

Transaktionale Informationssysteme

SoSe19

Benedikt Lüken-Winkels

April 23, 2019

Contents

1	1. Vorlesung	2
2	2.Vorlesung	4
2.1	Objekt-Modell	4
2.2	Concurrency Control	4

1 1. Vorlesung

Foliensatz 1

Orga

- **Vorlesung** Di, 14:15-15:45, H11
- **Übung** Mo, 13-14
- **Prüfung** mündlich 16.06. und 22.10.

Motivation

Bei vielen, kurzen Transaktionen (Änderungen) darf die Datenbasis nicht zerstört werden

- Rollback
- Administration der Aktionen auf der Datenbasis
- \Rightarrow Datenkonsistenz

Konsistenz Bewahrung der Korrektheit Daten im Fehlerfall

Generizität Abstraktion von Szenarien

Paralleler Zugriff Beispiel 1.1 (Folie 12)

Naive Parallelverarbeitung sorgt zum Konflikt

Optimistischer Ansatz Laufen lassen, bis ein Fehler Auftritt

Pessimistische Ansatz Zugriff blockieren

Fehlerhafte Ausführung Beispiel 1.2 (Folie 13)

Prozess wird durch Fehler unterbrochen

Rollback Sollten nicht alle Aktionen ausführbar sein, nicht ausführen (Komplett oder gar nicht)

Verteiltes Datensystem Beispiel 1.3 (Folie 14)

Verschiedene Datenbestände nicht korrekt synchronisiert (zB Client- und Serverwarenkorb),
Datensysteme sind verschieden und unabhängig voneinander (heterogen und autonom)

Transaktionale Eigenschaften

- Synchronisierung von Client und Serverinformationen
- Verifikation des Abschlusses einer Transaktion

Beispiel 1.4 (Folie 19)

Gesamte Aktion muss erfolgreich sein: Schlägt eine Transaktion im Block fehl, wirft eine Fehlermeldung (zB Prüfungsanmeldung und Bestätigung)

Workflow Management

Spezifikation von Workflows

- Wer bekommt welche Rolle

Workflow

- Geschäftsprozess (zB Beschaffung, Reiseplanung)
- Langlebig

Aktivität Teile eines Workflows, die von verschiedenen Akteuren ausgeführt werden

Architekturen

Einfache Server Struktur (Folie 27) Data Server: Datendarstellung

- Gekapselt in Objekten (Request, Reply)
- Ungekapselt als Tupel

Föderierte Systeme

- Alte Systeme müssen mit neuen Systemen kooperieren

Transaktionsmanagement

ACID (Folie 30)

- Atomarität: Ganz oder gar nicht
- Konsistenz: Konsistenzhaltung, waren die Daten Konsistent vor der Transaktion, sind sie es auch danach
- Isolation: Transaktionen beeinflussen sich nicht gegenseitig
- Dauerhaftigkeit: Wenn Transaktion erfolgreich, so ist sie in der Datenbank vorhanden

Anforderungen and Transaktionsmanagement (Folie 31)

- Concurrency Control
- !nachgucken!

Aufbau (Folie 32)

- Transaktionsmanagement sorgt für Synch der Zugriffe
- Datenbank-Cache: Lesen und Bearbeiten der Daten im DB-Cache. Schreiben geschieht später
- DB Seiten (Folie 37)

2 2.Vorlesung

Transaktion ist eine Sequenz von Operationen

Partiell geordnete Transaktionen

- Reflexiv
- Antisymmetrisch
- Transitiv

2.1 Objekt-Modell

Baumdarstellung einer Transaktion (welche Methode ruft welche Methode auf)

- Baum mit endlicher Höhe
- Innere Knoten sind Namen und Parameter von Operationen
- Blätter sind Seitenoperationen
- Besteht eine Ordnung auf einer Ebene, so ist die höhere Ebene auch geordnet

2.2 Concurrency Control

Klassische Synchprobleme

Verlust von ACID Eigenschaften, wenn Transaktionen unkontrolliert ausgeführt werden.

- Lost Update Problem: Keine Kommunikation zwischen Prozessen während eines Updates
- Inconsistent Read Problem: Transaktion₁ liest während Transaktion₂ läuft \Rightarrow Änderungen nicht abgeschlossen
- Dirty Read Problem: Ein Zwischenwert einer Transaktion wird für andere Transaktionen lesbar \Rightarrow Abbruch der Transaktion sorgt für Fehler durch gelesenen Wert

Schedules

Modell für verschränkte/parallele Ausführung von Transaktionen

- Transaction Manager entscheidet endgültig über die Reihenfolge in der Ausführung der Transaktionen verschiedener Prozesse
- Abort einer Transaktion: Rückführung aller Effekte
- Commit einer Transaktion
 - Abgeschlossen und Effekte für andere Transaktionen sichtbar
 - Muss atomar durchlaufen (ganz oder gar nicht)

Historie

- Vollständige Darstellung aller Transaktionen (committed und aborted)
- In der Praxis ist nur der erste Ausschnitt (Präfix) sichtbar \Rightarrow Schedule
- Serielle Historie: Transaktionen werden hintereinander ausgeführt (nicht parallel)
- Nicht-serielle Historie: Verschränkte Ausführung von Transaktionen

Shuffle Produkt Mischen von Transaktionen, die verschiedene Ausführungsreihenfolgen ergeben. Shuffle Produkt ist gültig, wenn

- Reihenfolge der Operationen beibehalten
- Keine Operationen hinzufügen
- Keine Operation entfernen

Aus der Menge der gültigen Produkte muss der Scheduler ein Element (eine Ausführungsmöglichkeit) wählen.

Total geordneter Schedule Entscheidungen werden auf Grund des Wissens aus den Präfixen einer Historie getroffen.

Partiell geordnete Historien Für Histoire S gilt:

- S ist vollständig: Transaktionen und Ergebnis (commit/abort)
- Eine Transaktion in S ist entweder committed oder aborted
- Ist eine Transaktion geordnet, so ist sie es auch in S
- Ordnung verschiedener Transaktionen auf dem gleichen Objekt