

=1=

Klausur zum DBS II

im ~~GW~~ SoSe 2008

Dr. Peter Kröger

1. Aufgabe: Multiple Choice

Für richtige Antwort: +1P
Für falsche Antwort: -1P (20 Punkte)
Die minimale Anzahl der Punkte ist 0.

1. Besitzen zwei Schedules gleiche Abhängigkeitsmengen, dann sind sie konfliktäquivalent. ja nein
☐ ☒
2. Bei einem Serialisierungsgraphen entsprechen die Knoten den Objekten und die Kanten den Abhängigkeiten der TAS. ja nein
☐ ☒
3. Wenn das 2 PL zum Einsatz kommt, dann ist die Serialisierbarkeit des Schedules gewährleistet. ja nein
☒ ☐
4. Das Loggranular soll immer kleiner oder gleich dem Sperrgranular sein, sonst sind Lost Updates möglich. ja nein
☒ ☐
5. Beim RAX-Sperrverfahren werden bei einem bestehenden A-Lock noch weitere Leser zugelassen. ja nein
☒ ☐
6. Die Ausschreibestrategie Force besagt, dass die TAS veränderte Seiten sofort nach jeder Schreiboperation in die Datenbank eingeführt werden. ja nein
☐ ☒
7. Die Compensation-Log-Record (CLR) dient zur Gewährleistung der Stempotenz der Undo-Phase. (Stempotenz: $\text{Undo}(\text{undo}(\dots(\text{undo}(x))\dots)) = \text{Undo}(x)$) ja nein
☒ ☐
8. Bei BOC werden in der Validierungsphase die noch laufenden TAS betrachtet. ja nein
☐ ☒

9. Serialisierbare Schedules sind rücksetzbar ja nein
☐ ☒ = 2 =
10. Können Zeitstempel statt Sperren auf Objekte zur Synchronisation zum Einsatz, müssen beim Rücksetzen einer TA die zugehörigen Zeitstempel ebenfalls zurückgesetzt werden. ja nein
☒ ☐
11. Beim NBL-Join kann es sowohl bei zu kleiner, als auch bei zu großer Wahl der Blockgröße zu Performance-Problemen kommen. ja nein
☒ ☐
12. Das WAL-Prinzip fordert, dass nach jeder Änderung der DB die zugehörigen UNDO-Info in die Log-Datei geschrieben werden müssen. ja nein
☐ ☒
13. Unter Latenzzeit versteht man die Zeit, die benötigt wird, um den Schreib-/Lesekopf auf den entsprechenden Zylinder zu positionieren. ja nein
☐ ☒
14. Hash-Join Verfahren können nur für Equi- und Natural-Join effizient genutzt werden. ja nein
☒ ☐
15. Beim Hash-Partition-Join (GRACE) müssen Tupel aus jedem Block der inneren Relation mit jedem Block der äußeren Relation auf Erfüllung des Joinprädikats untersucht werden. ja nein
☐ ☒
16. Daten in einem Data-Warehouse können verändert werden. ja nein
☐ ☒
17. Ein Data-Warehouse enthält immer die aktuelle Version der integrierten operationalen Datenbanken. ja nein
☐ ☒

=3=

18.

Bei der DICE-Operation ändert sich die Dimensionalität eines Datacubes.

ja nein
☐ ☒

19.

Beim Star-Schema gibt es für jede Dimension genau eine Dimensionstabelle.

ja nein
☒ ☐

20.

Beim Star-Yarn wird genau eine Faktortabelle mit mehreren Dimensionstabellen geknüpft.

ja nein
☒ ☐

2 Aufgabe: ACID-Prinzip

(4 Punkte)

Nennen sie die 4 Bestandteile des ACID-Prinzip und erläutern Sie diese kurz.

A - Atomicity : Änderungen einer TA kommen entweder ganz oder gar nicht in die DB.
(ALL OR NOTHING).

C - Consistency : Die TA überführt die DB in einem konsistenten zu einem anderen konsistenten Zustand.

I - Isolation : (Logischer Einbenutzer Betrieb)
Eine TA wird nicht von anderen TAs beeinflusst.

D - Durability : Änderungen von beendeten TAs bleiben persistent (dauerhaft) in die DB erhalten.

3. Aufgabe

(10 Punkte) ⁼⁴⁼

Man zeichne den Abhängigkeitsgraphen zu dem folgenden Schedule und gebe gegebenenfalls den äquivalenten seriellen Schedule.

