

다차원 동적 화일 구조를 아용한 데이터 분포의 추정

김 상 옥

Estimation of Data Distribution Using a Multidimensional Dynamic File Organization

Sang-Wook Kim

ABSTRACT

This paper presents a technique for estimating distribution of data stored in a database. This technique is very useful for accurate selectivity estimation, which is essential in query optimization and physical database design. To maintain data distribution, we employ the directory of the multilevel grid file, a multidimensional dynamic file organization. The major advantage of our technique is that data distribution information is maintained dynamically in the multilevel grid file. In contrast, other static methods such as the histogram method use static data structures, which requires periodic restructuring. Furthermore, we propose a method for keeping the abstract information of data distribution in main memory. This is advantageous in the situation where the size of main memory is not sufficient. Finally, We also suggest formulas for calculating selectivities of various queries based on our data distribution information.

1. 서 론

다차원 개념이 도입되는 CAD/CAM, VLSI, 지리 정보 처리 시스템 등의 보다 진보적인 응용 분야에서는 “X값이 20이고, Y값이 100인 점을 검색하라”와 같은 여러 개의 애트리뷰트로 구성된 질의가 빈번하-

게 사용된다. 다차원 동적 파일 구조(multidimensional file organization)는 이러한 다중 애트리뷰트 질의를 효과적으로 지원하기 위한 일종의 색인 구조이다. 현재까지 제안된 대표적인 다차원 동적 화일 구조로는 인터플레이션을 기반으로 하는 색인[Bur83], 다차원 선형 해형[Ouk83], 다차원 신장 해형[Oto84], K-D 트리[Ben79], K-D-B 트리[Rob81], BANG 화일[Fre87], 그리드 화일[Nie84], 그리고 계층 그리드

강원대학교 정보통신공학과 전임강사

화일[Wha91]등이 있다. 본 논문에서는 다차원 동적 화일 구조를 이용하여 시스템에 저장된 데이터의 분포를 효과적으로 추정하는 기법을 제안하고자 한다.

데이터베이스 시스템내에 저장된 데이터의 분포가 사용되는 가장 대표적인 경우는 선택률 추정이다. 선택률(selectivity)이란, "저장된 전체 레코드 수에 대한 질의조건을 만족하는 레코드 수의 비"로 정의되며, 여기에 전체 레코드 수를 곱하면 질의조건을 만족하는 레코드의 수를 알아낼 수 있다. 질의 최적화 과정에서는 선택률을 이용하여 질의를 만족하는 레코드의 수를 추정하며, 추정된 결과를 가지고 여러가지 처리 방식에 대한 각각의 응답시간을 예측한다[Bla77][Sel79]. 이중 가장 응답시간(response time)이 짧다고 추정되는 방법을 선택하는데 이를 액세스 경로 선택(access path selection)이라 하며, 올바른 액세스 경로의 선택 여부는 질의 처리의 효율성에 큰 영향을 미친다[Sel79]. 그러므로 선택률의 정확한 추정은 전체 시스템의 성능에 큰 영향을 미친다.

현재까지 널리 쓰이고 있는 가장 간단한 선택률 추정기법은 System R [Sel79]에서와 같이 시스템에 저장된 레코드들이 최대값과 최소값 사이에서 균일(uniform)하게 분포한다는 가정하에 질의를 만족하는 구간의 크기를 전체 구간의 크기로 나누는 방법을 이용한다.

이러한 균일분포 가정방식(uniform distribution assumption)의 장점은, 선택률을 추정할 때 사용되는 최대값, 최소값 등의 인자들을 관리하기가 쉽고 또한 계산이 단순하므로, 적은 오버헤드를 가지고 선택률을 쉽게 추정할 수 있다는 것이다. 그러나 저장된 데이터의 분포를 전혀 반영할 수 없으므로 저장된 레코드들이 균일분포 이외의 분포를 취하는 경우에는 추정된 선택률과 실제 선택률의 오차가 커진다는 단

점을 가지고 있다.

실제 데이터의 분포를 반영할 수 없는 균일분포 가정방식의 문제점을 해결하는 근본적인 방법은 시스템에 저장된 데이터의 실제 분포를 추정하는 것이다. 현재, 이를 위한 몇몇 방법이 제안되어 있으나[Pia84][Man88][Mur88][Che90], 이들이 사용하는 데이터 분포 정보는 어느 한 순간에 구성된 정적인 정보이므로, 레코드의 삽입과 삭제가 빈번히 발생하는 동적인 상황에서는 데이터의 분포정보를 주기적으로 재구성해야 한다는 문제점이 있다[Che90][Man88][Mur88].

