

Architektur von Datenbanksystemen Praktikumsbericht 1

Benedikt Kappes
Tuan-Si Tran

22. Oktober 2011

1 Einleitung

Im ersten Praktikum ist das Ziel die IBM DB2 - Datenbank kennenzulernen. Die Installation des Datenbanksystems auf Linux ist schon als VMWare-Image vorhanden, welche auch in diesem Praktikum benutzt wird. Dabei werden in diesem Team sowohl ein Laptop mit mechanischer Festplatte als auch ein Laptop mit SSD-Platte eingesetzt.

Im ersten Schritt werden Überlegungen im Vorfeld zur Größe der initialen leeren Datenbank und der zuschreibenden Daten gemacht. Als nächstes wird die Datenbank erzeugt und die Daten mit einer Java-Anwendung und unter Verwendung des JDBC-Treibers in die Datenbank eingetragen. Von Interesse sind hierbei Zeitdauer und die tatsächliche Größe der Datenbank bzw. der geschriebenen Daten. Im letzten Abschnitt wird die Auswirkung betrachtet, wenn beim Anlegen der Tabelle die Fremdschlüsselindexte nicht sofort mit erzeugt werden.

2 Vorbereitung

Im Laufe des erstens Praktikums muss die Datenbank mit Testdaten befüllt werden. Im Vorfeld soll dabei die Größe der Datenbank geschätzt werden, nachdem die Daten erzeugt und gespeichert worden sind. Dabei wurde mit Hilfe der Tabelle (siehe Anhang 6.1) die Größe eines Datensatzes ermittelt und mit der Anzahl der zu schreibenden Datensätze multipliziert.

Die folgende Auflistung zeigt die Hochrechnung für die drei Tabellen **Kunde**, **Produkt** und **Bestellung** an (siehe Tabelle 2.1).

Kunde			
Spaltenname	Typ	Byte Count	
KNR	Integer	4 Byte	
KNAME	Char	20 Byte	
KVORNAME	Char	15 Byte	
100 x 39 Byte			= 3 900 Byte
Produkt			
Spaltenname	Typ	Byte Count	
PID	Integer	4 Byte	
PRODUKTNAME	Char	20 Byte	
PREIS	Decimal(6,2)	4 Byte	

1000 x 28 Byte			= 28 000 Byte
Bestellung			
Spaltenname	Typ	Byte Count	
BID	Integer	4 Byte	
KNR	Integer	4 Byte	
PID	Integer	4 Byte	
DATUM	Date	4 Byte	
STATUS	Smallint	2 Byte	
750 000 x 18 Byte			= 13 500 000 Byte
Summe			= 13 531 900 Byte

Tabelle 2.1: Berechnung der Größe der Testdaten

Als Gesamtgröße für die reine Datenmenge wird in etwa 12,9 MB angenommen (bei 1024Byte = 1 KByte).

Fremdschlüsselindex

Wird der Fremdschlüsselindex nicht bei der Erzeugung der Tabelle angelegt, sollte das Eintragen der Daten schneller gehen. Nach unseren Erwartungen sollte das nachträgliche Erzeugen des Fremdschlüsselindex für die Tabelle *Bestellung* einige Zeit in Anspruch nehmen.

3 Durchführung

Über das Befehlszeilenfenster wurde die Datenbank *ArchDBS* erzeugt und den *Footprint* der leeren Datenbank ermittelt. Danach wurde das Java-Konsolenprogramm erstellt, das 100 Kundentupel, 1000 Produkttupel und 750 000 Bestellungen in die Datenbank eingetragen hat. Dabei wurde die Zeit gemessen, die für die Erstellung der 751100 Tupel benötigt wurden. Damit die Erzeugung der Testdaten performant ablaufen konnte, wurde `autocommit` auf `false` gesetzt und `PreparedStatement`s verwendet. Ebenso war es nötig, die maximale Größe der Logdateien (`LOGPRIMARY` oder `LOGSECOND`) zu erhöhen, damit die Erzeugung der 750000 Bestellungen auch mitgeloggt werden können. Die Größe der Datenbank wurde erneut ermittelt und mit dem Startwert verglichen.

