

Abgabetermin: 5.12.2018, 13:30 Uhr

☐ DES31UE Niklas Name Niklas Vest Aufwand in h 7
☐ DES32UE Niklas
☒ DES33UE Traxler Punkte _____ Kurzzeichen Tutor _____

1. Konfliktgraph**(10 Punkte)**

Betrachten Sie die folgenden Transaktionen und Ausführungspläne. Geben Sie die Konfliktrelationen an und zeichnen Sie die Serialisierbarkeits- bzw. Konfliktgraphen. Identifizieren Sie, ob jeder Ausführungsplan serialisierbar ist. Je nachdem notieren Sie den bzw. die äquivalenten seriellen Ausführungspläne oder die Konflikte.

a) $T_1 = w_1(x) \ r_1(y) \ w_1(y)$
 $T_2 = r_2(x) \ w_2(y)$
 $T_3 = w_3(x) \ w_3(y)$

$S_1 = w_1(x) \ r_2(x) \ w_2(y) \ r_1(y) \ w_1(y) \ w_3(x) \ w_3(y)$
 $S_2 = w_3(x) \ w_1(x) \ r_2(x) \ w_3(y) \ r_1(y) \ w_1(y) \ w_2(y)$

b) $T_1 = r_1(y) \ r_1(x) \ w_1(y) \ w_1(x)$
 $T_2 = r_2(z) \ w_2(x) \ w_2(y) \ w_2(z)$
 $T_3 = r_3(y) \ w_3(y)$

$S_3 = r_1(y) \ r_2(z) \ r_1(x) \ w_1(y) \ r_3(y) \ w_3(y) \ w_1(x) \ w_2(x) \ w_2(y) \ w_2(z)$
 $S_4 = r_1(y) \ r_1(x) \ w_1(y) \ r_3(y) \ r_2(z) \ w_2(x) \ w_1(x) \ w_2(y) \ w_3(y) \ w_2(z)$

c) $T_1 = r_1(x) \ r_1(z) \ w_1(x)$
 $T_2 = r_2(z) \ r_2(y) \ w_2(z) \ w_2(y)$
 $T_3 = r_3(x) \ r_3(y) \ w_3(y)$

$S_5 = r_1(x) \ r_2(z) \ r_1(z) \ r_3(x) \ r_3(y) \ w_1(x) \ w_3(y) \ r_2(y) \ w_2(z) \ w_2(y)$
 $S_6 = r_1(x) \ r_2(z) \ r_3(x) \ r_1(z) \ r_2(y) \ r_3(y) \ w_1(x) \ w_2(z) \ w_3(y) \ w_2(y)$

2. Fehlersicherheit von Transaktionen**(9 Punkte)**

Betrachten Sie die folgenden Ausführungspläne und stellen Sie fest, welcher Ausführungsplan strikt (ST), kaskadenlos (ACA) und/oder rücksetzbar (RC) ist. Begründen Sie Ihre Antwort für jeden Ausführungsplan.

$S_7 = r_1(y) \ r_1(x) \ w_1(x) \ w_2(x) \ r_1(x) \ w_1(y) \ \mathbf{a_1 \ a_2}$
 $S_8 = r_1(x) \ r_2(z) \ r_1(z) \ r_3(x) \ r_3(y) \ w_1(x) \ w_3(y) \ r_2(y) \ w_2(z) \ w_2(y) \ \mathbf{c_1 \ c_2 \ c_3}$
 $S_9 = w_1(x) \ w_1(y) \ r_2(z) \ \mathbf{c_1} \ w_2(x) \ r_2(y) \ w_2(y) \ \mathbf{c_2}$

	RC	ACA	ST
S_7			
S_8			
S_9			

Hinweis:

ST: Schedule s heißt strikt (engl. strict), falls folgende Bedingung gilt:

$$(w_j(x) \rightarrow_s p_i(x) \wedge j \neq i) \Rightarrow (a_j <_s p_i(x) \vee c_j <_s p_i(x), (p \in \{r, w\}))$$

D.h., es darf kein „geschriebenes“ Objekt einer noch nicht beendeten Transaktion gelesen oder überschrieben werden.

RC: s heißt rücksetzbar (engl. recoverable), falls folgende Bedingung erfüllt ist:

$$(T_i \text{ liest von } T_j \text{ in } s) \wedge (c_i \in s) \Rightarrow (c_j \rightarrow_s c_i)$$

Eine Transaktion darf erst dann ihr COMMIT durchführen, wenn alle Transaktionen, von denen sie gelesen hat, beendet sind.

