

Séance : Analyse du comportement d'un cache

L'objectif de ce travail dirigé est de présenter un problème de performance de politiques de gestion de mémoire virtuelle. Les applications de la gestion de zone de mémoire se retrouvent dans beaucoup de domaines de l'informatique et des télécommunications (hiérarchies de caches matériels, proxies internet, caches internet, caches d'adresses...)

L'accent sera plus précisément porté sur la méthodologie de modélisation de tels algorithmes et sur l'interprétation des résultats mathématiques obtenus par le traitement des modèles.

Modèle

La mémoire (virtuelle) d'un ordinateur est structurée en un ensemble de pages mémoires. Un programme, s'exécutant sur le processeur, appelle successivement les pages mémoire dont il a besoin, effectue le traitement associé à ces pages et replace la page traitée dans la mémoire. Pour l'utilisateur, la gestion des pages est entièrement transparente, la programmation se fait sans contrainte d'usage de la mémoire.

Le système d'exploitation est chargé de la gestion et de l'organisation de la mémoire. On supposera que la mémoire est localisée sur 2 sites physiques différents (la mémoire du processeur et le disque de pagination). Dans la plupart des ordinateurs, le temps d'accès à une page sauvegardée sur le disque est bien plus grand que le temps d'accès à une page située en mémoire. Le système d'exploitation cherchera donc à minimiser le nombre d'accès à des pages situées sur le disque, de tels accès sont appelés défauts de page car la page fait défaut dans la mémoire.

Pour notre étude, nous supposons que la taille de la page est fixée, que le nombre total de pages est égal à N et que la capacité de la mémoire est de M pages. On supposera les pages distinctes (pas de duplication de pages) et identifiées par un numéro de page de 1 à N . Les pages seront sauvegardées dans N places de 1 à M pour les pages en mémoire et de $M + 1$ à N pour les pages sauvegardées sur disque. En simplifiant, on pourrait caractériser l'état de la mémoire, à un instant donné, comme une permutation de $\{1, \dots, N\}$.

	Mémoire Virtuelle							
	Mémoire			Disque				
Adresses	1	2	3	4	5	6	7	8
Pages	P_3	P_7	P_2	P_6	P_5	P_1	P_8	P_4

Par exemple, lorsque le système à une capacité de $N = 8$ pages et que sa mémoire peut contenir $M = 3$ pages, un état du système est déterminé par un N -uple, ici $E = (P_3, P_7, P_2, P_6, P_5, P_1, P_8, P_4)$. Cela signifie, dans cet exemple, que la page P_6 est à l'adresse 4.

On envisage deux stratégies pour la gestion des pages :

La stratégie Move-To-Front Dans cette stratégie, on suppose que lorsqu'une page est appelée, elle a toutes les chances d'être rappelée prochainement (phénomène de localisation des données d'un programme). Après le traitement effectué sur la page, on la replace en première position "physique" dans la mémoire. Dans ce cas il faut décaler les autres pages et, le cas échéant, sauvegarder sur le disque la page la plus ancienne en mémoire. Dans les systèmes d'exploitation, cette stratégie est appelée LRU (pour Last Recently Used).

Il faut remarquer que les opérations de décalage de page en mémoire ou sur le disque se font par permutation d'adresses physiques de pages (pointeurs), opérations qui s'effectuent en un temps négligeable devant le temps d'un transfert mémoire/disque.

La stratégie Move-Ahead Dans cette stratégie, on suppose que lorsqu'une page est appelée, on doit attendre un peu pour savoir si elle est fréquemment appelée ou pas. On essaye de ne garder en mémoire que les pages fréquemment appelées sur une période de temps. Pour cela chaque page est associée à un rang dans le système. Lorsque une page est appelée, elle avance d'un rang.

Exemple : si la mémoire est dans l'état de la figure précédente et qu'on appelle la page P_5 , le système passe dans l'état :

- $(P_5, P_3, P_7, P_2, P_6, P_1, P_8, P_4)$ si la stratégie est Move-To-Front
- $(P_3, P_7, P_2, P_5, P_6, P_1, P_8, P_4)$ si la stratégie est Move-Ahead

Le système est modélisé par une chaîne de Markov dont les états sont des permutations de $\{P_1, \dots, P_N\}$. Les transitions de cette chaîne de Markov sont caractérisées par la stratégie de gestion de pages.

Exercice 1. Modélisation markovienne

indispensable

a. Justifier le choix d'une modélisation par une chaîne de Markov.

b. Si on suppose que l'espace mémoire est de $4Go$, et la taille du disque de Sawpp $4Go$, que la taille d'une page est de $4Ko$, quel est l'ordre de grandeur de la taille de l'espace d'états ?

Pour simplifier le problème, on fera les hypothèses suivantes :

Indépendance des demandes de pages La séquence des pages demandées forme une séquence de variables aléatoires $\{M_n\}_{n \in \mathbb{N}}$, indépendantes à valeur dans $\{1, \dots, N\}$.

Page la plus probable On supposera que l'une des pages, P_A , a la plus forte probabilité a d'appel et que, sans perte de généralité, toutes les autres pages seront demandées avec une même probabilité b , $b < a$.

Le processus aléatoire étudié sera donc $\{X_n\}_{n \in \mathbb{N}}$ la position de la page A (la plus probable) au n -ième appel de page du programme.

c. Justifier pourquoi étudier le processus $\{X_n\}$. Quel est maintenant la taille de l'espace d'états ?

Exercice 2. Étude théorique

indispensable

On commencera par étudier dans un premier cas la stratégie **Move-To-Front** puis on répondra ensuite aux mêmes questions pour la stratégie **Move-Ahead**.

a. Justifier pourquoi $\{X_n\}$ est une chaîne de Markov et donner son graphe de transition.

b. La chaîne est-elle irréductible ? Apériodique ?

c. Calculer la mesure stationnaire de la chaîne.

Exercice 3. Comparaison des deux stratégies : analyse numérique

à terminer à la maison

En prenant $N = 8$, $M = 3$, $a = 0.3$ et $b = 0.1$, étudier le vecteur de probabilité stationnaire pour chaque stratégie.

a. Calculer à l'état stationnaire la probabilité de défaut de page due à la page A .

b. Toujours à l'état stationnaire, calculer (numériquement) la probabilité de défaut de page.

c. Quelle semble être la stratégie à préférer selon vous ?

Exercice 4. Simulation

pour aller plus loin

On se servira du programme disponible sur la page web du cours (fichier `simulation_paginee.tar.gz`).

a. Vérifiez les résultats numériques obtenus précédemment.

b. Que peut on dire de la vitesse de convergence vers les mesures stationnaires ?

c. Calculer la seconde valeur propre des deux matrices de transition (numériquement ou analytiquement, à vous de choisir). Quelles informations nous apporte-t-elle ?

d. Tester la robustesse du modèle en étudiant une distribution de probabilité plus complexe. Retrouvez vous les mêmes résultats ?