Im nächsten Schritt wurde eine zweite Datenbank mit den entsprechenden Tabellen angelegt, wobei auf Foreign-Keys-Indexe verzichtet worden sind. Erneut wurde die Zeit gemessen und mit dem Wert aus der vorherigen Messung verglichen. Als letztes wurden die Fremdschlüsselindexe über die gefüllten Tabellen erzeugt.

4 Ergebnisse und Auswertung

4.1 Datenbankgröße

Der gemessene Footprint unserer leeren Datenbank ArchDBS war 128 MB. Nach Abruf der Informationen mittels `list tablespaces show detail` wuchs die Größe der Datenbank auf 160 MB an. Bei genauerer Betrachtung setzt sich die leere Datenbank wie folgt zusammen (Tabelle 4.1):

	Syscatspace	Tempspace1	Userspace1	Systoolspace
Seiten insgesamt	24576	1	8192	8192
Seitengröße (Byte)	jeweils 4096 Bytes			
Summe(Seite*Größe)	167776256 Bytes = 163844 KB = 160,003 MB			

Tabelle 4.1: Größe der Datenbank bei 1KB = 1024 Byte

Gründe für die Vergrößerung der Datenbank nach Aufruf des Befehls könnten der Verwaltungsaufwand der Datenbank zur Ermittlung der Tabellendetails sein.

Nach der Erzeugung der Testdaten war die Datenbank 192 MB groß. Unsere errechnete Größe lag allerdings bei 172,9 MB (160 MB + 12,9 MB). Der Fehler beträgt somit knapp zehn Prozent. Die Differenz hat hauptsächlich eine Ursache, die darin liegt, dass wir lediglich die genaue Größe der reinen Datenmenge berechnet haben, aber dabei die Größe der Index-Daten für die Tabellen vernachlässigen. Nach genauerer Betrachtung ergeben sich für die Indexe folgende Größenwerte, die anhand der Formel aus [IBM-2010] errechnet worden sind.

	KUNDE	PRODUKT	BESTELLUNG
Byte-Count Index-Attribut	jeweils 4 Bytes		
Index-Key-Overhead	jeweils 9 Bytes		
Anzahl Zeilen	100	1000	750 000
(ByteCount + Overhead)*Zeilen*2	2 600	26 000	19 500 000
Summe	19528600 = ca. 18,6 MB		

Tabelle 4.2: Berechnung der Größe der Index-Daten

Dadurch beanspruchen die Daten insgesamt mehr Speicherplatz als in unserer Rechnung. Besonders interessant ist hierbei, dass der Speicherung der Indexe mehr Platz einnehmen, als die Daten an sich.

Werden alle vorhandenen Zahlen addiert, so kommen wir auf 160MB Footprint + 12,9MB Datengröße + 18,6MB Index-Größe = 191,5MB Datenbank-

größe und sind somit sehr nah an dem gemessenen Wert der gesamten Datenbankgröße.

4.2 Zeitliche Messungen

Folgende Tabelle stellt die gemessenen Werte dar:

	SSD-Platte	mechan. Platte
mit angelegtem ForeignKey-Index(ArchDBS)	2:25 Minuten	2:53 Minuten
ohne ForeignKey-Index(ArchDB2)	1:24 Minuten	2:20 Minuten

Tabelle 4.3: Messwerte für die Eintragungsdauer der Testdaten

In dieser Tabelle ist zu sehen, dass das Eintragen in die Tabellen der Datenbank ArchDBS wesentlich länger dauert. Die gemessenen Werte stimmen auch mit unseren Erwartungen überein. Beim Eintragen der Daten ohne Fremdschlüsselindex muss das Datenbanksystem nicht überprüfen, ob die Schlüssel für **Kunde** und **Produkt** existieren und spart dadurch schon jeweils schon 750 000 Vergleiche für **Produkt** und **Kunde**.

