

Mobile und Verteilte Datenbanken

WS 2008/2009

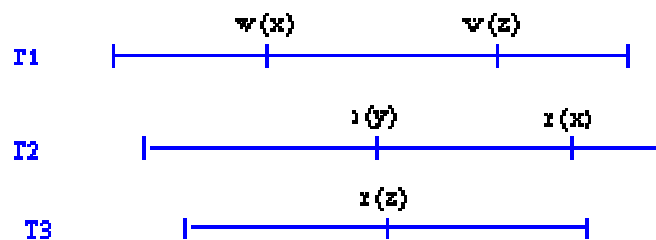
Blatt 7

Aufgabe 1: Nicht-blockierendes 2PC durch Prozeß-Paare

Blockierungen im 2PC-Protokoll können verhindert werden, wenn der Koordinator als Prozeß-Paar fehlertolerant realisiert wird. Dabei sendet der Koordinator-TM sämtliche für die Commit-Bearbeitung relevanten Zwischenzustände an einen Backup-TM in einem anderen Rechner. Nach Ausfall des Koordinators übernimmt der Backup-TM dessen Rolle. Wie hoch ist der Overhead einer solchen Lösung gegenüber einem 3PC-Protokoll?

Aufgabe 2: Vergleich von Synchronisationsverfahren

An einem Rechner sei folgender Schedule von drei Transaktionen mit Zugriff auf ausschließlich lokale Objekte gegeben, wenn keine Synchronisation erfolgt:



Bestimmen Sie für jedes der folgenden Synchronisationsverfahren die vorkommenden Blockierungen und Rücksetzungen sowie die sich ergebende Serialisierungsreihenfolge:

- Zeitmarkenverfahren (Basic Timestamp Ordering)
- rückwärtsorientierte, optimistische Synchronisation, bei der die Validierung ausschließlich gegenüber bereits beendeten Transaktionen erfolgt
- vorwärtsorientierte, optimistische Synchronisation, bei der die Validierung gegenüber laufende Transaktionen erfolgt
- RX-Sperrverfahren mit Wait/Die-Deadlock-Vermeidung
- RX-Sperrverfahren mit Wound/Wait-Deadlock-Vermeidung
- RX-Sperrverfahren mit Deadlock-Erkennung
- RX-Sperrverfahren mit Mehrversionen-Synchronisation und Deadlock-Erkennung.

Aufgabe 3: Deadlock-Erkennung

Wenden Sie den Algorithmus von Obermarck (wie er in R* Anwendung findet) zur Erkennung des folgenden Deadlocks an. Geben Sie jeden Zwischenschritt an (Annahme: $ts(T1) < ts(T2) < ts(T3)$). Wieviele Nachrichten werden zur Auflösung des Deadlocks benötigt?

