Extrait ANNALES DES TÉLÉCOMMUNICATIONS, tome 27, N[os] 1-2
janvier-février 1972

ALGORITHME D'ALLOCATION D'UNITÉ CENTRALE POUR CALCULATEURS EXPLOITÉS EN TEMPS PARTAGÉ

par

Rémi DESPRES
Ancien élève de l'Ecole Polytechnique
Ph. D. de l'Université de Berkeley *

et

Alain BACHE
Ingénieur E.N.S.E.E.I.H.T *

RÉSUMÉ. — *Une méthode nouvelle est développée pour l'allocation d'une ressource partageable au cours du temps entre des usagers identifiables. Bien que d'une portée assez générale, cette méthode est étudiée ici dans le cadre particulier de l'allocation d'unité centrale pour un calculateur exploité en temps partagé. En supposant prévisibles les durées de traitement une première version de l'algorithme est exposée et certaines propriétés remarquables sont mises en évidence. Puis, moyennant quelques extensions, l'algorithme est appliqué au cas où les temps de traitement s'éloignent notablement de leurs valeurs prévues. Enfin, si aucune prévision n'est disponible, des schémas types de comportement sont proposés pour y suppléer. L'application de la théorie à un cas concret est évoquée à propos du calculateur Ramsès II sur lequel l'algorithme d'allocation est en service depuis plusieurs mois.*

PLAN. — *Introduction.* ● I : ***L'algorithme*** I.1. *Position du problème : les processus et leurs états actif et bloqué ;* I.2. *Modèle théorique du traitement parallèle ; notion de priorité ;* I.3. *Hypothèse des durées de services prévisibles ;* I.4. *Notion de temps virtuel et d'échéance ;* I.5. *Principe de l'allocation à temps de traitement prévus (TTP) ;* I.6. *Propriétés de l'allocation TTP ;* I.7. *Conséquences des blocages anticipés ;* I.8. *Prise en compte des temps de changement de processus élu.* ● II : ***Le problème du choix des quantums ;*** II.1. *Responsabilité des prévisions ;* II.2. *Motivation des usagers ;* II.3. *Politique de choix ;* II.4. *Dispositif expérimenté sur le calculateur Ramsès II ;* ● *Conclusion.* ● *Bibliographie* (6 réf.).

INTRODUCTION

La mise en œuvre du calculateur Ramsès II, réalisé et exploité par le CNET, s'est accompagnée de recherches théoriques sur les systèmes d'exploitation en temps partagé. Un des problèmes traité de façon particulièrement approfondie a été celui de l'allocation d'unité centrale. Un algorithme inédit a été mis au point à cette occasion et fait l'objet du présent article. Il repose sur un principe, dit principe de l'allocation à temps de traitement prévus, ou en abrégé allocation TTP, dont on démontre qu'il conduit à une allocation optimale suivant un critère précis.

L'exposé empruntera sa terminologie au domaine de la programmation de systèmes mais il n'échappera pas au lecteur que la portée de l'algorithme proposé est plus générale que son application aux systèmes en temps partagé. Il est en fait applicable à tout système avec file d'attente vérifiant les conditions suivantes :

a) les usagers sont en nombre fini et identifiables ;

b) le service d'un usager peut à tout instant être interrompu au profit d'un autre usager.

L'allocation à temps de traitement prévus est d'autant plus précieuse que certaines ou la totalité des conditions suivantes sont par ailleurs vérifiées :

1. il faut un temps non négligeable pour remplacer un usager par un autre au poste de service ;

2. les usagers sont inégalement prioritaires ;

3. certains des usagers connaissent à l'avance la durée des services qu'ils demandent ;

4. les durées de service sont largement inégales aussi bien d'un usager à l'autre que d'une demande à l'autre pour un même usager.

L'ensemble de ces conditions est vérifié dans le cas de l'allocation d'unité centrale sur Ramsès II. L'expérience recueillie après neuf mois d'exploitation confirme les prédictions de la théorie.

L'algorithme TTP s'insère dans le courant actuel des recherches sur l'allocation d'unité centrale.

La première génération d'algorithmes reposait sur des recettes plus ou moins complexes nécessitant le réglage de nombreux paramètres pour conduire à un service satisfaisant [1, 2] ; le réglage devait se faire empiriquement soit en fonctionnement réel soit à partir de simulations. Deux tendances se dessinent depuis lors : l'une vise à clarifier les concepts et à fonder les algorithmes sur des bases théoriques plus rigoureuses [3, 4] ; l'autre vise à fournir aux usagers une allocation d'unité centrale personnalisée, chacun décrivant le service qu'il demande au moyen de paramètres [5, 6]. L'objectif implicite est de définir

* Au CNET-Issy, groupement INFORMATIQUE ET TRANSMISSION DE DONNÉES, département CALCULATEURS ÉLECTRONIQUES ET SYSTÈMES.

un algorithme simple adapté aussi bien à l'exécution en temps partagé qu'à l'exécution séquentielle.

I. L'ALGORITHME

I.1. Position du problème : les processus et leurs états actif et bloqué.

L'unité centrale est allouée alternativement à des *processus* (par processus nous désignons des travaux en cours d'exécution). Chacun de ceux-ci peut prendre deux états en fonction d'événements extérieurs comme les fins d'entrées-sorties, l'arrivée d'une certaine heure, etc. : l'état *actif* caractérise un processus demandeur d'unité centrale, l'état *bloqué* est celui d'un processus temporairement inapte à utiliser l'unité centrale.

Le problème de l'allocation d'unité centrale consiste à choisir à chaque instant l'un des processus actifs, qui devient ainsi le processus *élu*, et lui affecter l'unité centrale. Un processus actif qui n'est pas élu est dit *en attente*.

Nous utiliserons dans la suite les notations suivantes :

N : nombre des processus, c'est un nombre fixe,

$P_1, P_2, \ldots, P_N$: les N processus,

i : entier de 1 à N servant d'indice de processus,

$A_i(t)$: variable d'état affectée au processus P_i ; elle vaut OUI si P_i est actif, NON si P_i est bloqué,

$e(t)$: indice du processus élu. Si aucun processus n'est actif $e(t)$ n'est pas défini ; par convention nous poserons alors $e = 0$.

La figure 1 donne le diagramme des états des processus avec les transitions entre ceux-ci.

I.2. Modèle théorique du traitement parallèle ; notion de priorité.

En vue d'obtenir un étalon pour évaluer les performances de l'allocation TTP il nous est utile de raisonner sur le traitement parallèle. Celui-ci est défini par l'exécution simultanée de tous les processus actifs chacun étant exécuté d'autant moins vite que les processus actifs sont plus nombreux.

Il est possible d'introduire un système de priorités défini comme suit.

Une priorité π est attachée à chaque processus et les vitesses d'exécution de ceux-ci sont proportionnelles à leur priorité. Ainsi, si le processus P_1 possède une priorité π_1 supérieure à la priorité π_2 du processus P_2, il s'exécutera π_1/π_2 fois plus rapidement que ce dernier.

