Examen 2020

Consensus avec détecteur de défaillance ◊Ω

Consensus avec Ω

1- Vérification de l'accord :

→ Chaque processus Pi envoie une requête au leader, attend une proposition et décide de la valeur proposée.
→ Le problème potentiel réside dans le fait que les détecteurs de défaillance peuvent ne pas stabiliser immédiatement, ce qui pourrait entraîner des incohérences pendant la phase de sélection du leader, notamment si un leader est perçu différemment par différents processus. Par exemple, si un processus pense qu'un leader est défaillant tandis qu'un autre processus ne le fait pas, cela pourrait entraîner la sélection de différents leaders à différents moments.
Conclusion : L'algorithme ne garantit pas l'accord car il pourrait y avoir des processus qui choisissent des leaders différents avant que le détecteur de défaillance ne stabilise.

2- Vérification de la validité :

→ Chaque processus Pi envoie sa propre valeur vi comme proposition lorsqu'il répond à une requête de leader.
→ La valeur que chaque processus décide (après avoir reçu une proposition de leader) provient effectivement de l'une des propositions des processus et, donc, est valide. Cependant, comme chaque processus décide après avoir reçu une proposition, et qu'il n'y a pas de mécanisme garantissant qu'un processus ne choisit pas une valeur incorrecte à cause des défaillances du détecteur, la validité pourrait être compromise.
Conclusion : L'algorithme ne garantit pas totalement la validité car la sélection de la valeur pourrait être incorrecte si le détecteur de défaillance ne stabilise pas ou s'il y a des défaillances lors de la communication.

Consensus avec ◊Ω

3- Vérification de l'accord :

→ Chaque processus Pi envoie une requête au leader et attend une proposition. Ensuite, il envoie une requête d'unicité à tous les autres processus.
→ Un processus reçoit des réponses d'unicité (soit "Ack", soit "Nack"). Si plus de 3N/4 réponses sont des "Ack", alors le processus mémorise la valeur et décide de cette valeur.
→ Le critère "plus de 3N/4 Ack" assure qu'une majorité de processus confirment que le leader est unique.
Exécution sans défaillance :
→ Si tous les processus sont corrects et qu'il n'y a pas de défaillances, l'algorithme garantit qu'une majorité de processus confirmeront l'unicité du leader, ce qui permet d'assurer l'accord : tous les processus vont décider de la même valeur.
Conclusion : Oui, cet algorithme garantit l'accord dans le cas d'une exécution sans défaillance.

4- Vérification de la validité :

→ Chaque processus mémorise la valeur proposée par le leader après avoir reçu une majorité de "Ack".
→ La valeur v mémorisée par un processus doit donc être la proposition d'un leader correct, car un leader est un processus correct, et seul un leader correct peut obtenir 3N/4 "Ack" d'autres processus corrects.
Conclusion : Oui, l'algorithme garantit la validité, car la valeur choisie est celle d'un leader correct.

5- Vérification de la terminaison :

→ Si le détecteur de défaillance fonctionne correctement, tous les processus finiront par recevoir une proposition de leader et confirmeront l'unicité de ce leader via les messages d'unicité ("Ack" ou "Nack").
→ Cependant, si un ou plusieurs processus défaillants empêchent la propagation des messages "Ack", l'algorithme pourrait ne pas réussir à obtenir 3N/4 réponses positives et donc ne pas terminer.
Conclusion : L'algorithme garantit la terminaison si et seulement si tous les processus non défaillants peuvent communiquer et qu'il existe un leader correct. Si des défaillances surviennent, certains processus peuvent ne jamais obtenir assez de "Ack" pour terminer.