해쉬 기반의 다중 공간 조인 기법

°박준용*, 김명근*, 배해영*
*인하대학교 컴퓨터・정보공학과
e-mail: afetra@dblab.inha.ac.kr

Hash based Multi-way Spatial Join Method

°Jun-Yong Park*, Myoung-Keun Kim*, Hae-Young Bae*
*Dept. of Computer Science & Engineering, Inha University

요 약

최근 공간 데이터베이스 시스템은 공간 데이터 마이닝, 공간 그리드, LBS 등과 같은 환경의 여러 분야로 그 응용이 확대되고 있으며, 사용자들의 다양한 공간 질의 처리 요구에 부응하는 효율적인 공간 질의 처리의 필요성이 대두되었다. 특히 다중 공간 조인 질의와 같은 고비용의 공간 조인에 대한 효율적인 공간 질의 처리를 위해서는 단일 공간 조인 보다 더욱 많은 기하 계산과 공간 데이터의 특성을 이용한 다중 공간 조인 처리 방법이 필요하게 되었다.

본 논문은 고비용의 다중 공간 조인을 효율적으로 수행하기 위해 다중 공간 조인 연산을 효율적으로 처리하기 위하여 병렬적인 질의 수행을 위해 해쉬 기반의 다중 공간 조인 기법을 제안한다. 제안하는 기법은 기존 조인 대상이 되는 두 데이터 집합에만 공간 해쉬 조인을 수행하는 것을 확장하여 다중 공간 데이터 집합에 대해서도 조인이 수행 가능한 방법을 제안함으로써 최적의 조인 후보를 획득하여 효율적인 질의 처리를 수행하는 것을 보장한다.

1. 서론

공간 데이터베이스 시스템의 기본 연산인 공간 조인은 주어진 공간 조건을 만족하는 객체의 쌍들을 찾는 공간 질의의 한 종류로서 점질의(point query)나 질의 영역(region query)과는 달리 조인하는 데이터 집합에 대한 다중 주사(multi-scan) 방식의 질의이기 때문에 객체의 수가 증가함에 따라 연산 시간이 급격히 증가한다[1]. 그래서 공간 조인을 효율적으로 수행하기 위해 지금까지 공간에 대해 색인, 조인 방법, 그리고 조인 병렬화 연구가 진행되었다[1,2,3]. 특히, 다중(multi-way) 공간 조인 질의는 2-way 조인들을 연결시킨 조인 질의로서 실행 비행이 크고 질의 실행 계획 작성이 복잡하여[4,5] 데이터베이스 성능 향상을 위해서는 다중 공간 조인 방법에 적용할 수 있는 처리 방법이 필요하다.

본 논문[1]에서는 제안하는 해쉬 기반의 다중 공간 조인 처리 기법은 병렬 처리를 지원하면서 가장 성능이 좋은 조인 방법으로 평가 받는[6] 해쉬 조인을 다중 공간 데이터 집합에 적용한 기법이다. 공간 조인의 복잡하고 방대한 양의 처리 시간으로 인해 공간 조인은 여과 단계(filter stp)와 정제 단계(refinement step)로 불리는 2단계 처리 과정으로 전체 질의 처리 시간을 줄이기 위해 사용되어 왔는데[6] 이 방법은 여과와 정제 연산을 분리하여 각각에 대해 태스크 생성 및 할당을 해야 하는 추가적인 비용이 필요하다. 따라서 본 논문에서는 다중 공간 조인을 처리하기 위해 R-tree를 기반으로 한 해쉬 기반의 다중 공간 조인기법을 수행하여 최적의 조인 후보를 찾은 다음에 조인 후보에 대하여 병렬적으로 조인을 수행하는 해쉬 기반의 다중 공간 조인 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에

1) 본 연구는 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

서는 공간 데이터베이스 시스템에서 공간데이터를 효율적으로 관리하기 위해 R-tree 공간 인덱스를 이용한 다중 공간 조인 처리 방법과 기존의 공간 해쉬 조인 방법에 대하여 살펴본다. 그리고 3장에서는 본 논문에서 제시하는 해쉬 기반의 다중 공간 조인 기법에 대해 알아본 후, 마지막으로 4장에서는 결론 및 향후 연구에 대하여 기술한다.

