INF4410

Module 6 : Transactions réparties

6.1 Donnez un exemple d'ordonnancement entre deux transactions qui est cohérent sur chaque serveur mais incohérent globalement?

Serveur X, T passe avant U, correct localement:
T: Read(a); Write(a);
U: Read(a); Write(a);

Serveur Y, U passe avant T, correct localement:
U: Read(b); Write(b);
T: Read(b); Write(b);

Globalement il y a un problème car la transaction U pourrait lire la valeur initiale de b, puis la valeur modifiée par T de a pour les combiner en une nouvelle valeur de b et de a. Autrement dit, T précède U qui précède T pour respecter l'ordre sur chaque ordinateur, ce qui est impossible.

6.2 Quelle est l'interaction entre le verrouillage à deux phases, le détecteur d'interbloquage et le coordonnateur pour commettre en deux phases?

Un verrou mortel peut se produire pendant les opérations de verrouillage de données. Le coordonnateur n'entre pas encore en ligne de compte puisque les portions bloquées des transactions ne sont pas prêtes. Le détecteur d'interbloquage peut annuler certaines sous-transactions.

Lorsque tout les participants sont prêts, le coordonnateur fait le tour et demande à chacun s'il est prêt à se commettre. En seconde phase, il confirme que la transaction est commise. Si le coordonnateur fait défaut ou est retardé par un client qui répond lentement, les sous-transactions sur chaque système peuvent rester dans un état incertain pendant un long moment, en maintenant leurs verrous, ce qui est très mauvais pour la concurrence.

6.3 Dans le protocole d'acceptation de transaction à deux phases, le coordonnateur reçoit le vote de chaque participant et communique à tous le résultat. Une variante décentralisée demande plutôt à chaque participant de communiquer son vote à chaque autre participant qui peut ainsi déduire le résultat. Comment se comparent les nombres de messages requis dans chaque cas?

Dans le premier cas, il faut un message de chaque participant au coordonnateur (n - 1, le coordonnateur est lui-même un participant), et ensuite un message du coordonnateur à chaque participant (possiblement en multi-diffusion). Dans le second cas, la variante décentralisée, il faut un message de chaque participant à

chaque autre, soit n messages en multi-diffusion ou n(n - 1) messages simples, ce qui serait très coûteux si le nombre de participants est grand.

6.4 Un interblocage fantôme est un faux positif détecté par le détecteur global d'interblocages répartis. Comment cela peut-il se produire?

Le problème d'obtenir efficacement l'état global d'un système réparti a été discuté. Pour avoir un état global exact, il faudrait par exemple arrêter tous les processus le temps de prendre un cliché. A la place, le détecteur d'interblocages répartis fait l'union des graphes de dépendances pris en rapide succession des processus impliqués. Si les verrous peuvent être pris ou relâchés sans contrainte, les faux positifs serait plus probables. Cependant, un verrouillage en deux phases est normalement utilisé et les verrous peuvent seulement s'ajouter jusqu'à la fin de la transaction. Il subsiste le cas d'une transaction qui détient des verrous qui causeraient un interblocage mais qui est annulée dans le milieu de la séquence de lecture en rapide succession des graphes de dépendance des processus impliqués.

Il peut bien sûr arriver qu'un verrou se soit ajouté pendant que les graphes sont lus. Dans ce cas, un interblocage qui vient de se former pourrait ne pas être détecté. Ceci est sans conséquence puisqu'un interblocage demeure indéfiniment et sera nécessairement vu à la prochaine ronde de lecture des graphes de dépendance.

6.5 Si le coordonnateur d'une transaction tombe en panne après avoir enregistré sa liste d'intentions mais avant d'avoir écrit la liste des participants dans le journal et envoyé ses demandes de préparation à commettre aux participants, décrivez comment il pourra récupérer? Serait-il préférable d'écrire la liste des participants avant la liste d'intention?

Si le coordonnateur n'a pas encore envoyé ses demandes de préparation à commettre (vote d'acceptation de la transaction), la transaction peut être simplement oubliée au redémarrage. S'il n'a pas la liste des participants ou s'il n'a pas la liste des données, il ne peut rien faire. L'un n'est donc pas mieux que l'autre à sauver en premier. A la rigueur, la liste des participants pourrait être reconstruite si tous les participants se manifestent pour vérifier l'état de la transaction amorcée.