INF4410

Module 6 : Transactions

6.1 Un serveur gère plusieurs valeurs a1, ...an. Ce serveur offre deux opérations à ses clients: value = Read(i), Write(i, value). Les transactions T et U sont définies comme suit. Donnez deux sérialisations équivalentes de ces transactions?

T: x=Read(j); y=Read(i); Write(j,44); Write(i,33);

U: x=Read(k); Write(i,55); y=Read(j); Write(k,66);

6.2 Montrez pourquoi lorsqu'une transaction a relâché un verrou elle ne doit plus en obtenir afin de permettre l'équivalence avec la sérialisation?

6.3 Soit deux transactions T et U, lesquelles de ces sérialisations sont valides? Sont possibles avec un verrouillage en deux phases?

T: x = read(i); write(j,44);
U: write(i,55); write(j,66);

a) T x=Read(i); U write(i,55); T write(j,44); U write(j,66);

b) U write(i,55); U write(j,66); T x=Read(i); T write(j,44);

c) T x=Read(i); T write(j,44); U write(i,55); U write(j,66);

d) U write(i,55); T x=Read(i); U write(j,66); T write(j,44);

6.4 Expliquez pourquoi il ne faut pas relâcher les verrous avant de se commettre, même après la fin des opérations.

6.5 Une synchronisation optimiste est appliquée aux transactions T et U qui sont actives en même temps. Discutez ce qui arrive dans chacun des cas suivants:

T: x=Read(i); Write(j,44);
U: Write(i,55); Write(j,66);

a) T est prête en premier et la validation en reculant est utilisée? b) En avançant?

c) U est prête en premier et la validation en reculant est utilisée? d) En avançant?

6.6 Deux transactions T et U utilisent des compteurs de temps pour vérifier leur cohérence. Initialement ai = 10 au temps t0, et aj = 20 au temps t0.

Voici deux séquences possibles à vérifier pour les opérations de T et U:

T: Début;
U: Début;
U: Write(i,55);
U: Write(j,66);
T: x=Read(i);
T: Write(j,44);
U: Compléter;

T: Début;
U: Début;
U: Write(i,55);
U: Write(j,66);
U: Compléter;
T: x=Read(i);
T: Write(j,44);

6.7 Expliquez comment une transaction imbriquée s'assure que toutes les sous-transactions sont correctement commises ou annulées.

6.8 Les transactions T, U, et V sont définies comme suit. Décrivez le journal produit par cette application si un verrouillage en deux phases est utilisé et U acquiert i et j avant T? Décrivez comment ce journal peut être utilisé pour récupérer en cas de panne?

T: x=Read(i); Write(j,44);
U: Write(i,55); Write(j,66);
V: Write(k,77); Write(k,88);

6.9 Les transactions T et U utilisent un contrôle optimiste de la concurrence basé sur les compteurs de temps. Décrivez l'information écrite dans le journal si T débute en premier.

T x=Read(i);
U Write(i,55);
U Write(j,66);
T Write(j,44);
U Commit;
T Commit;

6.10 Les transactions T et U utilisent un contrôle optimiste de la concurrence basé sur la validation en reculant. Décrivez l'information écrite dans le journal si T débute en premier.

```
T x=Read(i);
U Write(i,55);
U Write(j,66);
T Write(j,44);
U Commit;
T Commit;
```