ACA: Schedule s vermeidet kaskadierende Abbrüche (engl. avoiding cascading aborts ACA), falls folgende Bedingung erfüllt ist:

$$(T_i \text{ liest } x \text{ von } T_j \text{ in } s) \Rightarrow (c_j \rightarrow_s r_i(x))$$

D.h., eine Transaktion darf nur Daten lesen, die zuletzt von einer bereits abgeschlossenen Transaktion geschrieben wurden.

3. Zwei-Phasen-Sperrprotokoll

(4 Punkte)

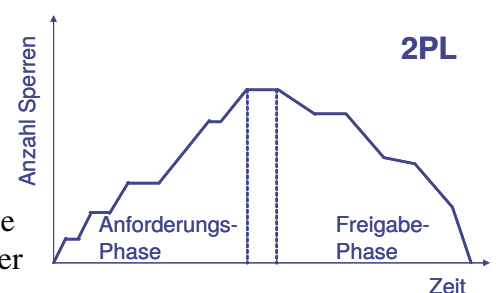
Gegeben ist der Ausführungsplan S_{10} mit folgender parallelausführung von Transaktionen:

$$S_{10} = r_1(x) \ r_2(y) \ w_1(x) \ r_3(z) \ r_2(x) \ r_3(x) \ w_3(z) \ w_2(z) \ c_1 \ c_2 \ c_3$$

Sie haben zwei Sperrbefehle $rl_i(x)$ und $wl_i(x)$ sowie einen Unlock-Befehl $u_i(x)$ zur Verfügung. Wie würde das einfache Zwei-Phasen-Sperrprotokoll bei diesem Ausführungsplan vorgehen? Geben Sie für S_{10} einen möglichen Ausführungsplan für das Zwei-Phasen-Sperrprotokoll an.

Der Zugriff auf gemeinsam benutzte Daten wird durch Sperren synchronisiert:

1. Schreibzugriff $w(x)$ nur nach Setzen einer Schreibsperre $wl(x)$ möglich
2. Lesezugriffe $r(x)$ nur nach $rl(x)$ oder $wl(x)$ erlaubt
3. Eine Schreibsperre $wl(x)$ darf nur auf Objekte erfolgen, die nicht bereits von einer anderen Transaktion zum Schreiben gesperrt sind
4. Nach $rl(x)$ nur noch $wl(x)$ erlaubt, danach auf x keine Sperre mehr; Sperren derselben Art auf ein Objekt werden innerhalb einer Transaktion maximal einmal gesetzt.
5. Nach $u(x)$ durch T_i darf T_i kein erneutes $rl(x)$ oder $wl(x)$ ausführen
6. Vor einem commit müssen alle Sperren aufgehoben werden
7. Wenn die Sperränderung zulässig ist, muss die Verschärfung von Sperren, z.B. von $rl(x)$ zu $wl(x)$, während der Wachstumsphase erfolgen.
8. Wenn die Sperränderung zulässig ist, darf die Abschwächung von Sperren, z.B. eine $rl(x)$ -Operation, die eine bereits gehaltene Schreibsperre $wl(x)$ abschwächt, nur in der Schrumpfungsphase erfolgen.
9. Eine Transaktion folgt dem Zwei-Phasen-Sperrprotokoll, wenn alle Sperroperationen (rl , wl) vor der ersten Entsperroperation einer Transaktion ausgeführt werden.



4. Theoriefragen zum Mehrbenutzerbetrieb

(1 Punkt)

Welche Aussagen treffen zu? Kreuzen Sie an.

