

März 2017

Aufgabe 3

Zeitstempel - basierende Synchronisation

#	T ₁	T ₂	readTS(A)	writeTS(A)	readTS(B)	writeTS(B)
1	SELECT A INTO valueA1		...5...	...0...	...0...	...0...
2	valueA1 = valueA1 + 10		-	-	-	-
3	UPDATE A = valueA1		...5...	...5...	...0...	...0...
4		SELECT A INTO valueA2	...10...	...5...	...0...	...0...
5		valueA2 = valueA2 * 2	-	-	-	-
6		UPDATE A = valueA2	...10...	...10...	...0...	...0...
7		SELECT B INTO valueB1	...10...	...10...	...10...	...0...
8		valueB = valueB1 + 10	-	-	-	-
9		UPDATE B = valueB1	...10...	...10...	...10...	...10...
10	SELECT B INTO valueB2		...10...	...10...	...10...	...10...
11	valueB2 = valueB2 - 10		-	-	-	-
12	UPDATE B = valueB2		...10...	...10...	...10...	...10...

Zeitstempel: TS(T₁) = 5
TS(T₂) = 10

→ bei "SELECT" → read
bei "UPDATE" → write

z.B. → #1 → SELECT A INTO value A1
→ Transaktion 1 wird verwendet, also TS(T₁) = 5
→ es wird A verwendet
→ es wird ein SELECT, also read verwendet
→ deshalb schreiben wir in die Spalte "readTS(A)" → 5
→ alles wird so fortgesetzt und die restlichen Werte von der oberen Zeile übernommen

Achtung! z.B. bei #10 → unsere aktuellen Werte sind bereits 10er (bei Zeile #9)

nun kommt aber ein "Select", "B" von T₁
→ wenn wir es wie oben machen würden, würde jetzt in "readTS(B)" → 5 stehen, jedoch war der frühere Wert größer (10) und deshalb bleibt 10 stehen!
also Achtung! den größeren Wert stehen lassen!

Jänner 2015

Aufgabe 3

Serialisierbarkeitsgraph 8

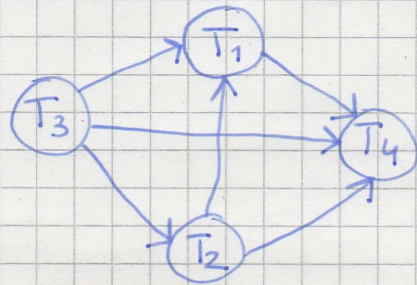
Wartegraph

a) Serialisierbarkeitsgraph

$r_1(A), r_2(B), r_3(A), w_2(B), w_3(C), r_4(D), r_1(B), r_1(C), r_2(C),$
 $r_3(C), w_4(C), c_1, c_2, c_3, c_4$

$w_2(B) \rightarrow r_1(B)$	$(T_2 \rightarrow T_1)$
$r_2(B) \rightarrow w_2(B)$	$(T_2 \rightarrow T_1)$
$w_3(C) \rightarrow r_1(C)$	$(T_3 \rightarrow T_1)$
$w_3(C) \rightarrow r_2(C)$	$(T_3 \rightarrow T_2)$
$r_1(C) \rightarrow w_4(C)$	$(T_1 \rightarrow T_4)$
$w_3(C) \rightarrow w_4(C)$	$(T_3 \rightarrow T_4)$
$r_2(C) \rightarrow w_4(C)$	$(T_2 \rightarrow T_4)$

→ wir schauen uns alle Kombinationen einer Variable an
"read + write",
"write + read" oder "write + write"
NICHT! → "read + read"



mögliche Reihenfolge der Serialisierung: man beginnt bei dem Knoten der KEINE eingehenden Kanten hat und entfernt diesen und all seine Kanten, usw.

→ $T_3 \rightarrow T_2 \rightarrow T_1 \rightarrow T_4$

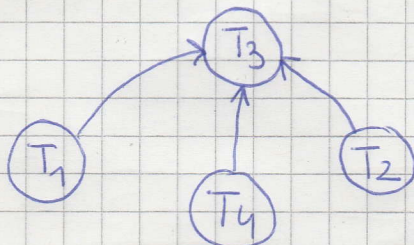
b) Wartegraph

$lockS_1(A), lockX_2(B), lockX_3(C), lockS_1(C), lockS_3(B), lockS_4(C), lockS_2(A)$

→ $lockX_2(B), lockS_3(B)$
→ $lockX_3(C), lockS_1(C)$
→ $lockX_3(C), lockS_4(C)$

T_3 muss auf T_2 warten
 $T_1 \rightarrow T_3$
 $T_4 \rightarrow T_3$

→ wieder nur Kombis die eine Schreibsperr enthalten
! KEIN → $lockS(x), lockS(x)$

