CHAPITRE 3

ORDONNANCEMENT DYNAMIQUE

Plan

CONTEXTE DE L'ORDONNANCEMENT DYNAMIQUE
PARTAGE DES UNITÉS CENTRALES ENTRE LES PROCESSUS
États et transitions d'états
Allocation des unités centrales aux processus
Ordonnancement temps réel
Inversion de priorité
Autres ordonnancements de même nature
CANAL D'ACCÈS AUX DISQUES ET ORDONNANCEMENT DES PROCESSUS
Organisation de l'espace sur les disques
Politique de déplacement du bras porte têtes
Politique de service rotationnel
Serveur disque de flux multimédia

Manuel 3

Ordonnancement dynamique

1. Contexte de l'ordonnancement dynamique

Ce chapitre traite de l'allocation dynamique d'une ressource commune qui doit être utilisée en exclusion mutuelle et qui peut éventuellement être réquisitionnée (on dit que le client est préempté). On fait l'hypothèse simplificatrice que toutes les autres ressources partagées dont les clients ont besoin leur ont été allouées, totalement ou en quantité suffisante pour la durée d'utilisation de cette ressource. On évacue ainsi les aléas futurs dus aux demandes d'autres ressources. On ne gère donc qu'une ressource et les critères d'allocation et d'optimisation qui régissent l'attente et le service ne prennent pas en compte les autres ressources. Ainsi l'attente et le service peuvent se traiter par un ordonnancement simple et la perte d'optimalité est compensée par un gain de temps de gestion. On ne peut pas faire d'ordonnancement statique car les occurrences des demandes et l'utilisation des ressources sont inconnues.

On s'intéresse à deux ressources matérielles importantes dans les systèmes : l'unité centrale, dont le temps de commutation d'un client à un autre est constant ; le canal d'accès au disque où le temps de commutation d'une requête à la suivante varie considérablement selon la position demandée des bras porte-têtes.

2. Partage des unités centrales entre les processus

2.1. États et transitions d'états

Dans la plupart des systèmes, il y a plus de processus que d'unités centrales. Celles-ci doivent être partagées. Un processus peut être dans deux états : soit il est bloqué parce qu'il lui manque des données ou parce qu'il attend un signal de synchronisation, soit il est actif, c'est à dire qu'aucune condition n'empêche son programme de s'exécuter. Pour tenir compte du partage de l'unité centrale (ou des unités centrales), il faut diviser cet état actif en deux nouveaux états, prêt et élu. Un processus actif est élu quand il utilise une unité centrale, sinon il est prêt.

Quand un processus quitte l'unité centrale, soit parce qu'il est préempté, soit parce qu'il se met en attente, le noyau du système doit poser un point de reprise tel que ce processus puisse reprendre plus tard son exécution là où il l'avait laissée. Le changement d'allocation d'une unité centrale s'accompagne d'une commutation de contexte réalisant la pose d'un point de reprise pour le dernier processus élu et le chargement du dernier point de reprise du nouvel élu. Cette commutation de contexte, faite par le noyau, prend un temps fixe, qui peut être long quand le contexte changé est important, ce qui est de règle dans des systèmes à espace d'adressage virtuel.

2.2. Allocation des unités centrales aux processus

Comme on ne connaît à l'avance ni les dates de déclenchement ni la durée d'exécution des processus, il faut réaliser une allocation dynamique au fur et à mesure des demandes. Si il y a attente des clients, on essaie de gérer cette attente avec un objectif global, comme réduire le temps d'attente moyen des requêtes, ou réduire le temps d'attente moyen des requêtes interactives en supposant que celles-ci sont les requêtes les plus courtes. Dans une application temps réel où on connaît la période d'arrivée et la durée d'exécution, on peut se donner comme objectif le respect des échéances temporelles.

L'ordonnancement des processus clients peut se fonder sur plusieurs politiques.

