고속 트렌드 분석을 위한 차원 계층구조 동적 생성 기법*

김 이 은°, 임 윤 선, 김 명
이화여자대학교 컴퓨터학과
{oceaner°, lys96, mkim}@ewha.ac.kr

Dynamic Generation of Dimension Hierarchies for Fast Trend Analysis

Yi-Eun Kim°, Yoonsun Lim, Myung Kim
Department of Computer Science & Engineering, Ewha Womans University

요 약

OLAP의 주요 목표는 대용량 데이터를 다차원적으로 분석하여 그 결과를 신속하게 제공함으로써, 사용자의 의사 결정을 지원하는 것이다. 다차원 분석을 용이하게 하기 위해 차원 계층이 사전에 정의되고 표준화된 연산들이 사용되는데, 그러한 연산들로는 롤업, 드릴다운, 슬라이스, 다이스, 피봇을 들 수 있다. 그러나 분석과정에서 기존에 정의된 데이터의 그룹핑 이외의 다른 그룹핑 방식이 필요할 때가 있으며, 그러한 그룹핑으로 전체 데이터를 분석하는 것을 트렌드 분석이라고 한다. 트렌드 분석은 기업의 의사 결정에 매우 중요한 요소이나, 사전에 정의된 계층을 사용하는 것이 아니므로, 질의 처리 시에 트렌드 분석을 신속하게 하기는 어렵다. 본 연구에서는 트렌드 분석을 고속화 하기 위한 방안으로 동적 계층 구조 생성을 제안한다. 특정 차원 기준으로 특정 범위 값으로 합산된 범위 합을 구하기 위해, 기존에 연구되었던 프리픽스섬 방식을 분석하여 문제점을 제시하였고, 새로운 기법을 제안하였다. 또한 분석 시에 디스크 접근을 효율적으로 하기 위한 큐브 저장 방식을 제안하였다. 본 연구에서 제안한 방식으로 트렌드 분석을 하였을 때 접근해야 하는 디스크 블록 수도 계산하여 제안한 방법의 효율성을 검증하였다.

1. 서 론

OLAP은 데이터베이스 내에 저장된 대용량 데이터를 다차원적으로 분석하여 그 결과를 온라인으로 신속하게 제공함으로써, 사용자의 의사 결정을 지원하도록 하는데 그 목적이 있다[1]. 사용자의 다차원적인 질의 처리를 신속하게 하기 위해, 일반적으로 다양한 요약 정보를 미리 계산하여 큐브에 저장해 놓으며, 표준화된 연산들인 롤업, 드릴다운, 슬라이스, 다이스, 피봇 등의 기본 연산들이 사용된다.

롤업이나 드릴다운은 차원마다 사전에 정의되어 있는 계층 구조를 따라 가면서 상세 정보를 보거나 점차적으로 요약되는 정보를 보는데 유용하다. 슬라이스나 다이스는 차원마다 원하는 범위를 정하여 해당 데이터를 OLAP 큐브에서 추출하여 보는 연산으로 원하는 조건을 만족하는 데이터의 확인에 유용하다.

때에 따라서는 사전에 정의되어 있는 계층 구조 이외에 동적으로 계층 구조가 필요한 경우가 있다. 예를 들어, 기간 차원에 일, 주, 월, 분기, 년의 순서로 계층이 이미 형성되어 있으나, 분석 도중에 5일 단위 또는 10일 단위의 요약 정보를 파악하는 것이 유용할 수도 있다. 또한 연령 차원을 기준으로 데이터를 분석할 때, 3년, 5년, 10년 단위 등 동적으로 원하는 기간을 설정할 필요가 있을 때도 있다.

이러한 모든 단위의 계층을 사전에 정의하고 큐브를 생성한다면 데이터 폭발 현상을 막을 수가 없게 된다[2]. 그리고 분석 과정을 통해 필요성을 알게 된 단위를 전혀 무시한다면 그러한 단위의 요약을 질의 처리 시에 실행할 때 데이터 전체를 스캔해야 하므로 처리 시간이 크게 느려지게 된다.

본 연구에서는 큐브의 특정 차원에 대해 차원 계층 구조를 동적으로 생성하는 방법에 대해 연구하였다. 우선 동적으로 생성될 계층도 전체 큐브에 대한 범위 합을 미리 구해 놓는 것과 같은 의미를 가지므로, 범위 합을 구하기 위해 기존에 연구되었던 프리픽스섬(Prefix Sum) 기법을 활용하였다. 또한 분석 시에 디스크 접근 시간을 줄일 수 있도록 큐브를 저장하는 방법도 고려하였다. 그리고 이러한 프리픽스섬 방식의 응용과 큐브 저장방식에 따라 계층을 생성해 놓으면, 트렌드 분석 시에 디스크 블록 접근이 감소하는가에 대한 분석을 통해 제안한 기법의 성능을 검증하였다.

