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 8 Abgabe: 3.6.1997

Kostenschätzung einfacher Anfragen

In dieser Aufgabe sollen verschiedene Alternativen für den physischen Entwurf einer Oracle-Datenbank betrachtet und bewertet werden. Die Bewertung erfolgt dabei in zwei Schritten. Im ersten Schritt sollen Sie mittels Analyse des Speicherbedarfs die zu erwartenden Anfragekosten schätzen. Im zweiten Schritt können Sie Ihre Schätzungen mit Hilfe einiger Performance-Messungen überprüfen.

Daten

Wir betrachten eine Relation `R`, die zwei numerische Attribute A und B enthält. Die beiden Attribute haben jeweils die Länge 6 Bytes. In dieser Relation sind 50 000 Tupel gespeichert (rechnen Sie einfach mit *345 Datenbankseiten* à 2 KBytes). Die Attributwerte von A sind aus dem Intervall $[0 \ldots 999]$ mit Gleichverteilung der Tupel über die Attributwerte. Die Attributwerte von B sind aus dem Intervall $[0 \ldots 999]$, wobei 40 000 Tupel über das Intervall $[0 \ldots 99]$ gleichverteilt und 10000 Tupel über das Intervall $[100 \ldots 999]$ gleichverteilt sind.

Problemstellung

Voraussetzung für die Bewertung eines Entwurfs ist die genaue Kenntnis der Anwendungslast. Wir haben dieses in der Praxis nichttriviale Problem stark vereinfacht und wollen die verschiedenen Entwurfsalternativen nur bezüglich der drei folgenden Anfragen beurteilen:

```
Q1:  SELECT COUNT(*)
     FROM R
     WHERE A = 321 AND B = 50;
Q2:  SELECT COUNT(*)
     FROM R
     WHERE A = 321 AND B <> 50;
Q3:  SELECT COUNT(*)
     FROM R
     WHERE A <> 321 AND B = 50;
```


Entwurfsalternativen

Für den physischen Datenbankentwurf der Relation `R` stehen verschiedene Möglichkeiten offen. Im Rahmen dieser Übung sollen Sie die folgenden sechs näher betrachten:

1. Es ist lediglich die Relation `R` gespeichert (d.h. es gibt keinerlei Indexe).
2. Es existiert ein Index (B*-Baum) über dem Attribut A (Relationenname `RA`).
3. Es existiert ein Index (B*-Baum) über dem Attribut B (Relationenname `RB`).
4. Es existiert sowohl über A als auch über B ein Index (B*-Baum, Relationenname `RAB`).
5. Es existiert ein „clustered“ Index über A, d.h. die Relation ist bezüglich A physisch sortiert (Relationenname `RA$`).
6. Es existiert ein kombinierter Index (B*-Baum) über A und B, d.h. die Attributwerte werden konkateniert und als ein Wert im Index gespeichert (Relationenname `RC`).

Alle Indexe ausser Alternative 5 seien mittelbar (indirekte B*-Bäume). Nehmen Sie an, dass jede Relation und jeder Index jeweils einen eigenen Tablespace belegen, also getrennt gespeichert sind, wobei die Seitengrösse 2 KBytes sei mit 70% Füllgrad. Ein TID oder Zeiger im B*-Baum habe eine Länge von 6 Bytes.

Messen der Kosten

Die berechneten Kosten können mit Messungen am Datenbanksystem ORACLE verglichen werden. Dazu steht Ihnen eine WWW-Schnittstelle zur Verfügung, welche auf einfache Art und Weise die Messung der Kosten erlaubt. Mit dem Kommando

```
netscape http://www-dbs.ethz.ch/is-cgi/uebung8.cgi
```

erhalten Sie diese graphische WWW-Oberfläche. Damit können Sie sowohl Ausführungspläne als auch die I/O- und CPU-Kosten für die verschiedenen Aufgaben ermitteln.

