Journal of Korea Multimedia Society Vol. 8, No. 3, March 2005(pp. 297-312)

데이터 웨어하우스의 다차원 온라인 분석처리 시스템을 위한 저장구조의 물리적 설계기법

이 종 학†

요 약

데이터 웨어하우스의 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. 먼저, 균일분포를 갖는 데이터에 대해서 집계 연산처리 성능이 다차원 파일구조상의 질의 영역의 모양과 다차원 파일구조의 도메인 공간을 이루는 페이지 영역의 모양 사이의 유사성에 따라 크게 영향 받음을 보이고, 이러한 특성을 이용하여 다차원 파일구조를 설계함으로써 다차원 온라인 분석처리의 성능을 향상시킨다. 그리고 편중된 분포에 대해서는 질의 영역별로 가중치를 부여한 정규화된 질의 영역의 모양을 이용함으로써 데이터의 분포에 따른 영향을 설계에 반영한다. 또한 본 논문에서는 실험을 통하여 이론적으로 제안한 MOLAP 저장구조의 물리적 설계기법이 실제 환경에서 정확히 동작함을 보인다. 실험결과에 의하면 이차원 파일구조의 경우 집계 연산처리를 위한 저장구조의 성능이 일곱 배 이상으로 향상됨을 확인 하였다. 삼차원 이상의 파일구조에 대해서는 더욱더 큰 성능향상이 예상된다. 이러한 성능의 향상은 제안된 MOLAP 저장구조의 물리적 설계기법이 매우 유용함을 나타내는 것이다.

A Physical Design Method of Storage Structures for MOLAP Systems of Data Warehouse

Jong-Hak Lee†

ABSTRACT

Aggregation is an operation that plays a key role in multidimensional OLAP (MOLAP) systems of data warehouse. Existing aggregation operations in MOLAP have been proposed for file structures such as multidimensional arrays. These file structures do not work well with skewed distributions. This paper presents a physical design methodology for storage structures of MOLAP that use the multidimensional file organizations adapting to a skewed distribution. In uniform data distribution, we first show that the performance of multidimensional analytical processing is highly affected by the similarity of the shapes between query regions and page regions in the domain space of the multidimensional file organizations. And than, in skewed distributions, we reflect the effect of data distributions on the design by using the shapes of the normalized query regions that are weighted with data density of those query regions. Finally, we demonstrate that the physical design methodology theoretically derived is indeed correct in real environments. In the two-dimensional file organizations, the results of experiments indicate that the performance of the proposed method is enhanced by more than seven times over the conventional method. We expect that the performance will be more enhanced when the dimensionality is more than two. The result confirms that the proposed physical design methodology is useful in a practical way.

Key words: Data Warehouse(데이터 웨어하우스), OLAP(다차원 온라인 분석처리), Physical Storage Structure Design(물리적 저장구조 설계)

※ 교신저자(Corresponding Author) : 이종학, 주소 : 경북 경산시 하양읍 금락1리 330(712-702), 전화 : 053)850-2746, FAX : 053)850-2704, E-mail : jhlee11@cu.ac.kr
접수일 : 2004년 7월 5일, 완료일 : 2004년 9월 6일

† 정회원, 대구가톨릭대학교 컴퓨터정보통신공학부 교수
※ 이 논문은 2003년 대구가톨릭대학교 교비해외파견 연구지원금에 의한 것임.

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[14]. 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[13,8,19].

OLAP에서는 데이터를 다차원 배열로 모델링하는 다차원 데이터 모델을 사용한다[14]. 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. 이러한 다차원 배열을 데이터 큐브(data cube)라 부른다. 다차원 데이터 모델은 차원들의 값의 변화에 따른 측정값의 변화를 분석하는 OLAP 사용자의 논리적 사고방식에 적합하다고 알려져 있다[14].

OLAP 시스템은 OLAP 데이터의 저장 방법에 따라 크게 관계형 OLAP(Relational OLAP: ROLAP)과 다차원 OLAP(Multidimensional OLAP: MOLAP)으로 구분된다[14]. ROLAP은 관계 데이터베이스 관리 시스템(Relational DBMS: RDBMS)을 기반 시스템으로 사용하는 것으로서, 테이블을 이용하여 OLAP 데이터를 저장한다. 반면에, MOLAP에서는 다차원 데이터를 효율적으로 저장, 관리할 수 있는 다차원 데이터베이스 관리 시스템(Multidimensional DBMS: MDBMS)을 기반시스템으로 사용하는 것으로서, 다차원 배열을 이용하여 OLAP 데이터를 저장한다[17,19].

ROLAP에서의 집계 연산처리에 대해서는 RDBMS에서의 기존 연구 결과를 포함하여 많은 연구 결과가 발표되었다[13,3]. 반면에, MOLAP에서의 집계 연산처리에 대해서는 많은 연구가 이루어지지 않았으며, 대부분의 기존 연구는 선계산된(precomputed) 집계 연산 결과를 저장하여 두는 정적인 방법을 주로 사용한다. 그런데, 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 없게 된다.

MOLAP에서의 동적인 집계 연산처리에서는 다차원 배열[19]과 압축된 다차원 배열[8]을 대상으로 한 집계 연산처리 방법이 연구되었다. 그러나 다차원 배열은 편중된(skewed) 분포를 갖는 데이터를 잘 처리하지 못하는 단점이 있으며, 압축된 다차원 배열은 각 차원의 값이 서로 유사한 셀들을 같은 페이지 내에 저장되게 하는 다차원 클러스터링의 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[15]을 제시한다.

다차원 파일구조는 다차원 클러스터링(multidimensional clustering)을 지원하는 파일구조로서, 여러 개의 속성으로 구성된 질의를 효과적으로 처리할 수 있다[6,12]. 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 최소화하는 것이 필요하다[2].

일차원 클러스터링을 위해서는 해당 속성 값으로 정렬(sorting)된 순서로 레코드들을 페이지에 각각 저장함으로써 클러스터링 특성이 하나의 속성에 의해서 독점되도록 한다. 반면, 다차원 클러스터링 기법을 위해서는 파일을 구성하는 각 속성의 값이 서로 유사한 레코드들을 같은 페이지 내에 저장시켜야 하므로 클러스터링 특성이 파일을 구성하는 모든 속성에 의해서 공유되어야 한다. 다차원 파일구조에서는 이를 위하여 레코드의 삽입과 삭제시의 상황에 따라 도메인 공간(domain space)[11]을 분할하거나 병합하고, 이 결과 생성된 도메인 공간의 각 영역내의 레코드들을 같은 페이지 내에 저장하는 방법을 사용한다.

이와 같이 다차원 파일구조에서는 여러 속성들이 클러스터링 특성을 공유하게 되므로 영역 분할전략

(region splitting strategy)의 변화를 통하여 속성 별로 클러스터링의 정도를 조정할 수 있다. 기존의 다차원 파일구조에 대한 대부분의 연구에서는 영역의 분할시 주로 순환 분할전략(cyclic splitting strategy) [9,11]을 사용하도록 하여 각 페이지와 대응되는 영역의 형태가 정방형이 되도록 함으로써 모든 속성들이 클러스터링 특성을 같은 정도로 공유하도록 하고 있다. 이것은 모든 속성에 대한 질의 조건의 구간 크기가 모두 같은 정방형의 질의 영역이 주로 사용된다는 것을 가정한 것이다.

그러나 실제로 OLAP 응용에서 사용자가 요구하는 질의는 일반적으로 데이터 큐브를 구성하는 각 차원에 따라 질의 조건의 구간 크기가 다르며, 큰 구간의 질의 조건이 특정 차원에 편향되게 주어지는 경향이 있다. 예를 들어, 어떤 차원을 위한 질의 조건에서는 단일 값만을 참조하는 반면에, 또 다른 속성을 위한 질의 조건에서는 비교적 큰 구간을 가지는 값들의 범위를 참조할 수 있다. 따라서 이러한 경우에는 실제의 질의 형태가 미리 가정한 정방형의 질의 형태와 다르게 나타나므로 질의처리 시의 성능이 저하된다.

