데이타 큐브에서 세분화된 뷰 실체화 기법
(Fine Granule View Materialization in Data Cubes)

김 민 정 † 정 연 돈 †† 박 웅 제 ††† 김 명 호 ††††
(Min Jung Kim) (Yon Dohn Chung) (Woong Je Park) (Myoung Ho Kim)

요 약 일반적으로 뷰라고 불리는 데이타 큐브의 일부를 실체화하여 저장하는 방법은 데이타 웨어하우스에서 많이 사용되는 기술이다. 뷰는 집계 함수로 정의되는 질의의 결과이다. 본 논문에서는 세분화된 뷰의 개념을 소개한다. 세분화된 뷰란 각 차원별로 정해진 구간에서의 집계 함수 결과이다. 이때 각 차원별로 나누는 구간은 질의의 접근 형태를 기준으로 설정된다. 세분화된 뷰의 표현 및 선택을 위하여 AND-OR 큐브 그래프와 AND-OR 최소 비용 그래프를 정의한다. 그리고, 이 구조체들을 이용하여 세분화된 뷰 실체화 기법을 제안한다. 실험을 통해 제안하는 방법의 성능을 평가한다.

Abstract Precomputation and materialization of parts, commonly called views, of a data cube is a common technique in data warehouses. The view is defined as the result of a query which is defined through aggregate functions. In this paper, we introduce the concept of fine granule view. The fine granule view is the result of a query defined through aggregate functions and the range on each dimension, where the subdivision of each dimension is based on queries' access patterns. For the representation and selection of fine granule views to materialize, we define the AND-OR cube graph and AND-OR minimum cost graph. With these structures, we propose a fine granule view materialization method. And, through experiments, we evaluate the performance of the proposed method.

1. 서 론

의사결정 지원시스템은 기업내의 비즈니스 구성원의 의사결정을 지원하기 위한 정보시스템으로, 사용자의 요구에 신속하게 대응하여 필요한 데이타를 추출하고 분석하는 기능을 제공한다. OLAP(On-Line Analytical Processing)은 실시간으로 데이타 웨어하우스 또는 직접 데이타베이스에 접근하여 다차원적인 정보분석을 하는 과정을 말한다. 일반적으로 OLAP 응용 시스템은 의사결정 지원시스템의 한 구성요소가 되며, 다차원적인 관점에서 사용자가 쉽고 선택적으로 데이타를 추출하고 검색할 수 있도록 한다.

OLAP 환경에서는 다양한 관점에서 데이타를 분석하고 검토하기 위해 다차원적인 데이타 모델인 데이타 큐브(data cube)를 사용한다. 데이타의 분석 관점을 나타내는 차원은 데이타 큐브의 각 축으로 구성되고, 데이타 큐브의 셀은 각 차원 항목의 조합에 의해 이루어진다. OLAP 질의는 일반적으로 특정 레코드의 값이 아닌 전체적인 경향을 분석하기 위한 것으로 많은 수의 집계(aggregation) 연산을 포함하는 특징을 갖는다. 또한, OLAP 시스템의 사용자는 의사 결정 시 다양하고 동적인 요청에 대한 신속한 질의 응답을 요구한다. 그러나, OLAP 질의는 방대한 양의 데이타에 대해 복잡한 집계 연산을 빈번히 수행해야 하므로 오랜 수행 시간을 필요로 한다. 이러한 문제점을 해결하기 위한 방법으로 비트맵(bit-map) 색인이나 조인(join) 색인 등과 같은 색인 기법을 활용하는 방법과 자주 요청되는 질의의 결과를 미리 계산하여 저장하는 실체화(materialization) 기법 등이 제안되었다[2, 3, 5, 7, 10]. 실체화 기법은 질의

· 본 연구는 한국과학재단 특정기초연구(과제번호 199901-303-007-3) 지원으로 수행되었음.
† 비 회 원 : LG전자기술원 연구원
lafwing@lg-elite.com
†† 비 회 원 : 한국과학기술원 전자전산학과 연구 교수
ydchung@dbserver.kaist.ac.kr
††† 비 회 원 : (주)다우데이타시스템연구소 연구원
ujpark@dbserver.kaist.ac.kr
†††† 종신회원 : 한국과학기술원 전자전산학과 교수
mhkim@dbserver.kaist.ac.kr
논문접수 : 2000년 9월 1일
심사완료 : 2001년 2월 15일

수행에 필요한 데이타 큐브의 일부를 미리 계산하여 기록해 둠으로써 실제 질의 처리 시에 저장된 결과를 이용하여 질의 응답 시간을 감소시키는 방법이다.

일반적으로 전체 데이타 큐브를 저장하는 경우 최적의 질의 수행 속도를 얻을 수 있지만, 많은 저장 공간이 요구되므로, 대용량 데이타베이스에 적용하기에는 부적절하다. 따라서, 사용 가능한 저장 공간의 크기를 고려하여 저장할 데이타 큐브의 부분을 효과적으로 선택하여 실체화하는 방법이 사용된다.

기존의 실체화 기법에 관한 연구는 집계 연산 애트리뷰트에 의해 분할되는 단위를 기반으로 이루어졌다[2, 7, 10]. 그러나, 집계 연산 애트리뷰트의 조합으로 이루어진 단위의 전체보다, 그 중에서 일부분에 대해 정보를 추출하고자 하는 질의에 있어, 기존 방법에서 가정하는 실체화 단위는 사용자의 관심영역을 반영하기에 부적절하다. 따라서, 사용자의 질의 패턴을 반영할 수 있는 세분화된 실체화 단위가 필요하다.