Pour être plus précis, introduisons les notations suivantes :

π_i désigne la priorité du processus P_i ; c'est un nombre positif constant au cours du temps,

$\gamma(t)$ est la *charge* de l'unité centrale à l'instant t, c'est-à-dire la somme des priorités des processus actifs :

$$\gamma(t) \equiv \sum_{k \text{ tels que } A_k(t) = \text{OUI},} \pi_k \tag{1}$$

$W_i(t)$ est le temps d'unité centrale affecté au processus P_i depuis le dernier instant où il est devenu actif.

Le traitement parallèle avec priorités inégales est alors défini comme suit :

$$\text{POUR TOUT } i \text{ ET TOUT } t \;\begin{cases} \text{SI } A_i(t) = \text{NON ALORS } W_i(t) = 0, \\ \text{SI } A_i(t) = \text{OUI ALORS } \dfrac{dW_i(t)}{dt} = \dfrac{\pi_i}{\gamma(t)}. \end{cases} \tag{2}$$

I.3. Hypothèses des durées de service prévisibles.

Supposons que chaque processus qui passe à l'état actif, présente une demande de service précise, c'est-à-dire qu'il réclame un traitement équivalent à la possession complète de l'unité centrale pendant un

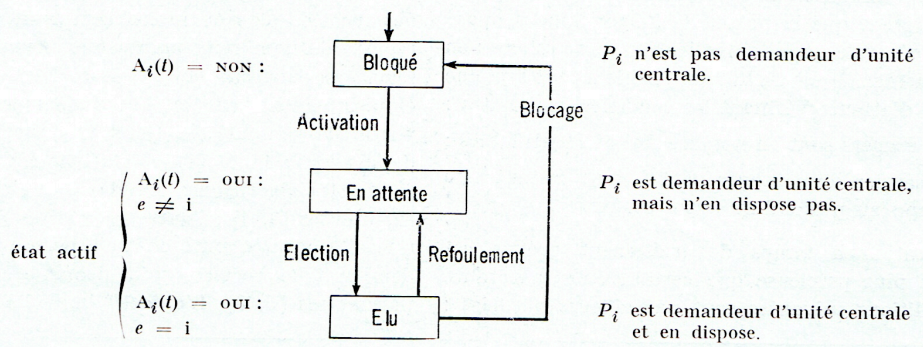


FIG. 1. — Etats des processus.

temps Q appelé *quantum*. (La façon dont le processus élabore sa demande en fonction des prévisions de blocage sera traitée plus tard.)

Provisoirement, nous supposerons qu'un processus ne se bloque jamais avant l'exécution complète du quantum qu'il a demandé. Par contre, il lui sera possible de se maintenir dans l'état actif en fin de quantum en présentant immédiatement une nouvelle demande. Le nouveau quantum demandé peut être différent du précédent.

Pour chaque processus actif nous pouvons définir $d_i(t)$ la date de demande du quantum en cours d'exécution à l'instant t (on a toujours $d_i(t) \leqslant t$). Soit, par ailleurs, $Q_i(t)$ ce quantum lui-même. Enfin, nous désignons par $f_i(t)$ la date future à laquelle le quantum en cours sera totalement exécuté.

Remarquons que $f_i(t)$ n'est pas calculable à l'instant t à moins que l'on sache comment évoluera la charge de l'unité centrale.

En effet, d'après (2) f_i et Q_i sont liés par la relation :

$$(3) \qquad Q_i(t) = \int_{d_i(t)}^{f_i(t)} \frac{d\tau}{\gamma(\tau)} .$$

La figure 2 illustre le déroulement d'une phase d'activité comprenant trois quantums.

$t_0(t)$: dernier instant antérieur à t telle que la charge soit nulle :

$$\left| \begin{array}{l} \gamma(t_0) = 0 , \\ \gamma(\tau) \neq 0 , \quad \text{pour } t_0 < \tau < t , \end{array} \right.$$

$q(t)$: date virtuelle à l'instant t définie par :

$$(4) \qquad q(t) = \int_{t_0(t)}^{t} \frac{d\tau}{\gamma(\tau)} .$$

Appelons maintenant *échéance* d'un processus actif P_i la date virtuelle de sa prochaine fin de quantum et désignons-la par la notation suivante :

$E_i(t)$: échéance de P_i à l'instant t définie par :

$$E_i(t) = q(f_i(t)) .$$

Montrons maintenant que, bien que $f_i(t)$ ne soit pas prévisible sans connaître la charge future, $E_i(t)$ est calculable dès la demande de quantum. En effet :

$$\begin{aligned} E_i(t) &= q(f_i(t)) , \\ &= \int_{t_0(t)}^{f_i(t)} \frac{d\tau}{\gamma(\tau)} , \quad \text{d'après (4)} \\ &= \int_{t_0(t)}^{d_i(t)} \frac{d\tau}{\gamma(\tau)} + \int_{d_i(t)}^{f_i(t)} \frac{d\tau}{\gamma(\tau)} , \end{aligned}$$

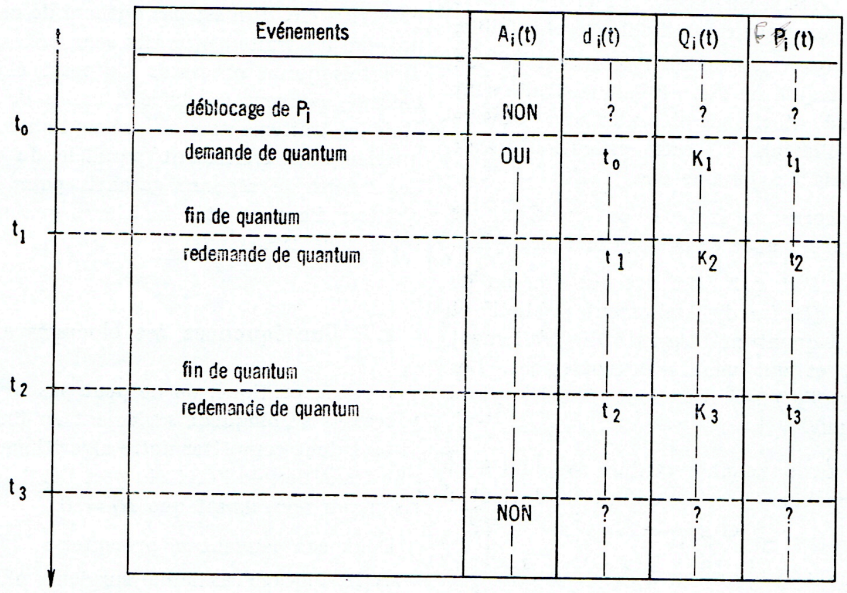


FIG. 2. — Exemple de phase d'activité comprenant trois quantums.

I.4. Notion de temps virtuel et d'échéance.

