

UNIVERSITE DE GENEVE
CENTRE UNIVERSITAIRE
D'INFORMATIQUE

FACULTE DES SCIENCES

MEMOIRE DE DIPLOME EN INFORMATIQUE

OPTIMISATION DE L'ACCES AUX DISQUES
POUR LES CDC 6000 INSTALLEES AU CERN

OCTOBRE 1977

MICHEL KUN

REMERCIEMENTS

Je tiens tout d'abord à remercier le professeur B. Levrat ainsi que les membres et collaborateurs du CUI du soutien qu'ils m'ont accordé. Mes remerciement vont tout particulièrement à Mme L. Noël qui a bien voulu accepter de dactylographier ce mémoire.

Je tiens également à exprimer ma reconnaissance au groupe système du CERN et plus particulièrement à J.D. Blake qui a toujours répondu à mes questions avec la plus grande amabilité. Enfin, je remercie tout spécialement F. Königshofer qui m'a constamment assisté de ses précieux conseils.

Table des matières

I INTRODUCTION (7 pages)

- A) Organisation de l'informatique au CERN I/1
- B) Nouvelle situation et liaisons entre ordinateurs I/3
- C) Implémentation prévue du réseau I/5

II PRESENTATION DU PROBLEME POSE (8 pages)

- A) Quelques éléments sur le système d'exploitation
des CDC 6000 en usage au CERN : SCOPE 3.4 II/1
- B) Quelques éléments sur le fonctionnement du NIP II/6
- c) Présentation du problème posé II/8

III METHODOLOGIE (8 pages)

- A) Méthode de mesure pour les périodes calmes III/3
- B) Méthode de mesure pour les périodes chargées III/7

IV IMPLEMENTATION (12 pages)

- A) Contrôle des canaux et des disques en activité IV/1
- B) Contrôle des disques utilisée IV/3
- C) Réalisation des mesures proprement dites IV/4
 - C1) Réalisation de la première situation : R IV/5
 - C2) Réalisation de la deuxième situation : CH IV/8
- D) Réalisation des graphiques et histogrammes IV/12

V MESURES (10 pages)

- A) Mesures pendant les périodes calmes V/1
- B) Mesures pendant les périodes chargées V/7

VI CONCLUSION (1 page)

VII ANNEXE I (3 pages)

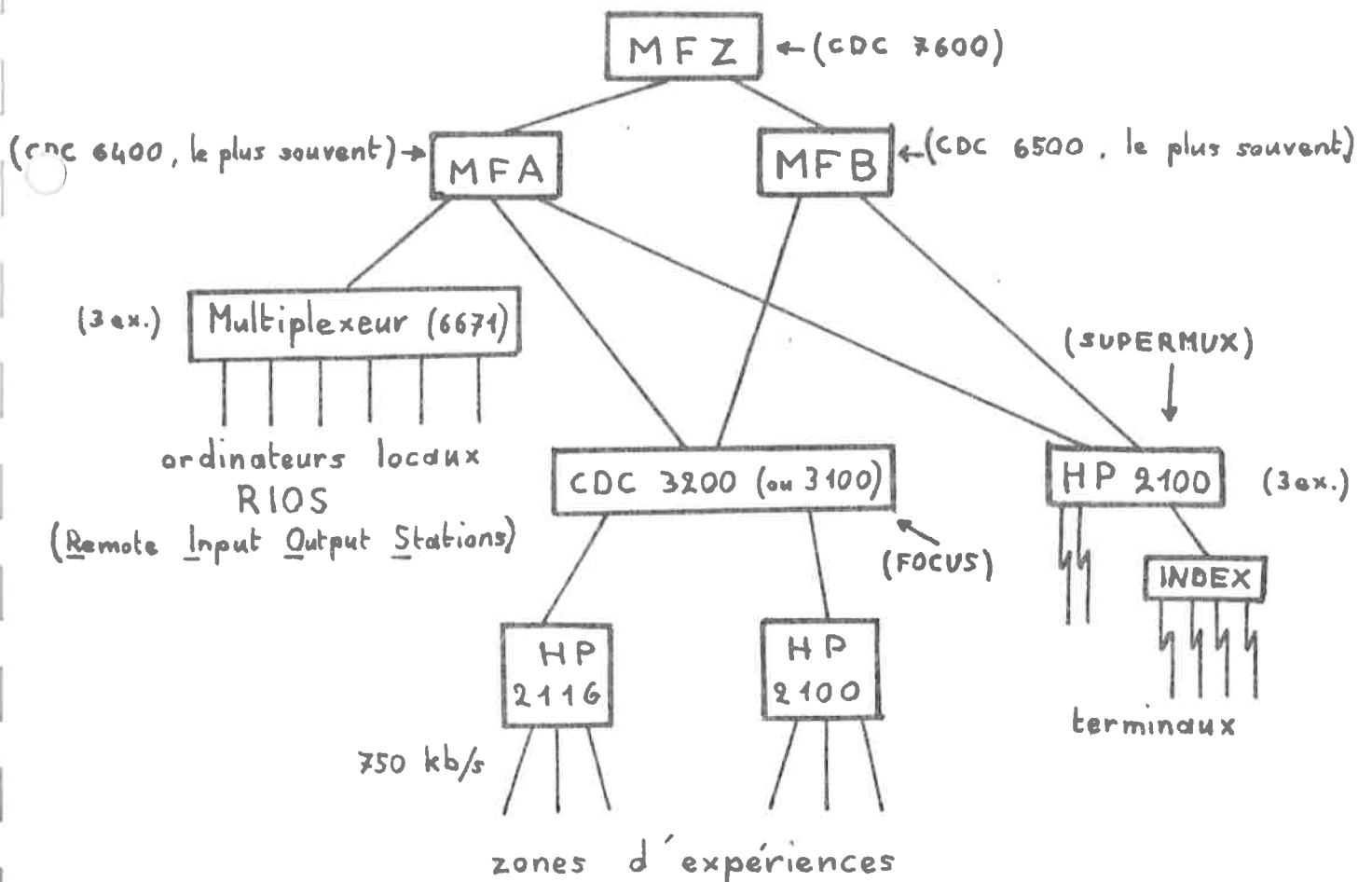
VIII REFERENCES (1 page)

I. INTRODUCTION

Ce travail a été effectué dans le cadre du réseau de transmission de données actuellement en cours de réalisation au CERN. Il s'agissait d'évaluer les différentes méthodes d'accès aux disques en vue d'optimiser le réseau sur ce point.

A. Organisation de l'informatique au CERN

Actuellement, le CERN compte environ 200 ordinateurs installés sur son site. La plupart d'entre eux sont des mini-ordinateurs, principalement des HP 2100, des Norsk Data Nord 10 et des PDP 11. Ces mini-ordinateurs sont utilisés sur les lieux d'expériences pour contrôler différents processus ainsi que pour stocker les données sur bandes magnétiques. Une autre partie de ces mini-ordinateurs est utilisée pour effectuer les liaisons avec les 6 ordinateurs importants que compte le CERN : une PDP 10, une CII 10070, une IBM 370'168 (nouvellement installée), une CDC 6400, une CDC 6500 et une CDC 7600. C'est sur ces trois derniers ordinateurs, reliés entre eux, que s'effectue l'essentiel du travail informatique. Ils se trouvent dans le centre de calcul électronique du CERN; en voici leur configuration :



Deux principaux groupes d'utilisateurs emploient les ordinateurs du CERN :

- 1) Les physiciens expérimentateurs qui cherchent à analyser le plus rapidement possible des échantillons d'une expérience en cours, analyses trop complexes pour être réalisées dans des temps raisonnables par des mini-ordinateurs. On cherche donc à obtenir la vitesse maximum qui se situe pour une série d'équipements autour de 2-3 Mbits/s. C'est dans ce but que FOCUS a été réalisé il y a plus de 8 ans déjà. FOCUS permet un accès multiple (une dizaine) à la CDC 3100 qui possède une capacité de disque limitée. La CDC 3100 joue le rôle de mémoire tampon ("buffer") entre les ordinateurs centraux et les machines locales qui procèdent à l'acquisition de données. Des échantillons de données sont accumulés sur un disque de la CDC 3100. Puis, à certains moments, des fichiers (programmes + données) sont transférés à haute vitesse de la CDC 3100 à la queue d'entrée des ordinateurs centraux où ils sont exécutés avec une grande priorité. Ensuite, les résultats sont retransmis à FOCUS qui les dirige vers les expérimentateurs. C'est ainsi que ces derniers contrôlent la bonne marche de leurs expériences dont les données stockées sur bandes magnétiques par des mini-ordinateurs de contrôle, seront par la suite traitées sur les ordinateurs centraux.
- 2) Tous les programmeurs, y compris les non professionnels, qui désirent préparer des programmes, les soumettre aux ordinateurs centraux, manipuler des fichiers, etc. ~~ils~~ utilisent essentiellement deux groupes d'équipement :
 - a) les ordinateurs locaux ("RIOS stations") ou les ordinateurs de stations extérieures reliées par les lignes PTT. Il y en a environ une cinquantaine dont la vitesse typique est de quelques dizaines de Kb/s.
 - b) les terminaux, au nombre de quelques centaines, qui ont une vitesse liée au temps de réaction de l'homme : 100 b/s. → 9600 b/s. C'est le système SUPERMUX qui

Que signifie RIOS ?

permet la liaison entre les terminaux et les ordinateurs centraux. Ce système est un concentrateur/multiplexeur frontal pour consoles asynchrones opérant dans un mini-ordinateur (un HP 2100). Chacun des HP peut contrôler jusqu'à 36 consoles. Actuellement, 2 HP contrôlent 60 portes : donc 60 terminaux peuvent être actifs simultanément.

B. Nouvelle situation et liaisons entre ordinateurs

Au CERN, les connections entre ordinateurs, terminaux et expériences ont été élaborées indépendamment les unes des autres au cours des années : par exemple, les connections entre les expériences et les ordinateurs centraux réalisées par FOCUS (cf. ci-dessus), celles entre des mini-ordinateurs et la CII 10070, celles entre les RIOS et les ordinateurs centraux, celles entre les terminaux et les ordinateurs centraux réalisées par SUPERMUX (cf. ci-dessus) et celles reliant entre eux les nombreux ordinateurs du système de contrôle du SPS.

Les progrès techniques dans la réalisation de tels systèmes, aussi bien du point de vue hardware que software, ainsi que la mise en place du SPS ont conduit à recommander l'installation d'un réseau - dont le but principal est la transmission de données - en vue de rationaliser, de standardiser et d'élargir les fonctions de quelques uns des systèmes existants. Ce réseau devrait également comprendre des liens avec des laboratoires extérieurs pour permettre une division plus adéquate du travail informatique entre les différents centres de calcul.

Les conséquences, sous-jacentes aux propositions précédentes, sur le design de ce réseau peuvent se regrouper en trois points :

- long terme,
- flexibilité,
- ~~reliabilité~~ et disponibilité.

Fiabilité

1) Long terme

Le réseau devrait être planifié dans l'hypothèse qu'il sera opératoire pendant une longue période, dix ans ou davantage. Ceci est nécessaire étant donné que quelques applications devront durer longtemps sans être interrompues et que les besoins essentiels à satisfaire ne changeront pas rapidement. L'équipement qui lui est attaché aura vraisemblablement une durée de vie plus courte, aussi bien dans le centre de calcul que tout autour, dans les laboratoires. Par conséquent, ce réseau doit permettre une adjonction facile de nouveaux équipements; il ne doit donc pas dépendre des propriétés particulières des ordinateurs installés actuellement au CERN.

2) Flexibilité

Sur une longue période de fonctionnement, la taille du réseau, l'importance de son utilisation globale ainsi que l'équilibre entre les diverses fonctions qu'il remplit changeront. Par ailleurs, il faut que de nouveaux liens puissent être ajoutés si nécessaire sans déranger le système existant. La possibilité de tels changements doit être comprise dans le design dès le départ. Ces considérations vont dans le sens d'une certaine modularité de l'architecture de ce réseau.

3) Reliabilité et disponibilité

Le réseau doit donner confiance aux utilisateurs dans le fait qu'ils pourront l'utiliser quand ils le désireront sans que leurs données soient modifiées ou détruites, même en cas de panne. Comme on admet que le réseau se développera avec le temps, le développement et le test de nouvelles fonctions doivent être faits en parallèle avec son utilisation normale, sans allouer x heures par semaine pour son développement.

Conclusions générales sous-entendues par ces principes

Ce réseau devrait idéalement être construit en parties indépendantes et autonomes, plutôt que de dépendre d'une seule partie compliquée, centralisée.

C. Implémentation prévue du réseau

Le réseau consistera notamment en :

- 1) Un ensemble de mini-ordinateurs (des MODCOMP II), appelé, noeuds du réseau, situés à différents points du site et reliés entre eux par des connections très rapides (3Mbits/sec.).
- 2) Une série de "protocoles", ou ensemble de règles définissant la manière dont se font les échanges de données entre un ordinateur utilisateur (appelé souscripteur) et un noeud.

Le réseau sera relié aux ordinateurs hôtes (les trois CDC du centre de calcul dans une première phase) par un programme, le NIP (Network Interface Program) qui sera en exécution en permanence dans chacune des CDC 6000. Le programme devra permettre à un souscripteur d'envoyer et de recevoir le résultat d'un job susceptible d'effectuer toutes sortes de manipulations de fichiers; il devra lui permettre également de mener à bien des opérations telles que le "job enquiry" et la manipulation de fichiers permanents à travers INTERCOM, le système temps partagé en usage sur les CDC de la série 6000.

Le NIP sera écrit essentiellement en SYMPL ("the CDC SYstem Programming Language"), un langage évolué, afin qu'il soit plus court, plus facile à bien structurer et, par conséquent, plus facile à tester et modifier.

Les plans actuels prévoient un noeud (au concentrateur) par hôte (donc par CDC 6000 dans notre cas). Le canal les reliant aura une vitesse de 3 Mbits/sec. Tout sera fait pour approcher le plus possible cette vitesse par la mise en place d'un software approprié, bien qu'il ne soit pas évident que :

- a) il y ait une exigence de cet ordre dans la pratique;
- b) le concentrateur puisse aller aussi vite;
- c) le NIP puisse atteindre cette vitesse.

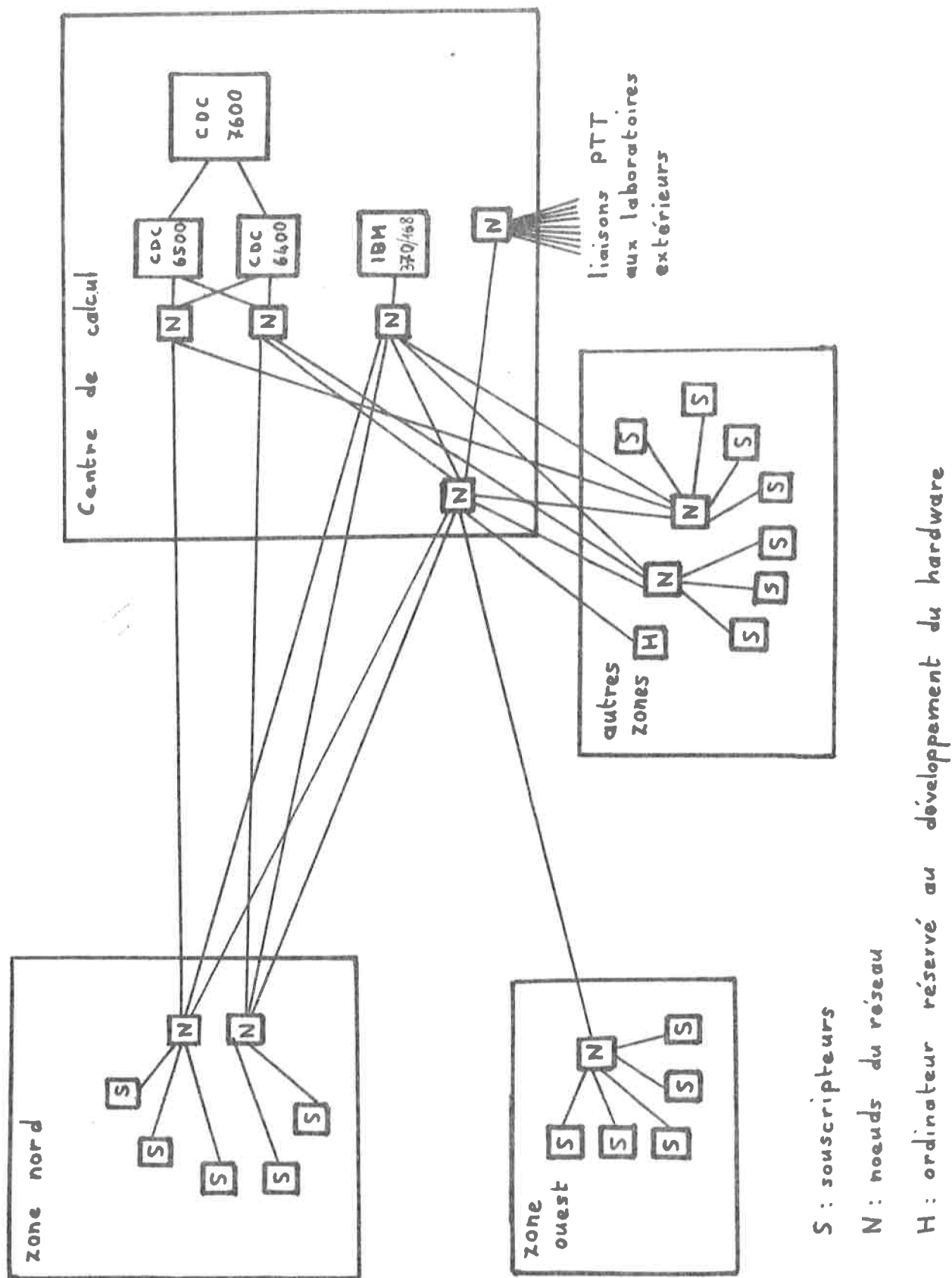


Schéma du réseau

Ce schéma est symbolique; en pratique, le nombre d'ordinateurs ainsi que leurs interconnexions seront différents.

