

분산 공간 DBMS에서의 효율적인 공간 릴레이션 분할 기법을 이용한 병렬 공간 조인 기법†

Parallel Spatial Join Method Using Efficient Spatial Relation Partition In Distributed Spatial Database Systems

고주일*, 이환재**, 배해영***

Ju-Il Ko, Hwan-Jae Lee, Hae-Young Bae

요약 분산 공간 데이터베이스 시스템들 사이에서 빈번히 수행되는 공간 조인 질의는 공간 데이터의 대용량성과 그 복잡성으로 인하여 공간 연산 수행 시 서버에 CPU 및 디스크 I/O 상의 부하를 일으킨다. 본 논문은 이러한 분산 공간 데이터베이스 시스템에서 수행 비용이 많이 드는 원격 사이트간의 공간 조인 질의를 병렬적으로 수행하는 기법을 제안한다. 본 기법은 조인에 참여하는 릴레이션들 중 하나를 이등분 하는 방법으로 공간 조인 연산을 분리한 후, 질의 수행에 참여하는 두 서버에게 조인 연산을 분배한다. 각 서버는 분할된 공간 조인 연산을 동시에 연쇄적으로 처리하고 결과를 병합하여 최종 조인 결과를 생성한다. 본 기법은 릴레이션을 효율적으로 분할하여 조인을 수행함으로써 공간 연산에 참여하는 객체의 수를 절반으로 줄이며 R-Tree 등 공간 인덱스의 탐색 횟수와 그 범위를 감소시킨다. 또한 릴레이션을 영역단위로 분할하여 객체의 수를 줄이고 참여 객체를 군집화 시킴으로써 조인 연산시에 디스크와 버퍼의 사용 효율을 높인다.

ABSTRACT In distributed spatial database systems, users may issue a query that joins two relations stored at different sites. The sheer volume and complexity of spatial data bring out expensive CPU and I/O costs during the spatial join processing. This paper shows a new spatial join method which joins two spatial relation in a parallel way. Firstly, the initial join operation is divided into two distinct ones by partitioning one of two participating relations based on the region. This two join operations are assigned to each sites and executed simultaneously. Finally, each intermediate result sets from the two join operations are merged to an ultimate result set. This method reduces the number of spatial objects participating in the spatial operations. It also reduces the scope and the number of scanning spatial indices. And it does not materialize the temporary results by implementing the join algebra operators using the iterator. The performance test shows that this join method can lead to efficient use in terms of buffer and disk by narrowing down the joining region and decreasing the number of spatial objects.

키워드 : 정보 통신, 공간 정보, GIS, 데이터베이스

1. 서론

분산 공간 데이터베이스 시스템에서 조인 조건에 공간 연산이 포함된 분산 공간 조인 질의는 공간 데이터의 대용량성과 그 복잡성으로 인하여 조인의 조건이

되는 공간 연산을 수행하는 서버에서 CPU 및 I/O상의 지역적 부하와 네트워크상의 전송 부하를 발생시켜 질의 처리시간을 증가 시킨다.

이런 문제들을 해결하기 위해, 기존의 문자 및 숫자 기반의 일반적인 분산 데이터베이스 시스템에서 사용

† 본 연구는 정보통신부의 대학 S/W 연구센터 지원사업의 연구 결과임

* 인하대학교 컴퓨터공학과 석사과정

ilovjc@hanmail.net

** (주)알티베이스 연구원

xcom73@freechal.com

*** 인하대학교 컴퓨터공학과 교수

hybae@inha.ac.kr

되는 세미조인[1, 2]을 기반으로 한 여러 가지 공간 조인 기법들과 질의 최적화 기법들이 제안되었다. 이 기법들은 데이터 전송량을 감소시켜 네트워크의 부하를 줄이며 공간 연산을 여과 단계와 정제 단계로 나누어 서로 다른 서버에서 수행함으로써 공간 연산 부하를 다소 분산시켰다는 장점이 있다. 하지만, 공간 연산을 수행하는 부하 가운데 대부분을 차지하는 정제 연산이 특정 서버에 집중되어 병목 현상을 발생시켜 전체적인 질의 처리시간이 지연되는 단점이 있다.