2. 관련연구

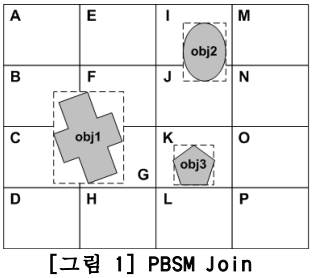
본 장에서는 R-tree 인덱스를 이용한 다중 공간 조인 기법과 두 가지 공간 해쉬 조인 기법에 대해 설명한다.

2.1 R-tree 인덱스를 이용한 다중 공간 조인

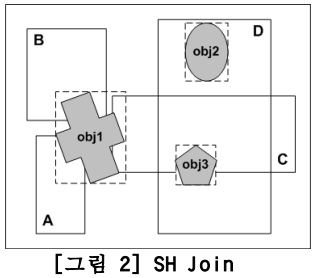
다중 공간 조인 질의 수행 비용은 네트워크 기술 발달로 점차적으로 데이터 전송 비용보다 공간 연산 수행시 발생하는 비용이 많아지게 되었다. 따라서 다중 공간 조인 처리의 최적화에는 공간 연산의 수행 비용의 최소화에 중점을 두는 연구가 필요하다. 공간 조인이란 주어진 공간 조건을 만족하는 모든 공간 객체의 쌍들을 찾는 연산의 한 종류로서 많은 디스크 I/O와 처리시간을 필요로 하므로, 효과적인 처리를 위한 알고리즘이 많이 진행되어 왔다[1]. 다중 공간 조인을 처리하기 위한 방법으로 2중 조인의 연속으로 처리하는 방법과 조인에 참가하는 모든 공간 객체 집합들에 대해 R-tree 인덱스가 존재할 경우, 다중의 R-tree 인덱스를 동시에 탐색하는 방법이 있다[7]. 여기서 R-tree 인덱스를 이용한 방법은 연속으로 2중 조인 처리와는 달리 중간 결과를 생성하지 않고, 일부 공간 객체 쌍에 대해 불필요한 정제 단계 연산을 수행하지 않기 때문에 보다 효율적이다[8]. 따라서 본 논문에서는 다중 공간 데이터 집합에 대하여 R-tree 인덱스를 구축하여 다중 공간 조인을 수행한다.

2.2 공간 해쉬 조인 기법

공간 해시 조인은 공간 데이터에 대해 해시 조인을 적용하는 기법으로 파티션을 기반으로 한 기법(PBSM Join: Partition Based Spatial-Merge Join)과 Seeded Tree를 기반으로 한 기법(SH Join: Spatial Hash-Joi)에서 제안되었다[2,9,10]. 파티션 기반의 공간 합병 조인 기법은 [그림1]과 같이 데이터 공간을 일정 수의 파티션으로 나누고 이를 버켓으로 사용하여 객체를 할당하게 되고, 객체를 할당하게 될 때 obj1과 같이 B,F,C,G의 여러 버켓에 하나의 객체가 중복되어 할당하게 된다. 파티션의 개수가 정해지고 타일이 각 파티션에 맵핑되고 나면 각 파티션에 할당된 객체들을 검사하여 공간 조인 후보에 대해서 조인을 수행하게 되는데 파티션에 속한 객체들을 메모리에 로드한 후 plane sweep 기법을 사용하여 조인을 수행한다. 이 기법의 단점으로는 비대칭적인 공간 해시 조인 기법이며 중복된 조인 결과가 생성되어 이를 검출하기 위한 추가적인 연산이 필요하다.



[그림 1] PBSM Join



[그림 2] SH Join

Seeded Tree를 기반으로 한 해시 함수를 이용하는 공간 해시 조인은 [그림2]와 같이 한 데이터 집합의 객체를 샘플링하여 트리를 구성한 다음 트리의 단말 노드를 버켓으로 만들고 샘플링한 데이터 집합의 객체를 다시 버켓에 할당한 후 다른 데이터 집합의 객체를 할당을 한다. Obj1과 같이 A,B,C의 여러 버켓에 하나의 객체가 중복되어 할당되고 Obj3는 C,D에 중복되어 할당되지만 샘플링한 데이터 집합의 객체는 하나의 버켓에 하나의 객체가 할당되므로 중복된 결과가 발생하지 않게 된다. 다른 데이터 집합의 객체들이 각각의 파티션에 할당된 후, 조인 후보에 대해 조인을 수행하는 단계를 진행한다. 한 파티션에 속한 객체들을 메모리에 로드하고 R-Tree 인덱스를 구성한 다음, 그 인덱스를 사용하여 조인을 수행는데, 만약 R-Tree를 구성하는 동안 메모리가 부족하게 되면 Outer에 속하는 객체에 LRU 정책을 적용하여 조인을 수행한다. 위의 두가지 공간 해쉬 기법은 공간 조인 알고리즘들은 조인의 대상이 되는 두 데이터 집합에 대해서만 집중되어 있다. 따라서 다중 공간 데이터 집합에 대한 조인을 처리하기 위한 처리 방법이 필요하다. 본 논문에서는 다중의 공간 데이터에 대한 병렬 처리를 위하여 해쉬 기반의 공간 조인 방법을 사용한다.