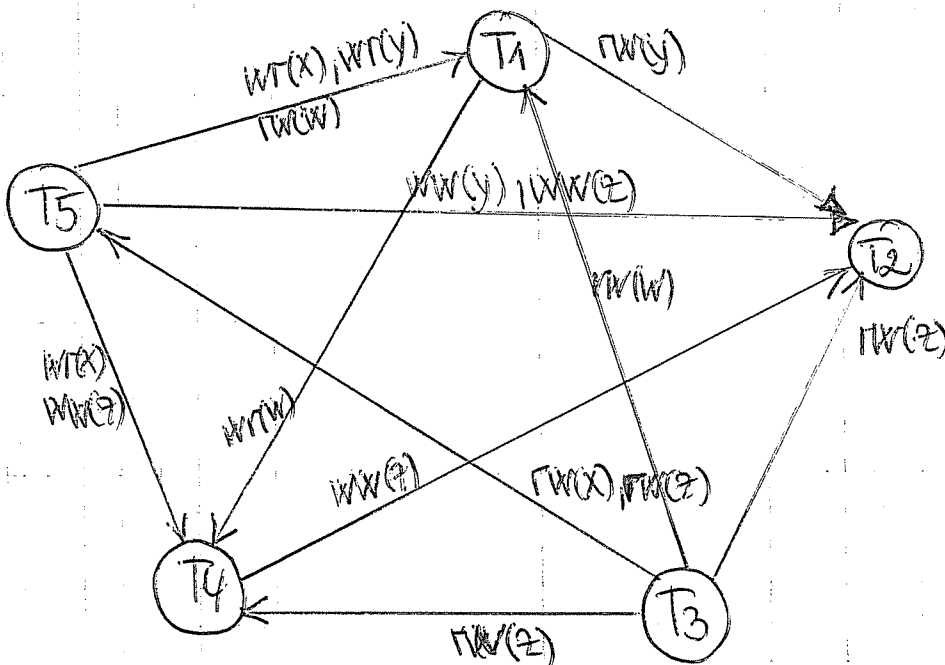
$$S = \left(r_3(x), w_5(x), r_3(w), w_5(y), r_3(z), r_1(y), r_4(x), w_5(z), r_5(w), w_1(w), w_4(z), w_2(y), r_1(x), w_2(z), r_4(w) \right)$$

Aktionen für x: $r_3(x), w_5(x), r_4(x), r_1(x)$

y: $w_5(y), r_1(y), w_2(y)$

z: $r_3(z), w_5(z), w_4(z), w_2(z)$

w: $r_3(w), r_5(w), w_1(w), r_4(w)$



Der Serialisierungsgraph ist zyklensfrei \Rightarrow also ist der Schedule serialisierbar. Ein äquivalenter serieller Schedule (der einzige hier).

