Datenbanksysteme

VU 184.686, WS 2022

Recovery

Johannes Fichte, Felix Winter

Institut für Logic and Computation
Technische Universität Wien

Acknowledgments

Die Folien sind eine Weiterentwicklung der Folien von Reinhard Pichler und Sebastian Skritek.

Der Inhalt basiert auf und behandelt Kapitel 10 des Lehrbuchs
(Kemper, Eickler: Datenbanksysteme – Eine Einführung).
Die meisten Beispiele und Abbildungen sind von dort entnommen.

Recovery (Fehlerbehandlung und Logging)

1. Speicherverwaltung
2. Fehlerkategorien
3. Angenommene Systemkonfiguration
4. Protokollierung (Logging)
5. Wiederanlauf nach einem Fehler (Recovery)
6. Sicherungspunkte (Checkpoints)

Inhalt

Inhalt

Speicherhierarchie

Verfügbare Speichermedien bilden typischerweise Hierarchie

- Tradeoff:
  - Schnell und teuer vs. langsam und billig
  - Schnell und flüchtig vs. langsam und persistent
  - Schnell und klein vs. langsam und groß
- "Zugriffslücke" zwischen den einzelnen Stufen

Typ	Zeit [s]	Preis [EUR/TB]
Register	$\sim 2 \cdot 10^{-10}$	
Cache	$\sim 5 \cdot 10^{-10}$	
DRAM	$\sim 1 \cdot 10^{-8}$	5600
Flash	$\sim 1 \cdot 10^{-5}$	100
Disk	$\sim 3 \cdot 10^{-3}$	20

Tabelle: Größenordnung Zugriffszeiten

Speicherhierarchie

Verfügbare Speichermedien bilden typischerweise Hierarchie

- Tradeoff:
  - Schnell und teuer vs. langsam und billig
  - Schnell und flüchtig vs. langsam und persistent
  - Schnell und klein vs. langsam und groß
- "Zugriffslücke" zwischen den einzelnen Stufen

(Vereinfachte) Annahme in LVA: Zweistufige Speicherhierarchie

- Datenbankpuffer (schneller Zugriff, flüchtig)
- Hintergrundspeicher (langsamer Zugriff, persistent)

Typ	Zeit [s]	Preis [EUR/TB]
Register	$\sim 2 \cdot 10^{-10}$	
Cache	$\sim 5 \cdot 10^{-10}$	
DRAM	$\sim 1 \cdot 10^{-8}$	5600
Flash	$\sim 1 \cdot 10^{-5}$	100
Disk	$\sim 3 \cdot 10^{-3}$	20

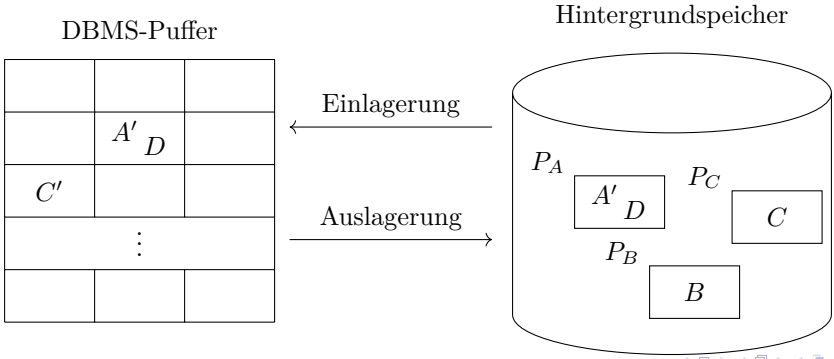
Tabelle: Größenordnung Zugriffszeiten

Speicherorganisation: Seiten

- Speicher unterteilt in Seiten (engl. Pages)
- Datensätze werden auf Seiten abgebildet
  - Vereinfachte Annahme in LVA: Datum A auf Seite P_A
- Level an Granularität in dem Daten zwischen Puffer und Hintergrundspeicher verschoben werden

Speicherorganisation: Seiten

- Speicher unterteilt in Seiten (engl. Pages)
- Datensätze werden auf Seiten abgebildet
  - Vereinfachte Annahme in LVA: Datum A auf Seite P_A
- Level an Granularität in dem Daten zwischen Puffer und Hintergrundspeicher verschoben werden



Inhalt

Hintergrundspeicher: Strategie zum Einlagern in Puffer

Update-in-Place ("direkte" Einbringstrategie):

- Jede Seite hat genau einen "Platz" im Hintergrundspeicher
- Beim Auslagern wird eine Seite auf diesen Platz kopiert
- Alter Zustand der Seite wird überschrieben

Hintergrundspeicher: Strategie zum Einlagern in Puffer

Update-in-Place ("direkte" Einbringstrategie):

- Jede Seite hat genau einen "Platz" im Hintergrundspeicher
- Beim Auslagern wird eine Seite auf diesen Platz kopiert
- Alter Zustand der Seite wird überschrieben

Twin-Block-Verfahren ("indirekte" Einbringstrategie):

- Für jede Seite zwei Kopien im Hintergrundspeicher + Bit das angibt welche Kopie aktuell ist.
- Beim Auslagern wird Seite auf aktuelle Kopie geschrieben

Hintergrundspeicher: Strategie zum Einlagern in Puffer

Update-in-Place ("direkte" Einbringstrategie):

- Jede Seite hat genau einen "Platz" im Hintergrundspeicher
- Beim Auslagern wird eine Seite auf diesen Platz kopiert
- Alter Zustand der Seite wird überschrieben

Twin-Block-Verfahren ("indirekte" Einbringstrategie):

- Für jede Seite zwei Kopien im Hintergrundspeicher + Bit das angibt welche Kopie aktuell ist.
- Beim Auslagern wird Seite auf aktuelle Kopie geschrieben

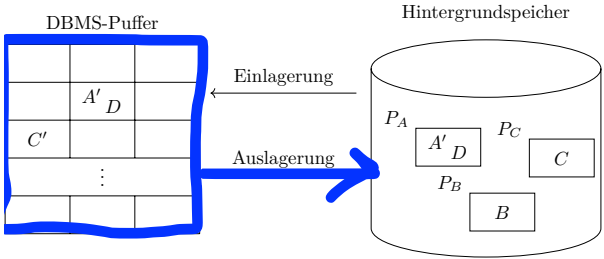
Schattenspeicherkonzept ("indirekte" Einbringstrategie):

- Nur für veränderte Seiten existieren zwei Kopien

Inhalt

Auslagerungstrategie Puffer

- Kein Auslagern (Verdrängen) während eines Zugriffs (FIX)
- Danach abhängig von gewählter Strategie
- *dirty* pages: Veränderte Seiten im Puffer



Auslagerungstrategie Puffer: Ersetzen von Seiten

(Ursache: Transaktion benötigt mehr Seiten, Lesen Puffer ← Hintergrund)

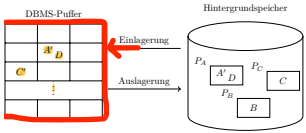
Wir unterscheiden nach der Ursache (warum ausgelagert werden muss).

Stragien zur Ersetzung von Pufferseiten:

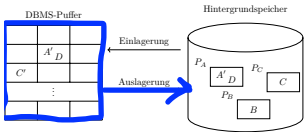
steal Jede nicht fixierte Seite ist prinzipiell ein Kandidat für die Auslagerung

nosteal Seiten die von einer noch aktiven Transaktion verändert wurden dürfen nicht ausgelagert werden.

Ursache:



Vorgang (Seite auslagern):



Auslagerungstrategie Puffer: Auslagern von Seiten (Änderungen)

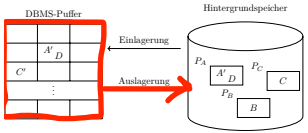
(Zeitpunkt: Transaktion fertig, Schreiben Puffer → Hintergrund)

Stragien zur Ersetzung von Pufferseiten:

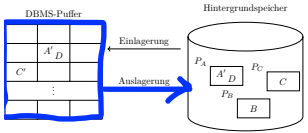
force Geänderte Seiten werden am Transaktionsende ausgelagert.

noforce Auslagerung am Ende einer Transaktion wird nicht erzwungen, *dirty pages* können im Puffer verbleiben.

Ursache:



Vorgang (Seite auslagern):



Erinnerung: Eigenschaften Transaktionen

Was wollen wir sicherstellen?