- La gestion de la file d'attente des clients peut conduire à les traiter à l'ancienneté ("FIFO"), ou avec des priorités fixes (déterminées de façon empirique, ou en fonction de la périodicité d'arrivée), ou encore avec des priorités variables (dépendant de l'attente écoulée, du service déjà reçu, de la nature de la requête, de l'échéance).
- La durée du service peut être laissée au choix du client et alors le processus s'exécute jusqu'à ce qu'il fasse un appel au système ; on peut l'interrompre et réquisitionner l'unité centrale quand arrive une requête plus prioritaire ; on peut allouer l'unité centrale pour une durée maximale appelée quantum.
- La structuration de l'allocateur permet d'introduire des étapes dans la politique d'allocation. On distingue ainsi un schéma sans recyclage où tous les clients sont dans une seule file d'attente, un schéma par tourniquet où l'attente est à l'ancienneté et où, après un service limité à un quantum, les clients sont remis en queue de la file d'attente, et un schéma par recyclage avec des files multiples où, après avoir reçu un service d'un quantum, les clients sont remis en queue dans une nouvelle file, moins prioritaire que la précédente. Dans ce dernier cas, toutes les files peuvent être associées à un même quantum, ou bien avoir des quantum plus grand au fur et à mesure que la priorité baisse (un choix classique est de doubler le quantum à chaque file).

La conception de systèmes interactifs où l'on souhaite favoriser les travaux à la console, à condition qu'ils restent de courte durée, a conduit à étudier les politiques par des simulations ou des mesures. Celles-ci montrent que, comparés à la politique à l'ancienneté, le tourniquet, et encore plus le recyclage multifiles, favorisent les requêtes courtes et défavorisent les requêtes longues, et cela d'autant plus qu'il y a un flux important de requêtes. S'il y a peu de clients en attente, il faut choisir la gestion la plus simple, l'ancienneté. Le tourniquet, simple à réaliser lui aussi, donne déjà de bons résultats et le recyclage multifiles n'est vraiment utile que s'il y a beaucoup de clients en attente. Aujourd'hui on trouve le tourniquet dans Unix et Linux, le recyclage multifiles dans certains Unix.

Tous les processus d'un système ne sont pas interactifs. Certains sont plus prioritaires parce qu'ils réalisent des fonctions systèmes (démons divers) ou des accès à des organes externes (horloge, alarmes). Ils sont alors gérés par des priorités fixes ou variables. Ce type de gestion cohabite souvent avec un niveau de priorité utilisateur où les processus sont gérés par quantum alloué par la méthode du tourniquet.

2.3. Ordonnancement temps réel

Les processus des applications temps réel sont souvent appelés des tâches et sont ordonnancés par priorités. Les tâches y sont périodiques (pour lire régulièrement les valeurs de capteurs ou envoyer des commandes ou un flux de sons ou d'images) ou apériodiques (pour traiter des alarmes). Leur exécution doit respecter des contraintes de temps qui sont exprimées par des délais critiques ou des dates d'exécution au plus tard (échéances). L'ordonnancement temps réel se détermine avec un modèle de tâche qui suppose connus les paramètres suivants :

date de réveil r_i, durée d'exécution maximale C_i, délai critique R_i et période d'exécution P_i. Ces paramètres permettent de calculer des priorités fixes ou variables qui servent à l'ordonnancement.

Ainsi, lorsque les tâches sont périodiques, l'ordonnancement à taux monotone ("rate monotonic") attribue une priorité fixe à chaque tâche et c'est la tâche de plus petite période qui est la plus prioritaire. Lorsque le traitement d'une requête doit être terminée au plus tard avant l'arrivée de la requête suivante (donc le délai critique est égal à la période), on dit que les tâches périodiques sont à échéance sur requête et dans ce cas on peut calculer une condition suffisante d'ordonnançabilité pour un ensemble de n tâches (et ainsi affirmer qu'il n'y aura pas de faute temporelle si les tâches se comportent bien selon les paramètres du modèle).

Cette condition suffisante est : $\sum Ci/Pi \leq n(2^{1/n} -1)$.

L'ordonnancement par échéance ("earliest deadline") attribue à chaque tâche une priorité dépendant de l'échéance ($d_i = P_i.+ R_i$) et c'est la tâche de plus petite échéance qui est la plus prioritaire. Comme les échéances relatives des tâches évoluent avec l'arrivée périodique de nouvelles requêtes, les priorités attribuées sont variables. Quand les tâches sont à échéances sur requête, on dispose de la condition nécessaire et suffisante d'ordonnançabilité : $\sum Ci/Pi \leq 1$.