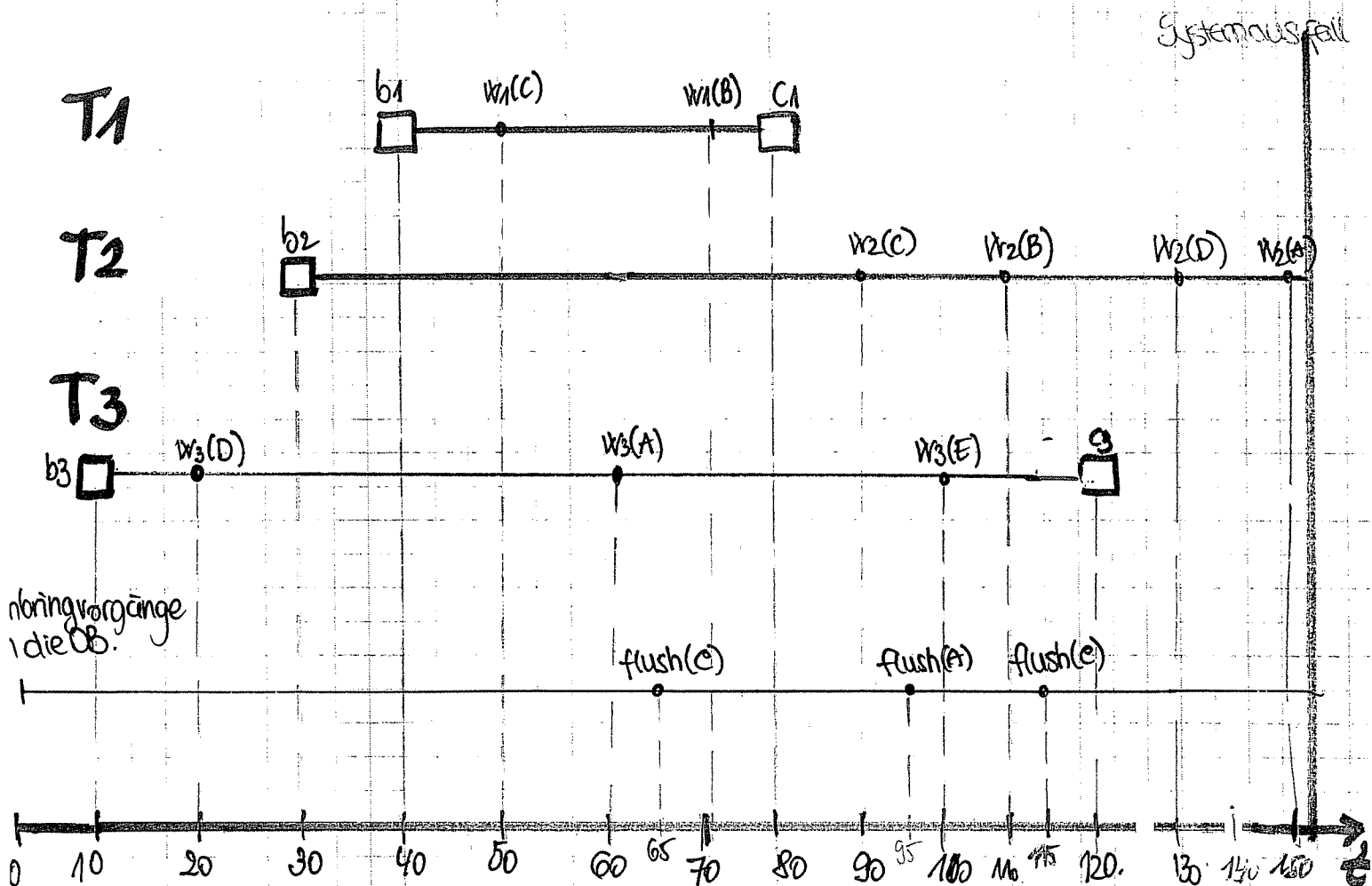
$T_3 \ T_5 \ T_1 \ T_4 \ T_2$

=5=

4. Aufgabe: Logging/Recovery

(8 Punkte)

- Parallel laufende T_1, T_2, T_3
- Datenelemente A, B, C, D, E , $X \in \{A, B, C, D, E\}$ sei in Seite P_X enthalten.
- Die modifizierten Seiten P_C, P_A werden durch $\text{flush}(P_C), \text{flush}(P_A)$ und $\text{flush}(P_C)$ zum Zeitpunkt 65, 95 und 115 aus dem Puffer verdrängt und in die DB eingebracht.
- Einbringstrategie: No-Atomic (Update in Place)
- Ausschreibestrategie: No-Force / Steal (wenn man flush machen will)
- Der letzte Checkpoint vor dem Systemausfall war zum Zeitpunkt 0 abgeschlossen.
- Keinerlei Checkpoints werden während der Ausführung gesetzt.
- Zu Beginn hat die LSN den Wert 0 und alle PageLSN werden auch auf 0 gesetzt.
- Gekennzeichnet Sie, wann Sie WAL- und CoWIT-Prinzip verwenden.



-6-

zur Log-File
hin zugefügte
Einträge

Zeit LSN	Aktion	Änderung DB-Puffer	Änderung DB	Log-Eintrag	
10	b ₃			10, T ₃ , BOTO	
20	w ₃ (D)	P _D , 20		20, T ₃ , P _D , R(D), U(D), 10	
30	b ₂			30, T ₂ , BOT, 0	
40	b ₁			40, T ₁ , BOT, 0	
50	w ₁ (C)	P _C , 50		50, T ₁ , P _C , R(C), U(C), 40	
60	w ₃ (A)	P _A , 60		60, T ₃ , P _A , R(A), U(A), 20	
65	flush(P _E)		P _C , 50		10-50 WAL
70	w ₁ (B)	P _B , 70		70, T ₁ , P _B , R(B), U(B), 50	
80	C ₁			80, T ₁ , LOT, 70	60-80 COMMIT
90	w ₂ (C)	P _C , 90		90, T ₂ , P _C , R(C), U(C), 30	
95	flush(P _A)		P _A , 60		-
100	w ₃ (E)	P _E , 100		100, T ₃ , P _E , R(E), U(E), 60	
110	w ₂ (B)	P _B , 110		110, T ₂ , P _B , R(B), U(B), 50	
115	flush(P _E)		P _C , 90		90 WAL
120	C ₃			120, T ₃ , LOT, 100	95-120 COMMIT
130	w ₂ (D)	P _D , 130		130, T ₂ , P _D , R(D), U(D), 110	
150	w ₂ (A)	P _A , 150		150, T ₂ , P _A , R(A), U(A), 130	

Eigentlich sollte hier 100 statt 95
stehen, da es ein double-Syncing
es nicht gibt.

5. Aufgabe Join-Verfahren

(4+4 Punkte)

Gegeben seien zwei Relationen, P mit 100 Plattenzugriffen
 S 50

- (a) Geben Sie in Pseudocode an, wie ein Equi-Join von R und S bezüglich des Attributes A mit Hilfe des Algorithmus NBL-Join gebildet wird.

1. Variante für Cashstrategie (andere Varianten auch OK)

Für alle Blöcke S_i aus S

{ Für alle Blöcke R_i aus R

{ Für alle Tupel s aus S_i

{ Für alle Tupel r aus R_i

{ Falls (Joinbedingung erfüllt)

{ $Q^* = Q^* \cup (r \cup s)$

{

}

}

}

}

Die äußere Relation
soll die kleinere sein
also S soll äußere Rel.

- (b) Wieviele Blockzugriffe sind bei der Berechnung des Joins von R und S mit Algorithmus NBL-Join mindestens nötig, wenn nur jeweils genau zwei Datendobjekte im Speicher gehalten werden können.

Gemerkt war:

mit der Strategie
aus (a).

1. Cash-Strategie

Die äußere Relation

Formel: $B_R + B_R \cdot B_S$

soll die kleinere sein!

Deshalb:

Blockzugriffe: $50 + 50 \cdot 100 = \underline{\underline{5050}}$