비공유 공간 클러스터 환경에서 효율적인 병렬 공간 조인 처리 기법

정 원 일† · 이 충 호†† · 배 해 영†††

요 약

기존의 단일 대용량 데이터베이스 서버에 인터넷 서비스 사용자들이 과도하게 몰릴 경우 서버에 발생하는 네트워크 통신량의 증가와 자원 사용량의 급격한 증가로 인해 서비스 처리 시간의 지연 및 서비스의 중단 현상이 발생할 수 있다. 이러한 문제들을 해결하기 위해 저비용의 여러 단일 노드를 고속의 네트워크로 연결하여 고성능을 제공하는 공간 데이터베이스 클러스터가 대두되었으나, 단일 노드에서 처리할 경우 전체 시스템의 성능을 저하시킬 수 있는 고비용의 공간 조인 연산에 대한 연구가 필요하다. 본 논문에서는 공간 데이터의 특성을 고려한 데이터의 분할과 부분 중복 기법을 사용하는 비공유 공간 데이터베이스 클러스터 환경에서 고비용의 공간 조인 연산을 효율적으로 수행하기 위한 논리적 분할 영역 및 병렬 공간 조인 기법을 제안한다. 제안 기법은 기존의 병렬 공간 조인 기법에서 나타나는 노드간 작업 생성 및 할당 단계가 필요하지 않으며 추가적인 메시지 전송이 발생하지 않으므로 고비용의 공간 조인 질의에 대해 기존의 비공유 구조를 위한 병렬 R-tree 공간 조인 기법보다 23%의 성능향상을 보인다. 또한, 각 클러스터 노드에서의 중복 정제(Refinement) 연산을 제거하므로 사용자에게 빠른 응답을 제공한다.

Efficient Parallel Spatial Join Processing Method in a Shared-Nothing Database Cluster System

Warnill Chung† · Chung-Ho Lee†† · Hae-Young Bae†††

ABSTRACT

Delay and discontinuance phenomenon of service are cause by sudden increase of the network communication amount and the quantity consumed of resources when Internet users are driven excessively to a conventional single large database server. To solve these problems, spatial database cluster consisted of several single nodes on high-speed network to offer high-performance is risen. But, research about spatial join operation that can reduce the performance of whole system in case process at single node is not achieved. So, in this paper, we propose efficient parallel spatial join processing method in a spatial database cluster system that uses data partitions and replications method that considers the characteristics of space data. Since proposed method does not need the creation step and the assignment step of tasks, and does not occur additional message transmission between cluster nodes that appear in existent parallel spatial join method, it shows performance improvement of 23% than the conventional parallel R-tree spatial join for a shared-nothing architecture about expensive spatial join queries. Also, It can minimize the response time to user because it removes redundant refinement operation at each cluster node.

키워드 : 공간 데이터베이스 클러스터(Spatial Database Cluster), 병렬 공간 조인(Parallel Spatial Join), 공간 색인(Spatial Index)

1. 서 론

최근 인터넷과 무선 통신 환경의 급속한 발전을 기반으로 웹 및 무선 단말기를 통한 다양한 지리정보 서비스들이 제공되고 있으며, 이러한 서비스를 제공 받으려는 사용자들 또한 지속적으로 증가하고 있다. 인터넷 환경의 서비스 이용자들은 특정 집단으로 분류하기가 힘들며, 이용자의 수 또한 몇 만에서 몇 백만에까지 이르기 때문에 기존의 단일 대용량 데이터베이스 서버에 인터넷 서비스 사용자들이 과도하게 몰릴 경우, 서버에 발생하는 네트워크 트래픽의 증가와 서버 자원 사용량의 급격한 증가로 인해 서비스 처리 시간의 지연 및 서비스의 중단 현상이 발생할 수도 있다[1]. 비공간 데이터를 제공하는 웹 서버의 경우, 이러한 문제들을 해결하기 위해 슈퍼컴퓨터와 같은 고가의 대형 컴퓨터를 사용하는 대신 저가의 여러 단일 노드들을 고속의 네트워크로 연결하여 고성능을 제공하는 클러스터 시스템이 대두되어 많은 연구가 진행되었으나, 공간 데이터를 위한 비공유 데이터베이스 클러스터 시스템에 관한 연구는 거의 이루어지지 않고 있다. 공간 데이터베이스 클러스터에

† 준 회 원 : 인하대학교 대학원 전자계산공학과
†† 준 회 원 : 인하대학교 지능형 GIS 연구센터 연구원
††† 종신회원 : 인하대학교 컴퓨터공학과 교수
논문접수 : 2003년 3월 18일, 심사완료 : 2003년 7월 21일

서 관리되어지는 공간 데이터는 비공간 데이터에 대해 다음과 같은 차이가 있다.

첫째 : 지속적인 갱신 연산이 발생하는 비공간 데이터와 달리 공간 데이터는 점 질의나 영역 질의등 선택 연산 위주의 서비스가 제공되는 정적인 특성을 갖는다[2].

둘째 : 공간 릴레이션은 복잡한 공간 데이터들로 이루어진 대용량의 특성을 지니며 인접한 영역의 공간 데이터에 대한 질의가 빈번하게 발생하는 지역적 인접성이 높다[1].

셋째 : 공간 데이터는 비공간 데이터에 비해 CPU 연산 비용과 디스크 입출력 비용이 훨씬 크며 특히, 조인 질의 수행의 경우에는 CPU 연산 비용이 디스크 입출력 비용보다 더 크다는 특징이 있다[3-5].

위와 같이 정적이며 지역적 인접성이 높고 고비용의 CPU 연산이 요구되는 특성을 가진 대용량의 공간 데이터에 기존의 비공간 데이터베이스 클러스터 시스템에서 사용하는 데이터의 분할 방법과 중복 방법을 동일하게 적용시키는 것은 매우 비효율적이므로 본 논문에서는 공간 데이터의 특성을 고려한 지역 단위의 데이터 분할 방법과 중복 방법을 이용한다. 이러한 공간 데이터베이스 클러스터 시스템을 이용하게 될 경우 웹 기반 사용자들의 대규모 공간 질의를 클러스터 내의 여러 노드로 분산시켜 빠르고 안정적으로 처리할 수 있으나, 고비용의 공간 조인 질의를 클러스터 내의 한 노드에서 단독으로 처리하게 할 경우 해당 노드에 부하가 집중되는 병목 현상이 발생하여 결국 전체 클러스터 시스템의 성능이 저하된다. 따라서, 본 논문에서는 지역 단위의 공간 데이터 분할 방법과 중복 방법을 이용하는 공간 데이터베이스 클러스터 환경에서 고비용의 공간 조인 연산을 효율적으로 수행하기 위한 병렬 공간 조인 기법을 제안한다.