본 논문의 구성은 다음과 같다. 2절에서 범위 질의를 효율적으로 처리하기 위한 기존 연구들을 분석하고, 3절에서 동적 차원 계층 생성을 위해 프리픽스섬을 계산하고 저장하는 방법을 제안하며, 4절에서 제안한 방법의 유용성을 검증한 후, 5절에서 결론을 맺는다.

2. 기존 연구

범위 질의의 속도 향상을 위해 많은 연구가 진행되어 왔다. 대표적인 방법이 프리픽스섬을 저장해 두는 것이다[3]. 프리픽스섬은 범위질의, 특히 범위 사이의 합산 값을 신속하게 계산하기 위한 방법이다. 그러나 이 방법은 범위가 미리 정해지지 않은 모든 차원의 멤버별로 범위를 계산해야 하는 동적 계층 생성 방법에는 적합한 방법이 아니다. 또한 이 연구 이후의 후속 연구들은 셀 값이 자주 수정되는 경우 큐브 데이터 갱신 비용이 비싸다는 점 때문에, 동적으로 셀이 수정되는 경우의 프리픽스섬을 효율적으로 계산하는 연구에 초점이 맞춰져 왔다[4, 5]. 그러나 앞선 연구와 마찬

* 본 연구는 2003년 한국과학재단 우수여성과학자 도약지원연구사업(R04-2001-000-00191-0(2003))지원에 의해 수행되었음.

가지로 동적 계층 생성 방법에 대한 연구는 이루어지지 않았다.

3. 동적 계층 생성을 위한 프리픽스섬

이 절에서는 프리픽스섬 큐브의 기본 개념을 활용하여 트렌드 분석을 하고자 하는 차원에 동적 계층 생성을 하는 방법을 제시한다. 큐브의 셀은 자주 수정되지 않는다고 가정하였다.

(1) 동적 계층 생성을 위한 프리픽스섬 계산 방법

범위의 합계 계산을 신속하게 하기 위해 사전에 합계 값을 저장해 놓는 방법인 프리픽스섬은 수정 비용을 고려하지 않는다면 범위 합 계산에 매우 유용하다. d차원 큐브 A가 있다고 하자. $A[i_1, i_2, \ldots, i_d]$의 프리픽스섬 $P[x_1, x_2, \ldots, x_d]$는 다음과 같이 계산된다[3].

$$P[x_1, x_2, \ldots, x_d] = Sum(0:x_1, 0:x_2, \ldots, 0:x_d)$$
$$= \sum_{i_1=0}^{w_1}\sum_{i_2=0}^{w_2}\cdots\sum_{i_d=0}^{w_d} A[i_1, i_2, \ldots i_d]$$

예를 들어, 그림 1(a)와 같은 3차원 데이터 큐브가 있을 때, 그림 1(b)는 이 큐브의 프리픽스섬이 계산되어 저장된 것을 나타낸다 [3].

index	1	2	3	4	5	6	7	8
1-1	3	5	1	2	2	4	6	3
1-2	7	3	2	6	8	7	1	2
1-3	2	4	2	3	3	3	4	5
2-1	3	2	1	5	3	5	2	8
2-2	4	2	1	3	3	4	7	1
2-3	2	3	3	6	1	8	5	1

(a) 데이터 큐브

index	1	2	3	4	5	6	7	8
1-1	3	8	9	11	13	17	23	26
1-2	10	18	21	29	39	50	57	62
1-3	12	24	29	40	53	67	78	88
2-1	15	29	35	51	67	86	99	117
2-2	19	35	42	61	80	103	123	142
2-3	21	40	50	75	95	126	151	171

(b) 프리픽스섬

그림 1. 3차원 큐브의 원본데이터와 프리픽스섬.

이러한 프리픽스섬 방법은 트렌드 분석에는 그대로 적용하기 힘들다. 그 이유는 범위가 주어지지 않는 모든 차원의 멤버별로 각 범위를 계산해야 하므로, 동적 계층 생성 방법에는 효율이 매우 떨어진다. 따라서 이런 경우에는 범위의 합을 구할 가능성이 낮은 차원의 속성 별로 분석하기를 원하는 차원의 방향으로 프리픽스섬을 구하는 것이 매우 효율적이다. 그림 2는 그림 1의 (a) 원본 데이터 큐브에서 가로 차원은 동적 계층을 구할 숫자 차원이고, 세로의 두 차원은 멤버가 각각 2개와 3개인 차원이고, 각 멤버당 프리픽스섬을 범위 차원의 방향으로 저장한 예이다. 본 연구에서 제안한 멤버 별로 프리픽스섬을 한 것을 MPS (Member Prefix Sum)라고 부르기로 한다.

index	1	2	3	4	5	6	7	8
1-1	3	8	9	11	13	17	23	26
1-2	7	10	12	18	26	34	35	37
1-3	2	4	8	11	14	17	21	26
2-1	3	5	6	11	14	19	21	29
2-2	4	6	7	10	13	17	24	25
2-3	2	5	8	14	15	23	28	29

그림 2. 3차원 큐브의 MSP.

