

Vorlesung Datenbanksysteme 2

Übung – Recovery Garbage
Collection

Aufgabe

Die Variablen in der Datenbank seien wie folgt vorgegeben: A=30, B=15, C=40 und D=20. Alle Variablen sind von T0 Initialisiert eingetragen.

- Warum ist Garbage Collection notwendig?
- Stellen Sie das Log-File sowie die Werte der Datenbank nach jedem Schritt für die UNDO-Strategie des Recovery-Managers dar. Wie wirken sich die Garbage Collections auf das Log-File aus?

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
...								
Stand beim Crash								

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊖ SYSTEMCRASH		

Aufgabe

a) Was ist das Ziel von Garbage Collection?

Um das Wachstum der Log-Datei(en) zu beschränken und Speicherplatz zu sparen.

Bei der **Garbage Collection** werden Logeinträge gelöscht, die nicht (mehr) für die Restart-Prozedur benötigt werden.

Garbage Collection Regel:

Ein Eintrag $[T_i, x, v]$ kann aus dem Log entfernt (removed) werden, wenn:

1. T_i abgebrochen wurde (Abort), oder
2. T_i committed hat, aber eine andere committed Transaktion einen Wert nach x geschrieben hat, nachdem T_i dies tat.
(v also nicht der letzte committed Wert von x ist)
3. wenn der RM kein Undo benötigt, v der letzte committed Wert von x ist, v der Wert von x in der stable Database ist und $[T_i, x, v]$ der einzige Eintrag im Log für x ist.

Undo/Redo-Strategie Log-File

Tabelle								
Operation	Logeintrag	Datenbasis				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		

UNDO-Strategie Log-File (1)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		

UNDO-Strategie Log-File (2)

Regel 2: T_i committed hat, aber eine andere committed Transaktion einen Wert nach x geschrieben hat, nachdem T_i dies tat.
(v also nicht der letzte committed Wert von x ist)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Värenderung des Logs!								

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		

UNDO-Strategie Log-File (3)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Vänderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---

T1	T2	T3
		READ (C) C := C + 42 WRITE (C)
Garbage Collection (1)		
	READ (B) B := 17 WRITE (B)	
READ (A) READ (D) D := 25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D := 26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊕ SYSTEMCRASH		

UNDO-Strategie Log-File (4)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Vänderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---

T1	T2	T3
		READ (C) C := C + 42 WRITE (C)
Garbage Collection (1)		
	READ (B) B := 17 WRITE (B)	
READ (A) READ (D) D := 25 WRITE (D)		
Garbage Collection (2)		
	READ (D) D := 26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		

UNDO-Strategie Log-File (5)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Vänderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---
COMMIT		30	17	82	25	T2,T3	T0,T1	---

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D)		
COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		

UNDO-Strategie Log-File (6)

Regel 2: T_i committed hat, aber eine andere committed Transaktion einen Wert nach x geschrieben hat, nachdem T_i dies tat.
(v also nicht der letzte committed Wert von x ist)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Vänderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---
COMMIT		30	17	82	25	T2,T3	T0,T1	---
GC(2) → Lösche Logeintrag [T0,D,20] !								

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊕ SYSTEMCRASH		

UNDO-Strategie Log-File (7)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40],[T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Vänderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---
COMMIT		30	17	82	25	T2,T3	T0,T1	---
GC(2) → Lösche Logeintrag [T0,D,20] !								
WRITE (D)	[T2,D,26]	30	17	82	26	T2,T3	T0,T1	---

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊕ SYSTEMCRASH		

UNDO-Strategie Log-File (8)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30],[T0,B,15], [T0,C,40], [T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Värenderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---
COMMIT		30	17	82	25	T2,T3	T0,T1	---
GC(2) → Lösche Logeintrag [T0,D,20] !								
WRITE (D)	[T2,D,26]	30	17	82	26	T2,T3	T0,T1	---
COMMIT		30	17	82	26	T3	T0,T1,T2	---

T1	T2	T3
		READ (C) C := C + 42 WRITE (C)
Garbage Collection (1)		
	READ (B) B := 17 WRITE (B)	
READ (A) READ (D) D := 25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D := 26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		

UNDO-Strategie Log-File (9)

Regel 2: T_i committed hat, aber eine andere committed Transaktion einen Wert nach x geschrieben hat, nachdem T_i dies tat.
(v also nicht der letzte committed Wert von x ist)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30], [T0,B,15] , [T0,C,40], [T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Värenderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---
COMMIT		30	17	82	25	T2,T3	T0,T1	---
GC(2) → Lösche Logeintrag [T0,D,20] !								
WRITE (D)	[T2,D,26]	30	17	82	26	T2,T3	T0,T1	---
COMMIT		30	17	82	26	T3	T0,T1,T2	---
GC(3) → Lösche Logeintrag [T0,B,15] & [T1,D,25]!								

T1	T2	T3
		READ (C) C := C + 42 WRITE (C)
Garbage Collection (1)		
	READ (B) B := 17 WRITE (B)	
READ (A) READ (D) D := 25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D := 26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊕ SYSTEMCRASH		

UNDO-Strategie Log-File (10)

Tabelle								
Operation	Logeintrag	Variablen				Listen		
		A	B	C	D	Aktiv	Commit	Abort
(Init.)	[T0,A,30], [T0,B,15] , [T0,C,40], [T0,D,20]	30	15	40	20	---	T0	---
WRITE (C)	[T3,C,82]	30	15	82	20	T3	T0	---
GC(1) → Keine Värenderung des Logs!								
WRITE (B)	[T2,B,17]	30	17	82	20	T2,T3	T0	---
WRITE (D)	[T1,D,25]	30	17	82	25	T1,T2,T3	T0	---
COMMIT		30	17	82	25	T2,T3	T0,T1	---
GC(2) → Lösche Logeintrag [T0,D,20] !								
WRITE (D)	[T2,D,26]	30	17	82	26	T2,T3	T0,T1	---
COMMIT		30	17	82	26	T3	T0,T1,T2	---
GC(3) → Lösche Logeintrag [T0,B,15] & [T1,D,25]!								

T1	T2	T3
		READ (C) C:=C+42 WRITE (C)
Garbage Collection (1)		
	READ (B) B:=17 WRITE (B)	
READ (A) READ (D) D:=25 WRITE (D) COMMIT		
Garbage Collection (2)		
	READ (D) D:=26 WRITE (D) COMMIT	
Garbage Collection (3)		
		READ (A)
⊗ SYSTEMCRASH		