Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Prof. H.-J. Schek *A. Feßler, U. Röhm, H. Schuldt, R. Weber*

Informationssysteme IS-G SS 1997

Übung 9 Abgabe: 10.6.1997

Aufgabe 1: B*-Baum

Ein B*-Baum vom Typ (k, k^*) hat folgende Eigenschaften:

1. Jeder Weg von der Wurzel zu einem Blatt hat die gleiche Länge.
2. Jeder Knoten – ausser Wurzel und Blättern – hat mindestens k und höchstens $2k$ Einträge. Blätter haben mindestens k^* und höchstens $2k^*$ Einträge. Die Wurzel hat entweder maximal $2k$ Einträge, oder sie ist ein Blatt mit maximal $2k^*$ Einträgen.
3. Jeder Knoten mit n Einträgen, ausser den Blättern, hat $n+1$ Kinder.

Geben Sie für den B*-Baum eine Formel an, mit der man die obere und untere Schranke für die Höhe des Baums bei gegebenem k, k^* und N (der Anzahl der eingetragenen Schlüssel) bestimmen kann.

Aufgabe 2: Anfrageoptimierung

Bei komplexen Anfragen hat die Reihenfolge der Operationsausführung starken Einfluss auf die Ausführungszeit der Anfrage.

Gegeben sind die Relationen $R(\underline{a}, b, c)$, $S(\underline{c}, d, e)$ und $T(\underline{e}, f, g)$.

Berechnet werden soll die Anfrage

$$\pi_{b,f}\ \sigma_{a=12}\ \pi_{a,b,c,d,f}\ \sigma_{g=10,f=2}\ ((R \bowtie T) \bowtie S)$$

a) Zeichnen Sie den Operatorbaum für die obige Anfrage.

b) Spielen Sie Optimierer und formen Sie den obigen Operatorbaum algebraisch so um, dass ein möglichst günstiger Operatorbaum entsteht. Nehmen Sie Umformungen zunächst am Algebra-Ausdruck vor.

c) Nehmen Sie an, dass alle Attributwerte der Attribute a, f und g jeweils unter 10 Werten gleichverteilt und unkorreliert sind. Die Werte der Fremdschlüssel ($R.c$ und $S.e$) sollen ebenfalls in Bezug auf die Schlüsselwerte gleichverteilt sein. Die Kardinalitäten der Relationen seien mit $Card(R) = 10$, $Card(S) = 10.000$, $Card(T) = 100.000$ gegeben.
Berechnen Sie die Größe der erwarteten Zwischenergebnismengen (Anzahl Tupel) bei der Ausführung des nichtoptimierten und des optimierten Operatorbaumes!

Aufgabe 3: Nested Block Join

Noch einmal der Algorithmus des Nested Block Joins von Folie 8-7 in anderen Worten:
Stehen im Hauptspeicher m Seiten für die Berechnung des Joins zur Verfügung, reserviert der Nested Block Join-Algorithmus k Seiten für die innere Schleife und $m-k$ für die äussere. Die äussere Relation, nennen wir sie R, wird portionsweise in Blöcken zu $m-k$ Seiten eingelesen. Für jede dieser Portionen muss die komplette innere Relation S in Portionen zu k Seiten eingelesen werden. Alle Tupel der Relation R, die sich auf den eingelagerten $m-k$ Seiten befinden, werden dann mit allen Tupeln aus S in den k Seiten verglichen. Man kann pro Durchlauf das Einlesen einer Portion von k Seiten sparen, wenn die innere Relation im Zick-Zack-Verfahren durchlaufen wird, also abwechselnd vorwärts und rückwärts.

a) Seien b_R und b_S die Anzahl der Seiten, die R respektive S belegen. Ermitteln Sie die Kostenformel (bzgl. Seitenzugriffen) für den Nested Block Join.

b) Erläutern Sie, warum die Kostenformel für den Nested Block Join minimal wird, wenn $k = 1$ und R die kleinere der beiden Relationen ist.