

Module 7 : Réplication

- 7.1 Trois ordinateurs offrent un service. Chaque ordinateur fait en moyenne une panne par 5 jours et cette panne prend 4 heures à réparer. Quelle est la disponibilité de ce système?

La probabilité de ne pas être fonctionnel est de $4/(5*24+4)=0.03$. La probabilité d'avoir les trois ordinateurs simultanément en panne est de 0.03^3 , pour une disponibilité de 0.999973.

- 7.2 Un jeu en réseau conserve une copie de son monde virtuel sur chaque client. Un serveur gère les communications de groupe. Comment devrait se faire la mise à jour pour les envois de projectiles qui sont lancés, frappent les opposants et les retiennent quelque temps? Pour le ramassage d'objets utiles?

La mise à jour des coordonnées des joueurs et du projectile devrait se faire de manière causale. Si A voit B et lance un projectile examinons quelques cas possibles:

Les messages de B vers A tardent. Lorsque A voit le projectile s'approcher de B, B est déjà loin. Lorsque le projectile devrait atteindre B, A doit contacter B, la réponse que B n'est pas touché arrivera après les mises à jour montrant que B est très loin. Si B prend trop de temps à répondre, le jeu pourrait être arrêté avec l'image du projectile qui arrive juste avant B.

Les messages de A vers B tardent. A voit très bien la position à jour de B et lance son projectile qui s'approche jusqu'à venir devant. B voit A arriver et lancer son projectile très en retard, il veut alors déplacer son personnage. Le message de projectile arrive alors. Si B a déjà bougé, A verra son projectile figé en attente de la réponse, verra B bouger et finalement verra le projectile manquer sa cible.

Lorsque A et B voient un objet à ramasser, il n'y a pas directement d'interaction entre les deux. Néanmoins, le premier arrivé doit pouvoir ramasser l'objet. Un ordonnancement total est donc approprié.

- 7.3 Expliquez la différence entre linéarisation et cohérence séquentielle?

Avec la linéarisation, les opérations sur chaque réplicat sont effectuées exactement dans le même ordre qui correspond à l'ordre réel. Ceci requiert un serveur central par où tout passe.

Avec la cohérence séquentielle, les opérations de chaque client arrivent dans le même ordre sur chaque réplicat. L'ordre relatif de ce qui est fait d'un client à l'autre peut varier entre les réplicats.

- 7.4 Pourquoi un client qui lit des valeurs d'un serveur secondaire obtient-il une cohérence séquentielle plutôt qu'une linéarisation?

Si un client lit une valeur d'un serveur secondaire (e.g. NIS), il obtiendra des valeurs qui sont mises à jour dans le même ordre que s'il parlait au serveur principal. Cependant, il pourrait être en retard par rapport à un client connecté au serveur principal.

- 7.5 Avec CODA, pourquoi les utilisateurs doivent-ils parfois intervenir manuellement?

Le contrôle de concurrence est optimiste sur CODA et les opérations ne peuvent être annulées. Ainsi, si deux utilisateurs ont mis à jour en même temps leur copie d'un fichier, le second à transmettre la mise à jour au serveur doit décider comment réconcilier les deux versions, plutôt que d'écraser le travail de l'autre.

- 7.6 Les items a et b sont répliqués sur Ax, Ay, Bm, et Bn. Les transactions T et U sont définies ainsi:

T: Read(a); Write(b,44);
U: Read(b); Write(a,55);

Montrez un ordonnancement de T et U, dans le contexte de verrous en deux phases pour les réplicats. Expliquez pourquoi les verrous ne suffisent pas à assurer la sérialisation?

T: x=Read(Ax); lock Ax;
T: Write(Bm,44); lock Bm;
U: x=Read(Bm); Attendre le verrou Bm;
T: Write(Bn,44); lock Bn;
T: Compléter; unlock Ax, Bm, Bn;
U: x=Read(Bm); lock Bm;
U: Write(Ax,55); lock Ax;
U: Write(Ay,55); lock Ay;

Supposons que Bm tombe en panne avant d'être verrouillé par U. U pourra procéder. La validation locale permet de vérifier si une copie qui n'a pas fonctionné n'a pas été recouverte. Dans le cas de T, lorsque Bm est revenu et a déjà été lu, il découvre que le verrou détenu ne tient plus et la transaction doit être annulée.