

UN MODELE D'ANALYSE DES PERFORMANCES D'ORDINATEURS MULTIPROGRAMMÉS A MÉMOIRE VIRTUELLE

D. Potier, J. Leroudier, M. Badel
IRIA/LABORIA

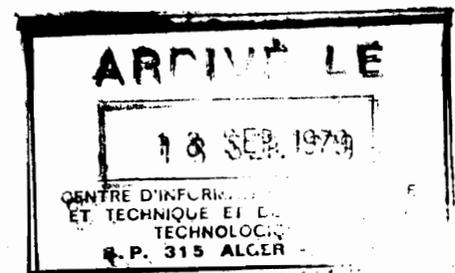
Abstract :

We present in this paper a global model of a virtual memory computer system. The model is analyzed and we point out simple relations between architecture parameters, program behaviour parameters and some performance measures such as resource utilization or overhead ratio. An analytical and a simulation method are used to solve the model and the comparison of the results which are obtained show that the analytical solution provides satisfactory results even if the necessary assumptions do not hold.



Résumé :

Nous développons dans cet article un modèle global d'ordinateur multiprogrammé à mémoire virtuelle. Le modèle est analysé et un certain nombre de lois de fonctionnement sont dégagées qui mettent en évidence des relations simples entre les paramètres de l'architecture, les paramètres de comportement des programmes et les mesures de performances : utilisation des ressources, taux d'overhead. Deux méthodes de résolution, par analyse mathématique et par simulation sont présentées et les résultats obtenus montrent que la méthode de résolution mathématique fournit, même dans le cas où les hypothèses nécessaires à son application ne sont pas satisfaites, des résultats comparables à ceux obtenus par simulation.



BIBLIOTHEQUE DU CERIST



BIBLIOTHEQUE DU CERIST

I - INTRODUCTION

Les années soixante avec la machine ATLAS[30] ont vu l'apparition des premiers ordinateurs à mémoire virtuelle [19]. Le mécanisme de traduction dynamique des adresses permettait de résoudre élégamment les problèmes de gestion et de partage de la mémoire dans les ordinateurs importants de l'époque conçus pour travailler en temps partagé. Les programmes pouvaient alors être déchargés de la mémoire principale et y être rechargés à des emplacements physiques différents, sans que soit nécessairement préservée la contiguïté logique de l'information dans la mémoire physique.

En plus d'un avantage propre au temps partagé, ce mécanisme permet d'exécuter des programmes dont l'espace d'adressage a une taille supérieure à celle de la mémoire physique de la machine, sans que pour cela l'utilisateur ait à construire des arbres de recouvrement (overlay). Cette possibilité est mise en oeuvre en ne chargeant que la partie utile d'un programme à un instant donné (les instructions en cours et les données accédées) et en modifiant cette partie utile à la demande au cours de l'exécution (chargement "à la demande"). C'est alors que le terme de mémoire virtuelle prend son plein sens et que le mécanisme de traduction dynamique des adresses est exploité au mieux de ses possibilités.

Une telle solution se généralisa rapidement à d'autres machines telles que GE 645 et IBM 360/67 sur lesquelles furent implémentés des systèmes d'exploitation qui utilisaient les mécanismes de traduction d'adresses : MULTICS [9] MTS [42], CP67 [34]. Actuellement, ces mécanismes sont disponibles sur les ordinateurs de haut de gamme récents tels que IBM 370 ou CII-IRIS 80 exploités sous les systèmes d'exploitation VM [27] ou VS chez IBM, et GEMAU [26] développé à l'Université de Grenoble.