이 방법으로 프리픽스섬을 구하기 위해서는 우선 동적 계층을 만들 범위 차원이 어떤 것인지 알아야 한다. 3차원(상품, 상점, 시간) 큐브의 세 차원 중 시간 차원에서만 동적 계층을 적용한다면, 각 상품 차원과 상점 차원의 모든 멤버에 대해 시간 차원을 따라서 프리픽스섬을 만든다. 그림 2에서는 가로 차원이 시간 차원이고 세로의 두 차원이 상품과 상점이 된다.

이처럼 각 멤버 당 프리픽스섬을 구해 놓은 MPS는 차원이 높아질수록 숫자 차원의 범위 합산 값을 계산하는 비용이 줄어든다. 전체의 프리픽스섬이 범위 합산 값을 구할 때 계산하는 식은 다음과 같다. 우선 $s(j)$를 다음과 같이 정의하자.

$\forall j \in D,$

$$s(j) = 1, \text{ if } x_j = h_j$$
$$-1, \text{ if } x_j = l_j - 1$$

그러면 다음 식이 성립한다.

$\forall\ j \in D,$

$$Sum(l:h, l:h, \ldots l:h)$$
$$= \sum_{\forall x\{\zeta-1, h_j\}} \{(\Pi\, s(i)) * P[x_1, x_2, \ldots, x_d]$$

이 방법은 n차원에 대해서 $2n$번 만큼의 계산을 필요로 한다. 그러나 1개의 차원만이 범위가 주어진 경우에 멤버당 프리픽스섬을 구하는 방법은 범위 합산 값을 구할 때, 이번 프리픽스섬에서 이전의 값을 빼는, 단 2번의 계산만이 필요하므로 효율적이다.

여러 개의 차원으로 구성된 큐브에서 한 차원만 범위 차원일 때 동적 계층을 생성하는 방법은 다음과 같이 나타낸다. H_n이 큐브의 동적 계층이고 C_d는 각 차원의 멤버를 나타내며 C_{nr}은 범위 차원의 범위 r 만큼의 n번째 멤버들을 나타낸다.

$$H_n = P[C_{nr}, C_2, C_3, \ldots C_d] - P[C_{(n-1)}, C_2, C_3, \ldots, C_d]$$

(2) 동적 계층 생성을 위한 프리픽스섬 저장 방법

프리픽스섬을 구한 후에 동적 계층을 생성할 때 집계 연산 결과를 저장하는 방식에 따라 질의 처리 효율이 크게 차이가 나게 된다. 우선 동적 계층이란, 숫자 차원을 따라 정해진 범위대로 합계 계산한 값을 저장하는 것이므로, 숫자 차원의 멤버 값들이 여러 블록에 나뉘어 저장되어 있을수록 유리하다. 즉 숫자 차원의 같은 멤버가 다른 블록에 반복되지 않도록 숫자 차원을 기준으로 다른 차원의 셀 값들을 모아서 그림 3과 같이 저장함으로써 디스크 블록 접근 비율을 최소화 하도록 하였다.

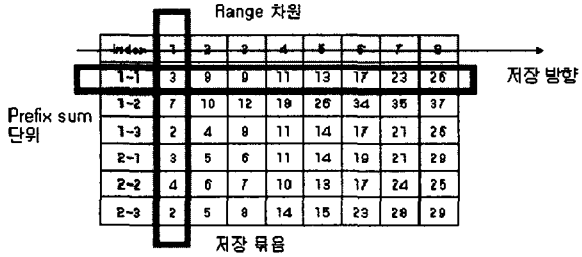

그림 3. MSP 저장 방법.

4. 실 험

먼저 MSP 방법으로 동적 계층을 생성하는 경우의 효율성을 분석하기로 한다. 본 논문에서는 각 차원의 멤버가 100개인 3차원으로 구성되어 있는 큐브에서 한 차원에 대해 범위 질의가 10단위로 발생하였을 경우를 실험하여 그림 4와 같은 결과를 얻었다. 이는 트렌드 분석을 하기 위해 계층을 생성한다고 하는 것은 큐브에 대해 특정 차원으로 범위 합을 구해나가는 것과 같은 것으로, 일반적 프리픽스섬보다 MSP가 항상 3/8배만큼의 비용이 들어 매우 효율적이라는 것을 알 수 있다.