따라서 본 논문에서는 OLAP 응용을 위한 다차원 파일구조의 영역 분할전략으로 인한 차원별 클러스터링 정도의 변화에 따라 같은 OLAP 연산에 대한 처리 성능이 달라짐을 보이고, 사용자 질의 정보를 기반으로 OLAP 연산들의 성능이 최적이 되도록 하는 MOLAP 시스템을 위한 저장구조의 물리적 설계기법을 제안한다. 제안한 기법은 MOLAP 연산을 위한 영역 질의 형태를 사전에 분석함으로써 질의처리 시 발생하는 페이지 액세스 수를 최소화할 수 있는 최적의 차원별 클러스터링 정도를 구하고, 이를 만족할 수 있는 영역 분할전략을 사용하여 다차원 파일구조를 구축한다.

본 논문의 구성은 다음과 같다. 제 2절에서는 관련 연구로서 기존의 MOLAP 시스템에서 집계 연산처리를 위하여 선계산된 집계연산 결과를 저장하여 두는 정적인 방법들과 이들이 가지는 한계점들을 살펴본다. 제 3절에서는 본 논문에서 MOLAP 저장구조의 물리적 설계기법에 적용할 다차원 파일구조의 주요 특징들을 소개한다. 제 4절에서는 먼저 다차원 파일구조를 이용한 MOLAP 집계 연산처리 방법을 설명하고, 이러한 집계 연산들을 가장 효율적으로 처리할 수 있는 다차원 저장구조를 구성하기 위한 물리적 설계기법을 제시한다. 그리고 제 5절에서는 성능 평가를 위한 실험 환경과 실험 결과를 제시한다. 마지막으로, 제 6절에서는 결론을 내린다.

2. 관련 연구

OLAP 응용은 사용자가 대화식으로 정보를 분석하는 시스템이므로, 질의 처리시 빠른 응답 시간을 요구한다. 이러한 요구를 만족시키기 위해 선계산한 결과를 다차원 배열에 유지하는 방법이 널리 사용되고 있다. Gray[5] 등에 의해 제안된 CUBE BY 연산은 집계 연산에서 데이터들을 여러 개의 그룹으로 나누는 기준이 되는 속성, 즉 그룹화 속성의 가능한 모든 조합에 대한 group by들을 모두 구하는 연산이다. CUBE BY 연산이 제안된 이후, CUBE BY의 결과를 선계산하여 유지하고 이를 이용하는 연구가 많이 이루어졌다[4]. CUBE BY 연산의 결과는 각 차원의 도메인에 'all'이라는 특수한 값이 추가된 다차원 배열의 데이터 큐브에 저장될 수 있으며[5], 이 특수한 값을 갖도록 확장된 데이터 큐브를 확장 데이터 큐브(extended data cube)[1]라 한다. n개의 차원 D_1, D_2, …, D_n으로 구성된 데이터 큐브는 $\Pi_{i=1}^{n}|D_i|$($|D_i|$는 차원 D_i의 카디널리티)개의 셀을 갖는 배열로 표현되고, 이에 대한 확장 데이터 큐브는 $\Pi_{i=1}^{n}(|D_i|+1)$개의 셀을 갖는 배열로 표현된다. 그리고 D_{g1}, D_{g2}, …, D_{gm}을 그룹화 속성으로 하는 집계 질의의 결과들은, 그룹화 속성을 제외한 속성, 즉 비 그룹화 속성들의 값이 'all'인 확장 데이터 큐브의 셀에 저장된다.

확장 데이터 큐브를 사용한 집계연산의 처리방법은 데이터 큐브 전체를 대상으로 하지만, MOLAP에서는 데이터 큐브의 임의의 영역을 대상을 하는 집계 질의가 자주 사용되게 된다. 본 논문에서는 이러한 질의를 영역-집계 질의(range-aggregation query)라 정의한다. 이 질의에는 각 차원에 대하여 임의의 연속적인 범위의 값들이 조건으로 주어진다. 이러한 조건을 범위 조건(range condition)이라 한다. 범위 조건은 나이, 수입, 시간 등과 같이 순서를 부여하는 것이 자연스러운 의미를 갖는 숫자형 차원에 대해서 빈번하게 발생한다[16,1]. 예를 들어, 보험 회사에 대한 OLAP 응용에서 나이, 수입, 년도를 차원으로 하

고, 판매량을 측정값로 하는 데이터 큐브를 생각해 보자. 이 때, 나이가 40~49이고, 수입이 40,000,000~60,000,000인 사람에게 2000~2003년에 판매한 보험을 년도 별로 구분하여 구하는 질의는 영역-집계 질의이다. 이러한 질의는 자료를 다각적인 측면에서 분석하기 위하여 OLAP 응용에서 매우 유용하게 사용된다.

이와 같은 영역-집계 질의는 특수한 경우를 제외하고는 확장 데이터 큐브를 사용해도 질의 처리 속도를 개선할 수 없다. 이것은 영역-집계 질의가 비 그룹화 속성에 대해 범위 조건을 갖기 때문이다. 비 그룹화 속성에 대해 범위 조건이 주어진 질의는 도메인의 일부분인 주어진 범위에 속하는 셀들만을 집계한 값이 필요하다. 그런데, 확장 데이터 큐브는 그룹화 속성들이 가질 수 있는 각 값에 대하여, 비 그룹화 속성에 대해 전체 도메인에 걸친 모든 셀들을 집계한 값만을 저장하고 있으므로, 이러한 질의를 처리하는데는 아무 소용이 없다. 그러나 영역-집계 질의에서 그룹화 속성에만 범위 조건이 주어진 특수한 경우는 확장 데이터 큐브로 쉽게 계산할 수 있다. 그 이유는, 집계 값은 각 그룹 별로 계산되며, 그룹화 속성은 그룹을 나누기 위해서만 사용되는 속성이므로 그룹화 속성에 대한 조건은 집계 값에 영향을 미치지 않기 때문이다.

영역-집계 질의는 그룹화 속성의 유무에 따라 두 가지로 분류할 수 있다. 하나는 그룹화 속성이 없는 질의로서, 임의의 영역을 지정하고 이 영역에 속한 모든 셀들을 대상으로 하나의 집계 값을 구한다. 다른 하나는, 그룹화 속성이 있는 질의로서, 임의의 영역을 지정하고 이 영역에 속한 셀들에 대하여 주어진 그룹화 속성들의 값의 조합에 따라 집계 값을 구한다. 전자의 질의를 영역-단일그룹 질의라 하고, 후자의 질의를 영역-그룹화 질의(range-groupby query)라 한다[1]. 그리고 영역-그룹화 질의에서 각 그룹을 결정하는 그룹화 속성들의 값의 조합을 그룹 키 값(group key value)이라 한다.

영역-단일그룹 질의를 처리하는 방법에 대해서는 Ho 등[1]에 의해 연구되었다. Ho 등은 확장 데이터 큐브와는 다른 요약 정보를 추가로 선계산하여 유지함으로써 영역-단일그룹 질의를 효율적으로 처리하는 방법을 제안하였다. 특히, 영역-단일그룹 질의 중에서 가장 자주 사용되는 집계 함수인 SUM인 영역-단일그룹 질의를 영역-합(range-sum) 질의라 하고, 이 질의의 처리를 위하여 요약 정보로서 전방-합(prefix-sum) 배열을 사용하는 방법을 제안하였다. 이 방법은 질의 영역의 크기에 상관없이 항상 동일한 개수(2^n개, n은 차원의 개수)의 셀을 액세스한다는 장점을 갖는다.

