MOLAP 시스템을 위한 다차원 저장구조의 설계기법

이종학, 이성원°
대구가톨릭대학교
{jhlee11, seongwon}@cu.ac.kr

A Design Method of Storage Structures for MOLAP Systems

Jong-Hak Lee, Seong-Won Lee°
Catholic University of Daegu

요 약

다차원 온라인 분석처리 시스템(MOLAP)에서 집계 연산은 중요한 기본 연산이다. 기존의 MOLAP 집계 연산은 다차원 배열구조를 기반으로 한 파일구조에 대해서 연구되어 왔다. 다차원 배열구조는 편중된 분포를 갖는 데이터에서는 잘 동작하지 못한다는 단점이 있다. 본 논문에서는 편중된 분포에도 잘 동작하는 다차원 파일구조를 사용한 MOLAP 저장구조의 물리적 설계기법을 제안한다. 실험결과에 의하면 이차원 파일구조의 경우 집계 연산처리를 위한 저장구조의 성능이 일곱 배 이상까지 향상됨을 확인하였다. 삼차원 이상의 파일구조에 대해서는 더욱더 큰 성능향상이 예상된다. 이러한 성능의 향상은 제안된 MOLAP 저장구조의 물리적 설계기법이 매우 유용함을 나타내는 것이다.

1. 서 론

데이터 웨어하우스에서 온라인 분석처리(On-Line Analytical Processing: OLAP)는 사용자가 의사 결정에 필요한 지식을 찾아내기 위해 대량의 데이터를 쉽게 분석할 수 있도록 도와주는 데이터베이스 응용이다[6]. 의사 결정에 있어서는 개별적인 레코드들보다 레코드들을 요약한 경향이 중요하기 때문에 상당수의 OLAP 질의들이 데이터를 요약하는데 사용되는 집계(aggregation) 연산을 포함하고 있다. 그런데, 집계 연산들은 처리 비용이 매우 큰 연산이기 때문에 집계 연산의 처리 성능은 OLAP 시스템의 성능에 큰 영향을 미치는 중요한 요소이다[1, 5, 4].

OLAP에서는 데이터를 다차원 배열로 모델링하는 다차원 데이터 모델을 사용한다[6]. 다차원 데이터 모델은 데이터를 분석의 대상이 되는 측정값(measure)들과 측정값을 결정하는 차원(dimension)으로 구분한다. 그리고 각 차원은 다차원 배열의 하나의 축(axis)으로 대응시키고 측정값은 배열의 셀(cell)에 저장된 값으로 대응시킨다. 다차원 데이터 모델은 차원들의 값의 변화에 따른 측정값의 변화를 분석하는 OLAP 사용자의 논리적 사고방식에 적합하다고 알려져 있다[6].

다차원 배열을 이용하여 OLAP 데이터를 저장하는 다차원 OLAP(Multidimensional OLAP: MOLAP)에서의 집계 연산처리에 대한 대부분의 기존 연구는 선 계산된(precomputed) 집계 연산 결과를 저장하여 두는 정적인 방법을 주로 사용한다. 그러나 이 방법은 일부 집계 연산들의 결과를 저장해두기 때문에 저장 공간의 오버헤드와 주기적인 갱신에 따른 오버헤드가 있을 뿐만 아니라, 특정의 집계 연산처리에만 효과가 있으며, 일반적으로 OLAP에서 필요로 하는 모든 집계 연산처리에 대한 효과는 크지 않게 된다.

MOLAP에서의 동적인 집계 연산처리에서는 다차원 배열과 압축된 다차원 배열[4]을 대상으로 한 집계 연산처리 방법이 연구되었다. 그러나 다차원 배열은 편중된 분포를 갖는 데이터를 잘 처리하지 못하는 단점이 있으며, 압축된 다차원 배열은 각 차원의 값이 서로 유사한 셀들을 같은 페이지 내에 저장되게 하는 다차원 클러스터링(multidimensional clustering)의 특성을 파괴하여 영역 질의 등 다른 OLAP 연산들의 성능이 저하되는 단점이 있다. 따라서 본 논문에서는 다차원 클러스터링 특성을 유지하면서 편중된 분포의 데이터들을 잘 처리할 수 있는 다차원 파일구조를 사용하여 OLAP 연산들의 성능을 최적으로 향상시킬 수 있는 MOLAP 저장구조의 물리적 설계기법[7]을 제시한다.