본 논문에서는 주어진 저장 공간의 효율적인 활용을 보장하는 세분화된 데이타 큐브 실체화 기법을 다루도록 한다. 데이타 큐브의 세분화를 위해 본 논문에서는 차원의 계층 구조를 이용하며, 세분화된 데이타 큐브를 표현하기 위해 AND-OR 큐브 그래프와 AND-OR 최소 비용 그래프를 정의한다. 그리고, 이를 이용한 AND-OR 큐브 그래프상의 부분 선택 방법을 제시한다. 또한, 실험을 통해 기존 방법과 제안하는 세분화된 실체화 기법의 성능을 비교하고 다양한 질의 유형에서의 적응 정도를 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 알아보고, 3장에서는 기존 연구의 문제점을 살펴보고 새로운 접근 방향에 대해서 논한다. 4장에서는 세분화된 데이타 큐브에서의 실체화 방법에 대해서 기술하고, 5장에서 실험 및 결과를 제시한다. 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

2.1 데이타 큐브와 격자 구조

데이타 큐브는 OLAP 시스템에서 베이스 데이타에 대해 다차원적인 관점을 제공하기 위해 사용되는 개념이다. 일반적으로 데이타 큐브는 집계 연산 시 사용되는 애트리뷰트, 즉, 각 차원을 이루는 애트리뷰트 조합에 대한 연산결과로 구성된다. 그림 1은 매장(s), 품목(p), 고객(c) 의 3차원 데이타 큐브의 예이다.

데이타 큐브에서 각 차원 애트리뷰트의 조합으로 나뉘어진 부분은 특정 부분을 이용하여 값을 계산할 수

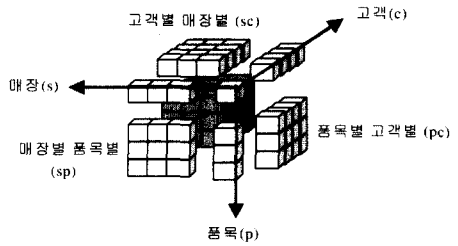

그림 1 3차원 데이타 큐브

있는 의존 관계(dependency relation : ≤)가 존재한다[1]. 위의 예에서 보면 매장(s) 차원에 대해서 group-by되어진 질의는 (s)에 대해서 집계 연산된 결과 뿐만 아니라 (s, p), (s, c) 또는 (s, p, c) 에 대해서 집계 연산된 결과에 의해서도 계산될 수 있다. (이를 (s) ≤ (s, p), (s) ≤ (s, c) 그리고, (s) ≤ (s, p, c)로 나타낸다.) 의존 관계는 특정 부분이 실체화됨으로써 실체화된 영역에 대한 질의 처리 뿐만 아니라, 그 영역과 의존 관계에 있는 다른 영역에 대한 질의 처리에도 영향을 주기 때문에, 실체화할 부분을 선택할 때 고려해야 할 중요한 요소가 된다. 차원 애트리뷰트에 의해 나누어진 데이타 큐브의 의존 관계는 격자(lattice) 구조에 의해 표현될 수 있다[1, 7].

그림 1의 데이타 큐브 예제의 차원 계층 구조와 격자 구조는 그림 2와 같이 나타낼 수 있다. 격자에서 각 노드들의 연결선은 의존 관계를 나타내며, 격자의 구성은 애

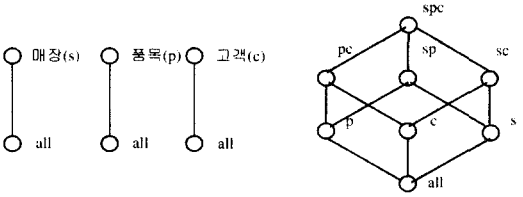

그림 2 차원의 계층 구조와 큐브 격자

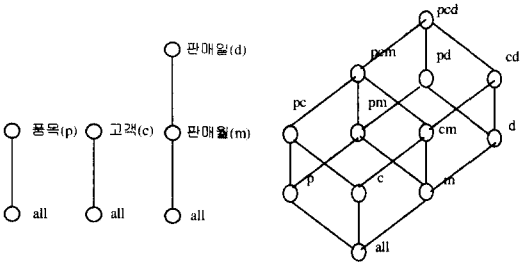

그림 3 의존 격자

트리뷰트의 관계에 따라 달라질 수 있다. 그림 2와 같은 일반적인 격자 구조를 하이퍼 큐브 격자(hypercube lattice) 라고 하고, 애트리뷰트의 계층구조가 다단계로 존재하는 경우의 격자 구조를 의존 격자(dependency lattice) 라고 한다. 그림 3은 의존 격자의 예를 보여준다.

2.2 뷰 (View) 선택 방법

집계 연산 애트리뷰트에 의해 구성된 큐브 격자에서 저장 공간의 제한을 넘지 않도록 실체화할 적절한 노드의 집합을 선택하는 문제를 뷰 선택 문제(view selection problem)라 한다[2]. 뷰 선택 문제는 NP문제로 최적의 해를 구하는 것은 시간이 매우 많이 걸린다[2]. 기존 연구는 격자의 각 노드를 뷰로 가정하고, 뷰 선택 문제에 대한 최적 근접의 해를 구하는 휴리스틱한 알고리즘들을 제시하였다.

먼저 [1]에서 제시된 방법은 ROLAP (Relational OLAP)환경을 가정하고, 비용 모델로 (격자의 각 노드에 대한 질의 비용이 해당 뷰의 튜플의 수에 비례한다는 전제하에) 튜플의 수를 사용한다. 뷰를 선택해 가는 과정은 현 상태에서 각 뷰를 실체화했을 때 얻을 수 있는 이익을 구하여, 그 중 단위 저장 공간 당 이익을 최대화하는 뷰를 선택하는 방법이다. 주어진 저장 공간을 넘지 않는 한도 내에서 위의 과정을 반복하여 원하는 만큼의 뷰를 선택한다.

[8]에서 언급되어진 알고리즘은 실체화 되어진 뷰에 대해서 색인을 구성함으로써 성능향상을 도모할 수 있다는 가정 하에, 색인을 실체화할 객체로 고려하여 선택하는 방법이다. 즉, 각 단계에서의 고려대상 집합은 아직 선택되지 않은 뷰와 이미 선택되어진 뷰의 색인들을 모두 포함하는 객체 집합으로 질의-뷰 그래프를 이용하여 그 중 이익을 최대화하는 객체를 선택한다.

3. 동 기

데이타 큐브의 특정 영역을 미리 계산하여 저장한 후 실제 질의 처리 과정에서 베이스 데이타 대신 저장된 결과를 이용함으로써 읽어야 하는 데이타의 수를 감소시켜 질의 처리를 효율적으로 수행하는 방법이 실체화 기법이다. 이때 어떤 부분을 선택하여 미리 계산하여 저장할 것인가 하는 문제는 전체적인 성능 향상 정도에 영향을 미치는 중요한 요인이 된다.