그러나 이와 같은 영역-단일그룹 질의의 처리에 사용된 방법을 영역-그룹화 질의의 처리에는 사용할 수 없다. 왜냐하면, 영역-그룹화 질의에서는 각 그룹 키 값에 따라 비 그룹화 속성들에 주어진 범위 조건을 만족하는 셀들의 집계 값을 구해야 한다. 그런데 영역-단일 그룹 질의는 그룹화 속성을 고려하지 않고 주어진 영역에 속하는 모든 셀들을 집계한 값을 구하기 때문이다.

이와 같이 지금까지의 MOLAP에서의 집계 연산 처리를 위한 연구에서는 특정의 집계 연산만을 빠르게 처리할 수 있는 방법으로 다차원 배열을 이용하여 연산의 결과를 선계산하여 이를 효율적으로 이용하는 방법들이다. 따라서 이러한 방법들은 일반적인 MOLAP의 모든 집계 연산을 잘 처리할 수 없을 뿐만 아니라, 다차원 배열은 편중된(skewed) 분포를 갖는 데이터를 잘 처리하지 못하는 단점이 있다. 또한 다차원 배열은 각 차원의 값이 서로 유사한 셀들을 같은 페이지 내에 저장되게 하는 다차원 클러스터링의 특성을 파괴하여 영역 질의에 해당하는 다양한 OLAP 연산들의 성능이 저하되는 단점이 있다. 따라서 본 논문에서는 다차원 클러스터링 특성을 유지하면서 편중된 분포의 데이터들을 잘 처리할 수 있는 다차원 파일구조를 사용하여 보편적인 영역 질의와 관련된 OLAP 연산들의 성능을 향상시킬 수 있는 물리적 설계기법을 제시한다.

3. 다차원 파일구조

본 절에서는 MOLAP 시스템의 저장구조로 이용할 다차원 파일구조에 대하여 소개한다. 먼저 다차원 파일구조에 관련된 다음과 같은 용어들을 정의한다[11]. *파일*은 *속성*들의 리스트로 구성된 *레코드*들의 모임이다. 파일의 구조는 레코드를 구성하는 속성들 중에서 일부에 의해서 결정되며, 이와 같이 파일을 구성하는데 참여하는 속성들을 *구성 속성(organizing attribute)*이라 한다. 그리고 두 개 이상의 구성 속성을 가지는 파일구조를 *다차원 파일구조*라 한다.

*도메인*은 한 속성이 취할 수 있는 모든 값들의 집합이며, 다차원 파일구조 내에서 하나의 축에 해당된다. 모든 속성에 대한 도메인들의 카티젼 곱(Cartesian product)을 *도메인 공간(domain space)*이라 하고, 도메인 공간의 일부분을 *영역(region)*이라 한다. 그리고 도메인 공간에서 데이터 페이지에 해당하는 영역을 *페이지 영역(page region)*이라 하고, 사용자 질의에 해당하는 영역을 *질의 영역(query region)*이라 한다.

다차원 파일구조는 다차원 클러스트링을 지원함으로써 여러 속성에 의한 검색, 즉 다중키 액세스를 효율적으로 처리한다. 다차원 클러스트링이란 각 구성 속성의 값이 서로 유사한 레코드들을 인접하게 저장하는 것을 말한다. 다차원 클러스트링을 지원하기 위하여 다차원 파일구조는 도메인 공간을 여러 개의 영역으로 분할하고 각 영역에 속하는 레코드들을 하나의 페이지에 저장한다. 그리고 도메인 공간의 분할 상태는 디렉토리에 저장한다. 다음은 이러한 다차원 파일구조의 한 예로서 계층 그리드 파일[11]을 소개한다.

계층 그리드 파일은 디렉토리와 데이터 페이지들로 구성된다. 디렉토리는 균형 트리 구조의 구조적 특성을 가지며, 각 단계 디렉토리는 전체 공간의 분할 상태를 반영한다. 디렉토리의 최하위 단계에 있는 엔트리(디렉토리 엔트리)는 데이터 페이지를 가리킬 뿐만 아니라, 그 페이지에 할당된 영역(페이지 영역)을 표현한다. 하나의 데이터 페이지는 디렉토리 엔트리에 의해서 표현된 영역 내에 속하는 데이터 레코드들만을 저장한다. 그리고 다단계의 디렉토리 구조는 재귀적으로 구성된다. 즉, 상위 단계의 디렉토리 엔트리는 차하위 단계의 디렉토리 페이지를 가리키며, 그 디렉토리 페이지가 가리키는 영역을 표현한다.

계층 그리드 파일은 레코드가 삽입되고 삭제되는 상황에 대해서 분할과 병합을 반복함으로써 동적 변화에 적응하는 특성을 가진다. 새로운 레코드가 삽입되는 경우, 다단계의 디렉토리를 루트로부터 최하위 디렉토리까지 탐색하여 그 레코드가 속하는 페이지 영역을 찾게 되고, 그 영역에 할당된 데이터 페이지에 레코드를 삽입하게 된다. 이 결과로 데이터 페이지의 용량이 초과되면(overflow), 해당 영역은 같은 크기를 갖는 두개의 영역으로 분할(half splitting)되고 각 영역에 해당하는 새로운 두개의 데이터 페이지가 할당되며, 레코드들은 두 데이터 페이지에 분산된다. 계층 그리드 파일에서 공간을 분할할 때 나타나는 가장 큰 특징은 분할이 요구되는 영역만을 분할시키는 부분 분할 방식(local splitting strategy)[11]을 취한다는 것이다. 이러한 분할 방식은 반드시 필요한 영역만을 생성시켜 디렉토리 엔트리 수의 증가를 억제한다. 이 결과 계층 그리드 파일의 디렉토리는 저장되는 객체의 분포나 서로 다른 속성간의 상관관계 등 데이터 특성에 큰 영향을 받지 않고 삽입되는 객체 수에 선형적으로 비례하여 증가하는[11] 특징을 가진다.

그림 1은 데이터 페이지의 용량이 3이라는 가정하에, 이차원 계층 그리드 파일의 성장 과정을 보이고 있다. 그림 1의 (a)는 전체 도메인 공간내에 세개의 레코드가 존재하는 초기 상태를 나타낸 것이다. 여기에 레코드를 하나 더 삽입하면 데이터 페이지의 용량이 초과되므로 전체 도메인 공간은 두개의 영역으로 분할되고, 새로운 데이터 페이지가 할당된다. 그리고, 레코드들은 분할 경계값을 따라 두 개의 데이터 페이지로 분산된다(그림 1(b)). 그림 1의 (c)와 (d)는 데이터 페이지 B와 C의 연속된 분할[1]에 따른 상태를 보여준다. 레코드들을 계속해서 삭제하면, 계층 그리드 파일의 데이터 페이지는 분할의 역순으로 병합하게 된다.

계층 그리드 파일의 디렉토리 엔트리는 영역벡터(region vector)와 다음 단계 페이지에 대한 포인터로 구성된다. N개의 속성을 갖는 계층 그리드 파일의 영역벡터는 N개의 해쉬값으로 구성되며, 해당 디렉

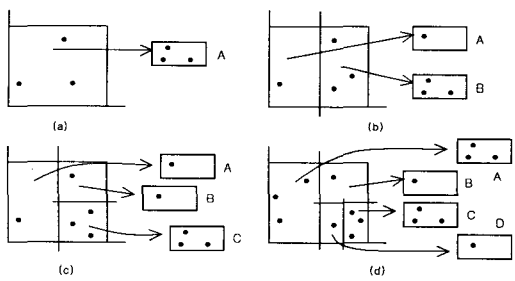


그림 1. 계층 그리드 파일의 동적 변화

1) 여기서는 두 축을 번갈아 가며 분할시키는 순환적 분할 전략을 가정하였다. 그러나 계층 그리드 파일에서는 페이지 영역을 분할할 때 분할 축을 임의로 선택할 수 있다. 따라서 분할 축을 선택하는 방법에 따라 속성들 사이의 클러스터링 정도를 달리 할 수 있다.

