

TP Transactions - Morgane Cadeau SN 2App

3 - Exemples et scénarios d'exécution

TMPP : politique de contrôle de concurrence pessimiste sans blocage avec propagation directe

TMPC : politique de contrôle de concurrence par certification avec propagation différée

TM2PL : protocole de verrouillage à deux phases avec propagation directe

a. Quelles seront les transactions validées, bloquées ou abandonnées ?

TMPP : Les transactions abandonnées seront T1 et T2, la transaction validée serait T3.

TMPC : La transaction abandonnée sera T3, les transactions validées seront T1 et T2.

TM2PL : Les transactions validées seront T1, T2 et T3. T2 sera bloqué par T3 car il essaiera d'écrire dans z alors que T3 l'a lu juste avant. T1 sera bloqué par T2 car T2 écrit une valeur dans y donc T1 est bloqué quand il essaye de lire la valeur de y.

b. Quel sera l'ordre série équivalent obtenu ?

TMPP : ordre série 6-10-15-16

TMPC : ordre série 5-8-11-12-4-7-9-13-14

TM2PL : ordre série 4-6-10-15-16-7-9-13-14-5-8-11-12

c. Quelles seront les valeurs de variables affichées par la commande list ?

TMPP : x = 3; y = 0; z = 0; ft1 = 0; ft2 = 0; ft3 = 1

TMPC : x = 0; y = 1; z = 2; ft1 = 1; ft2 = 1; ft3 = 0

TM2PL : x = 3; y = 1; z = 2; ft1 = 1; ft2 = 1; ft3 = 1

d. Vérifiez vos réponses à l'aide du shell

Mes réponses semblent correctes d'après le shell.

Mêmes questions avec le deuxième scénario :

a. Quelles seront les transactions validées, bloquées ou abandonnées ?

TMPP : Les transactions abandonnées seront T2 et T3, la transaction validée serait T1.

TMPC : La transaction abandonnée sera T3, les transactions validées seront T1 et T2.

TM2PL : Les transactions T1, T2 et T3 seront bloquées car chacune veut accéder à une valeur lue ou écrite par une autre transaction. Il y a donc interblocage.

b. Quel sera l'ordre série équivalent obtenu ?

TMPP : ordre série 5-8-11-12

TMPC : ordre série 5-8-11-12-4-7-9-13-14

TM2PL : ordre série nul car rien n'est validé

c. Quelles seront les valeurs de variables affichées par la commande list ?

TMPP : x = 0; y = 0; z = 0; ft1 = 1; ft2 = 0; ft3 = 0

TMPC : x = 0; y = 1; z = 2; ft1 = 1; ft2 = 1; ft3 = 0

TM2PL : x = 0; y = 1; z = 0; ft1 = 0; ft2 = 0; ft3 = 0

d. Vérifiez vos réponses à l'aide du shell

Mes réponses semblent correctes d'après le shell.

3. Politique d'accès sans contrôle de concurrence

Si T1 écrit une valeur dans x, T2 lit la valeur de x puis T1 annule, on a T2 qui aura lu une mauvaise valeur.

4 - Évaluation de protocoles

1 - Commentez les scénarios fournis

Scénario 1 (TM2PL) → Les trois transactions sont validées mais il y a beaucoup de temps d'attente et le temps total se rapproche plus du temps séquentiel que du temps optimal. En revanche, il n'y a pas de temps perdu, ce qui est une bonne chose. Il y a du temps d'attente à cause des verrous posés sur les variables.

Scénario 2 (TMPC avec T) → Les transactions T1 et T2 sont validées tandis que la transaction T3 a été annulée. Le temps optimal est atteint mais il y a tout de même 33 unités de temps perdues à cause de T3 qui a été annulée.

Scénario 3 (TMPC avec S) → Toutes les transactions sont validées. Le temps total n'est pas optimal mais est tout de même meilleur que TM2PL.

Dans ce cas là, le scénario 3 avec TMPC et les super transactions est le plus performant.

2 - Scénarios favorables à chaque politique fournie

TM2PL : Cette politique est plus efficace quand il y a peu d'accès concurrents à la mémoire transactionnelle, car un verrou est posé à chaque accès à une variable, que ce soit en lecture ou écriture, jusqu'à ce que la transaction soit validée.

TMPC : Cette politique est plus efficace lorsque les transactions sont sérialisables puisque c'est une politique optimiste donc la sérialisabilité n'est testée qu'à chaque commit.

TMPP : Cette politique est plus performante quand une transaction n'est plus sérialisable avant sa validation car TMPP est pessimiste et vérifie la sérialisabilité des transactions à chaque exécution.