2장에서 살펴본 것처럼 기존 연구에서는 실체화할 부분을 선택하는 문제를 뷰 선택 문제로 보았다. 즉, 미리 계산하여 저장할 부분을 선택하는 기본 단위로 큐브 격자의 각 노드를 기준으로 삼는다. 그러나, 사용자의 관심이 하나의 노드 전반에 걸쳐 균등하지 않은 응용에 있어서는 이러한 뷰 (즉, 큐브 격자에서의 한 노드) 단위의 실체화 기법이 효율적이지 못할 수 있다.

가령, 서울에 있는 사람들은 다른 지역에 비해서 서울과 관련된 정보에 대해서 많은 관심을 갖는다. 즉, 적용되는 OLAP 시스템의 목적과 특성에 따라 요청되는 질의 패턴이 모든 영역에 대해서 일정한 것이 아니라 특정 영역에 편중될 수 있다는 점이다. 따라서, 노드 단위로 선택하여 저장하는 것보다 질의 빈도수가 현저하게 높은 특정 영역을 선별하여 그 부분을 실체화한다면, 동일한 저장 공간을 활용하여 노드 단위 선택보다 성능을 향상시킬 수 있을 것이다. 예를 들어, 그림 4로 표현되는 음반 판매에 대한 데이타 큐브를 가정하자. p노드와 c노드의 질의 빈도수가 특정 영역에 치중해서 분포한다고 하자. 이 때 p노드의 일부분을 선택하여 저장하는 것이 가능하다고 하면, p노드 전체를 선택할 때에 비하여 적은 저장 공간을 사용하면서 비슷한 성능향상을 얻을 수 있다. 또한 남은 공간을 이용하여 추가적으로 다른 부분을 선택할 수 있게 된다. 즉, 가요에 해당하는 p노드의 부분과 10대에 해당하는 c노드의 부분을 선택하면 저장 공간의 측면과 성능 향상 측면에서 모두 효과적일 수 있다.

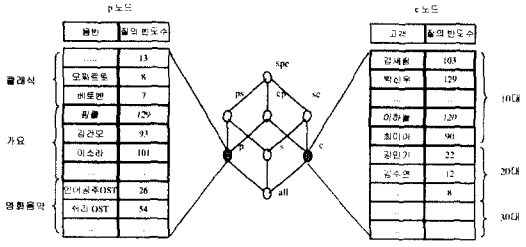

그림 4 불균등한 빈도수를 나타내는 예제

4. 세분화된 데이타 큐브에서의 부분 선택

본 논문에서는 데이타 큐브가 질의의 접근 특성에 따라 세분화된 단위로 분할(partition)되어 구성된다고 가정하고, 분할된 데이타 큐브에 대해서 각 단위간의 의존 관계를 고려하여 실체화할 부분을 선택하는 방법을 제안한다. 이를 위해, 세부 단위로 분할된 데이타 큐브를 표현하는 AND-OR 큐브 그래프와 특정 부분이 선택되어졌을 때 단위간의 관계를 나타내는 AND-OR 최소 비용 그래프를 정의하고, AND-OR 큐브 그래프상에서의 부분 선택 방법을 제안한다.

본 논문에서는 그림 5와 같이 각 차원이 세부 단위로 나누어진 다음, 이들의 조합으로 전체 데이타 큐브가 구

성된다고 가정한다. 차원의 분할은 사용자 질의를 분석하여 결정할 수 있다. (이 부분에 대한 자세한 설명은 [9]를 참조하라.) 그림 5는 2단계 계층 구조에 기반한 세분화된 데이타 큐브의 예를 보여준다. 각 세분화된 단위를 구별하기 위해 고유 식별자를 부여하여 나타내었다. 즉, {D0, D1}과 {P0, P1}에 의해 나타내어진 단위 영역은 u0의 식별자로 나타내었다.

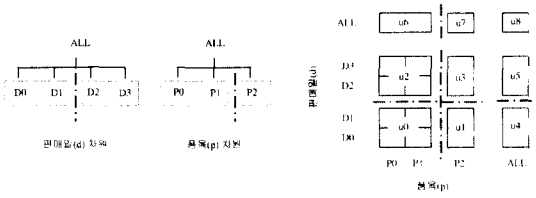

그림 5 차원 분할과 단위 구성

4.1 AND-OR 큐브 그래프

기존 연구[10]에서 애트리뷰트를 기준으로 분할된 데이타 큐브는 격자 구조로 표현되었다. 그러나, 본 논문에서는 데이타 큐브를 세분화된 단위로 분할하므로, 분할된 데이타 큐브를 표현하는 새로운 방법이 필요하다. 이 장에서는 데이타 큐브의 분할된 각 단위 영역들의 관계를 표현하기 위해 AND-OR 큐브 그래프를 정의하여 사용한다. [10]에서 제안한 AND-OR 뷰 그래프와 의미적으로 동일한 개념이다. 하지만, 본 논문에서는 애트리뷰트의 조합으로 정의되는 기존의 뷰 단위가 아닌, 본 논문에서 제안하고 있는 세분화된 뷰를 단위로 한다는 점에서 차이를 갖는다.

AND-OR 큐브 그래프는 일종의 방향성 그래프(digraph)이다. 세분화된 데이타 큐브의 단위 영역은 그래프의 정점(vertex)으로 표현된다. 각 정점은 의존 관계에 따라 방향선(directed edge)으로 연결된다. 방향선은 나오는 정점에 해당하는 단위 영역으로부터 들어가는 정점에 해당하는 단위 영역을 계산할 수 있는 의존 관계를 표현한다. 모든 방향선은 다음의 AND 혹은 OR의 관계를 갖는다.