Atomicity & Durability

	force	noforce
nosteal		
steal		

Tabelle: Kombinationen von Strategien

Erinnerung: Eigenschaften Transaktionen

Was wollen wir sicherstellen?

Atomicity & Durability

Welche Strategien könnten wir anwenden?

- nosteal of Seiten?
  Problem: Schlechter Durchsatz,
  ABER atomicity trivial
- force schreiben zum Commit?
  Problem: Dateizugriff langsam,
  ABER durability trivial

	force	noforce
nosteal		
steal		

Tabelle: Kombinationen von Strategien

Erinnerung: Eigenschaften Transaktionen

Was wollen wir sicherstellen?

Atomicity & Durability

Welche Strategien könnten wir anwenden?

- nosteal of Seiten?
  Problem: Schlechter Durchsatz,
  ABER atomicity trivial
- force schreiben zum Commit?
  Problem: Dateizugriff langsam,
  ABER durability trivial

	force	noforce
nosteal	trivial/ineffektiv	
steal		

Tabelle: Kombinationen von Strategien

Erinnerung: Eigenschaften Transaktionen

Was wollen wir sicherstellen?

Atomicity & Durability

Welche Strategien könnten wir anwenden?

- nosteal of Seiten?
  Problem: Schlechter Durchsatz,
  ABER atomicity trivial
- force schreiben zum Commit?
  Problem: Dateizugriff langsam,
  ABER durability trivial

	force	noforce
nosteal	trivial/ineffektiv	
steal		schwer/schnell

Tabelle: Kombinationen von Strategien

Strategie: Steal und noforce vs. Atomicity und Durability

steal: Warum Sicherstellen von Atomicity schwer?

- *Stehlen: existierende Seite auf den Hintergrundspeicher schreiben*
- Was, wenn die Transaktion Zugriff abbricht?
- Alten Wert zum Zeitpunt des Stehlens merken für Undo

noforce: Warum Sicherstellen von Durability schwer?

- Was passiert, wenn das System abstürzt bevor die geänderte Seite auf den Hintergrundspeicher geschrieben wurde?
- Schreibe so wenig wie möglich, an einer geeigneten Stelle, zum Zeitpunkt des Commits um Redo zu ermöglichen.

Inhalt

Drei Fehlerkategorien

Lokaler Fehler (in einer Transaktion)

- Nur eine Transaktion betroffen.
- Durch diese Transaktion verursachte Änderungen müssen (schnell!) zurückgesetzt werden (Lokales Undo)

Drei Fehlerkategorien

Lokaler Fehler (in einer Transaktion)

- Nur eine Transaktion betroffen.
- Durch diese Transaktion verursachte Änderungen müssen (schnell!) zurückgesetzt werden (Lokales Undo)

Fehler mit Verlust des DBMS-Puffers

- Durch abgeschlossene Transaktionen verursachte Änderungen müssen erhalten bleiben (Globales Redo).
- Durch nicht abgeschlossene Transaktionen verursachte Änderungen müssen zurückgesetzt werden (Globales Undo).

Drei Fehlerkategorien

Lokaler Fehler (in einer Transaktion)

- Nur eine Transaktion betroffen.
- Durch diese Transaktion verursachte Änderungen müssen (schnell!) zurückgesetzt werden (Lokales Undo)

Fehler mit Verlust des DBMS-Puffers

- Durch abgeschlossene Transaktionen verursachte Änderungen müssen erhalten bleiben (Globales Redo).
- Durch nicht abgeschlossene Transaktionen verursachte Änderungen müssen zurückgesetzt werden (Globales Undo).

Fehler mit Hintergrundspeicherverlust

- Wiederherstellung der Daten mittels Archivkopie

Drei Fehlerkategorien

Lokaler Fehler (in einer Transaktion)

- Nur eine Transaktion betroffen.
- Durch diese Transaktion verursachte Änderungen müssen (schnell!) zurückgesetzt werden (Lokales Undo)

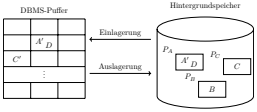
Fehler mit Verlust des DBMS-Puffers

- Durch abgeschlossene Transaktionen verursachte Änderungen müssen erhalten bleiben (Globales Redo).
- Durch nicht abgeschlossene Transaktionen verursachte Änderungen müssen zurückgesetzt werden (Globales Undo).

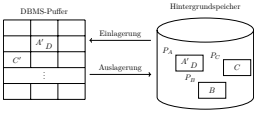
Fehler mit Hintergrundspeicherverlust

- Wiederherstellung der Daten mittels Archivkopie

Auswirkung der Stragie zur Auslagerung auf Recovery

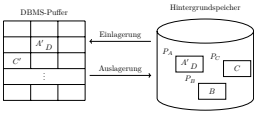


Auswirkung der Stragie zur Auslagerung auf Recovery



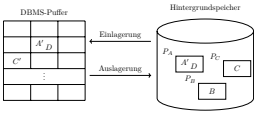
	$\rightarrow$	
	force (IMMER sofort auslagern)	noforce (Änderungen sichern, aber Seite im Puffer behalten)
$\leftarrow$ nosteal (Einlagern: nie Platz stehlen)		
steal (Einlagern: FIXierten nie Platz stehlen)		

Auswirkung der Stragie zur Auslagerung auf Recovery



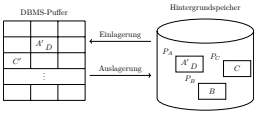
	$\rightarrow$	
	force (IMMER sofort auslagern)	noforce (Änderungen sichern, aber Seite im Puffer behalten)
$\leftarrow$ nosteal (Einlagern: nie Platz stehlen)	■ kein Redo ■ kein Undo	
steal (Einlagern: FIXierten nie Platz stehlen)		

Auswirkung der Stragie zur Auslagerung auf Recovery



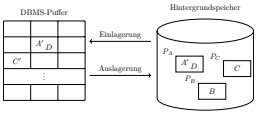
	$\rightarrow$	
	force (IMMER sofort auslagern)	noforce (Änderungen sichern, aber Seite im Puffer behalten)
$\leftarrow$ nosteal (Einlagern: nie Platz stehlen)	■ kein Redo ■ kein Undo	■ Redo ■ kein Undo
steal (Einlagern: FIXierten nie Platz stehlen)		

Auswirkung der Stragie zur Auslagerung auf Recovery



	→ force (IMMER sofort auslagern)	noforce (Änderungen sichern, aber Seite im Puffer behalten)
← nosteal (Einlagern: nie Platz stehlen)	■ kein Redo ■ kein Undo	■ Redo ■ kein Undo
steal (Einlagern: FIXierten nie Platz stehlen)	■ kein Redo ■ Undo	

Auswirkung der Stragie zur Auslagerung auf Recovery



	→ force (IMMER sofort auslagern)	noforce (Änderungen sichern, aber Seite im Puffer behalten)
← nosteal (Einlagern: nie Platz stehlen)	■ kein Redo ■ kein Undo	■ Redo ■ kein Undo
steal (Einlagern: FIXierten nie Platz stehlen)	■ kein Redo ■ Undo	■ Redo ■ Undo

Einige Nachteile von nosteal + force

- Erzwungenes Auslagern am Transaktionsende ist teuer
  - Von vielen Transaktionen benötigte Seiten (*"hot spots"*) würden ausgeschrieben obwohl sie im Puffer verbleiben.
  - Auslagern der Seiten müsste atomar geschehen ("alles oder nichts").
- Bei nosteal + force müssen Transaktionen immer gesamte Seite sperren.

Inhalt

Angenommene Systemkonfiguration

steal Auch Seiten die von einer noch nicht abgeschlossenen Transaktion modifiziert wurden können ausgelagert werden.

noforce Nach Transaktionsende müssen modifizierte Seiten nicht ausgelagert werden.

Update-In-Place Jede Seite hat genau einen Platz (eine Kopie) im Hintergrundspeicher.

Kleine Sperrgranulate auf Datensatzebene (siehe nächstes Kapitel "Mehrbenutzersynchronisation"): Verschiedene Transaktionen können gleichzeitig auf Seite zugreifen; Seite kann Änderungen von abgeschlossenen und von nicht abgeschlossenen Transaktionen enthalten.