다차원 파일구조는 다차원 클러스터링을 지원하는 파일구조로서, 여러 개의 속성으로 구성된 질의를 효과적으로 처리할 수 있다. 효과적인 클러스터링을 위해서는 레코드들을 그룹화 하여 페이지 단위로 저장할 때, 질의처리 시에 액세스되는 전체 페이지의 개수를 최소화하는 방안을 고려하여야 한다. 즉, 빈번히 함께 액세스되는 레코드들을 같은 페이지 내에 저장함으로써, 질의처리 시 액세스되는 페이지의 개수를 최소화 하는 것이 필요하다[2].

2. 다차원 파일구조를 이용한 집계 연산처리

먼저 집계 연산처리와 관련된 용어들을 정의하면 다음과 같다[8]. *집계 연산*은 데이터를 주어진 속성들의 값에 따라 여러 개의 그룹으로 나눈 후 주어진 집계 함수를 적용하여 각 그룹당 하나씩의 값을 구하는 연산이다. 구성 속성 중 집계 연산에서 레코드들을 여러 개의 그룹으로 나누는 기준이 되는 속성을 *그룹화 속성(grouping attribute)*이라 하고, 집계 함수가 적용되는 속성, 즉 분석의 대상이 되는 측정값을 나타내는 속성을 *집계 속성(aggregated attribute)*이라 한다. 그리고 그룹화 속성들의 도메인의 카티젼 곱을 *그룹화 도메인 공간(grouping domain space)*이라 하며, 그룹화 도메인 공간의 일부분을 *그룹화 영역(grouping region)*이라 한다. 다음으로, 집계 연산을 위하여 그룹화 도메인 공간을 한 개 이상의 그룹화 영역으로 분할한 경우, 이를 *집계 윈도우(aggregation window)*라 한다. 집계 윈도우를 대상으로 한 집계 연산을 *부분집계 연산(partial aggregation operation)*이라 한다.

페이지 영역 P를 그룹화 속성들의 집합이 G인 그룹화 도메인 공간으로 프로젝션한 결과를 G에 대한 P의 *페이지 그룹화 영역(page grouping region)*이라 하고 이를 Π_GP라 표시한다. 그리고 페이지 영역 P가 임의의 영역 Q와 겹칠 때 간단히 *페이지 P와 영역 Q가 겹친다*고 한다. 또한, 페이지 그룹화 영역 Π_GP가 집계 윈도우 W와 겹칠 때 *페이지 P와 집계 윈도우 W가 겹친다*고 한다. 마지막으로, 집계 윈도우 W와 겹치는 페이지를 W의 *부분집계 페이지(partial aggregation page)*라 한다. 다음은 지금까지의 용어를 사용한 집계 연산처리 방법에 대한 예제이다.

그림 1은 집계 연산처리를 위한 다차원 파일구조를 나타낸다. 그림 1의 다차원 파일구조에는 세 개의 구성 속성 X, Y, Z가 있으며, 전체 도메인 공간은 모두 여섯 개의 페이지 영역 A, B, C, D, E, F로 분할되어 있으며, 각 영역에 속하는 레코드들은 같은 데이터 페이지에 저장되어 있음을 나타낸다. 그림 1에서 X와 Z를 그룹화 속성이라 하면, 그룹화 도메인 공간은 X[0, 99] × Z[0, 99]이며, 이 공간의 일부분이 그룹화 영역이다. 그리고 집계 연산을 위해 분할된 X[0,49] × Z[0, 49], X[0, 49] × Z[50, 99], X[50, 99] × Z[0, 49], X[50, 99] × Z[50, 99]의 네 개의 그룹화 영역이 집계 윈도우이다. 그림 1에서 집계 윈도우 X[0, 49] × Z[0, 49]의 부분집계 페이지들은 C, D, E이다.