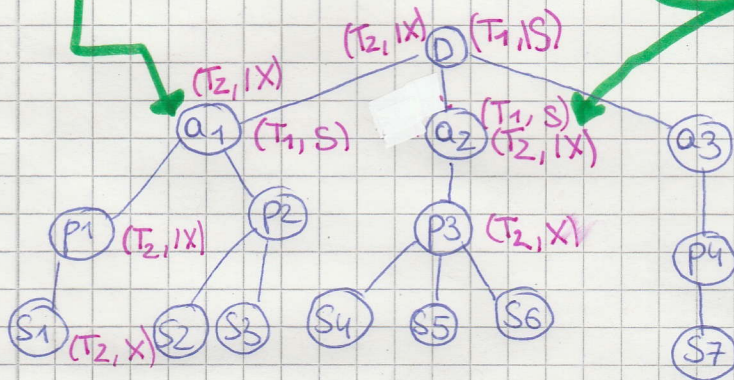


Multi-Granularity-Locking

(Aufgabe 2, 5. 2015)

Tabelle

	NL	S	X	IS	IX
S	✓	✓	-	✓	-
X	✓	-	-	-	-
IS	✓	✓	-	✓	✓
IX	✓	-	-	✓	✓



③. $(T_1, D, IS), (T_2, D, IX), (T_1, a_1, S), (T_2, a_2, IX), (T_2, p_3, X), (T_1, a_2, S), (T_2, a_1, IX), (T_2, p_1, IX), (T_2, s_1, X)$

Blockierung: bei $a_2 \rightarrow (T_2, IX), (T_1, S) \rightarrow IX$ und S vertragen sich nicht

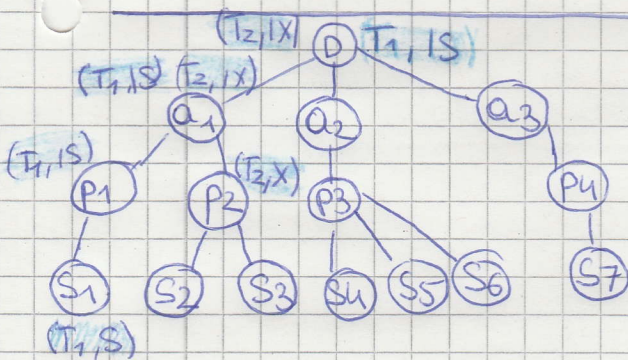
$a_1 \rightarrow (T_1, S), (T_2, IX) \rightarrow \text{---}$

Deadlock: entstehen wenn Transaktionen sich gegenseitig blockieren

hier: bei $a_2 \rightarrow$ zuerst T_2 , dann T_1
 $\rightarrow T_1$ wartet auf T_2

$\rightarrow$ bei $a_1 \rightarrow$ zuerst T_1 , dann T_2
 $\rightarrow T_2$ wartet auf T_1

$\hookrightarrow$ deshalb Deadlock



①. $(T_1, D, IS), (T_2, D, IX), (T_1, a_1, IS), (T_1, p_1, IS), (T_2, a_1, IX), (T_2, p_2, X), (T_1, s_1, S)$

Blockierung: hier kommt es zu keiner Blockierung, da sich die Sperren vertragen

(wenn keine Blockierung, dann kann es auch gar nicht zu einem Deadlock kommen)

Januar 2015

Aufgabe 2

Kostenberechnung

A₁: h → Kosten = 0, Anzahl der Seiten ⇒

Tupelanzahl · Tupelgröße
Bytes / Seite

$$\frac{60.000 \cdot 16}{1000} = \underline{960} : \text{Ant. Seiten!}$$

A₂: Access

→ Kosten: gleich Seitenanzahl

→ Seitenanzahl: $\frac{40.000 \cdot 60}{1000} = \underline{2400}$ Kosten: 2400

A₃: Tsem=3

Tupelanzahl: von Oben mal Selektion ($\text{Sel}_{\text{S.Em}} = 0,1$), Kosten = 0
 $= 40000 \cdot 0,1 = \underline{4000} = \text{Tupelanzahl}$

Tupelgröße: gleich wie Oben

Seitenanzahl: $\frac{4000 \cdot 60}{1000} = \underline{240}$

A₄: [Zwischenspeichern]

alles bleibt gleich wie Oben

NUR Kosten = entsprechen der Seitenanzahl

Kosten = 240

A₅: NL-M

Anzahl d. Tupel: $\text{AntT}_{\text{links}} \cdot \text{AntT}_{\text{rechts}} \cdot \text{Sel}_{\text{S/h}}$

$$= 60.000 \cdot 4000 \cdot \left(\frac{1}{40.000}\right) = \underline{6000}$$

Tupelgröße = $T_{\text{Größe links}} + T_{\text{Größe rechts}} = 16 + 60 = \underline{76}$

