OLAP에서 다차원 파일 구조를 사용한 큐브 생성 방법

김학경[0] 김진호 노희영
강원대학교 컴퓨터과학과
khk101@hotmail.com[0] {jhkim, young}@cc.kangwon.ac.kr

Effective Cube Computation using Multidimensional File Structure in OLAP

Hak-Kyoung Kim[0] Jin-Ho Kim Hi-Young Roh
Dept. of Computer Science, Kangwon National University

요 약

온라인 분석처리 시스템의 핵심 기술인 큐브를 효과적으로 산출하기 위한 많은 연구들이 이루어 졌다. 이러한 연구는 크게 온라인 분석처리 시스템의 결과 데이터를 저장하는 방식에 의해 MOLAP과 ROLAP으로 구분하여 이루어 졌다. 최근에 온라인 분석처리 시스템에서 큐브 산출에 대한 연구로 다중키 엑세스를 효율적으로 처리하는 다차원 파일 구조를 사용하여 집계 연산의 효율을 높이는 연구가 이루어졌다. 본 논문은 이러한 연구들을 바탕으로 다차원 파일 구조를 사용하여 효과적으로 큐브를 산출하고 결과 값을 미리 저장하는 일반적인 방법을 제안한다.

1. 서론

OLAP(On-Line Analytical Processing)은 '최종사용자가 다차원 정보에 직접 접근하여 대화 식으로 정보를 분석하고 의사결정에 활용하는 과정'[1]을 말한다. 즉, 사용자가 쉽게 이해 할 수 있으며 조작하기 쉬운 형태로 다양한 관점에서 분석의 대상이 되는 데이터를 빠르게 찾을 수 있어야 한다는 것이다. 이러한 OLAP시스템에서 의사 결정을 위한 다차원 질의를 효율적으로 처리하는 핵심기술을 CUBE(이하 큐브)라고 한다.

큐브는 데이터를 다차원 모델로 표현하기 위한 일반적인 방법이다. 큐브는 데이터를 분석의 대상이 되는 **사실**(혹은, 측정값)과 분석의 관점이 되는 **차원**으로 구성한다. 각 차원은 큐브의 축이 되고, 사실은 각 차원을 구성하는 항목들의 조합에 의해 만들어지는 셀에 저장이 된다.

큐브의 연산은 일반적으로 집계 연산이 많이 사용이 되며 집계연산은 비용이 크다는 단점을 가지고 있다. 이러한 단점을 해결하기 위하여 큐브에 저장이 될 집계 연산의 결과 값들을 미리 계산하여 저장해 두는 방법을 사용하며 집계 연산의 결과 값을 빠르고 효율적으로 산출하는 연구들이 많이 이루어지고 있다.

대표적으로 관계형 데이터베이스를 기반으로 sort와 hash를 사용하여 큐브를 산출하고 그 결과를 테이블로 저장하는 방법과[2][3][4] 다차원 데이터를 효과적으로 저장 관리 할 수 있는 다차원 파일구조를 저장 구조로 하는 방법이 그것이다[5][6]. 이러한 두 가지 방법의 차이에 의해 OLAP을 크게 ROLAP과 MOLAP으로 구분한다.

최근에는 큐브를 선계산하여 저장하는 방법 대신 실시간으로 집계연산을 수행하는 연구[7]와 MOLAP의 방법을 ROLAP에 적용하여 큐브를 산출하는 방법[6]에 대해서도 연구 되어지고 있다.

MOLAP의 큐브 산출 방법을 ROLAP에 적용하는 연구는 기존의 sort나 hash를 기반으로 한 방법에 비해 성능을 향상 시킬 수 있다는 연구 결과가 제시 되었다[6].

하지만 다차원 배열을 기반으로 하는 MOLAP의 기존 방법들은 편중된 분포를 갖는 데이터를 처리하는데 효율적이지 못하며, 또한 압축된 다차원 배열은 집계 연산이외의 다른 OLAP 연산들의 성능을 저하 시키는 단점이 있다.[7] 이러한 단점을 분리-포함 분할 다차원 파일 구조를 사용하여 집계 연산의 효율을 높이는 연구가 이루어 졌다[7]. 또한 기존의 방법에서 문제가 되는 저장 공간의 비용을 줄이기 위해 큐브의 결과를 선계산하여 저장하는 방식이 아닌 실시간으로 집계 연산을 처리하는 방법을 제안[7]하였는데 이것은 데이터의 크기가 크지 않고 차원이 적은 경우에 좋은 성능을 보이지만 대용량의 데이터를 처리해야하는 OLAP시스템에는 적합하지 않다.