Afin de raisonner indépendamment des variations de charge de l'unité centrale, il nous sera commode d'utiliser la notion de *temps virtuel*. Intuitivement, il s'agit du temps vrai ralenti par la charge de la machine ; il n'est défini que lorsqu'il existe des processus actifs.

Les notations seront les suivantes :

$$\begin{aligned} &= q(d_i(t)) + \frac{1}{\pi_i} \int_{d_i(t)}^{f_i(t)} \frac{\pi_i d\tau}{\gamma(\tau)} , \quad \text{d'après (4)} \\ &= q(d_i(t)) + \frac{1}{\pi_i} [W_i(f_i(t)) - W_i(d_i(t))] , \quad \text{d'après (2)} \end{aligned}$$

$$(5) \quad \bullet \quad E_i(t) = q(d_i(t)) + \frac{Q_i}{\pi_i} , \quad \text{d'après (3)} .$$

● Ce signe typographique indique que la formule était encadrée dans le manuscrit.

Puisque nous savons évaluer à l'avance les dates virtuelles des fins de quantums et puisque le temps virtuel est une fonction strictement croissante du temps, nous savons aussi prévoir l'ordre des fins de quantums en traitement parallèle.

C'est cette remarque qui est à la base de l'allocation TTP telle que nous la définissons dans la suite.

I.5. Principe de l'allocation à temps de traitement prévus (TTP).

Avant de traiter des situations plus complexes, énonçons le principe de l'allocation TTP dans le cas où tous les blocages ont lieu exactement en fin de quantum et où les temps de remplacement d'un élu sont négligeables.

Principe :

a) en fonction des instants de déblocage et des quantums demandés, déterminer ce que serait l'évolution future des processus dans le cadre du traitement parallèle défini précédemment,

b) toujours allouer la totalité de l'unité centrale au processus qui, en traitement parallèle, devrait terminer le premier. Si plusieurs ont la même échéance en choisir un, par exemple celui d'indice le plus faible,

c) lorsqu'un processus termine l'exécution de son quantum plus tôt qu'il ne l'aurait fait en traitement parallèle, ne pas l'autoriser à présenter une nouvelle demande avant l'instant où cette exécution aurait théoriquement pris fin dans ce cas.

L'algorithme correspondant à ce principe est donné sur la figure 3 avec deux notations nouvelles.

$T_i(t)$: variable d'état qui vaut OUI entre la fin de quantum *effective* du traitement prédictif et la fin de quantum *théorique* du traitement parallèle, et qui vaut NON autrement. Les états où $T_i(t) =$ OUI sont appelés états *temporisants*.

Un processus temporisant contribue toujours à la charge de la machine dont la définition devient :

$$\gamma = \sum \pi_i ,$$

i tels que $A(i) =$ OUI ou $T(i) =$ OUI .

$R_i(t)$: partie du quantum actuel restant à exécuter,

soit $R_i(t) \equiv Q_i(t) - [W_i(t) - W_i(d_i(t))]$.

Pour chaque transition entre états nous donnons entre parenthèses la condition de son déclenchement, et hors parenthèses les modifications de variables qui en résultent.

I.6. Propriétés de l'allocation TTP.

L'algorithme présenté a les caractéristiques suivantes,

Si les instants de demandes de quantums sont les mêmes qu'en traitement parallèle, les fins de quantums effectives sont toujours avancées au sens large par rapport à celles du traitement parallèle.

Il n'y a pas d'algorithme permettant, avec les mêmes instants de demandes de quantums, d'avancer certaines fins de quantums sans en retarder d'autres.

En d'autres termes, l'allocation TTP est strictement supérieure au traitement parallèle et aucun algorithme ne lui est strictement supérieur.

La démonstration de ces propriétés peut se faire par récurrence. En effet, si on admet qu'à partir d'un certain instant t il ne se présente plus de nouvelle demande de quantum, il est alors possible de calculer les dates effectives où prendront fin l'exécution des quantums des processus actifs à l'instant t. On montre alors que tous ces processus épuisent leur quantum plus tôt qu'en traitement parallèle à l'exception du dernier à utiliser l'unité centrale qui termine exactement à la même date.

Il est ensuite possible de remonter à la première demande de quantum précédant l'instant t puis à la précédente et ainsi de suite sans que la propriété soit modifiée.

Une autre propriété très importante de l'allocation TTP est qu'elle limite le nombre des changements du processus élu. Il n'est pas optimal de ce point de vue puisque les traitements dits *sans préemption*, c'est-à-dire tels qu'un processus élu reste élu jusqu'à son blocage, peuvent conduire à moins de chargements. Mais les traitements sans préemption sont nettement inférieurs au traitement parallèle du point de vue des temps de réponse garantis pour les quantums courts ; d'autant plus inférieurs que certains quantums sont très longs.

I.7. Conséquences des blocages anticipés.

Dans la pratique, on ne peut pas supposer que les processus se bloquent seulement en fin de quantum. Il faut donc compléter notre algorithme en prévoyant une transition de l'état élu vers l'état bloqué, temporisant ou non, avant que $R_t = 0$.

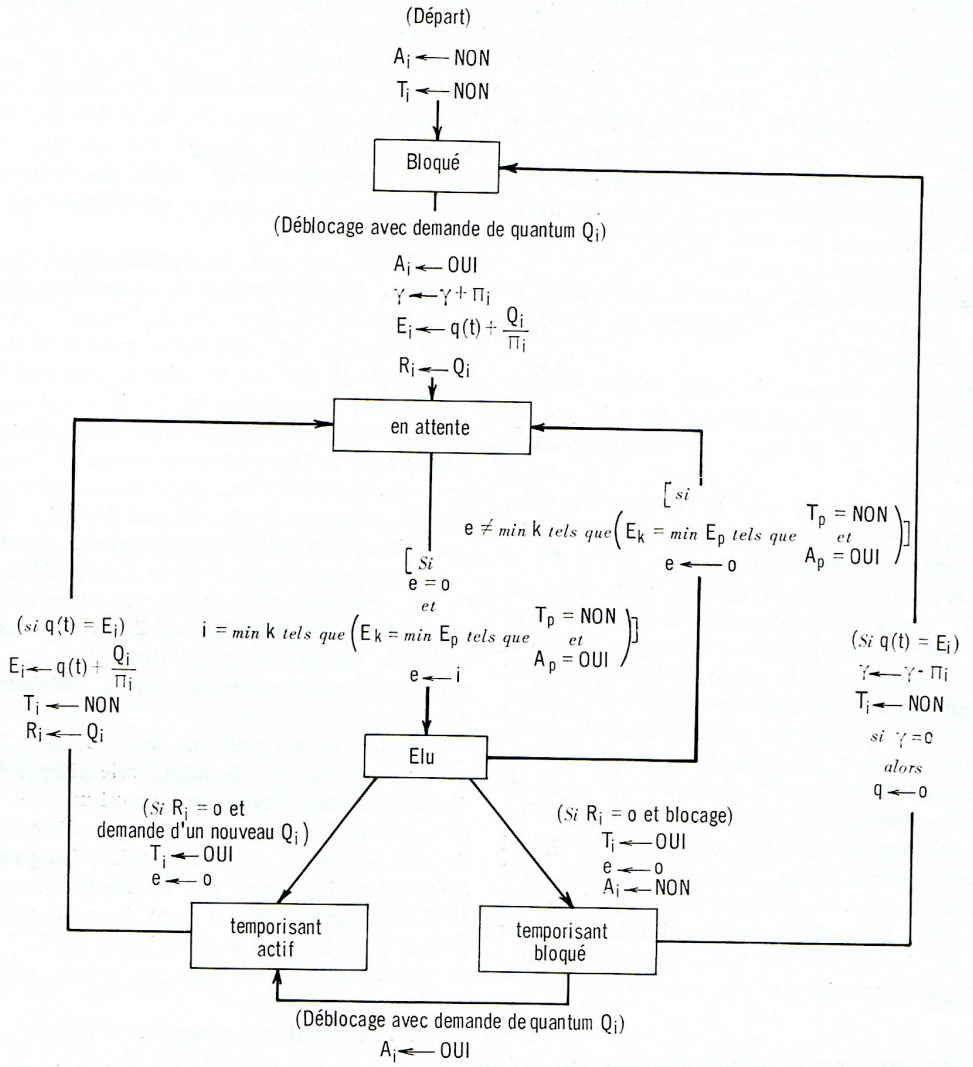
Deux cas peuvent se présenter :