Seitenanzahl = $\frac{6000 \cdot 76}{1000} = \underline{456}$

Kosten:

$$b_1 + 30 + \lceil b_1 / (32 - 30 - 1) \rceil \cdot (b_4 - 30) = \underline{202.590}$$

! HINWEIS!

normalerweise ist das "30" eigentlich "k=1", LVA Leitung könnte auch nicht erklären wieso sie genau das verwendet in Zukunft soll k angegeben werden!

A₆: [Zwischenspeichern]

alles gleich nur Kosten = Seitenanzahl

Kosten = 456

A₇: v

Kosten 0, Seiten: $\frac{2500 \cdot 120}{1000} = \underline{300}$

A₈: NL-M

$$b_6 + 30 + \lceil b_6 / (32 - 30 - 1) \rceil \cdot (b_7 - 30) =$$

$$456 + 30 + \lceil 456 / 17 \rceil \cdot (300 - 30) = \underline{123.606} = \text{Kosten}$$

$$\text{AntTupel} = \text{Ant}_{\text{links}} \cdot \text{Ant}_{\text{rechts}} \cdot \text{Sel}_{\text{v/h}} = 6000 \cdot 2500 \cdot (1/2500) = \underline{6000}$$

Tupelgröße = $T_{\text{Größe links}} + T_{\text{Größe rechts}} = 76 + 120 = \underline{196}$

Seitenanzahl: $\frac{6000 \cdot 196}{1000} = \underline{1176}$

Jänner 2016

Aufgabe 3

Kanonische Übersetzung 8

Optimierter Ausdruck

Fragmentierung:

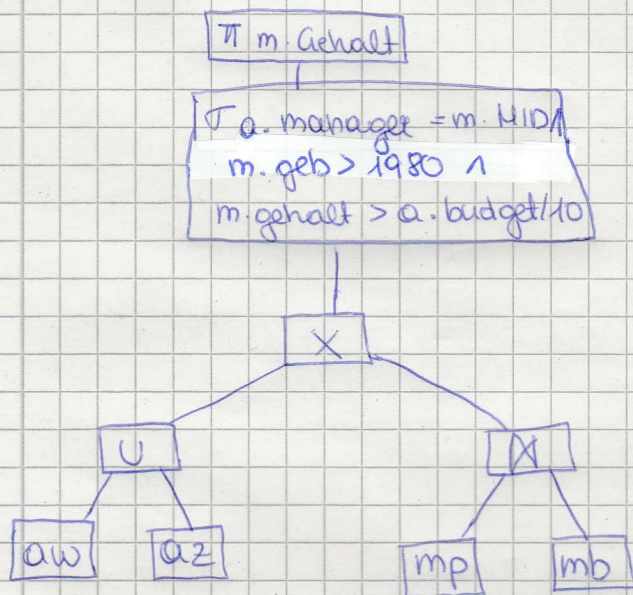
horizontal: Vereinigung

vertikal: Join

Kanonische Übersetzung

→ in die Projektion (π) schreiben wir das "SELECT"

→ in Selektion (σ) schreiben wir das "WHERE" und "and" wird zu " $\wedge$ "

