyen de communication privilégié entre CP-67 et les machines virtuelles. Ainsi ce système de mesure est divisé en 2 parties :

- Les variables décrivant un évènement et les instructions pour la mise à jour de leur valeur se trouvent dans l'hyperviseur lui-même.
- Le relevé, la sauvegarde et l'exploitation de ces variables sont effectués par un ensemble de programmes s'exécutant dans une machine virtuelle spécialisée que nous appellerons "la machine MESURE".

CHAPITRE I

De cette façon, le temps pris par CP-67 pour actualiser la valeur des variables est très faible, et l'influence de la machine MESURE interviendra dans les mesures au même titre que les autres machines virtuelles présentes dans le système.

3.1 Objectifs du système de mesure

L'analyse du contexte dans lequel est utilisé CP-67 et des services qu'il devait rendre nous a conduit à construire le système de mesure de telle sorte qu'il satisfasse trois besoins principaux :

- 1) surveillance : le système de mesure doit permettre de contrôler le système mesuré ; pour cela nous avons sélectionné certaines des variables pour les visualiser soit sur un terminal alphanumérique, soit sur un terminal de type machine à écrire. C'est l'administrateur du système qui doit examiner ces variables importantes lorsque l'hyperviseur présente un fonctionnement dégradé ou plus simplement pour juger de la situation. L'administrateur a la possibilité d'intervenir manuellement sur quelques paramètres de CP-67 ; par exemple lorsque les mesures montrent que l'activité de pagination est trop élevée, il peut modifier les valeurs limites (NBQ1, NBQ2, cf I.2.2.2) du nombre de machines virtuelles élues et ainsi réduire le nombre de

machines intervenant dans la multiprogrammation.

2) Sélection et contrôle des améliorations : c'est l'objectif principal que nous nous étions fixé lorsque nous avons développé ce système de mesure. En effet, il nous paraît fondamental de pouvoir déterminer quelles sont les parties de l'hyperviseur qui doivent être améliorées pour obtenir un meilleur rendement. Pour cela, nous avons été amenés à ajouter temporairement de nouvelles variables afin d'examiner en détail chacune des parties de CP-67 que nous voulions connaître, puis à supprimer ces variables quand le résultat était obtenu. Par la suite après l'introduction des modifications dans CP-67, nous voulions en mesurer l'influence et par comparaison, vérifier que le rendement global du système était pour le moins équivalent à ce qu'il avait été avant les changements.

3) Production de statistiques : les résultats des mesures conservés dans des fichiers permettent d'obtenir des statistiques portant sur de longues périodes. Ces statistiques pourront être utilisées pour différents types de travaux. Une première utilisation consiste à les exploiter pour servir d'entrée à un modèle de l'hyperviseur. Elles peuvent aussi être utilisées par l'administrateur du système d'exploitation afin de contrôler la répartition des

CHAPITRE I

travaux en fonction de l'heure dans la journée. Ces statistiques nous ont aussi permis de déterminer la période de comparaison lors d'une modification de l'hyperviseur.

3.1.1 Choix des variables

Le lot de variables que nous avons présentées au paragraphe I.2.3 est resté la base de notre choix ; de plus, issues directement de la pagination et des programmes gérant les entrées-sorties sur disques, nous avons ajouté les variables qui nous permettaient de mesurer l'activité des canaux d'entrée-sortie et le nombre de demandes en attente sur ces canaux. Toutes ces variables sont généralement celles que l'on rencontre dans d'autres systèmes de mesure ; par contre nous avons eu plus de difficultés pour choisir des variables concernant les utilisateurs de CP-67. En effet, de par sa conception, CP-67 ignore totalement le genre de travail réalisé par les différentes machines virtuelles. Après quelques tâtonnements, nous avons obtenu un ensemble de variables propres à l'installation, à l'hyperviseur CP-67 et aux objectifs que nous voulions atteindre.

3.1.2 Classification des variables

Les différentes variables que nous avons choisies peuvent être classées en 4 catégories :

CHAPITRE I

1) variables d'identification

Elles sont destinées à identifier chaque mesure :

- date de la mesure
- heure de la mesure
- numéro de la mesure

2) Variables instantanées

Ces variables n'ont une signification qu'au moment du prélèvement des valeurs de l'ensemble des variables. Il n'existe aucune relation entre les valeurs obtenues à deux instants différents. Dans cette catégorie sont rangés :

- le nombre d'utilisateurs connectés au système
- l'intervalle de temps qui sépare le relevé du précédent
- le nombre de pages des mémoires virtuelles qui résident sur tambours
- le nombre de pages des mémoires virtuelles qui résident sur disques

3) variables cumulatives

Ces variables recueillent soit le nombre d'apparitions d'un évènement, soit le temps pendant lequel l'unité centrale est dans un certain état. A l'initialisation de l'hyperviseur

CHAPITRE I

elles ont toutes une valeur nulle ; CP-67 incrémente ces variables suivant leur attribution. Les valeurs relevées représentent l'évolution de l'évènement attaché à chaque variable depuis l'initialisation de l'hyperviseur. Dans cette catégorie sont rangés :

- les temps passés par l'unité centrale en attente, en mode problème, en mode superviseur
- le nombre de défauts de page
- le nombre de pages écrites sur support externe
- le nombre de pages volées
- le nombre d'opérations d'entrée-sortie effectuées par les machines virtuelles
- le nombre d'instructions privilégiées simulées par l'hyperviseur

4) variables d'intégration

Nous avons introduit ce type de variable pour mesurer des phénomènes dont l'évolution est très rapide. Ces phénomènes sont représentés par la valeur d'une variable qui change fréquemment entre deux relevés de mesures, aussi nous nous sommes intéressés à la valeur moyenne prise par la variable entre les relevés. Pour chaque phénomène, CP-67 enregistre le nombre d'arrivées de l'évènement correspondant et calcule le produit :

CHAPITRE I

nombre * temps pendant lequel ce nombre est resté inchangé

Ce produit est ensuite ajouté à la valeur de la variable d'intégration attachée au phénomène, qui est une variable cumulative. La valeur moyenne est obtenue en divisant l'accroissement de cette variable par le temps qui a séparé 2 mesures.

C'est le seul cas où la mise à jour des variables demande à CP-67 l'exécution d'un nombre relativement important d'instructions (30 environ).

Ces variables nous permettent de mesurer :

- le nombre moyen de machines virtuelles interactives dans l'état élu
- le nombre moyen de machines virtuelles non interactives dans l'état élu
- le nombre moyen de demandes d'entrée-sortie en attente sur chaque canal

3.2 Relevé des mesures

Pour effectuer le relevé des valeurs des variables nous utilisons la machine virtuelle MESURE, qui emploie un moyen de communication propre à l'hyperviseur CP-67, ce moyen de

CHAPITRE I

communication est basé sur l'emploi de l'instruction "DIAGNOSE".

3.2.1 L'instruction "DIAGNOSE"

Sur les ordinateurs de la série IBM 360, il existe une instruction privilégiée qui n'est utilisée que par les programmes de maintenance de l'ordinateur : c'est l'instruction "DIAGNOSE" (23). Cette instruction permet de tester les circuits internes de la machine et produit des résultats différents suivant le type d'ordinateur. De ce fait, CP-67 ne pouvait simuler une telle instruction et il l'utilise comme moyen de communication entre une machine virtuelle et lui-même.

Une des formes de cette instruction permet à une machine virtuelle de demander le transfert dans sa mémoire virtuelle du contenu de mots de la mémoire réelle. La machine virtuelle qui exécute cette instruction fournit la liste des adresses en mémoire centrale et l'adresse d'un emplacement dans sa mémoire virtuelle, CP-67 prélève le contenu des mots adressés et les transfère dans la mémoire virtuelle.

3.2.2 La machine virtuelle MESURE

Le relevé des variables et leur sauvegarde sont effectués à des intervalles de temps réguliers par cette machine virtuelle. Lors de la mise en route du système de mesure, le programme chargé de relever les valeurs, obtient de CP-67 la liste des variables contenues dans l'hyperviseur, liste qui a été construite à la génération de l'hyperviseur. Dans cette liste se trouvent le nom de chaque variable et son adresse en mémoire centrale ; la machine MESURE examine cette liste et prépare une table d'adresses qui sera utilisée par les instructions DIAGNOSE qui seront exécutées pour relever les valeurs.

A la suite de cette initialisation, et plus généralement après chaque relevé, le programme qui contrôle la machine MESURE met dans l'horloge virtuelle de cette machine la valeur correspondant à l'intervalle entre deux mesures et place la machine virtuelle dans l'état attente.

Lorsque l'intervalle de temps est écoulé, l'interruption d'horloge qui marque cet événement redonne le contrôle à la machine MESURE, qui relève les valeurs des variables. Ce programme exécute l'instruction DIAGNOSE préparée au cours de l'initialisation et obtient ainsi un ensemble de valeurs qu'il sauvegarde immédiatement dans un fichier. Ce fichier retient donc l'historique du fonctionnement du système.

CHAPITRE I

Ultérieurement toutes ces informations seront analysées suivant les besoins et suivant les phénomènes que l'on veut étudier.

Parmi les valeurs qui viennent d'être recueillies un certain nombre d'entre elles sont traitées immédiatement pour matérialiser le fonctionnement global de l'hyperviseur depuis le dernier relevé et sont visualisées sur un terminal.

Le programme de contrôle de la machine virtuelle MESURE accepte deux commandes provenant de son terminal. La première permet de modifier la longueur de l'intervalle entre deux relevés ; la seconde permet de visualiser sur le terminal la moyenne des valeurs des différentes variables depuis la mise en route du système de mesure.

C'est à l'aide de ces deux commandes que l'administrateur du système peut surveiller le comportement de l'hyperviseur.

3.3 Exploitation des mesures

Elle se fait à partir du fichier qui a été sauvegardé ; cette exploitation est différente suivant le type de renseignements que l'on désire obtenir, ou le type de paramètres dont on veut observer l'évolution.

CHAPITRE I

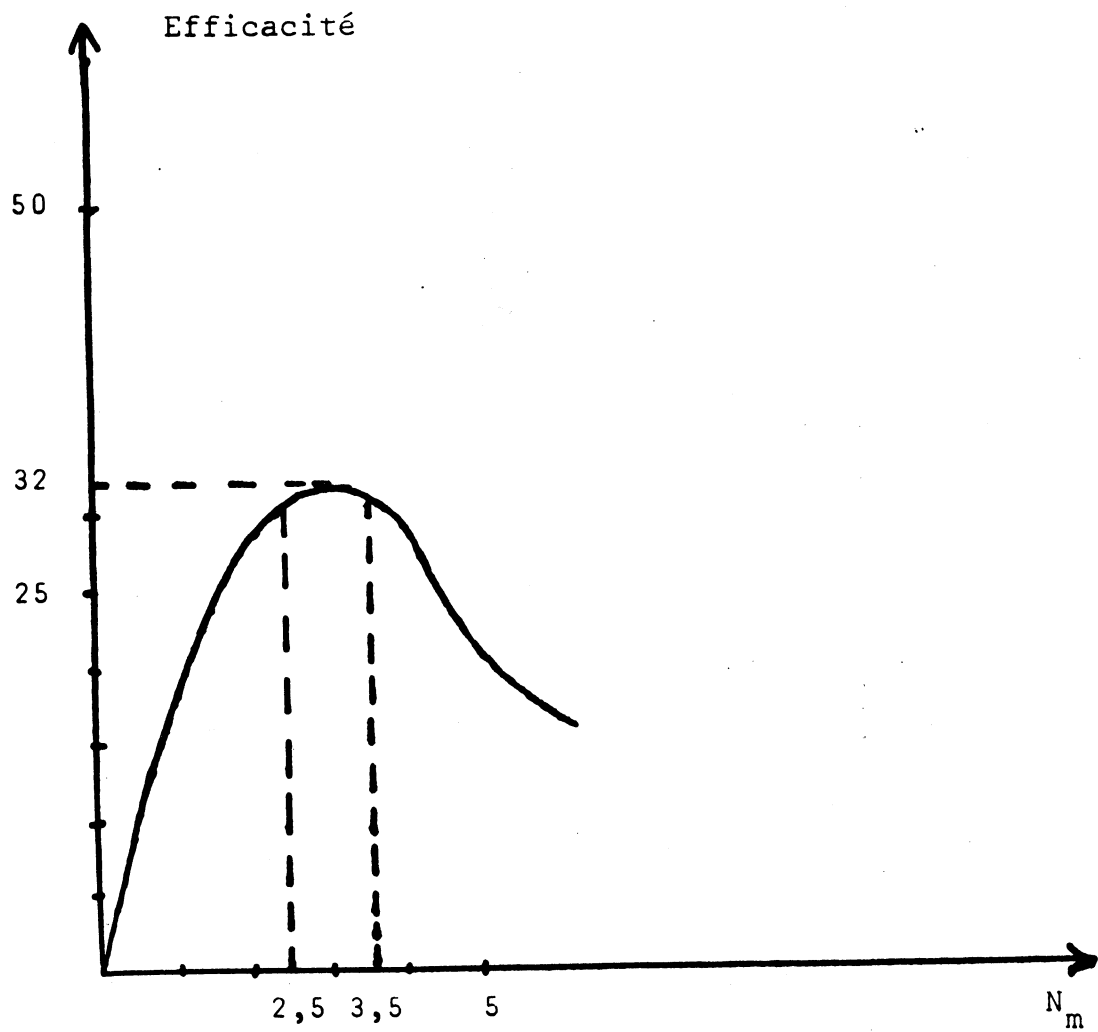
Pour exploiter ces fichiers, nous avons développé deux types de programmes ; les premiers permettent de tracer les courbes des valeurs moyennes des variables, en fonction du nombre d'utilisateurs présents dans le système ; les seconds tracent les courbes de ces mêmes variables en fonction du "niveau moyen de multiprogrammation", c'est à dire en fonction de la charge du système. Dans la suite, nous ne présenterons que des courbes obtenues à partir du deuxième type de programme.

3.3.1 Niveau moyen de multiprogrammation

Parmi toutes les machines virtuelles connectées à l'ordinateur, pendant une certaine période, seul un sous-ensemble de celles-ci est en compétition pour utiliser les différentes ressources à un instant donné, ce sont les machines virtuelles dans l'état élu. Ces machines sont celles qui peuvent exécuter des instructions et qui, par conséquent, consomment du temps d'unité centrale en mode problème. Nous définissons alors :

- le niveau de multiprogrammation instantané qui est le nombre de machines virtuelles dans l'état élu à un instant donné.

Ce nombre varie très souvent entre deux relevé de mesures, aussi nous ne pouvons raisonnablement mesurer que sa valeur



EFFICACITE

FIGURE I.3

CHAPITRE I

moyenne sur l'intervalle de temps entre deux mesures.