Inhalt

Inhalt

Protokollieren von Änderungsoperationen

Problem:

- Globales Undo: Hintergrundspeicher kann Änderungen nicht abgeschlossener Transaktionen enthalten (wegen steal)
- Globales Redo: Änderungen durch abgeschlossene Transaktionen können noch nicht im Hintergrundspeicher enthalten sein (wegen noforce)

Protokollieren von Änderungsoperationen

Problem:

- Globales Undo: Hintergrundspeicher kann Änderungen nicht abgeschlossener Transaktionen enthalten (wegen steal)
- Globales Redo: Änderungen durch abgeschlossene Transaktionen können noch nicht im Hintergrundspeicher enthalten sein (wegen noforce)

Lösung: Speichere zusätzliche Informationen (Logs)

- Für Undo/Redo benötigten Informationen
- Informationen über Beginn und Ende von Transaktionen

Abbildung: CC by Aapo Haapanen

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

LSN (Log Sequence Number):

- Eindeutige Kennung des Log-Eintrags / Zeitliche Abfolge
- Müssen monoton aufsteigend vergeben werden
- Erlaubt Ermittlung der chronologischen Reihenfolge

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

LSN (Log Sequence Number):

- Eindeutige Kennung des Log-Eintrags / Zeitliche Abfolge
- Müssen monoton aufsteigend vergeben werden
- Erlaubt Ermittlung der chronologischen Reihenfolge

TransaktionsID:

- Kennung der Transaktion die die Änderung durchgeführt hat

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

LSN (Log Sequence Number):

- Eindeutige Kennung des Log-Eintrags / Zeitliche Abfolge
- Müssen monoton aufsteigend vergeben werden
- Erlaubt Ermittlung der chronologischen Reihenfolge

TransaktionsID:

- Kennung der Transaktion die die Änderung durchgeführt hat

PageID:

- Kennung der betroffenen Seite
- Erstelle einen Log-Eintrag pro betroffener Seite

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

Redo:

- Informationen um Änderung nachzuvollziehen

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

Redo:

- Informationen um Änderung nachzuvollziehen

Undo:

- Informationen um Änderung rückgängig zu machen

Struktur der Log-Einträge

Definition (Struktur eines Log-Eintrags)

`[LSN, TransaktionsID, PageID, Redo, Undo, PrevLSN]`

Redo:

- Informationen um Änderung nachzuvollziehen

Undo:

- Informationen um Änderung rückgängig zu machen

PrevLSN:

- Zeiger auf vorhergehenden Log-Eintrag der Transaktion
- Aus Effizienzgründen benötigt (lokales Undo!)

Beispiel von Log-Einträgen

Beispiel

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

Physische oder Logische Protokollierung

Physische Protokollierung: Redo- und Undo-Einträge enthalten Zustände:

- Undo enthält before-image (= Zustand vor Änderung)
- Redo enthält after-image (= Zustand nach Änderung)

Physische oder Logische Protokollierung

Physische Protokollierung: Redo- und Undo-Einträge enthalten Zustände:

- `Undo` enthält before-image (= Zustand vor Änderung)
- `Redo` enthält after-image (= Zustand nach Änderung)

Logische Protokollierung: Redo- und Undo-Einträge enthalten Operationen:

- `Undo` enthält wie before-image aus after-image erzeugt wird
- `Redo` enthält wie after-image aus before-image erzeugt wird

Physische oder Logische Protokollierung

Physische Protokollierung: Redo- und Undo-Einträge enthalten Zustände:

- `Undo` enthält before-image (= Zustand vor Änderung)
- `Redo` enthält after-image (= Zustand nach Änderung)

Logische Protokollierung: Redo- und Undo-Einträge enthalten Operationen:

- `Undo` enthält wie before-image aus after-image erzeugt wird
- `Redo` enthält wie after-image aus before-image erzeugt wird

Einfache Annahme für Zahlen:

- Physisch = Absolute Werte, bspw. Vorher: 7, Danach 10, LOG: 7,10
- Logisch = Differenz der Werte, bspw. Vorher: 7, Danach 10, LOG:+3,-3

Physische oder logische Protokollierung

Festlegung für die Vorlesung

- Logische Protokollierung
- Warum?
  - Verständnis leichter
  - Annahme: Mehrbenutzer-Synchronisation kümmert sich um Isolation

Praxis

- Oft: Physische Protokollierung

Physische vs. Logische Protokollierung

Beispiel

	T_1	T_2
1	BOT	
2		BOT
3	read(A, a_1)	
4	write(A, $a_1 + 5$)	
5		read(A, a_2)
6		write(A, $a_2 + 10$)
7	abort	

Rücksetzen nur bei logischer Protokollierung möglich

Hinweis: Verletzt Isolation (siehe Concurrency Control)

Rolle der LSN (Log Sequence Number)

Problem: Beim Wiederanlauf (Recovery) muss das DBMS erkennen ob die Auswirkung eines Log-Eintrags bereits auf der Seite vorhanden ist.

Rolle der LSN (Log Sequence Number)

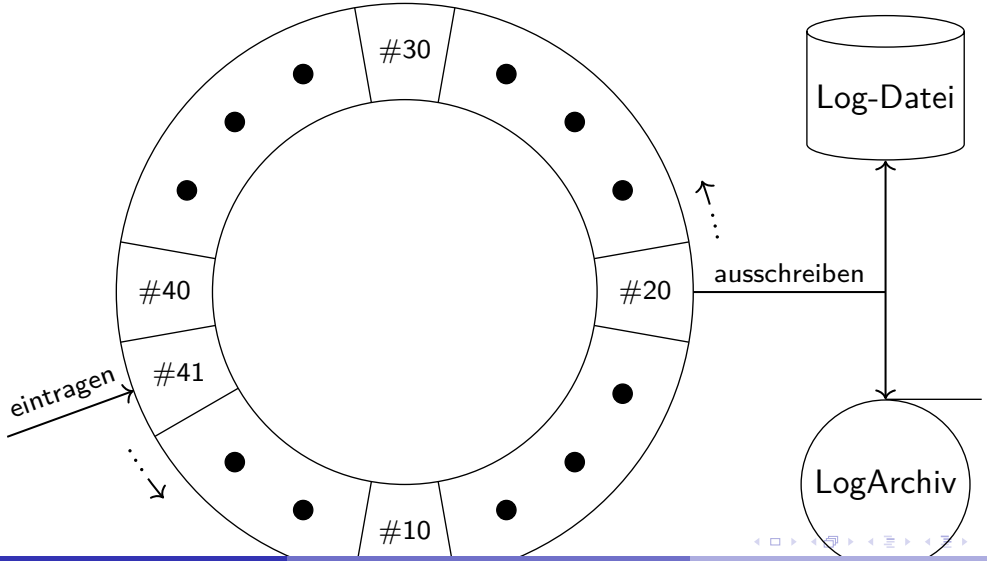
Problem: Beim Wiederanlauf (Recovery) muss das DBMS erkennen ob die Auswirkung eines Log-Eintrags bereits auf der Seite vorhanden ist.

Lösung Auch jede Seite hat ein Feld für eine LSN:

- Beim Anlegen eines Log-Eintrags: Schreibe LSN des Eintrags auf die Seite.
- Seite enthält LSN des jüngsten die Seite betreffenden Log-Eintrags

Inhalt

Schreiben von Log-Einträgen



Schreiben von Log-Einträgen

- Eigener Log-Puffer für Log-Einträge
  - Getrennt vom Puffer für Seiten
- Häufig: Ringpuffer
  - Kontinuierliches Ausschreiben vermeidet Last-Spitzen
- Log-Informationen werden doppelt ausgeschrieben:
  1. In ein temporäres Log (Hintergrundspeicher) für schnellen Zugriff
  2. In ein Log-Archiv für den Wiederanlauf nach einem Fehler mit Hintergrundspeicherverlust

Das WAL-Prinzip (Write Ahead Logging)

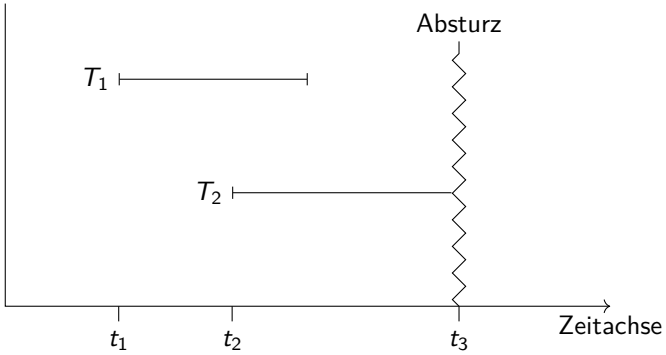
Die angenommene Konfiguration *steal*, *noforce*, *update-in-place* erfordert das **WAL-Prinzip** (*write ahead log*) einzuhalten:

- Bevor eine Transaktion festgeschrieben wird (erfolgreiches `commit`) müssen alle Log-Einträge der Transaktion ausgeschrieben werden (für Redo).
- Bevor eine modifizierte Seite ausgelagert werden darf, müssen alle zu dieser Seite gehörenden Log-Einträge ausgeschrieben werden (für Undo).
- Die chronologische Reihenfolge muss beim Ausschreiben erhalten bleiben.