이러한 문제를 해결하기 위해, 본 논문은 이러한 분산 환경의 대용량 공간 데이터베이스 시스템에서 수행 비용이 많이 드는 원격 사이트간 공간 조인 질의를 병렬적으로 수행하는 기법을 제안한다. 본 기법은 효율적 분할 기법에 따라, 먼저 공간 조인의 대상이 릴레이션들중에서 어떤 릴레이션을 분할할 것인지를 결정한다. 그리고 분할하기로 결정된 릴레이션을 균등한 수의 공간 객체가 포함 되도록 나누어 분할한 후 질의 수행에 참여하는 두 서버에 분배하여 병렬로 수행시킴으로써 질의 처리를 향상시킨다.

본 기법은 비용이 많이 드는 두 기본 릴레이션(Base Relation)들 간의 공간 조인 연산 수행 시 입력되는 객체의 수를 반으로 줄여 공간 연산에 의한 CPU 부하와 R-Tree[3, 4, 5]등의 공간 인덱스 탐색시 소요되는 디스크 I/O 비용을 줄여 전체적인 질의 수행시간을 단축시킨다.

2. 관련 연구

2.1 분산 질의 최적화 기법

기존의 일반적인 관계형 분산 데이터베이스를 위한 분산 질의 최적화 기법들은 조인 술어의 평가 비용이 거의 들지 않는 질의들에 대한 최적화 기법이기 때문에, 연산비용 보다는 전송 비용 최소화에 중점을 두고 설계가 된 알고리즘들이다.

하지만 네트워크 기술이 발달됨에 따라 전송부하가 점점 감소되고 있으며, 또한 분산된 다중 사이트간의 공간 조인 질의는 수행 비용의 대부분이 네트워크상의 전송 비용이 아닌 공간 연산 수행 시에 들게 되어 병목현상[6, 7]이 발생하게 되므로, 기존의 기법들을 그대로 공간 조인 질의의 최적화에 적용할 수 없다. 따라서 분산 공간 데이터베이스에 대한 공간 조인 질의의 최적화에는 공간 연산의 수행 비용의 최소화에 중점을 두고 각 사이트간의 연쇄적 질의처리(Pipelining)를 하기 위한 새로운 최적화 기법이 필요하다.

2.2 분산 공간 조인 기법

일반적인 관계형 분산 데이터베이스 시스템에서는 서로 떨어진 두 원격 사이트간의 질의 처리 시에 조인 연산은 세미조인(Semijoin)을 이용하여 수행한다. 세미조인을 이용한 조인 기법을 사용할 경우 튜플의 전체 어트리뷰트를 전송하지 않고 조인에 관련된 어트리뷰트들만 프로젝션하여 전송하므로 데이터 전송량의 감소로 인한 네트워크 부하 감소의 효과가 있다.[1]

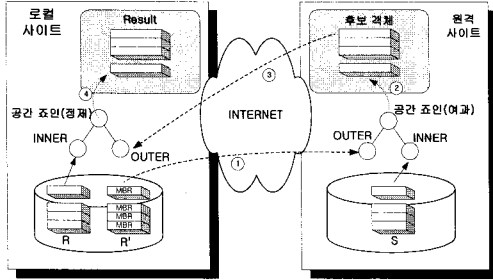
분산 공간 데이터베이스 시스템에서도 기존의 관계형 데이터베이스 시스템에서의 분산 조인 연산과 마찬가지로 세미조인을 이용한 조인 연산의 수행이 가능하다. 하지만 분산 공간 데이터베이스 시스템에서는, 네트워크를 통한 전송 부하가 최대의 걸림돌이던 기존의 관계형 데이터베이스와는 달리, 조인의 조건이 되는 공간 연산의 처리 부하가 전송부하보다 더 많은 비중을 차지한다. 그리고, 릴레이션의 크기가 대용량화 되고 객체가 복잡해 짐에 따라 공간 연산의 처리 부하는 조인 비용의 거의 대부분을 차지하게 된다. 따라서 세미조인 기법을 이용한 공간 조인의 수행 알고리즘은 데이터베이스의 크기에 따라 위의 두 가지 부하의 균형을 맞추어 설계되어야 한다.[6]

세미조인 기반의 분산 공간 조인 기법은 <그림 1>과 같으며, 세미조인 기반의 공간 조인 연산을 처리하는 알고리즘은 다음과 같다.