Du point de vue du software, deux points critiques essentiels peuvent limiter la vitesse du NIP :

- le nombre moyen d'instructions nécessaires au traitement d'un message qui doit être de moins de 1300 instructions SYMPL (pour une explication de ce chiffre, voir référence 3, point 3.2.1);
- la vitesse d'accès au disque ou, plus précisément la capacité de transfert de la mémoire centrale des 6000 aux disques.

Pour tenter de réduire au minimum le nombre d'instructions on pourrait écrire certaines parties du NIP en COMPASS, le langage assembleur des 6000, surtout les parties qui font intervenir un code particulièrement peu efficace parce que généré par SYMPL (cf. référence 3, point 3.4), mais il ne sera vraisemblablement guère possible d'intervenir sur ce point critique. En revanche, il est possible d'avoir une bonne idée du temps d'accès au disque en se livrant à toute une série de mesures en fonction de différents paramètres. Ces mesures devraient permettre de dégager la méthode optimum d'accès au disque pour le NIP. Tel est l'objet de ce travail.

II. PRESENTATION DU PROBLEME POSE

Il convient préalablement de donner quelques explications sur le système d'exploitation des CDC 6000 ainsi que des précisions sur le fonctionnement de NIP, tel qu'il est prévu pour le moment. Ceci nous permettra de présenter plus précisément le problème qui nous était posé.

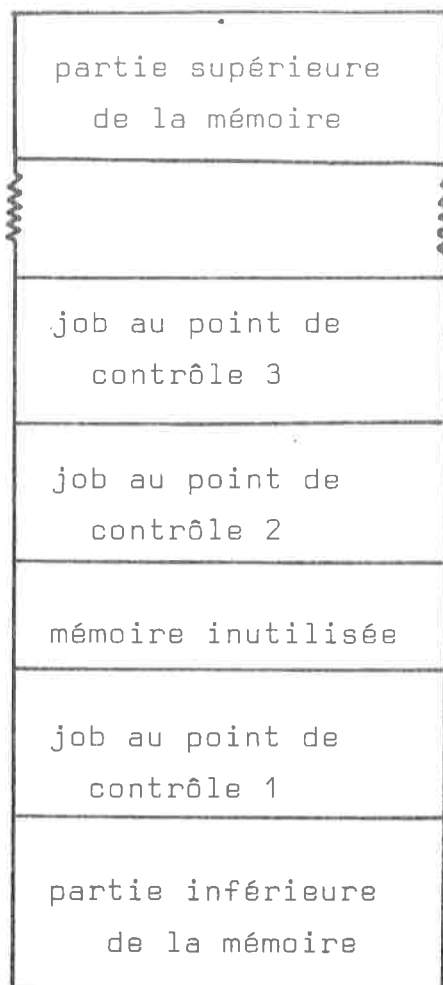
A. Quelques éléments sur le système d'exploitation des CDC 6000 en usage au CERN : SCOPE 3.4.

Le système d'exploitation des 6000, SCOPE 3.4, permet la multiprogrammation : 15 jobs peuvent être simultanément en exécution dans la mémoire centrale. Chaque job y occupe un ensemble contigu de mots formant un bloc. La première adresse d'un bloc est dénommée RA ("reference address"), la longueur de ce bloc FL ("field length").

Chaque fois qu'un job est provisoirement abandonné, le contenu actuel des registres, RA, FL ainsi que l'adresse de la prochaine instruction à exécuter - en tout 16 mots formant un bloc appelé paquet d'échange ("exchange jump package") - est stocké dans une zone de la mémoire centrale. Cette zone, extérieure au job, se trouve dans la partie inférieure de la mémoire et s'appelle aire du point de contrôle ("control point area"). Dans cette aire, on trouve également d'autres informations sur le job telles que son nom et son temps d'exécution par exemple.

Chaque job en mémoire centrale est relié à un point de contrôle ("control point") et chaque point de contrôle relie les éléments suivants à un job particulier :

- la longueur du champ en mémoire centrale (FL),
- les autres équipements et fichiers utilisés,
- l'aire du point de contrôle



La partie supérieure de la mémoire contient essentiellement l'information relative aux fichiers stockés sur disque. Quant à la partie inférieure de la mémoire, elle comprend notamment le résident de SCOPE, des tables du système, des pointeurs, des zones de communications entre la mémoire centrale et les processeurs périphériques.

Les processeurs périphériques (PP) sont de petits ordinateurs exécutant essentiellement les entrées/sorties. Ils sont associés à toutes les phases d'exécution d'un job, telles que : transfert sur disque d'un job lu par un lecteur de cartes, transfert d'un fichier entre un disque et la mémoire centrale, envoi de fichiers à l'imprimante, lecture et écriture sur bandes magnétiques, etc. Les PP sont les compléments des 6000

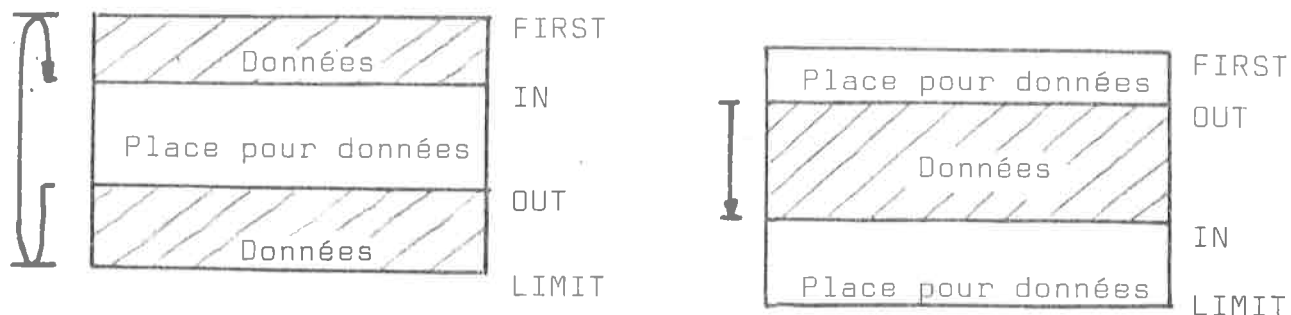
qui ne comprennent pas d'instruction d'entrée/sortie. En effet, les mémoires de masse ECS ("extended core storage"), pour lesquelles les 6000 ont une instruction d'entrée et une instruction de sortie, sont des dispositifs semblables à la mémoire centrale. C'est par leur intermédiaire d'ailleurs qu'on réalise le plus efficacement l'instruction "Move from CM to CM" qui n'existe pas sur les 6000. Ils sont utilisés également comme zone tampon ("buffer") entre la mémoire centrale et des dispositifs périphériques plus lents ainsi que comme mémoire comprenant une partie des programmes du système.

Un des processeurs périphériques, PPO, contient en permanence un programme appelé "Monitor" (MTR) dont une partie est également stockée dans la mémoire centrale. MTR contrôle toutes les fonctions de SCOPE, notamment la coordination des autres PP. C'est PPO qui, périodiquement, contrôle le contenu de RA+1. En effet, c'est cette position mémoire qui contient différentes requêtes au système, notamment les requêtes d'entrée/sortie.

Ces requêtes sont gérées tout d'abord par un programme en mémoire centrale CP.CIO qui, ensuite, les transmet au processeur approprié. C'est 1SP ("stack processor") qui prend en charge les entrées/sorties relatives aux disques. L'ensemble de ces requêtes est placé dans une table appelée pile des requêtes ("request stack"). Le rôle essentiel de 1SP est d'effectuer en priorité la requête impliquant un temps de "overhead" minimum. Ce temps comprend le temps de déplacement des têtes ainsi que le temps de commutation des têtes sur le disque.

La plus petite unité de mots pouvant être transmise à un équipement d'entrée/sortie s'appelle un PRU ("Physical Record Unit"). La taille d'un PRU varie selon l'équipement utilisé : pour les disques, un PRU comprend 64 mots de 60 bits. Les données sont transmises par l'intermédiaire d'une zone tampon circulaire ("circular buffer") qui peut être d'une taille quelconque, mais au moins supérieure à 1 PRU. Cette zone est

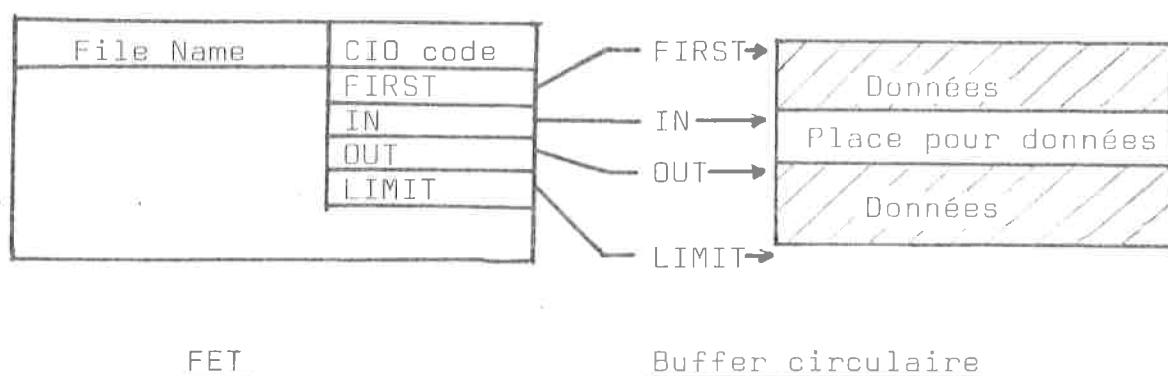
~~est~~ dite circulaire parce que les programmes gérant les entrées/sorties traitent le dernier et le premier mots de cette zone comme contigus. Par ailleurs, à l'intérieur de cette zone, chaque PRU doit être précédé d'un mot de contrôle.



Zone tampon (buffer) circulaire

- FIRST : adresse du premier mot du buffer } ne changent pas
 LIMIT-1: adresse du dernier mot du buffer } en cours d'exécution
- IN : adresse courante d'entrée : modifiée par SCOPE au cours d'une lecture, par le programme utilisateur au cours d'une écriture.
- OUT : adresse courante de sortie : modifiée par SCOPE au cours d'une écriture, par la programme utilisateur au cours d'une lecture.

Une opération d'entrée/sortie consiste à placer dans RA+1 le code de l'opération désirée ainsi qu'un pointeur à une zone comprenant entre autres les quatre adresses mentionnées ci-dessus. Cette zone, appelée FET ("File Environnement Table"), est une zone de communication que l'utilisateur doit créer à l'intérieur de son propre champ (FL). Chaque fichier manipulé d'une quelconque manière doit avoir une telle zone qui lui corresponde et qui ait été correctement préparée préalablement à toute requête d'entrée/sortie



Dans notre travail, nous avons considéré quatre fonctions d'entrée/sortie : READ/WRITE et READC/WRITEC. La différence essentielle entre ces 2 types d'entrée/sortie réside dans le fait que pour READC et WRITEC, c'est l'utilisateur qui prépare le mot de contrôle précédant chaque PRU transmis. Mais SCOPE ne vérifie ni ce mot de contrôle, ni d'autres éventuelles erreurs de programmation. En principe plus rapides que READ/WRITE, ces fonctions ne sont donc recommandées que pour la programmation de système. (??)

Il existe d'autres fonctions d'entrée/sortie, telles READIN et WRITOUT. Ces fonctions, en plus du buffer circulaire, utilisent une autre zone de mémoire dans laquelle l'utilisateur met ou reçoit les données. C'est le système qui s'occupe de faire le lien entre cette zone et le buffer circulaire, évitant ainsi à l'utilisateur le travail de modification des pointeurs IN ou OUT dans la FET.

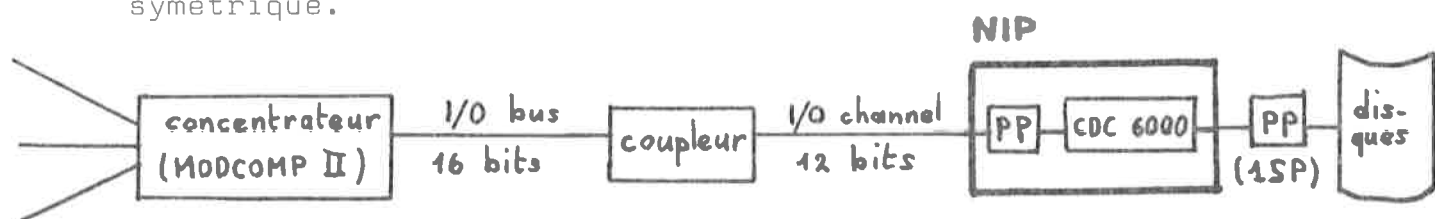
Pour nos mesures nous avons envisagé 2 manières de modifier ces pointeurs. Une manière consiste à faire une requête d'entrée/sortie, puis à utiliser la circularité du buffer en modifiant un des pointeurs (IN pour une sortie, OUT pour une entrée) au fur et à mesure que l'autre (OUT pour une sortie, IN pour une entrée) est modifiée par le système. On obtient ainsi un transfert continu de données par Changement continu des pointeurs,

d'où la façon dont nous nous réfèrerons plus loin à cette méthode : CH. L'autre manière consiste à attendre que la requête d'entrée/sortie soit terminée avant de réinitialiser le pointeur que le système a modifié (OUT pour une sortie, IN pour une entrée). Par ailleurs, pour éviter de gaspiller du temps de processeur central à attendre que la requête soit terminée, on demande à SCOPE de mettre le programme dans un statut de Rappel périodique ("periodic recall"), d'où la façon dont nous nous réfèrerons à cette méthode : R. Lorsque le programme est dans ce statut de rappel périodique, SCOPE peut activer un autre point de contrôle.

La période de rappel, c'est-à-dire le temps minimum avant que le programme soit réactivé, peut être spécifiée. Cependant, le programme est généralement réactivé après que 1 ou 2 PRUs aient été transmis, quelle que soit la période de rappel spécifiée.

B. Quelques éléments sur le fonctionnement de NIP

Rappelons tout d'abord que le NIP est le programme, en exécution dans les CDC 6000, qui réalisera le lien entre les ordinateurs nodaux (concentrateurs) et les ordinateurs centraux. Dans ce qui suit, nous ne considérerons que le trajet concentrateur → CDC 6000, le trajet inverse étant rigoureusement symétrique.