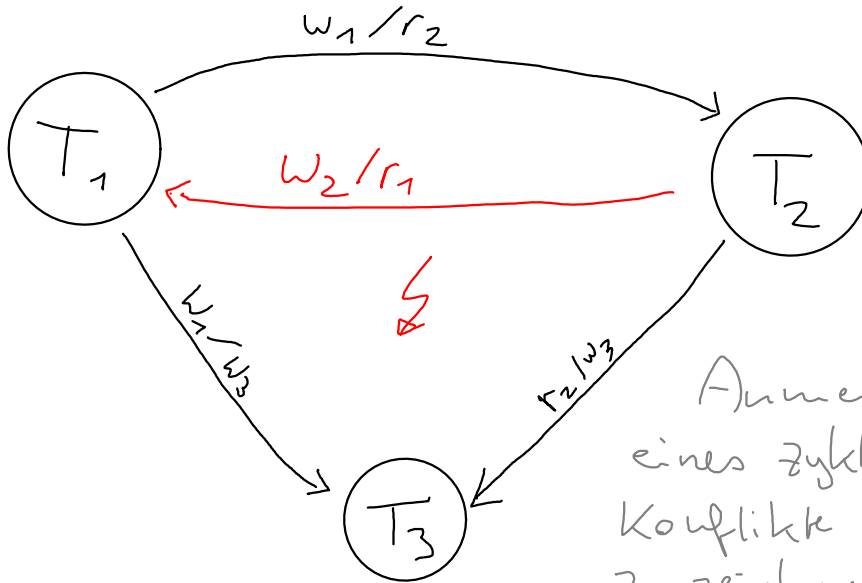
- ☐ Datenbanksysteme bewerten im Mehrbenutzerbetrieb die Fehlersicherheit, um eine Isolationsstufe zu wählen.
- ☐ Serielle Schedules gewährleisten, dass keine Konflikte auftreten, allerdings sinkt die Performance.
- ☐ Wenn ein Schedule strikt ist, können Sie davon ausgehen, dass er auch konfliktserialisierbar ist.

1. a)

vest

$$S_1 = \underline{w_1(x)} \underline{r_2(x)} \underline{w_2(y)} \underline{r_1(y)} \underline{w_1(y)} \underline{w_3(x)} \underline{w_3(y)}$$

$$C_{S_1} = \left\{ (\underline{w_1(x)}, \underline{r_2(x)}), (\underline{w_1(x)}, \underline{w_3(x)}), (\underline{r_2(x)}, \underline{w_3(x)}), \right. \\ \left. (\underline{w_2(y)}, \underline{r_1(y)}), (\underline{w_2(y)}, \underline{w_3(y)}), (\underline{r_1(y)}, \underline{w_3(y)}), \right. \\ \left. (\underline{w_1(y)}, \underline{w_3(y)}), (\underline{w_2(y)}, \underline{w_1(y)}) \right\}$$

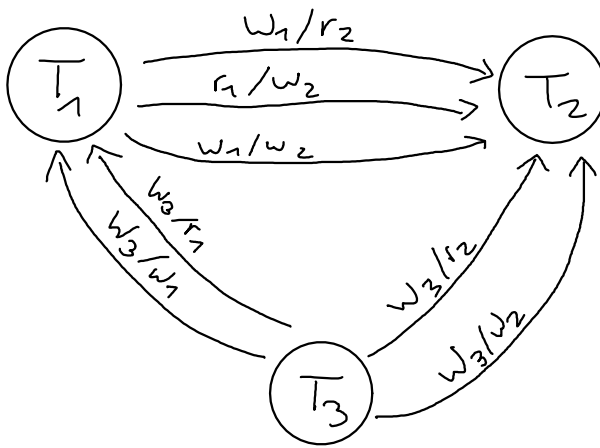


nicht
serialisierbar!

Anmerkung: Nach auftreten
eines Zyklus h re ich auf
Konflikte in den Graphen
zu zeichnen.

$$S_2 = \underline{w_3(x)} \underline{w_1(x)} \underline{r_2(x)} \underline{w_3(y)} \underline{r_1(y)} \underline{w_1(y)} \underline{w_2(y)}$$

$$C_{S_2} = \left\{ (\underline{w_3(x)}, \underline{w_1(x)}), (\underline{w_3(x)}, \underline{r_2(x)}), (\underline{w_1(x)}, \underline{r_2(x)}), \right. \\ \left. (\underline{w_3(y)}, \underline{r_1(y)}), (\underline{w_3(y)}, \underline{w_1(y)}), (\underline{w_3(y)}, \underline{w_2(y)}), \right. \\ \left. (\underline{r_1(y)}, \underline{w_2(y)}), (\underline{w_1(y)}, \underline{w_2(y)}) \right\}$$



serialisierbar!