Nous définissons :

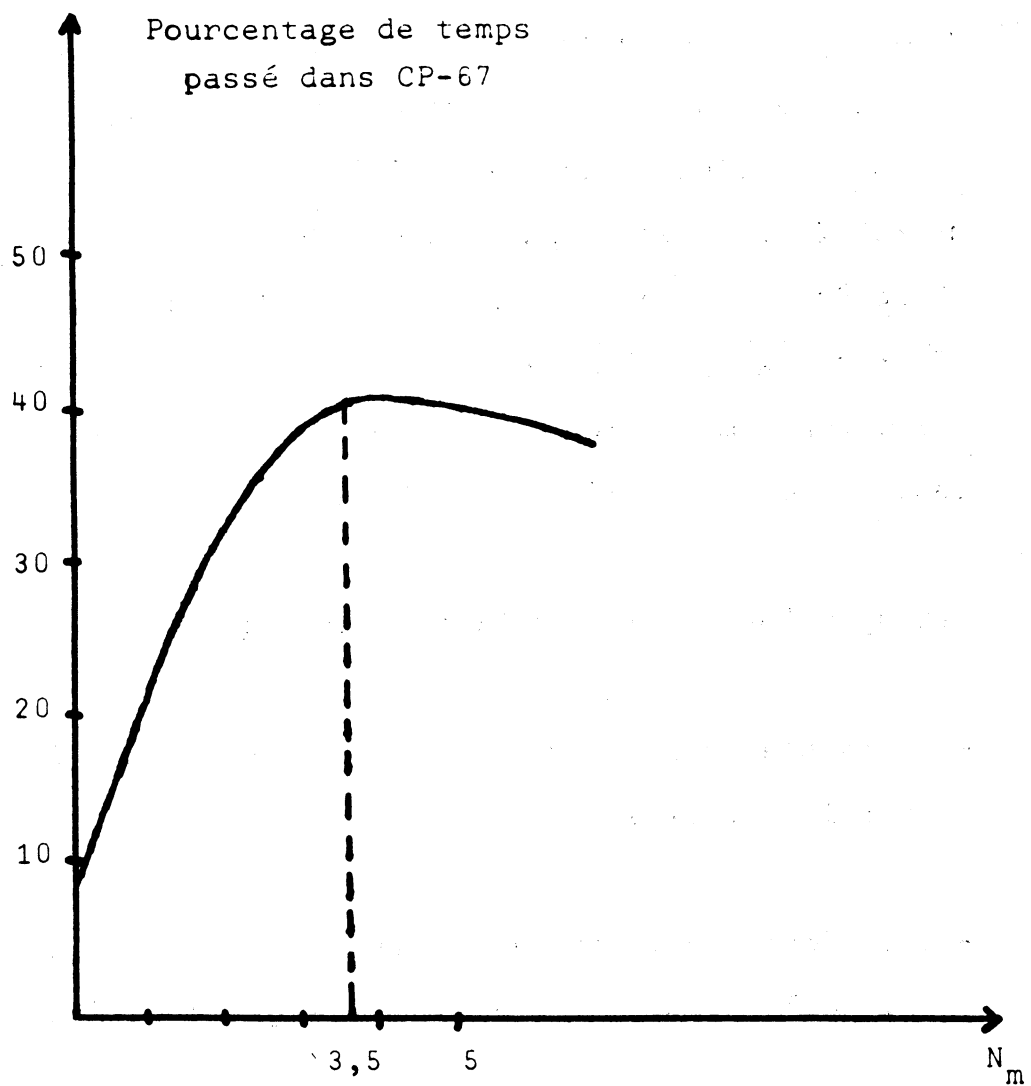
- le niveau moyen de multiprogrammation qui est la valeur moyenne du nombre de machines virtuelles dans l'état élu durant un intervalle donné.

C'est cette variable N_m qui est la variable indépendante de nos programmes d'exploitation des mesures.

3.3.2 Présentation et discussion de certaines courbes

Nous rappelons que les courbes présentées ici sont celles obtenues pendant le mois de juillet 1971, l'intervalle entre 2 mesures est de 100 secondes, et le nombre de mesures sauvegardées pendant cette période est d'environ 2500.

La figure I.3 présente le pourcentage de temps passé par l'unité centrale, en mode problème, en fonction de N_m . C'est ce pourcentage que nous avons appelé EFFICACITE. Nous remarquons que le maximum d'efficacité est atteint pour un niveau moyen de multiprogrammation compris entre 2,5 et 3,5. Au-delà de la valeur 3,5 l'efficacité décroît rapidement quand N_m augmente. Nous nous trouvons devant une situation anormale qu'il faut analyser plus en détail.



POURCENTAGE DE TEMPS D'UNITE CENTRALE

PASSE DANS CP-67

FIGURE I.4

CHAPITRE I

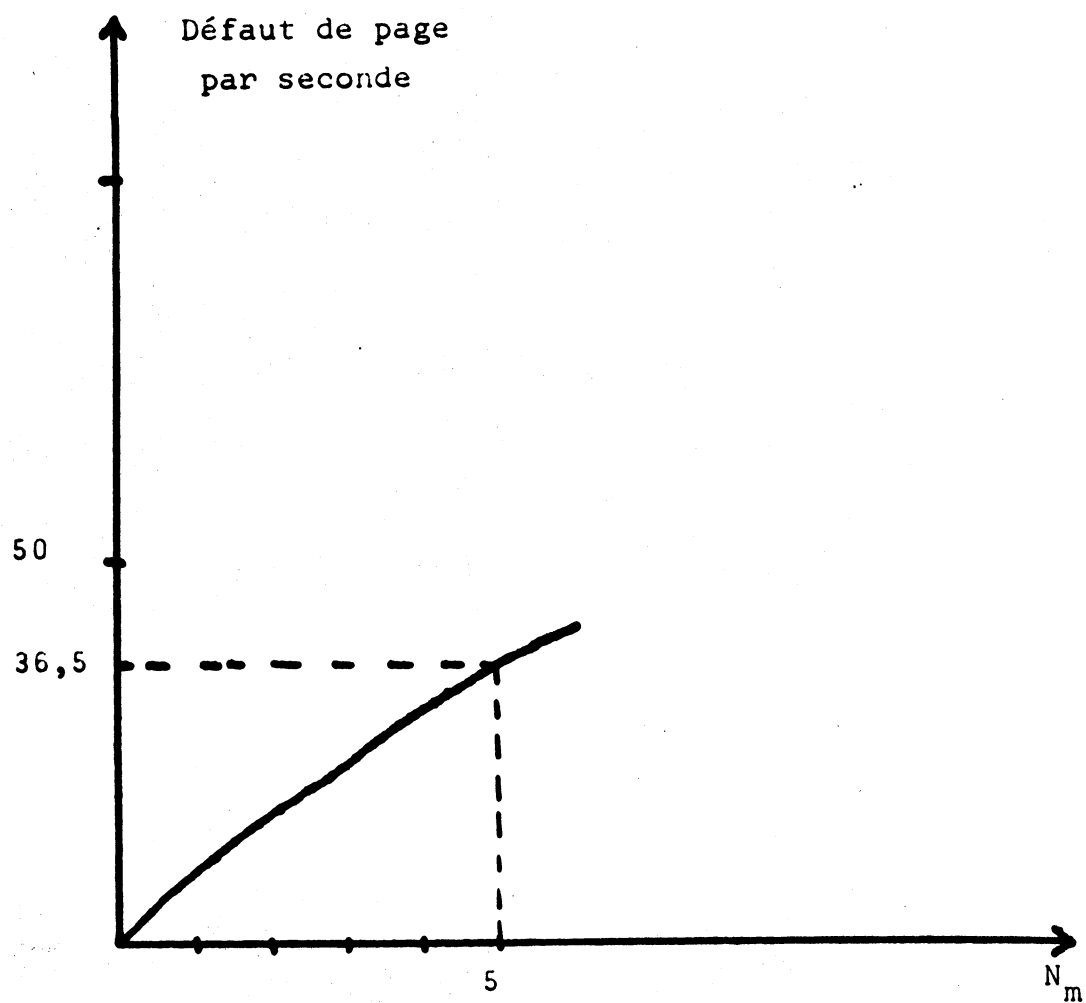
La figure I.4 présente le pourcentage du temps consommé par CP-67, en fonction de N_m . Nous observons que ce pourcentage croît régulièrement lorsque N_m varie de 0 à 3,5, puis au-delà de cette valeur de N_m , le temps passé dans l'hyperviseur diminue très lentement. La courbe précédente nous a montré, par contre, que l'efficacité diminue très rapidement à partir de $N_m = 3,5$; bien que l'activité des machines virtuelles décroît, le temps passé dans CP-67 reste important. Ce phénomène est connu sous le nom d'effondrement du système ("thrashing"), il se produit lorsque le niveau de multiprogrammation dépasse une certaine valeur. Dans notre cas, cette valeur limite semble être atteinte pour $N_m = 3,5$.

Les symptômes de l'effondrement sont :

- un faible pourcentage de temps passé par l'unité centrale à servir les utilisateurs
- une forte activité de pagination qui fait que le système continue à consommer beaucoup de temps d'unité centrale.

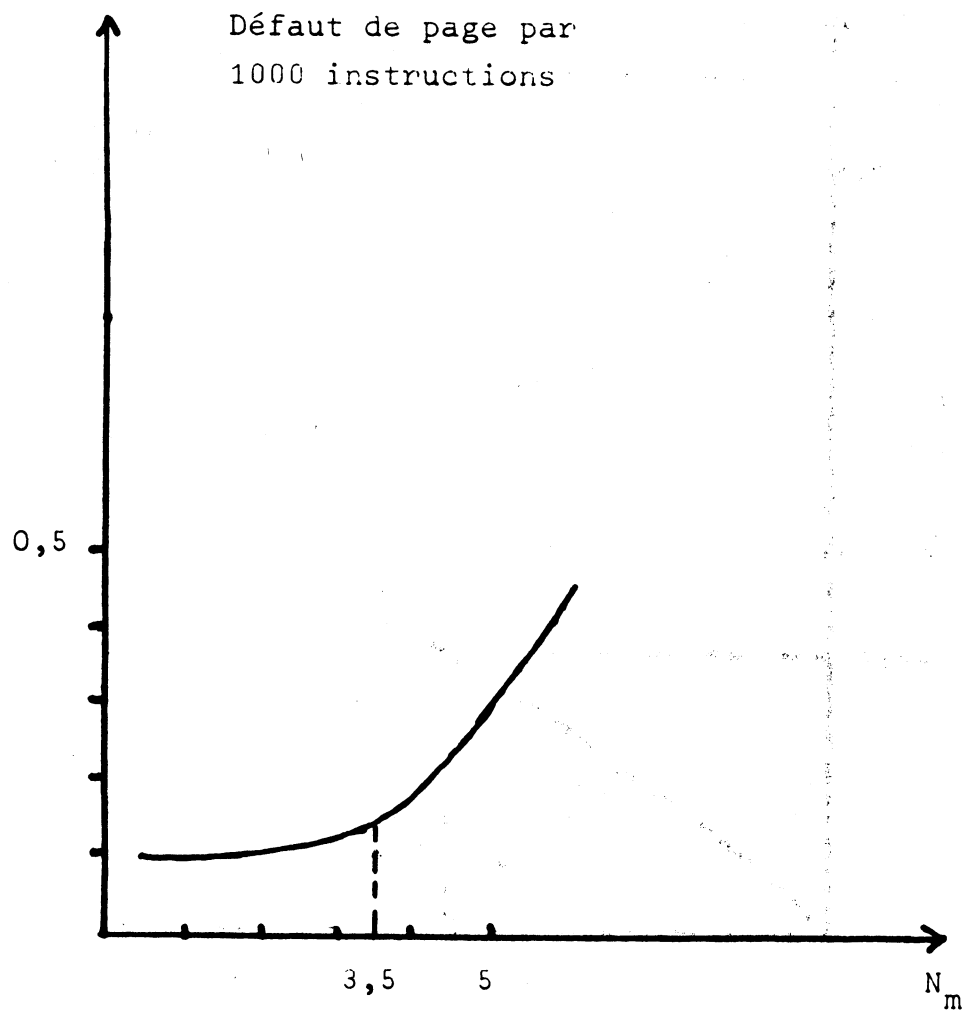
Afin de vérifier que nous sommes bien dans une situation d'effondrement, nous pouvons examiner les courbes qui décrivent l'activité de pagination.

La figure I.5 présente le nombre de défauts de page par seconde de temps écoulé, en fonction de N_m . Cette courbe est



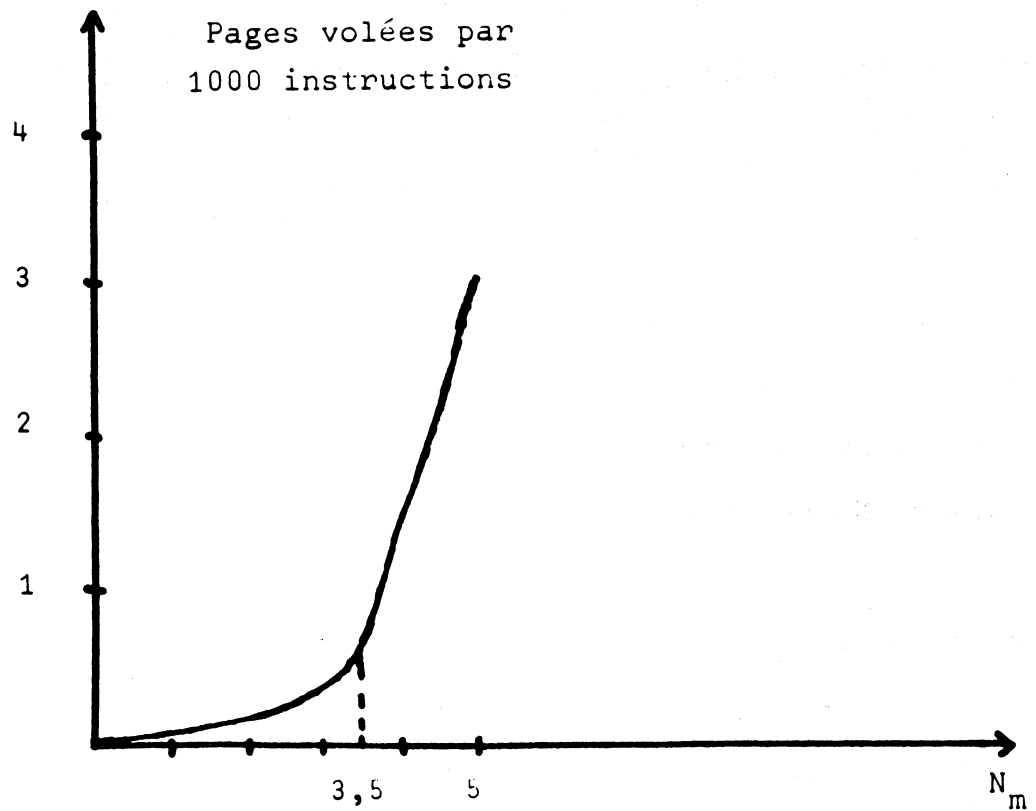
NOMBRE DE DEFAUTS DE PAGE PAR SECONDE

FIGURE I.5



NOMBRE DE DEFAUTS DE PAGE PAR 1000 INSTRUCTIONS

FIGURE 1.6



NOMBRE DE PAGES VOLEES PAR 1000 INSTRUCTIONS

FIGURE I.7

CHAPITRE I

pratiquement une droite, c'est à dire que le nombre de défauts de page rapporté au temps écoulé, reste proportionnel à N_m . Nous espérons que cette courbe présenterait un changement pour $N_m=3,5$; ce n'est pas le cas, la courbe n'est pas significative et ceci est dû au fait que l'unité de temps choisie est incorrecte. Il faut compter le nombre de défauts de page par seconde de temps passé en mode problème, ou bien par instruction exécutée par les machines virtuelles.