- 1) 릴레이션 R에서 튜플 한 개를 읽어 공간 객체의 근사 정보를 S의 사이트로 전송한다.
- 2) S의 사이트에서 S와 전송 받은 근사 정보를 이용하여 여과연산을 수행하여 후보 객체 리스트인 S'를 생성한다.
- 3) S'를 R의 사이트로 전송한다.
- 4) R의 사이트에서 읽은 튜플과 S'와의 정제(Refinement) 연산을 수행하여 적합한 객체를 선별하여 조인한다.
- 5) 1)~4)의 과정을 R의 모든 객체에 대하여 수행한다.

위의 알고리즘에 의하면 조인 연산은 릴레이션 R의 각각의 튜플과 조인 가능한 릴레이션 S의 튜플을 찾기 위한 일련의 검색(Selection) 연산으로 처리되며, 조인의 결과는 기존의 관계형 데이터베이스에서의 조인 연산과 동일하게 된다.[8] 위의 알고리즘이 기존의 공간 데이터베이스에서의 세미조인 기반의 조인 기법과 다른 점은 조인의 조건인 공간 연산의 수행을 두 개의 사이트에서 두 번 중복해서 하지 않고, 여과연산과 정제연산으로 나누어 수행한다는 것이다. 하지만, 공간 연산에서 가장 큰 부하로 작용하는 정제연산을 특

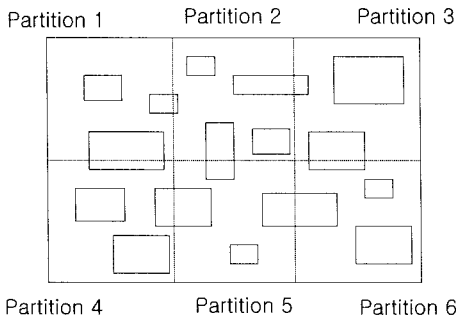
정 서버에서 모두 처리하므로, CPU에 걸리는 부하를 균등하게 분산시키지 못하는 단점이 있다.



〈그림 1〉 세미조인 기반의 분산 공간 조인

2.2 분할 기반 공간 병합 조인

분할 기반 공간 병합 조인(PBSM Join : Partition Based Spatial-Merge Join) 기법은 기존에 단일 서버 내에서의 병렬 조인 기법으로 제시된 병렬 버전의 정렬 병합(Sort-Merge) 계열의 조인 기법과 비슷하지만, 분할의 대상이 공간 릴레이션이라는 점과 릴레이션의 분할을 선행적인 값으로 하지 않고 이차원인 영역의 분할을 이용한다는 것이 다르다. PBSM 조인 기법은 일반적인 공간 연산의 수행단계와 같은 여파 단계와 정제 단계로 나누어 수행된다.



〈그림 2-2〉 분할 기반 공간 병합 조인 기법에서의 공간의 분할

PBSM 조인 기법의 여파 단계에서는 공간 릴레이션의 전체 영역을 버퍼의 메모리 크기에 의거하여 일정한 크기로 분할(Partitioning)하여 각 분할 영역에 MBR이 겹치는 객체들의 OID들의 리스트를 작성한다. 정제 단계에서는 두 릴레이션의 각 분할 영역들 사이에서 평면 스위핑(Plane Sweeping) 알고리즘의 순서로 조인을 수행하여 객체끼리의 정제연산을 하며,

결과로는 두 튜플의 OID 쌍들의 집합을 생성한다. 마지막으로 결과로 나온 OID의 쌍들에는 중복된 쌍이 있을 수 있으므로, OID쌍의 정렬하는 등의 방법을 통해 중복된 쌍을 제거한다.

PBSM 조인 기법은 비용이 많이 드는 공간 조인 연산을 병렬로 수행할 수 있어 수행 시간을 단축시킬 수 있다. 하지만, 조인의 수행 전에 각 분할 영역에 포함되는 OID들의 리스트를 작성해야 하므로 초기 응답시간이 오래 걸리고, 중복된 OID 쌍을 제거해야 하는 추가 부담도 있으며, 지역적으로 편중된(Skewed) 객체들을 처리하는 것에 대해서는 특별한 대안이 없는 단점이 있다.

3. 효율적인 공간 릴레이션 분할 기법을 이용한 병렬 공간 조인 기법

본 장에서는 둘 이상의 원격 사이트에 분산되어있는 공간 데이터베이스에 대한 질의 수행 시 포함되는 분산 공간 조인 질의를 효율적으로 수행할 수 있는 병렬 공간 조인 기법에 대하여 설명한다. 그리고, 본 기법을 수행하기 위한 분할 대상 릴레이션 선택 및 릴레이션 분할 알고리즘을 설명한다.