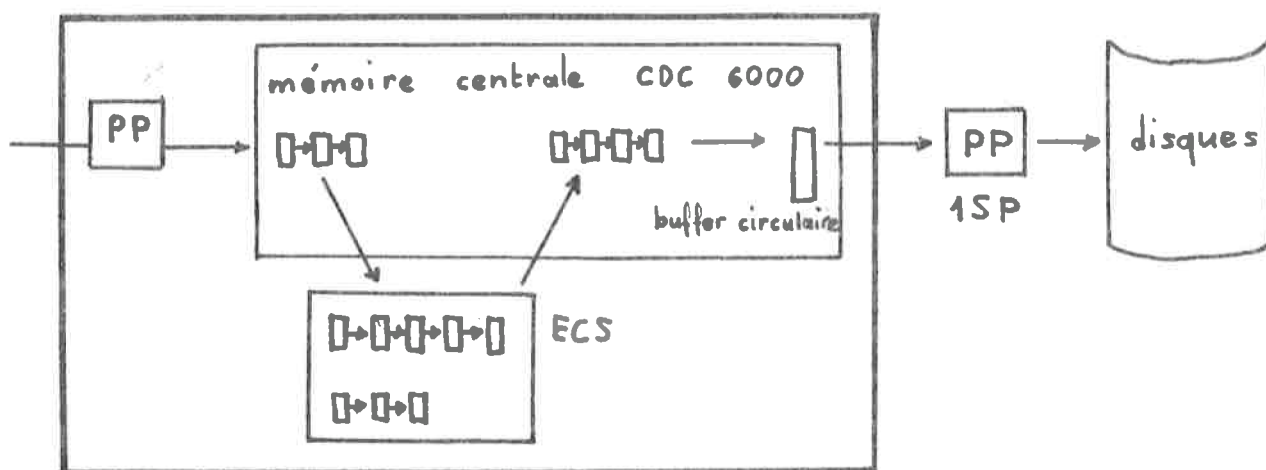


Les données échangées entre un souscripteur et l'ordinateur hôte le sont par blocs de mots appelés messages. La longueur d'un message de même que son organisation interne sont fixées par un ensemble de règles, un protocole. Un message se compose des données à transmettre ainsi que d'un en-tête. C'est dans cet en-tête que sont définis notamment le processus auquel

se rattache le message dans le souscripteur et l'ordinateur hôte auquel il est destiné, permettant ainsi d'établir ce qu'on appelle un lien logique.

C'est un PP, faisant partie intégrante du NIP, qui transmet les messages du concentrateur auquel il est rattaché à la mémoire centrale d'une CDC 6000 où ils sont organisés sous forme de queues. C'est sous cette forme également qu'ils sont provisoirement stockés dans la mémoire ECS, mais regroupés maintenant par liens logiques. Puis, de l'ECS, ils sont retransmis dans la mémoire centrale où ils sont regroupés dans un buffer circulaire avant d'être stockés sur disque comme fichiers permanents.

NIP



C. Présentation du problème posé

Optimiser la capacité de transfert au disque revient dans notre cas à répondre aux questions suivantes :

- 1) Est-il intéressant d'utiliser plus d'un buffer circulaire ?
- 2) Quelle doit être la taille de ce (de ces) buffer(s) ?
- 3) Faut-il transmettre les données de manière continue en modifiant le pointeur, IN ou OUT, au fur et à mesure (CH) ou faut-il attendre la fin d'une requête avant de transmettre les données suivantes (R) ?
- 4) Est-il préférable d'utiliser READC/WRITEC plutôt que READ/WRITE ?

Pour y répondre, il faut commencer par mesurer la vitesse de transfert au disque en fonction des différents paramètres sous-jacents à ces questions. C'est ce que nous avons fait et allons exposer dans ce qui suit. Enfin, déterminer la méthode optimum de transfert au disque consistera à confronter les résultats de ces mesures aux exigences du NIP : place limitée en mémoire centrale, nombre d'instructions limité pour traiter un message. Ceci fera l'objet de notre conclusion.

III. METHODOLOGIE

Afin de déterminer la méthode optimum de transfert au disque, il est nécessaire de comparer entre elles différentes mesures. Ces comparaisons n'ont un sens que pour autant que les conditions dans lesquelles les mesures sont effectuées soient identiques, ou du moins analogues. Plus précisément, il est souhaitable que :

1. Le nombre et le type de disques ainsi que le nombre de canaux utilisés soient identiques.

Pour établir ce point, il suffit de contrôler préalablement à chaque série de mesures le contenu d'une table du résident donnant l'état des différents équipements disponibles : EST ("Equipment Status Table").

2. Les disques utilisés soient les mêmes d'une mesure à l'autre.
Là encore, le contrôle était possible en faisant appel à une table du système nommée RBR ("Record Block Reservation table"). Cette table permet de retrouver notamment sur quel disque ont été stockés le premier et le dernier mots d'un fichier. Nous admettons que si ces deux mots sont stockés sur le même disque, l'ensemble du fichier se trouve sur ce disque.

3. La priorité accordée au job soit la même.

Pour ce faire, il nous a été possible de demander à l'opérateur d'introduire manuellement pour chaque job la priorité voulue. Cette requête s'est faite par une carte de contrôle PAUSE qui stoppait le job aussi longtemps que l'opérateur n'intervenait pas.

4. Le taux d'utilisation parallèle de l'ordinateur soit semblable.

C'est cette dernière condition qu'il était le plus difficile de contrôler. Cela découle de ce que les mesures effectuées durant des périodes à taux élevé d'utilisation parallèle de l'ordinateur (ou périodes "chargées") donnent des résultats très variables dont il semble difficile d'extraire une moyenne significative.

Pour obtenir une moyenne significative, il faudrait être capable d'établir une formule exprimant le degré d'utilisation de l'ordinateur à partir d'un certain nombre de statistiques telles que le nombre de requêtes dans la pile des requêtes, le nombre de points de contrôle actifs, le nombre de PP en activité, etc. Par ailleurs, en admettant que cela soit possible, il serait vraisemblablement nécessaire de calculer ce taux un grand nombre de fois au cours d'une même mesure afin d'obtenir un taux moyen significatif. Enfin, il n'est pas évident qu'on obtienne plus d'une fois le même taux moyen d'utilisation. Ce qui est nécessaire si l'on désire comparer les mesures.

Quoi qu'il en soit, CIA, le seul programme disponible au CERN effectuant des statistiques de cet ordre n'était pas fiable du tout. Il nous a donc fallu renoncer à contrôler directement le taux d'utilisation de l'ordinateur. La solution indirecte que nous avons utilisée finalement consiste à effectuer non pas une, mais plusieurs mesures successives pour une même méthode de transfert des données entre la mémoire centrale et le disque, chaque mesure étant déjà en elle-même une moyenne. Puis nous analysons l'évolution de ces mesures dans le temps à l'aide de graphiques et d'histogrammes. Si ces mesures restent constantes dans le temps, on est en droit d'affirmer que le taux d'utilisation parallèle de l'ordinateur était très faible (période "calme"). A priori, de telles conditions sont relativement faciles à obtenir pour peu que l'on effectue ces mesures de nuit ou durant les week-ends. Il semble donc possible de déterminer une méthode optimum pour ces conditions puisqu'alors les comparaisons deviennent possibles.

C'est la solution que nous avons retenue, pensant que la méthode optimum pour les périodes calmes serait au moins aussi efficiente que les autres pour les périodes chargées. Nous avons admis cette hypothèse que nous nous sommes bornés à vérifier par la suite autant que possible. L'essentiel des mesures a donc été effectué au cours de périodes "calmes".

A. Méthode de mesure pour les périodes calmes

Rappelons que, de manière générale, il s'agit de mesurer le temps de transfert d'un certain nombre de mots entre la mémoire centrale et un disque en faisant varier certains facteurs. Pour être significative, une telle mesure doit porter sur un grand nombre de mots.

Nous avons donc choisi que chaque mesure serait une moyenne établie sur 1000 requêtes d'entrée/sortie de n PRUs chacune (n PRUs étant la taille du buffer circulaire). Pour obtenir une telle mesure, ou le temps moyen de transfert par PRU (pour un buffer circulaire de n PRUs), il suffit de diviser le temps total de transfert par $n * 1000$. Ce temps varie évidemment d'une fois à l'autre en fonction de la taille du buffer circulaire, du degré d'utilisation parallèle de l'ordinateur, de la méthode de transfert utilisée; mais disons d'emblée, pour donner un ordre de grandeur, qu'il se situe très grossièrement autour de 30 secondes. Nous n'avons donc pas pris en considération le temps mis pour obtenir ce temps (1 appel à une routine du système : RTIME) qui est de l'ordre de 0,4 msec.

Puis nous répétons 100 fois la même mesure afin d'observer son évolution dans le temps, chacune de ces mesures étant séparée de la suivante par un temps de 10 secondes au moins. Donc, nous observons la variation d'une même mesure sur une période de l'ordre de 1 heure environ. Nous analysons ensuite les 100 résultats obtenus globalement, puis par groupe :

a) Globalement

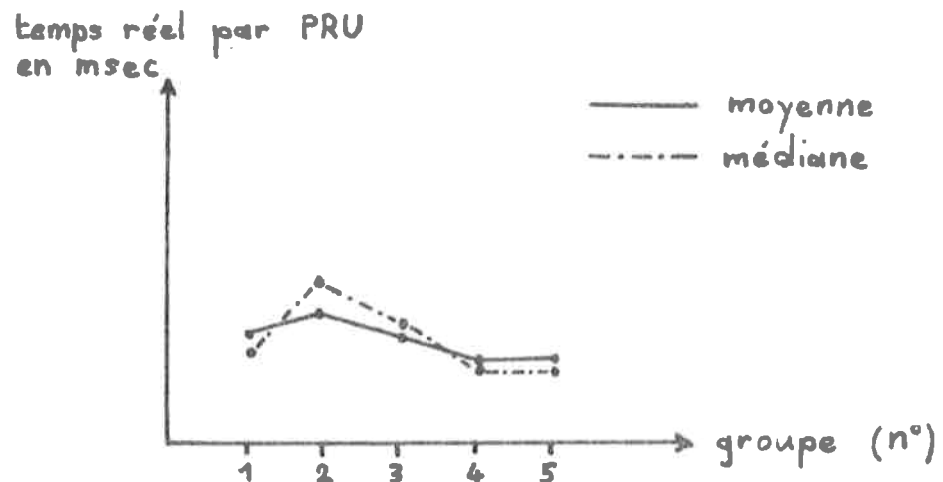
Nous produisons pour chaque ensemble de 100 mesures un histogramme ainsi qu'un certain nombre de calculs statistiques que nous n'avons finalement que peu utilisés, à part la moyenne et la médiane.

b) Par groupe

Nous définissons 5 groupes de mesures de la manière suivante :

1er groupe :	mesures n°	1	à	n°	20
2e groupe :	"	21	à		40
3e groupe :	"	41	à		60
4e groupe :	"	61	à		80
5e groupe :	"	81	à		100

Pour chacun de ces groupes, nous produisons un histogramme ainsi que quelques résultats statistiques qui à nouveau ne se sont révélés que de peu d'utilité, à l'exception de la moyenne et de la médiane. En effet, ce qui nous a finalement permis de sélectionner les résultats significatifs que nous recherchions était un graphique donnant la moyenne et la médiane en fonction du groupe de mesures

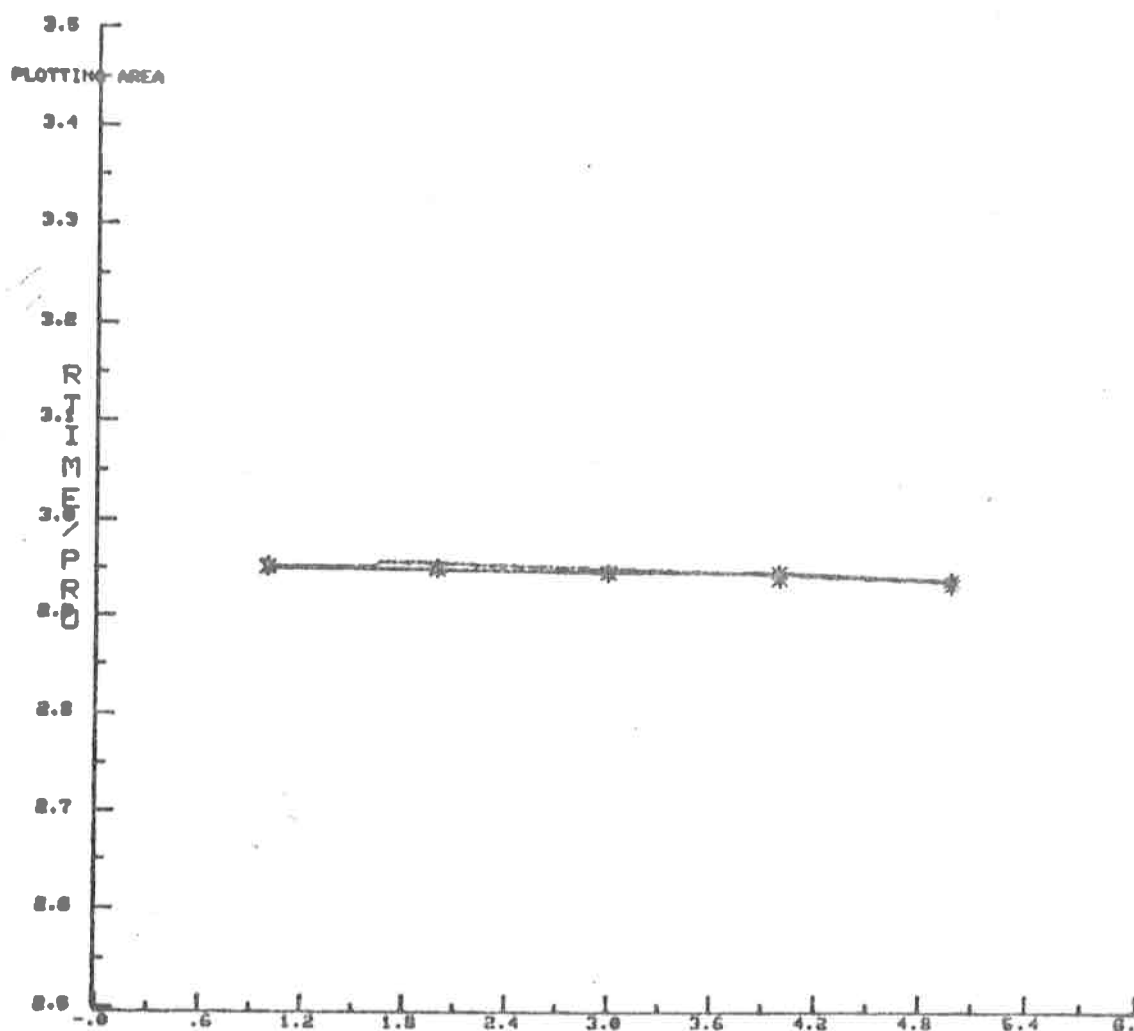


Deux critères essentiels déterminent le fait que le degré d'utilisation parallèle de l'ordinateur soit resté constant. Il faut que :

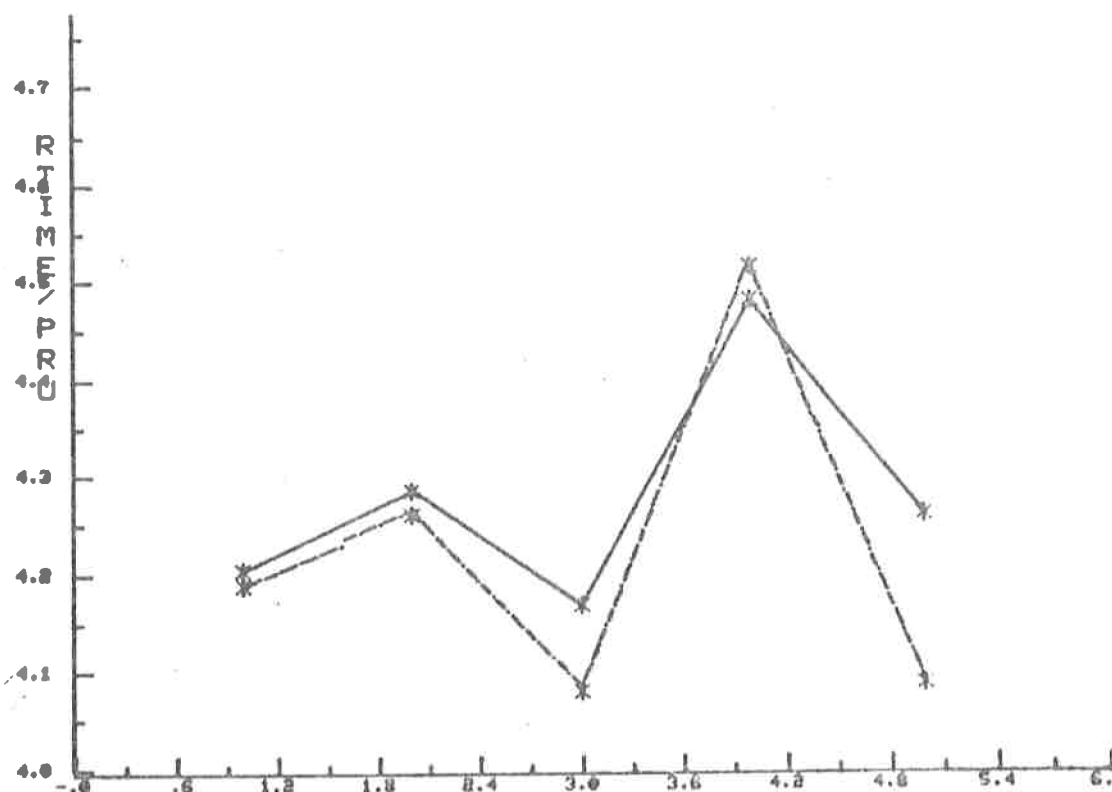
- 1) la moyenne reste la même pour chacun des groupes,
- 2) la moyenne soit proche de la médiane pour chaque groupe.