본 논문에서는 다차원 동적 화일 구조의 하나인 계층 그리드 화일[Wha91]을 이용함으로써 레코드가 삽입, 삭제되는 상황에서도 데이터의 분포 정보를 동적으로 유지할 수 있는 새로운 기법을 제시하고, 이를 이용하여 선택률을 추정 공식을 유도한다. 제안된 기법의 가장 큰 장점은 데이터의 분포를 동적으로 관리함으로써 기존의 기법들이 갖는 재구성의 오버헤드를 피할 수 있다는 것이다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구로서 히스토그램 기법의 특성과 그 장단점에 대하여 설명한다. 제 3장에서는 다차원 동적 해쉬 화일구조의 하나인 계층 그리드 화일의 특성에 대하여 설명한다. 제 4장에서는 계층 그리드 화일의 각 단계 디렉토리에서 데이터 분포 정보를 추출하는 방법에 대하여 기술하고, 이를 이용한 선택률 추정 공식을 제시한다. 제 5장에서는 결론을 내린다.

2. 관련 연구

본 장에서는 저장된 데이터의 분포를 추정하고자 하는 연구[Pia84][Mur88][Che90]로서 히스토그램 기법(histogram method)

이라고 불리우는 equal-width, equal-depth, variable-width의 세가지 기법에 대하여 간단히 소개하고자 한다.

Equal-width 기법[Pia84]에서는 모든 구간의 크기를 같도록 만든다. 따라서 전체범위를 몇 개의 구간으로 나눌 것인가에 따라 각 구간의 크기가 결정되며, 각 구간의 경계값과 그 구간내에 속하는 레코드의 수가 기록된다. 구간의 크기가 모두 같으므로 각 구간내에 속하는 레코드의 수를 가지고 데이터의 분포를 추정할 수 있으며, 이 데이터 분포 정보를 이용하여 선택률도 추정할 수 있다.

이 방법으로 추정되는 데이터 분포의 정확도는 분할되는 구간의 크기에 영향을 받게 되는데, 이 구간의 크기는 전체범위와 저장되는 레코드의 수 및 데이터의 분포에 따라 적절히 선정되어야 한다. 그러나 전체범위의 크기나 데이터의 분포는 상황에 따라 변동되는 것이므로 미리 예측할 수 없고, 한번 결정된 구간의 크기는 고정되므로 환경의 동적인 변화에 따라 선택률 추정값의 오차가 커질 수 있다.

Equal-depth 기법[Pia84]에서는 각 구간내에 속하는 레코드의 수를 같도록 만든다. 즉, 하나의 구간이 미리 결정된 수 만큼의 레코드들을 갖도록 구간의 크기를 할당시키므로 각 구간의 크기는 서로 다르다. 데이터 분포 정보내에는 구간내에 속하는 레코드의 수, 각 구간의 경계값 및 각 구간의 크기 등이 기록된다. 같은 수의 레코드들이 크기가 다른 구간내에 속하게 되므로, 각 구간의 상대적인 크기로써 구간내의 밀도를 알아낼 수 있으며, 또한 이를 기반으로 데이터의 분포를 추정할 수 있다.

Equal-depth 기법에서는 데이터 분포 정보를 구성할 때, 정렬(sorting)을 사용하여 구간마다 같은 수의 레코드들을 할당해 주기 때문에 데이터 분포 정보를 구성한 후에

발생되는 레코드의 삽입, 삭제, 갱신등의 변동은 각 구간이 같은 수의 레코드를 포함한다는 기본 원칙을 무너뜨린다. 따라서 보다 정확한 선택률의 추정을 위해서 주기적으로 데이터 분포 정보를 재구성 해야 하는 오버헤드가 발생한다.

끝으로 소개될 variable-width 기법[Che90]에서는 구간의 크기와 구간내에서의 레코드 수가 구간마다 다를 수 있으나, 하나의 구간내에서는 레코드들이 균일하게 분포되도록 구간경계를 결정한다. 즉, 각 구간이 균일한 분포(uniform distribution)를 가질 때까지 전체 데이터 공간을 반복 분할하여 데이터 분포 정보를 구성한다. 데이터 분포 정보는 각 구간의 경계값과 그 구간내에서의 레코드의 조밀도가 되며, 이를 이용하여 데이터의 분포를 추정해 낼 수 있다.