a) le blocage anticipé survient plus tard avec l'algorithme prédictif qu'avec le traitement parallèle. Nous dirons que le processus est *en retard*. Cette situation peut apparaître si, par exemple, le processus demandeur d'un grand quantum a longtemps attendu l'attribution de l'unité centrale, puis s'est trouvé bloqué dès le début de son exécution ;

b) le blocage anticipé survient plus tôt avec l'algorithme TTP. Nous dirons que le processus est *en avance*. Ce peut être le cas d'un processus dont le faible quantum ou la forte priorité a accéléré son accès à l'unité centrale.

Soit P_i un processus qui se bloque à l'instant t avec $R_i(t) \neq 0$, on peut calculer l'échéance qui aurait

1) Changements d'état de chaque processus P_i



2) Progression du temps

Au départ : $\gamma = 0$; $e \neq 0$,

à tout instant : $\begin{cases} si\ e \neq 0\ alors\ dq = dt/\gamma, \\ dR_e = -dt. \end{cases}$

FIG. 3. — Algorithme d'allocation TTP si les blocages n'ont lieu qu'en fin de quantum et si les changements de processus élus sont instantanés.

été la sienne si le quantum demandé avait été $Q_i(t) - R_i(t)$ au lieu de $Q_i(t)$: l'ajustement d'échéance s'écrit simplement $E_i \leftarrow E_i - R_i/\pi_i$, d'après (5).

I.7.1. Blocage anticipé d'un processus en avance.

Si l'échéance ajustée est dans le futur, c'est-à-dire si $q(t) < E_i$, P_i est un processus en avance et il suffit de le faire temporiser jusqu'à E_i pour que tout se déroule conformément au principe de l'allocation TTP. La surévaluation de Q_i peut seulement avoir changé l'ordre dans lequel les quantums auront été exécutés, avec pour conséquence que la fin de quantum de P_i a peut-être été moins avancée par rapport au traitement parallèle qu'elle ne l'aurait été avec un quantum exact.

I.7.2. Blocage anticipé d'un processus en retard.

Si au moment du blocage anticipé l'échéance ajustée est déjà dépassée, c'est-à-dire si $q(t) > E_i$, le problème est plus complexe. Il n'est plus possible de maintenir intégralement la relation avec le traitement parallèle car P_i est parvenu à son blocage moins vite que si

le traitement avait été parallèle.

Sur l'échelle du temps virtuel, le retard subi par P_i est au plus de $(Q_i - R_i)/\pi_i$. Les conséquences d'une surévaluation de quantum sont donc limitées si l'erreur est faible en valeur relative.

Les processus autres que P_i ne peuvent qu'être accélérés du fait du blocage anticipé de celui-ci. Il n'est donc pas porté atteinte à leur garantie de service, qui veut qu'ils soient exécutés au moins aussi vite qu'en traitement parallèle pourvu que leurs quantums ne soient pas surévalués.

La prise en compte des blocages anticipés de processus en retard intervient en deux points de l'algorithme d'allocation.

1. La transition menant de l'état bloqué temporisant à l'état bloqué doit être déclenchée par la condition $q(t) \geqslant E_i$ et non plus uniquement par $q(t) = E_i$.

2. Il faut prévoir qu'il peut rester des processus temporisants alors que tous les autres sont bloqués. En effet, après le blocage anticipé d'un processus en retard P_i le temps virtuel a progressé moins vite qu'il n'aurait dû, car la charge aurait dû être réduite de π_i dès l'échéance ajustée, or, celle-ci est dépassée. Quand apparaît cette situation, le temps virtuel est avancé jusqu'à l'échéance du premier processus temporisant. Dès lors, de deux choses l'une : ou le processus passe à l'état actif et accède à l'unité centrale, ou bien le processus passe à l'état bloqué. Il faut alors répéter l'opération. Ce recalage itératif du temps virtuel prend fin dès qu'un processus entre dans l'état actif ou quand tous les *temporisants* sont éliminés, c'est-à-dire quand la charge est annulée. Notons que ce recalage du temps virtuel permet aussi de compenser les imprécisions sur le calcul de $q(t)$ pourvu que ses approximations soient toujours par défaut.

I.8. Prise en compte des temps de changement de processus élu.

Dans les systèmes d'exploitation en temps partagé, les programmes sont le plus souvent conservés sur des mémoires auxiliaires, tambours ou disques magnétiques, moins coûteuses que la mémoire principale nécessaire à leur exécution. Dans ces conditions, les temps de passage d'une exécution de processus à l'exécution d'un autre ne sont pas négligeables. Pour certains travaux supposant un dialogue entre un programme et un usager humain, chaque transaction peut ne demander qu'un calcul très court ; le temps de chargement des programmes peut dans ce cas être non seulement appréciable mais prépondérant.

I.8.1. Principe d'imputation.

Nous posons comme principe que les temps perdus en chargement et déchargement de programmes doivent être imputés, comme s'il s'agissait de temps de calcul, aux processus qui en sont la cause.

Quels sont les temps de chargement et déchargement causés par un processus P_i ? En général, une réponse quantitative précise est impossible car dans les systèmes évolués une partie plus ou moins importante des transferts se déroule en parallélisme avec l'exécution d'autres processus. On se contentera donc d'une imputation forfaitaire. Celle-ci devra être supérieure aux temps de chargements et déchargements effectifs pour que le recalage du temps virtuel (paragraphe I.7.2) puisse compenser les erreurs d'évaluation.