La figure I.6 montre le nombre de défauts de page par 1000 instructions. Cette fois la courbe présente bien un changement de pente lorsque N_m dépasse la valeur 3,5.

La figure I.7 montre le nombre de pages volées aux machines virtuelles par 1000 instructions en fonction de N_m . Le changement de pente de cette courbe est très important lorsque N_m dépasse 3,5.

Nous pouvons alors expliquer l'effondrement du système lorsque le niveau de multiprogrammation augmente, cette situation de blocage est provoquée par le cumul de deux phénomènes :

- lorsque le niveau de multiprogrammation augmente, l'espace en mémoire réelle, réservé à chaque machine virtuelle diminue et chaque machine provoque plus de défauts de page

CHAPITRE I

- l'algorithme de pagination vole les pages des machines virtuelles qui participent à la multiprogrammation, ce sont des pages qui ont encore, en général, une forte probabilité d'être référencées, et cela augmente encore le nombre de défauts de pages.

Nous pouvons conclure à partir de ces courbes, que du point de vue de la pagination, l'hyperviseur CP-67 se comporte d'une manière satisfaisante, tant qu'il n'a pas atteint le point de saturation de la mémoire centrale ; mais le système sait mal reconnaître qu'il a atteint ce point de saturation et laisse le niveau de multiprogrammation s'accroître de façon exagérée.

CHAPITRE II

LE MODELE THEORIQUE DES WORKING SET

CHAPITRE II

1. INTRODUCTION

A la suite de la campagne de mesures effectuées sur l'hyperviseur CP-67, et après leur dépouillement et la mise en forme des résultats, nous avons fait le bilan de ce que nous avons recueilli et nous avons rapproché ces informations de l'étude des algorithmes de la pagination et du contrôleur. Notre but était de déterminer les qualités et les défauts que présentaient ces deux fonctions importantes du système, et par là-même, de proposer un certain nombre de modifications pour diminuer et même supprimer, si possible, les périodes pendant lesquelles le système présente de mauvaises performances.

Le principal reproche que l'on peut faire à la pagination est lié au "vol de pages". Les pages volées sont prises de façon aléatoire parmi les pages qui résident dans la mémoire centrale. Les différentes mesures que nous avons effectuées nous ont montré que tant que le nombre de pages volées est inférieur à 5 pages volées par seconde de temps écoulé, la situation du système reste bonne ; lorsque le nombre de pages volées augmente, il se produit un phénomène de "boule de neige" et alors le système passe une grande partie de son temps dans la pagination.

Quant au contrôleur, il est trop statique : le niveau de multiprogrammation est une variable dont la valeur n'est limitée que par une constante liée à la taille de la mémoire centrale.

CHAPITRE II

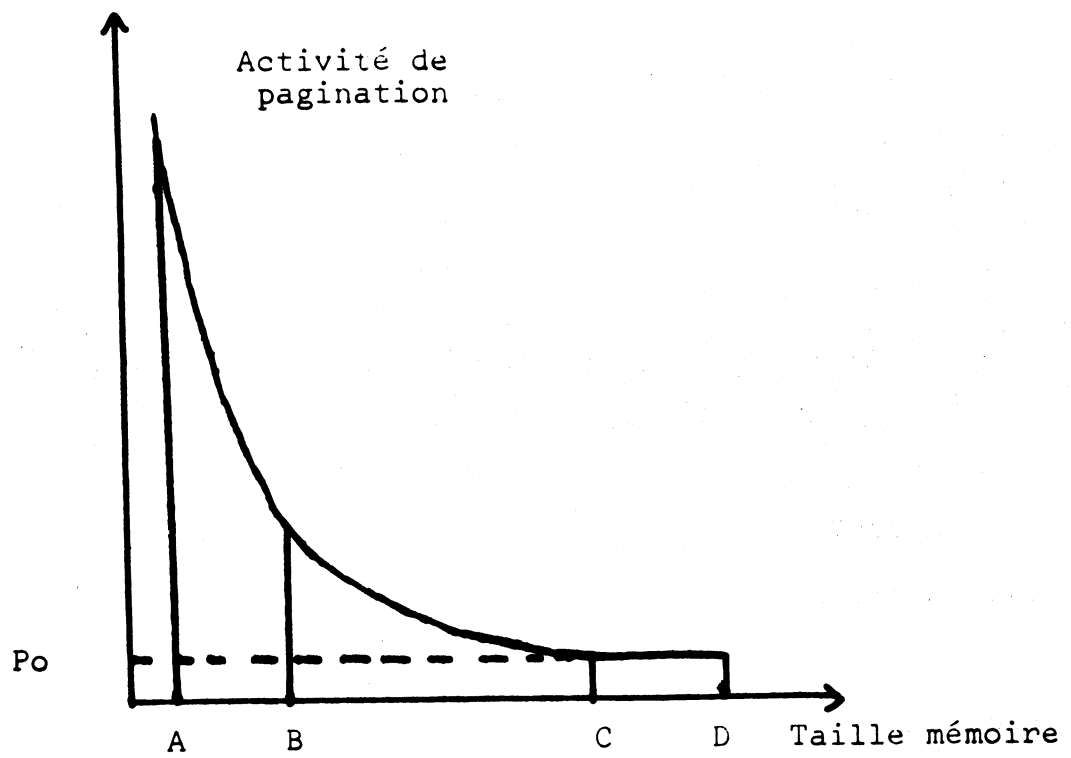
Nous pouvons aussi faire un autre reproche au contrôleur : les différentes machines virtuelles présentes dans le système ne sont pas organisées en files d'attente, aussi lorsqu'il doit choisir une machine virtuelle à activer, il est obligé d'examiner l'état de toutes les machines virtuelles avant de fixer son choix.

Toutefois, l'étude de ce contrôleur nous a montré certaines caractéristiques intéressantes ; notamment il sait différencier les machines virtuelles interactives et les machines virtuelles non interactives afin de favoriser les premières.

Ainsi à partir des conclusions que nous avons tirées de notre étude, il nous a semblé qu'il était possible d'améliorer le rendement du système en modifiant le contrôleur et la pagination, pour établir une certaine liaison entre ces deux fonctions. Ces modifications devaient répondre à un but précis : rendre ce contrôleur plus dynamique, c'est à dire que le niveau de multiprogrammation devait être lié à l'utilisation (et non plus à la taille) de la mémoire centrale. Pour atteindre cet objectif nous avons décidé de nous inspirer du modèle proposé par P.J. DENNING (8) connu sous le nom de modèle de "WORKING SET".

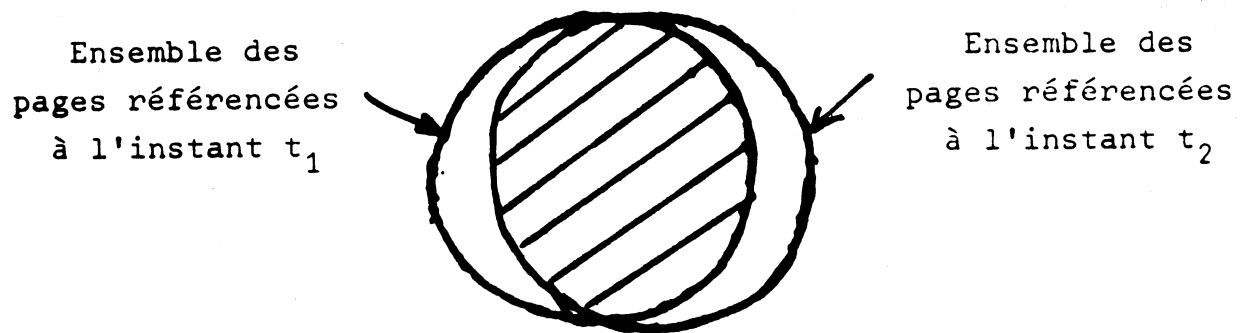
2. COMPORTEMENT DES PROGRAMMES DANS UN CONTEXTE DE PAGINATION

Les différents auteurs qui ont étudié les systèmes de pagination ont mis en évidence certaines caractéristiques du comportement des



ACTIVITE DE PAGINATION

FIGURE II.1



LOCALITE

FIGURE II.2

CHAPITRE II

programmes qui s'exécutent dans une mémoire paginée (3,7,8). Si, pour un programme donné, on trace la courbe donnant l'activité de pagination (en nombre de défauts de page) en fonction de la taille de la mémoire centrale, dans laquelle ce programme s'exécute, on constate que, quel que soit l'algorithme de remplacement utilisé, cette courbe a l'allure de celle de la figure II.1.

En faisant décroître la taille de la mémoire centrale à partir de la taille du programme, on distingue 3 zones significatives sur la courbe :

- une zone CD dans laquelle l'activité de pagination est constante ; dans cette zone le nombre de pages disponibles est supérieur ou égal au nombre P_0 de pages réellement utilisées lors de l'exécution du programme. Le nombre de défauts de page est alors égal à P_0 , il est dû au chargement des pages du programme.
- une zone BC dans laquelle le nombre de défauts de page croît lentement au fur et à mesure que la taille mémoire disponible décroît
- une zone AB dans laquelle une diminution faible de la taille de mémoire disponible entraîne une augmentation importante du nombre de défauts de page.

Pour chaque programme, il existe une taille de mémoire centrale se situant dans la zone BC qui réalise un bon compromis entre

CHAPITRE II

l'utilisation de la mémoire et le nombre de défauts de page acceptable pour le programme.

On peut faire une autre constatation, si l'on s'intéresse à la suite des pages référencées par un programme au cours du temps. En général, l'ensemble des pages référencées à un instant t_2 contient un certain nombre de pages qui étaient déjà référencées à l'instant t_1 si l'intervalle $t_2 - t_1$ est assez court (fig. II.2). Cette propriété est connue sous le nom de "localité des programmes".

3. DEFINITION DU WORKING SET (8)

Nous appelons working set d'un programme à un instant t , l'ensemble des pages qui ont été référencées dans l'intervalle $(t - \Delta t, t)$; ce working set est donc fonction de deux variables : le temps auquel est faite l'observation: t , la longueur de l'intervalle: Δt . Nous nous intéresserons plus spécialement au nombre de pages que contient cet ensemble : nombre que nous notons $W(t, \Delta t)$.

Cette fonction $W(t, \Delta t)$ possède un certain nombre de propriétés :

- $W(t, \Delta t)$ est une fonction croissante de Δt , en effet plus l'intervalle est grand, plus le nombre de pages référencées sera grand.

- La valeur $W(t, \Delta t)$ obtenue à l'instant t indique le nombre de pages qui ont été référencées pendant l'intervalle Δt ; en considérant les constatations faites précédemment, cette valeur pourra servir de prévision pour l'intervalle Δt suivant, et cette prévision sera acceptable si Δt est suffisamment petit.

- l'étude du comportement des programmes permet aussi d'établir le nombre de pages qui vont entrer dans le working set pendant l'intervalle Δt , nombre que nous notons $L(t, \Delta t)$. $L(t, \Delta t)$ est une fonction décroissante de Δt , en effet plus l'intervalle est grand, plus le nombre de pages référencées est grand et donc plus petit est le nombre de pages qui ne se trouvaient pas dans le working set.

4. ADAPTATION DU MODELE DE WORKING SET A L'HYPERVEISEUR CP-67

Pour pouvoir appliquer un tel modèle à CP-67, et pour répondre aux impératifs que nous nous étions fixés, nous avons utilisé un simulateur pour étudier le comportement des composants principaux d'un système (compilateurs et assembleurs) dans un environnement de pagination. Les résultats obtenus portaient sur l'étude de la variation de la taille du working set ($W(t, \Delta t)$) et du nombre de pages entrant dans le working set ($L(t, \Delta t)$) en fonction de la longueur de l'intervalle Δt (15).

CHAPITRE II

L'examen détaillé des résultats obtenus et l'étude du modèle proposé par DENNING nous ont fait adopter les deux principes ci-dessous pour réaliser les fonctions du contrôleur et de la pagination :

- la taille du working set calculé à un instant t est une bonne prévision de la demande du programme pour le prochain intervalle Δt ; de cette manière, nous pourrions contrôler et gérer plus efficacement la ressource mémoire centrale

- lorsqu'un conflit se produit au niveau de la mémoire centrale, c'est à dire lorsqu'aucun bloc libre n'est trouvé pour satisfaire une demande de page, nous refusons d'enlever une page à un working set, mais par contre nous diminuons le nombre de programmes concurrents pour la ressource mémoire centrale. Cela revient à diminuer le niveau de multiprogrammation.

CHAPITRE III

DESCRIPTION ET EVALUATION DES CHANGEMENTS APPORTES A L'HYPERVERSEUR

CHAPITRE III

Nous avons démontré que la pagination et le contrôleur doivent mutuellement échanger des informations pour que l'on obtienne de meilleures performances du système. Toutefois, il nous semble préférable de séparer la présentation de ces deux programmes pour faire apparaître clairement les attributions de l'un et de l'autre.

1. PRINCIPES DE FONCTIONNEMENT DE LA PAGINATION

Nous considérons qu'il est essentiel de préserver les working set d'une machine virtuelle dans l'état élu. En aucun cas ce principe ne devra être transgressé. Cette règle de conduite que nous appellerons "principe de working set" est une contrainte sévère que nous nous imposons pour tenter d'éliminer l'effondrement du système. En effet, si l'on diminue les working set des machines virtuelles, on peut atteindre le seuil à partir duquel les programmes qui résident dans ces machines ne peuvent plus s'exécuter de façon satisfaisante.

De ce fait, l'ensemble des blocs de la mémoire centrale, sur lequel opèrera l'algorithme de pagination, donc l'ensemble des pages qui sont susceptibles d'être remplacées est réduit aux pages qui n'appartiennent pas aux machines virtuelles élues.

