

1 Striktes und Rigoroses 2PL

Welchen Vorteil bietet rigoroses Zwei-Phasen-Sperren? Wie vergleicht es sich mit anderen Formen des Zwei-Phasen-Sperrens?

2 Deadlock Detection

Angenommen, die Transaktionen $T_1 - T_6$ laufen gleichzeitig auf einer Datenbank. Die folgende Tabelle zeigt, welche Datenobjekte a, \dots, j derzeit von den Transaktionen gesperrt sind und auf welche Freigaben von Sperren die Transaktionen warten.

Transaktion	hält Sperre auf	Wartet auf das Freigeben der Sperre auf
T_1	a, b	d, i
T_2	c	a
T_3	d, e	h, j
T_4	f, g	e, i
T_5	h	c
T_6	i, j	

Bitte erstellen Sie den Wait-For Graph und entscheiden Sie, ob ein Deadlock vorliegt. Im Falle eines Deadlocks, lösen Sie ihn bitte auf.

3 Recovery

Drei Transaktionen, T_1, T_2, T_3 , liefen gleichzeitig gemäß der folgenden Tabelle. Die Tabelle zeigt auch die geschriebenen Log-Dateien.

	T_1	T_2	T_3	Log
				[TID, DID, old, new]
1.	BOT			[T_1 start]
2.	$r(A, a_1)$			
3.	$a_1 = a_1 * 1.03$			
4.		BOT		[T_2 start]
5.	$w(A, a_1)$			[$T_1, A, 1, 1.03$]
				point 1
6.		$r(A, a_2)$		
7.			BOT	[T_3 start]
8.		$a_2 = a_2 + 10$		
9.	commit			[T_1 commit]
10.			$r(B, b_1)$	
12.		$w(A, a_2)$		[$T_2, A, 1.03, 10.03$]
13.		$r(C, c_1)$		
14.		$r(B, b_2)$		
15.		$c_1 = c_1 + b_2$		
16.		$w(c_1)$		[$T_2, C, 10, 13$]
17.			$b_1 = b_1 - 5$	
18.			$w(B, b_1)$	[$T_3, B, 3, -2$]
19.			commit	[T_3 commit]
				point 2
20.		commit		[T_2 commit]

1. Das System stürzt an Punkt 1 ab. Geben Sie an, was in jeder Phase des Wiederanlaufs geschieht.
2. Diesmal stürzt das System bei Punkt 1 nicht ab, bei Punkt 2 stürzt es allerdings wieder ab. Geben Sie an, was in jeder Phase des Wiederanlaufs geschieht.

4 Deadlock Prevention vs. Detection

Unter welchen Bedingungen ist es günstiger, (i) Deadlocks zu vermeiden, als (ii) Deadlocks zuzulassen und sie dann zu erkennen?

5 Conflict Graph vs. Wait-For Graph

Beschreiben Sie die Unterschiede und Gemeinsamkeiten zwischen Conflict Graphs und Wait-For Graphs.

Sie können den folgenden Schedules als Grundlage für Ihre Diskussion verwenden:

step	T_1	T_2
1	lock_X(A)	
2	lock_S(B)	
3	write(A)	
4	read(B)	
5	unlock(A)	
6	unlock(B)	
7		lock_X(B)
8		lock_S(A)
9		write(B)
10		read(A)
11		unlock(B)
12		unlock(A)
13	commit	
14		commit