제안 기법의 기반이 되는 공간 데이터베이스 클러스터 환경은 공간 데이터들을 일정한 지역 단위의 공간 릴레이션들로 분할하여 각 클러스터 노드에서 관리하도록 하며, 질의 요구가 빈번한 지역 릴레이션들만 다른 노드에 중복시켜 부하를 분산시키는 부분 중복 방법을 이용한다. 고가용성을 위해 새로운 클러스터 노드를 추가할 경우, 부하를 유발시키는 지역 릴레이션들에 대해서만 중복시킴으로써 고확장성을 지원한다. 또 모든 지역 단위의 릴레이션들에 대해 중복된 개수만큼 MBR 기반의 논리적인 분할 영역을 설정하여 중복 릴레이션들이 저장된 각 클러스터 노드들에게 할당하고 전역 메타 정보를 이용하여 관리하도록 한다. 사용자로부터 요구된 공간 조인 질의는 조인 대상 릴레이션들을 관리하는 모든 클러스터 노드에서 병렬적으로 수행되며, 각 노드는 자신에게 할당된 논리적 분할 영역의 공간 객체들에 대해서만 공간 조인 연산을 수행한다. 각 노드들에서 수행한 공간 조인 연산의 결과 튜플들은 파이프라인 방식[4, 15]으로 질의 처리 노드로 전송되며, 질의 처리 노드는 다른 노드들로부터 전송받은 질의 결과들에 대해 별도의 중복 결과 제거 작업 없이 사용자에게 전송한다. 제안된 공간 데이터베이스 클러스터에서의 병렬 공간 조인 기법은 조인에 참여하는 클러스터 노드마다 자신에게 할당된 논리적 분할 영역 내의 공간 객체들에 대해서만 공간 조인을 수행하므로 기존의 병렬 공간 데이터베이스 시스템에서 병렬 공간 조인 수행시 필요로 했던 각 노드간 작업 생성 및 할당 단계가 필요하지 않으며, 병렬 공간 조인 연산을 수행하는 중에 작업 분배를 위한 클러스터 노드간 메시지 전송 비용 등이 전혀 들지 않으므로 고비용의 공간 조인 질의를 빠르게 처리한다. 지역적 인접성을 고려한 논리적 분할 영역의 할당은 공간 조인 연산시 노드간 동일 객체에 대한 중복 디스크 I/O를 발생시키지 않으므로 질의 처리 시간을 단축시킨다. 또한, 공간 조인 연산의 여과 단계에서 논리적 분할 영역 경계 상의 공간 객체에 대해 객체 MBR 중점의 위치에 따른 여과 기법을 사용하여 고비용이 요구되는 정제 연산을 여러 노드에서 중복 수행하는 것을 방지하며, 병렬 공간 조인 결과들에 대해 중복된 결과들을 제거하기 위한 합집합(Union) 연산의 추가적인 작업을 수행하지 않고 해당 결과들을 곧바로 질의를 요청한 사용자에게 전송하므로 빠른 사용자 응답시간을 만족시킨다. 본 논문에서 기반으로 하고 있는 공간 데이터베이스 클러스터의 하드웨어 플랫폼은 여러 독립된 워크스테이션을 클러스터로 구성하여 대용량의 데이터들을 효율적이고 안정적으로 서비스하기 위해 폭 넓게 사용되고 있는 비공유(Shared-Nothing) 구조의 클러스터로 한다[6].

본 논문의 내용 구성은 다음과 같다. 2장에서 관련 연구 내용으로 비공간 데이터베이스 클러스터에서의 데이터 분할 및 중복 방법과 기존에 연구되어진 병렬 공간 조인 기법들에 대해 설명한다. 3장에서는 본 기법의 바탕이 되는 공간 데이터베이스 클러스터의 환경에 대해 설명하고, 논리적 분할 영역 및 병렬 공간 조인 기법에 대해 설명한다. 마지막으로 4장에서 성능평가를 하며, 5장에서 결론 및 향후 연구에 대해 설명한다.

2. 관련 연구

2.1 비공유 데이터베이스 클러스터에서의 데이터 분할 및 중복

데이터베이스 클러스터 시스템은 구조에 따라 공유 디스크 구조, 공유 메모리 구조, 비공유 구조로 분류할 수 있으며 이들 중 클러스터 시스템의 확장과 병렬 처리가 용이한 비공유 구조의 데이터베이스 클러스터가 폭 넓게 사용되고 있다[6]. 이러한 비공유 데이터베이스 클러스터에서는 대규모의 사용자 질의를 효율적으로 처리하기 위해 여러 노드 사이에 데이터를 분할하고 각 노드가 자신이 가지고 있는

분할된 데이터에 대한 질의를 수행하도록 하는 방법을 이용하고 있다. 일반적으로 데이터에 대한 분할 방식은 수평 분할, 수직 분할, 조합 분할로 나눌 수 있다. 이들 분할 기법의 분할 기법의 성능은 입력되는 질의의 종류에 따라 달라질 수 있으므로 어떤 분할 기법이 더 우수하다고 말할 수 없다. 그러나 데이터 분할 방식이 간단하고 결과 집합을 얻기 위한 조합 비용이 적게 드는 수평 분할 방식이 대부분의 클러스터 시스템에서 사용되고 있다[7].

수평 분할의 종류는 레코드 키 값에 따라 순차적으로 데이터를 분할하는 라운드-로빈(Round-robin) 분할 기법, 해쉬(Hash) 함수를 통한 값으로 데이터를 분할하는 해쉬 분할 기법, 키 값의 범위에 따라 데이터를 분할하는 범위 분할 기법 그리고 주어진 조건식에 따라 데이터를 분할하는 조건식 분할 기법이 있다[6]. 그러나 이러한 분할 기법에서는 데이터베이스 클러스터의 성능 향상에 한계가 있는데 그 이유는 다음과 같다.

첫째 : 여러 노드사이에 작업 부하가 고르게 되도록 데이터를 분할하기가 어렵다.
둘째 : 트랜잭션이 여러 분할을 접근할 경우 분산 트랜잭션을 관리해야 한다.
셋째 : 한 노드에서 발생한 고장으로 해당 데이터 분할을 접근할 수 없는 경우가 발생한다.

따라서, 분할된 데이터에 대해 적절한 데이터 중복이 요구되어진다[7]. 데이터의 중복 방법은 완전 중복과 부분 중복으로 분류된다. 완전 중복 방법은 데이터의 중복이 클러스터의 모든 노드에 저장되는 방식이고 부분 중복 방법은 데이터의 중복이 특정 노드들에만 저장되는 방식이다. 완전 중복 방법을 사용하는 경우 모든 노드에 동일한 데이터가 저장되므로 질의 처리에 대한 가용성이 향상되지만 갱신 트랜잭션의 처리시 모든 중복 데이터들에 대한 일관성을 유지시켜 주어야 하며, 가용성을 높이기 위한 새로운 클러스터 노드의 확장시 기존 노드의 모든 데이터를 전송해야 한다는 문제점이 있다. 이에 반해 부분 중복 방법은 가용성은 떨어지지만 중복 데이터들에 대한 일관성을 유지시키기가 용이하며 확장시 부하가 많이 발생하는 데이터들에 대해서만 중복시키면 되므로 확장 비용이 적다는 장점이 있다. 따라서 데이터베이스 클러스터가 처리해야 하는 질의가 읽기 위주의 트랜잭션들로 구성된 경우는 완전 중복 방법을 이용하여 시스템의 가용성을 높이도록 하며, 갱신 트랜잭션이 빈번하게 발생되는 경우는 부분 중복 방법을 이용하여 일관성을 유지시키기 위한 비용을 최소화한다.