Si ces conditions ne sont pas remplies, on peut être sûr qu'il y a eu des variations significatives du résultat des mesures à un moment donné, donc du taux d'utilisation parallèle de l'ordinateur.

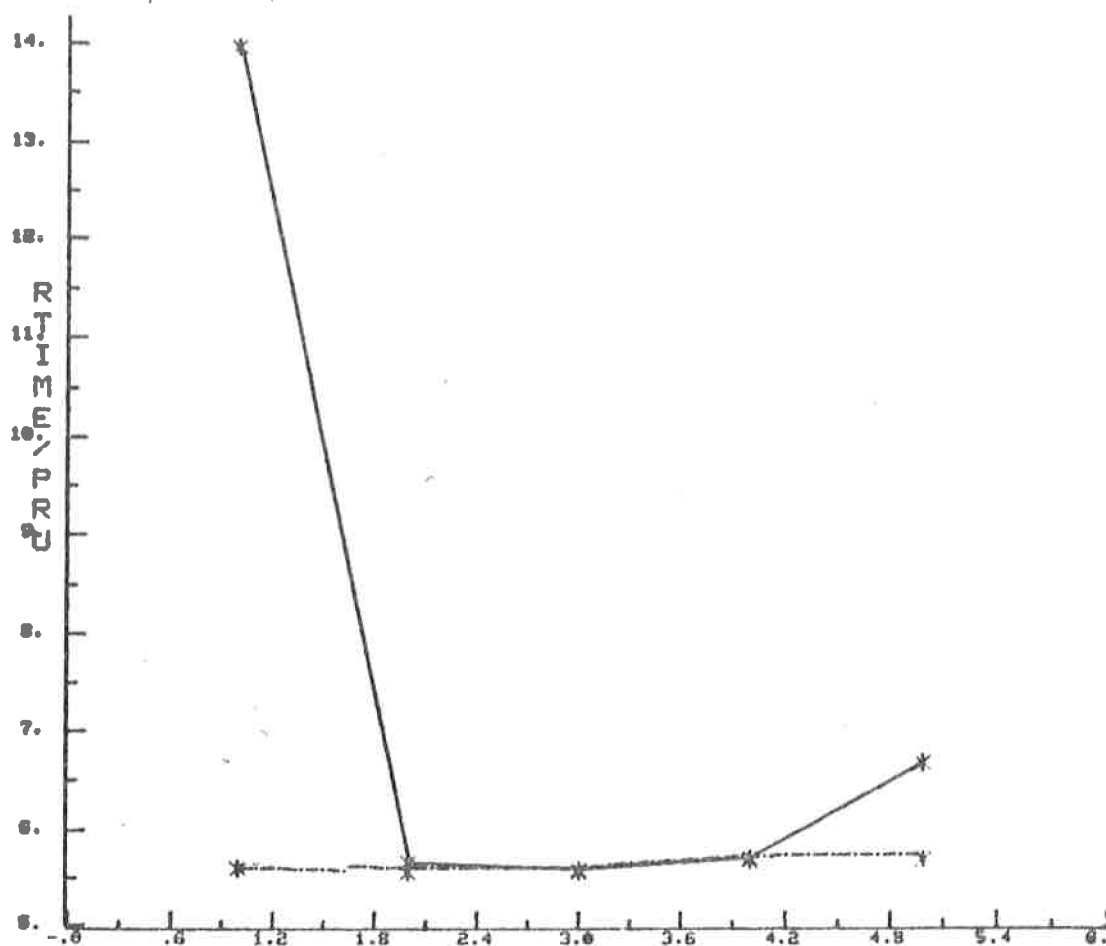
Par ailleurs, nous avons fait l'hypothèse que si ce taux restait constant au cours d'un test (donc pendant 1 heure environ), ceci impliquait qu'il était très faible pendant ce temps et que nous nous trouvions alors dans une période "calme". Voici un graphique illustrant fort bien une telle situation.



En revanche, dans le cas suivant, les variations étaient telles que les mesures ont dû être refaites.



Par contre, dans le cas suivant, malgré les fortes variations pour les groupes n° 1 et n° 5, nous avons considéré les résultats des groupes n° 2, 3 et 4 comme suffisamment significatifs pour les retenir.



B. Méthode de mesure pour les périodes chargées

Comme nous l'avons dit plus haut, en effectuant des mesures au cours de périodes chargées, nous ne visions que 2 buts :

- 1) vérifier l'hypothèse selon laquelle la méthode optimum pour les périodes calmes n'est pas pire que les autres pour les périodes chargées;
- 2) donner malgré tout quelques idées du temps de transfert entre la mémoire centrale et un disque pour les périodes chargées, ce qui, rappelons-le, est un des points critiques du réseau.

Pour réaliser ces mesures, nous avons dû procéder un peu différemment, ceci pour 2 raisons :

- 1) Les variations du degré d'utilisation parallèle de l'ordinateur variant très fortement au cours de telles périodes, il était exclu d'obtenir une moyenne significative et donc inutile de multiplier par trop les mesures.
- 2) Il fallait éviter de trop charger l'ordinateur afin de limiter les désagréments que de telles mesures entraîneraient pour les autres utilisateurs.

Tout ceci nous a finalement conduits à procéder de la manière suivante. Comme précédemment, chaque mesure représente le temps moyen de transfert par PRU (pour un buffer circulaire de n PRUs), cette moyenne étant effectuée sur un transfert de $1000 \cdot n \cdot 64$ mots de 60 bits. En revanche, nous n'avons répété cette mesure que 3 fois par test au lieu de 100. Puis nous passons au test suivant. Cependant, afin de faciliter les comparaisons, nous avons répété le même test un certain nombre de fois (3 ou 7 fois) au cours de la même période.

Pour chacune des méthodes testées, nous obtenions ainsi 9 (3×3) ou 21 (7×3) résultats que nous avons analysés de manière individuelle, par groupe de 3 et globalement.

- a) de manière individuelle en produisant un graphique donnant le temps moyen de transfert par PRU pour chacune des 9 ou 21 mesures;
- b) par groupe de 3
 - en calculant la moyenne et la médiane pour chacun de ces groupes;
 - en produisant un graphique représentant pour chacun de ces 3 ou 7 groupes de mesures l'évolution de leur moyenne et médiane respectives;
- c) globalement en effectuant un certain nombre de statistiques sur ces 9 ou 21 mesures (moyenne, médiane, ...).

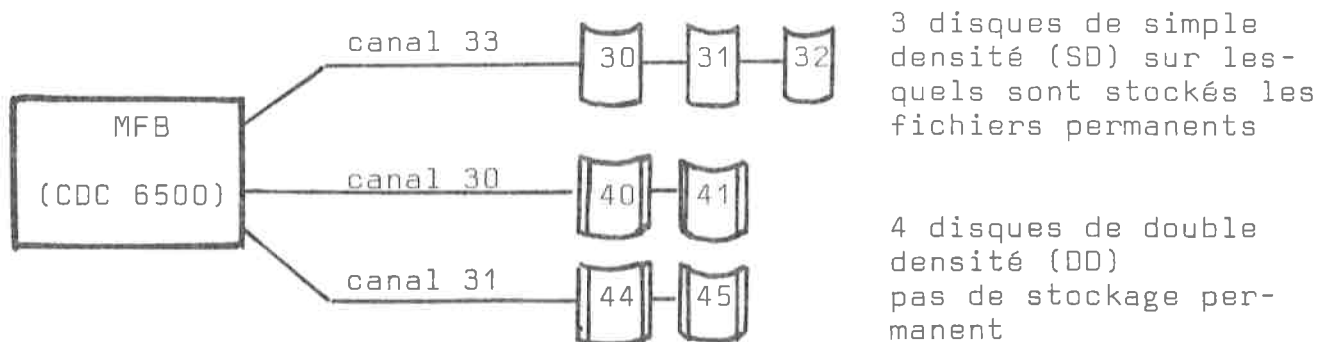
IV. IMPLEMENTATION

Dans ce chapitre nous allons préciser la manière dont nous avons réalisé certains points mentionnés précédemment :

- le contrôle des canaux et des disques en activité pendant nos mesures;
- le contrôle des disques utilisés pour nos mesures;
- les mesures proprement dites;
- les graphiques.

A. Contrôle des canaux et des disques en activité

Lors de nos mesures, la configuration relative aux disques était généralement la suivante :



Au début de chaque série de mesures, nous contrôlions cette configuration en relevant le contenu d'une table du système ("Equipement Status Table") et, plus précisément, le contenu relatif aux disques qui se trouvait dans les positions 4410₈ à 4427₈.

Les numéros des disques ("EST ordinal") figurant dans le schéma ci-dessus ont été obtenus en soustrayant à la position considérée l'adresse du premier mot de cette table (4360₈).

Nous donnons maintenant un relevé illustrant cette situation qui correspond aux périodes calmes :

C - DISPLAY - CENTRAL MEMORY

004410	4500	0001	0000	0131	0002	+	A	AY	B	F\	} AY/EST ord 30→32/canal 33 on
004411	4600	0001	0001	0131	0002	-	A	AA	B	F\	
004412	4400	0001	0002	0131	0002	9	A	BAY	B	F\	
004413	4000	0000	0003	4131	0002	5		CGY	D	F\	} AY/EST ord 33,36,37/canal 33 off
004414	0000	0000	0000	0000	0000					F\	
004415	0000	0000	0000	0000	0000					F\	
004416	4000	0000	0006	4131	0002	5		F6Y	B	F\	} AZ/EST ord 40,41/canal 30 on
004417	4000	0000	0007	4131	0002	5		G6Y	B	F\	
										F\	
004420	4000	0001	0000	0132	0003	5	A	AZ	C	F\	} AZ/EST ord 44,45/canal 31 on
004421	4000	0001	0001	0132	0003	5	A	AAZ	C	F\	
004422	0000	0000	0000	0000	0000					F\	
004423	0000	0000	0000	0000	0000					F\	} AZ/EST ord 44,45/canal 31 on
004424	4000	0001	0004	0132	0004	5	A	DAZ	D	F\	
004425	4000	0001	0005	0132	0004	5	A	EAZ	D	F\	
004426	0000	0000	0000	0000	0000					F\	
004427	0000	0000	0000	0000	0000					F\	

①

②

③

La colonne ① comprend un chiffre significatif du numéro de canal

2 correspond au canal 33

3 correspond au canal 30

4 correspond au canal 31

Une lettre dans la colonne ② indique que le disque est en usage, un chiffre qu'il est hors service.

La lettre Y dans la colonne ③ correspond à un disque de simple densité, Z à un disque de double densité.

Voici encore un relevé illustrant cette fois une autre situation, celle qui correspondait aux périodes "chargées" :

C - DISPLAY - CENTRAL MEMORY

004410	4500	0001	0000	0131	0002	+	A	AY	B	F\	} AY/EST ord 30→32/canal 33 on
004411	4600	0001	0001	0131	0002	-	A	AA	B	F\	
004412	4400	0001	0002	0131	0002	9	A	BAY	B	F\	
004413	4000	0001	0003	0131	0002	5	A	CAY	B	F\	} AY/EST ord 33,36/canal 33 on
004414	0000	0000	0000	0000	0000					F\	
004415	0000	0000	0000	0000	0000					F\	
004416	4000	0001	0006	0131	0002	5	A	FAY	B	F\	} AY/EST ord 37/canal 33 off
004417	4000	0000	0007	4131	0002	5		G6Y	B	F\	
										F\	
004420	4000	0000	0000	4132	0003	5		GZ	C	F\	} AZ/EST ord 40,41/canal 30 off
004421	4000	0000	0001	4132	0003	5		AGZ	C	F\	
004422	0000	0000	0000	0000	0000					F\	
004423	0000	0000	0000	0000	0000					F\	} AZ/EST ord 44,45/canal 31 off
004424	4000	0000	0004	4132	0004	5		DGZ	D	F\	
004425	4000	0000	0005	4132	0004	5		EGZ	D	F\	
004426	0000	0000	0000	0000	0000					F\	
004427	0000	0000	0000	0000	0000					F\	

B. Contrôle des disques utilisés

Comme nous l'avons déjà expliqué (chapitre III sous chiffre 2), nous avons jugé nécessaire de vérifier que le ou les disques utilisés ne variaient pas d'une mesure à l'autre. C'est pourquoi, après chaque mesure, il nous fallait consulter une table du système ("Record Block Reservation table") à partir de laquelle il était possible de déterminer le numéro du disque sur lequel était écrit le premier et le dernier mots du fichier que nous utilisions. Nous avons considéré que si le numéro du disque (EST ordinal) était identique pour ces deux mots, l'ensemble du fichier se trouvait sur ce disque. Pour parvenir à cette table, il était nécessaire de passer par 3 autres tables intermédiaires :

FET (File Environment Table)

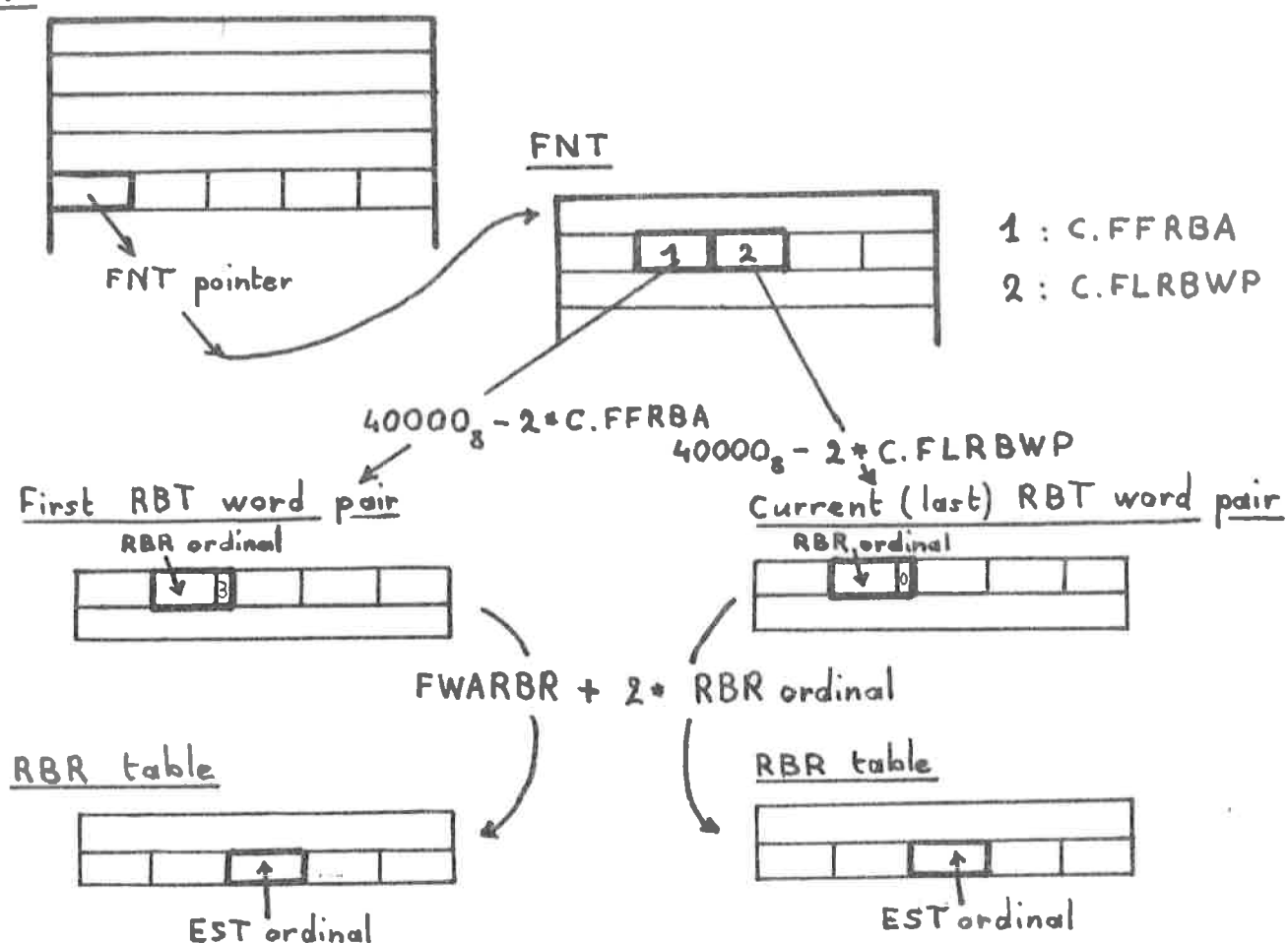
→ FNT (File Name Table)

→ RBT (Record Block Table)

→ RBR Table → EST ordinal

Une routine du système, UCM, nous a permis d'accéder aux tables situées en-dehors de notre propre champ (FL). Voici maintenant un schéma plus détaillé de ce que nous avons fait.