Quoi qu'il en soit, la responsabilité de chaque processus dans ces pertes de temps doit être clairement établie en observant les règles suivantes.

Un processus est responsable dans tous les cas des temps perdus lors du premier chargement qui suit sa demande de quantum et du déchargement consécutif à sa fin de quantum ou à son blocage.

En outre, il faut envisager le cas où il existe des processus actifs qui ont commencé leur chargement ou leur exécution, mais qui ont dû céder leur place avant d'avoir épuisé leur quantum ou de s'être bloqués.

Si le processus élu provoque alors le déchargement de tels processus pour trouver la place nécessaire à son installation en mémoire, il est responsable des pertes de temps qui en résultent ainsi que de celles qui résulteront des rechargements ultérieurs correspondants.

Considérons, par exemple, un processus P_1 déroulant un programme de calcul scientifique dépourvu d'entrées-sorties, avec un quantum de plusieurs minutes, et un processus P_2 qui, dialoguant avec l'usager, exécute à plusieurs reprises des quantums de 5 secondes. Supposons que les temps de chargement et déchargements soient tous égaux à 2 secondes ; le détail des imputations est donné sur le diagramme de la figure 4.

I.8.2. Algorithme complet.

La figure 5 présente une version de l'algorithme d'allocation TTP muni des deux extensions qui viennent d'être évoquées, à savoir :

— la possibilité d'envisager les blocages avant l'épuisement des quantums,

— la prise en compte des temps de chargement et de déchargement.

Sur la figure, S_i désigne une évaluation par excès, aussi précise que possible, des temps de chargements et déchargements imputables au processus P_i du fait de sa demande de quantum Q_i. On n'a pas introduit d'état de déchargement car on peut admettre que la phase de chargement d'un processus comprend le déchargement éventuel du processus élu précédent.

Une variable d'état supplémentaire est introduite sous le nom de X. Elle n'a de signification que s'il y a un processus élu et vaut NON tant que P_e est en phase de chargement, OUI lorsqu'il est en phase d'exécution.

L'état de X conditionne la décroissance de R_e (Fig. 5).

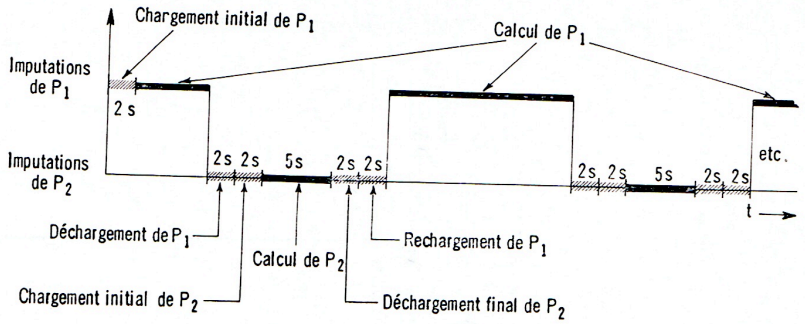


FIG. 4. — Imputations dans le cas où les temps de chargement et déchargement sont égaux.

II. LE PROBLÈME DU CHOIX DES QUANTUMS

II.1. Responsabilité des prévisions.

Dans le début de l'exposé, il est supposé que les processus ont la responsabilité de choisir leurs quantums.

Dans la pratique, le choix sera-t-il fait par le système lui-même, le programme exécuté ou l'usager ?

Le mieux est sans doute de permettre à chacun des trois de participer à la décision.

Le système offrira des valeurs par défaut choisies pour ne jamais apporter des ralentissements importants.

Les programmes peuvent contenir des appels au système pour fixer ou faire varier la progression de leurs quantums.

Qu'il fasse des prévisions directement ou à travers ses programmes, c'est toujours l'usager qui reste le plus apte à fournir des quantums optimaux. Pour cela, le système doit lui fournir les renseignements utiles sur les durées d'exécution.

II.2. Motivation des usagers.

Les principes mêmes de l'algorithme TTP incitent l'usager à soigner les prévisions des temps de calcul chaque fois que cela est possible.

En effet, le temps de réponse optimal est obtenu pour une prévision exacte.

Cependant, l'exactitude est loin d'être nécessaire pour obtenir un service voisin du traitement parallèle tant que les quantums restent dans des limites raisonnables.

Montrons sur un exemple très simple l'influence des prévisions sur le temps de réponse.

Supposons que deux processus P_1 et P_2, de même priorité, se débloquent simultanément et qu'ils soient les seuls processus actifs pendant la période étudiée.

Supposons que les valeurs exactes des temps de calculs qu'ils doivent consommer avant de se bloquer de nouveau soient respectivement :

$$C_1 = 10 \text{ minutes}, \quad C_2 = 5 \text{ minutes}.$$

En supposant que le déblocage de P_1 est toujours assorti d'une demande de calcul correspondant exactement à ses besoins, nous étudierons les temps de réponse T_1 et T_2 des deux processus quand la demande de P_2 varie.

Pour fixer les idées, notons d'abord que ces temps de réponse seraient en traitements parallèles :

$$T_1 = 15 \text{ minutes}, \quad T_2 = 10 \text{ minutes}.$$

1. Supposons que P_2 fournisse aussi une prévision exacte $Q_2 = 5$ mn, on voit alors que P_2 s'exécutera entièrement avant P_1 et les temps de réponse seront :

$$T_1 = 15 \text{ minutes}, \quad T_2 = 5 \text{ minutes}.$$

C'est pour P_2 la valeur optimale de son temps de réponse puisqu'il obtient 5 minutes de calcul en 5 minutes de temps réel.

2. Supposons que P_2 fournisse une prévision légèrement trop grande, par exemple $Q_2 = 8$ mn, l'ordre de passage du processus n'est pas modifié et le temps de réponse de P_2 reste optimal.