2.2 병렬 공간 조인 기법

공간 데이터베이스에서 공간 조인은 두 개의 공간 데이터 집합에 대해 포함, 교차, 겹침 등의 특정 공간 조건을 만족하는 객체 쌍의 집합을 찾아내는 연산이며 두 데이터 집합에 대한 다중 주사를 요구하므로 객체의 수가 증가함에 따라 연산 시간이 급격히 증가한다는 특징을 갖는다[3, 4, 8]. 이러한 공간조인에 관한 기존의 연구는 대부분 단일 프로세서를 이용한 공간 조인이 연구되어 왔으며, 실제 공간조인의 성능은 많은 향상을 가져왔다. 그러나 이러한 연구에도 불구하고, 단일 프로세서에서의 공간조인은 빠른 질의응답을 요구하는 사용자를 만족시키지 못하고 있다. 이것은 공간 조인의 대상이 되는 객체의 수가 많고, 개별 객체의 연산이 복잡하며, 단위 연산시간이 많이 걸리는 작업이기 때문이다[9]. 이러한 단일 프로세서에서의 한계를 극복하기 위해 병렬 공간 데이터베이스 시스템 환경 하에서 다중 프로세서를 이용한 병렬 공간 조인[4, 9-12]에 대한 많은 연구가 진행되어 왔으며, 분산 공간 데이터베이스 시스템 환경에서 두 개의 노드가 공간 조인 연산을 병렬로 수행하도록 하는 분산 병렬 공간 조인에 대해서도 연구가 되었다[2]. 공간조인의 특징은 대상 객체가 많고 객체 쌍의 연산에 있어서 정제 단계의 비용이 많이 소요되지만 각 객체 쌍들에 대한 연산을 분리하여 수행할 수 있는 독립적인 연산이다[12]. 또한, 기존의 단일 프로세서를 이용한 연구에서의 연구성과에 의하면, 여과 단계와 정제 단계를 구분하여 수행하는 것이 성능상의 이점을 제공한다는 특징이 있다[3, 8]. 이러한 객체 쌍들의 연산에서 의존성이 없는 점과 다단계로 수행할 수 있는 점은 병렬화 측면에서 매우 유리한 특징이라고 할 수 있다.

2.2.1 분산 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

관련연구 [2]에서는 분산 공간 데이터베이스 시스템에서의 병렬 공간 조인 수행시 사이트간 공간 조인 질의를 병렬적이며 연쇄적으로 수행하는 기법을 제안하였다. 이 기법은 최하위 사이트 간의 공간 조인 연산에 대해 공간 조인의 대상이 되는 영역을 균등한 수의 공간 객체가 포함 되도록 대상 릴레이션을 분할한 후 질의 수행에 참여하는 두 서버에 분배하여 병렬로 수행시켜 질의 처리를 향상시킨다. 그러나, 이 논문에서 제시하는 기법은 인터넷 환경을 기반으로 하여 공간 조인 수행 비용 뿐만 아니라 네트워크 상의 공간 데이터 전송 비용을 고려하지 않으며, 최하위 두 사이트 노드에서만 병렬 공간 조인을 수행하는 한계를 갖는다.

2.2.2 병렬 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

관련연구 [4]에서는 단일 할당 공간 색인 방식인 R*트리에서 공유 가상 메모리를 가지는 병렬 시스템에서의 공간 조인을 위한 방법들을 제시하고 있다. 이 논문에서는 모든 프로세서에서 접근 가능한 공유 가상 메모리를 사용하여 디스크로부터 메모리로 가져온 공간 객체를 공유함으로써 디

스크 입출력을 감소시킬 수 있으나, 일반적으로 사용되는 비공유 메모리 시스템에서는 성능저하의 요인이 된다. 또한, 노드를 서로 다른 프로세서에 할당함으로써 공간 객체의 디스크 입출력시에 병목현상이 발생하는 문제점이 있다[9].

관련연구 [10]에서는 단일 할당 공간 색인을 이용한 것으로서 비겹침-정규 데이터 공간 분할 방식을 사용하는 Quad 트리에서 병렬 공간 조인을 위한 방법에 대해서 다루고 있다. 이 연구에서는 SIMD(Single Instruction Multiple Data) array processor 환경에서 병렬 공간 조인 방법을 제시하고 있다. 그러나 SIMD는 특수 목적 응용에 주로 사용되는 시스템이며 또한 이 방법은 노드의 병합 과정에서 병합된 사분면들의 데이터들을 주기억장치 내에 적재하여 처리하므로 극단적인 경우에는 많은 병합 과정으로 인해 여러 번의 페이지 교체가 일어나 전체적인 수행속도가 느려지며, 공간 분할 경계상에 존재하는 객체들에 대한 중복 정제 연산 및 중복 결과 병합의 문제점을 갖고 있다[2].

한편, 관련 연구 [9]에서는 데이터 집합 공간을 일정한 면적을 갖도록 X, Y 좌표축으로 등분한 후 단일 또는 다중 할당 고정 그리드를 공간 색인으로 하여 태스크를 생성한다. 이렇게 생성된 태스크들에 대해 먼저 여과 연산을 병렬적으로 수행하여 후보 객체 쌍들로부터 중복을 제거한 후 다시 병렬적으로 정제 연산을 수행함으로써 중복 정제 연산을 제거했지만 여과와 정제 연산 각각에 대해 태스크 생성 및 할당 과정을 수행해야 하는 단점이 있다.

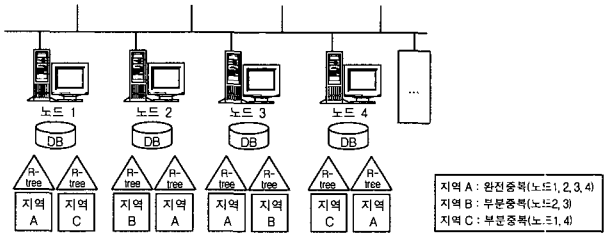
기존의 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법은 프로세서들간의 작업 분배의 문제, 디스크 병목 현상 및 중복 결과 제거에 대한 문제들을 발생시킨다. 이러한 문제들로 인해 병렬 공간 조인 수행시에 발생하는 추가적인 연산 비용은 본 논문에서 제안한 비공유 공간 데이터베이스 클러스터 환경에서의 병렬 공간 조인 기법에서 발생하지 않는다.

3. 비공유 공간 데이터베이스 클러스터에서의 병렬 공간 조인 기법

본 장에서는 병렬 공간 조인 기법의 기반이 되는 비공유 공간 데이터베이스 클러스터의 환경과 제안 기법을 수행하기 위한 확장된 메타 정보들에 대하여 기술한다. 또한 중복된 공간 릴레이션들에 대한 논리적 분할 영역과 이를 이용한 병렬 공간 조인 기법에 대해 기술한다.