FET



FWARBR : First Word Address Record Block Reservation Table
se trouve dans un mot situé en position n° 2 dans la mémoire
centrale :

adresse

2



C. Réalisation des mesures proprement dites

Le programme réalisant les mesures comprend un programme principal et 4 sous-programmes. Le programme principal, écrit en FORTRAN, effectue essentiellement les tâches suivantes :

- tout d'abord appeler le sous-programme, écrit en COMPASS, recherchant l'adresse du premier mot de la table RBR : FAWRBR;
- appeler un certain nombre de fois (généralement 100) un sous-programme, IODISK, écrit en COMPASS. Ce sous-programme effectue 1 mesure et recherche ensuite le (les) disque(s) utilisé(s) à cette fin;
- appeler 2 sous-programmes, écrits en FORTRAN, effectuant quelques calculs statistiques sur les résultats obtenus;
- enfin, imprimer tous ces résultats.

Ce programme variait essentiellement en fonction de la méthode utilisée pour effectuer le transfert entre la mémoire centrale et le(s) disque(s). Mais avant d'aller plus loin, rappelons les variables en jeu (cf. chapitre II, page 8).

- 1) le nombre de buffers circulaires,
- 2) la taille de ces buffers,
- 3) R ou CH (cf. chapitre 2, pages 5 et 6),
- 4) READ/WRITE ou READC/WRITEC

Ces différentes variables n'apportaient quasiment aucune modification du point de vue de la programmation. En effet, l'augmentation du nombre de buffers (1) n'engendrait qu'une duplication, à peu de choses près, de certaines parties de IODISK. Pour ce qui était de changer la taille de ces buffers (2), il suffisait de faire varier un symbole la définissant.

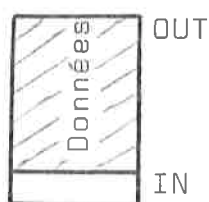
Enfin, la seule différence entre READ/WRITE et READC/WRITEC (4) résidait dans le fait que dans ce dernier cas, il était nécessaire de préparer le mot de contrôle précédant chaque PRU du (des) buffer(s) circulaire(s). Ceci se faisait au début de IODISK, à chaque appel de ce sous-programme. Venons-en maintenant à la seule variable (3) qui ait engendré une modification significative dans la programmation et, plus précisément dans IODISK.

C1. Réalisation de la première situation : R

Nous nous contenterons ici d'esquisser l'organigramme de la situation particulière où l'on écrit, en utilisant WRITE, sur deux fichiers simultanément un total de 1000 requêtes de n PRUs chacune. Dans notre jargon, nous symbolisons cette situation de la manière suivante : WRITE - R - 2 buffers - n PRUs. Mais, préalablement, voici quelques définitions qui faciliteront l'écriture de cet organigramme.

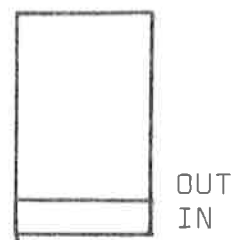
- lfn1 (lfn2) : nom logique du fichier n° 1 (n° 2).
- b1 (b2) : bit situé dans la FET n° 1 (n° 2), mis à 1 par le système dès que la requête relative au fichier n° 1 (n° 2) est effectuée. Au début de chaque nouvelle requête, ce bit est remis à zéro par le système.
- IN1,OUT1 (IN2,OUT2) : pointeurs IN et OUT (cf. chapitre II, page 4) du buffer n° 1 (n° 2). Dans notre cas, il nous suffira de modifier OUT1 (OUT2) après chaque requête pour retrouver l'état dans lequel nous avons initialisé ces pointeurs

buffer circulaire "plein"



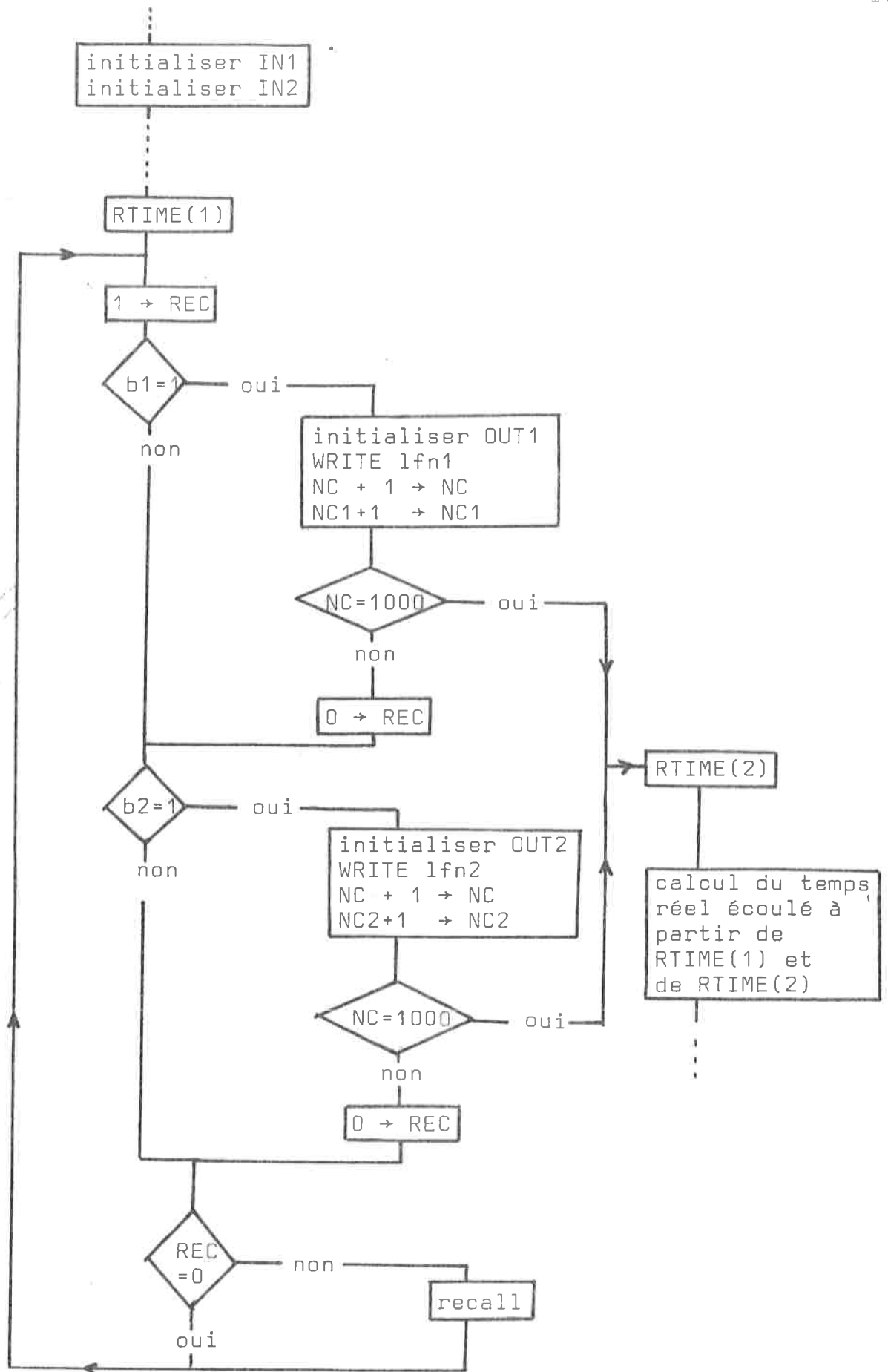
Etat initial des pointeurs

buffer circulaire "vide"



Etat des pointeurs à la fin d'une requête; OUT a été modifié par SCOPE.

- Rec : sémaphore prenant comme valeur 1 si aucune requête en cours n'est terminée, auquel cas, on met le programme sans le statut de rappel périodique. Autrement, Rec = 0 et on continue à effectuer le programme.
- NC : nombre total des requêtes effectuées.
- NC1 (NC2) : nombre de requêtes effectuées sur lfn1 (lfn2).



Nous nous sommes aperçus après coup que cette façon de faire comprenait une erreur. En effet, nous regardons RTIME(2) aussitôt que NC=1000, sans attendre que les dernières requêtes en cours soient terminées. Nous introduisons ainsi une erreur, inférieure à 2 pour mille dans le cas présent, ayant pour conséquence de rendre le résultat légèrement plus favorable. Mais cette erreur était tellement minime en regard de la précision qu'il était possible d'obtenir que nous avons renoncé à la corriger.

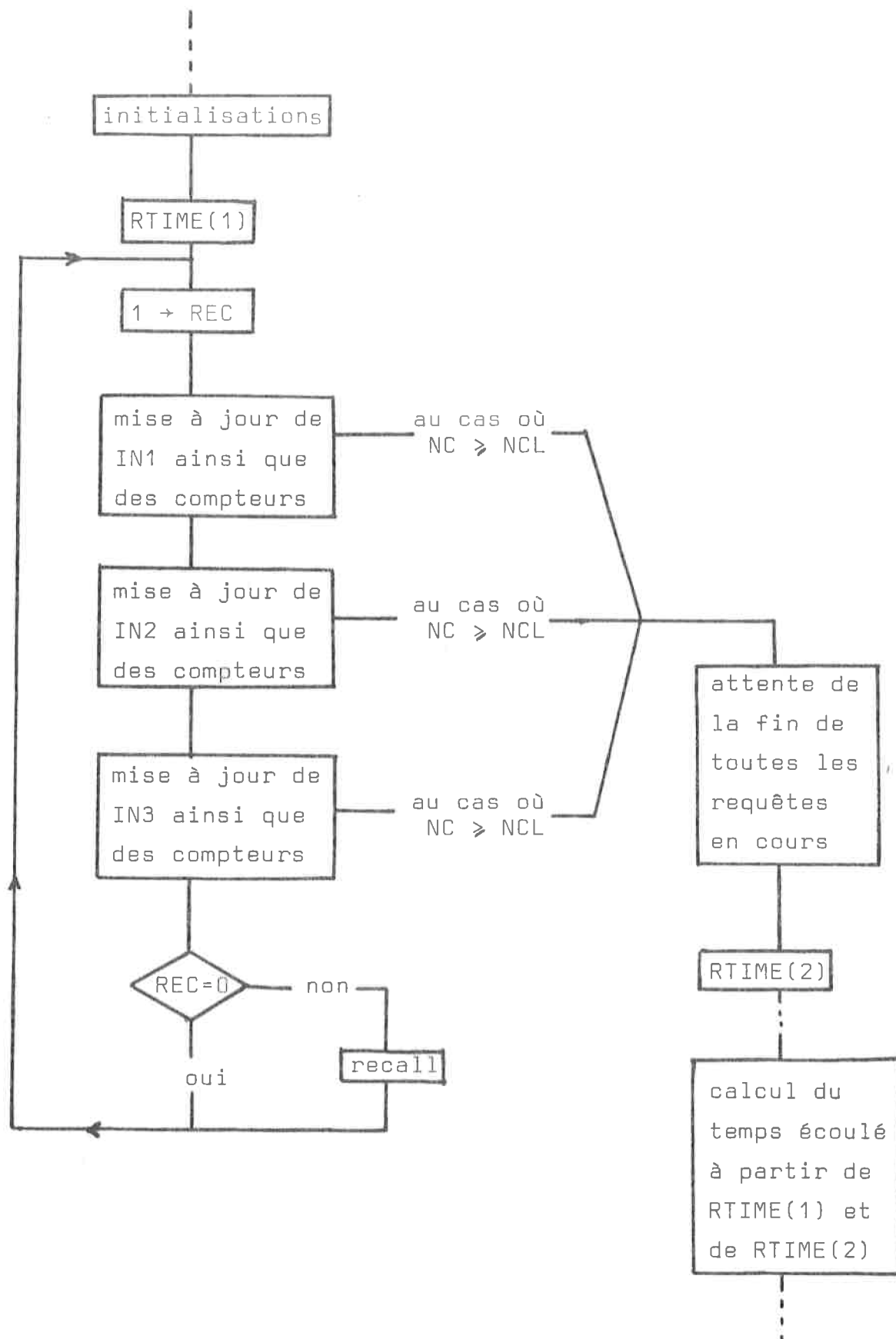
C2. Réalisation de la deuxième situation : CH

Ici encore, vu la symétrie des différents cas particuliers, nous n'envisagerons qu'un seul WRITE - CH - 3 buffers - n PRUs. Dans cette situation, il s'agit essentiellement de modifier les pointeurs IN des 3 buffers au fur et à mesure que le système modifie les pointeurs OUT. Cette modification continue des pointeurs permet d'éviter qu'un des buffers ne se vide complètement, ce qui entraînerait une interruption du processus de transfert. C'est pourquoi nous essayons de maintenir à tout instant les buffers aussi pleins que possible, ce qui revient à maintenir IN aussi proche de OUT que possible. Nous avons établi cette "proximité" à 1 PRU afin que le mot de contrôle précédant chaque PRU des 3 buffers se trouve toujours à la même place. Ceci rend possible l'utilisation de READ/WRITE de cette façon et permet, lorsque nous utilisons READC/WRITEC, de préparer les différents buffers une fois pour toute au début de IODISK.

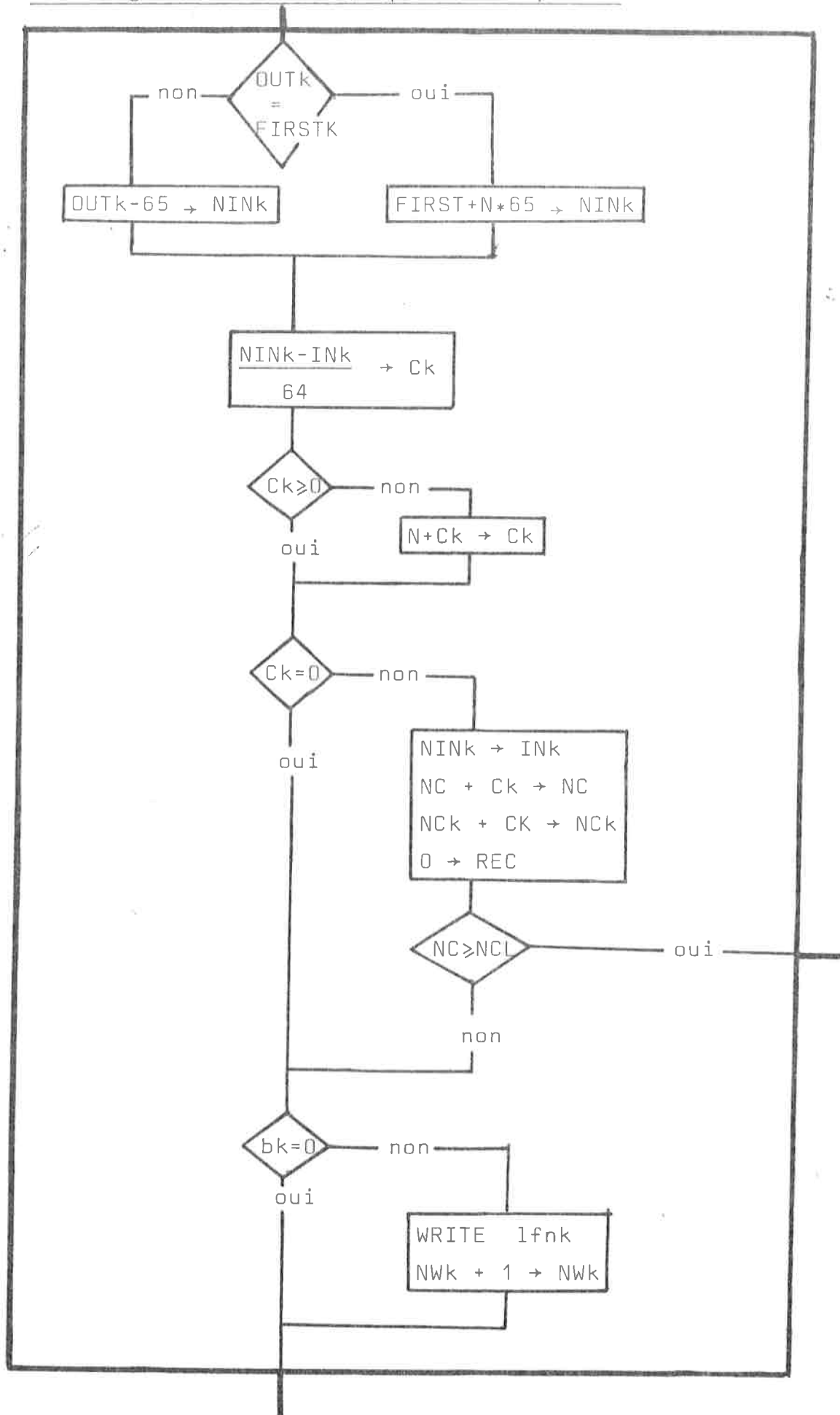
Nous allons maintenant esquisser l'organigramme de IODISK pour cette situation. Préalablement, nous donnerons quelques définitions destinées à en faciliter l'écriture.