그러나 이 방법 또한 한번 구성된 데이터 분포 정보는 이후에 발생하는 레코드의 삽입, 삭제 상황을 적절히 반영할 수 없으므로 equal-depth 기법의 경우와 마찬가지로 주기적인 재구성이 필요하다는 단점이 있다.

3. 계층 그리드 파일

본 논문에서 제안하는 기법의 핵심 아이디어는 주기적으로 데이터 분포 정보를 재구성해야 하는 히스토그램 기법과는 달리 레코드의 삽입, 삭제, 및 갱신에 따라 데이터 분포 정보를 동적으로 유지시킨다는 것이다. 본 장에서는 이를 위한 기반 자료 구조로서 사용되는 계층 그리드 화일[Wha91]에 대하여 소개하고자 한다. 제 3.1절에서는 계층 그리드 화일의 동적 특성에 대하여 설명하고, 제 3.2절에서는 계층 그리드 화일의 구조적 특성에 대하여 기술한다.

3.1 계층 그리드 화일의 동적 특성

계층 그리드 화일은 데이터 페이지(data page)와 디렉토리(directory)로 구성된다. 하나의 영역에 하나씩 할당되는 데이터 페이지는 레코드들을 저장하는 저장단위가 되며, 디스크 상에 존재한다. 하나의 데이터 페이지내에는 그 데이터 페이지가 할당된 영역내에 속하는 레코드들만이 저장될 수 있다. 디렉토리는 여러 영역으로 분할된 데이터 공간의 상태를 반영하며, 각각의 영역에 어느 페이지가 할당되어 있는가에 대한 정보를 갖는다.

계층 그리드 화일은 레코드가 데이터 공간에 삽입되고 삭제되는 상황에 대하여 분할(splitting)과 병합(merging)을 반복함으로써 동적 변화에 적응한다. 레코드가 삽입되는 경우, 레코드가 갖는 N개의 키값을 해싱하여 그 레코드가 속하는 영역을 찾게 되고, 그 영역에 할당된 데이터 페이지에 레코드를 삽입하게 된다. 이 결과로 데이터 페이지가 허용 용량을 초과하게 되면(overflow), 해당 영역은 같은 크기를 갖는 새로운 두 영역으로 분할되고 새로운 데이터 페이지가 하나 더 할당되어, 기존의 데이터 페이지에 있던 레코드들은 영역의 분할 경계값을 기준으로 두 데이터 페이지에 분산된다.

레코드를 삭제할 때에는 삭제될 레코드가 갖는 N개의 키값을 해싱하여 해당 데이터 페이지를 찾고, 데이터 페이지내에 해당 레코드가 존재하면 그 레코드를 삭제한다. 삭제 결과 데이터 페이지의 점유율이 지정된 최저 허용치 이하로 되면(underflow), 저장공간 이용률(storage utilization)을 높이기 위하여 병합을 수행한다. 병합과정은 다음과 같다. (1) 두 데이터 페이지내의 레코드들을 모두 하나의 데이터 페이지로 집중 저장하고, (2) 각 페이지가 할당된 두 영역을 하나의 영역으로 만든 후, (3) 합쳐진 영역에 (1)에서 생성된 데이터 페이지

를 할당시킨다.

계층 그리드 화일은 이러한 분할 및 병합 특성으로 저장되는 레코드의 양과 분포에 따라 상황에 적합한 구조로서 적용할 수 있다는 장점을 갖는다.

3.2 계층 그리드 화일의 구조

본 절에서는 계층 그리드 화일의 구조적 특성으로서 디렉토리 엔트리의 구조와 디렉토리의 구조에 대하여 설명하고, 디렉토리 구조로서 계층 구조를 선택한 이유에 대해서도 아울러 언급한다.

N개의 키로 구성된 데이터 공간내의 영역을 표현하는 디렉토리 엔트리는 N+1개의 애트리뷰트를 갖는 레코드로 표현되며, 따라서 디렉토리는 이러한 레코드들이 저장된 릴레이션이 된다. 각각의 키와 대응되는 N개의 애트리뷰트는 리전벡터(region vector)라 정의되며, 각 디렉토리 엔트리가 나타내는 영역의 위치 및 크기를 나타낸다. 나머지 1개의 애트리뷰트는 이 영역에 속한 레코드들이 저장되어 있는 데이터 페이지에 대한 포인터이다.

리전벡터는 각 키에 대응되는 N개의 해쉬값으로 구성된다. 한 디렉토리 엔트리의 리전벡터에서 i번째 해쉬값은 그 디렉토리 엔트리의 영역내에 속하는 모든 레코드들의 i번째 키를 해싱하였을 때 나타나는 해쉬값들의 공통 prefix가 된다.