Aufgabe 1: Indexhöhen bestimmen

Ermitteln Sie zuerst die Höhe der Indexe und die nötige Anzahl Zugriffe für die Suche im Index. Verwenden Sie die Werte 10 und 500 für die Suche von A und B mit den Entwürfen und tragen Sie ihre Resulate in die untenstehende Tabelle ein. Vergleichen Sie ihre Messungen mit ORACLE, indem Sie die graphische Schnittstelle benutzen (beachten Sie die Angabe *Logical Read Operations*).

Berechnung

Entwurf	Höhe	$A = 10$	$B = 10$	$A = 500$	$B = 500$
R: kein Index					
RA: Index über A					
RB: Index über B					
RAB: Indexe über A und B					
RA$: Clustered Index					
RC: Kombinierter Index					

Messung mit ORACLE

Entwurf	Höhe	$A = 10$	$B = 10$	$A = 500$	$B = 500$
R: kein Index					
RA: Index über A					
RB: Index über B					
RAB: Indexe über A und B					
RA$: Clustered Index					
RC: Kombinierter Index					

Aufgabe 2: Berechnung der Kosten

Berechnen Sie für jede der drei Anfragen Q_1 bis Q_3 und jede Entwurfsalternative die geschätzten I/O-Kosten und tragen Sie die Werte in die untenstehende Tabelle ein. Mit der graphischen WWW–Schnittstelle können Sie als Hilfestellung die Ausführungspläne ORACLEs für die einzelnen Anfragen ermitteln.

	Anfrage		
Entwurf	Q_1	Q_2	Q_3
R: kein Index			
RA: Index über A			
RB: Index über B			
RAB: Indexe über A und B			
RA$: Clustered Index			
RC: Kombinierter Index			

Es ist nicht nötig, für alle Anfragen und alle Entwürfe eine vollständige Analyse durchzuführen. Durch Überlegung lassen sich einige Resultate auf andere Entwürfe übertragen.

Aufgabe 3: Messen der Kosten

Nachdem Sie die verschiedenen physischen Entwürfe analytisch begutachtet haben, sollen Sie die Richtigkeit Ihrer Überlegungen mit Hilfe von Leistungsmessungen überprüfen. Verwenden Sie wieder das WWW-Formular zu dieser Aufgabe, um die I/O-Kosten (*Logical Read Operations*) mit ORACLE zu messen. Tragen Sie die Resultate in die folgende Tabelle ein und vergleichen Sie die Werte mit Ihren Berechnungen. Wo gibt es Abweichungen? Können Sie diese Abweichungen erklären? Warum sind bei einigen Entwürfen die physischen I/O's selbst bei der ersten Anfrage kleiner als die logischen I/O's? Wie erklären sich die Werte des Entwurfs `RA$`?

	Anfrage		
Entwurf	Q_1	Q_2	Q_3
R: kein Index			
RA: Index über A			
RB: Index über B			
RAB: Indexe über A und B			
RA$: Clustered Index			
RC: Kombinierter Index			

Aufgabe 4: Algebraische Optimierung

Gegeben seien die beiden bereits vertrauten Relationen

`Produkte (`PNr`, Bez, Gewicht, Preis, Lagerort, Vorrat)` und
`Bestellungen (`BestNr`, Monat, Tag, KNr, PNr, Menge, Summe, Status)`

sowie die folgende View

`BestInfo (BestNr, PNr, Bez, Gewicht, Menge)`
$:= \pi$[BestNr,PNr, Bez, Gewicht, Menge] (`Produkte` $\bowtie$ `Bestellungen`).

Es gelte die Integritätsbedingung:

π[PNr, Bez, Preis, Gewicht, Vorrat] (`Produkte`) =
π[PNr, Bez, Preis, Gewicht, Vorrat] (`Produkte` $\bowtie$ `Bestellungen`),

d.h. jedes Produkt wird mindestens einmal bestellt und es gibt nur Bestellungen für existierende Produkte.

Betrachten Sie nun die folgende Anfrage:

π[PNr, Bez] (σ[Gewicht > 100] (`BestInfo`))

Vereinfachen Sie diese Anfrage mit Hilfe der Äquivalenzregeln der Relationenalgebra und der oben angegebenen Integritätsbedingung, so dass die Anfrage ohne Join-Berechnung ausgeführt werden kann.