Institut für Informatik
Ludwig-Maximilians-Universität München

Prof. Dr. Stefan Conrad
Florian Krebs

24. Januar 2001

Data Warehouses

WS 2000/2001

Aufgabenblatt 3 / Online-Aggregation im Data Cube

Aufgabe 1.)

Eine häufige Anfrage in OLAP ist die Aggregation von Kennzahlen über mehr oder weniger große Bereiche des Datenwürfels. Zur Beschleunigung der Anfrage kann ein Datenwürfel als materialisierte Sicht vorberechnet und in einem MOLAP-System als mehrdimensionales (dichtes) Array gespeichert werden.

Ist z. B. der Kaufpreis die Kennzahl und die Region bzw. Kaufdatum die Dimensionen, so kann eine zweidimensionale Tabelle erzeugt werden, welche die aggregierten Werte der Verkäufe für jede Kombination aus Kaufdatum und Region enthält. Die Anfragen eines Analysten, der die Veränderung des Kundenverhaltens über die Zeit für verschiedenen Regionen ergründen will müssen so nicht immer wieder die zugrundeliegenden Rohdaten einlesen. Die Werte sind vielmehr bereits voraggregiert im Datenwürfel verfügbar und können so beliebig weiter verdichtet und analysiert werden.

Eine "Aggregations-Bereichsanfrage" bestimmt ein Hyper-Rechteck dieses Würfels und aggregiert die gespeicherten Werte (Summe aller Verkäufe in ... für den Zeitraum ...). Zur schnellen Bearbeitung dieser Anfragen, die ja möglichst auch bei sehr großen Datenmengen interaktiv ausgeführt werden sollten, betrachten wir im folgenden ein paar einfache Techniken die keinen weiteren Speicherplatz (zusätzlich zum Datenwürfel selbst) beanspruchen.

Sei A ein d-dimensionaler Würfel. Eine Zelle c = [c_1,..,c_d] (mit $0 \le c_i \le n_i - 1$) enthält die Kennzahl A[c]. Mit e : f beschreiben wir einen Bereich des Arrays bzw. Würfels d.h. die Menge aller Zellen mit $e_i \le c_i \le f_i$. SUM(A[e] : A[f]) ist dann die Summe aller dieser Zellen.

Betrachten wir die beiden Techniken "Prefix-Sum" (PS) und "Space-efficient Relative Prefix Sum" (SRPS) zunächst im eindimensionalen Fall.

- PS: Der Array enthält die Prefix-Summe der ursprünglichen Werte: $PS[j] = \sum_{k=0}^{j} A[k]$

- SRPS: A wird zunächst in Blöcke gleicher Größe unterteilt. Der "Anker" (am weitesten links stehende Zelle) von Block a : e enthält die Prefix-Summe von A, d.h. SRPS[a] = SUM(A[0] : A[a]). Alle anderen Zellen des Blocks a enthalten die "lokalen Prefix-Summen": SRPS[c] = SUM(A[a+1] : A[c]).

a) Die Grafik auf der folgenden Seite zeigt ein Beispiel. Vervollständigen Sie die fehlenden Werte für PS und SRPS (SRPS mit Blockgröße 3).

b) Führen Sie die Bereichsanfrage SUM(A[2] : A[5]) für alle 3 Beispiele aus. Geben Sie an, welche Zellen benötigt und wie diese verknüpft wurden.

c) Führen Sie das Update A[4] = 3 (auf den ursprünglichen Daten!) aus. Geben Sie das Ergebnisarray an und markieren Sie die betroffenen Zellen für PS und SRPS.

Original Array

4	6	2	1	7	5	3	2	8

SRPS Array

4	6	8	13	7	12			

PS Array

4	10	12	13	20	25			

d) Wie lassen sich die drei Techniken relativ zueinander bezüglich der Laufzeitkomplexität für Bereichsanfragen und Updates einordnen.

Die Grafik unten zeigt eine Anwendung von SRPS im 2-dimensionalen Fall. SRPS wird zuerst bezüglich der einen, dann der anderen Dimension angewendet. Zur Orientierung sind die Zellen A[2,4] und A[5,6] markiert. Die nachfolgenden Aufgaben beziehen sich auf die Grafik rechts.

e) Führen Sie die Bereichsanfrage SUM(A[2,4] : A[5,6]) aus. Geben Sie an, welche Zellen benötigt und wie diese verknüpft werden. Überlegen Sie dazu, welche Zellen die aggregierten Werte für welche Bereiche (Flächen) enthalten.

f) Führen Sie das Update (auf den ursprünglichen Daten!) A[4,2] = 1 aus. Geben Sie die betroffenen Zellen und deren neue Werte im SRPS Array an.

	0	1	2	3	4	5	6	7	8
0	3	5	1	2	2	4	6	3	3
1	7	3	2	6	8	7	1	2	4
2	2	4	2	3	3	3	4	5	7
3	3	2	1	5	3	5	2	8	2
4	4	2	1	3	3	4	7	1	3
5	2	3	3	6	1	8	5	1	1
6	4	5	2	7	1	9	3	3	4
7	2	4	2	2	3	1	9	1	3
8	5	4	3	1	3	2	1	9	6

Zeilen mit SRPS

	0	1	2	3	4	5	6	7	8
0	3	5	6	11	2	6	23	3	6
1	7	3	5	18	8	15	34	2	6
2	2	4	6	11	3	6	21	5	12
3	3	2	3	11	3	8	21	8	10
4	4	2	3	10	3	7	24	1	4
5	2	3	6	14	1	9	28	1	2
6	4	5	7	18	1	10	31	3	7
7	2	4	6	10	3	4	23	1	4
8	5	4	7	13	3	5	19	9	15

Spalten mit SRPS

	0	1	2	3	4	5	6	7	8
0	3	5	6	11	2	6	23	3	6
1	7	3	5	18	8	15	34	2	6
2	9	7	11	29	11	21	55	7	18
3	15	14	20	51	16	35	99	18	34
4	4	2	3	10	3	7	24	1	4
5	6	5	9	24	4	16	52	2	6
6	25	24	36	93	21	61	182	23	47
7	2	4	6	10	3	4	23	1	4
8	7	8	13	23	6	9	42	10	19