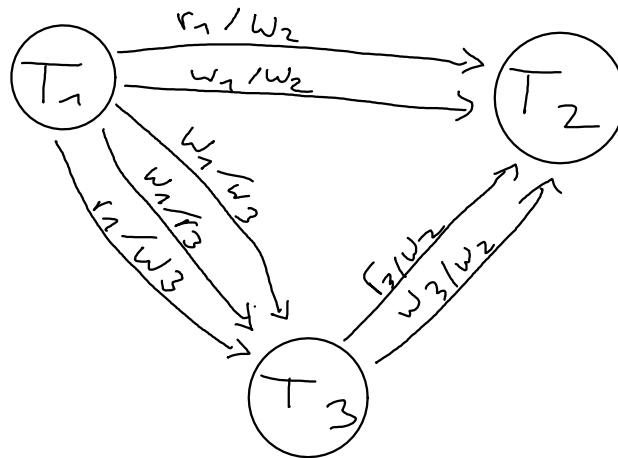
$T_3 \rightarrow T_1 \rightarrow T_2$

b)

$S_3 = \underline{r_1(y)} \underline{r_2(z)} \underline{r_1(x)} \underline{w_1(y)} \underline{r_3(y)} \underline{w_3(y)} \underline{w_1(x)} \underline{w_2(x)} \underline{w_2(y)} \underline{w_2(z)}$

Wechsel auf $[x]$ für bessere Lesbarkeit!

$$C_{S_3} = \left\{ (r_1[y], w_3[y]), (r_1[y], w_2[y]), (r_1[x], w_2[x]), \right. \\ (w_1[y], r_3[y]), (w_1[y], w_3[y]), (w_1[y], w_2[y]), \\ \left. (r_3[y], w_2[y]), (w_3[y], w_2[y]), (w_1[x], w_2[x]) \right\}$$



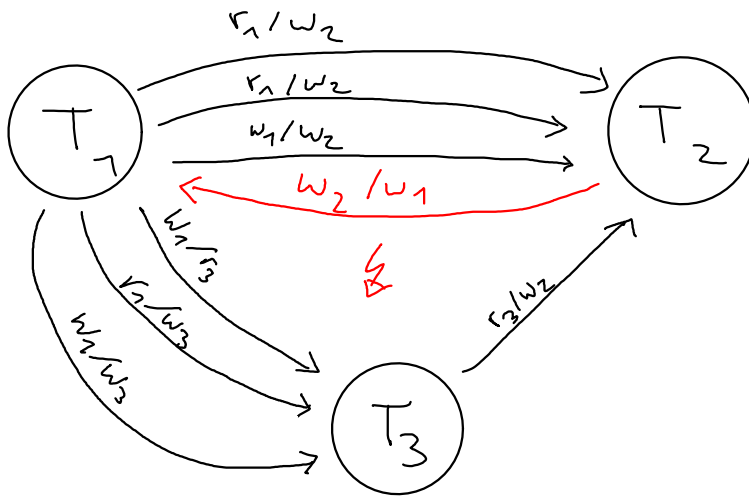
serialisierbar!

$T_1 \rightarrow T_3 \rightarrow T_2$

$S_4 = \underline{r_1(y)} \underline{r_1(x)} \underline{w_1(y)} \underline{r_3(y)} \underline{r_2(z)} \underline{w_2(x)} \underline{w_1(x)} \underline{w_2(y)} \underline{w_3(y)} \underline{w_2(z)}$

$$C_{S_4} = \left\{ (r_1[y], w_2[y]), (r_1[y], w_3[y]), (r_1[x], w_2[x]), \right. \\ (w_1[y], r_3[y]), (w_1[y], w_2[y]), (w_1[y], w_3[y]), \\ \left. (r_3[y], w_2[y]), (w_2[x], w_1[x]), (w_2[y], w_3[y]) \right\}$$

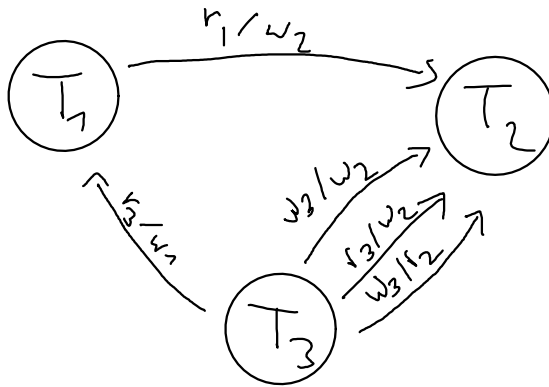
Verst



nicht
serialisierbar!

