



Übung zur Vorlesung
Einsatz und Realisierung von Datenbanksystemen im SoSe14

Moritz Kaufmann (moritz.kaufmann@tum.de)
<http://www-db.in.tum.de/teaching/ss14/impldb/>

Blatt Nr. 8

Aufgabe 1 Geben die Relation Klausur:

MatrNr	Vorbereitungszeit	Note
1	150	1.7
2	70	2.7
3	450	2.0
4	180	1.7
5	2500	1.3

- Formulieren Sie die Anfrage, die die MatrNr in der Skyline für die Attribute Vorbereitungszeit und Note erzeugt (kleiner ist jeweils besser) in SQL mit Hilfe des Skyline Operators.
- Formulieren Sie die Anfrage in SQL ohne Skyline Operator.
- Bestimmen Sie das Ergebnis der Anfrage.

a) SQL mit Skyline:

```
select MatrNr from Klausur k
skyline of k.Vorbereitungszeit min, k.Note min
```

b) SQL ohne Skyline:

```
select MatrNr from Klausur k
where not exists (
    select * from klausur dom
    where
        dom.Vorbereitungszeit <= k.Vorbereitungszeit and
        dom.Note <= k.Note and (
            dom.Vorbereitungszeit < k.Vorbereitungszeit or
            dom.Note < k.Note)
)
```

c) Ergebnis:

- 1) Ist in Skyline (Kann in Vorbereitungszeit nur von MatrNr 2 dominiert werden, dort ist aber Note schlechter)
- 2) Ist in Skyline (Minimum für Vorbereitungszeit)
- 3) Ist nicht in Skyline, dominiert von MatrNr 1
- 4) Ist nicht in Skyline, dominiert von MatrNr 1
- 5) Ist in Skyline (Minimum für Note)

Aufgabe 2

Schätzen Sie die Größe der Userdaten von Zalando für ein ganzes Jahr. Mögliche Parameter in der Schätzung sind Umsatz pro Bestellung, Größe einer Bestellung, etc.

Berechnen Sie zusätzlich noch die Größe der Produktdatenbank zusammen mit Kommentaren und Bewertungen.

Auf der Zalando Webseite und Wikipedia Artikel finden wir folgende Information:

1. Anzahl Artikel: 150.000 http://www.zalando.de/presse_zahlen-und-fakten/

2. Umsatz: 1,8 Mrd €

Für die Berechnung der Userdaten gehen werden folgende Relationen die wichtigsten sein: *Nutzer* und *Bestellungen*. Für das Schema der Usertabelle würden die Felder Kundennummer(4 Byte), ein Name(30 Byte), einem Geburtsdatum(4 Byte) und eine Adresse (70 Byte) Sinn machen. In Summe sind dies *108 Byte pro Kunde*.

Jede Bestellung besteht mindestens aus einer Id (jeweils 8 Byte) und einem Datum(4 Byte). Jeder Artikel der in der Bestellung vorkommt wird dann als Tupel ProduktId (8 Byte), Anzahl (4 Byte), Preis (8 Byte) und BestellungsId (8 Byte gespeichert). Als Schätzwert benutzen wir 3 Artikel pro Bestellung. Daraus folgt $8+4+3*(8+4+8+8) = 96$ *Byte pro Bestellung*.

Um die endgültige Größe der Userdaten zu schätzen, fehlen noch die Anzahl der Bestellungen und die Anzahl der User. Da die Anzahl der User nicht verfügbar ist benutzen wir einfach *20 Mio* als Schätzwert. Zur Schätzung der Anzahl von Bestellungen gehen wir von einem durchschnittlichen Umsatz von 45 €pro Bestellung aus. Zusammen mit dem Umsatz von 1,8 Mrd €ergibt dies 40 Millionen Bestellungen.

