범위 모자이크 질의와 범위 모자이크 상위-k 질의의 효율적인 수행[1)]

홍석진° 이석호
서울대학교 컴퓨터공학부
jinny@db.snu.ac.kr°, shlee@cse.snu.ac.kr

Efficient Execution of Range Mosaic Query and Range Mosaic Top-k Query

Seokjin Hong° Sukho Lee
School of Computer Science and Engineering, Seoul National University

요 약

범위 통계 질의는 범위 집계 질의와 같이 질의 영역 내에 포함된 데이타의 통계 정보를 반환하는 질의를 의미한다. 이 논문에서는 새로운 범위 통계 질의로 범위 모자이크 질의와 범위 모자이크 상위-k 질의를 소개한다. 범위 모자이크 질의는 질의 영역을 다차원 격자 형태로 분할 한 후, 분할된 각 셀에 대해 집계값을 구하는 질의이며, 범위 모자이크 상위-k 질의는 범위 모자이크 질의 결과 중 집계값을 기준으로 상위 k개의 셀을 구하는 질의이다. 이 논문에서는 집계 R-트리를 사용하여 두 종류의 질의를 효율적으로 수행하는 알고리즘을 제안한다. 또한, 실험 결과를 통해 제안된 알고리즘이 생성된 데이타와 실제 데이타 모두에 대해 좋은 성능을 나타내는 것을 보인다.

1. 서론

범위 집계 질의(range aggregate query) [1, 2]는 특정 질의 영역 내의 데이타 요소에 대한 합, 평균 등의 집계값을 반환하는 질의로, 범위 통계 질의(range statistical query)의 한 종류이다. 이러한 범위 집계 질의는 공간 데이타베이스 [3]와 데이타 웨어하우스 [4] 환경에서 분석을 위한 유용한 도구로 사용된다. 하지만 범위 집계 질의는 질의 영역에 대한 단일 집계값만을 반환하기 때문에, 질의 영역 내의 데이타의 분포를 알아내기에는 적합하지 않다. 이 논문에서는 질의 영역 내의 데이타에 대한 개략적인 분포를 알아내는 질의로, 범위 모자이크 질의 (range mosaic query)와 범위 모자이크 상위-k 질의 (range mosaic top-k query)를 제안한다. 범위 모자이크 질의는 질의 영역을 격자 형태의 모자이크로 나눈 후, 격자의 각 셀에 대한 집계값을 구하는 질의이며, 범위 모자이크 상위-k 질의는 범위 모자이크 질의 결과 중 집계값을 기준으로 상위 k개의 셀을 구하는 질의이다.

범위 모자이크 질의와 범위 모자이크 상위-k 질의는 다양한 분야에 적용할 수 있다. 움직이는 자동차의 위치를 저장하는 시공간 데이타베이스에서 특정 시간 구간 동안 주어진 영역 내에 있는 차량의 대략적인 분포를 구하는 질의는 대표적인 예라 할 수 있다. 질의 영역을 시공간의 3차원 격자로 나누고, 나뉜 각 영역에 대해 count 연산을 수행하는 범위 모자이크 질의를 통해, 특정 시간 구간 동안 주어진 영역 내의 차량 분포를 구할 수 있다. 범위 모자이크는 데이타 웨어하우스 환경에도 적용 가능하다. 예를 들어 고객의 수입과 나이를 차원 애트리뷰트로, 판매량을 값 애트리뷰트로 갖는 사실 테이블(fact table)이 있다고 할 때, "수입이 2000만원에서 5000만원 사이인 30, 40, 50대의 고객에 대해, 수입을 4 단계, 나이를 3 단계로 하여 그룹을 나눈 후 각 그룹별로 평균 판매량을 구하라."와 같은 질의를 생각해 볼 수 있다. 이 외에도 센서를 통해 측정된 기온이나 강수량의 분포, 또는 도시의 인구분포 등도 범위 모자이크 질의를 수행할 수 있는 좋은 대상이다.

이 논문에서는 범위 모자이크 질의와 범위 모자이크 상위-k 질의를 지원하기 위한 새로운 SQL 구문인 mosaic-by 절을 제안한다. mosaic-by 절은 group-by 절과 마찬가지로 레코드를 그룹짓는데 사용하며, 연속적인 도메인에 적용 가능한 구문이다. 또한 이 논문에서는 집계 R-트리(aggregate R-tree) [1, 2]를 기반으로 하여 두 종류의 질의를 효율적으로 수행하는 알고리즘을 제안한다.