c) $S_5 = r_1(x) \ r_2(z) \ \underline{r_1(z)} \ \underline{r_3(x)} \ \underline{r_3(y)} \ \underline{w_1(x)} \ \underline{w_3(y)} \ \underline{r_2(y)} \ \underline{w_2(z)} \ \underline{w_2(y)}$

$$C_{S_5} = \left\{ (r_1[z], w_2[z]), (r_3[x], w_1[x]), (r_3[y], w_2[y]), (w_3[y], r_2[y]), (w_3[y], w_2[y]) \right\}$$

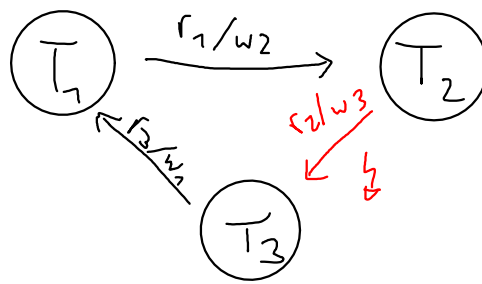


serialisierbar!

$$T_3 \rightarrow T_1 \rightarrow T_2$$

$S_6 = r_1(x) \ r_2(z) \ \underline{r_3(x)} \ \underline{r_1(z)} \ \underline{r_2(y)} \ \underline{r_3(y)} \ \underline{w_1(x)} \ \underline{w_2(z)} \ \underline{w_3(y)} \ \underline{w_2(y)}$

$$C_{S_4} = \left\{ (r_3[x], w_1[x]), (r_1[z], w_2[z]), (r_2[y], w_3[y]), (w_3[y], w_2[y]) \right\}$$



nicht
serialisierbar!

2)

	RC	ACA	ST
S_7	✓	✗	✗
S_8	✗	✗	✗
S_9	✓	✓	✓

$$S_7 = r_1(y) \ r_1(x) \ w_1(x) \ w_2(x) \ r_1(x) \ w_1(y) \ a_1 \ a_2$$

$$S_8 = r_1(x) \ r_2(z) \ r_1(z) \ r_3(x) \ r_3(y) \ w_1(x) \ w_3(y) \ r_2(y) \ w_2(z) \ w_2(y) \ c_1 \ c_2 \ c_3$$

$$S_9 = w_1(x) \ w_1(y) \ r_2(z) \ c_1 \ w_2(x) \ r_2(y) \ w_2(y) \ c_2$$

- RC verhindert durch ...
- ACA verhindert durch ...
- ST verhindert durch ...

3)

Schritt	T_1	T_2	T_3	Bemerkung	Locks		
					x	y	z
1	w[x]				$T_1[w]$		
2	r[x]				$T_1[w]$		
3		r[y]			$T_1[w]$	$T_2[r]$	
4		r[y]			$T_1[w]$	$T_2[r]$	
5	w[x]				$T_1[w]$	$T_2[r]$	
6	v[x]					$T_2[r]$	
7	commit					$T_2[r]$	
8			w[z]			$T_2[r]$	$T_3[w]$
9			r[z]			$T_2[r]$	$T_3[w]$
10			r[x]		$T_3[r]$	$T_2[r]$	$T_3[w]$
11		r[x]		T_2 wartet	$T_3[r]$	$T_2[r]$	$T_3[w]$
12			r[x]		$T_3[r]$	$T_2[r]$	$T_3[w]$
13			v[x]	T_2 wecken	$T_2[r]$	$T_2[r]$	$T_3[w]$
14		r[x]			$T_2[r]$	$T_2[r]$	$T_3[w]$
15			w[z]		$T_2[r]$	$T_2[r]$	$T_3[w]$
16			v[z]		$T_2[r]$	$T_2[r]$	
17			commit		$T_2[r]$	$T_2[r]$	
18		w[z]			$T_2[r]$	$T_2[r]$	$T_2[w]$
19		w[z]			$T_2[r]$	$T_2[r]$	$T_2[w]$
20		v[x]				$T_2[r]$	$T_2[w]$
21		v[y]					$T_2[w]$
22		v[z]					
23		commit					

4)

Verst

Welche Aussagen treffen zu? Kreuzen Sie an.

- ☐ Datenbanksysteme bewerten im Mehrbenutzerbetrieb die Fehlersicherheit, um eine Isolationsstufe zu wählen.
- ☒ Serielle Schedules gewährleisten, dass keine Konflikte auftreten, allerdings sinkt die Performance.
- ☐ Wenn ein Schedule strikt ist, können Sie davon ausgehen, dass er auch konfliktserialisierbar ist.