

Übungsblatt Recovery

Aufgabe 1: Recovery, Checkpointing

Das Datenbanksystem habe für folgenden Schedule zwei LRU-Cache-Slots reserviert (Undo/No Redo Strategie).

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			

- a) Es soll *Cache Consistent Checkpointing* verwendet werden.
 - i) Beschreiben Sie die Vorgänge bei der Ausführung des Schedules, indem Sie eine Tabelle vom Typ (1) (siehe unten) verwenden.
 - ii) Beschreiben Sie anschließend den Restart-Vorgang. Benutzen Sie hierzu eine Restart-Tabelle vom Typ (2) (siehe unten).
- b) Es soll *Commit Consistent Checkpointing* verwendet werden.
 - i) Beschreiben Sie die Vorgänge bei der Ausführung des Schedules, indem Sie eine Tabelle vom Typ (1) (siehe unten) verwenden.
 - ii) Beschreiben Sie anschließend den Restart-Vorgang. Benutzen Sie hierzu eine Restart-Tabelle vom Typ (2) (siehe unten).

Lösung 1)a)i) Cache Consistent Checkpointing, Normalbetrieb

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				

Lösung 1)a)ii) Cache Consistent Checkpointing, Restart

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	---	---

Lösung 1)b)i) Commit Consistent Checkpointing, Normalbetrieb

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				

Lösung 1)b)ii) Commit Consistent Checkpointing, Restart

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	---	---

Aufgabe 2: Recovery Checkpointing

Das Datenbanksystem habe für dieses Schedule drei LRU-Cache-Slots reserviert. Der Recovery Manager verwendet den *Undo/No Redo* Algorithmus. Es soll Commit Consistent Checkpointing verwendet werden.

	T1	T2	T3
1		READ (B)	
2		WRITE (B,7)	
3	WRITE (C,8)		
4	Commit Consistent Checkpoint		
5			WRITE (D,9)
6		WRITE (E, 8)	
7	READ (A)		
8	COMMIT		
9			COMMIT
10		COMMIT	

Kennzeichnen Sie in den Tabellen einen dirty Cache durch ein * und einen unbestätigten Wert in der Datenbasis durch Einklammern z.B. (<WERT>).

Lösung Aufgabe 2:

Tabelle Typ (1)															
Operation	Logeintrag	Slot 1		Slot 2		Slot 3		Datenbasis					Listen		
		X	V	X	V	X	V	A	B	C	D	E	Aktiv	Commit	Abort
(Init.)		-	-	-	-	-	-	1	2	3	4	5	-	T0	-
READ ₂ (B)		B	2	-	-	-	-	1	2	3	4	5	-	T0	-
WRITE ₂ (B)	[T ₂ , B, 7]	B*	7	-	-	-	-	1	2	3	4	5	T2	T0	-
WRITE ₁ (C)	[T ₁ , C, 8]	B*	7	C*	8	-	-	1	2	3	4	5	T2,T1	T0	-
CheckPoint 1	Warte auf T1 und T2!	B*	7	C*	8	-	-	1	2	3	4	5	T2,T1	T0	-
WRITE ₃ (D)	T3 wird gehalten!	B*	7	C*	8	-	-	1	2	3	4	5	T2,T1	T0	-
WRITE ₂ (E)	[T ₂ , E, 8]	B*	7	C*	8	E*	8	1	2	3	4	5	T2,T1	T0	-
READ ₁ (A)		A	1	C*	8	E*	8	1	<7>	3	4	5	T2,T1	T0	-
FLUSH(C)		A	1	C	8	E*	8	1	<7>	<8>	4	5	T2,T1	T0	-
COMMIT ₁		A	1	C	8	E*	8	1	<7>	8	4	5	T2	T0,T1	-
FLUSH(E)		A	1	C	8	E	8	1	<7>	8	4	<8>	T2	T0,T1	-
COMMIT ₂		A	1	C	8	E	8	1	7	8	4	8	-	T0,T1,T2	-
CheckPoint 1	[CP1]	A	1	C	8	E	8	1	7	8	4	8	-	T0,T1,T2	-
WRITE ₃ (D)	[T ₃ , D, 9]	A	1	D*	9	E	8	1	7	8	4	8	T3	T0,T1,T2	-
FLUSH(D)		A	1	D	9	E	8	1	7	8	<9>	8	T3	T0,T1,T2	-
COMMIT ₃		A	1	D	9	E	8	1	7	8	9	8	-	T0,T1,T2,T3	-