Ainsi le vol de pages qui apparaissait dans le précédent algorithme de CP-67 est éliminé, les fautes de pages effectuées

CHAPITRE III

par les différentes machines virtuelles sont dues à deux causes :

- acquisition du working set lorsqu'une machine virtuelle vient de passer dans l'état élu
- déplacement du working set au cours de l'exécution d'un programme dans une machine virtuelle élue.

L'utilisation des working set permet, si on le désire, de faire de la post-pagination ; en effet lors du calcul de la taille du working set, on peut identifier les pages qui le composent, et les recopier sur un support externe.

L'algorithme de remplacement, lorsque l'on traite un défaut de page, n'est pas imposé dans ce modèle. Parmi les différentes techniques de remplacement, nous avons choisi celle qui sélectionne l'emplacement le moins récemment utilisé.

2. PRINCIPES DE FONCTIONNEMENT DU CONTROLEUR

Les utilisateurs de l'hyperviseur CP-67 peuvent être rangés dans 2 catégories :

- ceux qui effectuent beaucoup d'entrées-sorties sur le terminal de leur machine virtuelle et qui demandent peu de temps d'unité centrale.

CHAPITRE III

- ceux qui au contraire demandent beaucoup de temps d'unité centrale et effectuent peu d'entrées-sorties sur le terminal.

Un utilisateur qui effectue de l'édition de fichiers avec sa machine virtuelle, opération qui demande peu de temps d'unité centrale, est mécontent lorsque le clavier de son terminal ne se déverrouille pas immédiatement après avoir envoyé une commande.

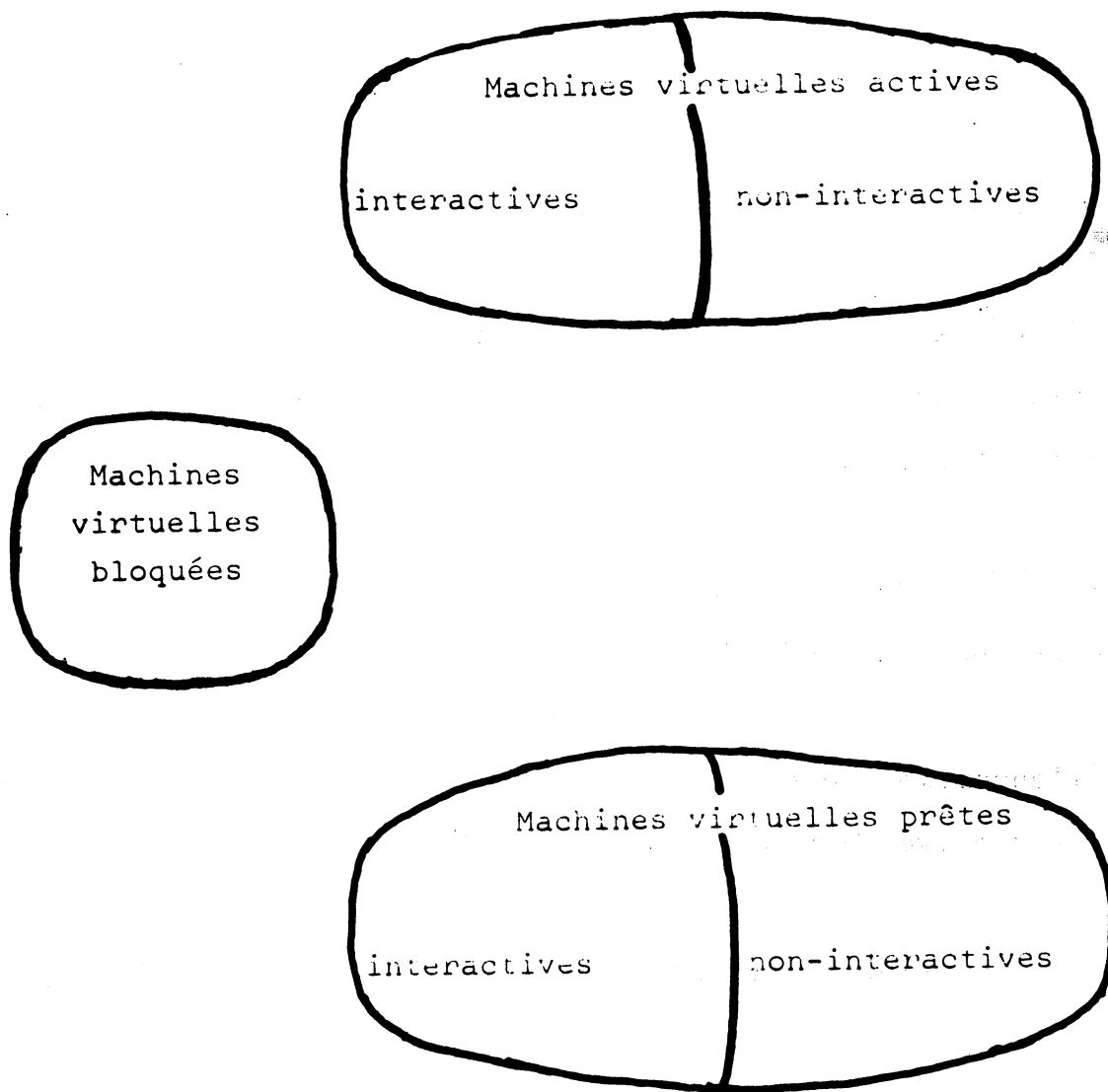
Il en est de même pour les utilisateurs qui examinent ou modifient le contenu de leur mémoire virtuelle, ou bien qui échangent des messages avec d'autres utilisateurs. Autrement dit, tous les utilisateurs qui effectuent des opérations interactives doivent obtenir un temps de réponse le meilleur possible. Après avoir consulté un certain nombre d'entre eux, nous avons établi qu'un temps de réponse de 1 seconde pour de telles commandes était satisfaisant, un temps de réponse de 2 secondes était accepté mais ennuyeux, un temps de réponse de 3 secondes ou plus était intolérable.

D'un autre côté, il y a des utilisateurs qui demandent à leur machine virtuelle des travaux non interactifs, par exemple l'assemblage ou la compilation d'un programme ; afin de favoriser les utilisateurs de la première catégorie, il nous a semblé que pour ces travaux non interactifs, on pouvait accepter des temps de réponse plus grands ; par exemple, le fait que l'exécution d'un assemblage dure 6 minutes au lieu de 4 ne doit pas gêner les utilisateurs qui demandent de tels travaux.

CHAPITRE III

Ces remarques préliminaires nous ont conduits à détailler la classification des machines virtuelles ; les machines peuvent être dans l'un des 5 ensembles de la figure III.1. C'est sur l'ensemble des machines virtuelles actives que s'effectue la multiprogrammation ; les machines virtuelles de l'ensemble prêt attendent elles, de pouvoir entrer dans l'ensemble actif. Une incidence immédiate d'un tel découpage est qu'il faut définir les règles de migration d'un ensemble dans l'autre ; de plus pour l'ensemble des machines virtuelles actives (celles qui participent à la multiprogrammation) il est nécessaire de fixer avec précision celle à qui le contrôleur donnera l'unité centrale à un instant donné.

Nous savons aussi qu'une machine virtuelle qui est dans l'ensemble actif ne peut y rester qu'un certain temps ; rappelons que l'on appelle potentiel le temps maximum pendant lequel une machine virtuelle peut rester dans l'ensemble actif. Ce potentiel n'est pas consommé en une seule fois par les machines virtuelles, on appelle quantum le temps maximum pendant lequel une machine virtuelle peut garder le contrôle de l'unité centrale. Lorsque la somme des quantum consommés atteint la valeur du potentiel, la machine virtuelle passe dans l'ensemble prêt. On peut noter dès maintenant qu'à la fin de chaque quantum, le contrôleur demande le calcul de la taille du working set qui a été nécessaire à la machine virtuelle. Cette valeur servira de prévision pour la demande de mémoire de cette machine lorsque le contrôleur lui attribuera de nouveau l'unité centrale.



REPARTITION DES MACHINES VIRTUELLES

FIGURE III.1

CHAPITRE III

A la différence du contrôleur d'origine de l'hyperviseur, seul le temps pendant lequel la machine a exécuté des instructions est pris en compte pour le calcul d'épuisement du quantum, et donc du potentiel.

2.1 Organisation des ensembles de machines virtuelles

Les différentes machines virtuelles présentes dans le système à un instant donné, sont rangées dans l'un des cinq ensembles suivants (figure III.1) :

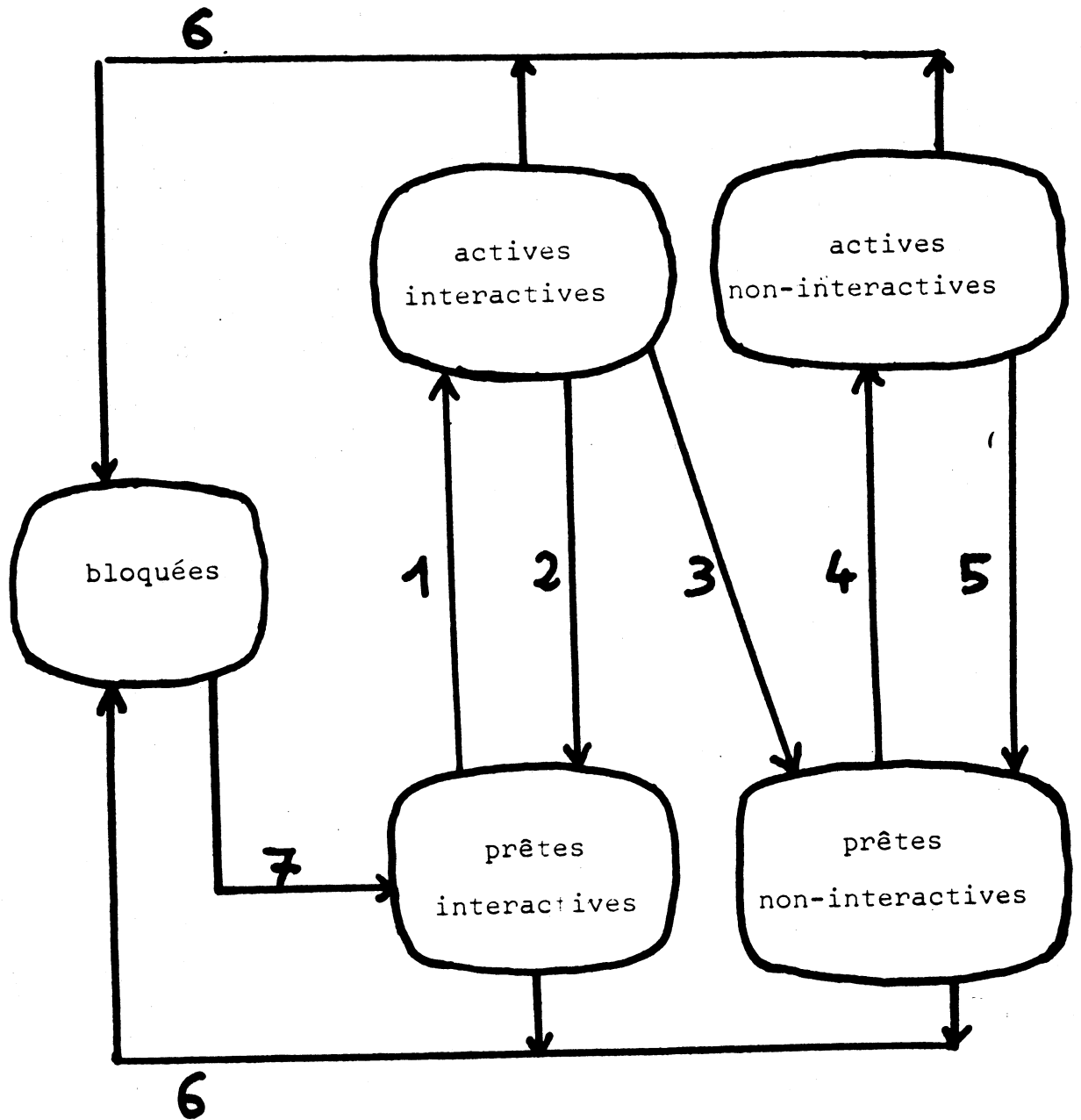
- l'ensemble des machines virtuelles actives et interactives que nous appelons RUSCON

- l'ensemble des machines virtuelles actives et non-interactives que nous appelons USBAT

Ces deux ensembles forment l'ensemble des machines virtuelles actives.

- l'ensemble des machines virtuelles prêtes et interactives que nous appelons RESCON

- l'ensemble des machines virtuelles prêtes et non-interactives que nous appelons RESBAT.



MIGRATION DES MACHINES VIRTUELLES

FIGURE III.2

Ces deux ensembles forment l'ensemble des machines virtuelles prêtes.

- l'ensemble des machines virtuelles bloquées que nous appelons BOS.

Chacun de ces ensembles définit une file d'attente différente. Ce sont les machines virtuelles des ensembles RUSCON et RUSBAT qui constituent le niveau de multiprogrammation ; c'est la somme des working set des machines virtuelles de ces deux ensembles qui indique l'occupation de la mémoire centrale.

2.2 Principes de migration des machines virtuelles

Le nombre de transitions entre les différents ensembles est assez important, aussi pour énoncer les principes que nous avons adoptés nous allons expliciter chacun des chemins qui apparaissent sur la figure III.2.

chemin 6 : il conduit les machines virtuelles dans l'ensemble des machines bloquées ; ces machines peuvent venir de l'un quelconque des quatre autres ensembles.

Cette transition s'effectue dans l'un des cas suivants :

- le programme qui s'exécute dans la machine virtuelle se met

CHAPITRE III

en attente d'une entrée d'informations depuis le terminal

- le programme se met en attente d'une interruption provenant de son horloge virtuelle

- l'utilisateur, ayant appuyé sur la touche "interruption" du terminal, frappe des commandes destinées à l'hyperviseur.

Les working set de ces machines ne doivent plus occuper la mémoire centrale aussi les blocs de mémoire centrale occupés par les working set des machines empruntant l'un des "chemins 6" sont marqués disponibles, et pourront être utilisés pour satisfaire des demandes ultérieures d'emplacements mémoire.

Lorsqu'une machine virtuelle est rangée dans cet ensemble, on ne calcule pas son working set, en effet, celui-ci ne serait d'aucune utilité pour prévoir la demande future de cette machine puisque l'entrée d'informations depuis le terminal, en général, déclenche de nouvelles opérations sans relation avec ce que faisait la machine virtuelle.

Chemin 7 : il conduit les machines virtuelles de l'ensemble bloqué à l'ensemble prêt interactif. Ce passage s'effectue lorsque l'évènement qui matérialise la fin de l'attente est arrivé ou bien lorsque l'utilisateur a fini d'entrer des commandes de l'hyperviseur, la machine virtuelle se trouve alors dans l'état prêt.