3. Supposons que P_2 fournisse une prévision beaucoup trop grande, par exemple : $Q_2 = 15$ mn.

L'ordre de passage du processus s'inverse :

Il vient alors :

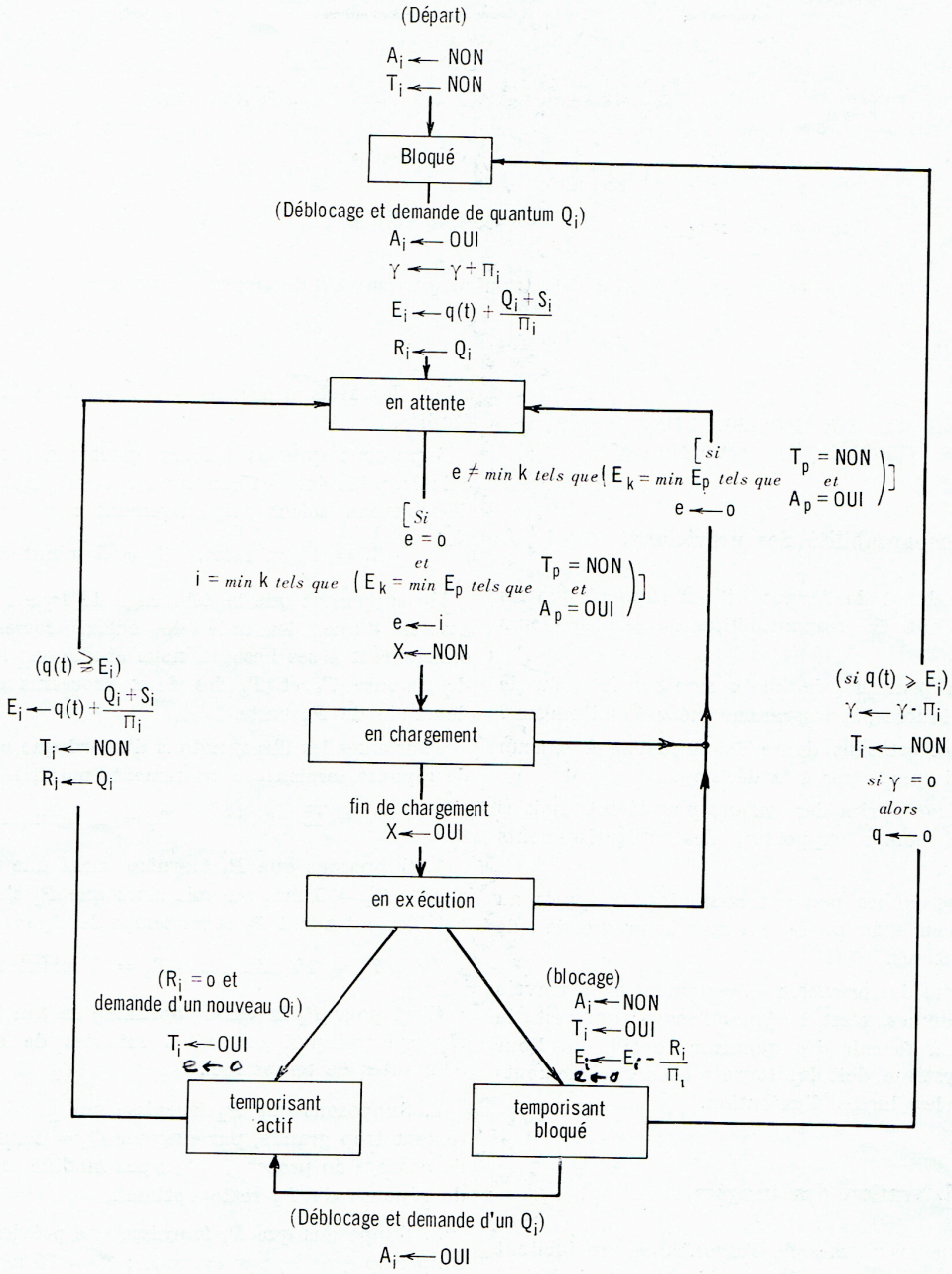
$$T_1 = 10 \text{ minutes}, \quad T_2 = 15 \text{ minutes}.$$

Le processus P_1 gagne 5 minutes par rapport au traitement parallèle tandis que P_2 les perd.

4. Supposons que P_2 demande un quantum légèrement trop court, soit $Q_2 = 4$ mn, et qu'il renouvelle la même demande pour terminer son calcul.

Le diagramme de la figure 6 donne l'état des processus P_1 et P_2 en fonction du temps.

1) Changements d'état de chaque processus P_i



2) Progression du temps

au départ : $\gamma = 0$; $e = 0$

à tout instant :

tant que $e = 0$ *et il existe* k *tel que* $T_k =$ OUI
faire : $q \leftarrow min\ E_k$ *tel que* $T_k =$ OUI

si $e \neq 0$ *alors* $dq = dt/\gamma$
si $X =$ OUI *alors* $dR_e = -dt$.

FIG. 5. — Algorithme d'allocation TTP avec blocages possibles à tout instant si les temps de changements de processus élu ne sont pas négligeables.

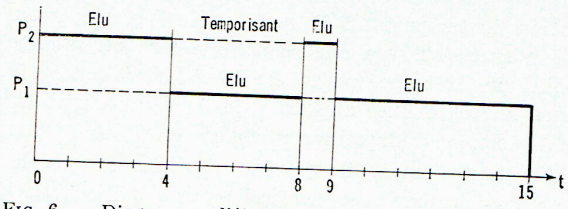


FIG. 6. — Diagramme d'état des processus P_1 et P_2 en fonction du temps.

Les temps de réponse sont alors :

$$T_1 = 15 \text{ minutes}, \quad T_2 = 9 \text{ minutes}.$$

Le processus P_2 reste encore avantagé par rapport au traitement parallèle mais beaucoup moins que dans le cas de la prévision exacte ou légèrement supérieure.

5. P_2 demande une suite de quantum très courts, disons de 10 millisecondes, choix qui, disons le tout de suite, est particulièrement déraisonnable.

Ici, le problème devient assez différent car le temps de chargement et de déchargement (dont nous n'avons pas tenu compte jusqu'à présent) ne peut plus être négligé puisque P_2 sera obligé de renouveler 300 000 fois sa demande de quantum avant de se bloquer.

Supposons pour simplifier que ce temps de chargement et de déchargement soit également de 10 ms. Le diagramme des changements d'états des processus au cours du temps sera la répétition de la maille élémentaire représentée figure 7.

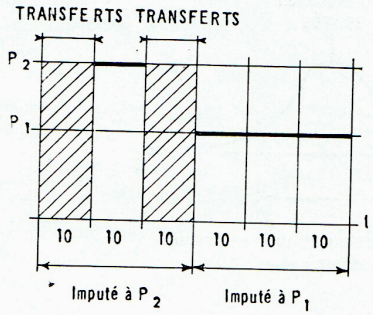


FIG. 7. — Maille répétitive du diagramme d'état des processus P_1 et P_2.

Les zones hachurées correspondent aux temps de commutation imputés au processus P_2.

L'étude de cette maille qui dure 60 ms montre que le processus P_1 calcule trois fois plus vite que le processus P_2 et que ce dernier consomme globalement trois fois plus de temps d'unité centrale pour le même temps de calcul. Le coût de l'exécution de P_2 est donc multiplié par trois.

Comme le processus P_1 calcule la moitié du temps, il terminera au bout du temps $T_1 = 20$ mn alors que P_2 n'aura reçu que $20/6 = 3{,}33$ mn de calcul utile d'où :

$$T_2 = 20 + (5\text{-}3{,}33) = 21{,}66 \text{ mn}.$$

Si nous voulons maintenir la comparaison avec le traitement parallèle, il faut tenir compte du fait que P_2 a consommé globalement 11,66 mn d'unité centrale, et les temps de réponse des processus en traitement parallèle seraient également $T_1 = 20$, $T_2 = 21{,}66$.