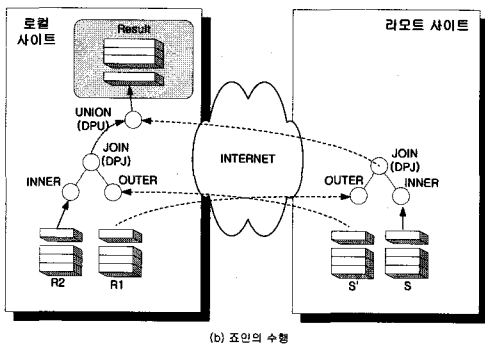
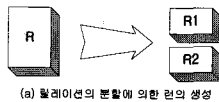
3.1 원격 사이트간의 병렬 공간 조인

앞서 관련 연구에서 살펴본바와 같이 분산 공간 데이터베이스 시스템에서는 원격 사이트간의 공간 조인시에 네트워크 부하보다는 공간 연산 즉 정제 연산의 부하가 더욱 크다는 것을 살펴 보았다. 그렇다면 정제 연산의 부하를 분산시킨다면 보다 효과적인 공간 조인 수행이 가능하다는 것을 알 수 있다. 따라서 본 논문에서는 이런 공간 연산중 정제 연산을 두 사이트가 병렬로 수행하게 함으로써 연산의 부하를 줄이는 병렬 공간 조인 방법을 제안한다.

병렬 공간 조인 기법은 크게 분석 단계, 그리고 질의 실행 단계의 2단계로 나누어 수행되며, 전체적인 수행 방법은 〈그림 2〉와 같다. 이 기법의 분석 단계에서는 조인에 참여하는 두 개의 릴레이션들 중 하나를 선택하여 릴레이션의 전체 영역을 2개로 분할한다. 질의 실행 단계에서는 각 서버에 한 개의 영역을 할당하여 각 영역에 속하는 튜플들과 영역이 분리되지 않은 릴레이션과의 조인을 동시에 수행한 후, 두 사이트에서 생성되는 조인의 결과 튜플들을 병합하여 최종 결과를 생성하여 클라이언트에게 전송한다.

질의 실행 단계에서는 새로운 대수 연산자를 사용하게 되는데, 이중 DPJ(Distributed Pipelined Join)

는 분산 환경에서 원격 사이트로부터 릴레이션을 입력 받아 자신의 사이트에 있는 릴레이션과 조인을 수행하는 노드이며, DPU(Distributed Pipelined Union)는 DPJ와 마찬가지로 분산 환경에서 자신의 사이트에서 입력되는 결과 릴레이션과 원격 사이트에서 입력되는 결과 릴레이션을 UNION연산하여 병합하는 대수 노드이다.[8, 9]



〈그림 2〉 병렬 공간 조인

〈그림 2〉에서 설명하는 병렬 공간 조인 기법은 〈그림 1〉의 세미조인 기반의 분산 공간 조인 기법과 비교하여 조인의 조건인 공간 연산 중 비용이 많이 드는 정제 연산의 수행을 두 서버에서 나누어 수행함으로써 각 서버에 CPU 비용을 분담시키며, 영역 분할을 통해 릴레이션을 분리한 후 조인을 수행함으로써 공간 연산 중 여과 단계에서 인덱스 탐색 범위를 줄여 디스크 및 버퍼에 소요되는 비용을 절감할 수 있다.

3.2 병렬 조인을 위한 효율적인 릴레이션의 분할

병렬 공간 조인 기법의 분석 단계에서는 릴레이션을 분할하여 조인 연산을 병렬로 처리할 수 있도록 하는데, 분할될 릴레이션 및 분할될 영역의 기준점을 잘못 선택할 경우 어느 한쪽 조인에 더 많은 부하가 편중되어 균등하게 병렬 조인을 할 수 없다. 본 장에서는 릴레이션을 분할할 때 두 개의 릴레이션 중 분할될 릴레이션을 선택하는 기준에 대한 설명을 하고, 선택된 릴레이션의 전체 영역을 두 부분으로 분할하는 기준점을 정하는 방법에 대하여 설명한다.

3.3.1 대상 릴레이션의 결정

원격 사이트에 있는 두 릴레이션을 공간 조인 할 때는 객체를 전송하는 네트워크상의 전송 부하와 각 서버에서 공간 조인 연산을 수행하는 CPU 및 디스크 I/O 부하가 큰 문제가 된다. 따라서 본 기법에서는 조인되는 두 개의 릴레이션 중에서 분할시킬 릴레이션을 결정할 때, 전송부하와 CPU 및 디스크의 부하를 최소한으로 줄이기 위하여 각 릴레이션의 전체 크기와 분할된 영역에 포함될 객체의 수를 고려한다.