정의 1: 단위 영역의 집합 $\{u_i \mid u_i$ 는 세분화된 데이타 큐브의 단위 영역$\}$으로부터 단위 영역 u_j 전체가 구해질 수 있는 의존 관계가 성립하면, u_i와 u_j는 **AND 관계**에 있다고 정의한다. □

AND-OR 큐브 그래프에서 AND 관계는 각 방향선을 가로 질러 'AND'를 표시한다. 예를 들어, { u1, u2, u3 } ≤ u4 이면 AND-OR 큐브 그래프에서는 각 단위 영역 u1, u2, u3, u4 를 각 정점 v1, v2, v3, v4 로 나타내고 u1, u2, u3와 u4의 의존 관계는 v1, v2, v3에서 v4로 연결되는 방향선의 AND 관계로 표현된다. 그림 6은 위의 예를 나타내고 있다. 본 논문에서는 표현을 단순화시키기 위해 방향선의 방향은 위에서 아래로 향한다고 가정한다. 또한, 하나의 단위 영역에서 하나의 단위 영역으로 연결되는 방향선은 AND 관계 표시를 생략할 수 있다고 한다. 하나의 단위 영역에 들어오는 AND 관계는 여러 경우가 생길 수 있다. 이러한 여러 AND 관계의 방향선들은 OR 관계를 나타낸다.

정의 2: 같은 단위 영역에 연결되어지는 여러 AND 관계들의 단위 영역 또는 단위 영역들의 집합에 대한 집합 $OU_j = \{ U_i \mid U_i$는 u_i와 AND 관계에 있는 단위 영역 집합$\}$와 u_j는 **OR 관계**에 있다고 정의한다. u_j는 OU_j의 임의의 원소 U_i에 의해 구해질 수 있다. □

AND-OR 큐브 그래프에서 하나의 정점에 들어가는 AND 관계의 여러 방향선은 OR 관계를 나타낸다고 할 수 있으므로 OR 관계는 특별히 기술하여 나타내지 않는다. 예를 들어, {u1, u2} ≤ u6, {u3} ≤ u6, {u4, u5} ≤ u6관계가 성립하면, {{u1, u2}, {u3}, {u4, u5}}는 u6와 OR관계에 있다. AND-OR 큐브 그래프는 그림 6과 같이 나타난다. 그림 7은 그림 5의 예제 데이타 큐브를 AND-OR 큐브 그래프로 나타낸 것이다. (편의상 AND 표시를 간략히 하였다.)

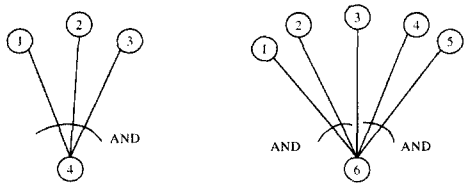

그림 6 AND 관계와 OR 관계

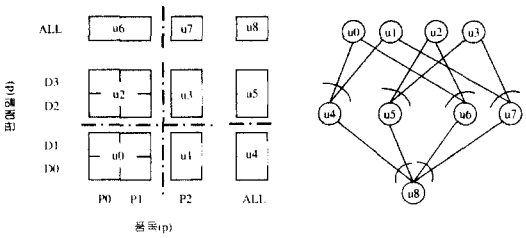

그림 7 AND-OR 큐브 그래프

분할된 데이타 큐브 C에 대한 AND-OR 큐브 그래프 CG(Cube Graph)=(V, E, A)는 방향성 그래프로, V는 C의 각 단위 영역들의 집합으로, E는 방향선이 들어가

는 정점과 나오는 정점의 레벨 차이가 1인 모든 AND 관계 방향선의 집합이다. A는 E에 포함되는 모든 방향선의 AND 관계에 대해서, 하나의 AND 관계를 구성하는 방향선의 집합을 원소로 하는 집합으로 정의한다. 각 정점은 해당하는 단위 영역에 대한 질의 빈도수와 저장 장소의 크기를 변수로 갖는다.

4.2 AND-OR 최소 비용 그래프

AND-OR 큐브 그래프를 사용하여 분할되어진 데이타 큐브를 기술한 후, 실체화 할 부분을 구하기 위하여 AND-OR 최소 비용 그래프를 이용한다. AND-OR 최소 비용 그래프는 AND-OR 큐브 그래프와 유사한 구조의 방향성 그래프이다. 그러나, 모든 정점간의 관계를 표시하는 것이 아니라, 선택되어진 정점을 기준으로 정점간의 필요한 관계만을 표시하기 위한 그래프이다. AND-OR 최소 비용 그래프의 모든 방향선은 선택과 보류 중 한가지로 나타내며, 각 정점에 대해 들어가는 하나의 AND관계 방향선은 반드시 선택된다. 이때 선택으로 표시되어진 AND 관계에서의 들어가는 정점과 매핑되는 질의의 최소 질의 비용을 나타낸다.

AND-OR 큐브 그래프 CG = (Vc, Ec, Ac)에 대한 AND-OR 최소 비용 그래프 MCG (Minimum Cost Graph) =(V, E, A)는 방향성 그래프로 V는 Vc의 모든 정점으로 구성되고, E는 Ec의 방향선 중에서 선택되어진 정점으로 들어가는 방향선을 제외한 것이다. Ac는 Ec의 방향선 가운데 하나의 AND 관계를 구성하는 방향선의 집합을 원소로 하는 집합이다. 각 정점은 해당하는 단위 영역에 대한 질의 빈도수와 저장 장소의 크기, 그리고 단위 영역에 대한 질의가 발생할 경우 필요한 질의 처리 비용을 변수로 갖는다.

AND-OR 최소 비용 그래프는 알고리즘 1에 의해 AND-OR 큐브 그래프로부터 구해질 수 있다. 이 과정을 통해 구해진 AND-OR 최소 비용 그래프는, 선택되어진 특정 정점을 이용한 각 정점에 대한 최소 질의 비용과 최소 비용 경로를 나타낸다. 그리고, 현재 상태에서 더 이상 고려될 필요가 없는 방향선을 제거하여 추가되어지는 정점의 선택 시에 고려되어야 할 검색 대상을 줄이고, 고려할 필요가 있는 방향선에 대해서는 선택과 보류의 의미를 부여함으로써 질의 계획(query plan)을 동시에 표현하고 있다.

