Programmation parallèle en mémoire partagée

Exercices pour le Module 3

INF8601 Systèmes informatiques parallèles

Michel Dagenais

École Polytechnique de Montréal

Département de génie informatique et génie logiciel

Contraintes d'ordonnancement

Expliquez par quel mécanismes possiblement présents sur des architectures modernes, les différentes contraintes $W \rightarrow R, W \rightarrow W, R \rightarrow R$ et $R \rightarrow W$ pourraient être relâchées afin de gagner en performance?

Contraintes d'ordonnancement

Expliquez par quel mécanismes possiblement présents sur des architectures modernes, les différentes contraintes $W \rightarrow R, W \rightarrow W, R \rightarrow R$ et $R \rightarrow W$ pourraient être relâchées afin de gagner en performance?

Les queues d'écriture vont souvent retarder les écritures par rapport aux lectures. Deux blocs de mémoire peuvent opérer en parallèle avec chacun leur queue d'écriture et queue d'invalidation, et causer des réordonnancements écriture écriture ou toute autre combinaison.

Mono versus multi-processeur

En quoi diffèrent les opérations de synchronisation comme les opérations atomiques sur un mono-processeur versus sur un multi-processeur? Lequel est plus coûteux?

Mono versus multi-processeur

En quoi diffèrent les opérations de synchronisation comme les opérations atomiques sur un mono-processeur versus sur un multi-processeur? Lequel est plus coûteux?

Sur mono-processeur, il suffit de se prémunir contre la préemption, ce qui est en général beaucoup plus court. Sur multi-processeur, un mécanisme est requis pour assurer que la nouvelle valeur soit disponible et modifiée par le processeur qui exécute l'opération atomique, sans interaction possible avec les autres processeurs, un peu comme pour les barrières mémoire.

Allocateur par processeur

On veut remplacer l'allocateur de mémoire d'un système d'exploitation multi-processeur par un allocateur qui maintient de la mémoire par processeur et la plupart du temps peut servir les allocations localement sur le processeur. Comment peut-on maintenir des structures "par processeur" en mémoire partagée? Comment changent les besoins en synchronisation dans ce cas?

Allocateur par processeur

On veut remplacer l'allocateur de mémoire d'un système d'exploitation multi-processeur par un allocateur qui maintient de la mémoire par processeur et la plupart du temps peut servir les allocations localement sur le processeur. Comment peut-on maintenir des structures "par processeur" en mémoire partagée? Comment changent les besoins en synchronisation dans ce cas?

On peut avoir des structures par processeur soit en utilisant le numéro de processeur comme indice dans un vecteur, ou en utilisant un registre. Il faut toutefois se prémunir contre la préemption ou la migration du thread courant vers un autre processeur.

Ordre relatif

3.4 Deux processeurs effectuent les opérations suivantes dans un système avec ordonnancement faible. Quels sont les combinaisons de valeurs possibles à la fin?

```
Processeur 1    Processeur 2
     { A == 1; B == 2 }
    A = 3;          x = A;
    B = 4;          y = B;
```

Ordre relatif

3.4 Deux processeurs effectuent les opérations suivantes dans un système avec ordonnancement faible. Quels sont les combinaisons de valeurs possibles à la fin?

```
Processeur 1    Processeur 2
     { A == 1; B == 2 }
    A = 3;          x = A;
    B = 4;          y = B;
```

```
x == 1, y == 2
x == 1, y == 4
x == 3, y == 2
x == 3, y == 4
```

Verrou

Les accès et verrous suivants sont exécutés sur deux processeurs, que peut voir ou ne pas voir un troisième processeur?

```
Processeur 1          Processeur 2
  *A = a;               *E = e;
  LOCK M                LOCK Q
  *B = b;               *F = f;
  *C = c;               *G = g;
  UNLOCK M              UNLOCK Q
  *D = d;               *H = h;
```

Verrou

Les LOCK empêchent les accès qui suivent de remonter avant. Par contre, *A = a peut aller après le LOCK. Le UNLOCK empêche les accès qui précèdent de le dépasser mais *D = d peut remonter avant. Le processeur 3 pourrait donc voir l'affectation de *D avant celle de *A.

Compteur d'octet

On vous demande de programmer un compteur d'octets, pour les statistiques de transmission par réseau, sur un système d'exploitation multi-processeur très performant où chaque processeur peut recevoir et envoyer des paquets. Cette valeur est mise à jour 1 million de fois par seconde mais lue environ une fois par 5 secondes. Comment vous y prendriez-vous?

Compteur d'octet

On vous demande de programmer un compteur d'octets, pour les statistiques de transmission par réseau, sur un système d'exploitation multi-processeur très performant où chaque processeur peut recevoir et envoyer des paquets. Cette valeur est mise à jour 1 million de fois par seconde mais lue environ une fois par 5 secondes. Comment vous y prendriez-vous?

On peut avoir un compteur par processeur et il suffit de sommer les compteurs à chaque 5 secondes pour générer les statistiques.

Structures partagées

Donnez des exemples de structures partagées dans un système d'exploitation multi-processeur?

Structures partagées

Donnez des exemples de structures partagées dans un système d'exploitation multi-processeur?

La liste des tâches, la liste des pages utilisées comme tampon E/S, la liste des périphériques...

Ordre de verrouillage

Deux verrous propres à chaque inode, i.A et i.B, sont requis avant de modifier un inode i dans un système. On vous suggère pour une raison obscure de prendre A avant B pour les inode pairs et B avant A pour les inode impairs. Est-ce que cela peut fonctionner?

Ordre de verrouillage

Deux verrous propres à chaque inode, i.A et i.B, sont requis avant de modifier un inode i dans un système. On vous suggère pour une raison obscure de prendre A avant B pour les inode pairs et B avant A pour les inode impairs. Est-ce que cela peut fonctionner?

Cette manière de procéder n'est pas incorrecte mais confondra les outils de vérification comme LockDep ainsi que les autres programmeurs qui regarderont ce code.

Rendez-vous efficace

Quel est le nombre total de messages requis pour un rendez-vous de n processus avec les méthodes du compteur, en arbre (chaque noeud en crée un autre à chaque niveau comme TBB) et en papillon? Quel est le délai si les messages peuvent circuler en parallèle sur le réseau mais une unité de temps t est requise pour l'envoi ou la réception d'un message sur un ordinateur donné? L'envoi ou la réception de deux messages prend 2t, mais il est possible de recevoir et envoyer un message (non reliés) simultanément en 1t.

Rendez-vous efficace

Quel est le nombre total de messages requis pour un rendez-vous de n processus avec les méthodes du compteur, en arbre (chaque noeud en crée un autre à chaque niveau comme TBB) et en papillon? Quel est le délai si les messages peuvent circuler en parallèle sur le réseau mais une unité de temps t est requise pour l'envoi ou la réception d'un message sur un ordinateur donné? L'envoi ou la réception de deux messages prend 2t, mais il est possible de recevoir et envoyer un message (non reliés) simultanément en 1t.

Avec un compteur, 2(n-1) messages sont envoyés mais le délai est de 2(n-1) aussi. En arbre, chaque noeud envoie un message en première phase puis reçoit un message en seconde phase, sauf la racine, pour 2(n-1) messages et un délai de $2 \log_2 n$. En papillon, chaque noeud envoie un message à chaque étape, $n \log_2 n$. Le délai est de $\log_2 n$.