On peut résumer l'enseignement tiré de cet exemple en traçant la courbe $T_2 = f(Q_2)$ donnant le temps de réponse en fonction du quantum choisi (Fig. 8).

Bien que cette courbe soit très particulière, la discontinuité au voisinage de la valeur exacte de la

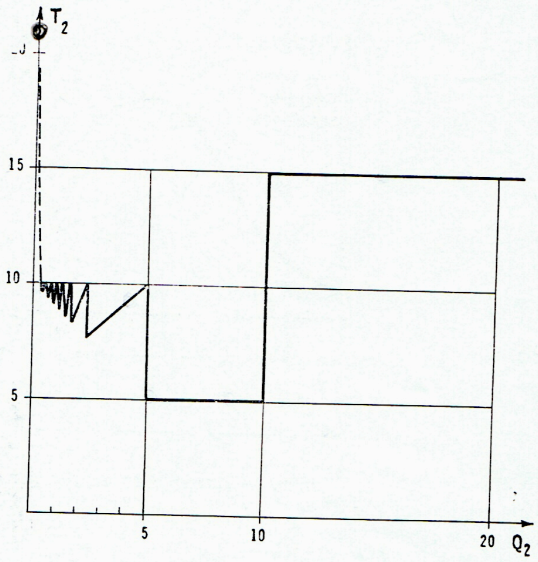


FIG. 8. — Temps de réponse en fonction du quantum choisi.

prévision reste une propriété générale. Quand le quantum choisi passe en dessous de la durée exacte du calcul, le temps de réponse devient au moins égal à celui du traitement parallèle.

Il en résulte que si l'usager connaît la durée du calcul avec une certaine marge d'erreur, il aura intérêt à fournir comme quantum la borne supérieure de son estimation.

II.3 Politique de choix.

Supposons maintenant que l'usager connaisse non pas la durée de son calcul, mais la distribution de probabilité de son temps de calcul. S'il connaît en outre le temps de chargement et de déchargement de son processus, il peut théoriquement en déduire la suite optimale des demandes de quantum qu'il doit présenter pour minimaliser l'espérance mathématique de son temps de réponse.

Dans la pratique, le temps de réponse ne sera pas substantiellement moins bon qu'en traitement parallèle si :

1. la durée totale de calcul est largement supérieure à la somme des temps de chargements et de déchar-

gements imputables au processus (pour que les pertes de temps dues aux transferts restent faibles) ;

2. le dernier quantum demandé n'est pas largement supérieur à la durée totale de calcul du processus (pour éviter le risque d'une attente inutile).

Considérons l'exemple suivant où la densité de probabilité p(x), définie par

Prob ($x \leqslant$ durée de calcul $< x + \mathrm{d}x$) = p(x) dx ,

a l'allure indiquée sur la figure 9.

ration du temps total d'exécution de la tâche, mais s'il admet qu'il y a une probabilité non négligeable pour que la tâche avorte en fin de compilation, il aura intérêt à fixer une suite de deux quantums C_1 et C_2 comme indiqué sur la figure 10.

De toute façon, il apparaît souhaitable que le système fournisse quelques modèles ou lois d'évolution des quantums dont les paramètres pourront être ajustés par l'usager de façon à mieux s'adapter à sa situation.

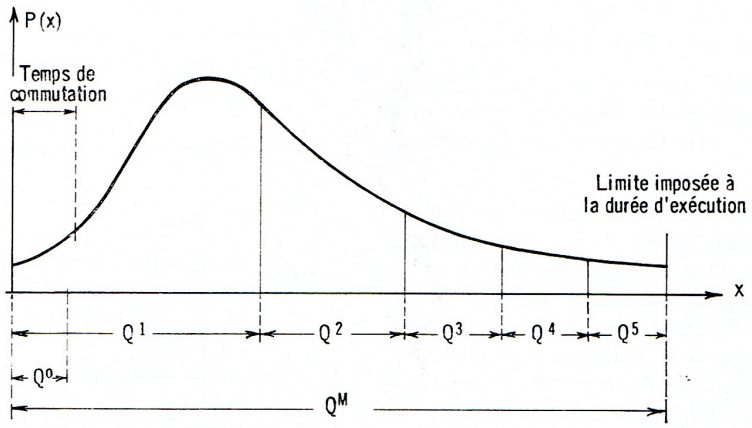


FIG. 9. — Exemple de distribution de probabilité du temps de calcul prévu.

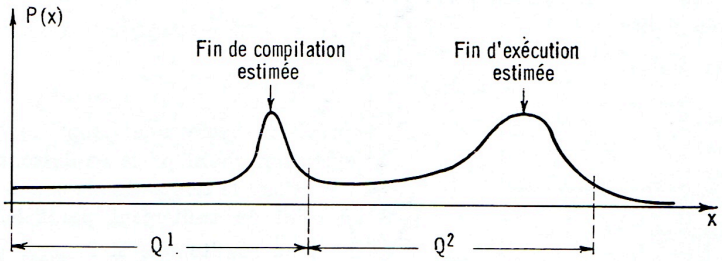


FIG. 10. — Choix des quantums par une compilation et une exécution de durées approximativement connues.

Une suite de demandes de quantum telle que Q^1 , Q^2 , ... Q^5 est tout à fait satisfaisante dans la pratique.

Par contre, prendre un seul quantum $Q^m = Q^1 + Q^2 + \ldots Q^5$ serait dangereux à cause de 2).

De même, prendre une suite de quantum voisin de Q_0 ou plus petit serait néfaste à cause de 1).

Autre exemple : soit un usager dont le programme, déjà mis au point, ne nécessite une fois lancé aucune intervention extérieure. Après avoir apporté quelques modifications, l'usager désire à nouveau compiler et exécuter ce programme. Il connaît donc avec une assez bonne précision les durées des phases de compilation et d'exécution.

S'il est absolument sûr de ses corrections, il peut fixer un quantum correspondant à une légère majo-

II.4. Dispositif expérimenté sur le calculateur Ramsès II.

II.4.1. Commande de choix des quantums.

Sur le calculateur Ramsès II, l'algorithme TTP est utilisé pour gérer les tâches FORTRAN (compilations, chargements et exécutions) soumises par les usagers. Pendant la phase d'exécution il y a possibilité de conversation entre l'usager et son programme FORTRAN.

L'usager dispose d'une commande qui lui permet de fixer un certain nombre de paramètres avant de lancer l'exécution d'une tâche ou encore de les modifier en interrompant momentanément la tâche en cours d'exécution.

Ces paramètres sont les suivants :

C : durée prévue de la première tranche de calcul (souvent la compilation et le chargement),

E : premier quantum demandé après un déblocage,

M : facteur de multiplication permettant d'obtenir une suite de quantums croissante ou décroissante lorsque ceux-ci sont épuisés sans qu'il y ait blocage.