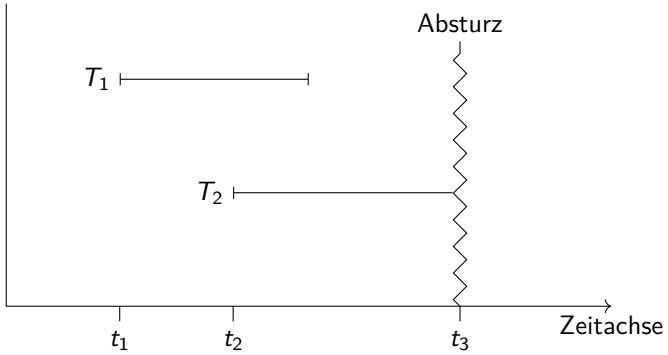
Inhalt

Inhalt

Wir unterscheiden zwei Arten von Transaktionen

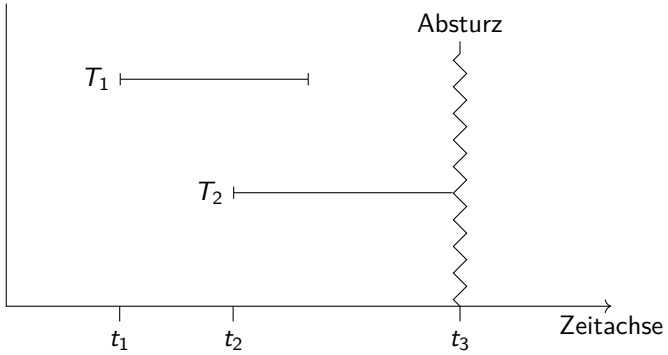


Wir unterscheiden zwei Arten von Transaktionen



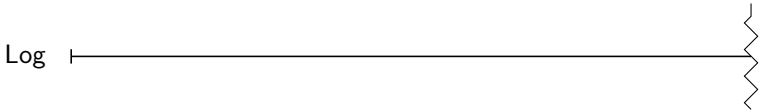
- Transaktionen vom Typ T_1 (Winner):
  Wirkung muss vollständig rekonstruiert werden.

Wir unterscheiden zwei Arten von Transaktionen

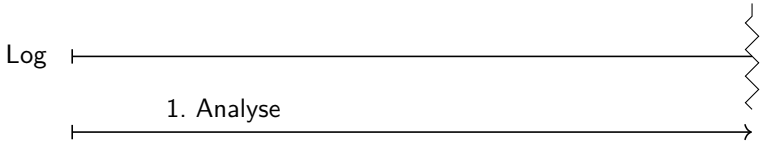


- Transaktionen vom Typ T_1 (Winner):
  Wirkung muss vollständig rekonstruiert werden.
- Transaktionen vom Typ T_2 (Loser):
  Wirkung muss vollständig rückgängig gemacht werden.

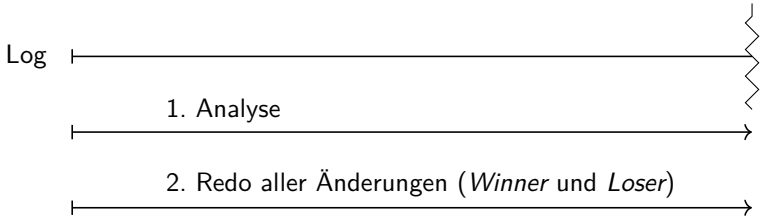
Drei Phasen des Wiederanlaufs



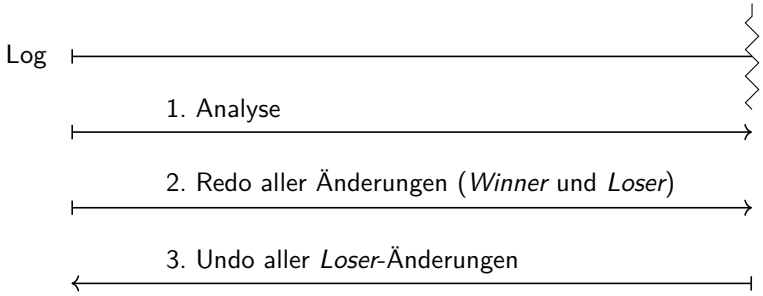
Drei Phasen des Wiederanlaufs



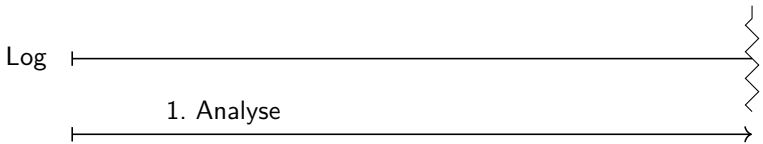
Drei Phasen des Wiederanlaufs



Drei Phasen des Wiederanlaufs

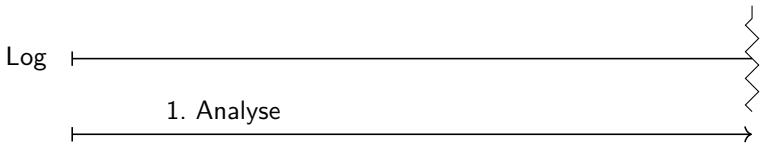


Drei Phasen des Wiederanlaufs: 1. Analyse



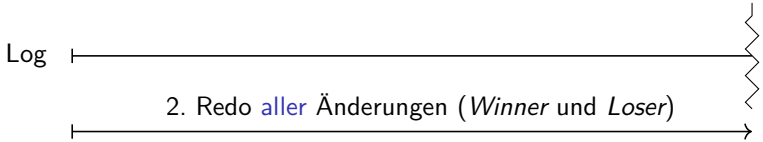
- Analysiere die gesamte Log-Datei

Drei Phasen des Wiederanlaufs: 1. Analyse



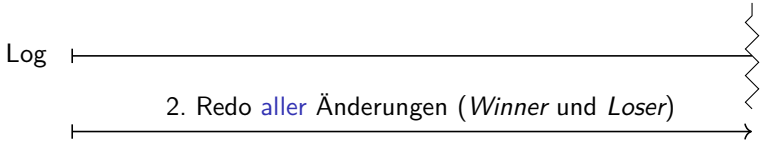
- Analysiere die gesamte Log-Datei
- Identifiziere Loser-Menge
  (= Transaktionen für die kein Abschluss protokolliert wurde)
- Finde höchste LSN für jede solche Transaktion

Drei Phasen des Wiederanlaufs: 2. Redo



Datenbasis wird auf exakten Stand bei Absturz gebracht

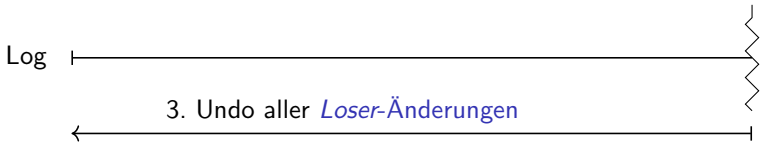
Drei Phasen des Wiederanlaufs: 2. Redo



Datenbasis wird auf exakten Stand bei Absturz gebracht

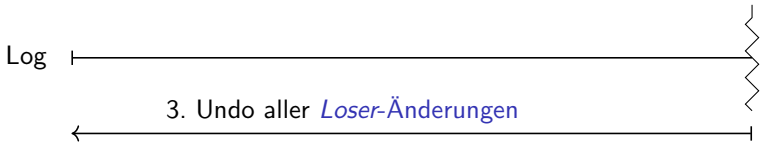
- Alle protokollierten Änderungen werden nachvollzogen
- In chronologischer Reihenfolge (d.h. geordnet nach LSN)
- Seiten-LSN bestimmt Zustand des Hintergrundspeichers:
  - LSN des Log Eintrags $>$ Seiten-LSN: Ausführen der Redo-Operation; Aktualisieren der Seiten-LSN
  - Sonst: Hintergrundspeicher enthält *after-image* der Operation