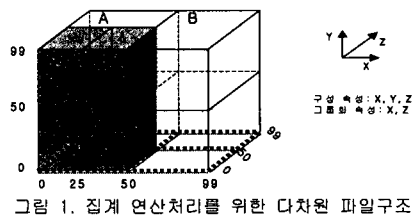


그림 1. 집계 연산처리를 위한 다차원 파일구조

집계 연산처리에서는 일반적으로 주 기억장치 크기의 한계 문제를 해결하기 위해서, 그룹화 도메인 공간을 여러 개의 집계 윈도우로 나누어 처리하는 방법을 사용한다[8]. 즉, 그룹화 도메인 공간을 대상으로 하는 집계 연산을 집계 윈도우를 대상으로 하는 부분집계 연산들로 나누어 수행하는 것이다. 집계 윈도우의 크기가 작아지면, 부분집계 연산의 결과 크기도 작아진다. 따라서 결과 테이블의 크기가 주어졌을

때, 부분집계 연산들의 결과가 결과 테이블에 들어가도록 집계 윈도우들을 선택하여 집계 연산처리에 사용할 수 있다.

이와 같은 기법을 사용한 집계 연산처리의 절차를 요약하면 다음과 같다. 첫 번째 단계로 그룹화 도메인 공간을 여러 개의 집계 윈도우로 분할한다. 그리고 두 번째 단계로 각 집계 윈도우를 하나씩 순회하면서 부분집계 연산을 수행한다. 두 번째 단계에서는 먼저 부분집계 연산에 사용될 영역 질의를 구성한다. 사용되는 영역 질의는 그룹화 속성들에 대해서는 해당 집계 윈도우를 대상으로 하고, 나머지 속성들에 대해서는 범위 조건이 주어진 속성의 경우에는 도메인의 일부분인 주어진 범위로 하고, 범위 조건이 주어지지 않은 속성의 경우에는 전체 도메인을 범위로 한다. 그리고 이러한 영역 질의들로서 집계 윈도우에 속하는 레코드들을 탐색하면서 부분집계 연산을 수행한다. 부분집계 연산의 중간 결과는 주기억장치의 결과 테이블에 유지된다.

이와 같이 다차원 파일구조는 집계 윈도우에 속하는 레코드들을 효율적으로 검색할 수 있다. 그 이유는 다차원 파일구조들은 다차원 클러스터링 특성을 지원함으로써 영역 질의들을 효과적으로 처리하기 때문이다. 그러나 다차원 파일구조를 집계 연산을 위해 사용하는 경우, 집계 연산에서 주어진 그룹화 속성과 각 속성에 주어진 범위 조건에 따라 질의 영역들의 모양이 일정하지 않고 다양하게 주어진다. 따라서 질의 영역들의 형태에 따라 집계 연산을 더욱더 효율적으로 처리할 수 있는 다차원 저장구조의 물리적 설계기법이 필요하다.

3. 다차원 저장구조의 설계원리 및 영역 분할전략

본 절에서는 설명의 편의를 위하여 구성 속성이 두 개인 이차원 도메인 공간상에서의 질의 영역과 색인 페이지 영역간의 상호관계를 통하여 다차원 저장구조의 설계원리와 페이지 영역 분할전략을 설명한다. 다차원 저장구조에서 영역 질의의 처리를 위하여 액세스해야 할 데이터 페이지의 개수는 주어진 질의 영역과 도메인 공간의 분할 상태를 나타내는 페이지 영역의 모양이 비슷할수록 액세스해야 할 페이지의 개수가 적게 된다[3]. 따라서 본 논문에서는 이와 같은 원리를 이용하여 MOLAP의 집계 연산처리를 위한 영역 질의들의 정보를 이용하여 다차원 파일구조를 구성함으로써 집계 연산처리의 성능을 향상 시키고자 한다.