Pour des tâches à échéances quelconques, cette condition est seulement nécessaire et une condition suffisante est alors : $\sum Ci/Ri \leq 1$.

Les applications multimédia requièrent en plus le contrôle de la gigue de requêtes périodiques. C'est l'écart par rapport à la période entre les dates de réponse de deux requêtes consécutives.

2.4. Inversion de priorité

On a fait l'hypothèse que les requêtes étaient indépendantes et ne demandaient pas d'autre ressource que l'unité centrale. Si des requêtes partagent dynamiquement aussi une ressource exclusive, l'ordonnancement peut être perturbé. On peut ainsi perdre le respect des échéances. Ainsi, avec des priorités fixes, un phénomène, l'inversion de priorité, fait s'exécuter des processus de moindre priorité avant le processus le prioritaire. Par exemple si le processus T2 de faible priorité occupait une ressource critique au moment où elle a été préemptée par le processus T1 de forte priorité et que celui-ci a aussi besoin de la ressource exclusive, au moment où T1 demande cette ressource, il est bloqué (à cause de l'exclusivité) et T2 reprend la main. Si maintenant un autre processus T3 de priorité intermédiaire entre T1 et T2 est déclenché, c'est T3 qui préempte T2 et occupe le processeur, retardant encore plus T1 en s'exécutant avant lui. Il y a inversion de priorité entre T1 et T3.

Le protocole de l'héritage de priorité empêche l'inversion de priorité en changeant la priorité d'un processus qui utilise une ressource critique. Lorsque d'autres processus attendent cette ressource, le processus hérite de la priorité maximale des processus demandeurs, lui inclus. Quand il rend la ressource, il reprend sa priorité initiale. Il existe d'autres protocoles de prévention qui tous changent la priorité (par exemple, le protocole de la priorité plafond).

2.5. Autres ordonnancements de même nature

On retrouve ce type d'ordonnancement pour d'autres ressources qui en général ne doivent pas être réquisitionnées. C'est le cas des files d'attente :
- des travaux dans un système à traitement séquentiel,
- des fichiers à éditer par une imprimante,
- des processus en attente d'allocation dynamique de mémoire (pour des données dynamiques d'un programme, pour des tampons d'entrée-sortie, pour des tampons de message),
- des sémaphores ou autres mécanismes de synchronisation des processus concurrents.

La préemption de l'utilisation de la ressource est impossible. On ajoute parfois un délai maximal d'attente au bout duquel le client est réveillé avec une indication de refus de service.

Si le client en attente a pu garder le processeur (l'unité centrale dans un multiprocesseur, un canal d'entrée-sortie pour un transfert), alors il peut utiliser la ressource sans délai. Sinon, une fois qu'il a reçu la ressource, il doit redemander le processeur, ce qui peut ajouter une attente due à l'ordonnancement de ce processeur. Il est parfois nécessaire de coupler allocation de ressource et allocation de processeur. Cela aboutit, soit à des ordonnancements plus complexes, soit à une structuration de système avec une ordre fixe d'allocation des diverses ressources.

3. Canal d'accès aux disques et ordonnancement des processus

3.1. Organisation de l'espace sur les disques

Les disques sont utilisés comme mémoire secondaire de va et vient des processus, comme support des fichiers et des bases de données, comme stockage intermédiaire de transfert d'images ou de messages dans les stations et les frontaux de communication.

La morphologie des disques contraint fortement les conditions d'accès. Un disque est un plateau magnétique tournant. L'information est stockée sur des pistes concentriques divisées en un nombre fixe de secteurs. L'accès est réalisé par une tête de lecture et d'écriture, qui peut être fixe, mais qui le plus souvent est portée par un bras porte tête, lequel se déplace mécaniquement pour se positionner sur la piste demandée. Il est fréquent que les deux faces du plateau, ou plusieurs plateaux, soient utilisées. Dans ce cas, le bras porte plusieurs têtes qui donnent accès à un ensemble de pistes appelé cylindre. L'adresse d'une requête se présente comme un numéro de cylindre, un numéro de piste et un numéro de secteur. Deux secteurs consécutifs sont séparés par un intervalle qui contient des informations de synchronisation, de repérage et de validation du prochain secteur. Le temps de défilement de cet intervalle permet aussi aux têtes d'accès de commuter d'écriture en lecture ou vice versa. Si le canal d'accès est assez rapide, on peut accéder à une suite de secteurs consécutifs pour n'importe quelle combinaison d'accès, écriture ou lecture. On peut aussi connaître le numéro du prochain secteur qui va défiler devant la tête d'accès. Le formatage du disque définit la taille d'un secteur, (habituellement entre 512 et 8192 octets), et partant, le nombre de secteurs par piste.