Ces machines virtuelles passent dans cet ensemble car l'entrée d'informations depuis le terminal leur attribue l'état interactif. Puisque à leur passage dans l'ensemble bloqué, on n'a pas calculé leur working set, à leur entrée dans cet ensemble on leur attribue une taille de working set par défaut.

Chemin 1 : Il conduit les machines virtuelles interactives de l'ensemble prêt à l'ensemble actif. C'est le contrôleur qui décide de ce passage en appliquant des critères de choix que nous détaillerons ultérieurement. Lorsqu'une machine virtuelle entre dans l'ensemble actif interactif le contrôleur lui attribue un potentiel de temps.

Chemin 2 : Il conduit les machines virtuelles interactives de l'ensemble actif, vers l'ensemble prêt ; ce passage est décidé par la pagination lorsque cet algorithme s'aperçoit qu'il y a une saturation de la mémoire centrale.

Chemin 3 : Il fait passer les machines virtuelles de l'ensemble actif interactif, à l'ensemble prêt non-interactif, c'est le contrôleur qui décide de ce passage. Une machine virtuelle emprunte le "chemin 3" soit lorsque la demande de la machine virtuelle n'était pas une demande interactive (par exemple demande d'assemblage d'un programme), soit lorsque la demande interactive est trop longue pour être traitée avant l'épuisement du potentiel.

CHAPITRE III

Chemin 4 : Il conduit les machines virtuelles non-interactives de l'ensemble prêt à l'ensemble actif comme pour le chemin 1. C'est le contrôleur qui décide de ce passage en appliquant ses critères de choix.

Chemin 5 : Ce chemin fait passer les machines virtuelles non-interactives de l'ensemble actif, à l'ensemble prêt. Les machines virtuelles empruntent ce chemin dans deux cas : lorsque l'algorithme de pagination s'aperçoit qu'il y a saturation de la mémoire centrale, ou bien lorsque ces machines ont épuisé leur potentiel avant d'avoir terminé le travail qui leur avait été demandé.

2.3 Critères de choix du contrôleur

Lorsque le contrôleur cherche à faire passer une machine virtuelle de l'ensemble prêt à l'ensemble actif, son choix va être déterminé par deux critères.

Critère de mémoire : afin de garder un contrôle de l'utilisation de la mémoire centrale, le contrôleur ne fera passer une machine virtuelle de l'ensemble prêt à l'ensemble actif que si le nombre de blocs disponibles en mémoire centrale est supérieur ou égal à la taille du dernier working set calculé (ou attribué par défaut) pour cette machine. Ce dernier working set sert de prévision pour la demande de mémoire de cette machine.

CHAPITRE III

Critère d'équité : le contrôleur essaie d'assurer que le temps d'unité centrale attribué aux machines virtuelles soit équitablement réparti entre ces différentes machines ; pour cela, il calcule, pour chaque machine virtuelle de l'ensemble prêt un coefficient d'équité G . G est le quotient obtenu en divisant le temps passé à attendre dans l'ensemble prêt par le temps passé en exécution, la dernière fois que la machine virtuelle était dans l'ensemble actif.

G est calculé chaque fois que le contrôleur doit choisir une machine virtuelle de l'ensemble prêt, la machine choisie sera celle ayant le coefficient G le plus grand et qui répond aux critères de mémoire. L'application de ces deux critères peut conduire à laisser une machine virtuelle dans l'ensemble prêt pendant un temps assez long parce que la taille du dernier working set calculé était très grande. Pour éviter cela le contrôleur va se servir du coefficient G : lorsqu'il calcule les coefficients de chaque machine, il calcule aussi la valeur moyenne

$$GM = (G_1 + G_2 + \dots + G_N) / N$$

si une machine virtuelle a un coefficient G supérieur à $k \times GM$, c'est cette machine qui sera choisie sans tenir compte du critère de mémoire. Ainsi les machines virtuelles ayant besoin de beaucoup de mémoire centrale pourront tout de même avoir accès à l'unité centrale.

CHAPITRE III

3. DESCRIPTION DE LA MISE EN OEUVRE DES NOUVEAUX ALGORITHMES

La modification des parties de CP-67 effectuant le travail de pagination et du contrôleur représente un changement important dans le système. Lorsque nous avons entrepris d'installer nos modifications, l'hyperviseur n'était plus un système expérimental, mais au contraire un système dont les structures étaient bien définies. Le fait d'introduire nos changements dans un système employé couramment a été très utile pour l'évaluation de nos travaux mais cela nous a posé quelques problèmes.

3.1 Calcul d'un working set, organisation de la mémoire centrale

Lorsqu'une machine virtuelle a épuisé son quantum, le contrôleur calcule la taille du working set qui a été utilisé pendant cette tranche de temps. La fonction qui effectue ce travail utilise pour cela un indicateur qui se trouve dans les "clés de protection mémoire" associées aux blocs de la mémoire centrale : dans les ordinateurs IBM 360, à chaque zone de 2048 octets (ou de 2K) de mémoire centrale est associé un octet de contrôle appelé "clé de protection mémoire" ; dans le cas particulier du 360/67 un bit de cet octet est mis à 1 lorsque la zone de 2K a été référencée (23).

La fonction qui calcule la taille du working set utilise aussi une table de CP-67 : la table de mémoire centrale (cf. I.2.1.3)

CHAPITRE III

dans laquelle, pour chaque bloc de la mémoire centrale, se trouve entre autres indications le "nom" de la machine virtuelle à qui ce bloc est attribué.

La fonction examine chaque entrée de cette table de mémoire centrale pour repérer les pages qui appartiennent à la machine virtuelle qui nous intéressent ; pour chacune de ces pages la fonction détermine si elle a été référencée à l'aide de l'indicateur dans la clé de protection associée.

Si cet indicateur est à 1, alors la page fait partie des pages référencées, et doit être comptabilisée dans la taille du working set. Si l'indicateur est à 0, cela signifie que cette page ne fait plus partie du working set ; dans ce cas la fonction indique, dans la table de pages qui décrit la mémoire de la machine virtuelle, que cette page est invalide. Ainsi, lorsque plus tard la machine virtuelle voudra de nouveau référencer cette page, une interruption pour page manquante se produire.

Les blocs de la mémoire centrale occupés par des pages n'appartenant plus à un working set sont considérés comme libres et peuvent donc être utilisés pour satisfaire les demandes ultérieures provenant des machines virtuelles actives et c'est seulement lorsque le bloc sera demandé que l'on recopiera, éventuellement, son contenu.

CHAPITRE III

Cet ensemble de blocs libres est repéré par une liste ; lorsqu'un bloc est libéré, il est chaîné à la fin de cette liste. Lorsque l'algorithme de pagination a besoin d'un bloc, il prend le premier de la liste. Ainsi les blocs de mémoire centrale sont sélectionnés suivant le principe du moins récemment utilisé ; de plus la pagination tient à jour le compte de blocs libres dans cette liste.

3.2 Traitement d'une interruption pour page manquante

Lors d'un défaut de page, il est possible que la page manquante se trouve toujours dans un bloc de la mémoire réelle, si cette page vient juste d'être retirée du working set, et si le bloc n'a pas été alloué pour contenir une autre page.

Le programme de pagination s'en aperçoit en examinant l'entrée de la table de pages correspondante : si cette entrée contient un numéro de bloc de la mémoire réelle, c'est que la page occupe encore ce bloc. Alors le bloc est enlevé de la liste des blocs libres et est réattribué à la machine virtuelle à qui appartient la page. Pour ce cas particulier, le traitement du défaut de page ne nécessite aucune opération d'entrée-sortie.

Lorsque la page demandée n'est plus dans un bloc libre, il faudra amener cette page à partir du support externe où elle réside. Pour cela il faut choisir un bloc de mémoire centrale

dans la liste des blocs libres ; si cette liste n'est pas vide, c'est le premier emplacement qui est choisi. Le contenu de ce bloc est copié à cet instant sur support externe, si la page a été modifiée par la machine virtuelle à qui elle appartient ; à la suite de cette copie éventuelle la page manquante pourra être lue dans le bloc de la mémoire centrale et ce dernier sera retiré de la liste des blocs libres.

Si la liste de blocs libres est vide, cela signifie que la mémoire centrale est trop utilisée, et que l'on a atteint le seuil à partir duquel, auparavant, naissait le vol de pages. Dans ce cas, conformément au principe que nous avons énoncé (cf III.1) il faut supprimer une des machines virtuelles actives. La "victime" sera choisie ainsi : c'est la première machine virtuelle appartenant à la même file d'attente que celle qui a provoqué l'interruption pour page manquante. Le working set de la "victime" est calculé, les blocs qu'il occupait sont mis dans la liste des blocs libres et la machine virtuelle est mise dans l'ensemble prêt ; on peut remarquer que cette machine suit soit le "chemin 2", soit le "chemin 5" suivant qu'elle est interactive ou non. Il est possible que cette opération ne libère aucun bloc de mémoire centrale (pages verrouillées pour une opération d'entrée-sortie) dans ce cas on réitère le processus. Lorsque l'exploration de la file d'attente où se trouve la machine virtuelle qui a provoqué la faute de page n'a libéré aucun bloc la recherche d'une "victime" se poursuit dans la deuxième file d'attente des machines virtuelles actives.

CHAPITRE III

Il existe pourtant un cas particulier : la seule machine virtuelle active est celle qui a provoqué l'interruption pour page manquante et le working set de cette machine occupe tous les emplacements disponibles en mémoire centrale. Dans ce cas la pagination demande le calcul de la taille du working set utilisé par cette machine ; ainsi ce calcul est effectué avant épuisement du quantum, et permet peut être de libérer quelques blocs. Enfin si cette solution ne donne pas de résultats cette machine virtuelle est retirée de l'ensemble des machines virtuelles actives ce qui a pour effet de libérer tous les blocs qu'elle occupait. De ce fait une machine virtuelle ayant un working set plus grand que la taille de mémoire centrale disponible sera fortement pénalisée.

3.3 Le contrôleur

En règle générale, on peut dire que le contrôleur effectue son travail lorsque CP-67 a fini de traiter une interruption ; c'est lui qui va décider du choix de la prochaine machine virtuelle qui prendra le contrôle de l'unité centrale.

Globalement, le contrôleur a deux rôles distincts : il doit examiner et éventuellement changer l'état de 2 machines virtuelles : celle qui contrôlait l'unité centrale au moment de l'interruption, celle à qui est attachée l'interruption (dans

certains cas c'est la même machine). Il doit choisir ensuite quelle est la machine virtuelle qui lui succèdera.

3.3.1 Détermination de l'état d'une machine virtuelle

Pour déterminer cet état, le contrôleur calcule le temps pendant lequel la machine virtuelle a exécuté des instructions : si elle a épuisé le quantum qui lui était alloué, le contrôleur calcule la taille du working set et place cette machine à la fin de la file d'attente ; de plus il vérifie si la machine virtuelle a épuisé son potentiel dans ce cas, il effectue la libération de tous les blocs de mémoire centrale occupés par ce working set, et fait passer la machine virtuelle dans une des deux files d'attente de l'ensemble prêt suivant ses caractéristiques.

Le contrôleur vérifie aussi si la machine virtuelle n'est pas dans l'état bloqué, ou bien si elle ne perd pas la qualité interactive ; si l'un de ces cas se produit, il transfère la machine virtuelle dans sa nouvelle file d'attente.

3.3.2 Choix d'une machine virtuelle

Dans son algorithme de choix, le contrôleur va faire en sorte de favoriser les machines virtuelles interactives ; de plus

CHAPITRE III

pour utiliser au mieux la ressource unité centrale, il cherchera à donner le contrôle de cette ressource aux machines virtuelles de l'ensemble actif avant de chercher à remplir la mémoire centrale.

Le contrôleur effectue ce choix en appliquant les critères qui suivent, dans l'ordre où ils sont énumérés, dès qu'un critère a permis la sélection d'une machine, il arrête l'examen des différentes files d'attente et passe le contrôle de l'unité centrale à cette machine.

- si la machine virtuelle qui a été interrompue peut toujours prendre le contrôle de l'unité centrale, c'est elle qui est choisie
- si la file d'attente de l'ensemble actif interactif n'est pas vide, c'est la première machine virtuelle de cette file qui est choisie
- si la file d'attente de l'ensemble actif, non-interactif n'est pas vide c'est la première machine virtuelle de cette file qui est choisie. Ainsi dans ces deux ensembles le choix se fait selon un ordre : premier arrivé, premier servi.
- si la file d'attente de l'ensemble prêt, interactif n'est pas vide le contrôleur choisira la première machine virtuelle qui répond au critère de mémoire centrale.

- si la file d'attente de l'ensemble prêt, non-interactif n'est pas vide, le contrôleur calcule pour les machines virtuelles de cet ensemble le coefficient d'équité : G et choisira la machine virtuelle qui satisfait aux deux critères : critère de mémoire et critère d'équité.
- si aucun des critères ne sélectionne de machine, l'unité centrale est mise en attente.

3.3.3 Remarques concernant le contrôleur

Lorsque nous avons installé ce nouveau contrôleur dans CP-67, il nous a fallu définir les valeurs des différents paramètres de ce composant ; les études faites auparavant sur le comportement des programmes dans un environnement paginé (15), et les constatations faites à partir de mesures nous ont permis de choisir la valeur de ces paramètres.

Pour les machines virtuelles actives, non interactives le quantum a été fixé à 30 millisecondes et le potentiel à 1 seconde.

Pour les machines virtuelles actives, interactives le quantum et le potentiel ont été fixés à 30 millisecondes.