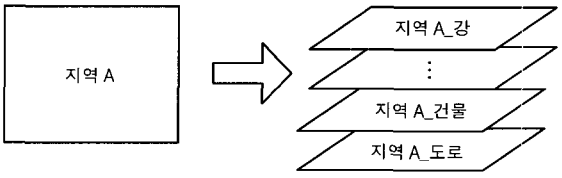
3.1 비공유 공간 데이터베이스 클러스터 환경

비공유 공간 데이터베이스 클러스터는 일반적으로 대규모의 사용자 질의를 처리해야 하는 웹 기반의 GIS 정보시스템의 후위 시스템으로 활용되고 있다. 웹 GIS 클러스터링 시스템은 웹 환경의 특성상 수많은 사용자들로부터 발생되는 대량의 트랜잭션들을 빠르게 처리하기 위해 사용자에게 요청받은 질의를 최대한 지역 노드에서 수행해야 한다[1]. 이러한 환경에서는 모든 클러스터 노드들이 동일한 공간 데이터를 가지고 있을 경우 고성능성에 있어서 최적이라고 할 수 있으나 대용량의 공간 데이터를 모든 노드에서 동일하게 저장, 관리하는 완전 중복 방법을 이용할 경우 저장 공간의 효율성에서 크게 떨어지며, 새로운 노드 추가시 해당 노드에 모든 공간 데이터를 전송해 주어야 하므로 고확장성의 측면에서도 비효율적이다. 이런 문제들을 해결하기 위해 공간 데이터의 분할 방법과 부분 중복 방법이 필수적이다. 하지만 특정 크기 이하의 공간 데이터 분할은 트랜잭션 처리시 여러 노드에 걸쳐 수행되는 분산 트랜잭션들을 발생시키게 되며 노드간 결과 전송으로 인한 내부 네트워크 부하와 다른 노드와의 동기화 수행 등으로 인해 처리시간의 지연을 야기시킨다. 따라서 웹 사용자들에게 지리정보 제공을 목적으로 하는 본 논문의 비공유 공간 데이터베이스 클러스터는 공간 데이터들을 일정한 지역을 단위로 하는 데이터 분할 방법을 적용시키며 공간 데이터의 접근 빈도수를 고려한 부분 중복 방법을 이용한다.



(그림 1) 부분 중복 방식의 비공유 공간 데이터베이스 클러스터

(그림 1)은 본 논문에서 기반으로 하고 있는 비공유 공간 데이터베이스 클러스터의 구조이다. 각 클러스터 노드에는 특정 크기의 지역 단위로 분할된 공간 릴레이션들이 저장되며 클러스터 노드의 고장시 데이터의 복구를 위해 적어도 하나의 백업(Backup) 본을 서로 다른 노드에 중복시킨다. 또한 사용자의 요구로부터 각 클러스터 노드에서 페이지 접근 빈도수, 트랜잭션의 응답시간, 최대 질의 처리 수를 평가하고[17], 사용자 요구의 패턴을 읽기 연산과 갱신 연산으로 분류하여, 읽기 연산이 빈번할 경우에는 복제를 수행하고 갱신 연산이 빈번할 경우에는 분할을 수행하여 서버의 부하를 분산시킨다.



(그림 2) 주제별 레이어들로 구성된 지역 A 공간 릴레이션의 예

(그림 2)는 여러개의 주제별 레이어(Layer)들로 구성된 지

역A 공간 릴레이션으로, 지역A 릴레이션은 해당 지역에 존재하는 강, 건물, 도로 등의 여러 레이어로 분류되는 공간 릴레이션들의 집합이다.

웹 GIS 클러스터링 시스템에서 제공하는 지리정보 서비스는 해당 지역에 속하는 기본 레이어들을 바탕으로 해야 정확한 위치 정보를 알 수 있다는 특성이 있다. 따라서 이런 레이어들이 분산되어 저장될 경우 다른 노드로부터 해당 레이어 정보를 가져와야 하는 부하가 발생하므로 해당 지역의 모든 레이어들을 하나의 지역 단위로 묶어서 관리한다. 일반적으로 공간 데이터를 대상으로 하는 공간 연산은 기존의 질의를 처리하는 것보다 고비용의 연산이 필요하게 되므로, 공간 질의 처리에서 공간 색인은 필수적이며, 본 비공유 공간 데이터베이스 클러스터 환경에서는 관리의 간편함과 연산의 간단함 때문에 가장 널리 이용되는 공간 색인인 R 트리를 이용한다[13].

3.2 메타 정보

비공유 공간 데이터베이스 클러스터에서는 사용자로부터 질의 수행을 요구 받은 노드에 데이터가 없을 경우 해당 데이터가 있는 노드를 찾아서 질의 수행을 요청해야 하므로 각각의 노드에서 관리되고 있는 데이터들의 정보를 서로 공유해야 한다. 이러한 메타 정보는 전역 메타 정보와 지역 메타 정보로 구성된다.

3.2.1 전역 메타 정보

비공유 공간 데이터베이스 클러스터에서 노드간의 정보를 교환하기 위해서는 모든 노드에서 참조 가능한 전역 메타 정보가 필요하다. 전역 메타 정보는 모든 클러스터 노드에 동일하게 저장되며 전역 메타 정보에 갱신이 발생하게 되면 모든 노드에 동일하게 반영시켜 주어야 하며, 현재 비공유 데이터베이스 클러스터 내에 연결되어 있는 다른 클러스터 노드들에 대한 정보, 공간 릴레이션들의 접근 빈도에 관한 정보 및 분할 및 중복된 릴레이션들에 대한 정보 등이 있으며 이러한 기본적인 전역 메타 정보 외에 본 논문에서 제안하는 기법을 수행하기 위해 아래와 같은 전역 메타 정보를 추가한다.

〈표 1〉 제안 기법을 수행하기 위해 확장된 전역 메타 정보

내 용	구 조
지역 릴레이션 구성 정보	(지역 릴레이션 명, 레이어 단위의 릴레이션 명)
지역 릴레이션 위치 정보	(지역 릴레이션 명, 클러스터 노드 ID)
지역 릴레이션 영역 정보	(지역 릴레이션 명, 지역 릴레이션 영역 MBR)
공간 객체들의 평균 중점	(레이어 단위의 릴레이션 명, X좌표의 값, Y좌표의 값)

<표 1>에서의 전역 메타 정보 중 지역 릴레이션 구성 정보는 지역 릴레이션 단위의 중복을 수행하기 위한 정보이며 릴레이션 위치 정보는 해당 릴레이션에 대한 공간 연산을 수행할 수 있는 클러스터 노드들을 찾아내기 위한 정보이다. 마지막으로 지역 릴레이션 영역 정보와 공간 객체들의 평균 중점은 본 논문에서 제안하고 있는 논리적 분할 영역을 생성하기 위한 정보이다.

