

Undo Tablespace statt Blockaden

Blick in die Vergangenheit

Thomas Klughardt
Senior System Consultant



Atomicity, Consistency und Isolation



Das ACID Modell

- Transaktionen in Oracle Datenbanken arbeiten ACID Konform.
 - ACID steht für Atomic, Consistent, Isolated und Durable.
- Das bedeutet:
 - Transaktionen werden ganz oder gar nicht durchgeführt.
 - Die Daten sind zueinander stets konsistent.
 - Andere Sessions sehen Änderungen erst, nachdem die gesamte Transaktion abgeschlossen ist.
 - Wenn die Transaktion abgeschlossen ist, sind die Daten dauerhaft gespeichert.
- Das ist gar nicht so einfach...



Das Oracle Transaktionsmodell

- Viele Datenbanken arbeiten mit Lesesperren.
 - Wird ein Datensatz geändert, darf er von anderen Sessions nicht gelesen werden.
 - Erst das Ende einer Transaktion hebt die Lesesperre auf.
 - Oft werden die Daten auch erst dann endgültig geschrieben.
- Eigenschaften bei Oracle – Multiversion Consistency
 - Optimistisches Commit Verhalten – meist erfolgt ein Commit.
 - › Änderungen werden sofort an die endgültige Stelle geschrieben.
 - Konsistente Abfragen
 - › Wenn wir eine Abfrage zum Zeitpunkt x starten, bekommen wir auch die Daten zum Zeitpunkt x.
 - Keine Sperren für Änderungen während des Lesens
 - › Während eine Abfrage läuft dürfen die zugrunde liegenden Daten geändert werden.
 - › Trotzdem ist das Ergebnis konsistent zum Startzeitpunkt der Abfrage.



Wie macht Oracle das?

- Wie macht Oracle das?
 - Wird ein Wert geändert, wird das sogenannte Before Image gespeichert.
 - › Der Wert selbst wird ersetzt.
 - Oracle weiß, wann welche Daten geändert wurden und wann eine Abfrage anfängt.
 - › Eindeutige globale Sequenz – System Change Number (SCN)
 - Falls nötig liest die Abfrage das Before Image und nicht den aktuellen Wert.
- Oracle wird dadurch zu einer hoch-performanten Datenbank.
 - Gerade im Multisession Betrieb skaliert Oracle besonders gut.
 - Keine Locking Probleme für Anwendungen.
 - Auch im Hochlastbetrieb konsistente Abfrageergebnisse.



Das Dirty Read „Feature“

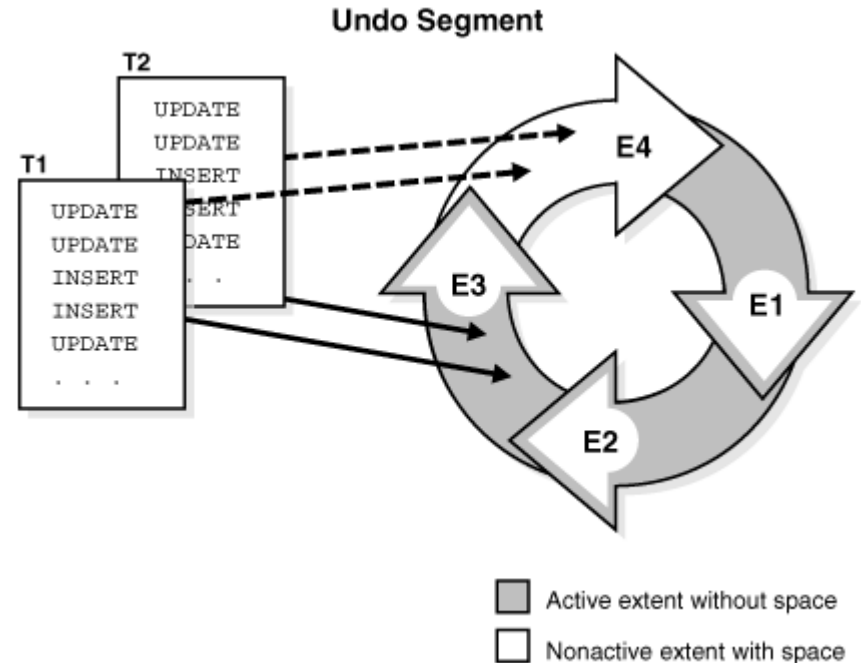
- Manche Datenbanken erlauben Dirty Reads.
 - Daten wie sie jetzt im Moment sind. Also auch:
 - › Nicht commitete Änderungen
 - › Später commitete Änderungen
- Dirty Reads verletzen das I in ACID (Isolation).
 - Abfrageergebnisse sind inkonsistent.
 - Master- und Detailtabellen passen nicht zusammen.
- Dirty Reads lesen aktuelle Daten...
 - Bei kompletter Denormalisierung nicht so schlimm
 - › Obwohl neuere Zeilen „überlesen“ werden können.
 - Wenn Dirty Reads benötigt werden, braucht man dann eine relationale Datenbank?

