Manipulations Multibases et Distribuées : Partie 3

Witold Litwin

Exercices

1. Voir tous ceux déjà dans les diapos du cours.

2. Soit la requête MBD du cours :

```
Use B0
Select R.A, X.B
From R, B1.R1 X
Where
R.C = V1 and X.D = V2 and R.E = X.E
```

On rappelle que c'est une requête distribue donc les deux bases B0 et B1 sont sur deux nœuds (SMBDs) distincts. On suppose que la taille de la table R après la restriction est 100 tuples. Celle de X après la restriction est de 1000 octets. Enfin, l'attribut E de jointure est la clé de R1. Est-ce que le plan d'exécution typique est le meilleur en ce qui concerne le coût de transfert par le réseau, en nombre d'octets transférés ? Est-il notamment préférable à celui basé sur une semi-jointure ? Motivez votre point de vue.

3. Supposez que la base **bnp** est sous MySQL et que **cic** est sous MsAccess. L'usager de **bnp** veut connaître le nom et la rue de chacune de 5 succursales de CIC à Paris, suivant 10 autres qui ne l'intéressent pas. Ceci, par rapport à l'ordre ascendant sur le code postal (donc l'arrondissement). Sous MySQL il utiliserait pour cela simplement le mot-clé LIMIT 5,10 dans son expression de sélection multibase et la clause OrderBy sur le code. MsAccess ne connaît cependant pas ce mot clé. Il n'a que le mot-clé TOP (comme pour SQL Server), où TOP X limite la sélection à X+*n*-1 tuples par rapport à l'ordre de sélection dans la clause Order By, où $n \geq 1$ sont les tuples ex-æquo par rapport à l'ordre sur les positions X…X+*n*-1. Montrez comment le SGMB, MySQL ici, pourrait permettre d'exprimer la requête originelle et pourrait peut-être générer la sous-requête envoyée vers MsAccess. Proposez le plan d'exécution qui vous semble optimal.

4. Il est bien connu que les equi-jointures internes avec les attributs de jointure sur un même domaine sont associatives : (A JOIN B) JOIN C = A JOIN (B JOIN C). Est-ce que cette propriété reste vraie, selon vous, quand les valeurs des attributs de jointures ont des précisions différentes ? Démontrez votre proposition.

5. Est-ce que la requête suivante est une transaction ACID ?

```
Use Banks vital cic
Update cl*
set  street = 'Charles de Gaulle"
where street = 'Etoile‘
```

6. Proposez une requête multibase où le modèle ACID convient et une où ce modèle ne convient pas.

7. Pourquoi 2PL strict est davantage usité que 2PL générique, où les verrous pourraient être lâchés avant la commande de commitment ?

8. Soit T1 et T2 deux transactions. On note Ri (B.D) la lecture d'une donnée D de base B par la transaction Ti et Wi(B.D) signifie l'écriture correspondante. Le début d'une transaction n'est pas noté explicitement. Si tout va bien, Ti se termine par Ci (Commit). On ne note pas explicitement d'avortement d'une transaction. Complétez et commentez les exécutions qui suivent, en supposant que les deux bases concernées sont sur un même SGMB. Les deux transactions sont comme suit :

T1 = R1(B1.D1), R1(B2.D2), W1(B2.D2 = D2 – D1), C1;
T2 = R2(B2.D2), R2(B1.D1), W2(B1.D1 = D2 + D1), C2;

a. Il n'y a aucun contrôle de concurrence. Quelles sont les exécutions sérialisables et pourquoi ?

1. R1(B1.D1), R1 (B2.D2), W1(B2.D2 = D2 – D1), C1, R2(B2.D2), R2(B1.D1), W2(B1.D1 = D2 + D1), C2;

2. R1(B1.D1), R1 (B2.D2), W1(B2.D2 = D2 – D1), R2(B2.D2), R2(B1.D1), C1, W2(B1.D1 = D2 + D1), C2;

3. R1(B1.D1), R1 (B2.D2), R2(B2.D2), W1(B2.D2 = D2 – D1), R2(B1.D1), C1, W2(B1.D1 = D2 + D1), C2;

b. On contrôle par 2PL strict. Après chaque Ri le gestionnaire des transactions voit s'il peut verrouiller (exclusivement) la donnée. Auquel cas l'exécution de Ti se poursuit. Il peut alternativement mettre Ti en attente, par l'opération WAi (Wait Ti). Enfin, par l'opération Ui, il déverrouille tous les verrous posé par Ti. Complétez par WA, U, et, après …, par la suite des opérations qui ont pu s'exécuter de T1 et T2 pour les ordonnancements commençant comme suit. Commentez vos propositions.

