

Résumé de la séance de questions/réponses avec Monsieur Boukhalfa (ASGBD)

Date : 05/01/2013

Question :

Est-ce que l'ordonnancement suivant est juste ?

R1(A) ----- OK

W1(A)----- OK

R1(A)----- OK (sachant que T1 ne se termine pas ici)

R2(A)----- OK

Réponse :

R2(A) non, car A est toujours verrouillée en exclusif par T1 (après W1(A)).

Question :

R3(B)	S3(B)	OK
R3(C)	S3(C)	OK
R1(B)	S1(B)	OK
W1(A)	X1(A)	OK
W3(B)	X3(B)	NON, T3 attend T1
W1(B)	X1(B)	NON, T1 attend T3

Normalement w3(B) s'exécute puisque on a fait R3(B) au début et donc t3 n'attends pas T1 ?

Réponse :

Non, ce n'est pas parce que R3(B) a été exécuté que W3(B) sera accepté sans condition. S'il y a une autre transaction qui a fait R(B) avec succès alors T3 doit l'attendre (T1 dans ton cas).

Question :

Peut-on utiliser l'algorithme de verrouillage pour montrer qu'un ordonnancement est sérialisable ?

Réponse :

Revoir le théorème vu en cours.

Question :

Dans la reprise après panne, qu'est-ce qu'on doit faire pour une transition qui a été active sans mises à jour lors de la panne ?

Réponse :

Lors de la reprise, si une transaction n'a effectué aucune mise à jour, alors il n'y a rien à faire. Par contre, il faut la relancer à partir du début car les données lus ont été perdus.

Question :

Si on annule une transaction qui a lu ou bien a écrit une val on doit faire une mise à jour des estampilles de ces variables ?

OU (je ne suis pas arrivée à distinguer la question à laquelle correspond cette réponse)

Dans l'estampillage, est-ce qu'à la fin de chaque transaction on fait une mise à jour des estampilles de lecture et d'écriture des variables ?

Réponse :

Si on autorise les mises à jour après le commit, les conséquences sont, pour les transactions où il n'y a pas commit, rien n'est fait car aucune mise à jour n'est effectuée. Si le commit a été exécuté mais les mises à jour ne sont pas matérialisées, il faut refaire la transaction. Donc, dans tous les cas, il n'y a pas d'opération défaire.

Question :

Dans l'estampillage, est-ce qu'on fait des mises à jour quand on annule une transaction ?

Réponse :

Dans l'algorithme d'estampillage, seules les estampilles des variables qui sont mises à jour. Quand une transaction est annulée, son estampille est mise à jour, elle prendra la plus grande +1.

Question :

V2P : lorsqu'il y a terminaison d'une transaction T (et donc libération des variables) , doit-on relancer le V2P après la fin de T que si T bloque d'autres transactions ?

Réponse :

En réalité, une fois la transaction se termine, le scheduler réveille les transactions en attente, et plusieurs transactions peuvent être en conflit pour une variable. Comme l'ordonnancement est séquentiel, d'habitude on termine ordonnancement jusqu'à la fin, ensuite on relance les autres.

Question :

Quand est-ce qu'on dit qu'un ordonnancement est correct et quand est-ce qu'on dit qu'un ordonnancement est sérialisable ? Et quelle est la différence entre les deux ?

Réponse :

C'est simple, sérialisable = correct + correspondance à une exécution série.

Ce sont deux choses différentes. Correct c'est par rapport à son effet sur la BD (les valeurs) par contre la correspondance à une exécution série c'est par rapport à la structure de l'ordonnancement (l'enchaînement des actions dans l'entrelacement). Ce dernier est prouvé en utilisant le graphe de précedence (regardez la correction de l'exo que j'ai mis sur le site web). [Lien](#)

Question :

Lorsque on parle sur les conséquences d'une commande sur les catalogues, on s'intéresse aux problèmes d'incohérence de base de données (contraintes non satisfaisables s'il y en a) ou sur les catalogues eux-mêmes (leurs contenus) ?

Réponse :

Les conséquences sur le contenu des catalogues.

Question :

Comment calculer le nombre d'accès aux informations d'une relation quand l'attribut donné est indexé et quand il n'est pas indexé ?

Réponse :

Quand l'attribut est indexé on suit le b-arbre, donc la profondeur de l'arbre. S'il n'est pas indexé, un accès séquentiel est requis, il faut lire toutes les pages.

Question :

Dans le verrouillage simple, lorsqu'une transaction fait un write elle libère la variable. Est-ce qu'elle a le droit de redemander cette variable en Read ou Write ?

Réponse :

Dans le verrouillage simple, la transaction libère la variable après n'avoir plus besoin d'elle. Donc, ce que tu dis n'est pas possible.