이 논문의 구성은 다음과 같다. 2절에서는 범위 통계 질의와 집계 R-트리에 대해 살펴본다. 3절에서는 범위 모자이크 질의와 범위 모자이크 상위-k 질의를 살펴본 후, 집계 R-트리 기반의 질의 처리 알고리즘을 소개한다. 4절에서는 실험 결과를 분석하고 5절에서 결론을 맺는다.

2. 관련 연구

범위 모자이크 질의는 범위 통계 질의의 일종이다. 이 절에서는 범위 통계 질의를 살펴본 후, 질의 처리를 위한 기본적인 자료구조인 집계 R-트리에 대해 살펴보도록 하겠다.

범위 통계 질의(range statistical query)는 특정 질의 영역 내의 데이타에 대한 통계적인 결과를 반환하는 질의로, 지금까지 연구된 범위 통계 질의로는 범위 집계 질의(range aggregate query) [1, 2]와 범위 상위-k 질의(range top-k query) [5] 등이 있다. 범위 집계 질의는 질의 영역 내의 데이타에 대한 합, 평균 등의 집계값을 반환하며, 범위 상위-k 질의는 질의 영역의 데이타 중 상위 k개를 반환한다.

이러한 범위 통계 질의를 효율적으로 수행하기 위해 사용되는 자료구조로 집계 R-트리(aggregate R-tree) [1, 2]가 있다. 집계 R-트리는 R-트리 [6]의 변형으로 자식 노드의 집계값을 부모 노드의 각 엔트리마다 저장하는 구조이다. 집계 R-트리를 통해 범위 통계 질의를 수행하면, 질의 영역에 완전히 포함되는 노드의 경우, 부모 노드의 엔트리에 저장되어 있는 해당 노드의 집계값을 사용하므로, 해당 노드 및 해당 노드의 자식 노드들을 더 이상 탐색할 필요가 없다. 따라서 노드 접근

1) 본 연구는 2005년도 두뇌한국21사업과, 정보통신부의 대학 IT연구센터(ITRC) 지원을 받아 수행되었습니다.

회수를 줄일 수 있고 질의 영역의 크기에 관계없이 거의 일정한 성능을 보인다는 장점이 있다. MRA R-트리(MRA R-tree) [2], 집계 큐브트리(aggregate cubetree) [1] 등이 이러한 집계 R-트리에 속한다.

3. 범위 모자이크 질의

3.1 Mosaic-by 절

이 논문에서는 범위 모자이크 질의를 표현하기 위해 SQL문에 mosaic-by 절을 추가하였다. Mosaic-by 절의 구문은 다음과 같다.

$$\text{MOSAIC}(g_1, g_2, \ldots, g_n) \text{ BY } d_1, d_2, \ldots, d_n$$

d_i는 차원 애트리뷰트이고, 각 차원 i는 g_i개의 동일한 크기의 구간으로 분할된다. 예를 들어 "MOSAIC(3, 2) BY a, b"와 같이 표현하면 레코드를 애트리뷰트 a, b에 대해 3×2 형태의 2차원 격자로 그룹지으라는 의미를 갖는다.

위에서 정의한 mosaic-by 절을 통해 아래와 같이 범위 모자이크 질의를 표현할 수 있다.

```
SELECT start(d1), end(d1), start(d2), end(d2), ...,
       start(dn), end(dn), aggregate(v)
FROM R
MOSAIC(g1, g2, ..., gn) BY d1, d2, ..., dn
WHERE d1>=m1 and d1<=M1 and d2>=m2 and
      d2<=M2 and ... and dn>=mn and dn<=Mn
```

위의 질의에서 질의 영역은 where 절을 통해 표현된다. 질의 영역 내의 레코드는 mosaic-by 절을 통해 n차원 격자로 그룹지어지며, 격자의 각 셀마다 집계 함수가 수행된다. 이 때, start(d_i)와 end(d_i)는 각 셀의 차원별 시작 좌표와 끝 좌표를 반환하는 함수이다.