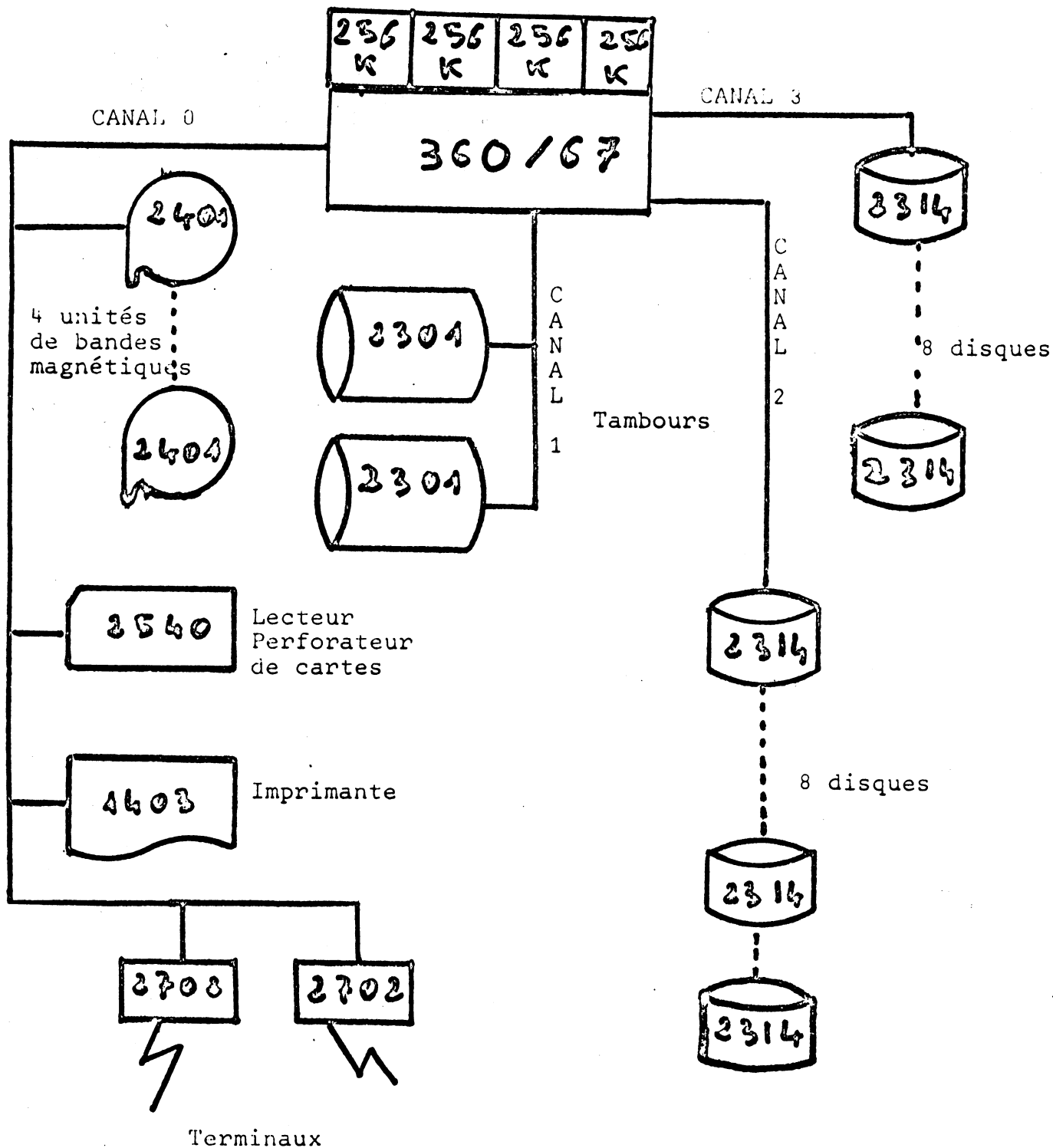
Le temps pris par l'hyperviseur CP-67 pour simuler les

CHAPITRE III

différentes opérations d'une machine virtuelle n'est pas compté dans le quantum et dans le potentiel : ces deux variables accumulent le temps pendant lequel une machine virtuelle exécute des instructions.

Pour appliquer le critère de mémoire, nous avons rencontré certaines difficultés : d'une part CP-67, dans certains cas, ne met pas à jour le nombre de blocs libres lorsqu'il prend un de ces blocs, d'autre part lorsqu'une machine virtuelle vient d'accéder à l'ensemble actif, il lui faut un certain temps avant d'accumuler son working set en mémoire centrale. Pour résoudre le premier problème, le contrôleur doit calculer le nombre de blocs libres en mémoire centrale chaque fois qu'il doit faire passer une machine virtuelle non-interactive de l'ensemble prêt à l'ensemble actif.

Pour résoudre le deuxième problème, il doit enlever à ce nombre de pages libres un nombre de blocs qui représentent le nombre de pages que les machines virtuelles, dans leur premier quantum, n'ont pas pu encore acquérir. Pour ranger les informations particulières à notre contrôleur, nous avons ajouté un bloc supplémentaire à la description de chaque machine virtuelle, ces blocs sont chaînés entre eux et constituent les différentes files d'attente.



CONFIGURATION EN OCTOBRE 1971

FIGURE III.3

CHAPITRE III

4. EVALUATION DES MODIFICATIONS

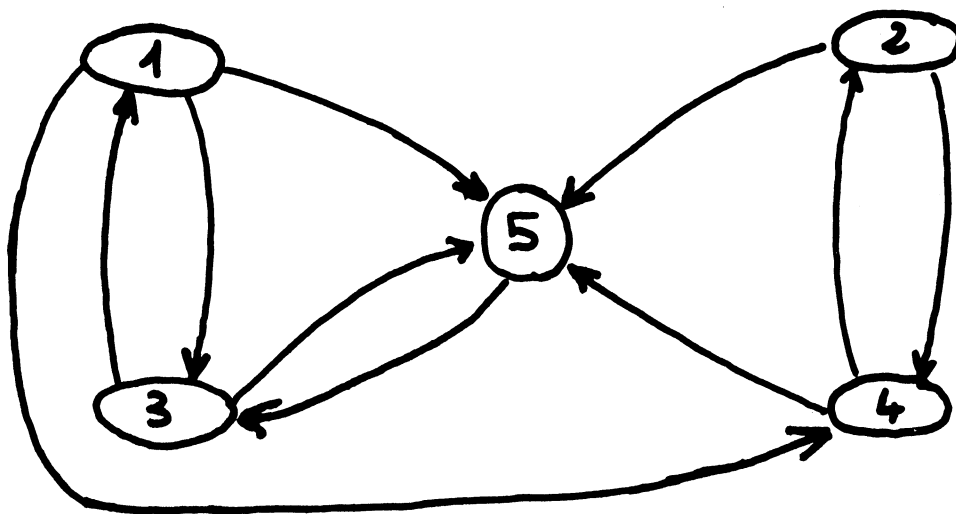
La configuration de l'ordinateur du CICG a été augmentée durant la période de réalisation de nos algorithmes ; les impératifs d'exploitation ne nous ont pas permis de mesurer nos modifications avec l'ancienne configuration, qui présentait (ch. I) un fonctionnement dégradé dans certaines conditions.

Les changements apportés sont les suivants (fig. III.3) :

- La mémoire centrale est passée de 512K octets à 1024K octets
- 20 unités de disques répartis sur deux canaux au lieu de 16.

L'augmentation de la taille mémoire avait rendu le phénomène de vol de pages beaucoup moins important. Toutefois, afin de pouvoir évaluer les modifications que nous avons introduites, nous avons utilisé l'outil de mesure en comparant les résultats obtenus pendant des périodes de 3 mois : les mesures se rapportant à CP-67 avant modifications ont été faites pendant les mois d'octobre, novembre et décembre 1971 ; les mesures faites après l'introduction de nos modifications portent sur les mois de janvier, février et mars 1972 ; la configuration de l'ordinateur étant identique pour ces 2 périodes.

Pendant ces périodes, le nombre maximum de machines virtuelles connectées au système à un instant donné a atteint 55, la moyenne



- Ensemble 1 : Machines virtuelles actives, interactives
- Ensemble 2 : Machines virtuelles actives, non-interactives
- Ensemble 3 : Machines virtuelles prêtes, interactives
- Ensemble 4 : Machines virtuelles prêtes, non-interactives
- Ensemble 5 : Machines virtuelles bloquées

TRANSITIONS ENTRE ENSEMBLES

FIGURE III. 4

CHAPITRE III

de ce nombre sur les 2 périodes étant de 34. Rappelons que lorsque l'hyperviseur est en fonction, le relevé des différentes variables s'effectue toutes les 100 secondes ; on obtient ainsi un certain nombre de résultats, soit directement à partir des valeurs des variables prélevées, soit par calcul en combinant plusieurs de ces valeurs.

4.1 Etude des files d'attente

Le dispatcheur que nous avons réalisé travaille sur cinq ensembles distincts de machines virtuelles, à chacun de ces ensembles est liée une file d'attente, nous rappelons dans la figure III.4 toutes les transitions possibles entre les différents ensembles. Nous nous sommes intéressés au nombre de transferts entre ces ensembles ; soit la matrice R dont un élément ij donne le nombre de transferts par seconde de l'ensemble i vers l'ensemble j .

La valeur moyenne des éléments de R est la suivante :

CHAPITRE III

	1	2	3	4	5
1	0	0	0	0,3	2,5
2	0	0	0	0,9	0,3
3	2,8	0	0	0	0
4	0	1,2	0	0	0
5	0	0	2,8	0	0

Moyenne de R

De même, nous avons pu calculer le vecteur V dont les éléments v_j représentent la longueur des files d'attente. Les valeurs obtenues sont les suivantes :

1	2	3	4	5
1,1	4,2	0,05	4,0	24,4

Moyenne de V

En utilisant les valeurs obtenues pour R et V, nous calculons la valeur moyenne du temps de résidence d'une machine virtuelle

CHAPITRE III

dans les différentes files d'attente ; nous construisons le vecteur T dont les éléments t_j sont obtenus par la formule :

$$t_j = \frac{v_j}{\sum_{i=1}^5 r_{ij}}$$

t_j représente le nombre de secondes de temps réel pendant lequel une machine reste dans un ensemble

	1	2	3	4	5
	0,42	5,04	0,02	5,35	9,5

Moyenne de T

Nous pouvons maintenant discuter certaines de ces valeurs :

- La prise en compte d'une requête interactive se fera lorsque la machine virtuelle aura attendu dans l'ensemble prêt interactif, puis lorsque cette machine virtuelle aura eu son temps d'exécution dans l'ensemble actif interactif, ceci demande donc un temps égal à $t_1 + t_3$ soit 440 millisecondes.

- Un terminal qui émet une requête place la machine virtuelle dans l'ensemble prêt interactif, où elle attendra en moyenne 20 millisecondes avant d'être admise dans l'ensemble actif interactif.

CHAPITRE III

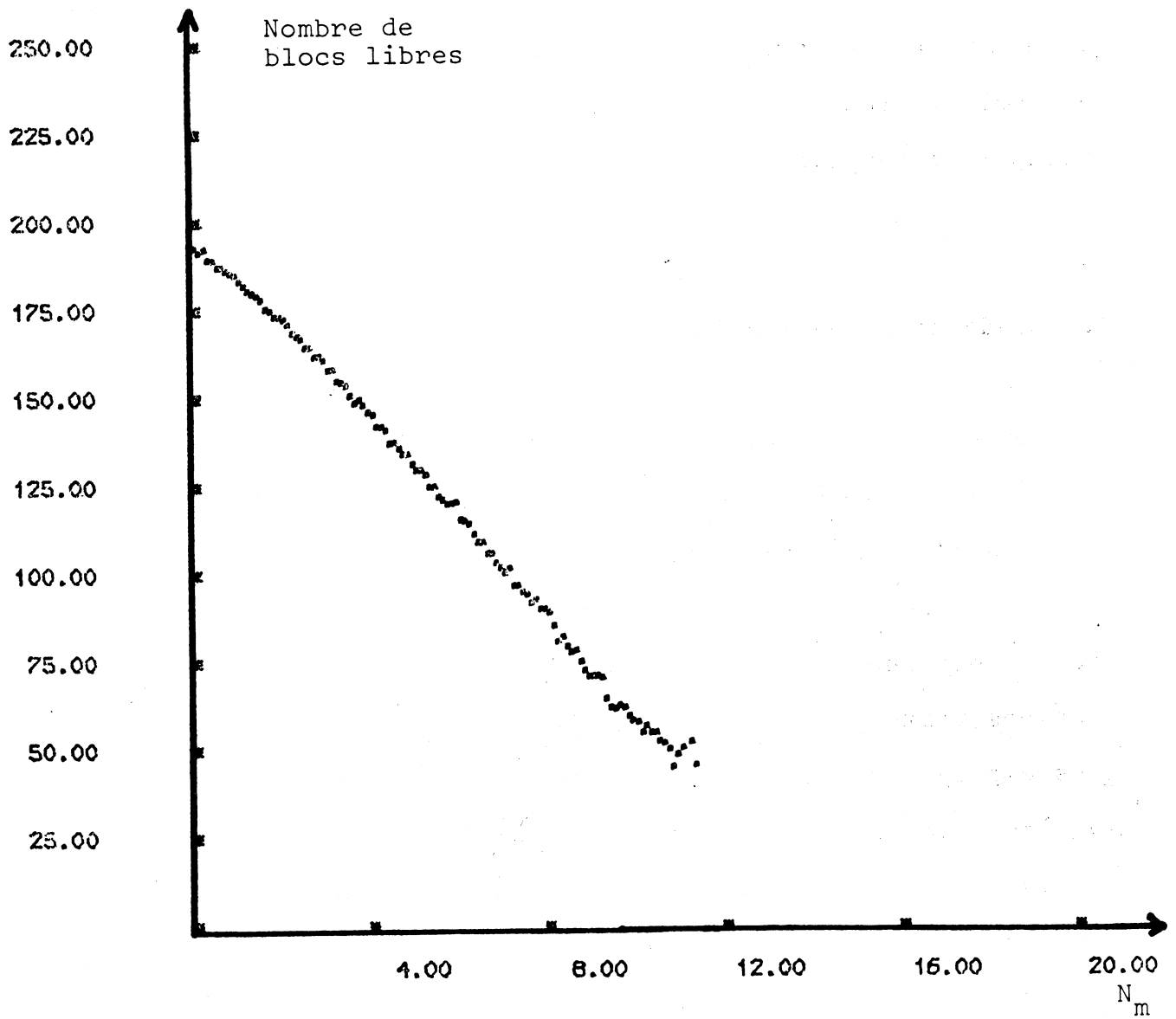
- Lorsqu'on examine la matrice R, on peut étudier les transitions de l'ensemble actif interactif (1), vers l'ensemble bloqué (5). La valeur $r_{15} = 2,5$ indique qu'il y a 2,5 transitions par seconde du premier ensemble vers le second, la somme des valeurs de la colonne 1 de la matrice R indique qu'il y a 2,8 machines qui arrivent dans l'ensemble actif interactif, par seconde. On remarque donc que 87 % de ces machines retournent dans l'ensemble bloqué c'est à dire que ces machines qui demandaient une opération interactive ont été satisfaites avant épuisement du potentiel (ce potentiel est égal à 30 millisecondes).

- Les 13 % représentent les machines virtuelles qui ont transité de l'ensemble actif interactif (1), vers l'ensemble prêt non interactif (4) ; cela signifie soit que la requête interactive demandait plus de 30 millisecondes de temps d'exécution, soit qu'il ne s'agissait pas de requêtes interactives.

- Il n'y a pas de transitions de l'ensemble actif interactif (1), vers l'ensemble prêt interactif (3), bien que la possibilité d'une telle transition existe. Ceci est dû au fait que le contrôleur n'a pas eu à évincer des utilisateurs interactifs pour manque de place en mémoire centrale.