이에 본 논문에서는 다차원 파일 구조를 사용하여 큐브의 결과를 미리 산출하여 저장하는 모델을 제시한다. 또한 이 모델을 기반으로 효율적으로 큐브를 산출하는 분할 영역 알고리즘을 제안한다.

2. 연구배경 및 동기.

여기서는 기존의 ROLAP/MOLAP 시스템들의 큐브 생성 방법들을 간략하게 소개한다. 또한 각 방법들의 문제점을 제시하고 이것을 개선하기 위한 본 논문의 동기를 밝힌다.

우선 기본적인 큐브 산출 방법에 대해 살펴보자. 예를 들어 Transactions라는 테이블이 Product(P), Date(D), Customer(C) 그리고 Sales(S)라는 에트리뷰트로 이루어져 있다고 가정하면 사용자는 다음과 같은 질의들을 통해 의사결정에 필요한 데이터를 찾는다.

- 상품(P)과 고객(C)에 대한 판매량(S)의 합.
- 날짜(D)별 고객(C)에 대한 판매량(S)의 합.
- 상품(P) 판매량(S)의 합.

여기서 P,D,C는 분석하고자 하는 관점인 차원이며 분석의 대상이 되는 S는 사실이 된다. 그리고 Transations 테이블을 사실 테이블이라고 한다. 예제에서와 같이 다차원 질의를 처리하기 위해서는 다중의 집계연산이 필요하며 이것을 간편하게 처리하기 위해 큐브라는 개념과 이것을 연산하는 방법이 제안되었다[2]. 큐브 연산은 차원이 n개일 때 2^n-1개의 집계 테이블을 생성하는 것을 의미하며 다음과 같이 SQL문을 확장하여 표현한다.

```
SELECT P,D,C, SUM(S)
FROM Transactions
CUBE-BY P,D,C
```

위와 같은 질의문의 결과는 PDC, PD, PC, DC, D, C, P, all의 집계테이블들을 생성하게 된다.

최종적으로 이렇게 산출된 집계 테이블의 저장구조에 따라 테이블의 형태로 저장을 하게 되면 ROLAP시스템이라고 하고 이

것을 다차원 배열의 형태로 저장하면 MOLAP시스템이 된다.

ROLAP 시스템에서 큐브 생성의 성능을 향상시키기 위한 많은 방법들이 제안 되었다. 이러한 방법들의 일반적인 공통점은 모든 집계 테이블을 노드로 하는 격자 탐색 그래프를(search lattice) 만들고 이것을 기반으로 큐브를 산출하는데 비용이 적게 드는 트리를 생성하여 이 트리를 메모리 사이즈에 맞게 분할하고 각각의 부분트리를 산출하여 합하는 방법을 사용한다. 이러한 방법들은 트리를 생성할 때 최소 비용의 부모를 찾거나, 정렬의 횟수를 줄이거나, 디스크 I/O의 횟수를 줄여 성능을 향상시키는 방법들이다[2][3][4].

MOLAP 시스템의 큐브 생성 방법은 기본적으로 ROLAP의 방법과 유사하나 다차원 분석에 용이한 다차원 배열을 사용하여 큐브의 결과를 산출하고 이것을 다차원 배열로 저장하는 방법을 사용한다.