- lfnk (k=1,2,3) :
- bk (k=1,2,3) : cf. chapitre IV page 5

- N nombre de PRUs compris dans 1 buffer.
- FIRST_k, IN_k, OUT_k pointeurs FIRST, IN et OUT du k^e buffer.
- NCL nombre total de PRUs que l'on désire écrire
 = 1000*N.
- NC nombre total de PRUs déjà écrits ou en
 train de l'être.
- NCK (K=1,2,3) nombre de PRUs déjà écrits ou en train
 de l'être sur lfnk.
- NINK (k=1,2,3) valeur que doit prendre IN_k pour être le
 plus proche possible de OUT_k.
- C_k nombre de PRUs écrits sur lfnk d'une fois
 à l'autre.
- NW_k nombre de fois que le k^e buffer s'est
 trouvé vide.



Mise à jour de INK ainsi que des compteurs



D. Réalisation des graphiques et histogrammes

Pour réaliser graphiques et histogrammes, nous avons utilisé les programmes de bibliothèque spécialement conçus à cet effet. Ces programmes nous permettaient même de choisir l'équipement sur lequel nous voulions les obtenir. Nous avons opté pour le terminal, étant donné qu'il était possible d'en photocopier l'écran grâce à un dispositif spécial.

V. MESURES

Dans ce chapitre, nous comparons les résultats de mesures effectuées dans des conditions analogues :

- priorité identique (parfois 10, mais généralement 30);
- même configuration relative aux disques (cf. chapitre IV, p. 1 et 2);
- utilisation des mêmes unités de disques.

Aussi avons-nous renoncé à préciser ces conditions pour les mesures dont nous allons maintenant exposer les résultats. De même, nous avons jugé inutile d'accompagner systématiquement ces résultats des graphiques dont ils sont tirés.

Par ailleurs, toutes les mesures que nous avons effectuées n'utilisaient qu'un seul canal à la fois. Nous avons adopté cette façon de faire car nous avons estimé que la méthode optimum recherchée était indépendante du nombre de canaux utilisés.

A. Mesures pendant les périodes calmes

1) Différence entre les entrées et les sorties

Hypothèses :

- a) s'il y a une différence, elle est indépendante :
 - du nombre de buffers ainsi que de leur taille,
 - du mode de transfert : CH ou R.
- b) la différence entre READ et WRITE, si elle existe, devrait être analogue à celle entre READC et WRITEC.

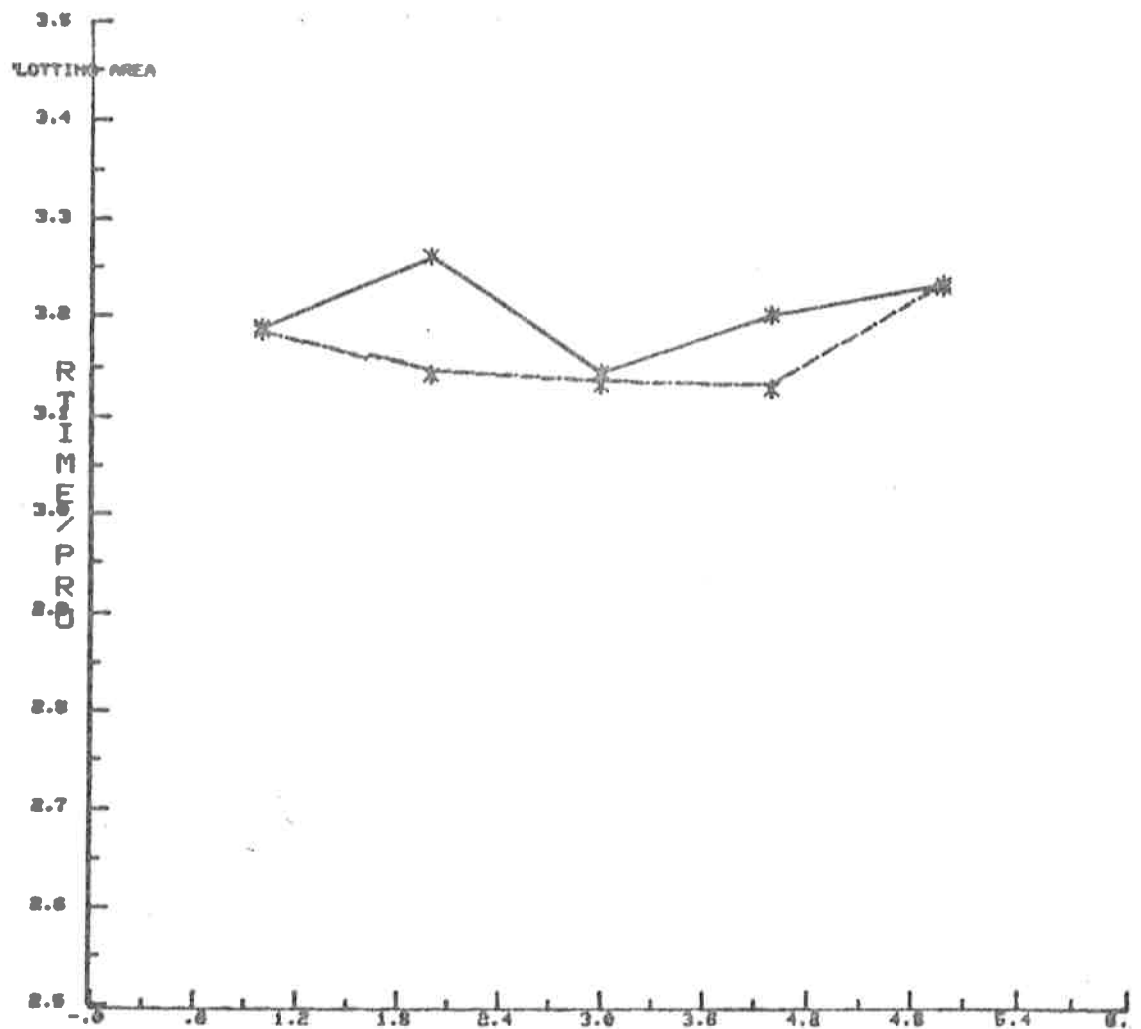
Nous avons donc choisi d'établir cette différence en utilisant READ/WRITE dans les conditions suivantes :

R - 2 buffers - 8 PRUs - 2 disques	
WRITE/WRITE	3.6 msec./PRU
READ/WRITE	3.4 msec./PRU
READ/READ	3.1 msec./PRU

Ces résultats ont été obtenus dans un temps où le taux d'utilisation parallèle de l'ordinateur allait en décroissant. C'est à ce fait que nous attribuons les différences observées et ce d'autant plus, qu'à une autre occasion, nous avons obtenu le résultat suivant :

WRITE/WRITE

R - 2 buffers - 8 PRUs - 1 disque 3.2 msec/PRU



Nous pouvons donc considérer qu'il n'y a pas de différence significative entre les entrées et les sorties. C'est pourquoi dorénavant nous ne prendrons en considération que les sorties.

2) Différence entre WRITE et WRITEC

Hypothèse : s'il y a une différence, elle est indépendante

- du nombre de buffers ainsi que de leur taille,
- du mode de transfert : CH ou R.

Nous avons donc choisi les conditions suivantes pour établir cette différence :

R - 2 buffers - 8 PRUs - 2 disques

WRITE 3.46 msec./PRU

WRITEC 3.44 msec./PRU

Ces résultats montrent clairement qu'il n'y a pas de différence significative entre WRITE et WRITEC.

3) Capacité de transfert au disque en fonction du nombre de buffers

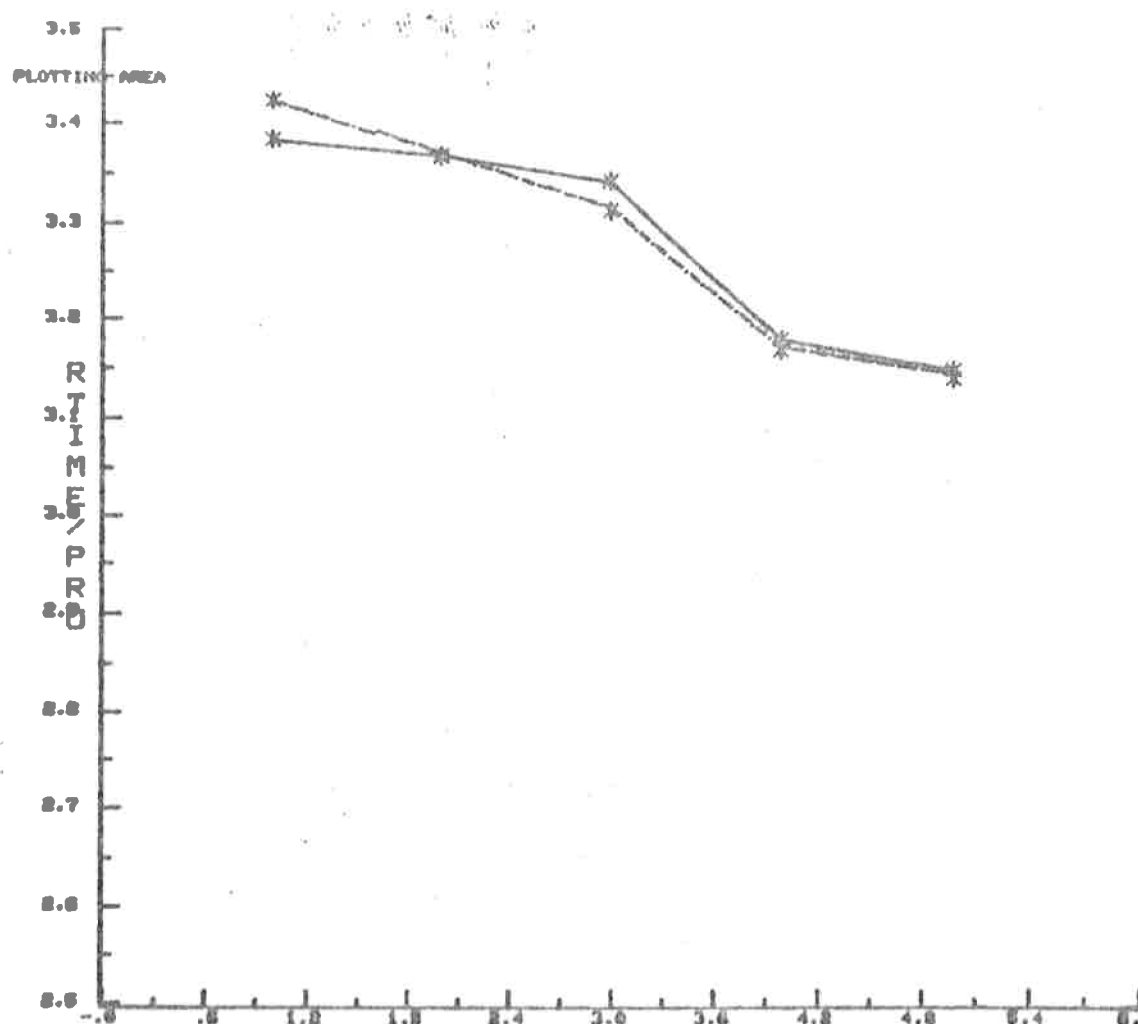
Hypothèse : s'il y a une différence, elle est indépendante de la taille des buffers ainsi que du mode de transfert (R - CH).

Pour ce test, nous avons choisi les conditions suivantes :

WRITEC - R - 12 PRUs

	1 disque	2 disques	
1 buffer	2.9	2.9	
2 buffers	3.1 (1)	2.8	
3 buffers	2.9	2.8	unité : msec./PRU
4 buffers	2.9	2.9	
5 buffers	2.9	2.9	

Le résultat (1) a été obtenu en ne tenant compte que du 5ème groupe de mesures, étant donné les variations du taux d'utilisation parallèle de l'ordinateur manifestes sur le graphique obtenu :



Nous pouvons voir clairement que le fait d'augmenter le nombre de buffers n'apporte aucun gain dans la capacité de transfert au disque.

4) Différence entre les modes de transfert R et CH

Ce test a été effectué pour les conditions suivantes :

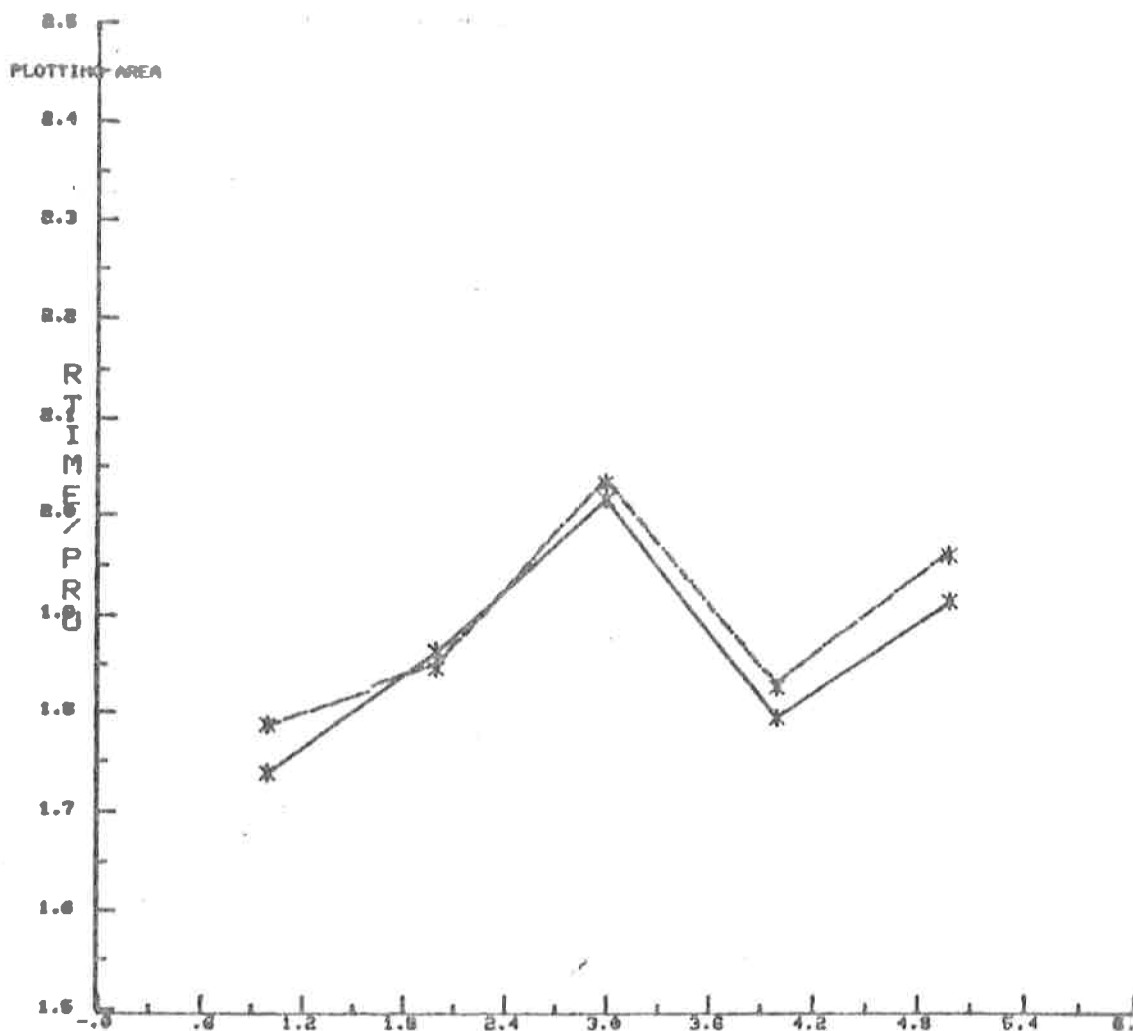
WRITE - 8 PRUs - 2 buffers - 2 disques

R 3.4 msec./PRU

CH 1.8 msec./PRU

Au cours de la seconde mesure, le taux d'utilisation parallèle de l'ordinateur a varié assez fortement (voir graphique ci-après). Nous avons malgré tout retenu les résultats des

groupes n° 1, 2 et 4 comme caractéristiques d'une période calme.



