GOETHE UNIVERSITÄT FRANKFURT AM MAIN

www.uni-frankfurt.de

Vorlesung Datenbanksysteme 2

Übung – Recovery Checkpointing

Aufgabe – Checkpointing

Das Datenbanksystem habe für folgenden Schedule ***zwei LRU-Cache-Slots*** reserviert. Der Recovery Manager unterstützt ***Undo/No Redo***.

a) Es soll ***Cache Consistent Checkpointing*** verwendet werden.

b) Es soll ***Commit Consistent Checkpointing*** verwendet werden.

Beschreiben Sie die Vorgänge bei der Ausführung des Schedules, indem Sie eine Tabelle vom Typ (1) und eine Restart-Tabelle vom Typ (2) verwenden.

T1	T2	T3	T4
	READ(A) A:=A+5 WRITE(A) COMMIT		
A:=3 WRITE(A) READ(B) B:=B+5			
Checkpoint (1)			
		READ(C) C:=C+5 WRITE(C)	
WRITE(B) COMMIT			
		ABORT	
Checkpoint (2)			
			READ(C) C:=C+5 WRITE(C)
☹ SYSTEMCRASH			

Cache Consistent Checkpointing

- aktive Transaktionen werden angehalten (blocked).
- alle „dirty“ Cache-Slots in stabile Datenbasis schreiben (flushed)
- Markierung in Logdatei setzten
- Der Restart muss nur die Updates von committed Transaktionen wiederherstellen, die im Log ***nach dem letzten Checkpoint*** auftreten.
- Der Restart muss nur die Updates von den Transaktionen rückgängig machen, die in der Aktiv- aber nicht in der Commit-Liste sind, oder in der Abort-Liste sind und nach dem letzten Checkpoint-Marker in dieser Liste stehen.

Cache Consistent Checkpointing

Tabelle Typ (1)											
Operation	**Logeintrag**	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**		
		X	V	X	V	A	B	C	Aktiv	Commit	Abort
(Initialisierung)	[T0,A,5],[T0,B,10], [T0,C,1]	-	-	-	-	5	10	1	-	T0	-
Stand beim Crash											

Restart-Tabelle vom Typ (2)

Listen: Aktiv={T4}, Commit={T0,T2,T1}, Abort={T3}

Tabelle Typ (2)									
Logeintrag	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**	
	X	V	X	V	A	B	C	undone	redone
Vor Recovery	-	-	-	-				---	---
Nach Flush	-	-	-	-				---	---

Ausführung - Tabelle vom Typ (1)

Tabelle Typ (1)											
Operation	**Logeintrag**	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**		
		X	V	X	V	A	B	C	Aktiv	Commit	Abort
(Initialisierung)	[T0,A,5],[T0,B,10], [T0,C,1]	-	-	-	-	5	10	1	-	T0	-
READ$_2$(A)		**A**	**5**	**-**	**-**	**5**	**10**	**1**	**-**	**T0**	**-**
WRITE$_2$(A)	**[T2,A,10]**	**A***	**10**	**-**	**-**	**5**	**10**	**1**	**T2**	**T0**	**-**
FLUSH(A)		**A**	**10**	**-**	**-**	**<10>**	**10**	**1**	**T2**	**T0**	**-**
COMMIT$_2$		**A**	**10**	**-**	**-**	**10**	**10**	**1**	**-**	**T0,T2**	**-**
WRITE$_1$(A)	**[T1,A,3]**	**A***	**3**	**-**	**-**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
READ$_1$(B)		**A***	**3**	**B**	**10**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
CheckPoint (1)	**[CP1]**	**A**	**3**	**B**	**10**	**<3>**	**10**	**1**	**T1**	**T0,T2**	**-**
READ$_3$(C)		**C**	**1**	**B**	**10**	**<3>**	**10**	**1**	**T1**	**T0,T2**	**-**
WRITE$_3$(C)	**[T3,C,6]**	**C***	**6**	**B**	**10**	**<3>**	**10**	**1**	**T1,T3**	**T0,T2**	**-**
WRITE$_1$(B)	**[T1,B,15]**	**C***	**6**	**B***	**15**	**<3>**	**10**	**1**	**T1,T3**	**T0,T2**	**-**
FLUSH(B)		**C***	**6**	**B**	**15**	**<3>**	**<15>**	**1**	**T1,T3**	**T0,T2**	**-**
COMMIT$_1$		**C***	**6**	**B**	**15**	**3**	**15**	**1**	**T3**	**T0,T2,T1**	**-**
ABORT$_3$		**C***	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**T3**
CheckPoint (2)	**[CP2]**	**C**	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**T3**
READ$_4$(C)		**C**	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**T3**
WRITE$_4$(C)	**[T4,C,6]**	**C***	**6**	**B**	**15**	**3**	**15**	**1**	**T4**	**T0,T2,T1**	**T3**
Stand beim Crash						3	15	1			