릴레이션의 전체 크기는 공간 정보 관리 서버에서 얻어온 메타 정보 중에서 튜플의 평균 크기와 릴레이션 내의 튜플의 개수(Cardinality)를 이용하여 대략적으로 구할 수 있다. 두 릴레이션의 크기 관계는 다음과 같은 3가지의 경우가 있을 수 있다.

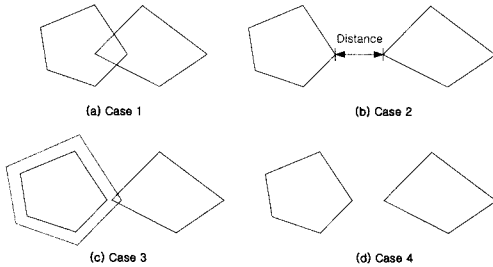
- A. $V(R) \gg V(S)$
- B. $V(R) \approx V(S)$
- C. $V(R) \ll V(S)$

A번과 C번의 경우는 두 릴레이션의 크기에 많은 차이가 있는 경우이며, B번의 경우는 두 릴레이션의 크기가 서로 비슷한 경우이다. 이렇게 릴레이션의 크기를 구하면 〈그림 2〉의 R1 및 R2의 크기는 구할 수 있으나, S'는 R2의 영역의 크기와 조인이 조건이 되는 공간 연산의 종류, 그리고 S의 전체 크기에 의해 정해지므로 S'의 전송량을 구하기 위해서는 조인 조건인 공간 연산의 종류를 고려해야 한다.

서로 다른 공간 데이터베이스 시스템사이의 공간 정보 유통을 위해 OGC(개방형 지리정보체계 협회)에서 공간 연산과 공간 객체 타입 등의 표준을 제정한 OGIS(The Open Geodata Interoperability Specification)에 정의된 공간 함수들[10] 중에서 공간 조인의 조건으로 사용될 수 있는 것은 두 객체간의 위상 관계를 나타내는 위상 연산자와 공간 분석 함수 중 거리(Distance) 연산 등의 수치 비교연산이 있을 수 있다. 이들을 조합하면 다양한 공간 조인 조건이 생성될 수 있지만 그 중 비슷한 성격을 가진 조건들을 제거하고 그 종류를 분류하면, 공간 릴레이션 R과 S가 공간 조인이 될 조건으로는 대표적으로 〈그림 3〉과 같은 4가지 종류의 것이 있을 수 있다.

위의 네가지 예 중에서 ①번은 R을 영역을 기준으로 R1과 R2로 분할할 경우, R2와 조인이 가능한 S의 대상 객체의 집합인 S'의 영역 역시 R2과 비슷하거나 약간 크다고 가정할 수 있다. ②, ③번의 조건은 조인 가능한 객체들이 서로 떨어져 있어도 후보가 되는 경우 이므로 R2의 영역보다 후보인 S'의 영역이

더 넓으며, ④번의 경우에 S'의 영역은 S의 영역 전체가 된다. 따라서 S'의 영역은 R2의 영역을 포함하는 더 넓은 영역을 가지게 되어, 두 릴레이션의 객체들이 균등하게 분포되어 있다고 가정할 때, R2보다 S'에 더 많은 객체들이 포함되게 된다.



〈그림 3〉 객체들의 위상 관계

조인의 수행 시 전송되어온 객체의 수는 곧 R-Tree와 같은 공간 인덱스에 대한 검색 횟수와 같은데, 이는 상당히 많은 CPU 및 디스크 I/O 부하를 일으키는 요소가 된다. 따라서 본 기법에서는 조인되는 두 릴레이션의 크기가 비슷할 경우 튜플의 개수가 더 많은 릴레이션을 분할하여 공간 인덱스에 대한 검색 연산의 횟수를 줄인다.

