**Praktikumsaufgabe 7**

**Aufbau Benchmark-Datenbank mit  
Leistungsmessungen**

Fach: Datenbanken und Informationssysteme

Semester: Wintersemester 15/16

Von: André Schlüß, Johannes Nowack, Timo Knufmann

**Teilaufgabe b): Schätzung der Größe für n-tps Datenbank**

Um eine annähernde Schätzung der Größe der Datenbank vornehmen zu können, wurden als erstes die Größen der einzelnen Datentypen bestimmt, welche mittels der Dokumentation des Datenbanksystems ermittelt wurden.

Folgende Datentypen und –größen sind von Relevanz:

* Integer: 4 Byte
* Char-String: 2 Byte pro Character. In den verwendeten Strings wurde UTF-8 als Kodierung gewählt. UTF-8 kann 1 – 4 Bytes groß sein. Es wird davon ausgegangen, dass nur Buchstaben, Zahlen und einfache Sonderzeichen verwendet werden. Diese sind in der „Basic Multilingual Plane“ von UTF-8 vorhanden, welche 2 Byte groß sind.

Diese Größen wurden mit der entsprechenden Anzahl in den jeweiligen Tabellen verrechnet, sodass sich die Größe pro Datensatz in einer Tabelle ergab. Anschließend wurde die Anzahl der Datensätze für n in den jeweiligen Tabellen mit einbezogen. Daraus ergibt sich folgende Tabelle:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Ein Datensatz: | Anzahl pro n |  | Größe für n: |
| Tabelle branches: |  | 100 | 1 |  | 100 |
| Tabelle accounts: |  | 100 | 100000 |  | 10000000 |
| Tabelle tellers: |  | 100 | 10 |  | 1000 |
| Tabelle history: |  | 50 | 0 |  | 0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| n= | 1 |  |  | Gesamt in Kilobytes: | 9766,69922 |

Wie unten rechts zu lesen, liegt die geschätzte Größe bei ca. **9.800 Kilobytes** für n = 1.

**Vergleich mit tatsächlicher Größe:**

Die tatsächliche Größe der Datenbank beträgt ca. **14.440 Kilobytes**. Aus der Differenz ergibt sich somit, dass die tatsächliche Größe um ca. 4.600 Kilobytes kleiner ist.

Bei der Ermittlung der Größe wurde als Engine InnoDB verwendet. Nach Umstellung auf MyISAM betrug die Größe der Datenbank laut dem DBMS nur noch 11.880 Kilobyte.

Die Veränderung des Speicherplatzbedarfs lässt sich darauf schließen, dass die beiden Engines die Datenbank uneinheitlich verwalten und daher unterschiedlich viel zusätzlichen Speicher benötigen.

Implizit bedeutet das, dass im Allgemeinen zusätzlicher Speicher für die Verwaltung der Datenbanken benötigt wird. Aus diesem Grund weicht die tatsächliche Größe der Datenbank letztlich von der geschätzten ab.

Diese Verwaltung schließt beispielsweise die Benutzer, Berechtigungen, Verwaltung der Primär- und Sekundärschlüssel ein.

**Teilaufgabe c): Laufzeitoptimierungen**

Die erste Messungen folgte auf der Entwicklermaschine, bei der ca. 1675 Sekunden für den Wert n = 10 benötigt wurde. Im Folgenden veranschaulichen Optimierungen am Programmcode die veränderten Laufzeiten.

**Optimierung 1: Verwendung von „Prepared Statements“:**

Die Idee bei der ersten Optimierung war, dass die Datenbank im Vorhinein bereits den SQL-Ausdruck kennt, lediglich müssen die Werte noch übertragen werden.

Durch die Optimierung erreicht man eine geringere Netzlast und die Überprüfung der Syntax erfolgt ausschließlich einmal. Im Gegensatz zu Beginn, als die Durchsicht für sämtliche SQL-Statements stattfand. Die Zeit für den Wert n = 10 betrug auf gewertetem System zurzeit 1215,18 Sekunden.

**Optimierung 2: Deaktivierung von „Auto-Commit“ im Quelltext:**

Der Grundgedanke der zweiten Optimierungen beruhte auf der Sachlage, dass das DBMS Zwischenspeicher verwendet, bei dem Änderungen jedoch nicht verzeichnet werden. Der Zwischenspeicher muss für jeden Commit neu angelegt werden und beim Ausführen dessen werden Veränderungen erst übernommen.

Als Resultat muss die Zwischenspeicherung angesichts der Deaktivierung nur einmalig für eine gewisse Menge an Statements in die Datenbank übernommen werden. Durch dieses Mittel soll eine beschleunigte Verarbeitung erfolgen. Die Zeit für n = 10 ergab auf gewertetem System 412,567 Sekunden.

**Optimierung 3: Änderungen an Datenbankeinstellungen**

Das Leitmotiv der dritten Optimierung basierte auf die Beschleunigung der Laufzeit durch Veränderungen von Parametern der Datenbank. In Erwägung wurden verschiedene Puffergrößen, Dateigrößen oder auch die Deaktivierung des Transaktionslogs gezogen.

Als Ergebnis lässt sich darstellen, dass angesichts größere Anzahl an zur Verfügung stehenden Ressourcen, Vorgänge schneller ausgeführt und bearbeitet werden können. In Folge dessen betrug die Zeit für den Wert n = 10 auf gewertetem System 209,089 Sekunden.

**Kommentierter Programmcode**