Le temps de résidence dans l'ensemble actif non-interactif, t_2 est de 5,04 secondes et le temps moyen d'attente pour accéder à

INSTITUT DE MATHEMATIQUES APPLIQUEES DE GRENOBLE



NOMBRE DE BLOCS LIBRES

FIGURE III.5

CHAPITRE III

cet ensemble t_4 est de 5,35 secondes, autrement dit pour qu'une machine virtuelle puisse avoir son potentiel dans l'ensemble non-interactif, c'est à dire pour pouvoir exécuter des instructions pendant une seconde, il lui faut 10 secondes de temps réel.

- Le temps de résidence dans l'ensemble bloqué $t_5 = 9,5$ indique qu'en moyenne une machine virtuelle émet une requête depuis son terminal toutes les 9 secondes, c'est ce qui est parfois appelé : le temps de réflexion au terminal.

4.2 Etude de la pagination

La courbe de la figure III.5 représente le nombre de blocs libres en fonction du niveau de multiprogrammation. Cette courbe a l'allure d'une droite ; le nombre de blocs disponibles en mémoire centrale est en moyenne de 198 blocs, la pente de la droite est en valeur absolue de 15. Dans les différentes machines virtuelles connectées à l'ordinateur, le système le plus employé est le système CMS (Cambridge Monitor System) ; ce système a la particularité de partager 6 pages entre tous les utilisateurs, ces 6 pages ne sont pas comptabilisées dans les working set de chaque machine virtuelle, ainsi on peut dire que le working set moyen que nous avons observé pour les machines virtuelles est de $15 + 6 = 21$ pages. Cette taille de working set correspond bien aux mesures que nous avons obtenues lorsque

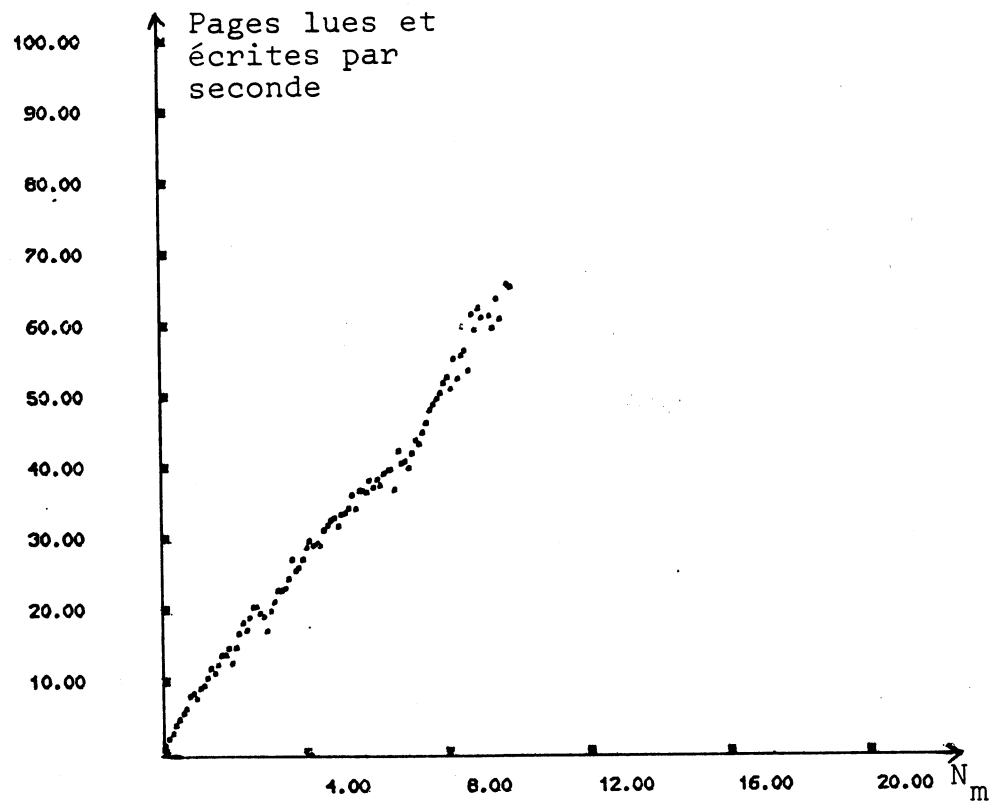
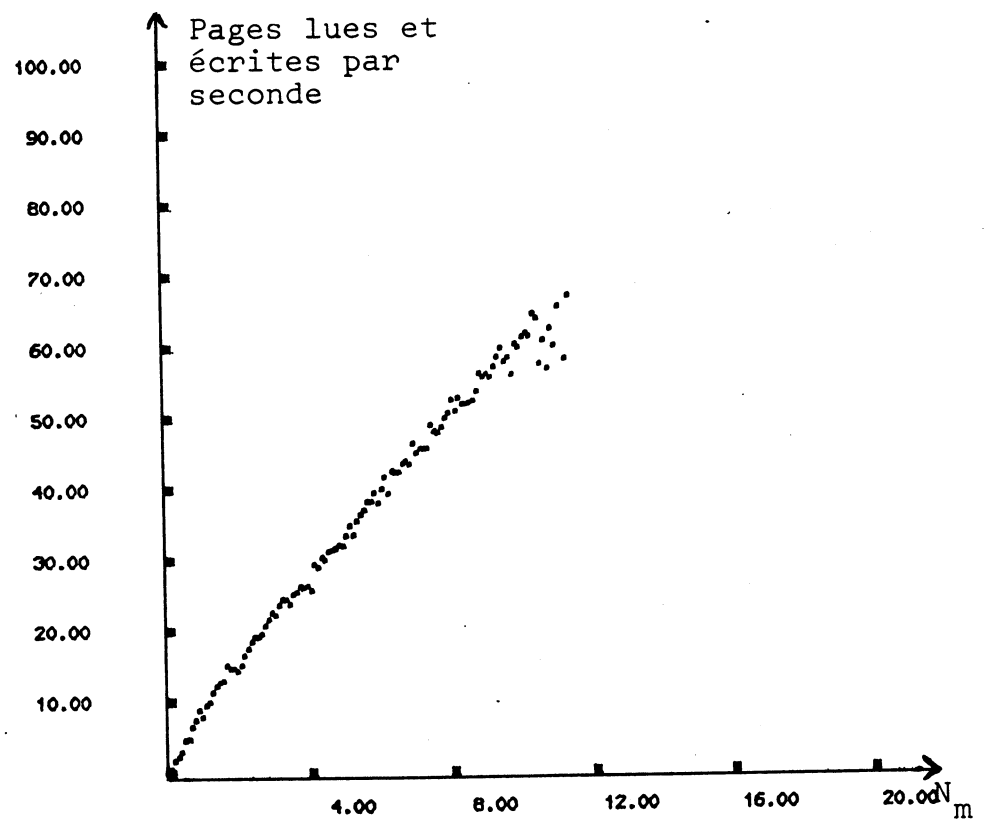


FIGURE III.6

NOMBRE DE PAGES LUES ET ECRITES

FIGURE III.7



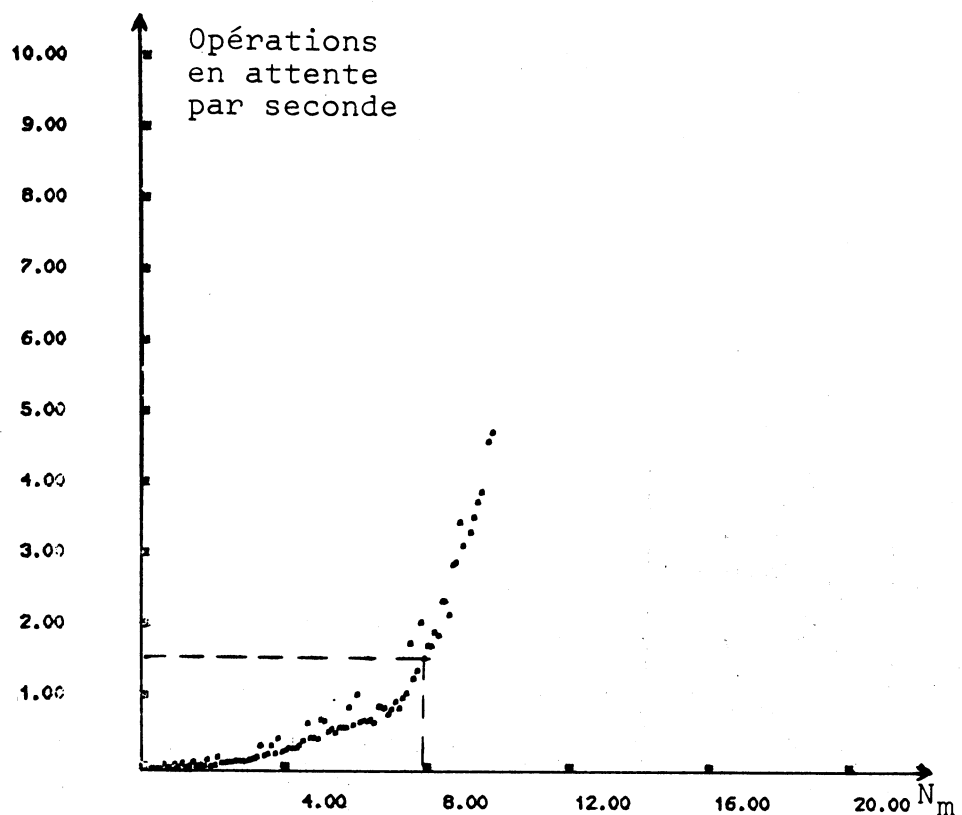
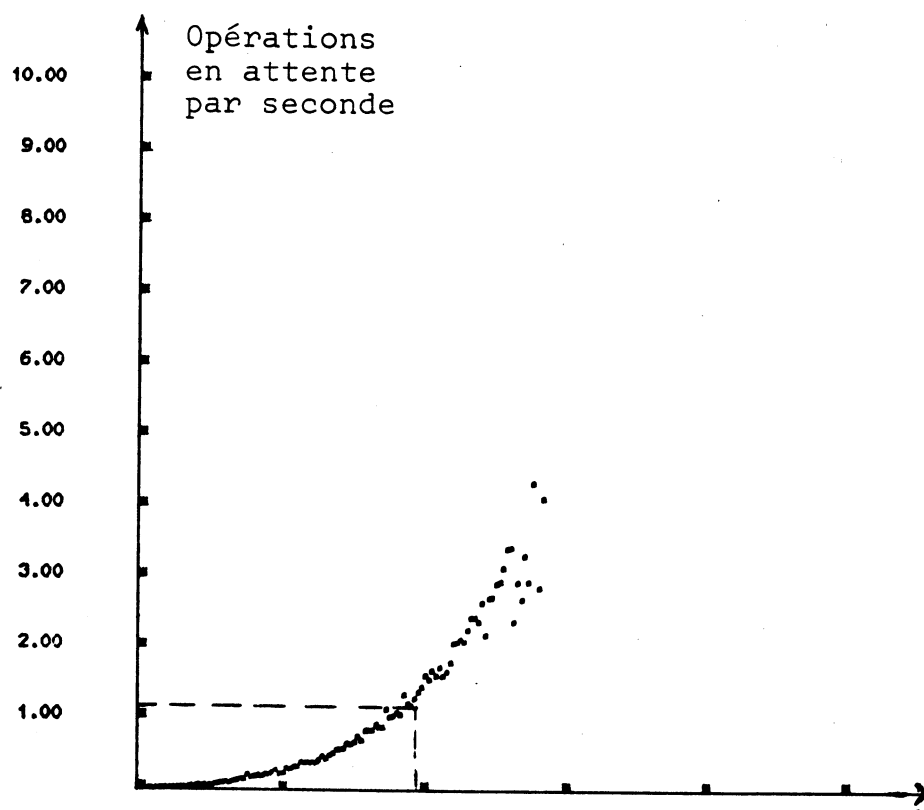


FIGURE III.8

ENTREES SORTIES DE PAGINATION EN ATTENTE

FIGURE III.9



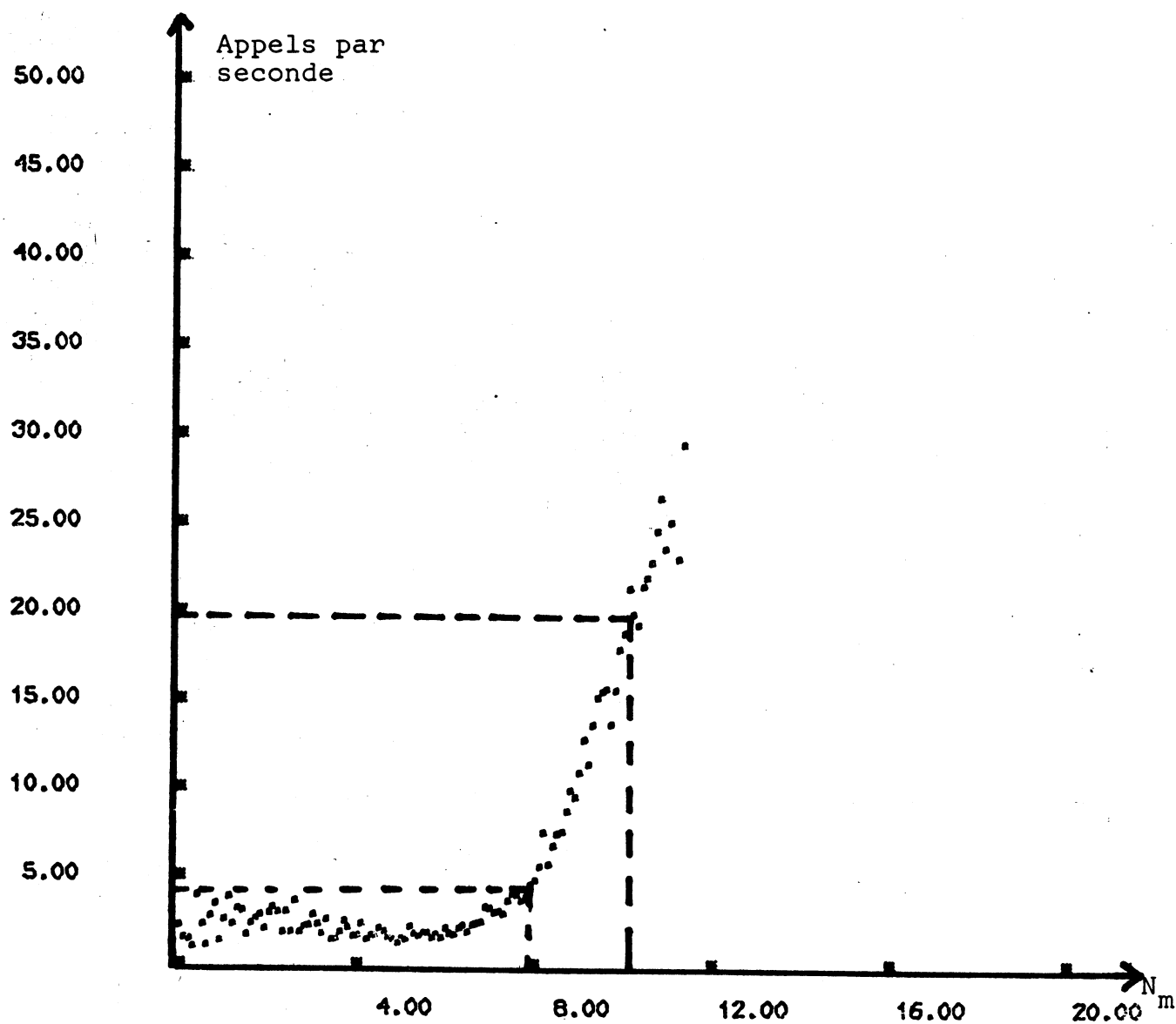
CHAPITRE III

nous avons étudié le comportement des composants du système CMS. Nous savons que la taille des working set peut varier rapidement en cours du temps, toutefois lorsque nous observons l'hyperviseur à des intervalles de l'ordre de 100 secondes, il est possible de parler de working set moyen pour les différentes machines virtuelles qui se partagent la mémoire centrale.