3.2.2 지역 메타 정보

지역 메타 정보는 각각의 지역 클러스터 노드에서 사용자 질의 처리를 수행하기 위해 관리되는 정보이고, 해당 노드에서 관리되어지고 있는 릴레이션들의 이름과 각각의 릴레이션 내에 있는 레코드와 필드들의 통계 정보 등이 있다. 본 논문에서는 이런 정보들 이외에 제안 기법을 수행하기 위해 필요한 다음과 같은 추가적인 정보를 관리한다.

〈표 2〉 제안 기법을 수행하기 위해 확장된 지역 메타 정보

내 용	구 조
논리적 분할 영역 정보	(지역 릴레이션 명, 논리적 분할 영역)

<표 2>에서 논리적 분할 영역 정보는 지역 릴레이션에 속하는 레이어 단위의 공간 릴레이션들에 대해 공간 조인 연산을 병렬로 수행할 경우 해당 지역 노드에서 수행할 할당 영역에 대한 정보를 나타낸다.

점갯수	객체타입	객체 MBR	객체의 점 리스트

(그림 3) 공간 객체의 헤더 정보

또한, (그림 3)과 같이 공간 릴레이션 내에 있는 레코드들은 공간 객체에 대한 빠른 접근과 효율적인 관리를 위해 객체의 타입, 크기 정보 및 효율적인 여과 단계와 빠른 공간 색인을 생성하기 위한 객체의 MBR 정보 등을 갖는 공간 헤더 정보 필드를 갖게 되며, 여기에 추가적으로 본 기법에서 제안하고 있는 여과 단계를 빠르게 수행하기 위한 객체 MBR의 중점 좌표를 저장한다.

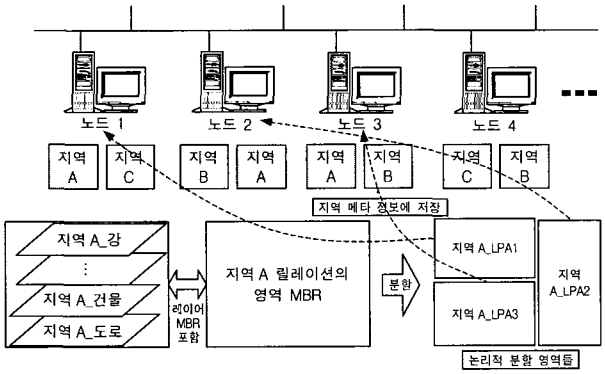
3.3 공간 릴레이션의 논리적 분할 영역

제안 병렬 공간 조인 기법은 백업과 부하분산을 목적으로 둘 이상의 클러스터 노드들에 중복 저장된 공간 릴레이션들에 대한 조인 질의 수행시 해당 릴레이션들을 저장하고 있는 모든 클러스터 노드들에서 공간 조인 연산을 병렬로 수행하도록 함으로써 고비용의 공간 조인 연산을 빠르게 처리하는 것을 목적으로, 공간 조인 연산에 참여하는 모든 클러스터 노드들이 별도의 작업생성의 과정 없이 병렬로 수행하기 위한 논리적 분할 영역에 대해 기술한다.

3.3.1 논리적 분할 영역

제안 기법은 병렬 공간 조인 수행시 별도의 작업 생성

및 관리에 따른 추가적인 연산 비용과 작업 할당과 결과 전송시 발생하는 추가적인 메시지를 제거하기 위해 중복된 지역 릴레이션들의 영역 MBR을 분할한 메타 정보인 논리적 분할 영역을 이용한다. 논리적 분할 영역을 사용하여 해당 지역의 릴레이션들을 관리하고 있는 모든 노드들에게 병렬 공간 조인 수행시 자신이 수행할 공간 조인의 영역을 지역 메타 정보로 설정해 놓을 경우 별도의 생성단계와 할당단계를 수행할 필요가 없어 공간조인 질의를 빠르게 처리할 수 있다.



(그림 4) 지역 A 릴레이션들에 대한 논리적 분할 영역의 할당

(그림 4)는 지역 A 릴레이션들에 대해 논리적으로 분할 영역을 할당하는 과정을 나타내고 있다. 이런 논리적 분할 영역은 중복된 지역 릴레이션들의 개수에 따라 분할하고, 중복 릴레이션을 저장하고 있는 모든 클러스터 노드에 할당되며 각 노드의 메타 정보에 기록되어 진다. <표 3>은 클러스터내의 한 노드인 노드 1에서 가지고 있는 이러한 논리적 분할 영역에 대한 메타 정보의 예이다.

〈표 3〉 클러스터 노드 1의 논리적 분할 영역에 대한 메타 정보의 예

지역 릴레이션명	논리적 분할 영역
지역 A	Rect(Node1_AminX, Node1_AminY, Node1_AmaxX, Node1_AmaxY)
지역 B	Rect(Node1_BminX, Node1_BminY, Node1_BmaxX, Node1_BmaxY)
...	...

논리적 분할 영역은 클러스터의 성능을 높이기 위한 재조정 혹은 클러스터 노드의 확장으로 인해 공간 릴레이션의 추가적인 중복이 발생할 경우 재분할된다. 논리적 분할 영역의 설정 및 재조정은 다음 절에서 설명한다.

3.3.2 논리적 분할 영역의 설정과 재조정

중복되지 않은 지역 릴레이션에 대한 논리적 분할 영역은 해당 지역 릴레이션내의 모든 레이어들을 포함하는 MBR 영역과 동일하다. 최초의 지역 릴레이션 중복으로 생성되는 논리적 분할 영역은 전역 메타 정보로 갖고 있는 객체 MBR의 평균 중점 값을 이용하여 설정하며 각 논리적 분할 영역에 포함되는 객체들의 개수가 비슷해지도록 한다. 이러한 논리적 분할 영역의 기준 점은 다음과 같은 과정을 통해 구할 수 있다.

먼저 (Xmin, Ymin, Xmax, Ymax) 값을 갖는 객체 MBR의 중점 (Xcent, Ycent)은 다음과 같은 식을 통해 구할 수 있다.

$$X_{cent} = \frac{X_{\min} + X_{\max}}{2}, \quad Y_{cent} = \frac{Y_{\min} + Y_{\max}}{2} \tag{1}$$

또한 레이어 단위의 릴레이션에 대한 평균 중점(Layer_Xcent, Layer_Ycent)을 구하는 식은 다음과 같고, m은 릴레이션에 속하는 객체의 개수이다.

$$Layer\_Xcent = \frac{\sum_{i=1}^{m} iXcent}{m},$$

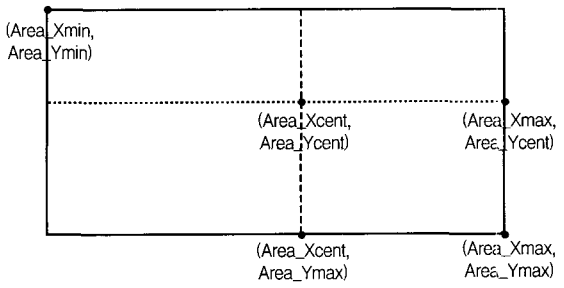
$$Layer\_Ycent = \frac{\sum_{i=1}^{m} iYcent}{m} \tag{2}$$