범위 모자이크 상위-k 질의 역시 mosaic-by 절을 사용하여 표현할 수 있으며 아래와 같이 select 절에 'TOP k' 구문을 사용하면 된다.

```
SELECT TOP k start(d1), end(d1), start(d2), end(d2), ...,
       start(dn), end(dn), aggregate(v)
FROM R
MOSAIC(g1, g2, ..., gn) BY d1, d2, ..., dn
WHERE d1>=m1 and d1<=M1 and d2>=m2 and
      d2<=M2 and ... and dn>=mn and dn<=Mn
```

3.2 범위 모자이크 질의의 수행

범위 모자이크 질의를 수행하는 간단한 방법으로는 RQA (Range Query and Aggregation, 범위 질의 후 집계) 기법이 있다. RQA 기법에서는 R-트리를 통해 해당 영역에 대한 범위 질의를 수행한 후, 결과로 얻어진 질의 영역 내의 모든 레코드를 각각 해당하는 모자이크 셀로 그룹짓는다. 그런 다음, 각 모자이크 셀별로 집계함수를 수행하여 결과를 구하게 된다. RQA 기법은 질의 영역 내의 모든 레코드를 접근해야 하므로 질의 영역이 크기가 커질수록 노드 접근 회수가 증가한다는 단점이 있다.

이 절에서는 집계 R-트리를 사용하여 범위 모자이크 질의를 수행하는 MCU(Multiple Cell Update, 다중 셀 갱신) 기법을 소개한다. MCU 기법은 집계 R-트리를 이용한 범위 집계 질의 처리 기법[1, 2]의 확장으로, 큐를 통해 집계 R-트리 노드를 순회한다. 큐에는 모자이크 셀에 부분적으로 겹치는 노드의 포인터를 유지한다. 질의 수행 방법은 다음과 같다. 큐에서 포인터를 하나 뽑아서 해당 노드를 읽고, 해당 노드의 엔트리에 대해 모자이크 셀에 완전히 포함되는 지 여부를 조사한다. 모자이크 셀에 완전히 포함되는 경우에는 저장되어 있는 집계값을 셀에 집계하고, 부분적으로 겹치는 경우에는 자식 노드의 포인터를 큐에 넣는다. 이러한 작업을 큐가 빌때까지 반복한다. 알고리즘 1에 MCU 기법의 자세한 알고리즘이 기술되어 있다.

```
1  Algorithm Range_Mosaic
2  Input: 집계 R-트리, 범위 모자이크 질의
3  Output: 각 모자이크 셀에 대한 집계값
4  begin
5      queue.enqueue(p_root_node);
6      repeat
7          p ← queue.dequeue();
8          n ← read_node(p);
9          foreach e in n.entries do
10             if c ← e.included(mosaic_grid) then
11                 c.aggregate(e.aggregate_value);
12             else if e.overlap(mosaic_grid) then
13                 queue.enqueue(e.child_node_pointer);
14             end if
15         end for
16     until queue.is_empty();
17 end
```

알고리즘 1. 범위 모자이크 질의 알고리즘 (MCU)

3.3 범위 모자이크 상위-k 질의의 수행

범위 모자이크 상위-k 질의는 범위 모자이크 질의 결과 중 상위 k개의 셀을 결과로 반환하는 질의로, 이전 절에서 설명한 RQA 및 MCU 기법을 통해서 수행할 수 있다. 하지만 이 경우 k 개의 결과에 포함되지 않는 모든 셀의 집계값을 구하게 되므로, 필요 없는 노드 접근이 발생하는 단점이 있다.

이 절에서는 범위 모자이크 상위-k 질의 처리를 위해, 결과에 포함될 가능성이 없는 셀을 미리 제외하는 CP(Cell Pruning, 셀 가지치기) 기법을 제안한다. CP 기법에서는 각 셀별로 집계값의 상한과 하한을 아래의 식을 통해 유지한다.

$$c.lower = e_1^i.value \oplus e_2^i.value \oplus \ldots \oplus e_n^i.value$$

$$c.upper = c.lower \oplus e_1^o.value \oplus e_2^o.value \oplus \ldots \oplus e_m^o.value$$

위 식에서 셀 c의 집계값의 상한과 하한은 각각 $c.upper, c.lower$로 나타낸다. $e_1^i, e_2^i, \ldots, e_n^i$은 셀 c에 완전히 포함된 엔트리를, $e_1^o, e_2^o, \ldots, e_m^o$은 셀 c와 부분적으로 겹치는 엔트리를 나타내며, $\oplus$는 집계함수를 의미한다.