이차원 공간상에서 데이터가 비균일하게 분포하게 되면, 도메인 공간내의 위치에 따라 데이터 밀집도가 다름으로 인하여 페이지 영역의 크기가 위치에 따라 달라지게 된다. 이와 같은 경우에는 각 질의 영역의 크기에 대해 위치에 따른 데이터 밀집도를 가중치(weight)로 곱한 질의 영역의 형태(*정규화된 질의 영역*)로서 페이지 영역의 최적 구간비를 계산할 수 있다. 즉, 서로 다른 크기의 페이지 영역들로 나누어져 있는 이차원 공간상에서, 임의의 위치에 주어지는 n개의 질의 영역 $q_i(x)\times q_i(y)$ $(i=1,\cdots,n)$에 대해 각 질의 영역의 레코드 밀집도를 $d_i(=nr_i/q_i(x)\times q_i(y)$, 단, nr_i는 질의 영역 내의 레코드 수이다.)라 할 때, 각 질의 영역과 교차하게 되는 페이지 영역의 총 개수를 최소로 하는 페이지 영역의 최적 구간비($p(x)$: $p(y)$)는 $\sum_{i=1}^{n} q_i(x)\sqrt{d_i}$: $\sum_{i=1}^{n} q_i(y)\sqrt{d_i}$로 계산한다[3].

다음은 다차원 저장구조의 페이지 영역 분할전략에 관한 설명이다. 아래 정리 1은 이차원 도메인 공간상에 임의의 위치에 주어지는 특정 모양의 한 질의 영역이 특정 크기의 한 페이지 영역과 교차하게 되는 질의 영역의 위치 범위의 크기는, 그 페이지 영역의 모양이 주어진 특정 질의 영역의 모양과 같을 때 최소가 됨을 나타낸다.

정리 1 구간비가 $q_x : q_y$인 $q_x\times q_y$형태의 질의 영역이 이차원 공간상에서 임의의 위치에 주어질 때, 크기가 B인 $p(x)\times p(y)$ 형태의 한 페이지 영역과 교차하게 되는 질의 영역의 위치 범위의 크기는 페이지 영역의 구간비가 주어진 질의 영역의 구간비와 같을 때 최소가 된다.

증명: 참고문헌[3] 참조. □

정리 1을 이용하면, 페이지 영역의 분할 시 분할된 페이지 영역의 구간비가 최적 구간비에 가깝게 되는 분할 축을 선택할 수 있다. 그림 2는 주어진 최적 구간비 $(a:b)$와 같은 모양을 갖는 $a\times b$형태의 질의 영역 Q가 이차원 도메인 공간상에 임의의 위치에 주어졌다고 가정하고, $p(x)\times p(y)$형태의 페이지 영역 P가 두 개의 페이지 영역으로 분할된 후의 한 페이지 영역과 교차하게 되는 질의 영역의 위치 범위(음영 부분)인 LQ를 나타낸다. 그림 2에서 (a)는 분할 축으로 X축을 선택한 경우의 LQ_x를 나타내고, (b)는 분할 축으로 Y축을 선택한 경우의 LQ_y를 나타낸다.

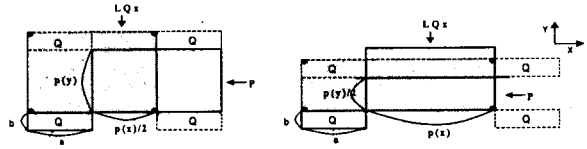
그림 2에서 X축을 분할한 경우의 LQ_x의 크기는

$$SIZE(LQ_x) = (p(x)/2+a)(p(y)+b) \quad (1)$$

이고, Y축을 분할한 경우의 LQ_y의 크기는

$$SIZE(LQ_y) = (p(x)+a)(p(y)/2+b) \quad (2)$$

이다.