Le temps d'accès moyen comprend le temps de déplacement du bras, le délai rotationnel et le temps de transfert. Les optimisations du service jouent sur le déplacement du bras porte tête, sur le délai rotationnel, ou par l'utilisation d'une mémoire cache pour anticiper les accès.

La préemption de l'utilisation du canal d'accès est impossible.

Chiffres pour deux disques durs	EIDE Ultra ATA100	SCSI Ultra 160
Taille	40 Gigaoctets	80 Gigaoctets
Coût au Go	1,90 € (12,50 francs)	16,46 € (108,00 francs)
Vitesse de rotation	7 200 t/mn	15 000 t/mn
Vitesse de transfert : piste interne	555 Mbits/s	700 Mbits/s
piste externe	100 Mbits/s	200 Mbits/s
en moyenne	24 à 41 Moctets/s	51 à 69 Moctets/s
Taille du cache	2 Mo	8 Mo
Temps moyen de déplacement du bras	9,5 ms	3,6 ms
Délai rotationnel moyen	4,16 ms	2 ms
Nombre de cylindres	16383	18479
Nombre de plateaux	2	4
Nombre de têtes	4	8
Densité de stockage (bits/inch max)	540.000	482.000
Densité des pistes (pistes/inch max)	58.000	38.000
Température	60° max	70° max
Mean time between failures - MTBF	600.000 heures de service	1.200.000 heures de service

3.2. Politique de déplacement du bras porte têtes

Le délai de déplacement mécanique du bras est de loin l'élément le plus important du temps de service et il est proportionnel à la distance à parcourir. On peut l'approcher avec une formule linéaire a + b * d où d est le déplacement (> 0), b la vitesse de déplacement et a un délai constant qui est pris pour vaincre l'inertie du bras, au départ comme à l'arrivée, puisque l'accès n'est possible que si le bras est arrêté.

Si les requêtes sont indépendantes entre elles, équiréparties sur toutes les pistes du disque, et si le bras porte tête reste sur la position demandée pour la requête précédente, le déplacement moyen pour une requête isolée vaut le tiers de l'intervalle de déplacement des têtes (si x est la position demandée, y la position précédente, c'est la valeur moyenne de |x - y|, avec x

et y des variables aléatoires équiréparties sur l'intervalle). Les politiques de service permutent les requêtes en attente pour réduire le déplacement total du bras et le temps d'attente moyen.

Le service à l'ancienneté ("FIFO") est le plus simple à mettre en oeuvre, mais le temps d'attente moyen est plus long que dans les autres disciplines. Ce service est utilisé lorsque les files d'attente sont vides ou peu chargées.

La politique du cylindre le plus proche ("shortest seek time first") sert la requête la plus proche de la position de la tête. Le débit est bon, le temps d'attente moyen est plus faible qu'à l'ancienneté, mais la variance est forte. Les requêtes demandant les pistes centrales sont les mieux servies et les requêtes pour des pistes extrêmes peuvent attendre indéfiniment (famine) si la charge est forte.

La politique de l'ascenseur ("scan") limite les changements de direction du bras en parcourant tout l'intervalle dans un sens (tant qu'il y a des requêtes à servir dans ce sens) puis dans l'autre. Le débit est bon, le temps d'attente moyen est plus faible qu'à l'ancienneté, la variance est plus faible qu'avec la politique du cylindre le plus proche, donc les pistes extrêmes sont atteintes plus souvent. Cette politique est utilisée lorsque les files d'attente sont moyennement chargées.