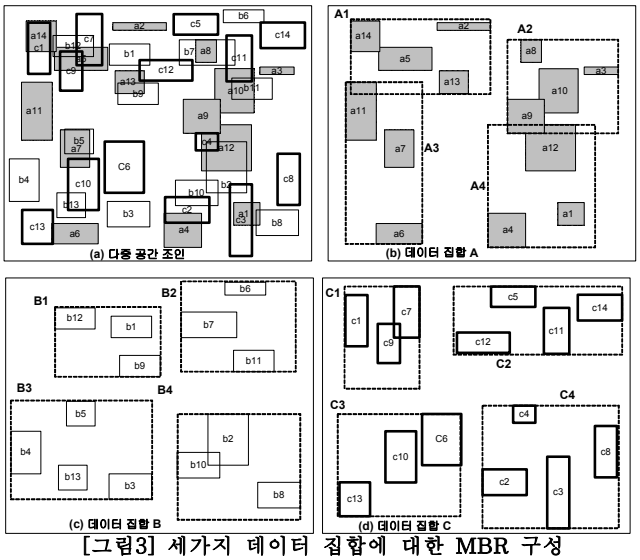
3. 다중 공간 조인의 최적화 방법

이 장에서는 본 논문에서 제안하는 기법인 최적의 다중 공간 조인을 수행하기 위해 R-tree 인덱스를 사용하는 해쉬 기반의 다중 공간 조인에 대해 설명한다. 본 논문에서는 폴리곤 데이터(Polygon Data)

인 다각형인 집합 A, B, C에 여러 다중 공간 조인 연산 중 겹침 연산(overlap operation)을 사용한다.

3.1 해쉬 기반의 다중 공간 조인

기존의 세 단계 과정을 통해 수행되는 공간 해쉬 조인[10]을 다중 공간 조인을 처리하기 위하여 확장하여 R-tree 인덱스를 사용하는 해쉬 기반의 다중 공간 조인을 제안한다.



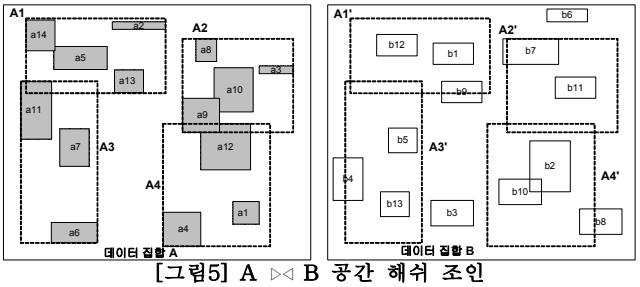
[그림3] 세가지 데이터 집합에 대한 MBR 구성

[그림4] 공간 객체의 MBR과 R-tree 인덱스 구성

[그림3]와 같이 세 가지 데이터 집합에 대하여 아래와 같이 A, B, C에 대하여 중복연산(overlap operation)으로 공간 조인이 정의될 때 이를 다중 공간 조인이라 하고, 이 다중 공간 조인은 A, B의 공간 조인과 B, C의 공간 조인이 AND 조건으로 연결되어 있음을 나타낸다.

$$A \bowtie B \bowtie C = A \bowtie B \text{ AND } B \bowtie C$$

이와 같은 다중 공간 조인에 대하여 확장된 해쉬 공간 조인은 우선 각 데이터 셋에 대해 조인 순서가 위와 같을시 데이터 집합 A의 객체에 대해 [그림 3-b]와 같이 MBR 값으로 네 개의 버켓(A1～A4)을 구성한 후 데이터 집합 A의 객체에 대하여 해쉬테이블을 구성한다. 조인할 대상인 데이터 집합 B에 동일한 버켓(A1～A4)을 적용하여 할당된 버켓에 속하는 B의 객체를 찾는다.