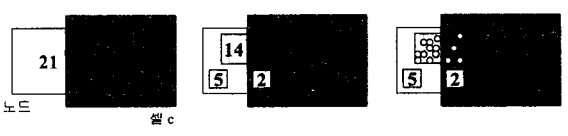


그림 1. 셀의 상한과 하한의 갱신

노드로부터 엔트리를 읽을때마다 해당 엔트리와 겹치는 셀의 집계값의 상한값과 하한값을 갱신한다. 그림 1은 셀의 집계값의 상한값과 하한값이 갱신되는 과정의 예이다. 첫 번째 그림에서는 집계값으로 32를 갖는 엔트리는 셀 c에 완전히 포함되고, 21을 갖는 엔트리는 부분적으로 겹치고 있다. 따라서 셀 c의 집계값의 상한값은 53, 하한값은 21이 된다. 질의가 수행될수록 두 번째, 세 번째 그림과 같이 하위 노드를 접근하게 되며, 그때마다 셀의 집계값에 대한 상한값과 하한값을 갱신하게

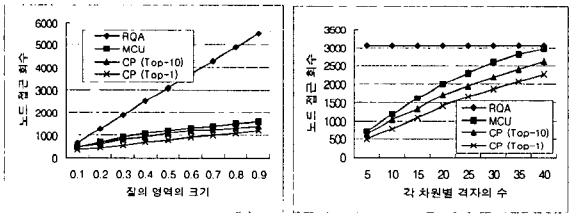
된다. 세 번째 그림과 같이 셀과 부분적으로 겹치는 엔트리가 하나도 없게 될 경우 집계값의 상한값과 하한값이 같아지며, 이 값이 바로 셀의 최종 집계값이 된다.

결과에 포함될 가능성이 없는 셀을 제외하기 위해, CP 기법에서는 질의가 수행되는 동안 상위 k개의 하한값을 유지한다. 셀의 상한값과 하한값이 갱신되는 동안 셀의 상한값이 상위 k개의 하한값보다 작아지는 경우 해당 셀은 질의 수행에서 제외된다. 이는 상한값이 상위 k개의 하한값보다 작은 셀은 더 이상 상위 k개의 결과에 포함될 수 없기 때문이다. 이러한 작업을 셀 가지치기라고 한다.

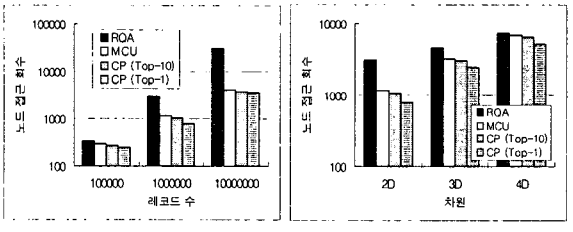
CP 기법의 알고리즘 골격은 알고리즘 1에 나와있는 MCU 기법과 비슷하다. 차이점을 살펴보면, 먼저 CP 기법에서는 노드의 엔트리를 처리할 때마다 셀 가지치기를 수행한다 (9행). 또한 10행의 e.included()와 12행의 e.overlap() 함수에서는 셀 가지치기에서 제외된 셀을 뺀 나머지의 셀과 엔트리와의 관계를 비교하게 된다.

4. 실험 및 분석

RQA, MCU, CP 기법의 성능을 실험을 통해 비교 평가 하였다. 2차원부터 4차원까지의 균등(uniform) 분포를 따르는 데이터셋과, 현실 세계를 반영하는 TPC-H 데이타셋 [7], Tiger/Line 데이타셋 [8]을 사용하였다. 각 실험은 주메모리 512MB, 하드 40GB인 펜티엄3 1GHz 시스템에서 수행하였다. R-트리와 집계 R-트리의 페이지 크기는 4KB로 하였다.