특정 지역에 해당하는 릴레이션들의 평균 중점(Area_Xcent, Area_Ycent)는 다음과 같은 식을 통해 알 수 있으며, n은 지역 릴레이션 내에 있는 레이어 단위의 릴레이션들의 개수이다.

$$Area\_Xcent = \frac{\sum_{i=1}^{n} iLayer\_Xcent}{n}$$

$$Area\_Ycent = \frac{\sum_{i=1}^{n} iLayer\_Ycent}{n} \tag{3}$$

이렇게 해서 구한 지역 릴레이션의 평균 중점을 이용하여 지역 영역을 분할하면 객체의 분포가 비슷한 두개의 영역으로 나눌 수 있게 된다. 이 때 식 (2)의 결과인 레이어 단위의 릴레이션내에 있는 객체들의 평균 중점은 메타 정보로 관리하고 있으므로 분할 영역을 설정할 때 식 (3)에 대해서만 계산한다.



(그림 5) 논리적 분할 영역의 기준 축

특정 지역에 해당하는 릴레이션들의 평균 중점(Area_Xcent, Area_Ycent)를 사용하여 두 개의 영역으로 분할하게 되는 기준 축은 (그림 4)와 같이 X축 또는 Y축이 될 수 있다. 이 때 분할된 두 영역의 크기가 비슷하도록 하는 축을 기준으로 선택하며 이것은 다음과 같은 방법을 통해 알 수 있다. 먼저 (그림 5)에서 X축으로 분할하였을 경우 생성되는 두 분할 영역은 각각 다음과 같고, AREA( )는 주어진 좌표로 생성되는 사각형의 넓이를 구하는 함수이다.

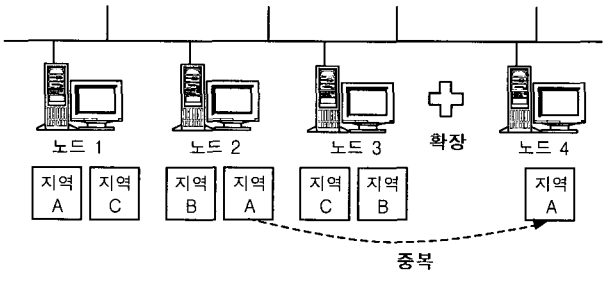
$$AxisX1\_AREA = AREA(Area\_Xmin, Area\_Ycent, Area\_Xmax, Area\_Ymax))$$
$$AxisX2\_AREA = AREA(Area\_Xmin, Area\_Ycent, Area\_Xmax, Area\_Ymax))$$

X축 분할로 생성된 두 영역의 차로 생기는 영역 AxisX_AREA는 다음과 같으며 Y축으로 분할하였을 경우도 동일한 방법으로 구할 수 있다.

$$AxisX\_AREA = ABS(AREA(AxisX1) - AREA(AxisX2))$$
$$AxisY\_AREA = ABS(AREA(AxisY1) - AREA(AxisY2))$$

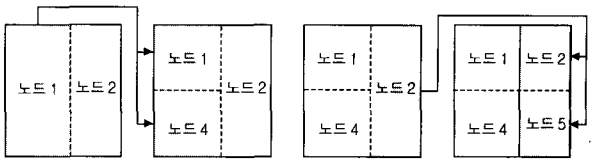
여기서 ABS()는 절대값을 반환하는 함수이며, 이렇게 구해진 AxisX_AREA와 AxisY_AREA를 비교하여 더 작은 쪽의 축을 논리적 영역의 분할 축으로 결정한다.

초기 지역 릴레이션의 중복으로 설정된 논리적 분할 영역은 동일 릴레이션이 중복될 때마다 재조정되어야 한다. 이러한 재조정을 매번 분할될 수만큼의 평균 영역을 구하여 클러스터 노드에 할당하는 것은 중복 수행시 부하로 작용하므로, 본 논문에서는 최초의 중복이 발생하였을 경우에만 위와 같은 방법으로 논리적 분할 영역을 설정하며 그 이후의 분할은 분할된 영역들 중 가장 큰 영역을 이등분하여 분할을 수행한다. 이 때, 분할된 영역상의 객체들이 편중되지 않도록 해당 영역의 길이가 긴 축에 대해 수직축을 분할 축으로 사용하여 분할된 영역이 정사분면에 가까워지도록 한다.



(그림 6) 노드 4의 확장을 통한 지역A의 중복

(그림 6)에서 지역 A 릴레이션을 저장하고 있는 노드 1과 노드 2의 논리적 분할 영역 중 더 큰 영역을 분할하여 새로 확장되는 노드 4에 할당해 준다.



(a) 노드 4의 확장 (b) 노드 5의 확장

(그림 7) 지역 A 릴레이션의 중복으로 인한 논리적인 영역 분할의 예

(그림 7)은 지역 A 릴레이션을 중복 저장하고 있는 노드 1과 노드 2의 초기 논리적 분할 영역과 노드 4와 노드 5의 확장시 논리적 분할 영역의 재조정 과정을 나타내고 있다.

3.4 병렬 공간 조인 기법

비공유 공간 데이터베이스 클러스터로 구성된 웹 GIS 클러스터링 시스템에서 사용자가 요구한 질의는 클러스터 노드들 중 하나의 노드로 전달된다. 해당 노드는 질의를 분석하여 자신이 처리할 수 있는 질의는 지역 노드에서 처리하고 그렇지 않은 질의에 대해서는 전역 메타 정보를 참조하여 질의를 처리할 수 있는 지역 노드로 포워딩시킨 후 최종 결과를 전송 받아 사용자에게 전달하게 된다. 본 절에서는 사용자의 공간 조인 질의를 논리적 분할 영역을 사용하여 병렬로 처리하는 기법에 대해 기술한다.

3.4.1 조인을 수행할 클러스터 노드들의 선정

공간 조인 질의를 처리해야 하는 클러스터 노드는 먼저 전역 메타 정보에서 관리되는 지역 릴레이션의 위치 정보를 통해 공간 조인 질의를 병렬로 수행할 클러스터 노드들의 리스트를 얻어온다. 이러한 노드들은 해당 릴레이션들을 이미 중복하여 저장하고 있으므로 별도의 데이터 전송없이 파스트리(parse tree) 형식의 질의문과 자신의 노드 ID를 각각의 노드들에게 전송하여 공간 조인 연산을 요청한다.

