ETH Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Dr. G. Alonso *C. Hagen, B. Kemme*

AR 1: Recovery

Theoretische Übung 3 Abgabe: 12.6.1997

Aufgabe 1: Undo/Redo

Wir betrachten einen Recovery-Algorithmus, der UNDO und REDO durchführt, aber die Datenobjekte nicht mit der *log sequence number* (LSN) des Log-Eintrags der letzten Änderung markiert. Um Log-Speicherplatz zu sparen, werden nur *after-images* aufgezeichnet. Es wird *commit consistent* Checkpointing verwendet.

Definieren Sie eine Datenstruktur für das Log (Update-Records, Checkpoint-Records, etc.). Beschreiben Sie die Algorithmen für die Standard-Recovery-Prozeduren *Write, Read, Commit, Abort, und Restart* sowie für das Checkpointing. Beschreiben Sie insbesondere, wie diese Prozeduren die Log-Struktur verwenden.

Welche Probleme treten auf, wenn *cache consistent* Checkpointing verwendet wird?

Aufgabe 2: Versionen

Im No-Undo/Redo-Algorithmus ist es möglich, *shadowing* zu verwenden, um es einer Transaktion zu ermöglichen, die Daten zu lesen, die sie selbst geändert hat: Wenn eine Transaktion T_i das Objekt x zum ersten mal schreibt, wird eine neue Version von x angelegt und in den Cache geschrieben. Die vorherige (Schatten-) Version von x wird nicht überschrieben.

Arbeiten Sie die Details dieses Ansatzes aus, indem sie Datenstrukturen und Algorithmen für die Recovery-Prozeduren (Write, Read, Commit, Abort, Restart) angeben.