알고리즘 1 AND-OR 최소 비용 그래프의 유도

```
1 G <- AND-OR 큐브 그래프
2 G의 상위 레벨로부터 하위 레벨까지 각 정점 v에 대해서 반복한다.
  2.1 정점 v가 선택된 정점이면, v에 들어오는 모든 방향선을 제거한다.
  2.2 정점 v가 선택된 정점이 아니면, 다음을 수행한다.
    2.2.1 정점 v에 들어오는 모든 AND 관계들에 대하여 질의 비용의 합을 구한다. 그리고, 이 값들 중에서 최소 값을 갖는 AND 관계를 선택한다.
    2.2.2 정점 v의 질의 비용 값을 위에서 구한 최소 값으로 설정한다.
  2.3 선택된 AND 관계로 연결된 방향선은 선택으로 표시하고, 그 외의 다른 방향선은 보류로 표시한다.
3 G는 AND-OR 큐브 그래프의 최소 비용 그래프이다.
```

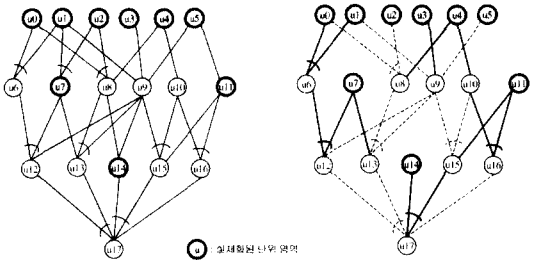

그림 8 AND-OR 최소 비용 그래프

그림 8은 일반적인 AND-OR 큐브 그래프가 주어졌을 때 AND-OR 최소 비용 그래프를 유도한 예이다. AND-OR 최소 비용 그래프에서 각 단위 영역에 대한 질의 계획은 해당하는 정점에서 선택 방향선을 따라 반대로 추적해서 얻을 수 있다. 즉, 위의 예제 그림에서 u17은 u14와 u15에 의해 구해질 수 있으며, u15는 u11에 의해 구해질 수 있다. 따라서, u17은 u14와 u11에 의해 최소 비용으로 구해질 수 있다. 그림 9는 u17의 질의 계획 그래프를 나타낸다.

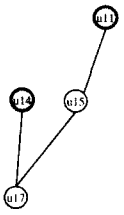

그림 9 u17에 대한 질의 계획

선택된 데이타 큐브의 부분은 실제 질의 처리 시에 참조되어야 한다. 일반적으로 요청된 SQL 질의는 질의 분석을 통해 동일한 결과를 얻을 수 있는 질의로 나누어져 재구성된다[3, 5]. 단위 영역 u17에 대해 요청되어진 질의는 질의 처리 과정에서 미리 계산되어 저장된 부분을 사용하기 위해 재구성되어지는데, 이 때 AND-

OR 최소 비용 그래프를 통해 구해진 질의 계획을 참조할 수 있다. 예를 들어, 정점 u17에 대한 질의가 다음과 같다고 하자.

```
SELECT  d1, d2, d3 ,  sum(a)
FROM  R
GROUP BY  d1, d2, d3
```

요청되어진 질의는 질의 계획의 끝 정점에 해당하는 단위 영역을 이용하도록 질의 처리 과정에서 다음과 같이 재구성될 수 있다. (Vn은 un의 각 단위 영역에 대한 뷰를 나타낸다고 가정한다.) 여러 단위 영역을 이용하여 계산되는 경우는 여러 개의 질의로 분할되어 UNION ALL 연산자를 통해 나온 여러 개의 결과 집합을 합친다[5].

```
SELECT  d1, d2, d3, sum(a)
FROM  V11
GROUP BY  d1, d2, d3
UNION ALL
SELECT d1, d2, d3, sum(a)
FROM  V14
GROUP BY  d1, d2, d3
```

4.3 부분 선택 기법

본 논문에서 고려하는 데이타 큐브에 대한 각 질의 비용은 기존 연구와 마찬가지로 질의를 처리하기 위해서 읽어야 하는 데이타의 개수로 해당하는 단위 영역의 저장 공간 크기로 간주한다[1, 7].

정의 3: 각 질의 w에 대한 응답 비용을 c(w)라고 하면, 전체 질의에 대한 질의 비용(C)은 각 질의의 요청 빈도수인 f(w)를 곱한 값의 합으로 구할 수 있으며, 평균 질의 비용 average C는 이 값에 빈도수의 합을 나눈 값이다.

$$C = \sum(c(w) \cdot f(w))$$

$$average\ C = \frac{\sum(c(w) \cdot f(w))}{\sum f(w)}$$

□

AND-OR 큐브 그래프에서 실체화할 정점들을 찾는 방법은 greedy 알고리즘을 이용한다. 매 단계 사용하는 단위 저장 공간을 고려하여, 최대로 질의 비용을 감소시키는 정점을 선택하는 방식이다. 이 때, 특정 부분이 선택되었을 때 전체 질의 비용을 나타내는 AND-OR 최소 비용 그래프를 이용하여 정점을 선택할 수 있다.

AND-OR 큐브 그래프에서 하나의 정점 v를 선택함으로 감소되는 비용은 v를 포함하는 모든 정점 w에서 감소되는 질의 비용의 합에 의해 계산된다. S를 실체화된 정점의 집합이라고 나타내고, 하나의 정점 v를 계산할 수 있는 S내의 최소 비용 정점들의 집합을 L(v)로 정의한다. 즉, $L(v) \le v$ 가 성립한다.

정의 4: S의 관점에서 하나의 정점 v에 대한 최소 질의 비용 C(v)는 다음과 같이 정의된다. 여기서, sizeof(v)는 v의 데이타의 수를 나타낸다.

(1) v가 S에 포함되면, C(v)=sizeof(v)

(2) v가 S에 포함되지 않으면, v의 최소 비용은 S내의 최소 비용을 갖는 정점들의 집합 L(v)의 비용의 합이다. 즉, $C(v) = \sum_{w \in L(v)} sizeof(w)$

정의 5: S의 관점에서 정점 u의 감소되는 질의 비용 RC(u,S)는 다음과 같이 정의된다. L(u) 에 속하는 각 정점 v에 대한 질의 감소 비용 RCv를 이용한다.

(1) w를 $v \le w$의 조건을 만족하는 S내의 최소 비용인 정점이라고 하자. w=L(v).

(2) 만약 C(u) < C(w) 라고 하면 RCv = C(w) - C(u), 그렇지 않으면, RCv = 0.

