

Gedächtnisprotokoll zur Prüfung 01672/01664

(Datenbanken II und Implementierungskonzepte für Datenbanksysteme)

Prüfer: Prof. Dr. Güting
Beisitzer: Simone Jandt
Datum: 23.11.2009
Dauer: ca. 25 min

Obwohl ich mich sofort nach meinen Prüfungen bemüht habe meine Erinnerungen niederzuschreiben, ist dieses Protokoll bestimmt unvollständig und auch nicht in korrekter Reihenfolge wiedergegeben. Dennoch der Appell an alle Prüflinge- selbst wenn einem nur noch zwei oder drei Themen einfallen, ist dies für die Lernenden besser als nichts – schreibt kurz auf, was Euch noch einfällt und schickt es an die Fachschaft. Auch wenn Eure Prüfung denen der anderen Protokolle entspricht, so wissen die Prüflinge zumindest dennoch, dass die Themen immer noch dieselben sind! Die folgenden Antworten habe ich nicht gegen das Skript geprüft und müssen daher nicht korrekt sein:

Implementierungskonzepte für Datenbanksysteme

Schichten des DBMS (von unten nach oben)

- Datenbank / Systemkatalog
- Geräte- und Speichermanager
- Zugriffspfadmanager /Recordmanager
- ... (vergleiche Skript Bild 1.3 (S. 13) – zu jeder Teilkomponente sollte eine kurze Erläuterung erfolgen (z.B. „Recovery Manager zuständig für UNDO von Transaktionen oder auch Restore nach Systemabsturz“, oder „Systemkatalog Manager verwaltet den Systemkatalog (Metadaten/Statistiken...))

Ablauf einer Anfrage:

- Lexikalische / syntaktische / semantische Analyse (+Autorisierung)-> interne Darstellung
Ab hier werden meine Erinnerungen leider etwas sehr ungenau, dies liegt wohl vor allem daran, dass ich mich hier ziemlich verzettelt habe und Prof. Dr. Güting mich öfters wieder einsammeln durfte.

Tipp: Wenn man Begriffe erwähnt, so sollte man damit rechnen, dass diese hinterfragt werden ☺ .

Ich erwähnte im Zusammenhang mit den elementaren Umformungen, die konjunktive und disjunktive Normalform. Hier kamen dann Nachfragen, wie diese denn aussähen (-> *Notation*) und für welche Teile der Abfrage man sie verwendet (-> *Selektion*).

In einem weiteren Erläuterungsversuch erwähnte ich den Operatorenbaum – hier kam dann die Rückfrage, wie dieser denn aufgebaut sei (-> *innere Knoten Operatoren, Blätter Relationen*).

Im Anschluss folgte eine Aufzählung der möglichen Optimierungsmaßnahmen (-> *1. Selektionen zusammenfassen, 2. Selektionen möglichst früh ausführen lassen, 3. Selektionen die direkt auf einen Join folgen, gemeinsam mit dem Join ausführen, 4. Projektionen ebenfalls möglichst nach vorne ziehen, jedoch nach Selektionen und 5. Relationen mit kleiner Ergebnismenge nach unten in den Baum verschieben, so dass nur mit möglichst kleinen Ergebnismengen gerechnet werden muss*).

Irgendwann erwähnte ich dann wohl noch, dass es verschiedene Möglichkeiten gibt Joins durchzuführen, diese sollte ich dann noch genauer ausführen und jeweils kurz erläutern (*in etwa meine Antwort: -> Nested-Loop-Join (jede Seite/jeder Datensatz der beiden Relationen werden miteinander kombiniert (vier For-Schleifen), Index-Nested-Loop-Join (Äußere Relation (evtl. nicht indiziert) wird vollständig durchlaufen und mit Hilfe eines Index der inneren Relation zugeordnet), Sort-Merge-Join (Beide Relationen sind/werden vorsortiert und miteinander verschmolzen), Simple Hash Join (Nur die Variante mit ausreichend Systempuffer erläutert: Kleinere Relation per Hashfkt. in einem Segment des Systempuffers verteilen und mit Hilfe der selben Hashfkt. angewandt auf die andere Relation in den Buckets nach Matchingpartnern suchen.) Grace-Hash-Join (beide Relationen partitionieren – und das eben genannte Verfahren (Simple Hash Join) jeweils auf die einzelnen zusammengehörigen Partitionen anwenden).*

Zwischendrin irgendwann die Frage: Was ist bei der Ermittlung der günstigsten Kosten die Schwierigkeit? (-> *Meine erste Antwort war, dass es sehr viele verschiedene Möglichkeiten gibt und das Optimum daher nur schwer zu ermitteln ist. Gegenfrage: Also mit genügend Rechenzeit ließe sich im Vorfeld ein optimaler Auswertungsplan erstellen? Mit ein paar weiteren Nachfragen fanden wir dann doch zusammen und die ursprünglich gewünschte Antwort war, dass die Schwierigkeit darin besteht, dass die Kostenschätzung nur eine Schätzung ist und auf Annahmen beruht, die auf Statistiken/Erfahrungswerten basieren (Anzahl Datensätze/Verteilung der Werte) und daher nur bedingt korrekt und optimal sein kann.*)