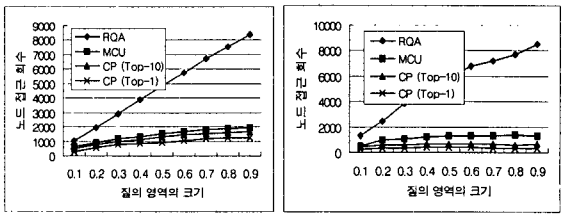
(a) 질의 영역의 크기 (b) 모자이크 셀의 개수



(c) 레코드 수 (d) 차원

그림 2. RQA, MCU, CP기법의 성능비교

그림 2는 균등 분포 데이타셋에 대한 실험 결과이다. 그림 2(a)는 질의 영역의 크기 변화에 따른 각 기법의 노드 접근 회수를 비교한 것이다. 10×10 모자이크를 사용하여 실험을 수행하였다. RQA기법의 경우 질의 영역의 크기가 커짐에 따라 노드 접근 회수가 크게 증가하는 반면, MCU와 CP의 경우 질의 영역의 크기가 커져도 노드 접근 회수가 크게 증가하지 않는다. 그림 2(b)는 질의영역의 크기를 50%로 고정한 후 모자이크 셀의 수를 변화시켜 가면서 실험한 결과이다. 차원별 격자의 수에 의해 모자이크 셀의 수가 결정된다. RQA의 경우 질의 영역 내의 모든 노드를 접근해야 하므로 항상 많은 노드 접근 회수를 보이는 반면, MCU와 CP의 경우에는 작은 수의 모자이크에 대해 매우 좋은 성능을 보이는 것을 알 수 있다. 또한 두 실험 모두 상위 일부 셀만을 구하는 CP의 경우 MCU보다 적은 노드 접근 회수를 보인다. 그림 2(c)는 레코드 수의 증가에 따른 노드 접근 회수를 측정한 결과이고, 그림 2(d)는 차원에 따른 실험 결과이다. 두 실험 결과 모두 MCU와 CP기법의 확장성을 보여주고 있다.



(a) TPC-H 데이타셋(H) (b) Tiger Line(TL)

그림 3. 실제 데이타셋에 대한 실험 결과

그림 3은 실제 데이타셋을 대상으로 실험한 결과이다. 그림 3(a)는 TPC-H의 스키마 중 orders 테이블의 레코드 150만개를 생성하여 사용한 결과이고, 그림 3(b)는 Tiger/Line 데이타 중 레코드 수가 약 140만 개인 2차원 데이타인 2002년의 미국 뉴욕 지역의 데이타를 사용한 결과이다. 두 실험 모두 균등 분포 데이타셋과 비슷한 결과를 보이는 것을 알 수 있다.

5. 결론

이 논문에서는 새로운 범위 통계 질의로 범위 모자이크 질의와 범위 모자이크 상위-k 질의를 소개하고, 각 질의의 효율적인 수행을 위한 집계 R-트리 기반의 MCU와 CP 기법을 제안하였다. 두 기법은 질의 영역 내의 모든 레코드를 접근하지 않고 적은 수의 노드 접근만으로 질의를 수행할 수 있으며, 실험 결과를 통해 제안된 알고리즘이 생성된 데이타와 실제 데이타 모두에 대해 좋은 성능을 나타내는 것을 알 수 있다.

참고 문헌

[1] Seokjin Hong, Byoungho Song, and Sukho Lee, "Efficient execution of range aggregate queries in data warehouse environments," Conceptual Modeling - ER 2001: 20th International Conference on Conceptual Modeling, Yokohama, Japan, pp. 299-310. Springer, 2001.

[2] Iosif Lazaridis and Sharad Mehrotra, "Progressive approximate aggregate queries with a multi-resolution tree structure," Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, California, May 21-24, 2001, pp. 401-412. ACM Press, 2001.

[3] Ralf Hartmut Guting, "An Introduction to Spatial Database Systems," VLDB Journal vol 3, No.4, pp.357-399, 1994.

[4] Surajit Chaudhuri, Umeshwar Dayal, "An Overview of Data Warehousing and OLAP Technology," SIGMOD Record vol.26, No.1, pp.65-74, 1997.

[5] Seokjin Hong, Bongki Moon, and Sukho Lee, "Processing range top-k queries in a sparse data cube," Proceedings of the International Conference on Information and Knowledge Engineering. CSREA Press, 2004.

[6] Antonin Guttman, "R-trees: A dynamic index structure for spatial searching," SIGMOD'84, Proceedings ACM SIGMOD International Conference on Management of Data, June 18-21, 1984, Boston, Massachusetts, pp. 47-57. ACM Press, 1984.

[7] TPC-H. http://www.tpc.org/tpch/, 2004.

[8] TIGER/Line. http://www.census.gov/geo/www/tiger/, 2002.