토리 엔트리가 나타내는 영역의 위치, 크기, 모양에 대한 정보를 갖는다. 영역벡터에서 i번째 해쉬값은 그 디렉토리 엔트리의 영역내에 속하는 모든 객체들의 *i*번째 속성을 해싱하였을 때 나타나는 해쉬값들의 공통 접두부(common prefix)가 된다. 또한 상위 단계의 한 영역은 그 영역에 대응하는 엔트리가 가리키는 부트리(subtree) 내의 모든 영역을 포함한다.

4. MOLAP를 위한 다차원 저장구조의 물리적 설계

본 절에서는 먼저 MOLAP에서 주어지는 집계 질의의 특징에 대하여 기술하고, 이러한 집계 질의들을 가장 효율적으로 처리할 수 있는 다차원 저장구조를 구성하기 위한 물리적 설계기법을 제시한다. 제 4.1절에서는 먼저 다차원 파일구조를 이용한 집계 연산처리 방법을 설명한다. 제 4.2절에서는 이차원 파일구조에 대해서 파일구조의 설계에 대한 기본원리를 기술한 다음에, 집계 연산처리를 위한 다차원 저장구조의 최적 조건과 함께 이 조건을 만족하는 이차원 파일구조의 영역 분할 전략을 제시한다. 그리고 제 4.3절에서는 일반적인 다차원 파일구조에 대해서 다차원 집계 연산을 최적으로 처리할 수 있는 다차원 저장구조의 물리적 설계기법을 기술한다.

4.1 다차원 파일구조를 이용한 집계 연산처리

먼저 집계 연산처리와 관련된 용어들을 정의하면 다음과 같다[18]. *집계 연산*은 데이터를 주어진 속성들의 값에 따라 여러 개의 그룹으로 나눈 후 주어진 집계 함수를 적용하여 각 그룹당 하나씩의 값을 구하는 연산이다[3]. 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 $\Pi_G P$라 표시한다. 그리고 페이지 영역 P가 임의의 영역 Q와 겹칠 때 간단히 *페이지 P와 영역 Q가 겹친다*고 한다. 또한, 페이지 그룹화 영역 $\Pi_G P$가 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 2는 집계 연산처리를 위한 다차원 파일구조를 나타낸다. 그림 2의 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 2에서 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 2에서 집계 윈도우 X[0, 49]×Z[0, 49]의 부분집계 페이지들은 C, D, E이다.

집계 연산처리의 가장 간단한 방법은 그룹화 도메인 공간을 하나의 집계 윈도우로 보고 집계 연산을 처리하는 방법으로, 먼저 주기억장치에 {그룹화 속성 값, 결과 값}의 엔트리로 구성되는 결과 테이블을 하나 준비한다. 그리고 파일을 한번 스캔하면서, 검색된 각 레코드에 대해 그룹화 속성들의 값을 키로

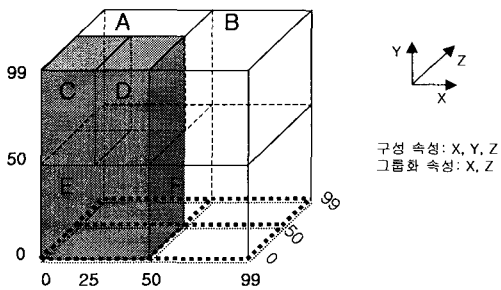


그림 2. 집계 연산처리를 위한 다차원 파일구조

하여 결과 테이블의 해당 엔트리를 찾아 집계 속성의 값을 엔트리의 결과 값에 집계한다. 이때, 해당 엔트리가 없으면 새로운 엔트리를 하나 삽입한다. 이 방법은 파일을 한번만 스캔하면서 집계 연산을 처리한다는 장점이 있다. 그러나 주 기억장치 크기의 한계로 인해 대용량 데이터베이스에는 유용하지 못하다.

주 기억장치 크기의 한계 문제를 해결하기 위해서, 그룹화 도메인 공간을 여러 개의 집계 윈도우로 나누어 집계 연산을 처리하는 방법을 사용한다. 즉, 그룹화 도메인 공간을 대상으로 하는 집계 연산을 집계 윈도우를 대상으로 하는 부분집계 연산들로 나누어 수행하는 것이다. 이것은 서로 다른 집계 윈도우에 속하는 레코드들은 그룹화 속성들의 값이 달라서 서로 다른 그룹에 속하므로, 하나의 부분집계 연산의 결과는 다른 부분집계 연산의 결과에 영향을 미치지 않기 때문에 가능하다. 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[18]. 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. 다음은 그림 2와 같은 다차원 파일구조의 경우 집계 연산처리 방법에 관한 예제이다.

우선, 그룹화 도메인 공간 X[0, 99]×Z[0, 99]를 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 집계 윈도우로 분할한다. 그리고 영역 질의를 사용하여 각 집계 윈도우에 대한 부분집계 연산을 수행한다. 예를 들어, 집계 윈도우 X[0, 49]×Z[0, 49]에 해당하는 부분집계 연산은 X[0, 49]×Z[0, 49], 그리고 Y[0, 99](속성 Y에 범위 조건이 없는 경우)로 구성된 영역 질의를 사용하여 수행된다. 즉, 그림 2에서 음영으로 표시된 막대에 속하는 레코드들을 액세스함으로써 X[0, 49]×Z[0, 49]에 해당하는 부분집계 연산이 처리된다.

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. 반면에, 관계형 데이터베이스에서 사용되는 테이블 저장구조는 다차원 클러스터링을 지원하지 않기 때문에, 집계 윈도우에 속하는 레코드들을 검색하는데 비효율적이다. 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 주어지기 때문에, 질의 영역들의 형태에 따라 집계 연산을 더욱더 효율적으로 처리할 수 있는 다차원 저장구조의 물리적 설계기법이 필요하다.

4.2 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 파일구조의 물리적 설계의 기본원리를 설명한다. 먼저, 이차원 도메인 공간 내에서 페이지 영역의 모양에 따라 주어진 질의 영역에 의해서 교차되는 평균 페이지 영역의 개수가 달라짐을 보인다. 데이터가 균일하게 분포한다고 할 때, 이차원 파일구조에서는 도메인 공간의 분할전략에 따라 일정한 크기의 페이지 영역들로서 페이지 영역의 모양이 다른 다양한 파일구조를 구성할 수 있다. 예를 들어, 그림 3은 두 개의 구성 속성 X와 Y에 대해서 256×16의 크기를 갖는 이차원 도메인 공간상에서 하나의 셀(cell)은 하나의 레코드를 나타내고, 한 데이터 페이지에 들어가는 레코드의 개수를 나타내는 블로킹 인수(blocking factor)가 16이라 할 때, 도메인 공간이 일정한 크기의 서로 다른 모양의 페이지 영역들로 분할된 상태를 보이고 있다. 즉, 그림 3의 (a), (b), 및 (c)는 각각 이차원 도메인 공간이 X축에 대한 Y축의 구간비가 각각 1/16, 16, 및 1인 페이지 영역들로 분할된 상태를 나타낸다.