따라서, $RC(u,S) = \sum_{v \le u} RCv$

정의 6: 정점 u의 단위 저장 공간 당 감소되는 질의 비용 $RC_s(u,S)$은 다음과 같다.

$v \le u$, c(L(v)) > c(u), |u| = sizeof(u) 일 때,

$$RCs(u,S) = \frac{RC(u,S)}{|u|} = \frac{1}{|u|}\sum(C(L(v)) - C(u)) \cdot f(v))$$

□

알고리즘 2는 AND-OR 큐브 그래프에서 정의한 단위 저장 공간 당 감소되는 질의 비용을 이용하여 정점의 집합을 선택하는 방법이다. 이때 매 단계 정점을 선택하고 AND-OR 최소 비용 그래프를 조정하기 위해 알고리즘 3을 이용한다. 알고리즘 2의 복잡도는, 매 단계 단위 저장 공간 당 감소되는 질의 비용을 계산하기위해각 정점의 서

알고리즘 2 AND-OR 큐브 그래프에서 부분 선택 방법

```
CG : AND-OR 큐브 그래프
1  M = ø
2  알고리즘 1을 이용하여 CG로부터 AND-OR 최소 비용
   그래프 MCG를 구한다.
3  MCG에서 상위 레벨부터 마지막 레벨까지 모든 정점 v
   에 대해 다음 과정을 반복한다.
   3.1  v가 M에 포함되면, 3으로 돌아간다
   3.2  v가 M에 포함되지 않으면, 서브 그래프의 각 정
        점 w에 대해서 단위 저장 공간 당 감소되는 질의
        비용의 합을 구한다.
4  MCG에서 S보다 크기가 작은 저장 공간 당 감소되는
   질의 비용을 최대로 하는 정점 v가 있는 경우 다음을
   수행한다.
   4.1  S = S - sizeof(v)
   4.2  M = M ∪ v
   4.3  MCG를 선택되어진 정점에 대해 알고리즘 3을 이
        용하여 조정한다.
   4.4  3으로 돌아간다.
5  MCG에서 S보다 크기가 작고, 저장 공간 당 감소되는
   질의 비용을 최대로 하는 정점 v가 없는 경우 끝낸다.
6  집합 M이 선택되어진 정점의 집합이다.
```

브 그래프에 포함되는 모든 정점에 대해서 살펴보아야 하므로, $O(k \cdot n \cdot d^{level})$가 된다. (여기서, k는 선택되어지는 정점의 수이고, n은 전체 정점의 수, d는 데이타 큐브의 차원의 수, level은 데이타 큐브의 레벨의 수를 나타낸다.) 이때 AND-OR 최소 비용 그래프를 이용함으로써 AND-OR 큐브 그래프에서 선택되어진 정점과 관련된 방향선을 제거하여 선택된 정점에 대한 검색 공간이 줄어든다. 실제로 선택된 점과 연결된 상위 레벨의 점들에 대해서 선택된 점의 서브 그래프에 포함되는 정점의 수 만큼을 감소시키는 효과가 나타난다.

알고리즘 3 AND-OR 최소 비용 그래프의 조정

```
MCG : AND-OR 최소 비용 그래프
sv : 선택되어진 정점

1  sv를 초기 노드로 하여 리스트 Q를 구성한다.
2  Q의 첫번째 원소를 pv로 하고, pv를 Q에서 삭제한다.
3  pv의 하위 정점 v에 대해 다음을 수행한다.
   3.1  pv에서 정점 v에 들어오는 방향선이 선택인 경우
        3.1.1  pv에서 v로의 AND 관계 값으로 정점의 최소
               질의 비용을 수정한다.
   3.2  pv에서 정점 v에 들어오는 방향선이 보류인 경우
        3.2.1  pv에서 v로의 AND 관계 값을 구한다.
        3.2.2  3.2.1에서 구한 값이 정점 v의 최소 질의 비용보다
               작은 경우
               3.2.2.1  pv에서 v로의 AND 관계의 방향선을 선
                        택으로 표시하고, 그 외의 다른 방향 선
                        은 보류로 표시한다.
               3.2.2.2  Q 에 v를 추가한다.
4  Q가 비어 있으면 끝내고, 그렇지 않으면 2로 돌아간다.
MCG는 선택된 정점을 반영한 AND-OR 최소 비용 그래프이다.
```

5. 실험 및 분석

이 장에서는 본 논문에서 제안한 세분화된 단위의 실체화 기법에 대한 성능 평가 실험과 분석을 제시한다.

5.1 실험 환경

본 실험에서 사용한 데이타 집합은 3차원 데이타베이스로, 차원 계층 구조는 3개의 레벨로 구성되어 있으며, 모든 차원은 2개의 단위 영역으로 분할되어 있다. 데이타 큐브내의 데이타는 균등하게 분포되어 있다고 가정하고 데이타 개수의 구성은 [6, 4]에서 사용한 크기 추정 방법을 이용한다. 질의 패턴에 대한 변수는 다음과 같다.

* 밀집 영역에 대한 질의 요청 정도 : 사용자의 질의들이 데이타 큐브의 부분에 대해서 집중되는 정도를 나타낸다. 데이타 큐브의 20% 영역에 집중되는 질의와 전체 질의의 비율에 따라 3가지 집합을 구성한다. (나머지 질의는 데이타 큐브 전체에 대해 균등한 분포를 나타낸다.)

-Q0 : 모든 질의가 전체 데이타 큐브에 대해서 균등한 분포를 나타낸다.

-Q60 : 20% 데이타 큐브를 60% 질의가 참조한다.

-Q80 : 20% 데이타 큐브를 80% 질의가 참조한다.

* 실제 질의 패턴 : 실체화시 예상한 질의와 실제 요청되는 질의 패턴의 구성에 따라 4가지 집합을 가정한다.

-P0 : 실제 질의는 예상 질의 패턴과 무관하다.

-P50 : 실제 질의는 50% 예상 질의와 50% 임의의 질의로 구성된다.

-P80 : 실제 질의는 80% 예상 질의와 20% 임의의 질의로 구성된다.

-P100 : 실제 질의는 100% 예상 질의로 구성된다. 즉, 예상 질의 패턴과 동일하다.