La politique de l'ascenseur à sens unique ("circular scan") ne parcourt les requêtes que dans un sens et en fin d'intervalle revient à la première piste du sens de parcours sans s'arrêter sur les requêtes rencontrées sur son retour. Le disque est traité comme s'il était un tore. Cette politique est utilisée lorsque les files d'attente sont fortement chargées.

3.3. Politique de service rotationnel

Le temps d'accès rotationnel dépend de la distance entre la tête d'accès et le secteur. Pour une requête isolée, le temps d'attente moyen vaut 1/2 tour de disque. Les politiques de service permutent les requêtes en attente pour réduire le temps d'attente moyen.

Le service à l'ancienneté ("FIFO") est le plus simple à mettre en oeuvre, mais le temps d'attente moyen vaut 1/2 tour de disque. Ce service est utilisé lorsque les files d'attente sont vides ou peu chargées.

La politique du plus court délai rotationnel sert en premier la requête la plus proche de la position actuelle de la tête et bien entendu une seule requête par secteur. On a donc une file d'attente par secteur et on sert, secteur après secteur, une requête par secteur. Cette politique n'est intéressante que s'il y a des cylindres avec beaucoup de requêtes en attente.

3.4. Disque serveur de requêtes périodiques liées à des flux multimédia

Les requêtes pour les flux multimédia (son, image video) sont périodiques, avec des contraintes temporelles d'échéances et de gigue. Les politiques de service sont adaptées. Citons quelques unes d'entre elles.

- Avec la politique de l'ascenseur ("scan"), le parcours des cylindres est permanent et on arrête le déplacement dès qu'il y a une requête pour un cylindre rencontré.
- Le parcours permanent peut suivre un ordre prédéterminé des cylindres, selon leur importance.
- Le service peut être un ordonnancement par échéances ("earliest deadline first").
- Si un ensemble de requêtes ont la même échéance, cet ensemble peut être servi par la politique de l'ascenseur ("EDF-SCAN").

EXERCICES

Choix de machines et de systèmes pour du temps réel

Jean possède une machine T avec un processeur ancien et un ordonnancement du processeur selon l'algorithme EDF (Earliest Deadline First) qui s'appuie sur l'échéance et la préemption. Il s'en sert pour gérer une application temps réel qui comprend deux processus A et B. Le processus A (processus de commande d'un moteur) est activé périodiquement et chaque activation demande 270 secondes de calcul et a une échéance de 320 secondes (autrement dit si le processus n'est pas terminé 320 secondes après son activation périodique, on a une faute temporelle); le processus B est déclenché par un manche à balai ("joystick"). Il demande 15 secondes de calcul et son échéance est de 21 secondes (au delà de 21 secondes, c'est la catastrophe).

On lui propose de changer sa machine pour une machine L avec un processeur dix fois plus rapide qui est associé à un ordonnancement à l'ancienneté sans préemption (A s'y exécute en 27 secondes et B en 1,5 seconde). Bien entendu les échéances ne changent pas.

Comparer le fonctionnement des deux machines dans le cas suivant :
- A l'instant 0, le processus A est déclenché,
- A l'instant 1 le processus B est déclenché.

On suppose nul le surcoût système pour la commutation de processus.
Au vu du résultat, Jean doit-il changer de machine ?

Réponse : La machine T (marque Tortue) respecte les échéances. En effet à t = 1, l'algorithme EDF préempte A pour élire B (intervalle 1..16). Puis quand B est fini, A est à nouveau élu (intervalle 16..285).
La machine L (marque Lièvre) ne respecte pas l'échéance de B. En effet il n'y a pas de préemption et A s'exécute pendant l'intervalle 0..27 avant que l'ordonnanceur n'élise B pendant l'intervalle 27..28,5. mais l'échéance de B (c'est 21) est dépassée.
Jean a intérêt à garder sa machine T ou bien à se programmer un ordonnanceur EDF pour L.

Suite de requêtes pour un disque

Soit une file d'attente pour un disque avec des requêtes invoquant les pistes dans l'ordre : 98, 183, 37, 122, 14, 124, 65 et 67
La tête de lecture écriture est initialement sur la piste 53 ;
Calculer le déplacement total de la tête pour les politiques suivantes : ancienneté, cylindre le plus proche, ascenseur, ascenseur sens unique

• Réponses : 640, 236, (299 ou 208), (322 ou 326)