

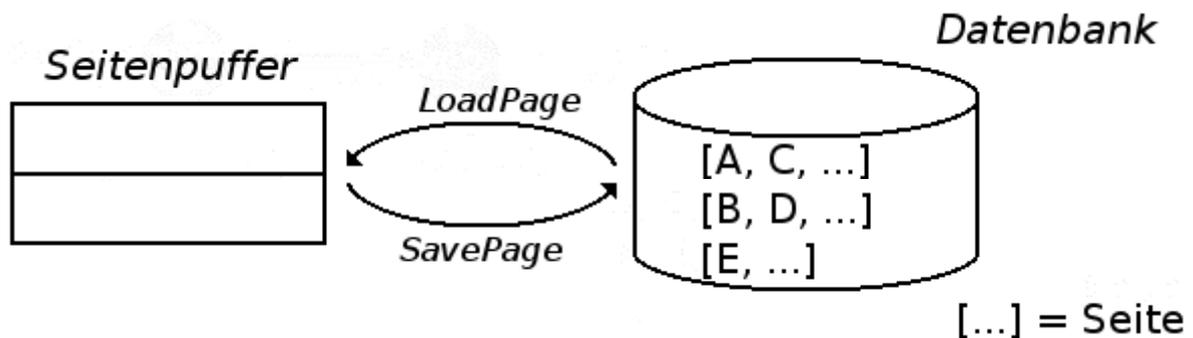
Datenbanksysteme II
SS 2007

Übungsblatt 3: Logging und Recovery

Besprechung: 30.07.2007

Aufgabe 3-1 Verdrängungs- und Ausschreibestrategien

Betrachten Sie das folgende Beispiel, bei dem die Objekte *A* und *C* sowie *B* und *D* jeweils auf derselben Seite in der Datenbank abgelegt sind. Im folgenden nehmen wir die Einbringstrategie *Update-in-Place* an.



Gegeben sei der folgende Schedule:

$S = r_1(B), r_2(C), w_2(C), w_1(B), r_1(E), w_1(E), EOT_2, r_1(A), w_1(A), EOT_1, r_3(D), w_3(D), EOT_3$

- Erläutern Sie, warum sich die Kombination *Force / No-Steal* für das direkte Einbringen (*Update-in-Place*) nicht realisieren lässt.
- Beschreiben Sie eine mögliche Abfolge von *loadPage*- und *savePage*-Operationen zwischen der Datenbank und dem *unbeschränkten* Seitenpuffer im Hauptspeicher. Dabei soll die Verdrängungsstrategie *No-Steal* benutzt werden. Geben Sie an, welche Ausschreibestrategie Sie verwenden
- Welche Auswirkungen ergeben sich, wenn Sie die Verdrängungsstrategie *Steal* benutzen und der Seitenpuffer eine *beschränkte Kapazität* von zwei Seiten hat? Geben Sie an, welche Ausschreibestrategie Sie verwenden.

Aufgabe 3-1 b)

Einbringstrategie:

Strategie Pufferverwaltung:

	<i>Aktion</i>	<i>Load/Save</i>	<i>Puffer</i>	<i>DB-Inhalt</i>
0				
1	$r_1(B)$			
2	$r_2(C)$			
3	$w_2(C)$			
4	$w_1(B)$			
5	$r_1(E)$			
6	$w_1(E)$			
7	EOT_2			
8	$r_1(A)$			
9	$w_1(A)$			
10	EOT_1			
11	$r_3(D)$			
12	$w_3(D)$			
13	EOT_3			

Aufgabe 3-1 c)

Einbringstrategie:

Strategie Pufferverwaltung:

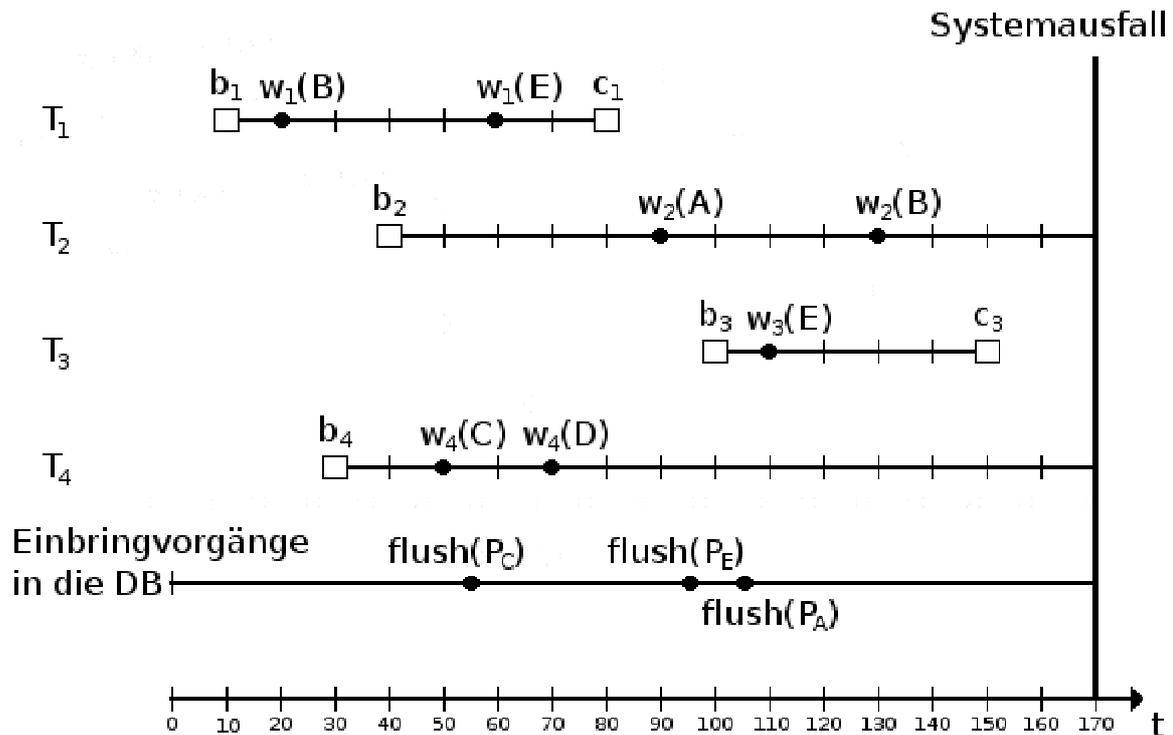
	<i>Aktion</i>	<i>Load/Save</i>	<i>Puffer</i>	<i>DB-Inhalt</i>
1-4	...			
5	$r_1(E)$			
6	$w_1(E)$			
7	EOT_2			
8	$r_1(A)$			
9	$w_1(A)$			
10	EOT_1			
11	$r_3(D)$			
12	$w_3(D)$			
13	EOT_3			