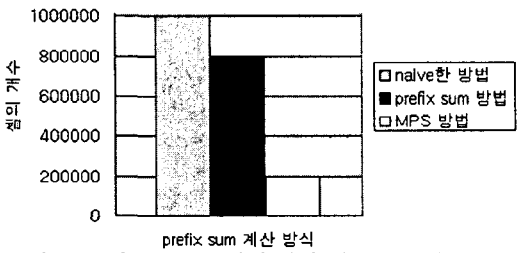

그림 4. 프리픽스섬 방법에 따라 접근해야 하는 셀의 개수

MSP를 본 논문에서 제안한 범위 차원을 고려하여 저장하였을 경우 디스크 접근이 얼마나 일어나게 되는지 분석하였다. 본 논문에서 실험은 3, 4, 5 차원에 대해 실험하였고, 접근 디스크 블록의 개수는 다음과 같은 식을 통하여 계산하여 그림 5와 같은 결과를 얻었다.

*접근 디스크 블록 수 = 계층 차원의 범위 접근 회수 * (한 블록 안에 들어가는 각 차원의 멤버 수)$^{n-1}$/전체 생성 디스크 블록 수*

그림 5의 그래프와 같이 본 연구에서 제안한 저장 방법은 데이터를 검색하기 위해 접근하는 디스크 블록의 수를 줄여 디스크 엑세스 시간을 현저히 절감시켰음을 알 수 있다.

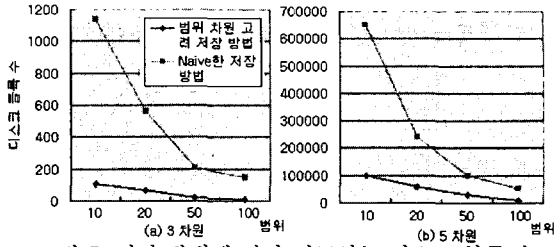

그림 5. 저장 방법에 따라 접근하는 디스크 블록 수

5. 결 론

트렌드 분석은 기업의 의사 결정을 내리는데 매우 중요한 요소이다. 본 연구에서는 이와 같은 트렌드 분석에 초점을 맞추어, 특정 차원에 대해 특정 범위 값으로 합산된 측정량을 알고자 하는 분석 시, 기본 연산처럼 효율적으로 결과를 도출하는 방법을 연구하였다. 기존에 연구되었던 프리픽스섬의 생성 방법을 개선하여, 프리픽스섬의 계산을 범위가 주어지지 않는 차원의 멤버들에 개별적으로 생성하도록 하였다. 또한 계층이 생성될 차원이 여러 개일 경우, 그 차원들을 제외한 나머지 차원들의 조합을 저장하고 범위 차원의 조합들을 차례로 저장해 나가도록 함으로써, 동적 계층을 만들 차원의 멤버 하나당 접근해야 하는 모든 차원 멤버의 조합을 저장하는 방법을 제안하였다.

본 연구에서는 제안한 계산 방법과 저장 방법을 사용하였을 때와, 이를 고려하지 않고 프리픽스섬을 사용했을 때의 동적 계층 생성에 있어서의 디스크 블록 접근을 비교 분석하였다. 계층의 범위가 작을수록 접근해야 하는 전체 셀의 수가 늘어나므로 본 연구에서 제안한 방법이 큰 효율을 보였고 전체적으로 디스크 블록 접근이 줄어드는 것을 알 수 있었다.

6. 참고 문헌

[1] Pilot Software, White Paper, "An Introduction to OLAP: Multidimensional Terminology and Technology," http://www.pilotsw.com/olap/olap.htm.

[2] E.Pourabbas, M.Raffanelli, "Characterization of hierarchies and some operators in olap environment," ACM Press, pp. 54-59, 1999, USA.

[3] C.-T.Ho, R.Agrawal, N.Megiddo, R.Srikant, "Range queries in OLAP datacubes," Proc. SIGMOD, 1997.

[4] Seok-Ju Chun, Chin-Wan Chung, Ju-Hong Lee, Seok, Lyong Lee, "Dynamic Update Cube for Range-Sum Queries," The VLDB Journal, 2001.

[5] S.Geffner,D.Agrawal,A. El. Abbadi, T. Smith, "Relative Prefix Sums: An Efficient Approach for Querying Dynamic OLAP Data Cubes," Proc. of ICDE, 1999.