Before Images



Was ist ein Before Image?

- „Vorher“ Abbild eines Datensatzes
 - Die „Undo Daten“
- Liegt in einem Undo Segment
 - Ringpuffer, wird periodisch überschrieben
- Benötigt für:
 - Rollbacks
 - Konsistentes Lesen
 - Flashback Feature
 - UND UPDATES und DELETES!

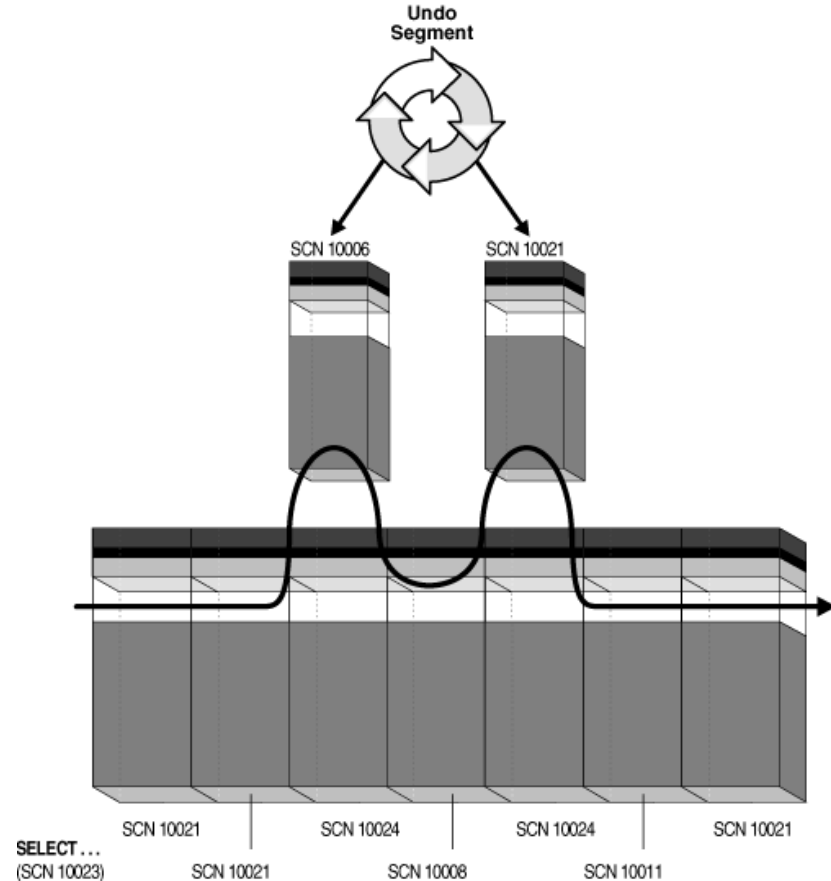


Quelle: Oracle Dokumentation: Concepts and Architecture Guide



Was passiert beim Lesen?