MOLAP의 방법에 대한 연구에서 주목할 만한 것은 MOLAP의 큐브 산출 방법을 사용하여 ROLAP에 적용하는 방법이 그것이다. 이러한 방법은 기존의 ROLAP의 방법 보다 전체 비용면에서 효율적임이 알려졌다[6]. 하지만 일반적인 MOLAP의 방법에서 사용하는 다차원 배열은 편중된 분포의 데이터를 잘 처리하지 못하며, 데이터의 클러스터링을 파괴하여 집계 연산 이외의 다른 OLAP연산의 성능을 저하 시키는 단점이 있다. 이러한 단점을 보완하기 위하여 다차원 배열이 아닌 다차원 파일 구조를 사용하여 집계 연산의 성능을 향상시키는 연구가 최근 진행되었다[7]. 이러한 연구의 결과로 사용자의 질의가 있을 때마다 집계 연산을 처리하는 동적인 방법을 제시하여 MOLAP의 집계 연산 처리의 성능을 크게 향상 시키고 또한 OLAP 시스템의 중요한 이슈중 하나로 사전 집계의 문제점인 데이터 폭발에 대한 좋은 대안을 내놓았다[7]. 하지만 이와 같은 방법은 데이터의 크기가 적은 경우에는 매우 효과적인 반면 대용량의 데이터를 처리해야 하는 OLAP시스템의 경우에 대한 방법으로는 적합하지 않다. 본 논문에서는 다차원 파일 구조를 사용하여 집계 연산의 효율을 개선시킨 방법[7]을 보완하며 다차원 파일 구조를 사용하여 기존의 연구들과 마찬가지로 큐브를 미리 산출하여 저장하는 방법을 제안한다.

3. 다차원 파일 구조를 이용한 CUBE의 생성.

본 절에서는 다차원 파일 구조를 이용한 집계 연산처리에 대한 설명을 하고 이것을 이용하여 큐브를 생성하는 방법을 제안한다.

3.1 다차원 파일 구조와 집계 연산.

다차원 파일 구조는 기존의 단일 키 파일 구조로 처리하기 힘든 것을 용이하게 처리하기 위하여 개발되었다. 이 개념은 CAD나 지리 정보 시스템 등의 데이터를 처리하는데 유용하며 또한 OLAP의 데이터에서도 좋은 성능을 보이는 것으로 나타났다[7].

우선 다차원 파일 구조에 관련된 용어와 특징을 설명한다[8].

표1은 다차원 파일 구조에 대한 용어를 정의한 표이다. 표1의 용어를 바탕으로 그림1을 예제로 각 용어를 설명한다.

그림1은 X,Y,Z 세 개의 구성 속성으로 이루어진 다차원 파일 구조에 대한 집계 연산의 예를 나타낸다. 도메인 공간은 모두 여섯 개의 영역으로 분할되어 있으며, 각 영역에 속하는 레코드들은 A,B,C,D,E,F의 페이지에 저장되어 있다.

표 1. 다차원 파일 구조의 용어들.

파일	속성들의 리스트로 구성된 레코드들의 모임
구성 속성	파일을 구성하는데 참여하는 속성들
다차원 파일 구조	두개 이상의 구성 속성을 가지는 파일 구조.
도메인	한 속성이 취할 수 있는 모든 값들의 집합
도메인 공간	모든 속성에 대한 도메인들의 카티젼 곱
영역	도메인 공간의 일부분.
페이지 영역	도메인 공간에서 페이지 P가 나타내는 영역
집계 윈도우	집계 연산을 위하여 도메인 공간을 한개 이상의 영역으로 분할한 경우.
그룹화 속성	구성 속성 중 집계 연산에서 레코드들을 여러 개의 그룹으로 나누는 기준이 되는 속성들(차원)
집계 속성	분석이 되는 측정값

그림에서 X와 Y는 분석의 관점이 되는 그룹화 속성이며 Z는 집계 속성이다. 도메인공간은 X:[0,99]*Y:[0,99]이며, 이공간의 일부분이 영역이다. 그리고, 집계 연산을 위해 분할된 X:[0,49]*Y:[0,49],X:[0,49]*Y:[50,99],X:[50,99]*Y:[0,49], X:[50,99]*Y:[50,99]의 네 개의 그룹화 영역이 집계 윈도우이다.