Zum Thema Kosten sollte dann noch kurz aufgezählt werden, woraus sich die Kosten zusammensetzen (-> *CPU (Rechenzeit im Hauptspeicher), notwendige Seitenzugriffe und schlussendlich Kommunikationskosten (bei verteilten Systemen zwischen den einzelnen Teilen des DBS (und nicht zwischen Server und anfragenden Client(!)).*

Datenbanken II

Serialisierbarkeit

Ein paralleles System von Transaktionen ist dann korrekt synchronisiert, wenn es serialisierbar ist, d.h. es mindestens eine gedachte serielle Ausführung derselben Transaktionen gibt, die (a) zur selben Ausgabe führt und (b) die Datenbank im selben Zustand hinterlässt.

Was ist eine Transaktion (und Nachfrage nach ACID)?

Eine Reihe von Befehlen/Operationen, die entweder vollständig und korrekt oder gar nicht ausgeführt werden.

- A atomicity ganz oder gar nicht
- C consistency DB wird von einem konsistenten Zustand in einen anderen überführt
- I isolation Jede Transaktion soll so ablaufen, als ob sie alleine wäre, d.h. Zwischenergebnisse anderer Transaktionen dürfen nicht sichtbar sein.
- D durability Eine erfolgreich abgeschlossene Transaktion überlebt jeden später auftretenden Fehler des DBS.

Welche Probleme können auftreten?

- Phantom Read
Eine Anwendung liest z.B. alle Kunden ein und merkt sich, dass 10 Kunden mit dem Buchstaben A beginnen. In einem zweiten Schritt liest sie nur die Datensätze mit diesen Kundennamen erneut ein und liest nun 11 Datensätze, da in der Zwischenzeit eine andere Transaktion einen weiteren Datensatz eingefügt hat.
- Lost Update
Zwei Transaktionen möchten ein gleiches Datum (Wert) um jeweils 10 erhöhen. Beiden lesen den ursprünglichen Wert 100 ein. Transaktion A rechnet $100+10=110$ und parallel ebenfalls Transaktion B ($100+10=110$). Beide schreiben ihr Ergebnis (110) nacheinander in die DB und obwohl 120 das Ziel gewesen wäre, steht nun 110 in der DB.
- Inkonsistente Sicht und hieraus folgend inkonsistente Datenbank
Eine Transaktion überweist von Konto A auf Konto B einen Betrag von 10€.
Erster Schritt Konto A wird um 10 € erleichtert
Zweiter Schritt Konto B wird um 10 € bereichert
Zwischen dem ersten und dem zweiten Schritt wird durch eine parallele Transaktion das Gesamtvermögen der Bank ermittelt und abgespeichert – hier sind die 10€ verloren gegangen, da nur die Abbuchung jedoch noch nicht die Gutschrift erfasst wurde.

Welche Möglichkeiten gibt es zur Vermeidung?

- präventiv/pessimistische Verfahren (z.B. 2 Phasensperrprotokoll / Zeitstempelverfahren)
- verifizierende/optimistische Verfahren (serialization graph certification, einige Verfahren die an dem Zeitstempelverfahren angelehnt sind)

Abhängigkeitsgraph?

Knoten stellen Transaktionen dar. Kanten bestehen zwischen zwei Transaktionen, wenn beide Transaktionen auf ein Datum zugreifen und mindestens eine von ihnen hierbei mit schreibendem Zugriff.

Arten von Sperren?

- X-Lock (exclusive - schreibender Zugriff)
- S-Lock (shared – lesender Zugriff)
- Intension-Locks auf den darüber liegenden Ebenen (IX, IS) (Hier kam dann kurz die Zwischenfrage, welche Ebenen es hier geben kann: Datensatz->Seite->Tabelle->Tablespace bzw. auch Indexseiten)
- Spezielle Locks auf z.B. Summenfeldern (Hot Spots)

Zwei-Phasen-Sperrprotokoll?

- Bevor auf ein Datum zugegriffen wird, muss ein Lock erworben worden
- Wenn der erste Lock freigegeben wird, darf keine weitere Sperre erworben werden.
- Spätestens mit dem Ende der Transaktion müssen alle Sperren freigegeben worden sein

Verschärfung?

- Preclaiming: Alle Sperren werden zum Anfang einer Transaktion erworben um z.B. Deadlocks zu vermeiden. Problem: Sperren sind hier meist nur auf übergeordneten Stufen möglich, da nur schwer im Detail festgestellt werden kann, welche Datenmengen betroffen ist.

- Strikte Zweiphasigkeit: Alle Sperren werden erst am Ende der Transaktion freigegeben (Vermeidet fortgesetzten Rollback)

Fazit

Auch ich kann Herrn Prof. Dr. Güting nur jedem empfehlen. Trotzdem ich immer sofort losgeplappert habe, was mir gerade so in den Sinn kam und auch recht häufig nachgefragt hatte, ob seine Frage in diese oder jene Richtung abzielt, blieb er sehr ruhig und gab mir auch außergewöhnlich viel Feedback.