Toutefois, les avantages apportés aux utilisateurs par la technique de la mémoire virtuelle se traduisent par une charge supplémentaire sur certaines ressources du système. D'une part, les algorithmes d'allocation dynamique des blocs d'information, ou page, ainsi que les tables nécessaires pour réaliser la traduction entre adresses virtuelles et adresses réelles augmentent le volume du système d'exploitation et réduisent d'autant l'espace mémoire disponible aux utilisateurs. D'autre part, par le jeu du mécanisme d'allocation dynamique, chaque fois qu'une page référencée ne se trouve pas en mémoire principale, le programme en cours d'exécution est interrompu, son contexte sauvegardé, les algorithmes de pagination sont exécutés et les transferts de la page requise et éventuellement de la page à décharger sont initialisés. Ces opérations entraînent un dépassement ("overhead") du temps d'unité centrale au détriment de l'exécution des programmes utilisateurs. Enfin, la pagination a pour effet d'imposer un échange important d'information entre la mémoire principale et la mémoire secondaire qui supporte la mémoire virtuelle. On conçoit alors que le mécanisme de pagination va faire apparaître, à forte charge, un goulot d'étranglement sur la mémoire secondaire en imposant des temps d'attente à chaque transfert de pages de plus en plus importants dûs à l'encombrement du canal de pagination. Lorsque le nombre de programmes qui se partagent les ressources du système augmente, le chargement à la demande des pages et la libre compétition des programmes dans la mémoire physique provoquent un phénomène en avalanche connu sous le nom d'écroulement ou thrashing [21, 38] qui peut conduire à une dégradation importante des performances du système.

Nous voyons donc qu'à côté de problèmes spécifiquement "système" liés à la gestion des espaces virtuels, le fonctionnement des machines à mémoire virtuelle pose des problèmes de performances qu'il est difficile d'appréhender directement en raison de la complexité même de l'organisation de ces machines. Durant ces dix dernières années, de nombreux travaux ont été publiés dans cette direction pour étudier les performances de différents algorithmes de pagination [11, 20, 22, 36] et pour évaluer les performances de systèmes à mémoire virtuelle [5, 24, 35, 37, 41]. Dans le même temps s'est développé l'étude des lois de comportement des programmes vis à vis des ressources qu'ils utilisent : mémoire centrale, unité entrée-sortie, unité centrale, et plusieurs modèles de ces comportements ont été proposés [8, 11, 13, 17, 31]. Associé à un modèle d'architecture de système, ce type de modèle permet de caractériser et d'analyser globalement les performances d'une architecture en faisant apparaître les relations entre les phénomènes de comportement de programme et l'organisation de l'architecture, et en permettant une analyse quantitative de ces relations [5].

Nous développons dans cet article un modèle global d'ordinateur multiprogrammé à mémoire virtuelle à partir de l'approche présentée dans [5]. Le modèle est organisé pour prendre en compte les aspects essentiels de tels systèmes, et ne représente donc pas le modèle d'un système particulier. Toutefois, le lecteur qui souhaiterait se référer à un système concret peut trouver un exemple simple dans le système CP-67 [34] ou VM [27]. Le but de cette étude est de proposer une analyse quantitative du fonctionnement d'un ordinateur multiprogrammé à mémoire virtuelle, et d'atteindre ainsi à une compréhension physique des phénomènes.

Le paragraphe 2 est consacré à la présentation du modèle utilisé et des hypothèses retenues. Dans le paragraphe 3 le modèle est analysé et un certain nombre de lois de fonctionnement sont dégagées qui mettent en évidence des relations simples entre les paramètres de l'architecture, les paramètres de comportement des programmes et les mesures de performances : utilisation des ressources, taux d'overhead. Deux méthodes de résolution, par analyse mathématique et par simulation sont présentées au paragraphe 4 et les résultats obtenus montrent que la méthode de résolution mathématique fournit, même dans le cas où les hypothèses nécessaires à son application ne sont pas satisfaites, des résultats comparables à ceux obtenus par simulation. Dans le dernier paragraphe, le modèle est utilisé pour étudier le comportement du système modélisé sous différentes hypothèses de fonctionnement et dégager quelques principes de gestion d'un système multiprogrammé à mémoire virtuelle.