



8. Übungsblatt

(Testwoche: 8. - 10. Juni 2010)

Einführung in Datenbanksysteme
Datenbanken für die Bioinformatik

Heinz Schweppe, Katharina Hahn

Aufgabe 1 (B-Bäume)

4 Punkte

Betrachten Sie ein DBMS mit folgenden technischen Daten:

- Die Blockgröße ist 2KB (also 2048 Byte).
- Verweise auf Datensätze bzw. Knoten sind 12 Byte groß.
- Ein Blockheader ist 56 Byte groß.

Es existiert ein B-Baum, bei der der Suchschlüssel 8 Byte groß ist. Bestimmen Sie die Höhe des B-Baums, wenn insgesamt 1 000 000 Datensätze in diesem indexiert sind.

Von der Blockgröße bleiben nach Abzug des Headers noch 1992 Byte Platz übrig für die Suchschlüssel und Verweise. Ein Suchschlüssel ist das Element mit dem verglichen wird und nimmt 8Byte Platz ein. Ist das Gesuchte Element kleiner als der Suchschlüssel steigt man in den B-Baum anhand des Verweise (12 Byte groß) eine Ebene tiefer. In jedem Knoten gibt es n Suchschlüssel und n+1 Verweise. Damit subtrahiert man von 1992 die Größe eines Verweises → 1980 Byte und teilt dies durch 20 Byte (8 Byte für den Schlüssel und 12 Byte für den Verweis) → 99 Schlüssel und 100 Verweise passen in einen Knoten.

Für den Fall das der Baum maximal belegt ist:

Baum Höhe 1: 99 Elemente

Baum Höhe 2: $99 + 100 \cdot 99 = 9999$ Elemente

Baum Höhe 3: $99 + 100 \cdot 99 + 10.000 \cdot 99 = 999.999$ Elemente

Baum Höhe 4: $99 + 100 \cdot 99 + 10.000 \cdot 99 + 1.000.000 \cdot 99 = 99.999.999$ Elemente

Somit ist die Höhe des Baumes für 1.000.000 Elemente für einen maximal belegten Baum 4.

Aufgabe 2 (Indexierung, B-Bäume)

4 Punkte

In einer Datenbank wird eine typische Anfrage auf eine Tabelle häufig durchgeführt. Diese Anfrage liefert durchschnittlich 100 Elemente zurück. Die verwendete Festplatte hat eine maximale Zugriffszeit (seek + latency) von 8ms und einen Datendurchsatz von mindestens 15MB/s. Die Blockgröße beträgt 8KB und jeder Block ist zu mindestens 80% gefüllt.

Berechnen Sie, ab welcher Gesamtanzahl von Datensätzen in der Tabelle die Benutzung eines B-Baumes effizienter ist als eine sequentielle Suche. Schätzen Sie dabei alle Berechnungen nach oben ab und gehen Sie von einer Höhe des Baumes von 3 aus.

Wir schätzen zur Berechnung ab, dass ein Datensatz = 64 Byte Daten entspricht.

Eine Höhe von 3 entsprechen 4 Zugriffe pro Datensatz: 3 innerhalb des Baumes, um die Block-ID zu finden und ein Zugriff, um den Block selbst zu lesen. Daraus ergeben sich bei 100 Datensätzen 400 Random-Zugriffe. Die Anfrage zu beantworten dauert also durchschnittlich:
 $400 \cdot (8\text{ms} + 8\text{KB} / 15\text{MB} \cdot \text{s}) = 3200\text{ms} + 3200/15\text{ms} = 3413,33\text{ms}$

Wieviel Datensätze x sequentiell zu suchen dauert länger als 3413,33ms? Auf jedem Block sind $0,8 \cdot 128$ Datensätze gespeichert.

$$8\text{ms} + (x / (0,8 \cdot 128)) \cdot 8\text{KB} / 15\text{MB} \cdot \text{s} > 3413,33\text{ms}$$

$$8ms + (x/102,4 * 8/15 * ms) > 3413,33ms$$

$$x/102,4 * 8/15 * ms > 3406,93ms$$

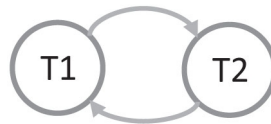
$$x > 654132$$

Aufgabe 3 (Konfliktserialisierbarkeit)**9 Punkte**

Gegeben sind die drei Schedules für die Transaktionen T1 und T2 (sowie T3 in Aufgabenteil c). Überprüfen Sie, ob diese serialisierbar sind. Geben Sie alle **Konfliktpaare** an und zeichnen Sie jeweils den entsprechenden **Konfliktgraphen** auf. Geben Sie für jeden konfliktserialisierbaren Ausführungsplan (Schedule) einen **äquivalenten seriellen Plan** an, z.B. T3 → T1 → T2.

- a) r1(a), w1(a), r1(b), w1(b), r2(b), w2(b), r2(c), w1(c)

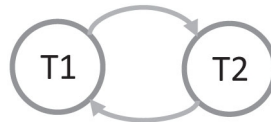
Konfliktpaare: (r1(b), w2(b)), (w1(b), r2(b)), (w1(b), w2(b)), (r2(c), w1(c))
Konfliktgraph:



Nicht serialisierbar, da Zyklus.

- b) r1(c), r2(b), w1(a), w2(b), r1(b), r2(c), w1(b), w2(c)

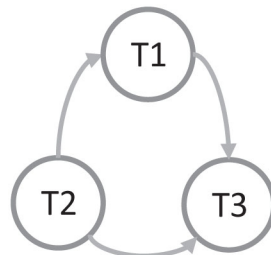
Konfliktpaare: (r1(c),w2(c)), (r2(b), w1(b)), (w2(b), r1(b)), (w2(b), w1(b))
Konfliktgraph:



Nicht serialisierbar, da Zyklus.

- c) r1(a), r2(b), r3(a), w3(a), r1(b), w1(b), r2(c), r1(c), r2(d), r1(d), w1(d), w3(c)

Konfliktpaare: (r1(a), w3(a)), (r2(b), w1(b)), (r2(c), w3(c)), (r1(c), w3(c)), (r2(c), w3(c)), (r2(d),w1(d))
Konfliktgraph:



Äquivalenter serieller Plan: T2 → T1 → T3

Aufgabe 4 (Konfliktserialisierbarkeit)

5 Punkte

Zeigen Sie, dass es serialisierbare Historien gibt, die ein Scheduler auf Basis des 2PL-Protokolls nicht zulassen würde. Zeigen Sie also, dass die Klasse der serialisierbaren Historien größer ist als die Klasse aller nach dem 2PL-Protokoll generierbarer Historien.

w2(a), r1(a), r2(a) → Kann so vom 2PL-Scheduler nicht erzeugt worden sein (no lock after unlock), da T1 ein a liest, das von T2 gesperrt ist. Dieser Schedule ist trotzdem serialisierbar (T2 → T1).