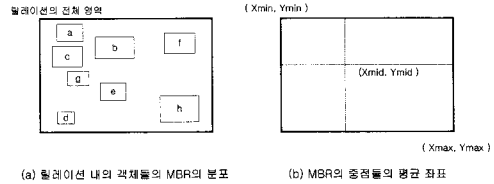
위에 설명한 두 가지 기준에 의하여 조인이 되는 두 개의 릴레이션중 분할할 릴레이션을 선택할 경우, 두 릴레이션의 크기가 많은 차이가 있으면 분할된 릴레이션 쪽이 더 작은 영역을 전송하게 되므로 크기가 더 큰 릴레이션을 분할하여야 하며, 두 릴레이션의 크기가 비슷한 경우는 어느 쪽을 분할해도 전송하는 데이터의 양은 같지만 공간 인덱스의 탐색 연산의 횟수를 줄이기 위하여 튜플의 수가 많은 릴레이션을 분할하여야 한다.

3.3.2 분할 영역의 결정

병렬 공간 조인 기법에서 제안하는 영역 분할 기법은 빠른 응답속도와 연쇄적인 질의 처리를 위하여 계산비용이 적게 들고 중복 튜플이 생성되지 않도록 하는 것을 우선 목표로 한다. 본 영역 분할 기법에서는 다음과 같이 릴레이션이 차지하는 전체 영역을 둘로 분할하여 그 중 하나의 영역에 MBR이 INTERSECT 되는 객체와 그렇지 않은 객체가 나뉜다.

분할의 대상이 되는 릴레이션에 객체가 분포되어 있고 객체들의 MBR들이 〈그림 4 (a)〉와 같이 분포되어 있다고 가정하자. 만일 영역을 X축 혹은 Y축을 따라서 정확하게 2등분을 하여 객체들을 나눌 경우, 두 개의 런 중 하나의 런에 객체들이 편중되는 현상이 발

생하게 된다. 이러한 편중현상을 방지하기 위해서는 영역을 분할할 때 객체들의 분포 상황을 감안해야 한다. 병렬 조인 기법에서는 영역 분할의 기준점으로 모든 객체들의 MBR의 중점(CPM : Central Point of MBRs)의 평균좌표(APC : Average Point Of CPMs)를 사용한다.



〈그림 4〉 릴레이션 내의 분포된 객체 MBR들의 평균좌표

객체의 MBR의 중점을 구하는 함수인 MidX()는 다음과 같은 식으로 간단히 구현된다.

$$MidX(a) = \frac{X_{max} + X_{min}}{2} \dots\dots\dots [수식 2]$$

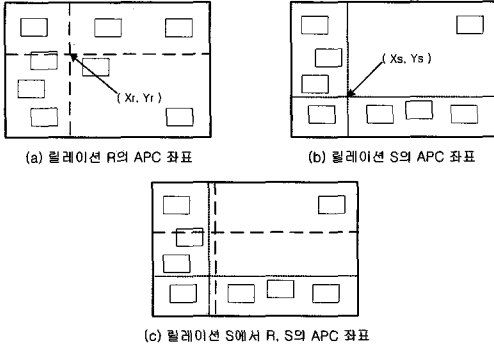
MidY(a) 역시 MidX와 같은 식으로 구할 수 있으며, 이 두 가지 함수를 이용하여 릴레이션의 APC 좌표를 구할 수 있다. 〈그림 4 (b)〉에서 APC 좌표 중 X축 좌표값인 Xmid는 다음과 같은 식으로 구할 수 있다.

$$Xmid = \frac{MidX(Envelope(a)) + MidX(Envelope(b)) + \dots + MidX(Envelope(h))}{Cardinality} \dots\dots\dots [수식 3]$$

APC의 Y축 좌표값인 Ymid 역시 Xmid와 같은 방법으로 구할 수 있으며, APC 좌표의 X축 혹은 Y축을 기준으로 하여 영역을 분할하여 객체의 MidX() 혹은 MidY() 값이 영역에 포함되는지를 체크하여 분할하면 객체들의 수를 균등하게 분할할 수 있다. 하지만 이때 공간 연산이 두 사이트에서 고르게 수행되도록 하기 위해 분할 축을 선택할 필요가 있다.

아래 〈그림 5〉를 보면 분할 대상 릴레이션 R이 X축 또는 Y축으로 어떻게 분할됨에 따라 R1, R2의 공간 연산 후보 객체들의 수가 달라짐을 알 수 있다. Xr 값을 기준으로 Y축(세로 방향)으로 R을 분할할 경우 R1, R2의 공간 조인이 가능한 S의 후보 객체의 수는 비슷하다. 하지만 Yr 값을 기준으로 X축(가로 방향)으로 분할할 경우 R1, R2의 공간 조인 가능한 후보 객체의 수는 특정 분할 영역으로 편중됨을 볼 수 있다. 따라서 위의 예에서는 R1과 R2와 조인되는 후보 객

체수를 비슷하게 하여 병렬 수행도를 높이기 위해서 X_r 값에 따라 세로 방향으로 분할해야한다.