질의 패턴에 대한 변수는 Q80, P80을 기본 값으로 가정한다. 본 실험에서 사용한 성능 평가 기준은 다음과 같다.

* 전체 질의 비용

요청되는 질의 q에 대한 질의 비용을 c(q)라고 하면, 전체 질의 비용은 모든 질의에 대한 질의 비용의 합으로 표현된다.

$$C = \sum c(q)$$

* 평균 질의 비용

요청된 질의 집합을 Q로 표기하면, 전체 질의 비용을 요청된 질의 개수로 나눈 값으로 나타낸다.

$$average\ C = \frac{\sum c(q)}{|Q|}$$

* 성능 향상 비율

기존에 제안된 애트리뷰트 기반 분할 방법에 의한 평균 질의 비용을 Ca, 제안하는 방법에 의한 평균 질의 비용을 Cp로 나타내면, 성능 향상 비율은 다음과 같다.

$$Performance\ Improvement(PI) = \frac{Ca - Cp}{Ca}$$

평균 성능 향상 비율은 주어진 저장 공간 s에 따른 평균적인 성능 향상 비율에 대한 평균값으로 나타낸다.

$$average\ PI = \frac{1}{|S|} \int \frac{Ca - Cp}{Ca}\, ds$$

5.2 실험 결과 및 분석

본 논문에서 수행되어진 실험은 다음과 같다.

* 실험 1. 일정한 저장 공간이 주어졌을 때 기존 방법과 제안하는 방법의 전체 질의 비용을 비교한다.

* 실험 2. 예상 질의 패턴 및 밀집 영역에 질의들이 집중하는 정도에 따른 성능 향상률의 변화에 대해서 살펴본다.

* 실험 3. 실제 질의 패턴의 구성에 따라 평균 질의 비용의 변화를 살펴본다.

* 실험 4. 실제 요청되는 질의 패턴의 구성에 따라 기존 방법과 비교한 성능 향상정도를 살펴본다.

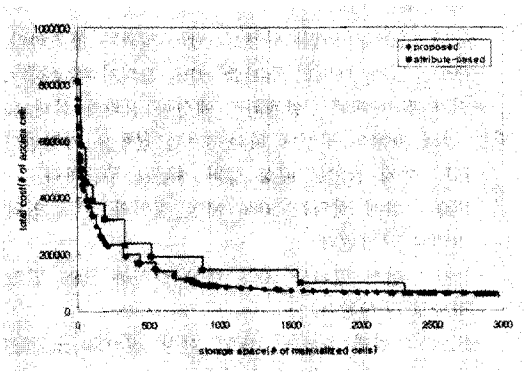

그림 10 [실험 1] 기존 방법과의 전체 질의 비용의 비교

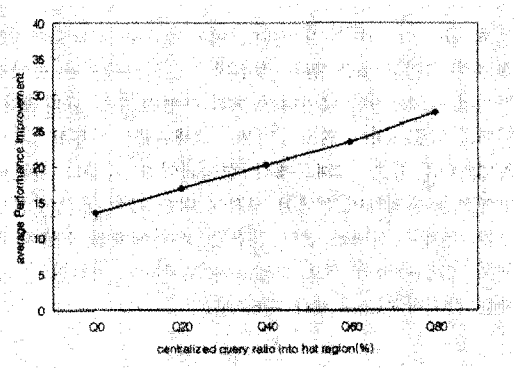

그림 11 [실험 2] 밀집 영역에 대한 질의 집중도에 따른 비교

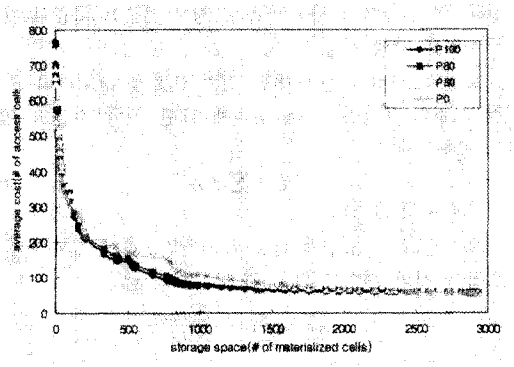

그림 12 [실험 3] 질의 요청 패턴에 따른 평균 질의 비용의 변화

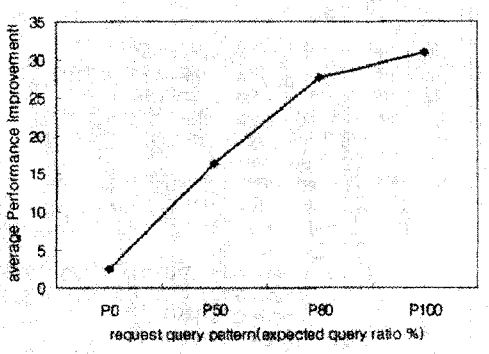

그림 13 [실험 4] 질의 요청 패턴에 따른 성능 향상률의 변화

그림 10은 기존 방법과 제안하는 방법에 의해 선택되어진 데이타 큐브의 부분을 실체화하였을 때 사용하는 저장 공간에 따라 전체 질의 비용 변화를 나타낸 것이다. 전체 질의 비용 변화는 하나의 뷰 또는 단위가 선택되어짐에 따라 단계적으로 감소하는 그래프를 나타낸다. 대부분의 경우에 제안하는 기법이 기존 방법보다 질의 비용이 작게 나타나며, 평균적으로 20~30%의 질의 비용 감소를 보인다. 질의 비용 감소의 이유는 크게 두 가지로 분석될 수 있다. 먼저, 제안하는 방법은 기존 방법보다 더 세분화된 단위로 선택하는 것으로, 단위의 크기를 줄임으로써 그래프의 단계 폭을 감소시켜 주어진 저장 공간을 효과적으로 사용할 수 있게 된다. 즉, 낭비하는 공간을 줄여서 많은 부분을 저장하여 전체적인 질의 비용이 감소한다고 볼 수 있다. 또한, 고려하는 단위를 세분화하여 의미 있는 영역으로 나누어 선택함으로써, 자주 요청되는 질의에 대해서 선별적인 선택이 가능하게 되어 요청되는 질의의 많은 부분을 효과적으로 처리함으로써 질의 비용이 감소한 것으로 분석할 수 있다.