Drei Phasen des Wiederanlaufs: 3. Undo



Entferne Effekte nicht abgeschlossener Transaktionen

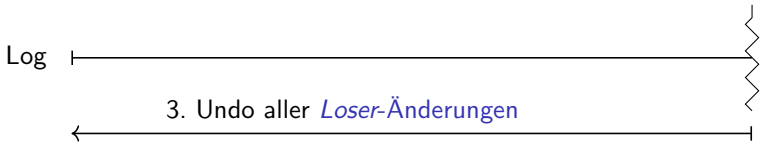
Drei Phasen des Wiederanlaufs: 3. Undo



Entferne Effekte nicht abgeschlossener Transaktionen

- Für jeden Log-Eintrag einer Loser-Transaktion: Verwende Undo-Information zur Wiederherstellung des *before-image*
- In umgekehrter chronologischer Reihenfolge (nach LSN)

Drei Phasen des Wiederanlaufs: 3. Undo



Entferne Effekte nicht abgeschlossener Transaktionen

- Für jeden Log-Eintrag einer Loser-Transaktion: Verwende Undo-Information zur Wiederherstellung des *before-image*
- In umgekehrter chronologischer Reihenfolge (nach LSN)
- Erzeuge speziellen Log-Eintrag für jedes ausgeführte Undo

Fehlertoleranz (Idempotenz) des Wiederanlaufs

Problem: auch während des Wiederanlaufs kann ein Fehler mit Verlust des Puffers auftreten.

Fehlertoleranz (Idempotenz) des Wiederanlaufs

Problem: auch während des Wiederanlaufs kann ein Fehler mit Verlust des Puffers auftreten.

Anforderung:

$$undo(undo(\cdots(undo(a))\cdots)) = undo(a)$$
$$redo(redo(\cdots(redo(a))\cdots)) = redo(a)$$

Fehlertoleranz (Idempotenz) des Wiederanlaufs

Problem: auch während des Wiederanlaufs kann ein Fehler mit Verlust des Puffers auftreten.

Anforderung:

$$undo(undo(\cdots(undo(a))\cdots)) = undo(a)$$
$$redo(redo(\cdots(redo(a))\cdots)) = redo(a)$$

Realisierung:

- Redo-Phase: mittels LSN (in Log-Eintrag bzw. Seiten-LSN)

Fehlertoleranz (Idempotenz) des Wiederanlaufs

Problem: auch während des Wiederanlaufs kann ein Fehler mit Verlust des Puffers auftreten.

Anforderung:

$$undo(undo(\cdots(undo(a))\cdots)) = undo(a)$$
$$redo(redo(\cdots(redo(a))\cdots)) = redo(a)$$

Realisierung:

- Redo-Phase: mittels LSN (in Log-Eintrag bzw. Seiten-LSN)
- Undo-Phase: mittels CLRs (compensation log record)

Compensation Log Record (CLR)

Definition (Struktur eines Compensation Log Eintrags)

$\langle \texttt{LSN}, \texttt{TransaktionsID}, \texttt{PageID}, \texttt{Redo}, \texttt{PrevLSN}, \texttt{UndoNxtLSN} \rangle$

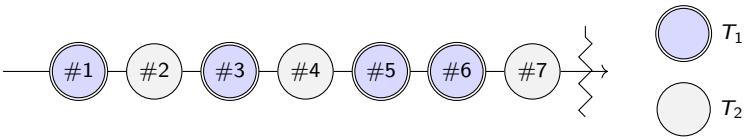
Keine Undo-Information:

- Rücknahme von Undo-Schritten nicht nötig:
  Entweder erfolgreich ausgelagert oder *Redo*

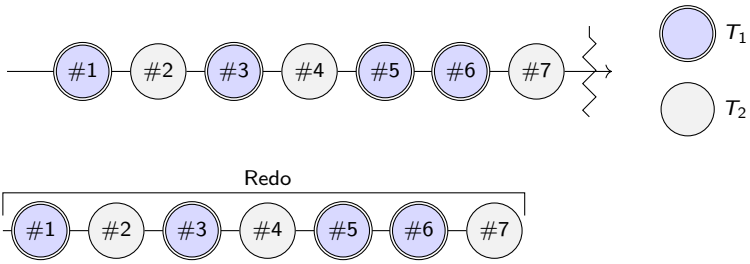
UndoNxtLSN:

- Verweis auf nächste rückgängig zu machende Änderung derselben Transaktion
- `PrevLSN` des rückgängig gemachten Log-Eintrags

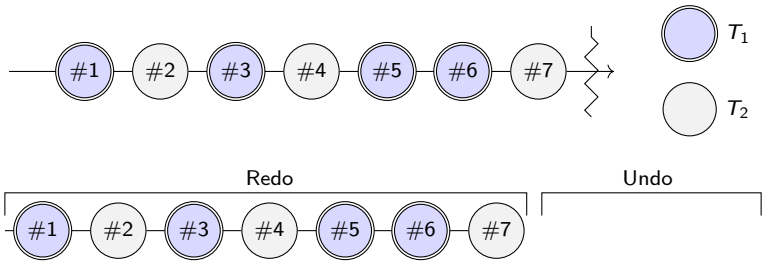
Beispiel: Kompensationseinträge im Log



Beispiel: Kompensationseinträge im Log

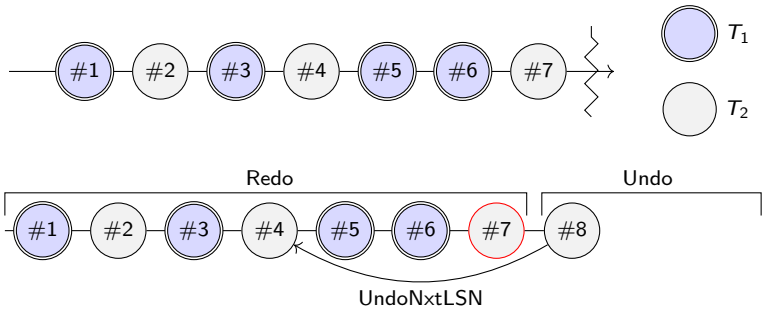


Beispiel: Kompensationseinträge im Log



CLRs für rückgängig gemachte Änderungen und für BOT:

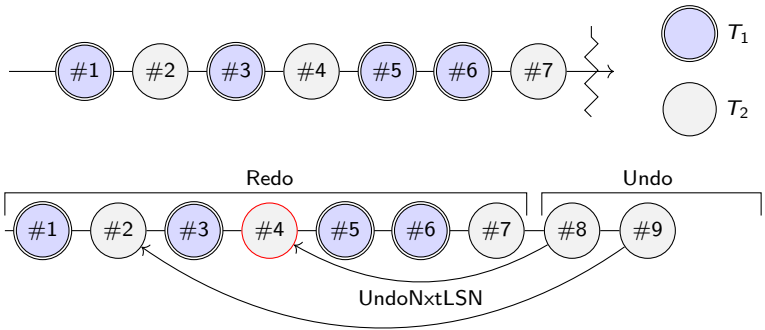
Beispiel: Kompensationseinträge im Log



CLRs für rückgängig gemachte Änderungen und für BOT:

- #8 ist CLR für #7

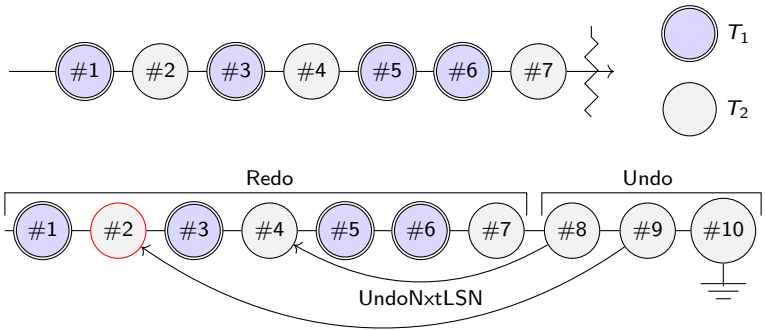
Beispiel: Kompensationseinträge im Log



CLRs für rückgängig gemachte Änderungen und für BOT:

- #8 ist CLR für #7
- #9 ist CLR für #4

Beispiel: Kompensationseinträge im Log



CLRs für rückgängig gemachte Änderungen und für BOT:

- #8 ist CLR für #7
- #9 ist CLR für #4
- #10 ist CLR für #2 (= BOT)

Beispiel Log-Einträge und Wiederanlauf

Beispiel

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$[\#1, T_1, \texttt{BOT}, 0]$
$[\#2, T_2, \texttt{BOT}, 0]$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$
$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$
$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$
$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

Beispiel Log-Einträge und Wiederanlauf

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$
$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$
$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$
$$\langle \#10, T_2, \text{-}, \text{-}, \#9, 0 \rangle$$

Wiederanlauf mit CLR

2. Redo:

- bearbeite gesamtes Log inklusive CLRs

3. Undo:

- Verwalte Liste aller Loser + höchste aktuelle LSN pro Loser
- Für Loser mit maximaler aktueller LSN:
  - "Normaler" Logeintrag:
    (1) Undo; (2) erstelle CLR;
    (3) setze aktuelle LSN auf `PrevLSN`
  - CLR:
    (1) Setze aktuelle LSN auf `UndoNxtLSN`

ARIES

(Algorithms for Recovery and Isolation Exploiting Semantics)

Erinnerung/Zusammenfassung

- Familie von Algorithmen zur Wiederherstellung nach Fehlerfall
- noforce/steal-Strategie
- WAL (Write Ahead Logging)

Weiter mit einem Beispiel....

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$
$r_1(A, a_1)$
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

Seiten

P_A LSN: #0
$A = 75$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$
$r_1(A, a_1)$
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$

Seiten

P_A LSN: #0
$A = 75$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*
1	2
2	3
3	1

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$

Seiten

P_A LSN: #0
$A = 75$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*
1	2
2	3
3	1

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$

Seiten

P_A LSN: #0
$A = 75$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*
1	2
2	3
3	1

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2]

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*
1	4
2	3
3	1

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*
1	4
2	3
3	1

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #0
$C = 10$

aktive Trans.

Tr.	*LSN*
1	4
2	5
3	1

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #0
$B = 120$	

P_C	LSN: #6
$C = 35$	

aktive Trans.

Tr.	*LSN*
1	4
2	5
3	6

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2]
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #7
$B = 170$

P_C LSN: #6
$C = 35$

aktive Trans.

Tr.	*LSN*
1	4
2	7
3	6

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$ [35]
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #7
$B = 170$	

P_C	LSN: #6
$C = 35$	

aktive Trans.

Tr.	*LSN*
1	4
2	7
3	6

Beispiel: Logging und Recovery

Historie

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$ [35]
$w_1(C, c_1 + 25)$
...

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #7
$B = 170$	

P_C	LSN: #8
$C = 60$	

aktive Trans.

Tr.	*LSN*
1	8
2	7
3	6

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #7
$B = 170$	

P_C	LSN: #8
$C = 60$	

aktive Trans.

Tr.	*LSN*
1	8
2	7
3	6

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #8
$C = 60$	

aktive Trans.

Tr.	*LSN*
1	8
2	7
3	9

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #10
$C = 85$	

aktive Trans.

Tr.	*LSN*
1	8
2	10
3	9

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #10
$C = 85$	

aktive Trans.

Tr.	*LSN*
1	8
2	10

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #10
$C = 85$	

aktive Trans.

Tr.	*LSN*
1	8
2	10

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #10
$C = 85$	

aktive Trans.

Tr.	*LSN*
1	8
2	10

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle \#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7 \rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

aktive Trans.

Tr.	*LSN*
1	8
2	7

Beispiel: Logging und Recovery

Historie

...
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle \#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7 \rangle$
Absturz

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

aktive Trans.

Tr.	*LSN*
1	8
2	7

Beispiel: Logging und Recovery

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Beispiel: Logging und Recovery

1. Analysephase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*Status*	
1	loser	
2	loser	
3	winner	

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2]
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	2
2	3
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	2
2	3
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	2
2	3
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	4
2	3
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$ ✓
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	4
2	3
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #4
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	4
2	3
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	4
2	5
3	1

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1] ✓
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #0
$B = 120$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	4
2	5
3	6

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$ ✓
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$ ✗
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$ ✓
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$ ✗
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #7
$B = 170$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	4
2	7
3	6

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1] ✓
[#7, T_2, P_B, B+=50, B-=50, #5] ✗
[#8, T_1, P_C, C+=25, C-=25, #4] ✓
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #7
$B = 170$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	8
2	7
3	6

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1] ✓
[#7, T_2, P_B, B+=50, B-=50, #5] ✗
[#8, T_1, P_C, C+=25, C-=25, #4] ✓
[#9, T_3, P_B, B-=75, B+=75, #6] ✗
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #9
$B = 95$

P_C LSN: #8
$C = 60$

Transaktionen

Tr.	*LSN*
1	8
2	7
3	9

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1] ✓
[#7, T_2, P_B, B+=50, B-=50, #5] ✗
[#8, T_1, P_C, C+=25, C-=25, #4] ✓
[#9, T_3, P_B, B-=75, B+=75, #6] ✗
[#10, T_2, P_C, C+=25, C-=25, #7] ✗
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #9
$B = 95$

P_C LSN: #10
$C = 85$

Transaktionen

Tr.	*LSN*
1	8
2	10
3	9

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1] ✓
[#7, T_2, P_B, B+=50, B-=50, #5] ✗
[#8, T_1, P_C, C+=25, C-=25, #4] ✓
[#9, T_3, P_B, B-=75, B+=75, #6] ✗
[#10, T_2, P_C, C+=25, C-=25, #7] ✗
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #9
$B = 95$

P_C LSN: #10
$C = 85$

Transaktionen

Tr.	*LSN*
1	8
2	10

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2] ✓
[#5, T_2, P_A, A+=0, A-=0, #3] ✗
[#6, T_3, P_C, C+=25, C-=25, #1] ✓
[#7, T_2, P_B, B+=50, B-=50, #5] ✗
[#8, T_1, P_C, C+=25, C-=25, #4] ✓
[#9, T_3, P_B, B-=75, B+=75, #6] ✗
[#10, T_2, P_C, C+=25, C-=25, #7] ✗
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩ ✗

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #9
$B = 95$

P_C LSN: #12
$C = 60$

Transaktionen

Tr.	*LSN*
1	8
2	12

Beispiel: Logging und Recovery

2. Redo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$ ✓
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$ ✗
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$ ✓
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$ ✗
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$ ✓
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$ ✗
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$ ✗
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$ ✗

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

Transaktionen

Tr.	*LSN*
1	8
2	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

Transaktionen

Tr.	NxtU	LSN
1	8	8
2	12	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle \#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7 \rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

Transaktionen

Tr.	NxtU	LSN
1	8	8
2	12	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle \#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7 \rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

Transaktionen

Tr.	NxtU	LSN
1	8	8
2	7	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

Transaktionen

Tr.	NxtU	LSN
1	8	8
2	7	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #12
$C = 60$	

Transaktionen

Tr.	NxtU	LSN
1	8	8
2	7	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2]
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩
→ ⟨#13, T_1, P_C, C-=25, #8, #4⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #9
$B = 95$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN
1	4	13
2	7	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #9
$B = 95$	

P_C	LSN: #13
$C = 35$	

Transaktionen

Tr.	NxtU	LSN
1	4	13
2	7	12

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2]
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩
→ ⟨#14, T_2, P_B, B-=50, #12, #5⟩

Seiten

P_A LSN: #5
$A = 25$

P_B LSN: #14
$B = 45$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN
1	4	13
2	5	14

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A	LSN: #5
$A = 25$	

P_B	LSN: #14
$B = 45$	

P_C	LSN: #13
$C = 35$	

Transaktionen

Tr.	NxtU	LSN
1	4	13
2	5	14

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$
$\rightarrow\ \langle\#15,\ T_2,\ P_A,\ \text{A-=0},\ \#14,\ \#3\rangle$