〈그림 5〉 분할 릴레이션 R의 X, Y축 분할에 따른 릴레이션 S의 객체들의 포함 관계

정리하면 어떤 축으로 분할할 것인지를 판단하는 방법은 다음과 같다. 먼저 조인될 두 릴레이션 R과 S의 APC값을 얻은 뒤 두 APC의 X, Y값의 차를 각각 구한다. 그리고 차가 적은 축의 값을 기준으로 분할하면 된다. 예를 들면 〈그림 5 (c)〉에서는 X값의 차이가 Y값의 차이보다 적으므로 X_r 값을 기준으로 릴레이션 R을 분할하기로 선택된다.

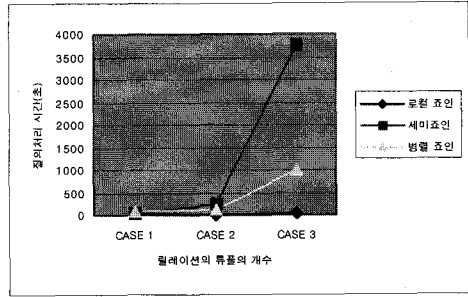
4. 성능평가

본 평가에서 사용된 릴레이션들에 대한 상세 정보 아래의 〈표 1〉과 같다.

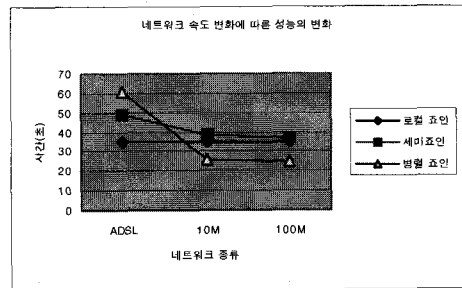
〈표 1〉 조인에 참여한 릴레이션의 상세 정보

	참여 릴레이션	릴레이션의 전체 크기 (Kbyte)	튜플의 총 개수 (개)
CASE 1	산	1900	181
	강	147	325
CASE 2	건물 경계	700	2840
	골목길	800	2703
CASE 3	건물	5800	45232
	도로	5500	33379

네트워크 환경이 ADSL일때 〈표 1〉의 각 경우에 대해 두 릴레이션간의 의 INTERSECT 연산을 수행한 결과는 〈그림 6〉과 같고 네트워크 환경의 변화에 따른 CASE1의 릴레이션에 대한 INTERSECT 연산의 결과는 〈그림 7〉과 같다.



〈그림 6〉 릴레이션의 크기 변화에 따른 성능의 변화



〈그림 7〉 네트워크 환경 변화에 따른 성능의 변화

위의 〈그림 6〉의 결과를 보면 릴레이션의 크기가 커짐에 따라 수행 시간이 증가하는 쪽이 세미조인 기반의 기법보다 병렬 조인 기법의 경우가 더 작은 것을 알 수 있다. 병렬 조인 기법에서의 서버간 데이터 전송량이 세미조인 기반의 기법보다 더 많은 것을 감안하면, 릴레이션의 크기가 커짐에 따라 공간 조인 시의 CPU 및 I/O 부하가 네트워크 부하보다 더 큰 비율로 증가함을 알 수 있다. 〈그림 7〉을 보면, ADSL의 경우처럼 네트워크 환경이 좋지 않은 경우에는 병렬 조인 기법이 세미 조인 기법보다 더욱 많은 수행시간이 걸린다. 네트워크 환경이 좋아질수록 병렬 조인 기법은 세미조인보다 더 나은 성능을 보이며 10Mbps에 가까워지면 로컬 조인 보다도 더 나은 성능을 보인다. 이는 공간 연산 부하를 분산하여 병렬 시스템에서의 효과가 나타나는 것으로 볼 수 있다.

5. 결론

분산된 여러 사이트에서 관리되는 대용량의 공간 데이터베이스를 효율적으로 공유하기 위해서는 분산 공간 데이터베이스 시스템의 사용이 필수적이다. 이러한 분산 공간 데이터베이스 시스템들 사이의 질의에 빈번히 포함되는 공간 조인 질의는 공간 데이터의 대용량성과 그 복잡성으로 인하여 공간 연산 수행시 서버에 CPU 및 디스크 I/O 상의 부하를 일으킨다.