Die Gesamtgröße der Userdatenbank ist somit ungefähr $20.000.000*108 + 40.000.000*96 = 6.000.000.000$ *Byte* = 5,6GB

Um die Produktdatenbank zu berechnen müssen wir uns auf mehr Schätzungen verlassen. Zentral sind auch hier wieder zwei Relationen: *Artikel* und *Kommentare*. Die wichtigsten textit pro Artikel werden *Beschreibung, Anzahl, Preis, Name, etc* sein. Die Tabelle wird wahrscheinlich von der Beschreibung dominiert werden, aus diesem Grund nehmen wir einfach einen Schätzwert für die Gesamtgröße pro Artikel (1024 Byte). Ähnlich ist es bei den Kommentaren pro Artikel, hier wird die Größe von dem Kommentar selbst dominiert (zumindest im Durchschnitt, einzelne Kommentare werden auch relativ wenig Inhalt haben). Hier gehen wir von einem Schätzwert von 320 Byte pro Kommentar aus. Im Durchschnitt gehen wir auch von 3 Kommentaren pro Artikel aus, also insgesamt 960 Byte Kommentardaten. Insgesamt ergibt dies ein Datenbedarf von $1024 + 960 = 1984$ Byte pro Artikel. Damit verbraucht die Produktdatenbank $1984 * 150.000 = 297600000$ Byte which is 283 MB.

Aufgabe 3

HyPer schafft 120.000 Transaktionen pro Sekunde. Pro Transaktion werden 120 Byte in die Log geschrieben. Berechnen Sie den benötigten Durchsatz zum Schreiben der Log.

Die Datenbank läuft für einen Monat und stürzt dann ab. Es wurde kein Snapshot erstellt. Berechnen Sie die Recoveryzeit. Gehen Sie davon aus, dass die Recovery durch die Festplatte limitiert ist (100 Mbyte / s). Wieviel Log Einträge werden pro Sekunde reconvert?

Durchsatz = $120.000 * 120 = 14400000 = 13,7$ MBytes/s.

LogEinträge = $120.000*60*60*24*30=31104000000$

LogGröße = $\text{LogEinträge}*120 = 33$ TB

RecoveryZeit = 4,1 Tage

RecoveryDurchsatz = 873813 tx / s.

Aufgabe 4

In traditionellen Datenbanksystemen sind die Festplatte und der Buffermanager oft der Hauptgrund für Performanceengpässe. Wie ändert sich dies in Hauptspeicherdatenbanken, wo sind die neuen Flaschenhälse?

Der Unterschied zwischen traditionellen Datenbanksystemen und Hauptspeicherdatenbanken ist, dass wir in der Speicherhierarchie ein paar Stufen nach oben gehen. Hauptspeicher ist teurer, aber gleichzeitig auch schneller und hat eine geringere Latenz. Genauso wichtig ist aber auch, dass die Daten nun alle in einem Adressraum liegen. Bei der Nutzung von Festplatten muss das Datenbanksystem explizit die Daten von der Festplatte in den Speicher laden. Beim Hauptspeicher ist der Wechsel zwischen RAM, L3 und L1 Cache transparent für das System. Das heißt ein Buffermanager wird nicht mehr benötigt. Auch wenn der Wechsel zwischen den verschiedenen Hauptspeicherhierarchien für das System nicht explizit sichtbar ist, so ist es doch in der Performance bemerkbar. Der Latenzunterschied zwischen RAM und L3 ist ähnlich groß wie zwischen Festplatten und RAM. Die Datenbank muss nun so strukturiert werden, dass möglichst viele Operationen auf Daten in den schnelleren Speicherschichten ausgeführt werden. Ein weiterer neuerer Flaschenhals sind die Lockingverfahren. Im Vergleich zu einfachen Operationen wie Lesen, Schreiben, Addition, etc. ist ein Lock zu erstellen viel teurer. Viele Hauptspeichersysteme versuchen daher Locks zu vermeiden.

Gruppenaufgabe

Wie ändert sich die Bedeutung des Redo-Log und Undo-Log in Hauptspeicherdatenbanken im Vergleich zu klassischen Datenbanken? Wo werden sie gespeichert?

Da die Daten nicht mehr auf der Festplatte gespeichert werden, müssen sie (falls es keine Snapshots gibt) bei einem Neustart komplett auf Basis des Redo-Logs wiederhergestellt werden. Da nie die Daten einer nicht commitierten Transaktion auf der Platte landen wird die Undo-Log nur benötigt um im Fall eines Aborts die Änderungen der Transaktion zurückzusetzen. Außerdem reicht es aus die Redo Log Daten erst beim Commit zu schreiben. Aus diesem Grund sind die Daten der Undo-Log nur zur Laufzeit der Transaktion notwendig und müssen nicht auf die Festplatte geschrieben werden. Während der Wiederherstellung der Datenbank können dann einfach alle Redo Log Einträge wiederhergestellt werden.