Seiten

P_A LSN: #15
$A = 25$

P_B LSN: #14
$B = 45$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN
1	4	13
2	3	15

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50, A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0, A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25, C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50, B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25, C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75, B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25, C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A	LSN: #15
$A = 25$	

P_B	LSN: #14
$B = 45$	

P_C	LSN: #13
$C = 35$	

Transaktionen

Tr.	NxtU	LSN
1	4	13
2	3	15

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$
$\rightarrow\quad \langle\#16,\ T_1,\ P_A,\ \text{A+=50},\ \#13,\ \#2\rangle$

Seiten

P_A	LSN: #16
$A = 75$	

P_B	LSN: #14
$B = 45$	

P_C	LSN: #13
$C = 35$	

Transaktionen

Tr.	NxtU	LSN
1	2	16
2	3	15

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #16
$A = 75$

P_B LSN: #14
$B = 45$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN
1	2	16
2	3	15

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

[#1, T_3, BOT, #0]
[#2, T_1, BOT, #0]
[#3, T_2, BOT, #0]
[#4, T_1, P_A, A-=50, A+=50, #2]
[#5, T_2, P_A, A+=0, A-=0, #3]
[#6, T_3, P_C, C+=25, C-=25, #1]
[#7, T_2, P_B, B+=50, B-=50, #5]
[#8, T_1, P_C, C+=25, C-=25, #4]
[#9, T_3, P_B, B-=75, B+=75, #6]
[#10, T_2, P_C, C+=25, C-=25, #7]
[#11, T_3, COMMIT, #9]
⟨#12, T_2, P_C, C-=25, #10, #7⟩
$\rightarrow$ ⟨#17, T_2, BOT, #15 ⟩

Seiten

P_A LSN: #16
$A = 75$

P_B LSN: #14
$B = 45$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN
1	2	16

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle\#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7\rangle$

Seiten

P_A LSN: #16
$A = 75$

P_B LSN: #14
$B = 45$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN
1	2	16

Beispiel: Logging und Recovery

3. Undo-Phase

Logeinträge

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$
$[\#5,\ T_2,\ P_A,\ \text{A+=0},\ \text{A-=0},\ \#3]$
$[\#6,\ T_3,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#1]$
$[\#7,\ T_2,\ P_B,\ \text{B+=50},\ \text{B-=50},\ \#5]$
$[\#8,\ T_1,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#4]$
$[\#9,\ T_3,\ P_B,\ \text{B-=75},\ \text{B+=75},\ \#6]$
$[\#10,\ T_2,\ P_C,\ \text{C+=25},\ \text{C-=25},\ \#7]$
$[\#11,\ T_3,\ \texttt{COMMIT},\ \#9]$
$\langle \#12,\ T_2,\ P_C,\ \text{C-=25},\ \#10,\ \#7 \rangle$
$\rightarrow \quad \langle \#18,\ T_1,\ \texttt{BOT},\ \#16 \rangle$

Seiten

P_A LSN: #16
$A = 75$

P_B LSN: #14
$B = 45$

P_C LSN: #13
$C = 35$

Transaktionen

Tr.	NxtU	LSN

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

Redo-Phase:

- Log-Einträge #1 – #8
- Beachte Seiten-LSN

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

Redo-Phase:

- Log-Einträge #1 – #8
- Beachte Seiten-LSN

Undo-Phase:

- Folge UndoNxtLSN

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}=50, A\text{+}=50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}=100, C\text{-}=100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}=50, B\text{-}=50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}=100, A\text{+}=100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}=100, \#7, \#4 \rangle$$

Redo-Phase:

- Log-Einträge #1 – #8
- Beachte Seiten-LSN

Undo-Phase:

- Folge UndoNxtLSN

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

Redo-Phase:

- Log-Einträge #1 – #8
- Beachte Seiten-LSN

Undo-Phase:

- Folge UndoNxtLSN

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

Redo-Phase:

- Log-Einträge #1 – #10
- Beachte Seiten-LSN

Wiederanlauf mit CLR – Beispiel

Beispiel

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

Redo-Phase:

- Log-Einträge #1 – #10
- Beachte Seiten-LSN

Undo-Phase:

- Folge UndoNxtLSN
- UndoNxtLSN = 0
  $\Rightarrow$ T_2 ist fertig

ARIES

Zusammenfassung

- Beschriebenes Vorgehen bildet Kern des ARIES Verfahrens.
- Weitere Verfeinerungen dienen hauptsächlich der Reduktion von Zugriffen auf den Hintergrundspeicher.
- Sehr flexible Recovery Methode

ARIES

Zusammenfassung

- Beschriebenes Vorgehen bildet Kern des ARIES Verfahrens.
- Weitere Verfeinerungen dienen hauptsächlich der Reduktion von Zugriffen auf den Hintergrundspeicher.
- Sehr flexible Recovery Methode

- Kernkomponenten von ARIES:
  - WAL
  - Wiederholung der vollständigen Historie im Redo-Schritt.
  - Erzeugen von CLRs im Undo-Schritt.

Inhalt

Lokales Zurücksetzen einer Transaktion

1. Ermittle letzten Log-Eintrag der Transaktion
2. Lokales Undo: Änderungen der Transaktion werden rückgängig gemacht
   - Log wird entgegen der chronologischen Reihenfolge durchlaufen
   - CLRs werden angelegt

Lokales Zurücksetzen einer Transaktion

1. Ermittle letzten Log-Eintrag der Transaktion
2. Lokales Undo: Änderungen der Transaktion werden rückgängig gemacht
   - Log wird entgegen der chronologischen Reihenfolge durchlaufen
   - CLRs werden angelegt

Effizienz:

- Üblicherweise wird für jede Transaktion Zeiger auf letzten Log-Eintrag verwaltet.
- Rückwärtsverkettung mittels `PrevLSN` erlaubt schnelles Durchlaufen des Logs.
- Bei realistischer Größe sind die meisten Log-Einträge aktiver Transaktionen noch im Puffer.

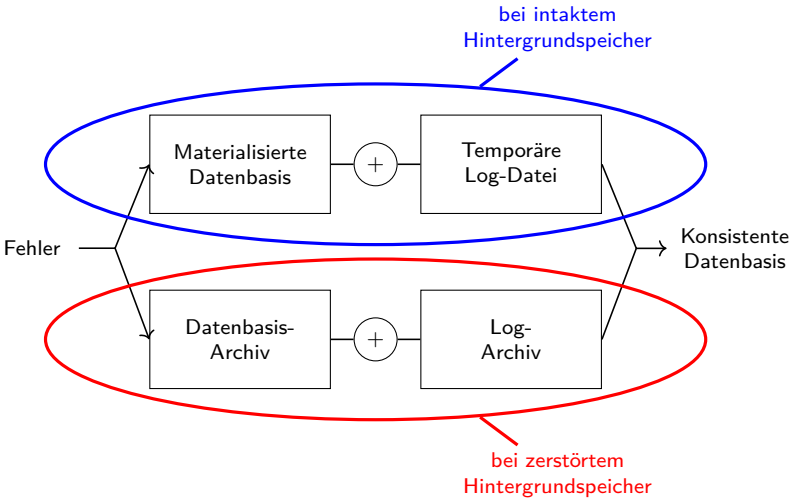
Partielles Zurücksetzen einer Transaktion

1. Ausgehend vom letzten Log-Eintrag, setze Änderungen zurück und erzeuge CLRs (wie bei lokalem Undo).
2. Unterschied: Gehe nicht bis zu `BOT`, sondern setze Transaktion nur bis zum gewünschten Punkt zurück.

Erlaubt die Umsetzung von Savepoints

Inhalt

Media-Recovery



Inhalt

Warum Sicherungspunkte (Checkpoints)

Problem bei vorgestellter Recovery-Methode:

- Es muss immer das gesamte Log durchlaufen werden.
- Das Log wird mit der Zeit immer größer.

Warum Sicherungspunkte (Checkpoints)

Problem bei vorgestellter Recovery-Methode:

- Es muss immer das gesamte Log durchlaufen werden.
- Das Log wird mit der Zeit immer größer.