본 기법은 릴레이션을 분할하여 조인을 수행함으로써 공간 연산에 참여하는 객체의 수를 절반으로 줄여 R-Tree 등의 공간 인덱스의 탐색 횟수와 그 범위를 기존의 세미조인 기반 공간조인 기법에 비해 절반으로 감소시켜 디스크와 버퍼의 사용 효율을 높였다. 또한 디스크 I/O 및 메모리 연산 비용을 줄여 사용자에게 결과가 전달되기 시작하는 초기 응답시간을 단축시켰다.

향후 연구 과제로는 분할 영역을 결정하는데 작용하는 각 서버들의 성능 차이, 릴레이션의 복사본(Replica)의 존재 유무 등 다양한 요소들을 감안한 최적의 조인 계획 생성에 관한 연구가 필요하다.

참고 문헌

- [1] M. Tamer zsu, Patrick Valduriez, "Principles of Distributed Database Systems, Prentice Hall, 1991.
- [2] Zhe Li, Kenneth A. Ross, PERF Join: An Alternative To Two-way Semijoin And Bloomjoin, Proc. of Int'l Conf. on Information And Knowledge Management, pp. 137 - 14 , 1995.
- [3] Guttman, A., "R-Trees: An Dynamic Index Structure for Spatial Searching," Proc. Of ACM SIGMOD Int'l Conf. on Management of Data, pp.47-57, 1984.
- [4] T. Sellis, N. Roussopoulos, C. Faloutsos, R+-tree: A Dynamic Index for Multi-Dimensional Objects, Proc. of Int'l Conf. Very Large Data Bases, pp. 16-27, 1996.
- [5] Kriegel, N.B., B. Seeger, "The R*-tree : An Efficient and Robust Access Method for Points and Rectangles," Proc. Of ACM SIGMOD, pp.322-331, 1990.
- [6] Kian Lee Tan, Beng Chin Ooi, David J. Abel, Exploiting Spatial Indexes for Semijoin-Based Join Processing in Distributed Spatial Databases, IEEE Transaction on Knowledge And Data Engeering, Vol. 12, No. 6, pp. 920-937, 2000.
- [7] Ashraf Aboulnaga, Jeffrey F. Naughton, Accurate Estimation of the Cost of Spatial Selections, Proc. of the 16th International Conference on Data Engineering, Feb. Mar., 2000.
- [8] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, Mc. Graw Hill, 1997.
- [9] Hector Garcia-Molina, Jeffry D Ulman, Jennifer Widom, Database System Implementation, Prectice Hall, 2000.
- [10] Donovan A. Schneider, David J. Dewitt, A Performace Evaluation of Four Parallel Join Algorithms in a Shared Nothing Multiprocessor Environment, Proc. of ACM SIGMOD Int'l Conf. on Management of data, Vol. 18, Issue 2, pp. 110-121, June 1989.
- [11] Jignesh M. Patel, David J. Dewitt, Partition Based Spatial-Merge Join, Proc. Of ACM SIGMOD, Vol.25, Issue 2, pp. 259-270, 1996.
- [12] T. Brinkhoff, H. Kriegel, B. Seeger, Efficient Processing of Spatial Joins Using R-Trees, Proc. of ACM SIGMOD Int'l Conf. on Management of data, Vol. 22, Issue 2, pp. 237-246, 1993.



배해영

1976년 인하대학교 응용물리학
공학사

1978년 연세대학교 전자계산학
공학석사

1990년 숭실대학교 전자계산학 공학박사
1982년-현재 인하대학교 컴퓨터공학부 교수
1999년-현재 정보통신부 지정 지능형 GIS 연구센터
센터장

46 개방형GIS학회논문지 제4권 제1호

2001년-현재 건설교통부 국가 GIS 민간자문위원 위
원장
2001년-현재 국무조정실-정보화평가위원회(GIS분야)
위원
2001년-현재 ETRI 컴퓨터소프트웨어기술 연구소 자
문위원



고주일

2002년 인하대학교 컴퓨터공학 졸업
2003년-현재 인하대학교 컴퓨터공학
석사과정



이환재

2000년 인하대학교 컴퓨터공학
졸업
2002년 인하대학교 컴퓨터공학
공학석사
2002년-현재 (주)알티베이스 연구원