(a) X축 분할 (b) Y축 분할

그림 2. 분할 후의 한 페이지 영역과 교차하게 되는 질의 영역의 위치 영역

정리 1에 의하여 LQ의 크기는 페이지 영역의 구간비가 질의 영역의 구간비와 같을 때 최소가 되므로, 이 LQ의 크기가 작을수록 페이지 영역의 구간비가 질의 영역 Q의 구간비에 가깝게 된다. 따라서 이 LQ의 크기가 작게 되는 축을 분할 축으로 선택함으로써 분할 후의 페이지 영역의 구간비를 주어진 최적 구간비에 더 근접하게 할 수 있다.

4. MOLAP 저장구조의 물리적 설계기법

제 3절에서의 이차원 파일구조에 대한 질의 영역과 페이지 영역간의 상호 관계와 영역 분할전략을 N차원으로 확장한 다차원 파일구조에 대하여 다차원 집계연산을 최적으로 처리할 수 있는 다차원 저장구조의 물리적 설계 과정은 다음과 같은 세 가지 단계로 구성된다. 첫째, MOLAP 시스템에서 주어지는 영역-집계 질의들의 처리에 필요한 n개의 질의 영역에 대하여 정규화를 취한다. 즉, N차원의 도메인 공간에 주어진 임의의 질의 영역 $q(1)\ X\ q(2)\ \cdots\ q(j)\ \cdots\ X\ q(N)$에 대한 정규화는 다음과 같다. 즉, 먼저 질의 결과에 의한 질의 영역내의 레코드 개수 nr을 이용하여 레코드 밀집도 d를 $nr/q(1)q(2)\cdots q(i)\cdots q(N)$로 구하여, 질의 영역을 이루는 각 축의 구간 $q(j)$에 가중치 $d^{1/N}$을 곱하여 정규화된 질의 영역의 형태 $q(1)d^{1/N}X\ q(2)d^{1/N}\cdots q(j)d^{1/N}\cdots X\ q(N)d^{1/N}$를 얻는다.

둘째, 정규화된 모든 질의 영역에 대해서 각 축별로 구간의 크기를 합산한 값의 비율로서 페이지 영역의 최적 구간비(a(1):a(2):…:a(j):…:a(N))를 구한다.

셋째, 최적 구간비에 가장 가까운 페이지 영역들로 구성된 다차원 파일구조를 구축한다. 여기서는 제 3절의 영역 분할정책을 N차원으로 확장하여 적용한다. 즉, 계속되는 레코드의 삽입으로 다차원 파일구조의 데이터 페이지에 오버플로우가 발생하면, 이 데이터 페이지에 대응하는 페이지 영역은 구간 이등분 정책을 사용하여 같은 크기의 두 영역으로 분할되고, 원 데이터 페이지의 레코드들은 분할된 페이지 영역에 대응하는 두 개의 데이터 페이지로 나뉘어 저장된다. 이때 페이지 영역의 구간 이등분 정책으로 제 3절의 영역 분할정책을 N차원으로 확장하여 적용하는 것이다. 즉, 둘째 단계에서 결정된 최적 구간비 (a(1):a(2):…:a(j):…:a(N))와 같은 모양을 가지는 가상의 질의 영역(a(1) X a(2) … a(j) …X a(N))이 임의의 위치에 주어진다고 가정하고, 분할이 요구되는 페이지 영역(p(1) X p(2) … p(j) …X p(N))이 각 축에 대해 구간 이등분에 의한 분할 후의 한 페이지 영역과 교차하게 되는 질의 영역의 위치 범위의 크기(예를들어, j번째 축의 구간을 이등분 했을 때 그 크기는 (p(1) + a(1))(p(2) + a(2)) … (p(i)/2 + a(i)) … (p(N) + a(N)이다.)를 각각 계산한 다음 그 값이 가장 작게 되는 축을 분할 축으로 선택한다.