- Erinnerung: SCN – globale Sequenz
 - System Change Number (SCN)
 - Jede Transaktion bekommt eine SCN
 - Blöcke kennen ihre Version
 - › Es kann mehrere Blöcke mit verschiedenen Versionen geben.
- Oracle sucht den aktuellsten Block
 - Im Cache
- Wenn die SCN zu groß ist wird auf Undo Daten zugegriffen.



Quelle: Oracle Dokumentation: Concepts and Architecture Guide



Historie – Rollback Segmente

- Wurden früher (bis 9i) in Rollback Segmenten gespeichert.
 - Rollback Segmente mussten von Hand angelegt, gesized und gelöscht werden.
 - Transaktionen benutzten „ihre“ Rollback Segmente.
 - Best Practice:
 - › Große Rollback Segmente für Anwendungen mit großen Transaktionen.
 - › Kleine Rollback Segmente für Anwendungen mit kleinen Transaktionen.
 - › Rollback Segmente auf zur IO Verteilung verschiedene Platten.
- Nicht wirklich zeitgemäß, oft problematisch
 - Stripe and Mirror Everything (SAME) erlaubt keine gezielte IO Verteilung.
 - ORA-01555: Snapshot too old



Historie – Undo Tablespace

- Mit 9i wurde alles besser, auch wenn es viele erst mit 10g merkten:
 - Undo Tablespace und automatisches Undo Management
 - › Mit 9i eingeführt
 - › Mit 10g Standard
- Multipurpose Datenbank
 - Welcher DBA kennt schon jede Anwendung bis ins Detail?
- Oracle kann sich um das Undo Management kümmern.
 - Bei Striping und Mirroring keine merklichen Nachteile.
 - Dafür viele Vorteile...



Undo Tablespace



Historie – Undo Tablespace

- Before Images liegen jetzt im Undo Tablespace.
- Natürlich wichtige Komponente – alles geht durch den Undo Tablespace
 - › Jede Änderung schreibt
 - › Es wird viel daraus gelesen
 - › Viel IO Last
- Nicht sehr granular -> alles geht durch den/die Undo Tablespace(s)
 - Hat sich in der Praxis aber sehr bewährt.



Undo Tablespace

- Mehrere Undo Tablespaces möglich
- Wird über Instanzparameter gesteuert
 - Interessant für Oracle RAC Datenbanken

Initialization Parameter	Description
UNDO_MANAGEMENT	If AUTO or null, enables automatic undo management. If MANUAL, sets manual undo management mode. The default is AUTO.
UNDO_TABLESPACE	Optional, and valid only in automatic undo management mode. Specifies the name of an undo tablespace. Use only when the database has multiple undo tablespaces and you want to direct the database instance to use a particular undo tablespace.



Historie – Undo Tablespace

- Before Images liegen jetzt im Undo Tablespace.
- Natürlich wichtige Komponente – alles geht durch den Undo Tablespace
 - › Jede Änderung schreibt
 - › Es wird viel daraus gelesen
 - › Viel IO Last
- Nicht sehr granular -> alles geht durch den/die Undo Tablespace(s)
 - Hat sich in der Praxis aber sehr bewährt.



Konfiguration für den Undo Tablespace

- Er sollte definitiv nicht zu klein sein!
- Wie lange sollen Daten vorgehalten werden?
 - `undo_retention`
- Müssen die Daten tatsächlich so lange vorgehalten werden?
 - `retention_guarantee`

**Vorsicht! – Ist Retention Garantie gesetzt, dürfen die benötigten Undo Daten nicht vor der Retention überschrieben werden!
Die Datenbank wartet in dem Fall, bis die Retention abgelaufen ist!**



Konfiguration für den Undo Tablespace

- **retention_guarantee** setzen wir also besser nicht...
- Trotzdem sollten die Undo Daten ausreichend lange bereitstehen.
 - **undo_retention** ruhig großzügig setzen
 - Entsprechend groß muss der Undo Tablespace sein.
 - Es kann sinnvoll sein, mehrere Undo Tablespaces zu haben.
 - › Z.B Oracle Real Application Cluster
- Oft ergeben sich ungeplante Anforderungen.
 - **dpexport ... FULL=y FLASHBACK_SCN=<SCN>**
 - **SELECT * FROM a,b,c,d,e,f,g,h WHERE ...;**



Flashback



Was geht noch – außer konsistenten Abfragen?