공간 조인 연산 수행을 요구 받은 각 클러스터 노드들은 자신의 지역 메타 정보에 있는 조인 대상 릴레이션들의 논리적 분할 영역을 참조하여 자신이 수행할 공간 조인의 영역을 알아낸 후 해당 영역 내의 공간 객체들에 대해서만 공간 조인 연산을 수행하게 된다. 예를 들어 클러스터 내의 노드 1, 노드 2, 노드 3에 중복 저장된 공간 릴레이션 R과 S를 다음과 같은 조건으로 조인을 한다고 가정하자.

$$SELECT \times FROM\ R, S\ WHERE\ CONTAINS(R.Obj, S.Obj);$$

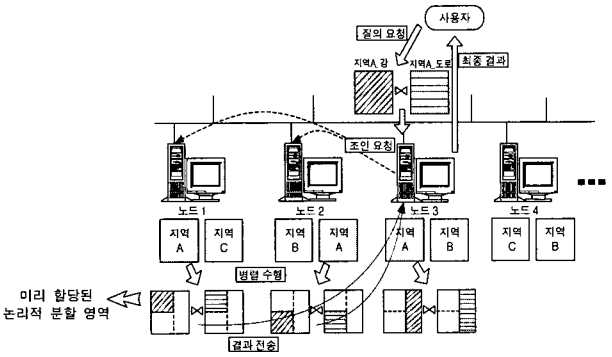
위의 조인 질의문은 각각의 노드 1, 노드 2, 노드 3에 설정된 논리적 분할 영역 LPA_Node 1, LPA_ Node 2, LPA_ Node 3에 의해 다음과 같은 3개의 질의문으로 분리된다.

① $SELECT \times FROM\ R, S\ WHERE\ INTERSECTS(LPA\_Node1, R.OBJ)\ AND\ CONTAINS(R.OBJ, S.OBJ);$

② *SELECT×FROM R, S WHERE INTERSECTS (LPA_Node2, R.OBJ) AND CONTAINS (R.OBJ, S.OBJ);*

③ *SELECT×FROM R, S WHERE INTERSECTS (LPA_Node3, R.OBJ) AND CONTAINS (R.OBJ, S.OBJ);*

공간 조인 연산 수행을 요청 받은 각 클러스터 노드는 자신의 논리적 분할 영역에 겹쳐지는 공간 객체들에 대해서만 공간 조인 연산을 수행한 후 결과 레코드들을 공간 조인 연산을 요청한 노드에게 전송하게 된다.

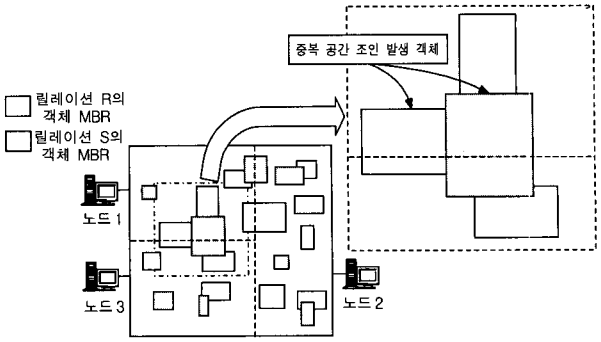


(그림 8) 논리적 분할 영역을 이용한 병렬 공간 조인 수행의 예

(그림 8)은 지역 A 릴레이션 내에 있는 강과 도로 레이어에 대한 공간 조인 질의에 대해 논리적 분할 영역을 이용한 병렬 공간 조인 수행의 예로, 사용자로부터 질의를 요구받은 질의 처리 노드는 전역 메타 정보를 참조하여 조인 대상 릴레이션들을 중복하여 저장하고 있는 노드들을 검색한 후 해당 노드들에게 사용자 질의를 전송하여 병렬로 공간 조인 연산을 수행할 것을 요청한다. 공간 조인 연산 수행을 요청받은 노드들은 자신의 지역 메타 정보에 설정된 논리적 분할 영역에 포함되는 공간 객체들에 대해서만 공간 조인 연산을 수행하게 된다. 각각의 노드들에서 병렬적으로 수행된 공간 조인의 결과들은 질의 처리 노드에게 전달되며 질의 처리 노드는 최종 결과들을 사용자에게 전송하게 된다.

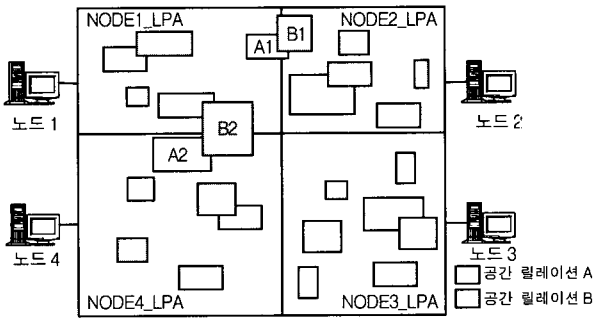
3.4.2 여과 단계에서의 중복 후보 객체 제거 기법

제안 기법은 객체 MBR의 중점을 이용한 여과 단계를 수행하여 분할 영역 경계 상에 중첩되는 공간 객체들에 대해 고비용의 정제 연산이 중복 수행되는 것을 방지한다. 질의 수행을 요청 받은 지역 노드는 자신의 지역 메타정보에 있는 논리적 분할영역에 겹치는 객체들에 대해 공간 조인 연산을 수행한다. 하지만 논리적 분할 영역과 같이, 공간 분할을 기반으로 병렬 공간 조인 연산을 수행할 경우에는 (그림 9)와 같이 분할 영역 경계선 상에 존재하는 객체들에 대해 고비용의 공간 조인 연산이 중복 수행되는 문제가 발생한다.



(그림 9) 중복 공간 조인이 발생하는 객체의 예

분할 영역의 경계선에 존재하는 공간 객체들에 대한 고비용의 중복 조인 연산을 제거하기 위해 공간 객체들의 MBR 중점의 위치를 이용한 여과 기법을 이용한다. 공간 릴레이션 내의 모든 튜플에는 대용량의 공간 객체를 빠르게 접근하기 위한 공간 헤더 정보를 가지고 있으며, 이 헤더 정보 내에는 객체의 MBR 정보 및 본 논문에서 제안하는 여과 기법을 수행하기 위한 MBR 중점의 값이 저장되어 있다. 본 기법에서는 논리적 분할 영역 경계선 상에 존재하는 공간 객체들을 MBR 중점의 위치에 따라 선택적으로 여과를 수행함으로써 여러 노드에서 고비용의 정제 연산이 중복 수행되는 것을 방지한다.



(그림 10) 클러스터 노드 상에서 공간 릴레이션 A와 B의 공간 조인 연산

(그림 10)은 클러스터 노드들에 중복 저장된 공간 릴레이션 A와 B에 대한 공간 조인 연산을 나타낸다. 공간객체 A1과 B1은 논리적 분할영역 NODE1_LPA과 NODE2_LPA의 경계선상에 존재한다. 이는 여과 단계에서 기준 릴레이션을 설정한 후 기준 릴레이션과 후보 객체들의 MBR 중점의 위치에 따라 최종 후보객체 쌍을 출력해 중복되는 후보 객체 쌍들로 인한 중복 정제 연산이 발생하지 않도록 한다. 여과 연산시 기준 릴레이션과 객체 MBR의 중점 위치에 따라 후보 객체 쌍으로 출력될 수 있는 경우는 다음과 같다.