5. 성능 평가

본 절에서는 MOLAP 저장구조의 물리적 설계기법의 유용성을 다양한 실험을 통하여 제시한다. 제 5.1절에서는 성능평가를 위하여 사용된 실험 환경에 대하여 기술하고, 제 5.2절에서는 실험 결과를 제시하고 이를 분석한다.

5.1 실험 환경

본 실험에서는 MOLAP 저장구조로 사용한 다차원 파일구조로 계층 그리드 파일을 사용하여 100,000개의 레코드를 포함하는 이차원과 삼차원의 두 종류의 다차원 저장구조를 구축하였다.

이차원 저장구조의 구축에 사용한 레코드의 분포 특성은 균일 분포와 비균일 분포로 구분한다. 비균일 분포의 데이터로는 각 축의 값이 $[-2^{31}, 2^{31}-1]$인 도메인 내에서 표준 편차 σ가 $2^{31}\times 2/5$인 $N(0,\sigma^2)$의 정규 분포를 취하게 한다. 그리고 삼차원 저장구조의 구축에 사용한 데이터는 비균일 분포의 데이터로서, 각 축의 값은 $[-2^{31}, 2^{31}-1]$의 구간 내에서 표준 편차 σ가 $2^{31}\times 2/5$인 $N(0,\sigma^2)$의 정규 분포를 취하도록 한다.

집계 연산의 패턴을 구성하기 위하여 사용한 질의 영역들의 형태는 이차원의 질의 영역인 경우에는 질의 영역의 구간비가 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256, 1:512, 및 1:1024인 각각에 대해서, 질의 영역의 크기에 따라 다음과 같이 구성한다: (1) 크기가 도메인 공간의 1/200로서 대영역인 L1, L2, L4, L8, L16, L32, L64, L128, L256, L512, 및 L1042형태의 질의 영역, (2) 크기가 도메인 공간의 1/2000로서 중영역인 M1, M2, M4, M8, M16, M32, M64, M128, M256, M512, 및 M1024형태의 질의 영역, (3) 크기가 도메인 공간의 1/20000로서 소영역인 S1, S2, S4, S8, S16, S64, S128, S256, S512, 및 S1024형태의 질의 영역 등이다. 그리고 삼차원의 질의 영역인 경우에는 크기가 도메인 공간의 1/20000로서 소영역인 경우에만 한정하여 질의 영역의 구간비가 각각 1:1:1, 1:2:4, 1:4:16, 1:8:64, 및 1:16:256인 S1_1_1, S1_2_4, S1_4_16, S1_8_64, 및 S1_16_256 형태의 질의 영역 등을 사용한다.

5.2 실험 결과

첫 번째 실험에서는 이차원 균일 분포의 데이터에 대하여 이차원의 계층 그리드 파일로서 서로 다른 구간비의 페이지 영역을 갖는 여러 개의 MOLAP 저장구조들을 생성하고, 각각에 대하여 다양한 형태의 질의 영역들을 갖는 집계 연산들을 처리할 때 발생하는 평균 페이지 액세스 수를 측정하였다. 실험에 사용된 질의 영역의 형태는 대영역인 L1, L2, L4, 중영역인 M8, M16, M32, 소영역인 S64, S128, S256 등이다. 그림 3은 실험 결과를 그래프로 나타낸 것이다. 그림 3에서 가로 축은 이차원 MOLAP 저장구조를 구성하는 페이지 영역의 구간비를 나타내고 세로 축은 각 질의 영역의 평균 페이지 액세스 수를 나타낸다. 모든 질의 영역들에 대하여 각 축의 구간 크기를 더한 값의 비는 1:57로 나타났으며, 그림 3에서 알 수 있는 바와 같이 이 비율과 가장 유사한 1:64를 페이지 영역의 구간비로 가지는 이차원 MOLAP 저장구조에서 가장 좋은 성능을 보인다. 이와 같은 실험 결과는 데이터가 균일하게 분포하면, 주어진 다양한 형태의 질의 영역들에 의해 교차되는 페이지 영역의 개수를 최소로 하는 MOLAP 저장구조를 구성하는 페이지 영역의 최적 구간비는 모든 질의 영역들에 대해 각 축별로 구간 크기를 더한 값의 비로서 계산할 수 있음을 보이는 것이다.