Nous pouvons comparer l'activité de pagination entre l'ancien hyperviseur et notre hyperviseur modifié. La figure III.6 présente le nombre total de pages lues et écrites, par seconde, en fonction du niveau moyen de multiprogrammation (N_m) pour l'ancien système ; la figure III.7 présente la même courbe pour le nouveau système ; jusqu'à un niveau de multiprogrammation de 8 ces deux courbes coïncident, au-delà il apparaît une légère divergence : la courbe de l'ancien système présente un changement de pente, la pagination augmente plus rapidement lorsque N_m dépasse la valeur 8.

Ceci se confirme si nous examinons les courbes qui indiquent le nombre d'opérations d'entrée-sortie en attente de traitement sur le canal réservé à la pagination : la figure III.8 présente cette courbe pour l'ancien hyperviseur, la figure III.9 présente cette courbe pour le système modifié. Comme précédemment pour N_m inférieur à 8 les 2 courbes coïncident, au-delà de cette valeur de N_m la courbe de l'ancien système s'accroît plus rapidement que celle du nouveau.

INSTITUT DE MATHEMATIQUES APPLIQUEES DE GRENOBLE



NOMBRE D'APPELS DE LA FONCTION
CALCUL DE BLOCS LIBRES

FIGURE III.10

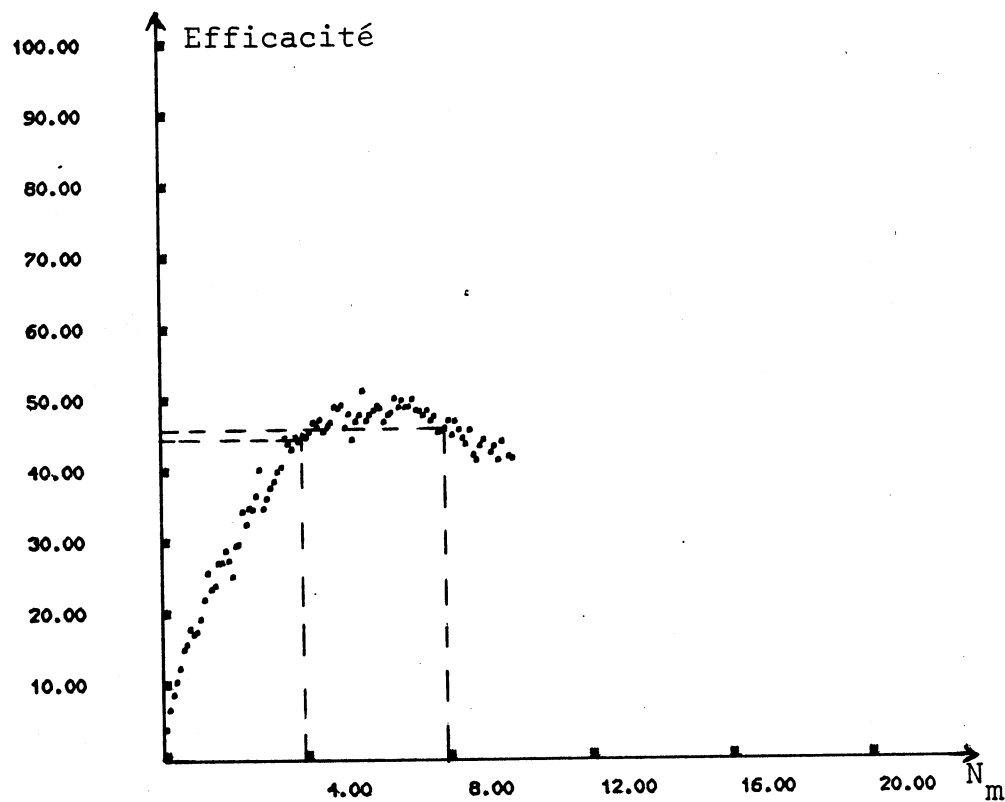
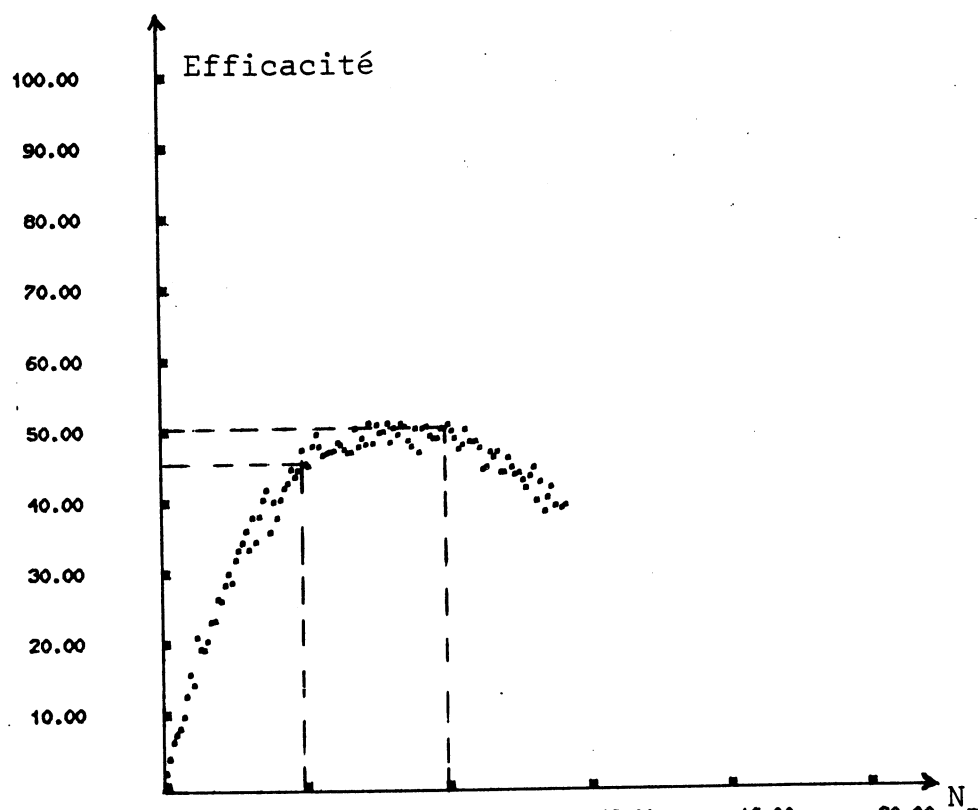


FIGURE III.11

EFFICACITE

FIGURE III.12



4.3 Etude de l'utilisation de l'unité centrale

Nous pouvons aussi comparer les temps d'utilisation de l'unité centrale entre l'ancien système et notre système modifié. Mais auparavant, examinons la courbe qui décrit l'activité de la fonction de calcul du nombre de blocs libres en mémoire centrale, fonction que nous avons été obligés d'ajouter. La figure III.10 présente le nombre d'appels par seconde de cette fonction suivant N_m . Nous voyons que jusqu'à ce que $N_m = 8$ le nombre d'appels est d'environ 3 par seconde, ce qui n'est pas très important ; au-delà de cette valeur le nombre d'appels croît très rapidement, par exemple pour $N_m = 10$, le nombre d'appels est de 20 par seconde. Ceci nous montre que lorsque N_m dépasse 8, le temps pris par le contrôleur pour choisir la prochaine machine virtuelle va augmenter de façon importante.

Nous pouvons maintenant examiner les 2 courbes qui représentent l'efficacité c'est à dire le pourcentage du temps total de l'unité centrale passé en mode problème. Les figures III.11 et III.12 donnent cette efficacité en fonction du niveau moyen de multiprogrammation, respectivement pour l'ancien et pour le nouveau système ; pour les valeurs de N_m inférieures à 4, les 2 courbes sont croissantes et sont sensiblement confondues. Pour N_m compris entre 4 et 8 ces courbes atteignent leur valeur maximale ; au-delà de $N_m = 8$ les 2 courbes décroissent. On peut remarquer toutefois que la courbe du système modifié décroît moins rapidement. Cette diminution du temps passé en mode

CHAPITRE III

problème est dûe à plusieurs facteurs dont les 2 plus importants sont :

- l'accroissement du temps passé dans le contrôleur pour choisir une machine virtuelle

- la formation de files d'attente pour les demandes d'entrée-sortie aussi bien sur les disques que sur les tambours.

Ainsi, à l'aide de l'outil de mesure présenté au chapitre I, nous avons pu évaluer le comportement des algorithmes du contrôleur et de la pagination dans un environnement d'utilisation normale. De cette série de résultats, nous pouvons constater que dans le système modifié se comporte mieux que l'ancien lorsque l'on approche la saturation. Toutefois, nous n'avons pas pu comparer les temps de réponses des deux systèmes mais des sondages que nous avons effectués auprès des utilisateurs nous ont montré que le fait d'avoir introduit un critère d'équité dans le choix des machines virtuelles, a diminué globalement le temps d'attente de chaque utilisateur devant son terminal.

CONCLUSION

Dans ce mémoire, nous avons donc décrit un travail continu qui a été effectué pendant une période de 2 ans : réalisations d'un outil de mesure destiné à mieux connaître le fonctionnement de l'hyperviseur CP-67 dans son environnement normal d'exploitation, puis étude et mise en oeuvre de nouveaux algorithmes dans cet hyperviseur, enfin évaluation des modifications apportées.

L'outil de mesure fonctionne toujours et il est employé pour rechercher de nouvelles possibilités d'amélioration, et pour contrôler le bon fonctionnement de l'hyperviseur. Nous avons pu comparer notre réalisation avec d'autres travaux équivalents (2, 9) ; dans tous les cas, nous retrouvons les mêmes caractéristiques. Le résultat le plus important à partir de cet

CONCLUSION

outil a été la mise en évidence du phénomène d'effondrement. Les différentes courbes présentées ont été obtenues à partir des valeurs moyennes prélevées pendant toute la durée du fonctionnement de l'hyperviseur, pour une étude plus précise, on pourrait se servir des fichiers de statistiques en sélectionnant les intervalles de fonctionnement où le système est fortement chargé.

La mise en oeuvre des nouveaux algorithmes du contrôleur et de la pagination nous a permis de montrer qu'il est possible de baser ces algorithmes sur le modèle de working set. Toutefois, l'application stricte de ce que nous avons appelé le "principe de working set" : interdiction totale du vol de pages, peut conduire à augmenter le temps pris par le contrôleur lorsqu'il effectue le choix d'une machine virtuelle.

De cette réalisation, nous retiendrons une règle qui nous semble essentielle dans la conception d'un algorithme de gestion de l'unité centrale : il faut que les parties du système pagination et contrôleur, échangent des informations afin de gérer le mieux possible la mémoire centrale.

REFERENCES

1. Arden, B.W. and Boettner, D.
"Measurement and performance of a multiprogramming system"
Second symposium on operating systems principles, Princeton University, 1969
2. Bard Y.
"CP-67 measurement and analysis, overhead and throughput"
Workshop on system performance evaluation
Harvard University 1971
3. Belady, L.A.
"A study of replacement algorithms for a virtual storage computer"
IBM systems J. 5,2 1966
4. Bellot, M. et al
"Systèmes de programmation générateurs de machines virtuelles"
Cours polycopié, Grenoble 1973
5. Campbell, D.J. and Heffner, W.J.
"Measurement and analysis of large operating systems during system development"
FJCC 1968
6. Cantrell, H.N. and Ellison, A.L.
"Multiprogramming system performance measurement and analysis"
SJCC 1968
7. De Meis, W.M. and Weizer, N.
"Measurement and analysis of a demand paging time sharing system"
Proceedings of the 24th National Conference, ACM, 1969
8. Denning P.J.
"Resource allocation in a multiprocess computer system"
Tech report MAC-TR-50, MIT project MAC
9. Doherty, W.J.
"Scheduling TSS/360 for responsiveness"
Proceedings AFIPS 1970, FJCC

10. Dupuy, J.P.
"Description, fonctionnement et utilisation du Basic Counter Unit"
Note technique, Grenoble 1971
11. Hans, C.
"Contribution à l'architecture de mécanismes élémentaires pour certains systèmes générateurs de machines virtuelles"
Thèse d'état, Grenoble 1973
12. Karush, A.
"Two approaches for measuring the performance of time sharing systems"
Second symposium on operating system principles
Princeton University, 1969
13. Kimbleton, S.R. and Moore, O.G.
"A probabilistic framework for system performance evaluation".
Workshop on system performance evaluation
Harvard University 1971
14. Morganstein, S.J., Winograd J. and Herman, R.
"SIM/61 - A simulation measurement tool for a time shared demand paging operating system".
Workshop on system performance evaluation
Harvard University, 1971
15. Rodriguez-Rosell, J.
"Experimental data on how program behavior affects the choice of scheduler parameters"
Third symposium on operating system principles
Stanford University, 1971
16. Rodriguez-Rosell, J. and Dupuy, J.P.
"The evaluation of a time-sharing page demand system"
Proceedings AFIPS 1972, SJCC
17. Rodriguez-Rosell, J. and Dupuy, J.P.
"The design, implementation and evaluation of a working set dispatcher"
CACM, April 1973

18. Saltzer, J.H. and Gintell, J.W.
"The instrumentation of Multics"
Second symposium on operating systems principles
Princeton University, 1969
19. Scherr, A.L.
"Time-sharing measurement"
Datamation, April 1966
20. Schwetman, H.D. and de Line, J.R.
"An operational analysis of a remote console system"
Proceedings AFIPS 1969, SJCC
21. Wallace, V.L. and Mason, D.L.
"Degree of multiprogramming in page-on-demand system"
CACM, June 1969
22. CP Program logic manual
Form GY20-0590-0
IBM Corp. Technical publications dept.
23. IBM System/360 Model 67. Functional characteristics.
Form A27-2719
IBM Corp. Technical publications dept.