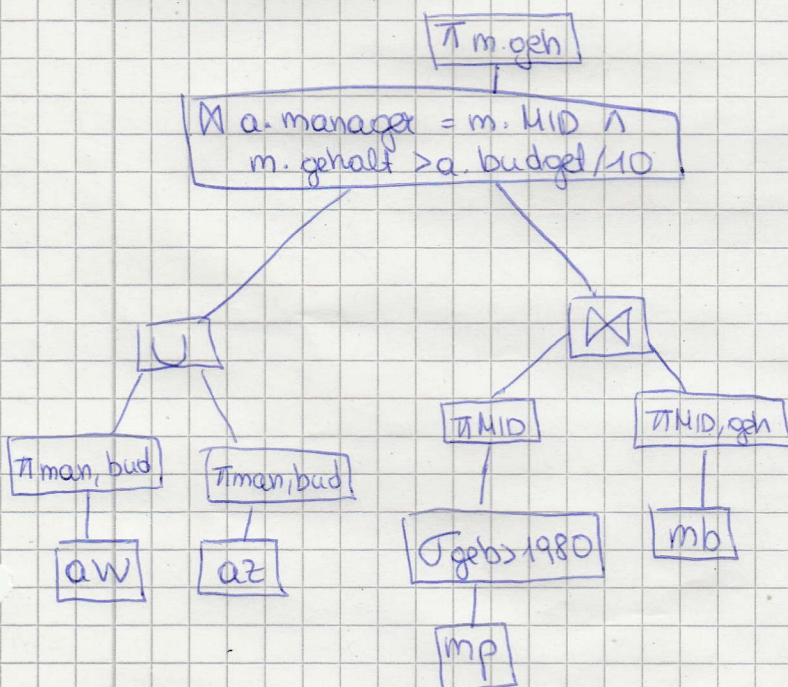


Kreuzprodukt zum Verknüpfen

→ horizontal: Abteilung Wien (aw)
Abteilung Zürich (az)

→ vertikal: Mitarbeiter Privat (mp)
Mitarbeiter Beruf (mb)

Optimierter Ausdruck



→ select (π) bleibt gleich

→ wir beginnen unten und schauen welche Attribute gebraucht werden und (π) projizieren sie raus

→ wir brauchen:
aw: manager, budget
az: manager, budget
mp: MID
mb: MID, Gehalt

→ zusätzlich selektieren wir gleich am Anfang " $\sigma \text{geh} > 1980$ " raus aus "mp"

3.05.17

Aufgabe 3

Histrie von Transaktionen

A=100, B=200, C=100

	A	B	C	c ₁	c ₂	a ₃	b ₃	b ₂
4	100	200	100	100				
5					100			
6						100		
7	200							
8	300							
9							200	
10		300						
11			300					300
		400						

Schritt 4: $r(C, c_1) \rightarrow c_1 = 100$

S5: $r(C, c_2) \rightarrow c_2 = 100$

S6: $r(A, a_3) \rightarrow a_3 = 100$

S7: $w(A, c_2 \cdot 2) \rightarrow c_2 \cdot 2 \Rightarrow 100 \cdot 2 \Rightarrow 200$ $A=200$ $A+=100$
Veränderung v. 100 auf 200

[#4, T₂, P_A, A+=100, A-=100, #2]
 ↑ wird einfach durch nummeriert
 ↑ Transaktion
 ↑ welche Variable
 ↑ Veränderung
 ↑ gegenüber
 ↑ vorheriger Logeintrag! dieser! Transaktion - also einfach in der Spalte schauen wo zuletzt "rechts" etwas eingetragen wurde

S8: $w(A, c_1 + 200)$
 $c_1 + 200 = 100 + 200 = 300 \rightarrow A=300$, Veränderung von 200 auf 300 = +100

[#5, T₁, P_A, A+=100, A-=100, #1]

S9: $r(B, b_3) \rightarrow b_3 = 200$

S10: $w(B, a_3 + b_3) \rightarrow 100 + 200 = 300 \rightarrow B=300$, Veränderung = +100

[#6, T₃, P_B, B+=100, B-=100, #3]

S11: $w(C, c_2 + 200) \rightarrow 100 + 200 = 300 \rightarrow C=300$, Veränderung = +200

[#7, T₂, P_C, C+=200, C-=200, #4]

S12: $r(B, b_2) \rightarrow b_2 = 300$

S13: $w(B, b_2 + c_2) \rightarrow 300 + 100 = 400 \rightarrow B=400$, Veränderung: +100

[#8, T₂, P_B, B+=100, B-=100, #7]

S14: [9, T₂, commit, #8]

! Systemabsturz!

S15: $\langle \#10, T_3, P_3, B=100, \#6, \#3 \rangle$
 Klammeren jetzt 00 Undo, also "2. Wert" von oben Anfangsnummer des zurückgesetzten Log-Eintrags vorhergehender Logeintrag
 des zurückgesetzten (einfach 1. und letzte Zahl des alten Log-Eintrags)

→ es werden nur die Transaktionen zurückgesetzt, die bis zum Systemabsturz nicht "committed" wurden → hier also T_1 und T_3

→ man geht von unten nach oben durch und setzt alles nacheinander zurück

S16: $\langle \#11, T_1, P_3, A=100, \#5, \#1 \rangle$

S17: $\langle \#12, T_3, (BOT), \#10, 0 \rangle$

hier nimmt man den "letzten Log-Eintrag" von den schon zurückgesetzten dieser Transaktion

↳ man sucht also wo ab dem "Systemabsturz" der letzte Log-Eintrag mit dieser Transaktion existiert und nimmt seine AnfangsNr.

S18: $\langle \#13, T_1, (BOT), \#11, 0 \rangle$