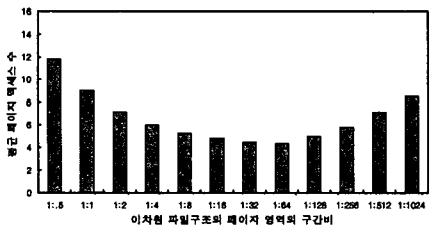


그림 3. 이차원 균일분포 데이터에 대한 서로 다른 구간비의 페이지 영역을 갖는 이차원 파일구조별 영역 질의처리 성능

두 번째 실험에서는 이차원 비균일 분포의 데이터에 대하여 첫 번째 실험과 동일한 실험을 수행하였다. 본 실험에서는, 먼저 각 질의 영역들을 정규화하고, 정규화된 질의 영역들에 대하여 각 축의 구간 크기를 더한 비는 1:3.5로 계산되었으며, 이 비율과 가장 유사한 구간비의 페이지 영역을 가지는 이차원 MOLAP 저장구조에서 가장 좋은 성능을 보였다.

세 번째 실험에서는 삼차원 비균일 분포의 데이터에 대하여 두 번째 실험과 동일한 실험을 수행하였다. 실험에 사용된 질의 영역의 형태는 S1_1_1, S1_2_4, S1_4_16, S1_8_64, 및 S1_16_256 등의 다섯 가지로, 집계 연산의 영역 질의 패턴을 구성하기 위하여 각각 200개씩 도메인 공간상에 균일하게 분포하도록 하였다. 정규화된 모든 질의 영역들에 대하여 각 축의 구간 크기를 더한 값의 비는 1:6:68로 계산되었으며, 이 비율과 같은 구간비의 페이지 영역을 갖는 MOLAP 저장구조에서 가장 좋은 성능을 보인다.

마지막으로, 네 번째 실험에서는 이차원 MOLAP 저장구조에 대하여 물리적 설계기법을 이용하여 구성한 이차원 파일구조가 얼마나 성능개선 효과가 있는 지를 알아본다. 먼저, 여섯 가지의 이차원 질의 영역의 형태인 M1, M4, M16, M64, M256, 및 M1024에 대하여, 각 형태별로 1000개의 질의 영역들이 도메인 공간상에 균일하게 주어지는 여섯 가지의 질의 패턴을 생성한다. 그리고 각 질의 패턴에 대하여 최적의 구간비(질의 패턴을 구성하는 질의 영역들의 구간비와 동일)를 갖는 페이지 영역들로 구성된 이차원 파일구조를 생성하여 그 질의 패턴을 처리할 때 발생하는 평균 페이지 액세스 수를 구하고, 이 값에 대한 구간비가 1:1인 페이지 영역들로 구성된 이차원 파일구조에서 같은 질의 패턴을 처리할 때 발생하는 평균 페이지 액세스 수의 비율을 측정한다. 그림 4는 이에 대한 실험 결과를 나타낸 것이다. 가로축은 각 질의 패턴을 구성하는 질의 영역들의 구간비를 나타내며, 세로축은 제안된 기법을 사용하는 경우의 성능 이득이 몇 배인가를 나타낸다.

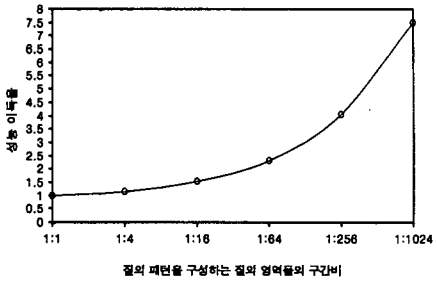


그림 4. 집계 연산을 구성하는 질의 영역의 구간비별 이차원 MOLAP 저장구조의 성능 효율.