Ces résultats nous montrent que l'utilisation du mode de transfert au disque CH rend ce transfert beaucoup plus rapide que R.

5) Taille optimum du buffer pour WRITEC-CH-1 buffer

Nous appelons "taille optimum du buffer" la taille minimum pour laquelle la capacité de transfert est maximum.

4 PRUs	1.8 msec./PRU
7 PRUs	1.7 msec./PRU
10 PRUs	1.5 msec./PRU
13 PRUs	1.5 msec./PRU
16 PRUs	1,5 msec./PRU

Selon la définition que nous venons d'en donner, 10 PRUs est la taille optimum du buffer.

Jusqu'à présent, toutes les mesures ont été effectuées sur des disques de double densité (DD), qui, au moment où nous faisons nos mesures, n'étaient pas utilisés pour le stockage de fichiers permanents. Seuls les disques de simple densité (SD) étaient alors utilisés à cette fin. Or le NIP utilisera essentiellement des disques réservés au stockage permanent. C'est pourquoi nous avons voulu établir la différence entre ces deux types de disques.

6) Différence entre les disques SD et DD

Nous avons testé cette différence pour WRITEC-CH-1 buffer

	<u>DD</u>	<u>SD</u>
4 PRUs	1.8 msec./PRU	1.8 msec./PRU
7 PRUs	1.7 msec./PRU	1.75 msec./PRU
10 PRUs	1.5 msec./PRU	1.6 msec./PRU
13 PRUs	1.5 msec./PRU	1.6 msec./PRU
16 PRUs	1.5 msec./PRU	1.7 msec./PRU

Le taux d'utilisation parallèle de l'ordinateur était plus important pour le test relatif aux disques de simple densité.

Malgré ce fait on peut dire qu'il n'y a pas de différence significative entre ces deux ensembles de mesures donc entre les deux types de disques considérés.

Les résultats précédents nous amènent à penser que la méthode optimum de transfert au disque est la suivante :

WRITEC - CH - 1 buffer - 10 PRUs

Pour établir ce résultat, nous avons comparé R et CH en utilisant un buffer de dimension réduite (8 PRUs). C'est pourquoi nous avons jugé utile de tester le mode de transfert R avec un buffer de grande dimension : 56 PRUs. Nous avons choisi cette dimension parce qu'elle correspond à celle d'un RB ("Record Block"), l'unité de stockage sur disque.

7) WRITEC - R - 56 PRUs - 1 disque

1 buffer	1.9 msec./PRU
2 buffers	1.85 msec./PRU
3 buffers	1.85 msec./PRU

Ce test donne des résultats assez proches des meilleurs que nous avons obtenus jusqu'à présent. En outre, nous pouvons constater une fois de plus que l'augmentation du nombre de buffers n'apporte aucun gain dans la capacité de transfert au disque.

B. Mesures pendant les périodes chargées

Quatre raisons nous ont poussés à effectuer des mesures durant les périodes où le taux d'utilisation parallèle de l'ordinateur est particulièrement important (chaque jour ouvrable aux environs de 11 h. et de 15 h.) :

- Nous voulions vérifier que l'augmentation du nombre de buffers n'entraîne aucun gain dans la capacité de transfert aux disques.
- Nous voulions établir que la taille optimum du buffer pour WRITEC - CH - 1 buffer que nous avons déterminée pour les périodes calmes (une dizaine de PRUs) n'est pas catastrophique pour les périodes chargées.
- Nous voulions comparer les 2 modes de transfert (R - 56 PRUs et CH - 12 PRUs) ayant donné les meilleurs résultats pendant les périodes calmes.
- Nous voulions avoir une idée de la capacité de transfert qu'il était possible d'obtenir alors.

1) Capacité de transfert en fonction du nombre de buffers

Ce test a été effectué pour WRITEC - R - 12 PRUs - 1 disque. Cependant, pour faciliter les comparaisons, nous avons testé dans le même temps une méthode de référence :

WRITEC - CH - 1 buffer - 12 PRUs.

Chacune des moyennes suivantes a été établie à partir de 21 (7 x 3) mesures (unité : msec./PRU) :

	moyenne	médiane	minimum	maximum
1 buffer	14.88	14.52	5.27	40.64
2 buffers	9.57	8.28	4.28	16.73
3 buffers	13.51	10.75	4.59	50.41
référence	3.36	3.	1.74	6.63

Une fois encore, nous pouvons constater qu'il est inutile d'augmenter le nombre de buffers.

2) Taille optimum du buffer pour WRITEC-CH-1 buffer

Chacune des moyennes suivantes a été établie à partir de 9 (3 x 3) mesures (unité : msec./PRU) :

	moyenne	médiane	minimum	maximum
7 PRUs	3.23	3.34	1.7	4.37
10 PRUs	3.01	2.88	1.98	4.72
13 PRUs	3.02	3.45	1.77	3.88
16 PRUs	3.15	3.47	1.86	4.33

Nous pouvons constater que, pendant les périodes chargées, l'augmentation de la taille du buffer n'apporte aucun gain dans la capacité de transfert aux disques.

3) Comparaison entre R-56 PRUs et CH-12 PRUs

Nous avons comparé les trois situations suivantes :

- a) WRITEC - R - 56 PRUs - 1 buffer
- b) WRITEC - R - 56 PRUs - 2 buffers
- c) WRITEC - CH - 12 PRUs - 1 buffer

A cette fin, nous avons effectué un mardi entre 14h.17 et 16h.30 9 tests dans l'ordre suivant :

- | | | |
|------------------|------------------|------------------|
| 1) 1er test de a | 2) 1er test de b | 3) 1er test de c |
| 4) 2e test de a | 5) 2e test de b | 6) 2e test de c |
| 7) 3e test de a | 8) 3e test de b | 9) 3e test de c |

Chaque test comprenant 3 mesures, chacune des moyennes ci-dessous a donc été établie à partir de 9 mesures (unité : msec./PRU)

		moyenne	médiane	minimum	maximum
R-56 PRUs	a	3.88	3.17	2.32	9.4
	b	4.67	3.16	2.43	9.92
CH-12 PRUs	c	3.51	3.33	1.76	5.12

On peut voir d'emblée que, conformément à notre attente, les situations a et b peuvent être considérées comme équivalentes. D'autre part, il semble que la situation c soit légèrement plus efficiente : elle semble notamment plus stable, surtout si l'on considère la maximum relevé.

En observant les graphiques des 9 tests donnés en annexe (cf. annexe I), on constate que le taux d'utilisation parallèle de l'ordinateur était particulièrement important dès la 3ème mesure du 5ème test, et ce jusqu'à la fin de la 1ère mesure du 9ème test. Ce sont donc les tests n° 6, 7 et 8 qui correspondent le plus à notre intention : comparer R - 56 PRUs (tests 7 et 8) et CH - 12 PRUs (test 6) durant une période chargée; c'est à partir de ces trois tests que nous obtenons les valeurs suivantes :

	moyenne	maximum
R - 56 PRUs	6 msec./PRU	9.9 msec./PRU
CH - 12 PRUs	4.2 msec./PRU	5.1 msec./PRU

Nous allons maintenant récapituler les résultats relatifs à ces deux modes de transfert - qui, dans l'ensemble, se sont montrés les plus efficaces - afin d'en faciliter la comparaison.

		<u>périodes calmes</u>	<u>périodes chargées</u>	
		moyenne	moyenne	maximum
R-56 PRUs	msec./PRU	1.85	6	9.9
	Mbits/sec	<u>2.08</u>	<u>0.64</u>	<u>0.39</u>
CH-12 PRUs	msec./PRU	1.5	4.2	5.1
	Mbits/sec	<u>2.56</u>	<u>0.91</u>	<u>0.75</u>

Les résultats obtenus en utilisant CH - 12 PRUs sont plus performants. Pour les périodes chargées, par exemple, le plus mauvais de ces résultats est encore meilleur que la moyenne des résultats relatifs à R - 56 PRUs. Cette différence nous semble significative même si les résultats ne sont qu'approximatifs. Nous pouvons donc répertorier comme suit les différences essentielles entre ces deux modes de transfert :

	<u>R - 56 PRUs</u>	<u>CH - 12 PRUs</u>
nombre de lignes de code	moins grand	plus grand
place nécessaire en mémoire centrale (vu la dimension du buffer)	plus importante	moins importante
capacité de transfert	moindre	meilleure

VI. CONCLUSION

Les résultats précédents nous poussent à recommander l'utilisation du mode de transfert CH - 12 PRUs dans le cadre du NIP. Nous pensons que son seul inconvénient - un plus grand nombre de lignes de code - ne contrebalance pas ses avantages et, plus particulièrement sa meilleure capacité de transfert.

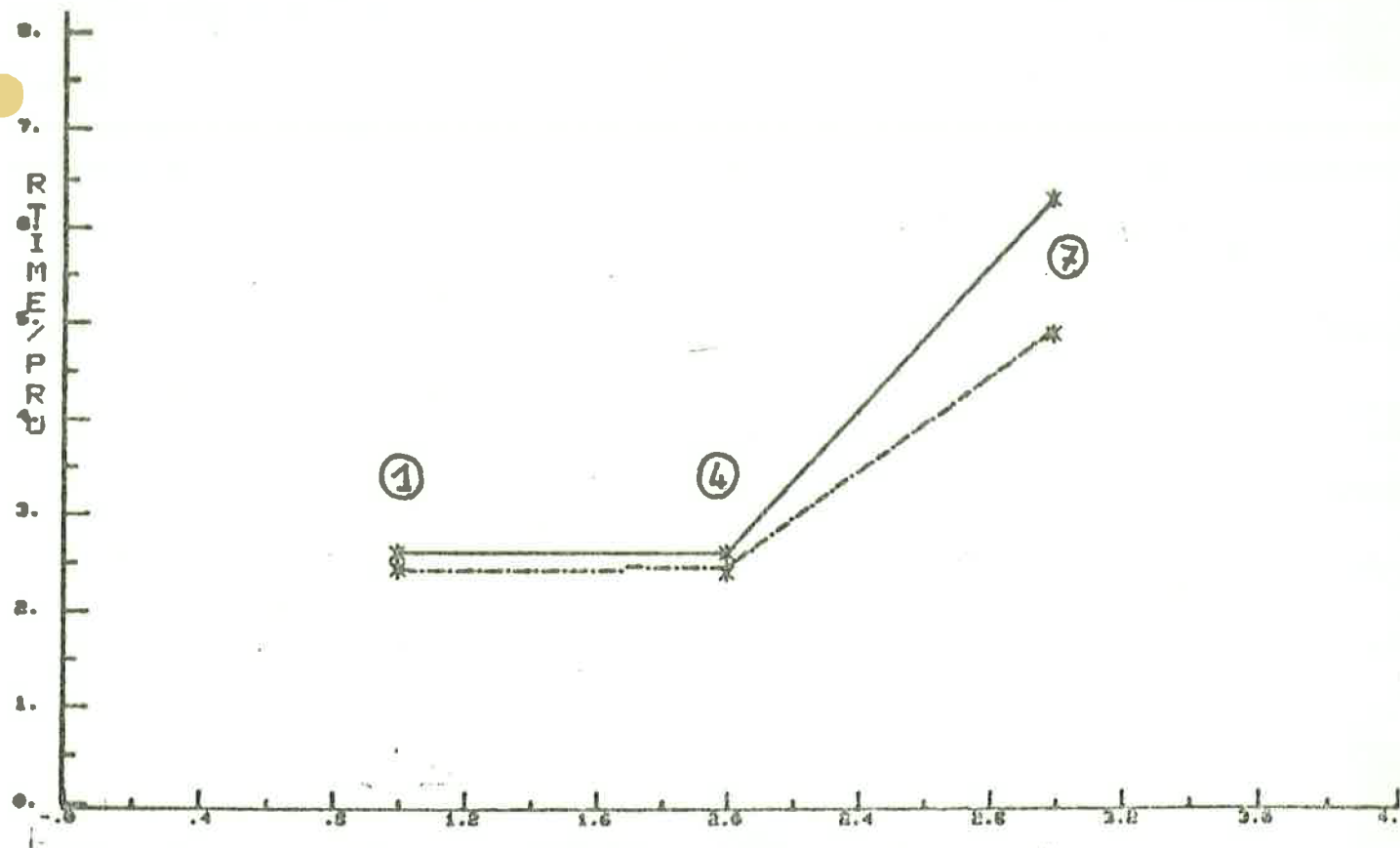
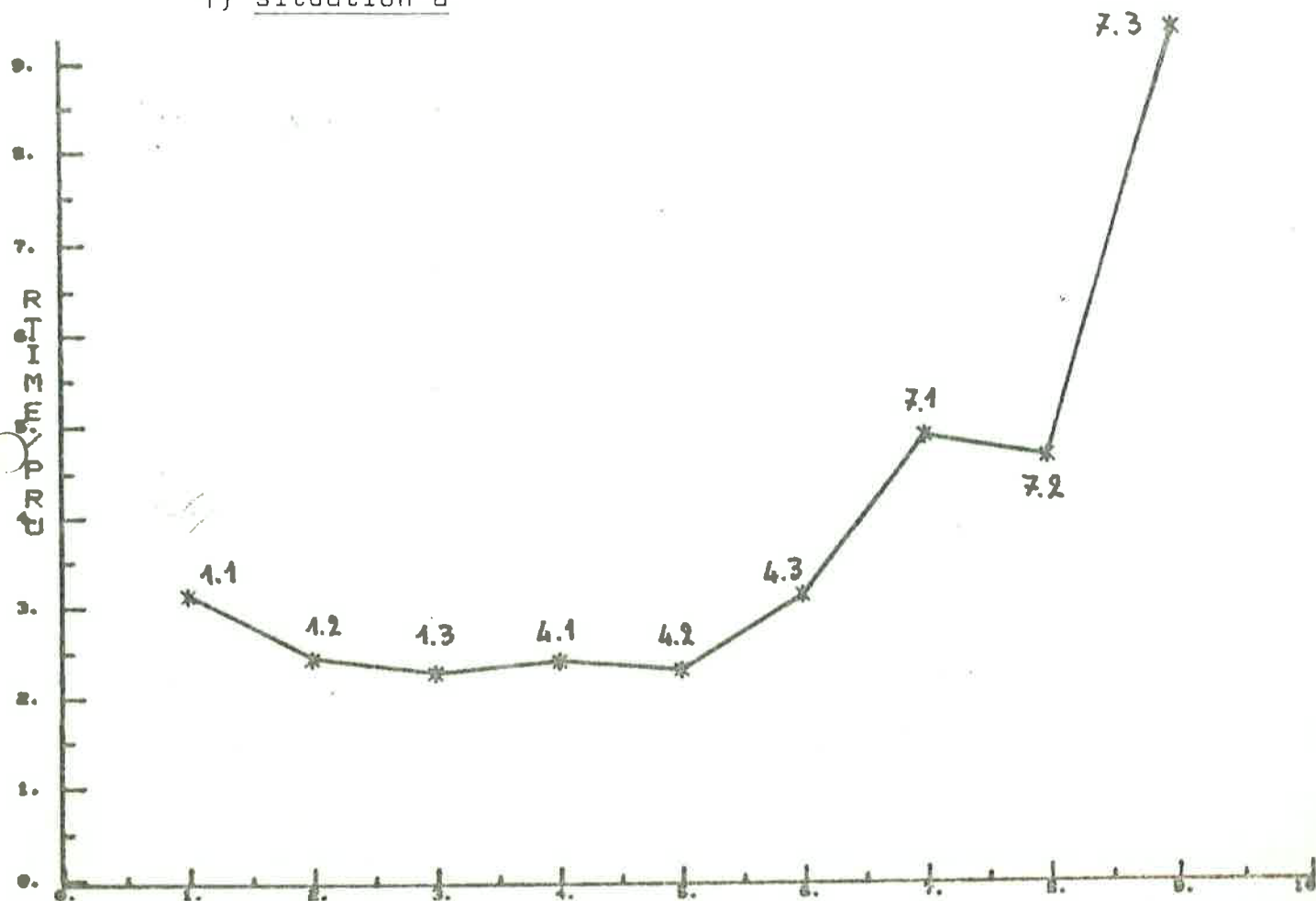
Même si cette situation était retenue, la vitesse souhaitée (3 Mbits/sec) ne sera pas atteinte. Cependant, elle pourrait être vraisemblablement améliorée, pendant les périodes chargées, par l'utilisation d'un canal supplémentaire. Si, en plus, ce canal supplémentaire était réservé à l'usage exclusif du NIP, la diminution consécutive du temps de "overhead" permettrait peut-être de se rapprocher de la capacité de transfert mesurée pendant les périodes calmes : 2.5 Mbits/sec.