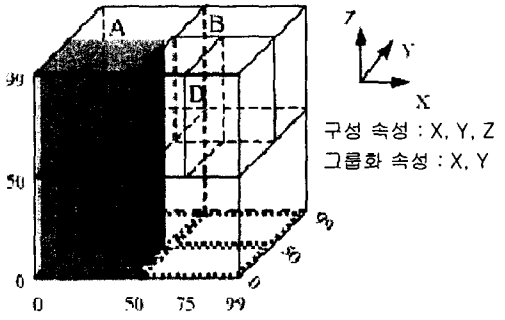

그림 1. 다차원 파일 구조

다차원 파일 구조에서의 가장 간단한 집계 연산은 도메인 공간 전체를 하나의 집계 윈도우로 보고 집계 연산을 처리하는 방법이다. 하지만 이 방법은 주 기억장치 크기의 한계로 인해 실제로 사용하기 어렵다. 주 기억장치의 문제를 해결하기 위해, 도메인 공간을 여러 개의 집계 윈도우로 나누어 집계 연산을 처리하는 방법을 사용한다.

다차원 파일 구조의 장점은 다차원 클러스터링을 지원한다는 것이다. 즉 다중키 액세스를 효율적으로 처리할 수 있다. 다차원 파일구조는 다차원 클러스터링을 위해 도메인 공간을 여러 개의 영역으로 분할하고 각 영역에 속하는 레코드들을 하나의 페이지에 저장한다[7].

이러한 장점을 바탕으로 다음 절에서는 다차원 파일 구조를 사용하여 큐브를 생성하는 방법을 설명한다.

3.2 다차원 파일 구조를 이용한 큐브 생성 방법.

3.1절에서는 간략하게 다차원 파일 구조에 대해 알아보았다. 본 절에서는 다차원 파일 구조를 이용하여 큐브를 생성하는 방법을 설명한다.

그림 1의 예에서 사실 Z에 대한 X, Y 두 차원의 큐브를 산출한다고 하면 XY, X, Y, all의 집계를 산출하여야 한다. 집계는 우선 X 차원의 순서로 다차원 파일 구조를 스캔하여 결과를 산출한다. 그 후 Y 차원의 순서로 다시 다차원 파일 구조를 스캔하여 결과를 산출한다.

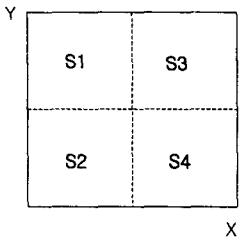

그림 2. 그림1의 집계 윈도우

그림 2에서 X차원의 순서(S_2, S_4, S_3, S_1)로 각각의 집계 윈도우를 이용하여 부분집계를 처리하면 XY, X, all의 값을 구할 수 있다. 나머지 Y의 집계는 Y차원의 순서로 다시 한번 다차원 파일 구조를 읽고 그 결과를 산출한다.

여러 차원으로 구성된 큐브를 산출하는 방법으로 위에서 설명한 방법을 확장하여 다시 간략하게 설명한다. 3차원 이상으로 구성된 큐브를 산출 할 때는 위에 방법과 마찬가지로 각 차원의 순서로 다차원 파일을 스캔하며 큐브를 산출하고 또한 최소비용의 부모에서 자식을 산출하는 방법을 사용한다. 다음 그림은 4차원의 예를 각 차원의 순서로 집계하는 과정을 나타낸 그림이다.

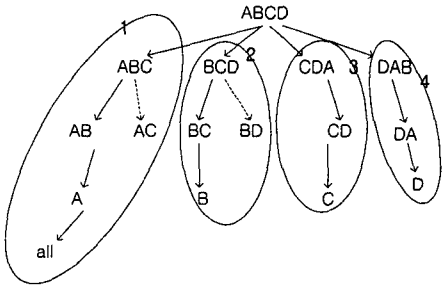

그림 3. 4차원 큐브의 산출 과정.

그림 3은 차원의 순서 A,B,C,D에 의해 큐브를 산출하는 과정을 트리로 나타낸 것이다. 그림에서 번호 1,2,3,4는 각 차원의 순서로 집계를 하는 과정의 순서를 나타내고 실선은 각 차원의 순서로 집계 시에 한번에 구해지는 것들을 나타내며 점선은 네번의 다차원 파일을 스캔 후 구하지 못하고 남은 집계들을 최소 부모의 집계 결과에서 구하는 부분을 뜻한다.

3.3 제안한 방법을 ROLAP에 적용.

3.2 절에서 제안한 방법은 다차원 파일 구조로 되어있는 DBMS에서 최적의 성능을 보인다. 또한 제안한 방법은 ROLAP에도 적용이 가능하며 ROLAP에서도 좋은 성능을 보일 것으로 예상 된다. 본 논문에서 제안한 방법을 ROLAP에 적용하기 위해서는 3단계의 과정을 통해 가능하며 그 과정은 다음과 같다.

