Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Peer Kröger
Felix Borutta, Markus Mauder

München, 08.06.2016

Datenbanksysteme II

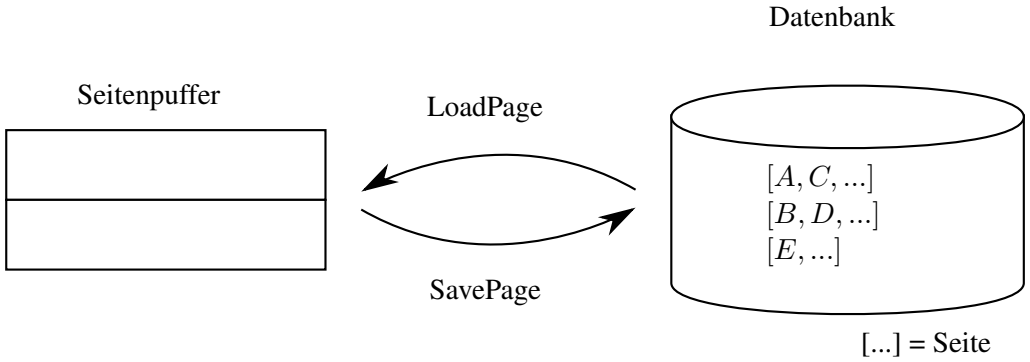
SS 2016

Übungsblatt 5: Logging

Besprechung: 13. – 15.06.2016

Aufgabe 5-1 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:

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

a) Erläutern Sie, warum sich die Kombination *Force* / *No-Steal* für das direkte Einbringen (*Update-in-Place*) nicht realisieren lässt.

b) 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.

c) 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 5-2 WAL-Prinzip und Commit-Regel

Erläutern Sie anhand des Beispiels aus Aufgabe 5-1 die Probleme, die bei Verletzung des WAL-Prinzips oder der Commit-Regel auftreten können.

Aufgabe 5-1 b)

Einbringstrategie:

Strategie Pufferverwaltung:

	Aktion	*Load/Save*	*Puffer*	*DB-Inhalt*
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 5-1 c)

Einbringstrategie:

Strategie Pufferverwaltung:

	Aktion	*Load/Save*	*Puffer*	*DB-Inhalt*
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			