[그림5] A ⋈ B 공간 해쉬 조인

데이터집합 A, B에 대해 4개의 할당된 버켓에 속하는 객체를 조사한 결과는 아래와 같고,

데이터 집합 A 객체에 대해 구성된 해쉬테이블을

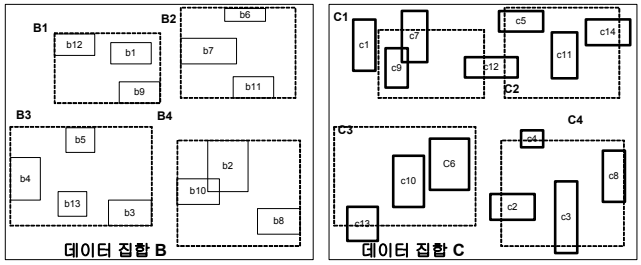
A1={a5,a13,a14} A2={a8,a9,a10} A3={a7} A4={a1,a12}

A1'={b1,b9,b12} A2'={b7,b11} A3'={b4,b5,b13} A4'={b2,b8,b10}

이용해서 데이터 집합 B에 해쉬함수를 적용하여 조인을 수행할 최종 조인 후보는 아래와 같다.

A1'={b9,b12} A2'={b7,b11} A3'={b5} A4'={b2}

데이터 집합 B의 b3과 b6은 어떤 버켓의 범위에도 중첩되지 않기 때문에 제거되고, 다중 공간 해쉬 조인은 [A1, A1′]와 같이 버켓에 속하는 쌍을 찾아 조인을 수행하게 된다. A ⋈ B에 대한 조인 후보를 찾아보면 (a1, b2), (a5, b12), (a7, b5), (a8, b7), (a9, b2), (a10, b11), (a12, b2), (a13, b9), (a14, b12)로 조인 대상의 후보가 결정된다. [그림6]의 B ⋈ C 공간 해쉬 조인은 A ⋈ B와 같은 방법으로 데이터 집합 B의 MBR 크기로 버켓을 생성(B1～B4)하여 데이터 집합 C에 속하는 조인 후보를 찾고,



[그림6] B ⋈ C 공간 해쉬 조인

버켓에 속하는 B ⋈ C 공간 해쉬 조인 후보는 아래와 같다.

B1={b9,b12} B2={b7, b11} B3={b5} B4={b2}

B1'={c7,c9,12} B2'={c5,c11,c12,14} B3'={c6,c10,c13} B4'={c2,c3,c4,c8}

위의 후보들에 대해서 A ⋈ B에 수행되는 데이터 집합 B의 조인 후보인 {b2, b5, b7, b9, b11, b12}에 대해서만 해쉬 테이블을 구성하여 B ⋈ C에 참여하는 데이터 집합 C의 조인 후보를 선택하면 아래와 같다.

B1'={c7,c9,12} B2'={c11,c12} B3'={Ø} B4'={c3,c4}

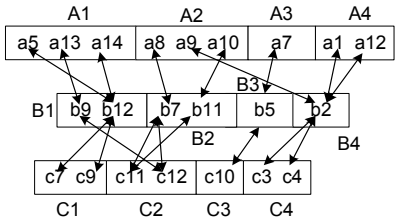
결국, B ⋈ C의 조인 후보로 (b2, c3), (b2, c4), (b5, c10), (b7, c11), (b7, c12), (b9, c12), (b11, c11), (b12, c7), (b12, c9)가 결정된다. 이와 같은 방법으로 세 가지 데이터 집합 이상의 다중 공간 해쉬 조인에 대해서도 버켓을 설정하여 조인후보를 찾고, 이전 단계에서 찾은 조인후보를 다음 조인에 실질적으로 참여하는 후보를 선택하는 과정을 통해 다중 공간 해쉬 조인을 수행하게 된다.