1. R1(B1.D1), R1 (B2.D2), W1(B2.D2 = D2 – D1), C1, R2(B2.D2), R2(B1.D1), W2(B1.D1 = D2 + D1), C2;

2. R1(B1.D1), R1 (B2.D2), W1(B2.D2 = D2 – D1), R2(B2.D2), C1,… ;

3. R1(B1.D1), R2(B2.D2), R1 (B2.D2), R2(B1.D1),…;

c. On contrôle toujours par 2PL strict. Mais, cette fois-ci, après chaque Ri le gestionnaire des transactions voit s'il peut poser un verrou partagé sur la donnée, autorisant une lecture par une autre transaction. Auquel cas l'exécution de Ti se poursuit. Il issue autrement le WAi. Chaque Wi donne lieu à un verrou exclusif ou une attente. Comme auparavant, complétez par WA, U etc. l'exécution de T1 et de T2 comme suit. Commentez vos propositions.

1. T1 = R(B1.D1), W(B1.D1) ; T2 = R(B1.D1), W(B1.D1) ;

Le début de l'exécution : R1(B1.D1), R2(B1.D1), W2(B1.D1), W1(B1.D1)…

2. Proposez trois exécutions concurrentes de T1 et de T2 du point (a). On dit qu'un protocole 1 est davantage concurrentiel pour certaines transactions s'il offre davantage d'exécutions concurrentes correctes. Es-ce le cas pour les verrous partagés ici par rapport à ceux exclusifs seulement ?

9. Supposez que les deux bases B1 et B2 ci-dessus sont distribuées, c. à d. sur deux nœuds (SGMBs) distincts. On utilise 2PC, en lançant T1 à partir de la base B0 sur son nœud propre également. La notation Tj désigne la sous-transaction sur la base j de T1. L'ordre du vote pour la base Bj est noté RCj. La réponse est Yj (Yes) or Nj (No). Le commit final est noté Cj. En supposant le succès de la transaction, montrez les opérations issues et reçus par T1 sur B0, et celles constituant d'une même manière les sous-transactions sur B1 et B2.

10. En quoi consiste le blocage d'un participant du 2PC et quand il peut arriver ?

11. A quoi sert un protocole de terminaison ajouté au 2PC standard ?

12. Proposez un protocole de terminaison qui permettrait à chaque participant de terminer 2PC si tous les autres (N-1) participants l'ont terminé. Supposez que (i) le réseau fonctionne normalement pour les messages entre les participants, que (ii) tous les participants fonctionnent et que (iii) le coût du protocole est de 2(N-1) messages

13. Prouvez qu'il n'y a pas de protocole de terminaison qui éviterait à coup sûr le blocage d'un participant.

14.Prouvez que le protocole basique de dates de valeur est libre de verrou mortel.

15.Les deux transactions T1 et T2 du point 8.a ci-dessus, s'exécutent maintenant avec les dates de valeurs, V1 = 100 et V2 = 200 et le protocole basique du cours. Analysez les exécutions du point 8. Comment elles auraient changé et pourraient être complétées ? Identifiez chaque transaction par sa date de valeur. Notez Av l'opération de suppression (abort) de la transaction Tv.

16. Le meta-voyagiste E-Rêves veut utiliser les dates de valeur pour ses transactions de réservation de billets d'avions, venant de voyagiste différents. Proposez une solution réaliste pour une transaction typique montrée dans le cours pour EDreams. Supposez que les dates de l'utilisation de 2PC

17.A quoi sert ODBC ?

18. Le Select Into multibase est à l'heure actuelle réalisé par ODBC par les Fetch d'un tuple à la fois. Chaque Fetch coûte 1ms au moins sur un réseau local. Estimez le gain de temps pour votre stratégie optimisée de la requête de l'exercice (2) ci-dessus. Notez que vous simulez la fonction Estimate de MSQL.

19. Est-ce que un des articles dans le matériel de support du cours pourrait vous être utile dans le cas général de l'optimisation correspondante à celle de la question (18) ? Motivez votre réponse en quelques phrases.

20.L'utilité de fonctions API par rapport aux celles dites SQL de ODBC.

21.Comment la DCE gère-t-elle la base de temps globale ?