Aufgabe 3-2 *WAL-Prinzip und Commit-Regel*

Erläutern Sie anhand des Beispiels aus Aufgabe 3-1 die Probleme, die bei Verletzung des WAL-Prinzips oder der Commit-Regel auftreten können.

Aufgabe 3-3 Normalablauf

Gegeben sei ein DBMS, das die parallel laufenden Transaktionen T_1 , T_2 , T_3 und T_4 verwaltet. Dabei ändert Transaktion T_1 die Objekte B und E , T_2 die Objekte A und B , T_3 das Objekt E und schließlich Transaktion T_4 die Objekte C und D . Dabei sei das Objekt $X \in \{A, B, C, D, E\}$ jeweils in der Seite P_X enthalten. Die modifizierten Seiten P_C , P_E und P_A werden durch die jeweilige Operation $\text{flush}(P_Y)$ ($Y \in \{C, E, A\}$) zum Zeitpunkt $t = 55$, $t = 95$ bzw. $t = 105$ aus dem DB-Puffer verdrängt und in die DB eingebracht. Die Operationen b_i bzw. c_i markieren Transaktionsbeginn (BOT) bzw. Commit (EOT) der jeweiligen Transaktion T_i .



Das DBMS setzt ein physiologisches Logging-Verfahren ein und benutzt dabei Non-Atomic, Steal und No-Force als Einbring-, Seitenersetzungs- bzw. Ausschreibestrategie. Es werden indirekte Sicherungspunkte benutzt, der letzte vor dem Systemausfall war zum Zeitpunkt 0 schon abgeschlossen (dabei wurden alle Seiten erfolgreich ausgeschrieben). Während der Ausführung der Transaktionen werden keinerlei Sicherungspunkte gesetzt. Zu Beginn hat die Log-Sequence-Number LSN den Wert 0 und alle Page-LSN werden ebenfalls auf 0 gesetzt. Hat ein Log-Eintrag keinen Vorgänger, so wird prevLSN entsprechend mit 0 initialisiert.

- (a) Führen Sie die in der Abbildung gezeigten Aktionen der Transaktionen nacheinander durch und vervollständigen Sie dabei die Tabelle auf der nächsten Seite.

Benutzen Sie dabei folgende Notation für die Log-Information:

- $R(X)$: Redo-Information für Objekt X
- $U(X)$: Undo-Information für Objekt X
- BOT : Transaktionsbeginn
- EOT : Transaktionsende

- (b) Wie sieht der Inhalt der Datenbank zum Zeitpunkt des Systemfehlers aus?

Aufgabe 3-3 a)

Zeit (LSN)	Aktion	Änderung im DB-Puffer (Page-Id, LSN) (Hauptspeicher)	Änderung in DB (Page-Id, LSN) (Platte)	Log-Eintrag im Log-Puffer (LSN, TA-Id, Page-Id, R(X), U(X), PrevLSN) bzw. (LSN, TA-Id, BOT/EOT, PrevLSN) (Hauptspeicher)	Zur Log-Datei hinzugefügte Log-Einträge (Platte)
10	b_1			10, T_1 , BOT, 0	
20	$w_1(B)$	$P_B, 20$		20, T_1 , R(B), U(B), 10	
30	b_4			30, T_4 , BOT, 0	
40	b_2			40, T_2 , BOT, 0	
50					
55					
60					
70					
80					
90					
95					
100					
105					
110					
130					
150					

Aufgabe 3-4 *Ablauf der Crash-Recovery*

- (a) Aus welchen Phasen besteht die Crash-Recovery nach dem Systemfehler?
- (b) Führen Sie mit Hilfe der Log-Datei aus Aufgabe 3-3 die Crash-Recovery durch.
Benutzen Sie hierzu
- vollständiges Redo
 - selektives Redo

Begründen Sie kurz die einzelnen Recovery-Aktionen.