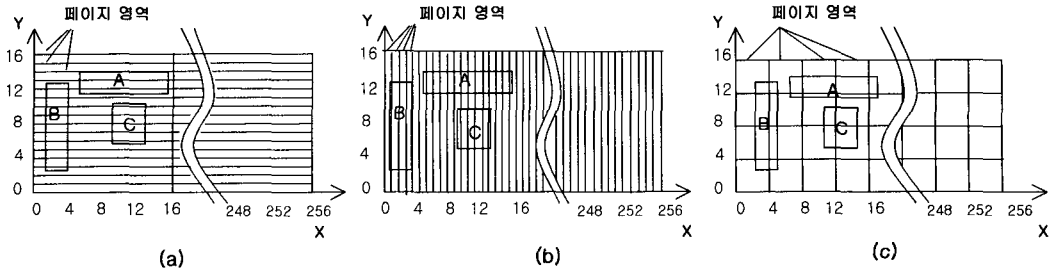


그림 3. 서로 다른 모양의 페이지 영역을 갖는 이차원 도메인 공간의 분할 상태

다차원 파일구조에서 질의처리의 성능은 도메인 공간의 분할 상태를 나타내는 페이지 영역의 구간비에 따라 달라진다. 예를 들어, 그림 3에서와 같이 서로 다른 페이지 영역의 구간비로 구성된 세 종류의 각 이차원 파일구조에 대해서, 크기는 모두 같고 모양이 서로 다른 세 종류의 질의 영역 A(구간비=1/4), B(구간비=4), 및 C(구간비=1)가 주어졌을 때, 이 질의 영역들과 교차하게 되는 페이지 영역들의 개수[2]는 다음과 같음을 알 수 있다. 즉, 질의 영역 A의 경우에는 그림 3의 (a)에서는 3개이고, (b)에서는 12개이며, (c)에서는 8개이다. 질의 영역 B의 경우에는 그림 3의 (a)에서는 12개이고, (b)에서는 3개이며, (c)에서는 8개이다. 그리고 영역 C의 경우에는 그림 3의 (a)에서는 6개이고, (b)에서는 6개이며, (c)에서는 4개이다.

이와 같이 영역 질의의 처리를 위하여 액세스해야 할 데이터 페이지의 개수는 주어진 질의 영역과 도메인 공간의 분할 상태를 나타내는 페이지 영역의 형태에 따라 많은 차이가 있으며, 주어진 질의 영역의 모양과 페이지 영역의 모양이 비슷할수록 액세스해야 할 페이지의 개수가 적어짐을 관찰할 수 있다. 따라서 본 논문에서는 이와 같은 원리를 이용하여 MOLAP의 집계 연산처리를 위한 영역 질의들의 정보를 이용하여 다차원 파일구조를 구성함으로써 집계 연산처리의 성능을 향상 시키고자 한다.

다차원 파일구조에서는 도메인 공간을 구성하는 페이지 영역의 모양에 따라 주어진 질의 영역에 의해서 교차되는 페이지 영역들의 평균 개수가 달라지는 특징이 있다. 참고문헌[7]에서는 이러한 특징을 이용하여 다차원 공간내의 데이터의 균일분포와 비균일분포 각각에 대하여 주어진 질의 영역들에 대해 페이지 영역의 평균 액세스 횟수를 최소로 하는 페이지 영역의 최적 구간비를 계산하는 방법을 제안하였다. 본 절에서는 이를 소개하고, 집계 연산처리를 위한 다차원 파일구조에 대한 페이지 영역의 최적 구간비를 이와 같은 방법으로 계산한다.

다음은 이차원 공간상에서 데이터가 균일하게 분포할 때 페이지 영역의 최적 구간비를 계산하는 방법에 대한 정리이다. 데이터가 균일하게 분포하면 도메인 공간을 구성하는 페이지 영역들의 크기가 일정하게 되며, 주어진 질의 영역들에 의해 교차되는 페이지 영역들의 개수를 최소로 하는 페이지 영역의 최적 구간비는 모든 질의 영역들에 대해 각 축별로 구간 크기를 더한 값의 비로서 계산할 수 있다[7]. 즉, 크기가 $p(x)\times p(y)$로 일정한 페이지 영역들로 나누어져 있는 이차원 도메인 공간상에서, 임의의 위치에 주어지는 n개의 질의 영역 $q_i(x)\times q_i(y)$ $(i=1,\cdots,n)$에 대해 각 질의 영역과 교차하게 되는 페이지 영역의 총 개수를 최소로 하는 최적 페이지 영역의 구간비 (p(x) : p(y))는 $\sum_{i=1}^{n} q_i(x) : \sum_{i=1}^{n} q_i(y)$이다.

다차원 공간상에서 데이터가 비균일하게 분포한다는 것은 도메인 공간내의 위치에 따라 데이터 밀집도가 다름으로 인하여 페이지 영역의 크기가 위치에 따라 달라짐을 의미한다. 즉, 밀집도가 높은 곳은 밀집도가 낮은 곳에 비하여 많은 페이지가 할당되므로 각 페이지 영역의 크기는 작아지게 된다. 따라서 비균일 분포의 경우에는 질의 영역에 의해 교차되는 페이지 영역의 개수가 질의 영역의 크기뿐만 아니라 질의 영역이 주어진 위치의 데이터 밀집도에도 비례하게 되므로, 균일분포에서와 같이 페이지 영역의 최적 구간비를 모든 질의 영역의 각 축별로 구간 크기를 단순히 더한 값의 비율로서 구할 수 없다.

이와 같은 경우에는 각 질의 영역의 크기에 대해 위치에 따른 데이터 밀집도를 가중치(weight)로 곱

2) 이들은 질의를 처리하기 위하여 액세스하게 되는 데이터 페이지들의 개수와 동일하다.

한 질의 영역의 형태를 *정규화된 질의 영역(normalized query region)*이라 하고, 이러한 질의 영역의 정규화를 통하여 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 하면 된다.

계층 그리드 파일에서는 새로운 객체의 삽입으로 페이지의 용량이 초과되면, 이 페이지에 대응하는 페이지 영역은 같은 크기를 갖는 두개의 영역으로 분할된다. 이때 분할되는 페이지 영역의 분할 축으로서 분할된 영역의 구간비가 최적 구간비에 가깝게 되는 축을 선택함으로써, 객체의 지속적인 삽입으로 인한 연속된 분할시에 도메인 공간내의 모든 페이지 영역의 구간비를 최적 구간비에 근접하도록 유도할 수 있다.

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 : 아래 그림 4는 이차원 도메인 공간에서 $q_x\times q_y$ 형태의 질의 영역 Q가 임의의 위치에 주어질 때, 크기가 $B(=p(x)\times p(y))$인 특정 페이지 영역 P와 교차하게 되는 위치의 범위를 질의 영역 Q의 좌상점이 위치할 수 있는 영역(음영 부분) LQ로 나타낸 것이다.

그림 4에서 LQ의 크기 $SIZE_LQ(p(x),p(y))$은 다음 식과 같다.

$$SIZE\_LQ(p(x),\ p(y))=(p(x)+q_x)(p(y)+q_y) \qquad (1)$$

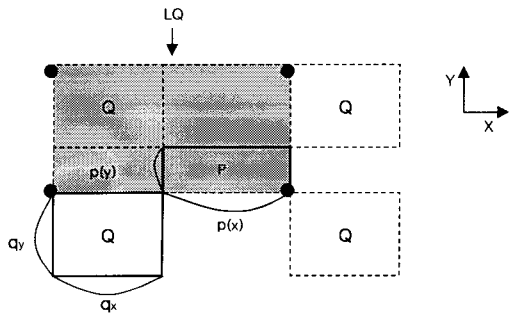


그림 4. 임의의 한 페이지 영역과 교차하게 되는 질의 영역의 위치 영역

$p(x)\times p(y)=B$에 의해서, 수식(1)의 $p(y)$를 $\frac{B}{p(x)}$로 치환하면,

$$SIZE\_LQ(p(x),\frac{B}{p(x)}) = (p(x)+q_x)(\frac{B}{p(x)}+q_y)$$
$$=B+\frac{q_xB}{p(x)}+p(x)q_y+q(x)q_y \qquad (2)$$

따라서 수식(2)의 값을 최소로 하는 $p(x)$를 구하면, $p(x)=\sqrt{(q_x/q_y)B}$이다. 또한, 이러한 $p(x)$에 대한 $p(y)$는 $p(x)\times p(y)=B$에 의하여 $p(y)=\sqrt{(q_y/q_x)B}$이다. 그러므로, $SIZE_LQ(p(x),\ p(y))$를 최소로 하는 페이지 영역 P의 구간비 $p(x):p(y)=q_x:q_y$이다.