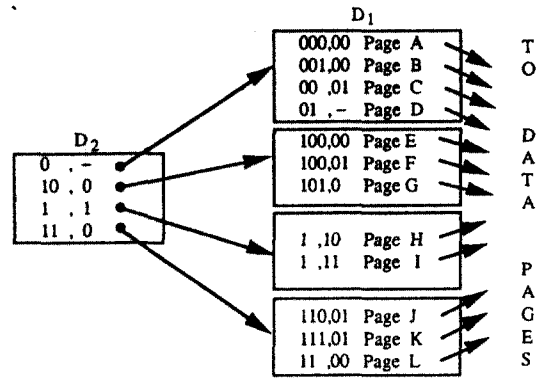
다음은 계층 그리드 화일의 디렉토리 구조로서 계층화 구조를 선택한 이유를 알아보자. 전체 디렉토리 엔트리들이 단순히 일차원의 배열구조로 저장된다면, 하나의 레코드가 어느 영역에 속하는가를 알아보기 위해 최악의 경우 모든 디렉토리 엔트리들을 조사하여야 한다. 이러한 비효율성을 제거하기 위하여 계층 그리드 화일이 취하는 디렉토리의 구조가 계층구조이다. 즉, 같은 영역에 속하는 레코드들을 같은 데이

타 페이지에 넣고 이 영역을 표현하는 디렉토리 엔트리를 디렉토리 D1에 유지하는 것과 마찬가지로, D1을 페이지 단위로 저장하기 위하여 인접한 몇 개의 디렉토리 엔트리들을 같은 디렉토리 페이지에 저장하고, 이 디렉토리 엔트리들이 나타내는 영역들을 모두 포함하는 보다 큰 영역의 디렉토리 엔트리를 D1의 상위단계 디렉토리 D2에 유지시키자는 것이다. 만일 디렉토리 D2의 디렉토리 엔트리들을 하나의 디렉토리 페이지에 저장할 수 없게 되면, 같은 방식으로 그 상위에 디렉토리 D3를 두게되며, 이것은 최상위 디렉토리의 엔트리들이 하나의 디렉토리 페이지내에 유지될 수 있을 때까지 반복된다.

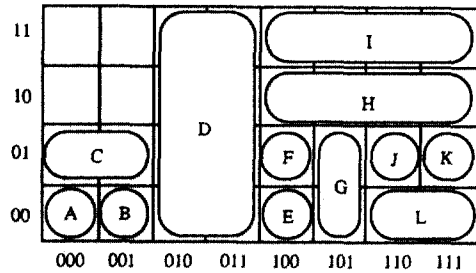
그림 1은 두개의 키를 갖는 이단계 계층 그리드 화일의 구조를 표현한 것이다. 그림 1(a)는 전체 디렉토리 구조를 나타낸 것이며, 그림 1(b)는 디렉토리 D1이 나타내는 데이터 공간의 분할을 도면화한 것이다. 그림 1(b)내의 곡면형 사각형들은 각각 D1의 디렉토리 엔트리가 나타내는 영역들을 나타낸 것이며, 내부의 문자는 그영역에 할당된 데이터 페이지를 의미한다. 따라서 디렉토리 D1에는 12개의 디렉토리 엔트리가 존재한다. D2는 4개의 디렉토리 엔트리를 가지고 있고 D1을 위한 디렉토리에 해당한다. D2의 디렉토리 엔트리 (10, 0)는 첫번째 키와 두번째 키의 해쉬값의 prefix가 각각 10, 0인 레코드들이 속한 영역을 나타내며, 이 영역은 D1에서 (100, 00), (100, 01), (101, 0)의 3개의 영역으로 다시 세분된다. 따라서 상위 디렉토리에서 하위 디렉토리로 내려올수록 보다 구체화된 데이터 공간의 분할상태가 나타난다.

4. 계층 그리드 화일을 이용한 데이터 분포의 추정

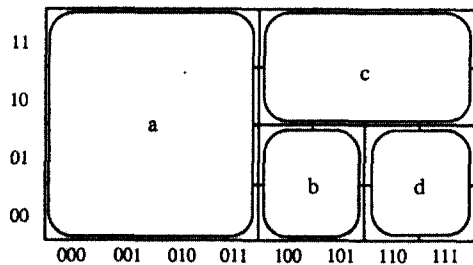
본 장에서는 계층 그리드 파일의 각 단계



(a) The Structure of a Two-level Multilevel Grid File Directory.



