

Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken im SoSe20*

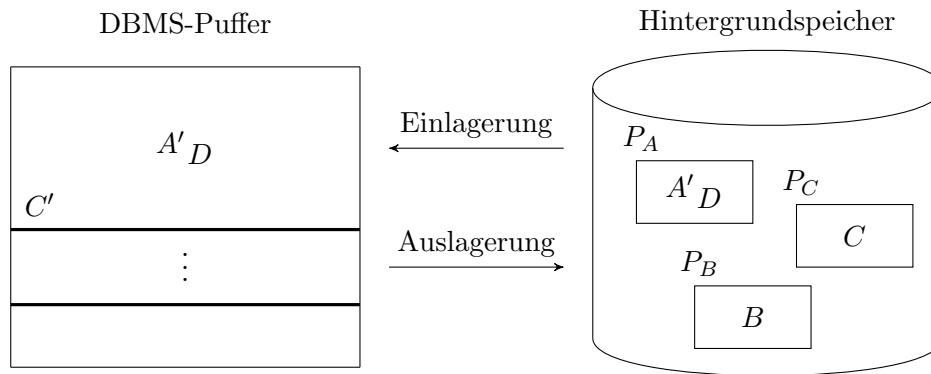
Maximilian {Bandle, Schüle}, Josef Schmeißer (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss20/impldb/>

Blatt Nr. 01

Hausaufgabe 1

Demonstrieren Sie anhand eines Beispiels, dass man die Strategien *force* und \neg *steal* nicht kombinieren kann, wenn parallele Transaktionen gleichzeitig Änderungen an Datenobjekten innerhalb einer Seite durchführen. Betrachten Sie dazu z.B. die unten dargestellte Seitenbelegung, bei der die Seite P_A die beiden Datensätze A und D enthält. Entwerfen Sie eine verzahnte Ausführung zweier Transaktionen, bei der eine Kombination aus *force* und \neg *steal* ausgeschlossen ist.



Hausaufgabe 2

Überlegen Sie sich, bei welcher Seitenersetzungsstrategie bei einem Wiederanlauf eine *Redo*- bzw. eine *Undo*-Phase notwendig ist. Verwenden Sie in diesem Zusammenhang den Begriff *dirty*. Welche der beiden Phasen entfällt bei einer Hauptspeicherdatenbank?

Hausaufgabe 3

In Abbildung 1 ist die verzahnte Ausführung der beiden Transaktionen T_1 und T_2 und das zugehörige *Log* auf der Basis logischer Protokollierung gezeigt. Wie sähe das *Log* bei physischer Protokollierung aus, wenn die Datenobjekte A , B und C die Initialwerte 1000, 2000 und 3000 hätten?

Hausaufgabe 4

Leider erhalten wir einen Fehler mit Hauptspeicherverlust der in Abbildung 1 gezeigten Ausführung nach Schritt 13. Welche Transaktion ist ein *Winner*, welche ein *Loser*? Geben Sie alle nötigen Kompensations-Rekord (CLR) an.

Hausaufgabe 5

Warum ist es für die Erzielung der Idempotenz der Redo-Phase notwendig, die – und nur die – LSN einer tatsächlich durchgeführten Redo-Operation in der betreffenden Seite zu

Schritt	T_1	T_2	Log
			[LSN, TA, PageID, Redo, Undo, PrevLSN]
1.	BOT		[#1, T_1 , BOT , 0]
2.	$r(A, a_1)$		
3.		BOT	[#2, T_2 , BOT , 0]
4.		$r(C, c_2)$	
5.	$a_1 := a_1 - 50$		
6.	$w(A, a_1)$		[#3, T_1 , P_A , $A-=50$, $A+=50$, #1]
7.		$c_2 := c_2 + 100$	
8.		$w(C, c_2)$	[#4, T_2 , P_C , $C+=100$, $C-=100$, #2]
9.	$r(B, b_1)$		
10.	$b_1 := b_1 + 50$		
11.	$w(B, b_1)$		[#5, T_1 , P_B , $B+=50$, $B-=50$, #3]
12.	commit		[#6, T_1 , commit , #5]
13.		$r(A, a_2)$	
14.		$a_2 := a_2 - 100$	
15.		$w(A, a_2)$	[#7, T_2 , P_A , $A-=100$, $A+=100$, #4]
16.		commit	[#8, T_2 , commit , #7]

Abbildung 1: Verzahnte Ausführung zweier Transaktionen und das erstellte Log

vermerken? Zeigen Sie für die folgenden Szenarien anhand von Beispielen mit logischer Protokollierung, dass die Idempotenz nicht sichergestellt werden kann.

- a) LSN-Einträge werden in der Redo-Phase nicht auf Datenseiten geschrieben.
- b) LSN-Einträge von Log-Records, für die die Redo-Operation nicht ausgeführt wird, werden trotzdem in die Datenseiten übertragen.

Beantworten Sie außerdem folgende Frage:

- c) Wie wird die Idempotenz der Undo-Phase sichergestellt, wenn ein Kompensations-eintrag geschrieben wurde und dann noch vor der Ausführung des Undo das Datenbanksystem abstürzt?