그림 4에서 나타난 바와 같이 집계 연산에 필요한 질의 영역의 모양이 정방형(구간비가 1:1)에서 멀어질수록 제안된 물리적 설계기법을 사용하는 경우의 성능개선 효과가 뚜렷해짐을 볼 수 있다. 즉, 질의 영역의 구간비가 1:1024인 경우 집계 연산처리 성능이 일곱 배 이상으로 향상됨을 볼 수 있으며, 구간비가 더 커질수록 더욱더 향상될 수 있음을 나타낸다.

6. 결 론

데이터 웨어하우스의 MOLAP 시스템에서 집계 연산은 중요한 기본 연산이다. 그리고 집계 연산은 처리 비용이 매우 큰 연산이기 때문에 집계 연산처리의 성능은 시스템의 성능에 큰 영향을 미치는 중요한 요소이다. 본 논문에서는 다차원 클러스터링 특성을 유지하면서 편중된 분포의 데이터들을 잘 처리할 수 있는 다차원 파일구조를 사용하여 집계 연산처리의 성능을 최적으로 보장할 수 있는 다차원 저장구조의 물리적 설계기법을 제안하였다. 실험 결과에 의하면, 주어진 질의 패턴과 데이터 분포에 따라 최적의 MOLAP 저장구조를 구성할 수 있었으며, 이차원 파일구조의 경우 질의 영역의 모양이 편향된 정도에 따라 기존의 정방형 모양의 페이지 영역으로 구성된 이차원 파일구조에 비해 집계 연산에 필요한 영역 질의처리의 성능이 그림 4에서와 같이 급격히 향상되는 것으로 나타났다. 특히, 질의 영역의 구간비가 1:1024인 경우에는 영역 질의처리의 성능이 일곱 배 이상으로 향상됨을 볼 수 있었다. 이것은 제안된 기법이 실제적으로 매우 유용함을 보여주는 것이다.

참고 문헌

[1] C. T. Ho, R. Agrawal, N. Megiddo, and R. Srikant, "Range Queries in OLAP Data Cubes," In *Proc. Int'l Conf. on Management of Data*, pp. 73-88, ACM SIGMOD, Tucson, Arizona, June 1997.

[2] C. T. Yu et al., "Adaptive Record Clustering," *ACM Trans. on Database Systems*, Vol. 10, No. 2, pp. 180-204, June 1985.

[3] J. Lee, Y. Lee, K. Whang, and I. Song, "A Region Splitting Strategy for Physical Database Design of Multidimensional File Organizations," In *Proc. Int'l Conf. on Very Large Data Bases*, pp. 416-425, Athens, Greece, Aug. 1997.

[4] J. Li, D. Rotem, and J. Srivastava., "Aggregation Algorithms for Very Large Compressed Data Warehouses," In *Proc. Int'l Conf. on Very Large Databases*, pp. 651-662, Edinburgh, Scotland, UK, Sept. 1999.

[5] S. Agarwal et al., "On the Computation of Multidimensional Aggregations," In *Proc. Int'l Conf. on Very Large Data Bases*, pp. 506-512, Mumbai(Bombay), India, Sept. 1996.

[6] S. Chaudhuri, and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," *ACM SIGMOD Record*, Vol. 26, No. 1, pp. 65-74, Mar. 1997.

[7] S. Finkelstein et al., "Physical Database Design for Relational Databases," *ACM Trans. on Database Systems*, Vol. 13, No. 1, pp. 91-128, Mar. 1988.

[8] Y. Lee, K. Whang, Y. Moon, and I. Song, "A One-Pass Aggregation Algorithm with the Optimal Buffer Size in Multidimensional OLAP," In *Proc. Int'l Conf. on Very Large Data Bases*, pp. 790-801, Hong Kong, China, Aug. 2002.