Restart-Tabelle vom Typ (2)

Listen: Aktiv={T4}, Commit={T0,T2,T1}, Abort={T3}

Tabelle Typ (2)									
Logeintrag	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**	
	X	V	X	V	A	B	C	undone	redone
Vor Recovery	-	-	-	-	3	15	1	---	---
[T4,C,6]	**C***	**1**	**-**	**-**	**3**	**15**	**1**	**C**	**---**
Nach Flush	-	-	-	-	3	15	1	---	---

Commit Consistent Checkpointing

- warten bis alle gerade aktive Transaktionen T_i entweder committed sind oder abgebrochen wurden
- keine neuen Transaktionen zulassen (bis Checkpoint End)
- alle „dirty“ Cache-Slots in stabile Datenbasis schreiben(fluschen)
- Markierung in Logdatei setzten
- Redo/Undo nur für Transaktionen, die ***nach letzten Checkpoint aktiv waren***

Commit Consistent Checkpointing

Tabelle Typ (1)											
Operation	**Logeintrag**	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**		
		X	V	X	V	A	B	C	Aktiv	Commit	Abort
(Initialisierung)	[T0,A,5],[T0,B,10], [T0,C,1]	-	-	-	-	5	10	1	-	T0	-
Stand beim Crash											

Restart-Tabelle vom Typ (2)

Listen: Aktiv={T4}, Commit={T0,T2,T1}, Abort={T3}

Tabelle Typ (2)									
Logeintrag	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**	
	X	V	X	V	A	B	C	undone	redone
Vor Recovery	-	-	-	-				---	---
Nach Flush	-	-	-	-				---	---

Ausführung - Tabelle vom Typ (1)

Tabelle Typ (1)											
Operation	**Logeintrag**	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**		
		X	V	X	V	A	B	C	Aktiv	Commit	Abort
(Initialisierung)	[T0,A,5],[T0,B,10], [T0,C,1]	-	-	-	-	5	10	1	-	T0	-
READ$_2$(A)		**A**	**5**	**-**	**-**	**5**	**10**	**1**	**-**	**T0**	**-**
WRITE$_2$(A)	**[T2,A,10]**	**A***	**10**	**-**	**-**	**5**	**10**	**1**	**T2**	**T0**	**-**
FLUSH(A)		**A**	**10**	**-**	**-**	**<10>**	**10**	**1**	**T2**	**T0**	**-**
COMMIT$_2$		**A**	**10**	**-**	**-**	**10**	**10**	**1**	**-**	**T0,T2**	**-**
WRITE$_1$(A)	**[T1,A,3]**	**A***	**3**	**-**	**-**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
READ$_1$(B)		**A***	**3**	**B**	**10**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
CheckPoint (1)	**Warte auf T1!**	**A***	**3**	**B**	**10**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
READ$_3$(C)	**T3 wird gehalten!**	**A***	**3**	**B**	**10**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
WRITE$_1$(B)	**[T1,B,15]**	**A***	**3**	**B***	**15**	**10**	**10**	**1**	**T1**	**T0,T2**	**-**
FLUSH(A)		**A**	**3**	**B***	**15**	**<3>**	**10**	**1**	**T1**	**T0,T2**	**-**
FLUSH(B)		**A**	**3**	**B**	**15**	**<3>**	**<15>**	**1**	**T1**	**T0,T2**	**-**
COMMIT$_1$		**A**	**3**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**-**
CheckPoint (1)	**[CP1]**	**A**	**3**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**-**
READ$_3$(C)		**C**	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**-**
WRITE$_3$(C)	**[T3,C,6]**	**C***	**6**	**B**	**15**	**3**	**15**	**1**	**T3**	**T0,T2,T1**	**-**
ABORT$_3$		**C***	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**T3**
CheckPoint (2)	**[CP2]**	**C**	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**T3**
READ$_4$(C)		**C**	**1**	**B**	**15**	**3**	**15**	**1**	**-**	**T0,T2,T1**	**T3**
WRITE$_4$(C)	**[T4,C,6]**	**C***	**6**	**B**	**15**	**3**	**15**	**1**	**T4**	**T0,T2,T1**	**T3**
Stand beim Crash						3	15	1			

Restart-Tabelle vom Typ (2)

Listen: Aktiv={T4}, Commit={T0,T2,T1}, Abort={T3}

Tabelle Typ (2)									
Logeintrag	**Slot 1**		**Slot 2**		**Datenbasis**			**Listen**	
	X	V	X	V	A	B	C	undone	redone
Vor Recovery	-	-	-	-	3	15	1	---	---
[T4,C,6]	**C***	**1**	**-**	**-**	**3**	**15**	**1**	**C**	**---**
Nach Flush	-	-	-	-	3	15	1	---	---