(b) Regions Represented by Directory Entries in D1.



(c) Regions Represented by Directory and Its Domain Space Partition

Fig. 1. Two-level Multilevel Grid File Directory and Its Domain Space Partition

디렉토리가 동적인 환경에서도 지속적으로 데이터 분포 정보를 유지할 수 있음을 밝히고, 이를 이용하여 선택률을 추정하는

방법을 제시하고자 한다. 제 4.1절에서는 계층 그리드 화일의 각 단계 디렉토리가 어떻게 데이터 분포에 대한 정보를 유지할 수 있는가를 보이고, 제 4.2절에서는 데이터 분포 정보를 주기억 장치내에서 어떻게 관리할 것인가에 대하여 기술한다. 제 4.3절에서는 이를 이용하여 선택률을 추정하는 방법에 대하여 설명한다.

4.1 계층 그리드 화일의 디렉토리 내에서 관리되는 데이터 분포 정보

계층 그리드 화일에서는 레코드의 양과 분포의 변화에 따라 자신의 디렉토리 구조를 동적으로 적응시키므로 각 단계 디렉토리에서는 항상 데이터 분포에 따른 데이터 공간의 분할 상태가 반영된다. 따라서 계층 그리드 화일에서는 각 단계 디렉토리에서 반영되는 데이터 공간의 분할 상태를 이용하여 저장된 데이터의 분포를 추정할 수 있다.

제 3.2절에서 소개된 그림 1의 예를 살펴보자. 최하위 디렉토리 D1을 도면화시킨 그림 1(b)를 보자. 영역을 나타내는 곡면형 사각형은 디렉토리 엔트리와 일대일 대응되고, 각 디렉토리 엔트리에는 하나의 데이터 페이지가 할당되어 있으므로, 각 영역에는 하나의 데이터 페이지가 할당되어 있다. 따라서 아래와 같은 가정 1(a)에 의하여 각 디렉토리 엔트리가 나타내는 곡면형 사각형의 넓이로써 전체 데이터 공간 내에서 데이터 분포 상태를 파악할 수 있다.

가정1(a): 모든 데이터 페이지내에는 같은 수의 레코드들이 존재한다.

가정 1(a)에 의하여 각 데이터 페이지에 해당되는 영역에는 같은 수의 레코드들이 존재하므로 좁은 영역내에서는 레코드들이 조밀하게 분포됨을 알 수 있고, 넓은 영역

에서는 레코드들이 희박하게 분포됨을 알 수 있다. 따라서 최하위 단계의 디렉토리에 존재하는 디렉토리 엔트리들의 집합은 전체 데이터 공간 내에서의 데이터의 분포를 나타내는 척도가 된다.

다음은 루트 디렉토리 D2를 도면화시킨 그림 1(b)를 살펴보자. 루트 디렉토리는 네 개의 디렉토리 엔트리들로 구성되어 있으므로, 전체 데이터 공간은 네 개의 영역(a,b,c,d)로 분할되어 있다. 또한 각 영역은 루트 디렉토리의 엔트리와 일대일 대응되고, 각 디렉토리 엔트리에는 하위단계 디렉토리의 페이지가 하나씩 할당되어 있다. 이러한 루트 디렉토리를 데이터 분포 정보로 사용하기 위하여 가정 1(a)와 유사한 형태의 가정 1(b)를 세운다.

가정1(b): 같은 단계의 디렉토리 페이지내에는 같은 수의 디렉토리 엔트리들이 존재한다.

즉, 가정 1(b)에 의하여 루트 디렉토리의 엔트리는 같은 수의 레코드들을 대표하게 되므로 영역의 크기로써 각 디렉토리 엔트리가 나타내는 영역에서의 데이터 분포 상태를 파악할 수 있다. 그러므로 최하위 디렉토리에서의 방식과 마찬가지로 루트 디렉토리내의 엔트리들도 전체 데이터 공간 내에서의 데이터 분포 상태를 나타내는 척도로서 사용될 수 있다.

일반적으로 계층 그리드 화일이 L개의 디렉토리 단계를 가질 때, 시스템에 저장된 데이터의 분포는 가정 1(a)와 가정 1(b)를 일반화시킨 아래와 같은 가정 1에 의하여 L개의 디렉토리에 모두 반영된다.

가정 1: 같은 단계내의 디렉토리 엔트리들이 나타내는 영역에는 같은 수의 레코드들이 존재한다.