A chaque épuisement d'un quantum Q celui-ci est remplacé par le produit MQ.

M est un nombre positif supérieur ou inférieur à l'unité. La durée d'exécution prévue peut aussi être augmentée ou diminuée progressivement.

N : le nombre maximal de multiplication du quantum Q par le facteur M. Quand le nombre de quantums épuisés successivement atteint la valeur N, Q cesse de progresser.

Il est à noter que l'emploi de cette commande est facultative. Si l'usager ne veut pas l'utiliser le système fixe les valeurs des paramètres C, E, M et N. Les valeurs standards des paramètres C et E sont suffisamment grandes pour que les performances du système ne soient pas perturbées. Seul l'usager qui les utilise aura un manque à gagner si elles ne sont pas optimales.

II.4.2. Valeurs fixées par défaut des paramètres.

Les valeurs par défaut sont en fonction du temps moyen de chargement et de déchargement τ :

$$C = 8\,\tau, \quad E = 2\,\tau, \quad M = 1{,}5, \quad N = 2.$$

La suite des quantums proposés est donc après le premier blocage :

$$Q_1 = 2\,\tau,$$
$$Q_2 = 3\,\tau,$$
$$Q_3 = 4{,}5\,\tau,$$
$$Q_4 = 4{,}5\,\tau.$$

Etudions les effets de cette suite sur la garantie de service de l'usager.

Pour ne pas tenir compte de la charge de la machine, nous utiliserons l'échelle du temps virtuel pour repérer les événements.

Soit x la quantité de calcul à effectuer. Elle peut s'écrire :

$$x = \sum_0^n Q_i + y, \quad \text{avec } y \leqslant Q_{n+1}, \; Q_0 = 0.$$

Le temps de réponse garanti sur l'échelle virtuelle est donc :

$$T' = \sum_0^{n+1} Q_i + (n+1)\,\tau,$$

alors qu'avec la prévision exacte, il serait comme en traitement parallèle :

$$T'' = \sum_0^n Q_i + y + \tau.$$

Le ralentissement virtuel est :

$$\frac{T'}{T''} = \frac{\sum_0^n Q_i + \tau + n\tau + Q_{n+1}}{\sum_0^n Q_i + \tau + y}.$$

Il passe par un maximum relatif à chaque fois que $y \dashrightarrow 0 +$ c'est-à-dire que la quantité de calcul excède de très peu un nombre exact de quantums.

Ces maximums relatifs ont pour valeur :

$$\frac{T'}{T''} = 1 + \frac{Q_{n+1} + n\tau}{\sum_0^n Q_i + \tau}.$$

Dans le cas envisagé, ces valeurs successives sont données dans le tableau I.

TABLEAU I

n	$\sum_0^n Q_i$	Q_{n+1}	$\sum_0^{n+1} Q_i$	x	T'/T''
0	0	2	2	ε	3
1	2	3	5	$2 + \varepsilon$	2,33
2	5	4,5	9,5	$5 + \varepsilon$	2,1
3	9,5	4,5	13	$9{,}5 + \varepsilon$	1,78
4	13	4,5	17,5	$13 + \varepsilon$	1,57

Le ralentissement peut surtout apparaître pour les durées de calcul très courtes, c'est-à-dire pour les tâches de dialogue avec l'usager.

Mais quand $x \dashrightarrow \infty$ $T'/T'' \dashrightarrow 5{,}5/4{,}5 = 1{,}22$.

CONCLUSION

Nous avons présenté un algorithme applicable à toute ressource demandée de façon répétée par un nombre fini d'usagers et commutable d'un usager à l'autre si nécessaire. L'unité centrale d'un ordinateur exploité en temps partagé n'est qu'un cas particulier d'une telle ressource.

L'algorithme permet d'affecter aux usagers des coefficients de priorité tel que la vitesse moyenne de service de chacun soit proportionnelle à sa priorité et, bien entendu, inversement proportionnelle à la somme des priorités en concurrence.

Si les usagers peuvent prévoir exactement la durée du service qu'ils demandent, ils seront servis dans un temps minimal compatible avec la même garantie aux autres usagers. S'ils ne le peuvent pas, ils décomposent leur demande en une suite de sous-demandes. Dans ce cas et pourvu que la progression des durées de celles-ci soit judicieuse, le temps de service effectif le moins bon restera toujours de l'ordre du temps de service garanti en cas de prévision exacte.

L'extension du principe de l'allocation à temps de traitement prévus au cas de plusieurs ressources de même type n'a pas été abordé dans cet article mais ne soulève pas de difficulté majeure ; ce serait le cas pour un système à plusieurs unités de traitement ayant accès à des mémoires ou des organes communs et exploité en temps partagé.

La réalisation effective du programme d'allocation sur RAMSÈS II *est due non seulement aux auteurs mais aussi à M. Henriot, membre de l'équipe de recherche en programmation des systèmes du département* CALCULATEURS ÉLECTRONIQUES ET SYSTÈMES DU CNET.

Manuscrit reçu le 27 *octobre* 1971.

BIBLIOGRAPHIE

[1] CORBATO (F. J.), MERWIN-DAGGETT (M.), DALEY (R. C.), An experimental time-sharing system (Système expérimental de temps partagé), *in* Programming systems and languages. *McGraw-Hill*, New York (1967), pp. 683-698.

[2] VYSSOTSKY (V. A.), CORBATO (F. J.), GRAHAM (R. M.), Structure of the Multics supervisor (Structure du superviseur Multics). Proceedings fall joint computer conference (1965). *AFIPS Press*, U. S. A. (1965), **27,** pp. 203-212.

[3] KLEINROCK (L.), A continuum of time sharing scheduling algorithms (Un continuum d'algorithmes d'allocation en temps partagé). Proceedings of the spring joint computer conference, *A.F.I.P.S. Press*, U. S. A. (1970), **36,** pp. 453-458.

[4] HANSEN (P. B.), An analysis of response time ratio scheduling (Une analyse de l'allocation d'après les rapports de temps de réponse). Proceedings of the *I.F.I.P.* Congress 1971, *North Holland publishing Co.* (août 1971), Fasc. TA-3, pp. 150-154.

[5] SALTZER (J. H.), Traffic control in a multiplexed computer system (La commande du trafic dans un système informatique multiplexé). Thèse MAC-TR-30, *M.I.T. Cambridge Massachussets*, U. S. A. (juillet 1966), 79 p.

[6] BERSTEIN (A. J.), SHARP (J. C.), A policy driven scheduler for a time sharing system (Un allocateur fondé sur les politiques d'utilisation pour système en temps partagé). *Communications of the A.C.M.*, U. S. A. (févr. 1971), **14,** nº 2, pp. 74-78.

Imprimerie Jacques et Demontrond, 26, rue Ernest-Renan - Besançon