

MASTER INFORMATIQUE DE L'UNIVERSITÉ BORDEAUX 1
 BASES DE DONNÉES AVANCÉES INF 559 (CHRISTIAN RETORÉ)
 EXERCICES ACCÈS CONCURRENTS (31 DÉCEMBRE 2007)

Le cours a présenté trois techniques de gestion des accès concurrents :

1. énoncé du problème : sérialisabilité, sérialisabilité par permutation, test par construction du graphe de précédence ;
2. verrous simples : transactions cohérentes (actions précédées de verrouillage ; déverrouillage ultérieur), schéma légaux (au plus un verrou sur un élément) ; verrous de plusieurs types (partagés et exclusifs) , modification des notions de cohérence et de légalité en conséquences
3. estampillage (time stamps) simple ou avec copies multiples
4. validation

Un rappel peut-être utile (niveau lycée) : étant données deux suites $a_1 \dots a_n$ et $b_1 \dots b_p$ sans élément commun il y a $C_{n+p}^n = C_{n+p}^p = (n+p)! / n! p!$ suites différentes contenant chaque élément de chaque suite exactement une fois, et préservant l'ordre interne à chaque suite : a_i avant a_j si et seulement si $i < j$ et b_i avant b_j si et seulement si $i < j$. Il est amusant de montrer cela par récurrence ; cela conduit à une formule $f(n, p) = f(n-1, p) + f(n, p-1)$ avec $f(0, p) = f(n, 0) = 1$ et on conclut en utilisant la formule du triangle dit de Pascal : $C_{n-1}^{p-1} + C_{n-1}^p = C_n^p$.

Pour les exercices de réflexion, pas d'exercice type, mais vous pouvez réfléchir aux questions suivantes : quel type de correction garantissent les techniques de gestion des accès concurrents ? comment les garantissent-elles ?

Notation $o_i(E)$: la transaction i effectuée sur l'élément E l'opération x qui peut être :

- r lecture (read)
- w écriture (write)

Exercice A

Dans un système par verrouillage l'opération x ci-dessus peut aussi être :

- l verrouillage (lock)
- u déverrouillage (unlock)

On considère les transactions suivantes :

- $S_1 : r_1(A); A := A + 2; w_1(A); r_1(B); B := B * 3; w_1(B);$
- $S_2 : r_2(B); B := B * 2; w_2(B); r_2(A); A := A + 3; w_2(A);$
- $T_1 : l_1(A); r_1(A); A := A + 2; w_1(A); u_1(A); l_1(B); r_1(B); B := B * 3; w_1(B); u_1(B);$
- $T_2 : l_2(B); r_2(B); B := B * 2; w_2(B); u_2(B); l_2(A); r_2(A); A := A + 3; w_2(A); u_2(A);$
- $U_1 : l_1(A); r_1(A); A := A + 2; w_1(A); l_1(B); r_1(B); B := B * 3; w_1(B); u_1(A); u_1(B);$
- $U_2 : l_2(B); r_2(B); B := B * 2; w_2(B); l_2(A); r_2(A); A := A + 3; w_2(A); u_2(B); u_2(A);$

(A.i) Trouver un schéma d'exécution de S_1 et S_2 (c'est-à-dire T_1 et T_2 sans verrouillage ni déverrouillage) qui soit sérialisable mais pas sérialisable par permutation.

(A.ii) Donner un schéma d'exécution de T_1 et T_2 interdit par les verrous.

(A.iii) Combien y a-t-il de schémas légaux pour T_1 et T_2 ?

(A.iv) Combien y a-t-il de schémas légaux et sérialisables pour T_1 et T_2 ?

(A.v) Combien y a-t-il de schémas légaux et sérialisables par permutation pour T_1 et T_2 ?

(A.vi) T_1 et T_2 sont-elles des transactions à deux phases ?

(A.vii) U_1 et U_2 sont-elles des transactions à deux phases ?

(A.viii) Combien y a-t-il de schémas légaux d'exécution de U_1 et U_2 ? (en ne prenant en considération que les opérations effectives, les r et les w).

Exercice B

Dans un système avec des verrous partagés, l'opération de déverrouillage est inchangée, mais l'opération de verrouillage l ci-dessus est remplacée par deux opérations de verrouillage :

xl verrouillage avec un verrou exclusif

sl verrouillage avec un verrou possiblement partagé

On considère deux schémas d'exécution construits sur trois transactions 1, 2, 3 :

$\mathcal{S}_1 : r_1(A); r_2(B); r_3(C); w_1(B); w_2(C); w_3(D);$

$\mathcal{S}_2 : r_1(A); r_2(B); r_3(C); r_1(B); r_2(C); r_3(D); w_1(C); w_2(D); w_3(E);$

(B.i) Pour chacun des deux schémas :

- Insérer des verrous partagés avant chaque lecture si elle n'est pas suivie d'une écriture du même élément par la même transaction.
- Placer un verrou exclusif en face de toutes les autres écritures ou lectures.
- Placer les déverrouillages à la fin de chaque transaction.

(B.ii) Décrire l'exécution de la suite d'opérations ainsi obtenue.

Exercice C

Notation spécifique pour l'estampillage :

- st_i : la transaction i commence.

On considère les séquences d'opérations suivantes, ou st_i signifie que la transaction i commence :

$\mathcal{S}_0 : st_1; st_2; st_3; st_4; r_1(A); w_1(A); w_4(A); r_3(A); w_3(A); w_2(A); r_2(A); st_5; st_6; st_7; w_5(A); r_2(A); r_7(A); r_6(A)$

$\mathcal{S}_1 : st_1; st_2; st_3; st_4; w_1(A); w_3(A); r_4(A); r_2(A);$

$\mathcal{S}_2 : st_1; st_2; st_3; st_4; w_1(A); w_4(A); r_3(A); w_2(A);$

$\mathcal{S}_3 : st_1; st_2; st_3; st_4; w_1(A); w_2(A); w_3(A); r_2(A); r_4(A);$

(C.i) Décrire l'exécution de chacune des suites d'opérations avec un système d'estampillage simple.

(C.ii) Décrire l'exécution de chacune des suites d'opérations avec un système d'estampillage multiple.

Exercice D

Notation spécifique pour les événements utiles à un système par validation :

- R_iX la transaction i commence, et son ensemble d'éléments lus est X .
- V_i la transaction i essaie de valider.
- W_iX la transaction i finit, et son ensemble d'éléments écrits est X .

On considère les suites suivantes d'événements :

$\mathcal{S}_0 : R_1\{A, B\}; R_2\{B, C\}; V_1; R_3\{C, D\}; V_3; W_1\{A\}; V_2; W_2\{A\}; W_3\{B\};$

$\mathcal{S}_1 : R_1\{A, B\}; R_2\{B, C\}; V_1; R_3\{C, D\}; V_3; W_1\{C\}; V_2; W_2\{A\}; W_3\{D\};$

$\mathcal{S}_2 : R_1\{A, B\}; R_2\{B, C\}; R_3\{C\}; V_1; V_2; V_3; W_1\{A\}; W_2\{B\}; W_3\{C\};$

$\mathcal{S}_3 : R_1\{A, B\}; R_2\{B, C\}; R_3\{C\}; V_1; V_2; V_3; W_1\{C\}; W_2\{B\}; W_3\{A\};$

$\mathcal{S}_4 : R_1\{A, B\}; R_2\{B, C\}; R_3\{C\}; V_1; V_2; V_3; W_1\{A\}; W_2\{C\}; W_3\{B\};$

Décrire dans chacun des cas l'exécution de la séquence d'opérations dans un système par validation.