정리 1을 이용하면, 페이지 영역의 분할 시 분할된 페이지 영역의 구간비가 최적 구간비에 가깝게 되는 분할 축을 선택할 수 있다. 그림 5는 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 5에서 (a)는 분할 축으로 X축을 선택한 경우의 LQ_x를 나타내고, (b)는 분할 축으로 Y축을 선택한 경우의 LQ_y를 나타낸다.

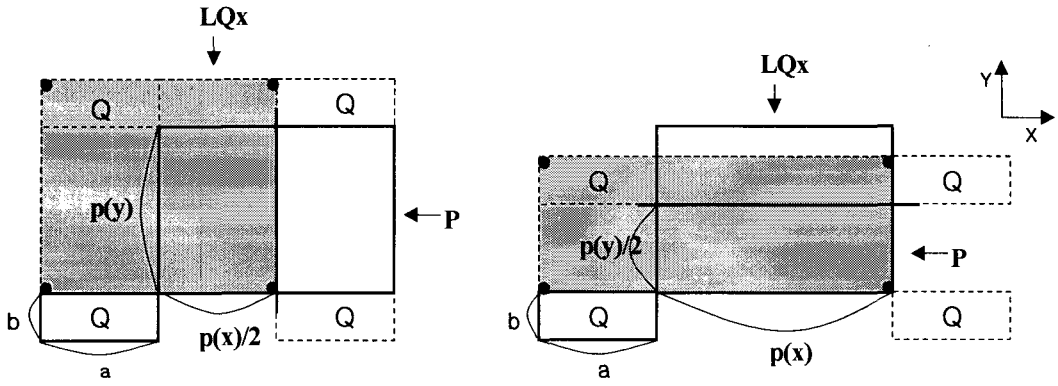
그림 5에서 X축을 분할한 경우의 LQ_x의 크기는

$$SIZE(LQ_x) = (p(x)/2+a)(p(y)+b) \qquad (3)$$

이고, Y축을 분할한 경우의 LQ_y의 크기는

$$SIZE(LQ_y) = (p(x)+a)(p(y)/2+b) \qquad (4)$$

이다.



(a) X축 분할 (b) Y축 분할

그림 5. 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. 즉, $(p(x)/2+a)(p(y)+b) < (p(x)+a)(p(y)/2+b)$이면 X축을 분할하고, 그렇지 않으면 Y축을 분할함으로써 이차원 파일구조를 구성하는 페이지 영역들의 모양이 최적의 구간비가 되도록 한다.

4.3 MOLAP 저장구조의 물리적 설계기법

아래 그림 6은 제 4.2절에서의 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 알고리즘을 나타낸다.

그림 6에서 나타난 MOLAP 시스템을 위한 다차원 파일구조의 전체 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 $q(1)\times q(2) \cdots q(j) \cdots \times q(N)$에 대한 정규화는 다음과 같다. 즉, 먼저 질의 결과에 의한 질의 영역내의 레코드 개수 nr을 이용하여 레코드 밀집도 d를 $\frac{nr}{q(1)q(2)\cdots q(i)\cdots q(N)}$로 구하여, 질의 영역을 이루는 각 축의 구간 $q(j)$에 가중치 $d^{1/N}$을 곱하여 정규화된 질의 영역의 형태 $q(1)d^{1/N}\times q(2)d^{1/N}\cdots\ q(j)d^{1/N}\cdots\times q(N)d^{1/N}$를 얻는다.

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

셋째, 최적 구간비에 가장 가까운 페이지 영역들로 구성된 다차원 파일구조를 구축한다. 여기서는 제 4.2절의 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 4.2절의 영역 분할정책을 N차원으로 확장하여 적용하는 것이다. 즉, 둘째 단계에서 결정된 최적 구간비(a(1):a(2)$\cdots$:a(j)$\cdots$:a(N))와 같은 모양을 가지는 가상의 질의 영역(a(1)$\times$a(2) $\cdots$ a(j) $\cdots\times$a(N))이 임의의 위치에 주어진다고 가정하고, 분할이 요구되는 페이지 영역(p(1)$\times$p(2) $\cdots$ p(j) $\cdots\times$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)) $\cdots$ (p(i)/2+a(i)) $\cdots$ (p(N)+a(N)이다.)를 각각 계산한 다음 그 값이 가장 작게 되는 축을 분할 축으로 선택한다.

• **설계정보:**

MOLAP 시스템에서 주어지는 영역-집계 질의들의 처리에 필요한 n개의 질의 영역에 대하여,

(1) 각 질의 영역의 형태 $q_i(1)\times q_i(2)\cdots q_i(j)\cdots\times q_i(N)(i=1,\cdots,n)$

(2) 각 질의 영역에 포함되는 레코드의 개수nr_i $(i=1,\cdots,n)$

• **알고리즘:**

단계 1: 각 질의 영역의 정규화 $(i=1,\cdots,n)$

(1) 각 질의 영역의 레코드 밀집도 d_i를 구한다.

$$d_i=\frac{nr_i}{q(1)q(2)\cdots q(j)\cdots q(N)}$$

(2) 밀집도 d_i로서 정규화된 질의 영역의 각 축의 구간 크기를 구한다.

$$\begin{aligned} q_i'(1) &= q_i(1)\times d_i^{1/N} \\ q_i'(2) &= q_i(2)\times d_i^{1/N} \\ &\vdots \\ q_i'(j) &= q_i(j)\times d_i^{1/N} \\ &\vdots \\ q_i'(N) &= q_i(N)\times d_i^{1/N} \end{aligned}$$

단계 2: 최적 페이지 영역의 최적 구간비$(a(1):a(2)\cdots:a(j)\cdots:a(N))$ 결정

$$a(1):a(2)\cdots:a(j)\cdots:a(N) = \sum_{i=1}^{n} q_i'(1) : \sum_{i=1}^{n} q_i'(2) \cdots : \sum_{i=1}^{n} q_i'(j) \cdots : \sum_{i=1}^{n} q_i'(N)$$

단계 3: 최적 구간비에 가장 가까운 페이지 영역들로 구성된 다차원 파일구조의 구축

(1) 다차원 저장구조에 레코드 삽입

(2) 데이터 페이지에 오버플로우가 발생하면, 데이터 페이지 분할

⇒ 대응하는 페이지 영역 $(p(1)\times p(2)\cdots p(j)\cdots\times p(N))$의 분할전략:

다음 식들의 값들 중 최소가 되는 차례의 축을 분할 축으로 선택

- 첫 번째: $(p(1)/2+a(1))(p(2)+a(2)\cdots(p(j)+a(j))\cdots(p(N)+a(N))$
- 두 번째: $(p(1)+a(1))(p(2)/2+a(2)\cdots(p(j)+a(j))\cdots(p(N)+a(N))$

⋮

- j 번째: $(p(1)+a(1))(p(2)+a(2)\cdots(p(j)/2+a(j))\cdots(p(N)+a(N))$

⋮

- N 번째: $(p(1)+a(1))(p(2)+a(2)\cdots(p(j)+a(j))\cdots(p(N)/2+a(N))$

그림 6. MOLAP 시스템을 위한 다차원 파일구조의 물리적 설계 알고리즘

5. 성능 평가

본 절에서는 MOLAP 저장구조의 물리적 설계기법의 유용성을 다양한 실험을 통하여 제시한다. 실험의 목적은 MOLAP 저장구조를 구성하는 페이지 영역의 모양에 대한 MOLAP 집계 연산처리에 필요한 다양한 영역 질의에 대한 질의 영역들의 모양과 크기, 그리고 파일에 저장된 레코드들의 분포 등 여러 가지 인자들의 변화에 대하여 제안된 설계기법의 유용성을 실제 실험을 통하여 검증하는 것이다. 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 두 종류의 다차원 저장구조를 구축하였다. 하나는 X축과 Y축으로 구성된 이차원의 저장구조이고, 다른 하나는 X축, Y축, 및 Z축으

로 구성된 삼차원의 저장구조이다.