그림 11은 밀집 영역에 요청되는 질의의 비율에 따른 기존 방법의 비교 결과이다. 질의가 집중되는 비율과 성능 향상의 정도가 비례적으로 변하는 것을 알 수 있다. Q0의 경우, 즉, 요청되는 질의의 영역이 데이타 큐브 전체에 대해서 균등하게 분포하는 경우에도 10% 이상의 성능 향상을 나타내고 있다. 실험 2를 통해 제안한 방법에서 밀집 영역을 효과적으로 나타낼 수 있도록 단위 영역을 구성하는 것이 중요하다는 것을 알 수 있다.

그림 12의 결과는 제안하는 방법이 예상 질의 패턴과 실제 요청되는 질의가 다른 경우에 대해서 평균 질의 비용을 비교한 것으로 부분적으로 차이가 있지만 대부분의 경우 전체적으로 유사한 질의 감소 패턴을 보인다. 전체적으로 감소하는 비율은 예상 질의 패턴과 유사한 분포로 질의가 요청될수록 큰 폭으로 감소함을 볼 수 있다.

그림 13은 예상 질의 패턴과 요청 질의 패턴이 다르게 나타날 때, 기존 방법과 비교하여 제안한 방법의 성능 향상 정도를 나타낸다. 요청되는 질의가 예상한 질의와 유사할수록 기존 방법에 비해 높은 성능 향상을 나타냄을 알 수 있다.

본 논문에서 제안한 방법은 다양한 질의 요청 환경에 대해서 기존 방법보다 나은 결과를 나타내고 있다. 차원별로 분할되어진 단위 영역이 질의들이 집중적으로 접근하는 부분들을 효과적으로 표현할 수 있도록 구성되는 경우에 현저한 질의 비용 감소에 따른 성능 향상을 나타낼 수 있다. 또한 실제 질의 패턴과 유사한 질의 분포를 예상하는 것도 성능 향상의 중요한 요인이 될 수 있다.

6. 결 론

데이타 큐브의 일부분을 실체화하여 선별적으로 저장하는 기법은 OLAP 시스템에서 질의 응답 시간을 줄이는 효과적인 방법으로, 실체화하는 부분은 사용자의 질의 요청 패턴을 잘 반영하고 데이타 큐브 내에 존재하는 의존 관계를 잘 고려하여 선택해야 한다. 기존 연구 [10]에 있어서 선별의 기본 단위는 집계 연산 애트리뷰트의 조합에 의해 구성되는 뷰 단위로, 질의들이 집중적으로 접근하는 부분들을 효과적으로 반영하지 못하였다. 본 논문에서는 데이타 큐브에서 세분화된 단위의 실체화 방법을 제시하였다. 세분화된 큐브의 표현을 위해 AND-OR 큐브 그래프와 AND-OR 최소 비용 그래프를 정의하였다. 그리고, 실체화할 부분을 선택하는 알고리즘을 제시하였다.

본 논문에서는 제안한 방법의 유용성을 평가하기 위해 다양한 질의 유형에 대해서 평균 질의 비용을 기준으로 성능 향상 정도를 측정하였다. 실험 결과는 제안한 방법이 기존 방법보다 주어진 저장 공간을 효과적으로 사용하고 성능 향상 측면에서 좋은 결과를 나타내고 있음을 보여준다. 또한 질의 패턴의 다양한 환경에 대해서도 제안한 방법이 우수한 결과를 나타낸다는 것을 알 수 있다.

참 고 문 헌

[1] V. Harinarayan, A. Rajaraman, and J. C. Ullman. Implementing data cubes efficiently. In Proceedings of ACM SIGMOD, pages 205-227, 1996.

[2] H. Karloff, M. Mihail. On the Complexity of the View-Selection Problem. In Proceedings of ACM PODS, 1999.

[3] A. Y. Levy, A. Rajaraman, A. O. Mendelzon, Y. Sagiv, D. Srivastava. Answering queries using views. In Proceedings of ACM PODS, 1995.

[4] K. A. Ross, D. Srivastava. Fast Computation of Sparse Datacubes. In Proceddings of the 23rd International VLDB Conference, pages 116-125, 1997.

[5] D. Srivastava, S. Dar, H. V. Jagadish and A. Y. Levy. Answering Queries with Aggregation Using Views. In Proceddings of the 22rd International VLDB Conference, pages 116-125, 1996.

[6] A. Shukla, P. M. Deshpande, J. F. Naughton and K. Ramasamy. Storage Estimation for Multidimensional Aggregates in the Presence of Hierarchies. In Proceeding of the 22nd Int. VLDB Conference, pages 522-531, 1996.

[7] A. Shukla, P. M. Deshpande and J. F. Naughton. Materialized View Selection for Multidimensional Datasets. In Proceeding of the 22nd International VLDB Conference, pages 522-531, 1996.

[8] H. Gupta, V. Harinarayan, A. Rajaranan, and J. C. Ullman. Index Selection for OLAP, In proceedings of ICDE, 1997.

[9] M. J. Kim. Fine Granule View Materialization in Dimension Hierarchy-based Data Cube, MS Thesis, KAIST, 2000.

[10] H. Gupta, Selection of Views to Materialize in a Data Warehouse, In proceedings of ICDT, pages 98-112, 1997.

김 민 정

1998년 연세대학교 컴퓨터과학과 학사. 2000년 한국과학기술원 전산학과 석사. 현재 LG전자기술원 연구원. 관심분야는 데이타베이스, OLAP, 데이타 마이닝 등임

정 연 돈

정보과학회논문지 : 데이타베이스 제 28 권 제 1 호 참조

박 웅 제

1993년 서강대학교 컴퓨터공학과 학사. 1995년 한국과학기술원 전산학과 석사. 1995년 ~ 현재 한국과학기술원 전산학과 박사과정. 1995년 ~ 현재 (주)다우데이타시스템 연구소 책임 연구원. 관심분야는 OLAP, Data Warehouse, 분산 컴퓨팅, KMS 임.

김 명 호

정보과학회논문지 : 데이타베이스 제 28 권 제 1 호 참조