3.2 실질적인 다중 공간 조인

다중 공간 조인 A ⋈ B AND B⋈ C 에 대한 조인 후보는 [그림7]과 같이 얻어지고, [그림8]은

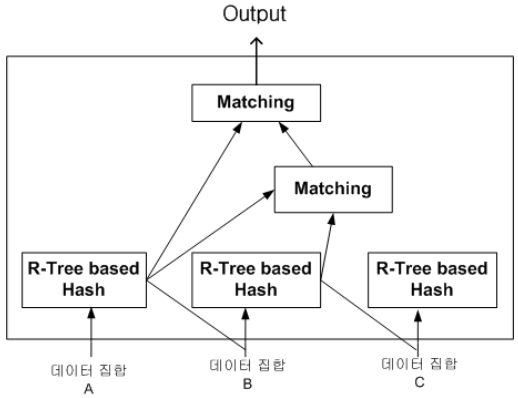
a	B	C
a1	b2	c3
a1	b2	c4
a5	b12	c7
a5	b12	c9
a7	b5	c10
a8	b7	c11
a8	b7	c12
a9	b2	c3
a9	b2	c4
a10	b11	c11
a12	b2	c3
a12	b2	c4
a13	b9	c12
a14	b12	c7
a14	b12	c9

[그림7] 조인 후보 테이블



[그림8] 조인 연계 그래프

각 버켓(A1～A4)에 대하여 병렬적으로 공간 조인을 수행하는 과정을 나타낸 것으로 (a1,b2,c3), (a1, b2, c4)의 다중 공간 조인에서 (a1, b2)와 같이 반복적으로 수행되는 조인에 대해, 단 한번의 조인을 수행하여 다중 공간 조인 방식을 취한다. 이와 같은 해쉬 기반의 다중 공간 조인에 대한 전체적인 과정은 [그림9]와 같이 표현될 수 있다.



[그림9] 해쉬 기반의 다중 공간 조인

A ⋈ B AND B ⋈ C에 대한 R-tree 인덱스를 사용하는 해쉬 기반의 다중 공간 조인은 실질적인 조인 연산시 불필요한 조인 access 제거하여 병렬적으로 최적의 다중 공간 조인을 수행하게 한다.

4. 결론 및 향후 연구

본 논문에서는 고비용의 다중 공간 조인을 효율적인 수행하기 위해 기존 공간 해쉬 조인을 확장하여 해쉬 기반의 다중 공간 조인 기법을 제안하여. 조인에 참여하는 실질적인 조인 후보들을 파악하여 병렬적으로 질의처리를 할 수 있게 하였다.

향후 연구로는 해쉬 조인의 성능에 영향을 주는 자료불균형(data skew)이 다중 공간 해쉬 조인에 미치는 영향과 효율적인 처리를 위한 연구가 필요하다.

참고문헌

[1] T. Brinkhoff, H.P. Kriegel, B. Seeger, "Efficient Processing of Spatial Joins Using R-Trees", Proc. ACM SIGMOD Int. Conf. pp237-246, 1993

[2] M.L. Lo, C.V. Ravishankar, "The Design and Implementation of Seeded Trees : An Efficient Method for Spatial Joins", IEEE Trans. Knowledgement and Data Engineering, Vol 10, No 1, pp136-152, 1998.

[3] T. Brinkhoff, H.P. Kriegel, B.Seeger, "Parallel Processing of Spatial Joins Using R-Trees, Proc. 12th IEEE Data Engineering pp258-265, 1996

[4] Priti Mishra, Margaret H. Eich, "Join Processing in Relational Databases", ACM Computing Surveys, Vol 24, No 1, pp 63-113, March 1992

[5] Goetz Graefe, "Qeury Evaluation Techniques for Large Databases", ACM Computing Surveys, Vol 25, No 2, pp 77-170, June 1993

[6] J.A. Orenstein, "Spatial Query Processing in an Object-Oriented Database System", Proc. of ACM SIGMOD, pp 326-336, 1986

[7] N. Mamoulis and D. Papadias, "Multiway Spatial Joins", ACM Trans. on Database Systems(TODS), Vol 26, No 4, pp 424-475, 2001

[8] D. Papadias, N. Mamoulis and Y. Theodoridis, "Constraint-based Processing of Multiway Spatial Joins", Algorithmica, Vol 30, No 2, pp 188-215, 2001

[9] Jignesh M. Patel, David J. DeWitt, "Partition Based Spatial-Merge Join," SIGMOD Conf., pp 259-270, 1996

[10] Philippe Rigaux, Agnes Voisard, Michel O. Scholl, "Spatial Database: With Application to Gis," Morgan Kaufmann Publishers, 2001