- Die Undo Segmente enthalten die Vorher Abbilder.
 - Und sie kennen ihren genauen Zeitpunkt.
- Abfragen gegen die Daten aus der Vergangenheit!
- Wichtig:
 - Wie lange werden die Daten in der Vergangenheit vorgehalten?
 - › **undo_retention**
 - › Undo Tablespace Größe



Flashback

- Flashback Query
 - **SELECT ... AS OF <SCN/TIMESTAMP> ...;**
 - **SELECT * FROM "BIN\$TDHqmJRKR5u+Hrg6PGD2kw==\$0";**
- Flashback Table
 - **FLASHBACK TABLE ... TO <SCN/TIMESTAMP> ...;**
 - **FLASHBACK TABLE ... TO BEFORE DROP;**
- Einfacher Weg, von User Fehlern zu recovern.
 - Besonders stark in Kombination mit dem LogMiner.



Flashback – statt Recovery nach User Fehlern

The screenshot shows the Toad for Oracle LogMiner interface. The main window displays a table of database operations. The table has five columns: SCN, Time Stamp, Username, Operation, and SQL Redo. The operations are listed in chronological order, with the most recent operation at the bottom. The operations include internal operations, deletions, and insertions. The SQL Redo column contains the SQL statements that were executed.

SCN	Time Stamp	Username	Operation	SQL Redo
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	DELETE	delete from "SYS"."WRI\$_OPTSTAT_HISTGRM_HISTORY" where "OBJ#" = '17453' and "INTCOL#'
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	DELETE	delete from "SYS"."WRI\$_OPTSTAT_HISTGRM_HISTORY" where "OBJ#" = '17453' and "INTCOL#'
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	DELETE	delete from "SYS"."WRI\$_OPTSTAT_HISTGRM_HISTORY" where "OBJ#" = '17453' and "INTCOL#'
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486720	12.11.2014 17:16:49	UNKNOWN	INTERNAL	
74486721	12.11.2014 17:16:49	UNKNOWN	COMMIT	commit;
74486722	12.11.2014 17:16:50	UNKNOWN	INSERT	insert into "PERFSTAT"."STATS\$MUTEX_SLEEP"("SNAP_ID","DBID","INSTANCE_NUMBER","MUTE
74486722	12.11.2014 17:16:50	UNKNOWN	INSERT	insert into "PERFSTAT"."STATS\$MUTEX_SLEEP"("SNAP_ID","DBID","INSTANCE_NUMBER","MUTE

The interface also shows a menu bar (File, Edit, Search, Editor, Session, Database, Debug, View, Utilities, Rerun, Window, Help), a toolbar, and a status bar at the bottom with "Autocommit is OFF" and "CAPS NUM INS" indicators. The window title is "Toad for Oracle - [SYS@LOKAL12C - LogMiner]".



Flashback

- Verschiedene Anwendungsfälle:
 - Vergleich Jetzt zu Vorher
 - Auditing
 - › Vorsicht, Undo Daten werden irgendwann überschrieben.
 - Troubleshooting
 - › Was hat sich geändert?
 - Viele mehr...
- Übrigens: Flashback Database ist ein anderer Mechanismus.
 - Nutzt die Flashback Logs



Takeaways



Zum Mitnehmen

- Die Oracle Datenbank hat sehr ausgefeilte Transaktionsmechanismen.
- Es lohnt sich, sich tiefer damit zu beschäftigen. Planung, Konzeption und Troubleshooting sind dann einfacher.
- Viele Weitere Informationen in der Oracle Dokumentation

Describes Oracle Database architecture and essential topics for database administrators and developers.

besonders im Concepts and Architecture Guide

- Wenn Sie uns besuchen möchten: Dell Software Stand
 - Dritter Stock vor Raum Tokyo.

