Examen 2022

Protocole de cohérence causale

10. Vivacité faible de Lamport

Si un événement e précède causalement un événement e′, alors HL(e)<HL(e′) où HL(x) est l'horloge locale associée à l'événement x.
Sa contraposée serait :
Si HL(e) ≥ HL(e′), alors e ne précède pas causalement e′.
Cela signifie que si l'horloge locale d'un événement e est plus grande ou égale à celle d'un événement e′, alors e ne peut pas être causé par e′.

11. Protocole de cohérence causale léger

a- *Mise à jour précédant causalement sa dernière écriture.*

Le site doit répercuter cette mise à jour sur sa copie locale, car la mise à jour reçue est valide et appartient à un ordre causal antérieur. Ignorer la mise à jour irait à l'encontre de la cohérence causale, car cela briserait l'ordre causale des opérations.
→ Justification : Si la mise à jour précède causalement la dernière écriture locale, elle doit être appliquée avant cette écriture, respectant ainsi l'ordre causal.

b- *Mise à jour causalement indépendante de la dernière écriture*

Si un site reçoit une mise à jour indépendante causalement de sa dernière écriture, il doit aussi répercuter cette mise à jour sur sa copie locale. Dans ce cas, il n'y a pas de lien causal entre la mise à jour reçue et les opérations locales précédentes, donc la mise à jour peut être appliquée sans conflit.
→ Justification : Comme les opérations sont indépendantes, il n'y a pas de problème à appliquer la mise à jour, car cela ne change pas l'ordre causal de la séquence.

c- *Mise à jour succédant causalement à la dernière écriture*

Si un site reçoit une mise à jour succédant causalement à sa dernière écriture sur la copie locale de X, cette mise à jour doit également être répercutée sur la copie locale. En effet, la mise à jour succédant causalement à une opération locale représente une évolution logique et doit être appliquée pour respecter l'ordre des événements.
→ Justification : Le site doit appliquer cette mise à jour, car elle représente une évolution de l'état et doit suivre l'ordre causal, où chaque opération se produit dans un ordre précis.

d- *Messages échangés et algorithmes*

X.Reads()

1. Le site s envoie une demande de lecture à tous les autres sites pour récupérer la dernière valeur de X.
2. Chaque site envoie sa valeur actuelle de X, avec l'horloge associée.
3. Le site s reçoit les valeurs et les horloges des autres sites et choisit la valeur la plus récente, respectant l'ordre causal.

```
1. émettre "demande de lecture" à tous les sites.
2. pour chaque site \( i \) :
    a. recevoir la valeur \( X_i \) et son horloge \( HL_i \).
3. choisir la valeur \( X_{max} \) associée à l'horloge la plus récente.
4. retourner \( X_{max} \).
```

X.Writes(v)

1. Le site sss écrit la valeur vvv dans sa copie locale de X.
2. Ensuite, il envoie cette valeur à tous les autres sites avec l'horloge associée à cette écriture.
3. Chaque site reçoit la mise à jour et l'applique si l'ordre causal le permet.

```
1. écrire la valeur \( v \) dans \( X_s \) sur le site \( s \).
2. émettre "mise à jour \( v \)" avec l'horloge \( HL_s \) à tous les sites.
3. pour chaque site \( i \) :
    a. recevoir la mise à jour et appliquer si possible.
```

Traitement de réception des messages

Lorsqu'un site reçoit un message de mise à jour, il doit appliquer la mise à jour si l'ordre causal le permet.

```
1. si la mise à jour reçue est valide selon l'ordre causal :
    a. appliquer la mise à jour à \( X \).
    b. mettre à jour l'horloge locale.
```

12. Non-bloquant

Les opérations X.Reads() et X.Writes(v) sont non-bloquantes, car elles ne nécessitent pas de synchronisation explicite entre les sites. Chaque site peut exécuter ses opérations indépendamment, et la cohérence causale est maintenue par l'échange de messages asynchrones.

→ Explication :

- Lorsqu'un site exécute X.Reads(), il envoie des demandes à tous les autres sites et reçoit des réponses de manière asynchrone.
- Lorsqu'un site exécute X.Writes(v), il envoie la mise à jour de manière asynchrone à tous les autres sites, et l'application de la mise à jour peut se faire sans attendre de réponse immédiate.

13. Protocole léger vs protocole classique

a- *Gain du protocole léger*

Le protocole léger présente plusieurs avantages par rapport au protocole classique, notamment :

- Moins de messages échangés : Au lieu de diffuser chaque mise à jour à tous les sites et de gérer une diffusion causale complète, le protocole léger envoie simplement des messages de mise à jour et gère l'ordre causal de manière plus directe et plus simple.
- Plus de parallélisme : Les opérations X.Reads() et X.Writes(v) peuvent être exécutées indépendamment sur chaque site sans nécessiter de synchronisation complexe.
- Moins de données échangées : Les messages échangés dans le protocole léger contiennent juste la valeur de X et l'horloge associée, ce qui réduit le volume des données comparé à une diffusion causale qui pourrait impliquer des informations plus détaillées.

b- *Désavantages du protocole léger*

Les désavantages du protocole léger par rapport au protocole classique sont :

- Moins de robustesse : Le protocole classique avec une diffusion causale garantit que toutes les mises à jour sont appliquées dans le bon ordre causal, ce qui peut être plus robuste pour des environnements complexes.
- Plus complexe à gérer : Bien que plus léger en termes de communication, le protocole léger peut être plus difficile à implémenter, car il repose sur des règles spécifiques d'ordre causal qui peuvent nécessiter une gestion subtile des horloges et des messages.