Lösung:

- Erzwinge (gelegentlich) das Ausschreiben von geänderten Seiten = Sicherungspunkt (Checkpoint).
- Dadurch wird das Log erst ab einer bestimmten LSN benötigt.
- Die benötigte minimale LSN hängt von Zeitpunkt und Art des Sicherungspunktes ab.

Warum Sicherungspunkte (Checkpoints)

Problem bei vorgestellter Recovery-Methode:

- Es muss immer das gesamte Log durchlaufen werden.
- Das Log wird mit der Zeit immer größer.

Lösung:

- Erzwinge (gelegentlich) das Ausschreiben von geänderten Seiten = Sicherungspunkt (Checkpoint).
- Dadurch wird das Log erst ab einer bestimmten LSN benötigt.
- Die benötigte minimale LSN hängt von Zeitpunkt und Art des Sicherungspunktes ab.

Achtung! Checkpoint $\neq$ Savepoint!

Arten von Sicherungspunkten

Transaktionskonsistente Sicherungspunkte

- "Beste" Qualität für Recovery, sehr teuer anzulegen

Aktionskonsistente Sicherungspunkte

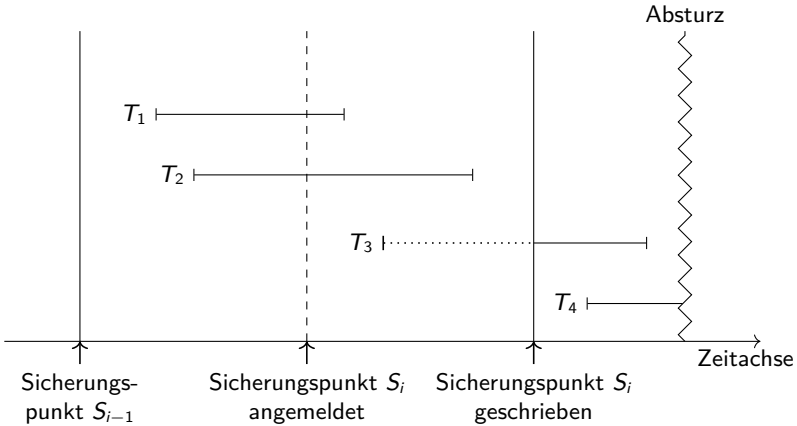
- *Undo* benötigt "ältere" Log-Einträge, erzeugt Last-Spitzen

Unscharfe (Fuzzy) Sicherungspunkte

- *Undo* und *redo* benötigen "ältere" Log-Einträge, kontinuierliches Ausschreiben.

Transaktionskonsistente Sicherungspunkte

Stop-New



Transaktionskonsistente Sicherungspunkte

Idee:

- Hintergrundspeicher soll alle Änderungen von zum Zeitpunkt S_i abgeschlossenen Transaktionen enthalten.
  $\Rightarrow$ kein Redo über S_i hinaus nötig.
- Zum Zeitpunkt S_i darf es keine aktiven Transaktionen geben.
  $\Rightarrow$ kein Undo über S_i hinaus nötig.

Transaktionskonsistente Sicherungspunkte

Idee:

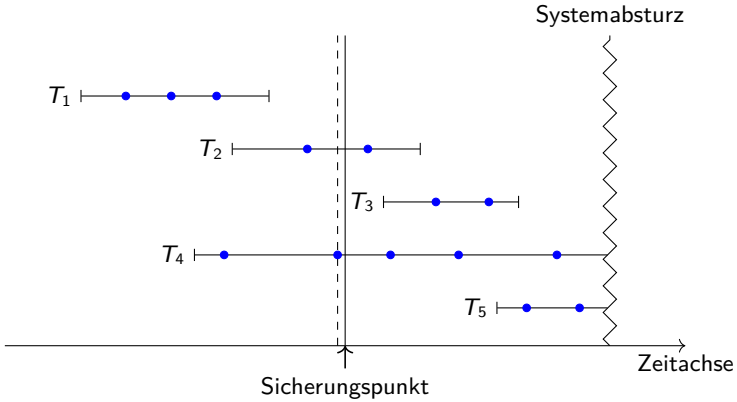
- Hintergrundspeicher soll alle Änderungen von zum Zeitpunkt S_i abgeschlossenen Transaktionen enthalten.
  $\Rightarrow$ kein Redo über S_i hinaus nötig.
- Zum Zeitpunkt S_i darf es keine aktiven Transaktionen geben.
  $\Rightarrow$ kein Undo über S_i hinaus nötig.

Problem:

- Zwischen Anmelden und Festschreiben des Sicherungspunktes dürfen keine neuen Transaktionen gestartet werden.
- Führt zu üblicherweise nicht akzeptablen Wartezeiten.

Aktionskonsistente Sicherungspunkte

Stop-Write



Aktionskonsistente Sicherungspunkte

Idee:

- Alle aktiven Änderungsoperationen sollen beendet werden.
- Danach ausschreiben aller modifizierten Seiten
- Es wird nur das Starten der nächsten Änderungsoperation (aber nicht das Starten neuer Transaktionen) verzögert.

Aktionskonsistente Sicherungspunkte

Idee:

- Alle aktiven Änderungsoperationen sollen beendet werden.
- Danach ausschreiben aller modifizierten Seiten
- Es wird nur das Starten der nächsten Änderungsoperation (aber nicht das Starten neuer Transaktionen) verzögert.

Wiederanlauf:

- Analyse-Phase setzt bei S_i auf.
- Kein Redo über S_i hinaus nötig.
- Im allgemeinen ist ein Undo über S_i hinaus nötig; bis zur *MinLSN* (= die kleinste LSN der zum Sicherungszeitpunkt aktiven Transaktion).

Unscharfe (fuzzy) Sicherungspunkte

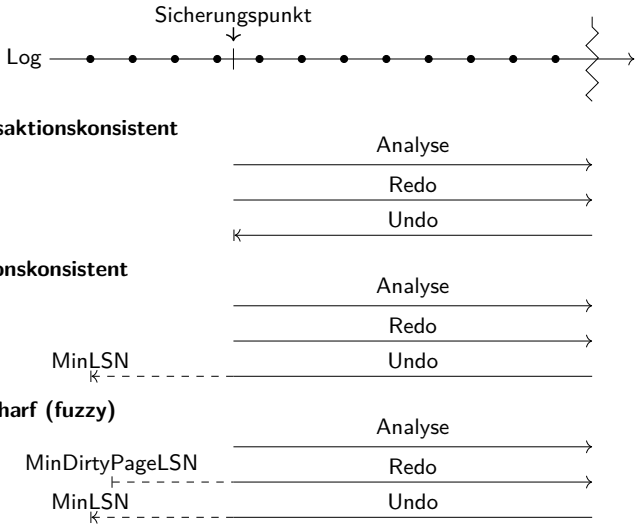
Idee:

- Seiten sollen kontinuierlich ausgeschrieben werden. Kein abruptes Ausschreiben vieler Seiten wegen Sicherungspunkt.
- Zum Sicherungspunkt werden nur die Kennungen aller modifizierten Seiten (= Dirty Pages) ausgeschrieben.
- Zusätzlich wird *MinDirtyPageLSN* (= minimale LSN, deren Änderungen noch nicht ausgeschrieben wurden) verwaltet und bei einem Sicherungspunkt ausgeschrieben.

Problem:

- „hot-spots“ (laufend benötigte Seiten) werden lange nicht ausgeschrieben. Ausschreiben wird erzwungen, wenn eine Seite mehrmals in der „Dirty Pages“-Menge war.

Drei Sicherungspunkt-Qualitäten – Zusammenfassung



Lernziele

- Was sind Seiten?
- Welche Strategien zur Ersetzung/Auslagerung von Pufferseiten gibt es? Vorteile? Nachteile?
- Welche Einbringsstragien gibt es und wie funktionieren sie?
- Welche Fehlerkategorien gibt es?
- Welche Informationen enthält ein Log-Eintrag?
  - Unterschied zwischen logischer und physischer Protokollierung?
  - Was ist die LSN? Wozu braucht man sie?
- Was ist das WAL-Prinzip?
- Wie funktioniert der Wiederanlauf nach einem Fehler mit Buffer Verlust?
  - Phasen des Wiederanlaufs und wie funktionieren sie?
  - Wie wird die Fehlertoleranz des Wiederanlaufs umgesetzt?
  - Was sind CLRs?
- Wozu dienen Sicherungspunkte und welche Arten gibt es?