Quoi qu'il en soit, ce travail nous a permis d'entrer en contact avec le centre de calcul électronique du CERN, de prendre conscience de la multiplicité des tâches qu'un centre d'une telle importance implique. En outre, il nous a permis de nous familiariser avec le système d'exploitation des CDC 6000 : SCOPE 3.4. Enfin, il nous a donné l'occasion de développer une technique de mesure du temps réel pour un ordinateur utilisé en multiprogrammation.

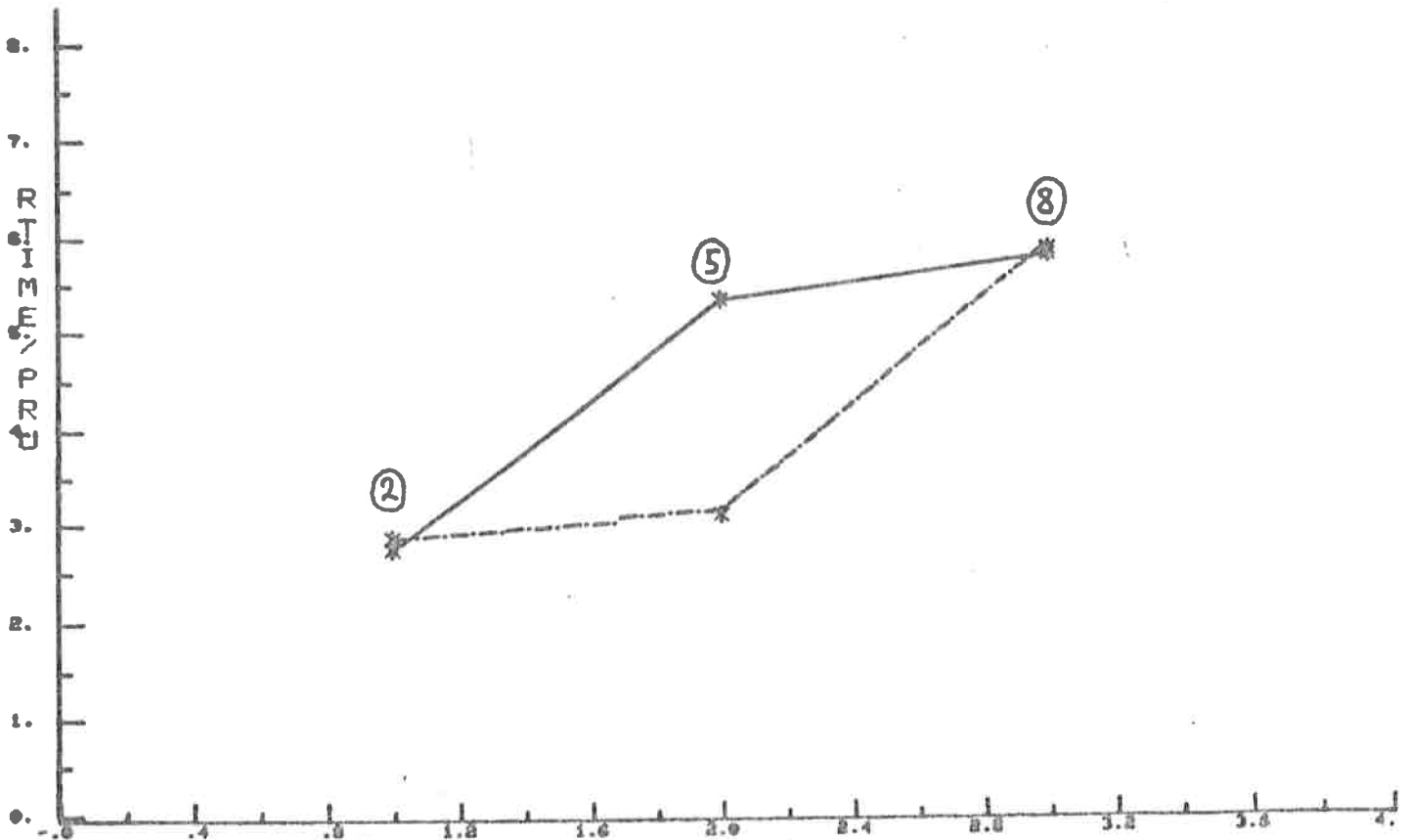
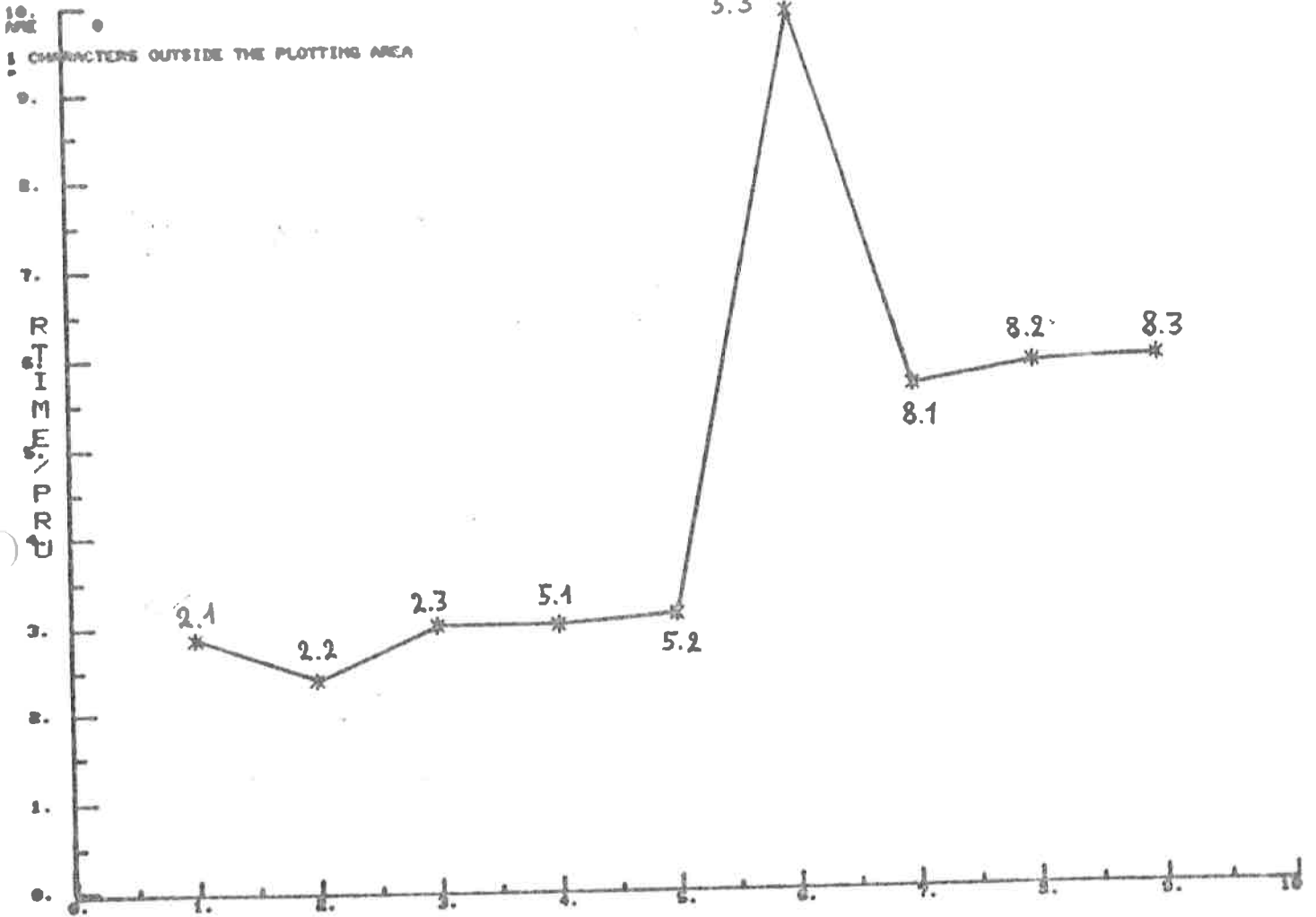
VII ANNEXE I

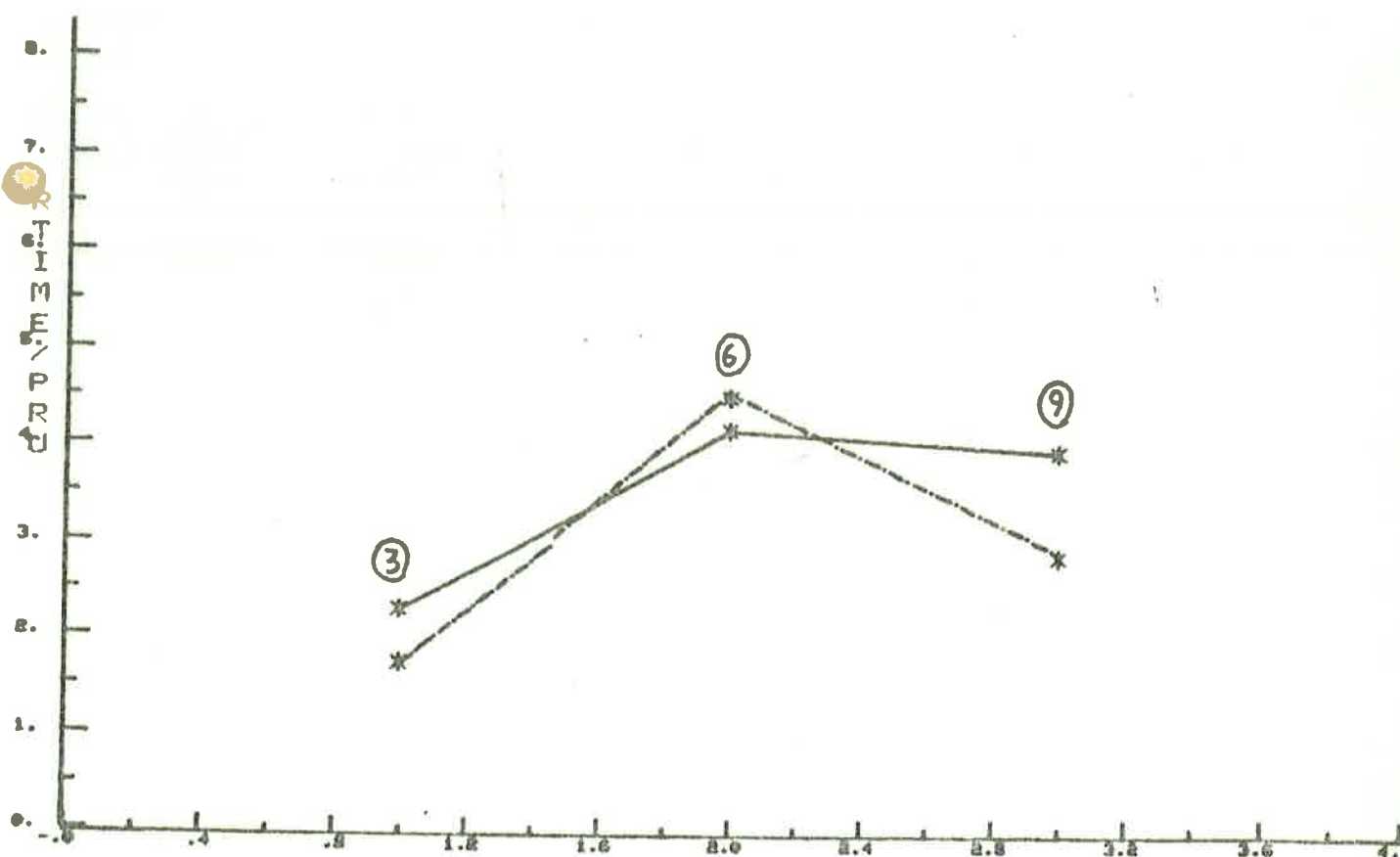
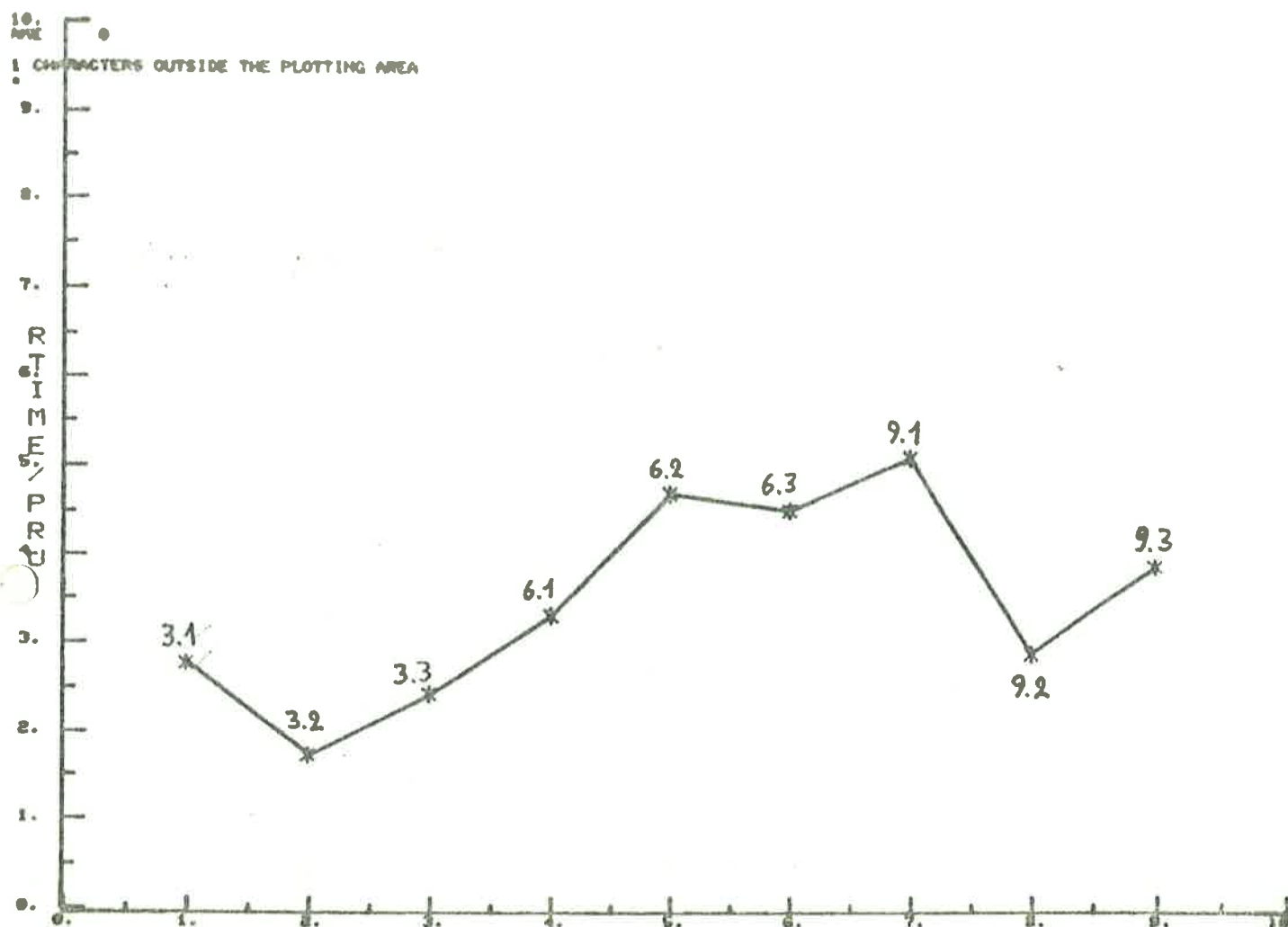
Graphiques de 9 tests destinés à comparer R-56 PRUs à CH-12 PRUs au cours d'une période chargée (cf Ch V, p. 9)

1) situation a



2) situation b



3) situation c

VIII. REFERENCES

- 1) CERN Data Communication Network and Decentralised
Computing Facilities : Pojett Description
M.G.N. Hine 25.2.75
- 2) The 6000 Network Interface Program - General Description
J.D. Blake 2.5.75
- 3) The 6000 Network Interface Program - General Design Outline 1
F. Königshofer 17.6.75
- 4) An Introduction to the CERN Network
H. Davies 3.9.75
- 5) The Focus Remote Access System
E.M. Palandri 1970
in Proc. Int. Conf. on Data Handling in High-Energy Physics
CERN 70-21, p.551 sqq
- 6) Supermux : A Multi-Host Front End Concentrator System For
Asynchronous Consoles
Collectif 8.9.74 CERN 74-19
- 7) Control Data
SCOPE Reference Manual
6000 Version 3.4
- 8) Control Data
6000 Computer Systems Programmers Reference Manual
6000 Version 3.4