이차원 저장구조의 구축에 사용한 레코드의 분포 특성은 균일 분포와 비균일 분포로 구분한다. 균일 분포의 레코드들은 각 축의 값이 $[-2^{31}, 2^{31}-1]$인 도메인 내에서 균일 분포하게 하고, 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 패턴의 구성을 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 대영역(Large)인 L1, L2, L4, L8, L16, L32, L64, L128, L256, L512, 및 L1042형태의 질의 영역, (2) 크기가 도메인 공간의 1/2000로서 중영역(Medium)인 M1, M2, M4, M8, M16, M32, M64, M128, M256, M512, 및 M1024형태의 질의 영역, (3) 크기가 도메인 공간의 1/20000로서 소영역(Small)인 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 이차원 MOLAP 저장구조들을 생성하고, 각각에 대하여 다양한 형태의 질의 영역들을 갖는 집계 연산들을 처리할 때 발생하는 평균 페이지 액세스 수를 측정한다. 실험에 사용된 질의 영역의 형태는 대영역인 L1, L2, L4, 중영역인 M8, M16, M32, 소영역인 S64, S128, S256 등이며, 집계 연산의 질의 패턴을 구성하기 위하여 대영역 질의 형태는 10개씩 도메인 공간상의 중앙에 집중되도록 생성하고, 중영역 질의 형태는 100개씩 도메인 공간상의 좌측하단에 집중되도록 생성하며, 대영역 질의 형태는 1000개씩 도메인 공간상의 우측상단에 집중되도록 생성한다. 그림 7은 실험 결과를 그래프로 나타낸 것이다. 그림 7에서 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 7에서 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. 따라서 MOLAP 저장구조로 다차원 파일구조를 이용하는 경우, 이와 같은 최적 구간비를 가지는 페이지 영역들로 구성함으로써, MOLAP 시스템의 집계 연산처리의 성능을 더욱더 향상 시킬 수 있다.

두 번째 실험에서는 이차원 비균일 분포의 데이터에 대하여 첫 번째 실험과 동일한 실험을 수행하였다. 즉, 이차원 비균일 분포의 데이터로서 서로 다른 구간비의 페이지 영역을 갖는 여러 개의 이차원 MOLAP 저장구조들을 생성하고, 각각에 대하여 첫 번째 실험에 주어진 질의 패턴의 영역 질의들을 처리할 때 발생하는 평균 페이지 액세스 수를 측정하였다. 본 실

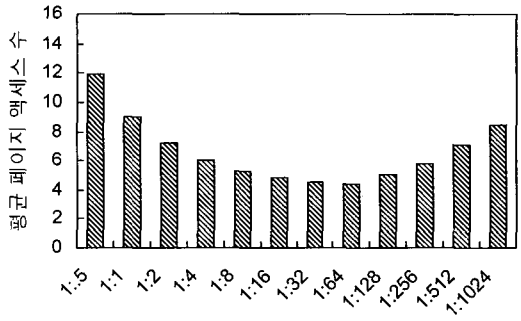


그림 7. 이차원 균일분포 데이터에 대한 서로 다른 구간비의 페이지 영역을 갖는 이차원 파일구조별 영역 질의처리 성능

험에서는, 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. 이와 같은 실험 결과는 데이터가 비균일하게 분포하면, 주어진 다양한 형태의 질의 영역들에 의해 교차되는 페이지 영역의 개수를 최소로 하는 페이지 영역의 최적 구간비는 정규화 과정을 통하여 주어진 질의 영역들에 대해 각 축별로 구간 크기를 더한 값의 비로서 계산할 수 있음을 보이는 것이다. 또한, 이는 같은 질의패턴에 대해서 첫 번째 실험에서와 같은 균일 분포의 데이터로 구성된 이차원 MOLAP 저장구조의 최적 구간비인 1:57과는 매우 다른 구간비의 페이지 영역이 최적이 됨을 알 수 있다.

세 번째 실험에서는 삼차원 비균일 분포의 데이터에 대하여 두 번째 실험과 동일한 실험을 수행한다. 즉, 삼차원 비균일 분포의 데이터로서 서로 다른 구간비의 페이지 영역을 갖는 여러 개의 삼차원 MOLAP 저장구조들을 생성하고, 각각에 대하여 다양한 형태의 질의 영역들을 갖는 집계 연산들을 처리할 때 발생하는 평균 페이지 액세스 수를 측정한다. 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 한다. 그림 8은 실험 결과를 그래프 형태로 나타낸 것이다. 정규화된 모든 질의 영역들에 대하여 각 축의 구간 크기를 더한 값의 비는 1 : 6 : 68로 계산되었으며, 그

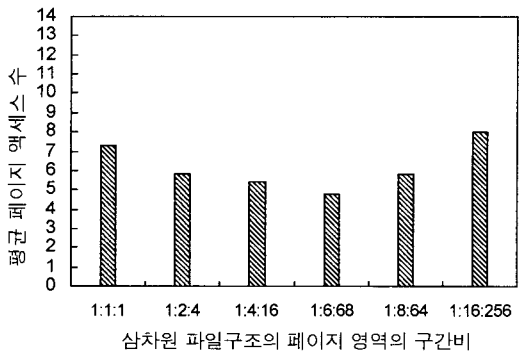


그림 8. 삼차원 비균일 분포 데이터에 대한 서로 다른 구간비의 페이지 영역을 갖는 삼차원 파일구조별 영역 질의 처리 성능

림 8에서 알 수 있는 바와 같이 이 비율과 같은 구간비의 페이지 영역을 갖는 MOLAP 저장구조에서 가장 좋은 성능을 보인다. 이와 같은 실험결과는 삼차원의 MOLAP 저장구조에 대해서도 두 번째 실험의 이차원 MOLAP 저장구조에서와 마찬가지로, 다양한 형태의 질의 영역들에 의해 교차되는 페이지 영역의 개수를 최소로 하는 페이지 영역의 최적 구간비는 정규화 과정을 통하여 주어진 질의 영역들에 대해 각 축별로 구간 크기를 더한 값의 비로서 계산할 수 있음을 보이기 위한 것이다. 또한, 이 실험의 결과는 제 4.3절에서 제시한 MOLAP 저장구조의 물리적 설계 알고리즘의 실용성을 입증하는 것이다.

네 번째 실험에서는 삼차원 비균일 분포를 가지는 데이터에 대하여 서로 다른 구간비의 페이지 영역을 갖는 여러 개의 삼차원 MOLAP 저장구조들을 생성하고, 각각에 대하여 고유의 질의 영역 형태를 갖는 집계연산들을 처리할 때 발생하는 평균 페이지 액세스 수를 측정한다. 이 실험의 목적은 삼차원 비균일 분포 데이터에 대하여 도메인 공간상의 임의의 위치에 주어지는 특정 형태의 질의 영역에 의해 액세스되는 평균 페이지의 개수는 삼차원 파일구조의 페이지 영역의 구간비가 주어진 질의 영역의 구간비와 같게 될 때 최소로 됨을 확인하는 것이다. 삼차원 MOLAP 저장구조들의 구축에 사용된 페이지 영역의 구간비는 1:1:1, 1:2:4, 1:4:16, 1:8:64, 및 1:16:256 등 다섯 가지이며, 질의 영역의 형태는 S1_1_1, S1_2_4, S1_4_16, S1_8_64, 및 S1_16_256 등의 다섯 가지이다. 이러한 각 질의 영역의 형태별로 1000개의 질의 영역을 도메인 공간상에 균일하게 생성하고, 이들 질의를 처리하는데 발생하는 평균 페이지 액세스 수를 측정한다. 그림 9는 이에 대한 실험 결과를 그래프로 나타낸 것이다. 모든 형태의 질의 영역에 대하여, 그 질의 영역의 구간비를 페이지 영역의 구간비로 가지는 MOLAP 저장구조에서 가장 좋은 성능을 보였다. 이와 같은 실험 결과는 삼차원 MOLAP 저장구조에 대해서도 본 논문에서 제안한 MOLAP 저장구조의 물리적 설계기법이 유효함을 보이는 것이다.