Das nachträgliche Erzeugen des Fremdschlüsselindex lief allerdings viel schneller ab als wir es erwartet haben. Die Aktion dauerte lediglich 4 Sekunden. Wir haben angenommen, dass alle Fremdschlüsselindexe nochmal angelegt werden müssen, ebenso für die 750 000 Bestellungen. Allerdings haben wir nicht beachtet, dass lediglich die Tabelle **Bestellung** Fremdschlüsselindexe auf die anderen beiden Tabellen besitzt. Dadurch werden maximal 1100 Keys überprüft und als Fremdschlüsselindex angelegt, was nur im Vergleich zu 750 000 nur ein kleiner Bruchteil ist. Ebenso wird mit anlegen der Tabellen **Kunde** und **Produkt** Primary-Key-Indexe angelegt. Infolgedessen findet die nachträgliche Suche nach den vorhandenen Einträgen in der Datenbank wesentlich schneller statt.

5 Zusammenfassung

In diesem Praktikum haben wir gesehen, dass eine leere initiale Datenbank bereits 120 MB einnimmt und mit einer Ausführung eines Systemsbefehls bereits auf 160 MB anwuchs. Das Eintragen der Testdaten ergab, dass diese Daten mehr Speicherplatz einnehmen, als es rein rechnerisch sein sollte, was darauf zurückzuführen war, dass die Speicherung der Indexe einen unerwartet großen Platz einnahm, der größer ist als die Testdaten selbst. Schließlich hat sich noch herausgestellt, dass das Eintragen der Daten in eine Tabelle ohne vorher angelegtem Fremdschlüsselindex wesentlich schneller läuft, da dort nicht abgeglichen werden muss, ob der Key auch existiert.

6 Anhang

6.1 Byte-Count der Datentypen

Table 4. Byte Counts of Columns by Data Type

Data type	VALUE COMPRESSION is active ¹	VALUE COMPRESSION is not active	
		Column is nullable	Column is not nullable
SMALLINT	4	3	2
INTEGER	6	5	4
BIGINT	10	9	8
REAL	6	5	4
DOUBLE	10	9	8
DECIMAL	The integral part of $(p/2)+3$, where p is the precision	The integral part of $(p/2)+2$, where p is the precision	The integral part of $(p/2)+1$, where p is the precision
DECFLOAT(16)	10	9	8
DECFLOAT(34)	18	17	16
CHAR(n)	$n+2$	$n+1$	n
VARCHAR(n)	$n+2$	$n+5$ (within a table)	$n+4$ (within a table)
LONG VARCHAR ²	22	25	24
GRAPHIC(n)	n^2+2	n^2+1	n^2
VARGRAPHIC(n)	n^2+2	n^2+5 (within a table)	n^2+4 (within a table)
LONG VARGRAPHIC ²	22	25	24
DATE	6	5	4
TIME	5	4	3
TIMESTAMP(p)	The integral part of $(p+1)/2+9$, where p is the precision of fractional seconds	The integral part of $(p+1)/2+8$, where p is the precision of fractional seconds	The integral part of $(p+1)/2+7$, where p is the precision of fractional seconds
XML (without INLINE LENGTH specified)	82	85	84
XML (with INLINE LENGTH specified)	INLINE LENGTH +2	INLINE LENGTH +4	INLINE LENGTH +3
Maximum LOB ³ length 1024 (without INLINE LENGTH specified)	70	73	72
Maximum LOB length 8192 (without INLINE LENGTH specified)	94	97	96

Abbildung 6.1: Angaben zum Byte-Count, vgl. [IBM-2011]

Literaturverzeichnis

[IBM-2010] IBM: *IBM DB2 9.7: Database Administration Concepts and Configuration Reference*. Handbuch, 09.2010, Seite 310f.

[IBM-2011] IBM: *DB2 Solution Information Center* URL: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0000927.html>, zuletzt besucht am 20.10.2011