표 2. ROLAP에 적용하는 3단계 과정.

1. 테이블을 스캔하여 다차원 파일 구조로 로드한다.
2. 로드된 다차원 파일 구조의 데이터를 제안한 방법으로 집계를 산출한다.
3. 산출된 결과를 다시 테이블로 변환한다.

4. 다차원 파일 구조를 사용한 큐브 산출 알고리즘.

본 절에서는 3.2절에서 제안한 방법의 알고리즘을 표3으로 소개한다.

표 3. 다차원 파일 구조를 이용한 큐브 산출 알고리즘

다차원 파일 큐브 알고리즘.
입력 :
(1) OLAP 데이터를 저장하고 있는 다차원 파일 구조. R-file
(2) 그룹화 속성들의 집합 A(차원)
(3) 집계 속성 M (사실)
출력 :
그룹화 속성을 원소로 하는 집합의 부분집합들의 집계 결과.
알고리즘 :
1. 각 영역들로 이루어진 r-file의 그룹화 속성의 카티젼 곱으로 된 도메인 공간을 집계 윈도우로 분할.
2. 각 집계 윈도우를 차원의 순서로 집계.
2.1 첫 번째 차원의 순서로 되어있는 각각의 집계 윈도우를 집계하기 위하여 영역 질의를 구성.
2.2 주 기억 장치에 현재 차원 순서를 Prefix로 하는 집계 결과와 그 자식들의 집계 결과에 대한 앤트리를 생성
2.2 2.1에서 구성한 영역질의를 통해 r-file을 검색.
2.3 검색된 각 레코드에 대해 A의 값을 키로 하여 결과 테이블의 해당 앤트리에 M의 값을 저장.
3. 모든 차원의 순서로 단계 2를 반복
4. 누락된 집계 테이블을 최소 부모에서 산출.

5. 결론.

본 논문은 OLAP 시스템의 핵심이 되는 기술인 큐브를 효율적으로 산출하기 위하여 다차원 파일 구조를 사용하여 집계 연산의 성능을 향상시켜 큐브의 결과를 산출하는 방법을 제안하였다. 제안한 방법의 구현과 성능 평가가 앞으로 필요하며 제안한 알고리즘의 보완을 통해 보다 좋은 성능을 발휘 할 수 있도록 해야 한다. 현재 제안한 방법의 구현은 다차원 파일 구조의 하나인 계층 그리드 파일(Multilevel Grid File: MLGF)를 이용하여 구현 중에 있으며 기존에 연구 되었던 방법들과의 비교를 통해 성능을 평가할 계획이다.

참고 문헌

[1] 조재희, 박성진 "OLAP 테크놀로지" Sigma Consulting Group, 2000

[2] J. Gray, A.Bosworth, A.Layman, and H.Pirahesh. "Data Cube: A relational aggregation operator generalizing group-bys, cross-tabs and sub-totals" Technical Report MSR-TR-95-22, Microsoft Research, Advance Technology Division, Microsoft Corporation, Redmond, 1995

[3] S. Agawal, R. Agrawal, P. Deshpande, J. Naughton, S. Sarawagi and R. Ramakrishnan. "On the Computation of Multidimensional Aggregates". In VLDB'96, 1996

[4] K.A. Ross and D. Srivastava. "Fast computation of Sparse datacubes" In VLDB'97, 1997

[5] Li, J., Rotem, D., and Srivastava, J., "Aggregation Algorithms for Very Large Compressed Data Warehouses" In VLDB'99, 1999

[6] Y.Zhao, P.Deshpande, and J.F.Naughton "An array-based algorothm for simultaneous multidimensional aggregates" In SIGMOD 1997, 1997

[7] 이영구, 문양세, 황규영 "다차원 온라인 분석처리에서 분리-포함 분할 다차원 파일 구조를 사용한 원-패스 집계 알고리즘" 정보과학회 논문지 : 데이터베이스 제 28권 제2호, 2001

[8] Whang, K. et al., "Dynamic Maintenance of Data Distribution for selectivity Estimation," The VLDB Journal, Vol. 3, No. 1, 1994