마지막으로, 다섯 번째 실험에서는 이차원 MOLAP 저장구조에 대하여 물리적 설계기법을 이용하여 구성한 이차원 파일구조가 기존의 순환 분할전략(즉, 구간비 = 1:1)으로 구성한 이차원 파일구조와 비교하여 얼마나 성능개선 효과가 있는 지를 알아본다. 먼저, 여섯 가지의 이차원 질의 영역의 형태인 M1, M4,

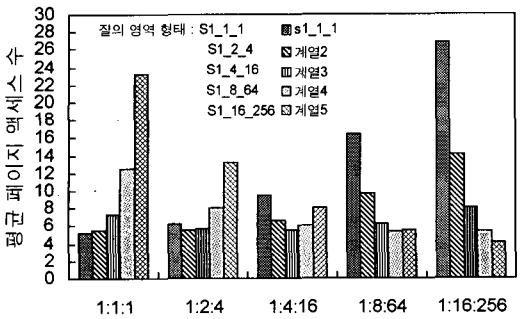


그림 9. 삼차원 비균일 분포의 데이터에 대한 서로 다른 구간비의 페이지 영역을 갖는 삼차원 파일구조에 대한 질의 영역의 형태별 질의처리 성능

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 10은 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.

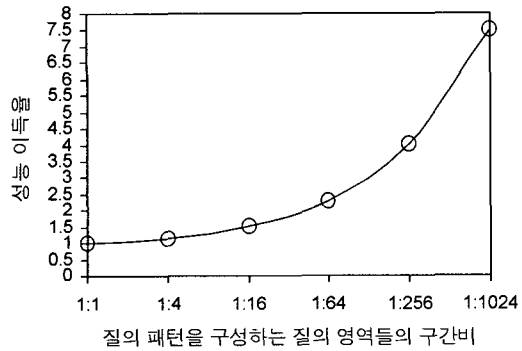


그림 10. 집계 연산을 구성하는 질의 영역의 구간비별 MOLAP 저장구조의 물리적 설계기법에 의해서 생성된 이차원 파일구조의 성능 효율.

그림 10에서 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. 이러한 결과는 제 4절에서 제시한 MOLAP를 위한 다차원 저장구조의 물리적 설계기법의 성능개선 효과를 잘 나타내는 것이다.

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. 먼저, 다차원 파일구조를 이용한 집계 연산처리는 다차원 도메인 공간상에서 여러 개의 영역 질의로 처리됨을 보이고, 이러한 영역 질의들을 처리하기 위하여 액세스하는 데이터 페이지의 개수를 최소로 하는 최적의 다차원 파일구조를 구축하였다.

본 논문에서 제안한 MOLAP 저장구조의 물리적 설계기법은 다차원 파일구조를 구성하는 도메인 공간의 분할 상태를 나타내는 페이지 영역들의 모양이 도메인 공간상의 영역 질의가 위치하는 질의 영역의 모양과 일치할 때, 질의처리 시에 액세스되는 데이터 페이지의 수가 최소로 되는 원리를 바탕으로 한다. 제안한 설계기법에서는 먼저, 도메인 공간상의 데이터 분포 특성을 고려하기 위하여 주어진 각 질의 영역에 대하여 레코드 밀집도를 가중치로 부여하여 정규화 과정을 거친다. 그리고 정규화된 모든 질의 영역을 대상으로 각 축별로 구간의 크기를 합산한 값의 비율로서 페이지 영역의 최적 구간비를 결정하고, 이 최적 구간비에 최대한 가까운 모양의 페이지 영역들로 구성된 다차원 파일구조를 구축한다.

또한, 본 논문에서는 이와 같은 MOLAP 저장구조

의 물리적 설계기법의 성능평가를 위하여, 다차원 파일구조의 하나인 계층그리드 파일을 대상으로 페이지 영역의 모양이 최적 구간비에 근접하도록 하는 영역 분할전략을 제시하고, 이를 이용하여 다양한 실험을 수행하였다. 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 10에서와 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.

참 고 문 헌

[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] G. Graefe, "Query Evaluation Techniques for Large Databases," *ACM Computing Surveys*, Vol. 25, No. 2, pp. 73-170, June 1993.

[4] I. S. Mumick, D. Quass, and B. S. Mumick, "Maintenance of Data Cubes and Summary Tables in a Warehouse," In *Proc. Int'l Conf. on Management of Data*, pp. 100-111, ACM SIGMOD, Tucson, Arizona, June 1997.

[5] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tabs, and Subtotals," In *Proc. Int'l Conf. on Data Engineering*, pp. 152-159, IEEE, New Orleans, Louisiana, Feb. 1996.

[6] J. L. Bentley, "Multidimensional Binary Search Trees in Database Applications," *IEEE Trans. on Software Eng.*, Vol. 5, No. 4, pp. 333-340, July 1979.

[7] 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.

[8] 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.

[9] J. Nievergelt and H. Hinterberger, "The Grid File: An Adaptable Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol. 9, No. 1, pp. 38-71, Mar. 1984.

[10] J. T. Robinson, "The K-D-B Tree: A Search Structure for Large Multidimensional Dynamic Indexes," In *Proc. int'l Conf. on Management of Data*, ACM SIGMOD, pp. 10-18, 1981.

[11] K. Whang and R. Krishnamurthy, *Multilevel Grid File*, IBM Research Report RC 11516, 1985.

[12] L. Harada et al., "Query Processing Method for Multi-Attribute Clustered Relations," In *Proc. Int'l Conf. on Very Large Data Bases*, pp. 59-70, Brisbane, Australia, Aug. 1990.

[13] 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.

[14] 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.

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

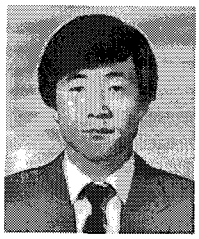
[16] S. Geffner et. al., "Relative Prefix Sums: An Efficient Approach for Querying Dynamic OLAP Data Cubes," In *Proc. Int'l Conf. on*

Data Engineering, pp. 328-335, IEEE, Sydney, Australia, Mar. 1999.

[17] Y. Kotidis and N. Roussopoulos, "An Alternative Storage Prganization for ROLAP Aggregate Views Based on Cubetrees," In *Proc. Int'l Conf. on Management of Data*, pp. 249-258, ACM SIGMOD, Seattle, Washington, June 1998.

[18] 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.

[19] Y. Zhao, P. M. Deshpande, and J. F. Naughton, "An Array-Based Algorithm for Simultaneous Multidimensional Aggregates," In *Proc. Int'l Conf. on Mangement of Data*, pp. 159-170, ACM SIGMOD, Tucson, Arizona, June 1997.



이 종 학

1982년 경북대학교 전자공학과(전자계산 전공) 졸업(학사)
1984년 한국과학기술원 전산학과 졸업(공학석사)
1997년 한국과학기술원 전산학과 졸업(공학박사)
1991년 정보처리기술사
1984년~1987년 금성통신(주) 부설연구소 주임연구원
1987년~1998년 한국통신 연구개발본부 선임연구원
1998년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수
관심분야 : 객체 데이타베이스, 다차원 파일구조, 물리적 데이터베이스 설계, 데이터 웨어하우스, 생물정보학 등