각 단계의 디렉토리가 갖는 데이터 분포 정보의 특성을 보면, 크게 데이터 분포에 대한 정확도와 엔트리 처리를 위한 오버헤드로 구분된다. 따라서 요구되는 정확도와

오버헤드를 고려하여 데이터 분포 정보를 위한 디렉토리 단계를 적절히 선정하여야 한다.

4.2 주기억 장치 내에서의 데이터 분포 정보의 관리

참고 문헌 [Wha94]에서는 데이터 분포 관리를 위한 자료 구조로서 계층 그리드 화일의 디렉토리 자체를 이용하고 있다. 그러나 시스템에 저장된 데이터의 양이 많아지면 디렉토리가 주기억 장치내의 버퍼에 상주할 수 없게 되므로 데이터 분포 정보를 얻기 위해서는 디스크 액세스가 불가피하게 된다. 그러나 “질의 최적화를 위한 선택질의 추정” 등과 같은 순간적인 응답을 요구하는 응용에서는 디스크 액세스로 인한 시간의 지연은 바람직하지 않다. 본 절에서는 이러한 문제를 해결하기 위하여 각 애트리뷰트에 대한 데이터 분포 정보를 구성하여 이를 주기억 장치에 상주 시키는 방법을 제시한다.

애트리뷰트 별 데이터 분포 정보의 구성 예가 그림 2에 나타나 있다. 각 애트리뷰트별로 나타나는 각 요소는 그 애트리뷰트에서 아직 분할이 발생하지 않은 영역마다 하나씩 할당된다. 예를 들면, 그림 2에서의 첫번째 애트리뷰트 p는 분할이 발생하지 않은 영역이 00, 01, 1이므로 세개의 요소를 갖게 되고, 두번째 애트리뷰트 q는 00, 01, 10, 11 네개의 영역에 분할이 아직 발생하지 않았으므로 네개의 요소를 갖게 된다. 또한 한 요소의 값은 그 요소의 구간 내에 포함되는 디렉토리 엔트리 수의 합이 되며, 만일 디렉토리 엔트리가 나타내는 영역의 일부만이 요소의 구간에 속하게 되면, 그 디렉토리 엔트리가 나타내는 영역 전체 크기를 하나로 보았을 때, 요소의 구간과 교차되는 부분의 상대적인 크기를 더해주게 된다. 따라서 그림 2에서 첫번째 애트리뷰트 p의 두번째 요소값은 요소 구

간과 교차되는 부분의 크기를 고려하여 더하면, 그림에서와 같이 $1/4 + 1/4 + 1/2 + 1 = 2$ 가 된다.

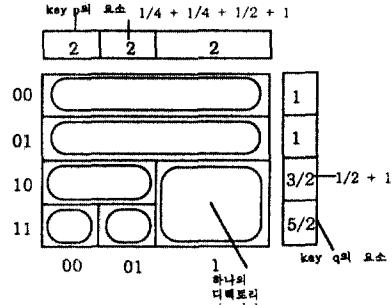


Fig. 2. Constructing Data Distribution Information Using a Multilevel Grid File Directory

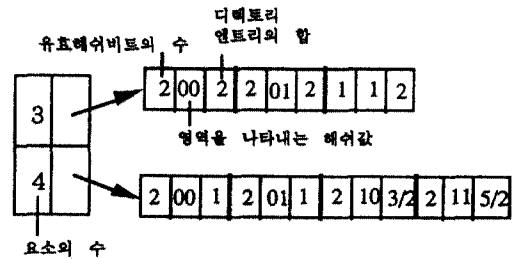


Fig. 3. A Data Structure for Maintaining Data Distribution Information

데이터 분포에 대한 확률 밀도 함수의 효과적인 추출을 위하여 데이터 분포 정보는 그림 3과 같은 자료 구조로서 주기억 장치내에서 관리된다. 요소의 수는 각 애트리뷰트에 해당되는 요소 개수를 의미하며, 유효 해쉬 비트 수는 유효 해쉬 비트 스트링의 크기로서 각 차원의 요소 구간의 크기를 파악하기 위하여 사용되는데 요소 구간의 크기는 전체 구간의 크기를 정규화하였을 때의 상대적인 크기로서 정의된다. 즉, 유효 해쉬 비트 수가 2이면 요소 구간의 크기는 $1/2^2$ 이 되고, 유효 해쉬 비트 수가 3이면 요소 구간의 크기는 $1/2^3$ 이 되므로 유효 해쉬 비트의 수를 v 라 할 때, 각 요소 구간의 크기는 $1/2^v$ 이 된다. 유효 해쉬 비트는 디렉토리 엔트리가 나타내는 영

역 혹은 질의 영역과의 교차 여부를 알기 위해 기록되며, 각 요소 구간에 속하는 디렉토리 엔트리 수의 합이 요소값이 된다.

한 애트리뷰트 a에 대한 레코드 존재 확률 밀도 함수(probability density function) f(a)와 a에서의 임의의 범위 (low, high)를 가지면, low부터 high까지의 범위에 해당

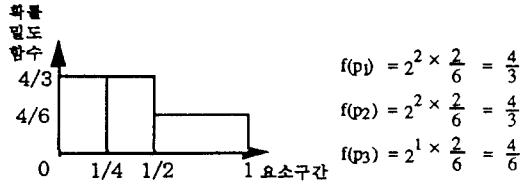
되는 레코드 존재 확률을 $\int_{low}^{high} f(a)da$ 에 의해 구할 수 있다. 여기서 각 디렉토리 엔트리가 표현하는 영역내의 데이터 분포를 균일 분포(uniform)로 가정하면, 각 데이터 분포 정보 요소의 구간내 확률 밀도 함수는 상수가 된다. 즉, v를 요소 구간 i를 표현하는 애트리뷰트 a의 해쉬값의 유효 비트 수라 하고, n_i 를 i내의 디렉토리 엔트리 수의 합, 그리고 N을 선택된 디렉토리 단계내의 전체 엔트리 수라 놓으면, a의 요소 구간 i내에서의 확률 밀도 함수 $f(a_i)$ 는 다음과 같이 구해진다.

각 요소 구간의 크기는 $1/2^v$ 이고, 그 요소 구간내의 디렉토리 엔트리 존재 확률은 n_i/N 이므로, 각각의 디렉토리 엔트리가 나타내는 데이터 영역내의 데이터 레코드의 수가 일정하다고 가정하면, $f(a_i) \times 1/2^v = n_i/N$ 가 성립된다. 따라서 $f(a_i) = 2^v \times n_i/N$ 이다.

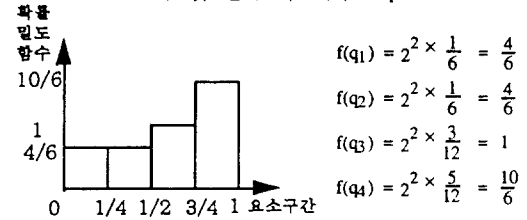
그림 2에서의 데이터 분포 정보를 바탕으로 각각의 확률 밀도 함수 $f(a_i)$ 와 그 분포를 나타내는 히스토그램은 그림 4와 같다. 히스토그램에서 y축은 확률 밀도 함수값이고, x축은 각 애트리뷰트의 도메인 전체를 1로 정규화시킨 각 요소 구간이 된다.

4.3 선택률의 추정

본 절에서는 제 4.2절의 방식으로 구한 확률 밀도 함수를 가지고, 다양한 질의에 대하여 선택률을 추정하는 기법에 대하여 논의한다.



a) 첫 번째 애트리뷰트 (p)



b) 두 번째 애트리뷰트 (q)

Fig. 4. Histogram Representing a Probability Density Function

1) 애트리뷰트 a에 대한 부분매치 질의

$f(a_i) \times \Delta a$ 로 구한다

$f(a_i)$ 는 질의 영역이 포함되는 요소 구간 i의 확률 밀도 함수이며, Δa 는 (1/유효한 a값의 수)로써 추정한다.

2) 애트리뷰트 a에 대한 범위 질의

$\sum_i (f(a_i) \times \Delta a_i)$ 로 구한다.

$f(a_i)$ 는 질의 영역과 교차되는 각 요소 구간 i의 확률 밀도 함수이며, Δa_i 는 질의 영역과 교차되는 각 요소 구간에서 교차되는 영역의 크기를 나타낸다.

3) 완전매치 질의

$\Pi_a (f(a_i) \times \Delta a)$ 로 구한다.

즉, 각 애트리뷰트에 대한 부분매치 질의의 선택률을 구한 후, 서로 다른 애트리뷰트간의 상호 종속이 없다는 가정하에 각각의 선택률을 곱하여 구한다.

4) 다중 애트리뷰트 범위 질의

$\Pi_a(\sum_i(f(a_i) \times \Delta a_i))$ 로 구한다.

즉, 각 애트리뷰트에 대한 범위 질의의 선택률을 구한 후, 서로 다른 애트리뷰트간의 상호 종속이 없다는 가정하에 각각의 선택률을 곱하여 구한다.

5. 결 론

본 논문에서는 다차원 동적 화일 구조를 이용하여 시스템에 저장된 데이터의 분포를 효과적으로 추정하는 기법을 제안하였다.

실제 데이터의 분포를 반영할 수 없는 System R에서의 균일분포 가정방식의 문제점을 해결하기 위하여 몇몇 방법이 제안되어 있으나[Pia84][Mur88][Che90], 이들이 사용하는 데이터 분포 정보는 어느 한 순간에 구성된 정적인 정보이므로, 레코드의 삽입과 삭제가 빈번히 발생하는 동적인 상황에서는 데이터의 분포정보를 주기적으로 재구성해야 하는 오버헤드가 있었다.

본 논문에서는 다차원 동적 화일 구조의 하나인 계층 그리드 화일[Wha91]을 이용함으로써 레코드가 삽입, 삭제되는 상황에서도 데이터의 분포 정보를 동적으로 유지할 수 있는 새로운 기법을 제시하고, 이를 이용하여 선택률을 추정 공식을 유도하였다. 또한, 데이터 분포 정보를 주기억 장치 내에서 관리하는 방법과 자료 구조를 제시함으로써 주기억 장치의 용량이 제한된 경우에도 이를 극복할 수 있도록 하였다. 제안된 기법의 가장 큰 장점은 데이터의 분포를 동적으로 관리함으로써 기존의 기법들이 갖는 재구성의 오버헤드를 피할 수 있다는 것이다.

참 고 문 헌

- [Ben79] Bentley, J.L., "Multi-dimensional Binary Search Trees in Database Applications," *IEEE Tran on Software Engineering*, Vol. SE-5, No. 4, pp. 333-340, July 1979.
- [Bla77] Blasgen, M.W. and Eswaran, K.P., "Storage and Access in Rel-ational Databases," *IBM Systems Journal*, Vol. 16, No. 4, pp.363-337, 1977.
- [Bur83] Burkhard, W.A., "Interpolation-Based Index Maintenance," In *Proc. 2nd ACM Symp. on Principles of Database Systems*, ACM SIGMOD, pp. 79-89, 1983.
- [Che90] Chen, M.C.et al., "Selectivity Estimation Using Homogeneity Measurement," In *Proc. Intl. Conf. on Data Engineering*, IEEE, Los Angeles, pp. 304-310, Feb. 1990.
- [Fre87] Freeston, M., "The BANG File: A New Kind of Grid File," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 260-269, 1987.
- [Man88] Mannio, M.V. et al., "Statisticational Profile Estimation in Database Systems," *ACM Computing Surveys*, Vol. 20, No. 3, pp.191-221, Sept. 1988.

- [Mur88] Muralikrishna, M. and DeWitt, D., "Equi-Depth Histograms for Estimating Selectivity Factors for Multi-Dimensional Queries," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 28-36, June 1988.
- [Nie84] Nievergelt, J. et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol. 9, No. 1, pp. 38-71, Mar. 1984.
- [Ouk84] Ouksel, M. and Scheuermann, P., "Storage Mapping for Multidimensional Linear Hashing," In *Proc. 3rd ACM Symp. on Principles of Database Systems*, ACM SIGMOD, pp. 90-105, 1984.
- [Oto84] Otoo, E.J., "A Mapping Function for the Directory of a Multidimensional Extendible Hashing," In *Proc. 10th Intl. Conf. on Very Large Databases*, pp. 493 - 506, 1984.
- [Pia84] Piatetsky, S.G. and Connell, G., "Accurate Estimation of the Number of Tuples Satisfying a Condition," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 256-276, 1984.
- [Rob81] Robinson, J.T., "The K-D-B Tree: A Search Structure for Large Multidimensional Dynamic Indexes," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 10-18, 1981.
- [Sel79] Selinger, P.G. et al., "Access Path Selection in a Relational Database Management System," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 23-34, May 1979.
- [Wha91] Whang, K.Y. and Krishnamurthy, R., "The Multilevel Grid File: A Dynamic Hierarchical Multidimensional File Structure," In *Proc. 2nd Intl. Conf. on Database Systems for Advanced Applications*, pp. 449-459, Apr. 1991.
- [Wha94] Whang, K.Y., Kim, S.W., and Wiederhold, G., "Dynamic Maintenance of Data Distribution for Selectivity Estimation," *The VLDB Journal*, Vol. 3, No. 2, pp. 29-51, 1994.