① 두 공간 객체의 MBR 중점이 논리적 분할 영역 내에 있는 경우

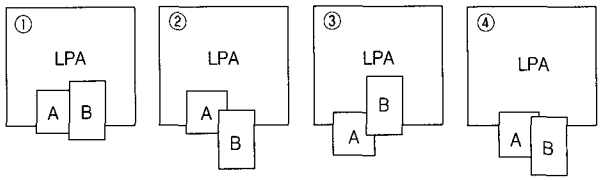
② 기준 릴레이션 내의 공간 객체 MBR 중점만 논리적

분할 영역 내에 있는 경우

③ 기준 릴레이션 내의 공간 객체 MBR 중점만 논리적 분할 영역 외에 있는 경우

④ 두 공간 객체의 MBR 중점이 논리적 분할 영역 외에 있는 경우

이 경우를 도식화하면 (그림 11)과 같다. 여기서 LPA는 논리적 분할 영역이고 A는 기준 릴레이션 내에 있는 공간 객체 A의 MBR을 의미한다.



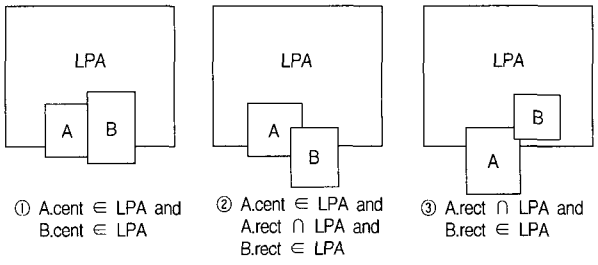
(그림 11) 논리적 분할 영역 경계선 상에서 후보 객체 쌍이 될 수 있는 경우

(그림 11)에서 기준 릴레이션의 공간 객체 MBR 중점이 자신의 논리적 분할영역 내에 있는 ①과 ②의 경우만 최종 후보객체 쌍으로 출력하여 정제연산을 수행하고 ③과 ④의 경우는 해당 영역 하단에 존재하는 논리적 분할 영역을 관리하는 노드에서 수행하게 된다. 그러나 이 경우에도 아래의 (그림 12)(c)와 같이 다른 논리적 분할 영역 내에 완전히 포함되는 공간 객체들이 여과 수행에서 누락되는 경우가 발생할 수 있다. 그러므로 LPA의 논리적 분할 영역을 갖는 클러스터 노드에서 최종 후보 객체 쌍으로 출력할 수 있는 경우는 다음과 같으며, (그림 12)는 아래 경우에 대한 조건식을 도식화한 것이다.

① 두 공간 객체의 MBR 중점이 논리적 분할 영역 내에 있는 경우

② 기준 릴레이션 내의 공간 객체의 MBR 중점만 논리적 분할 영역 내에 있는 경우

③ 기준 릴레이션외의 공간 객체의 MBR이 논리적 분할 영역에 완전히 포함되는 경우



(그림 12) 논리적 분할 영역 경계 상의 공간 객체가 후보 객체가 되는 경우

여과 연산은 모든 공간 릴레이션마다 구축된 R 트리 공간 색인을 이용한다. 이러한 중복 후보 객체 제거 기법을 적용한 R 트리 공간 색인 기반의 여과 알고리즘은 다음과 같다.

```
INPUT :
  R, S : R-tree Node
  LPA : Logical Partition Area
SpatialJoinFilter (R, S, LPA)
  E_R, E_S : Entity of R and S ;
  E_R.rect, E_S.rect : MBR of E_S and E_R
  E_R.cent, E_S.cent : Central Point of E_S and E_R
  E_R.ref, E_S.ref : Page ID of E_S and E_R
  for (all E_S ∈ S) {
    for (all E_R ∈ R with E_R.rect ∩ E_S.rect ≠ Ø)
      if ( (Get the Next Entity of R) and (Get the Next Entity
           of S) ) {
        E_R.cent ← Get the Central Point of E_R ;
        E_S.cent ← Get the Central Point of E_S ;
        if (E_R.cent ∈ LPA and E_S.cent ∈ LPA)
          Output (E_R, E_S) ;
        else if (E_R.cent ∈ LPA and E_R.rect ∩ LPA and E_S.rect
                 ∩ LPA)
          Output (E_R, E_S) ;
        else if (E_S.rect ∈ LPA and E_R.rect ∩ LPA)
          Output (E_R, E_S) ;
      }
  }
```

(알고리즘 1) R-tree 공간 색인 기반의 여과 알고리즘

3.4.3 정제 수행 및 공간 조인 결과의 전송

여과 연산을 통해 출력되는 후보 객체 쌍에 대해 지역 노드에서는 두 후보 객체에 대한 정제 연산을 수행한다. 이러한 정제 연산은 기존의 공간 조인에서 보편적으로 사용되는 평면 스위핑(plane-sweeping) 기법[8, 18, 19]의 정제 부분을 변형하여 사용한다. 평면 스위핑은 다차원의 공간 데이터를 한 축(x 축, y 축 등등)을 기준으로 정렬한 다음, 정렬한 순서대로 축 방향으로 스위핑하면서 데이터를 처리하는 방식이다.

```
INPUT :
  ParseTree : Spatial Join Parse Tree
  RelationID : Region Relation ID
BOOL ParallelSpatialJoin ( ParseTree, RelationID )
  NodeID : Current Cluster Node ID ;
  ClusterNodes[ ] : Cluster Node List ;
  NodeCount : Integer Variable ;
  ResultRecords : Query Result Records ;
  ClusterNodes ← GetClusterNodesFromMeta ( RelationID ) ;
  NodeCount ← ClusterNodes.Count ;
  for ( NodeCount is bigger than 0 )
    Forward the Query to the related Cluster Node ;
    Decrease NodeCount by 1 ;
  end for

  CreateThread ( RecieveResult, ClusterNodes )
  while ( true )
    Get ResultRecords from LocalSpatialJoin ( ParseTree ) ;
    if ( ResultRecords is not NULL ) than Send the result
       records to User ;
    else return TRUE ;
  end while
```

(알고리즘 2) 질의처리 노드에서의 공간 조인 알고리즘

(알고리즘 2)에서 조인에 참가할 노드들을 선정하고, 각 노드들에 해당 질의를 전달합니다. 각 지역 노드는 정제 연산을 통해 조인 조건을 만족하는 결과 레코드들을 파이프라인 방식[4, 15]으로 공간 조인 질의 수행을 요청한 질의처리 노드에게 전송하며, 질의처리 노드에서는 전달받은 결과 레코드들에 대해 별도의 중복 제거 과정없이 곧바로 사용자에게 전송하게 된다. 파이프라인 방식 처리는 조인 중간 결과를 디스크를 통하지 않고 다른 프로세서에게 직접 전달하므로 효율적이다.

4. 성능 평가

4.1 평가 환경

본 실험에서 제안 기법과 기존 기법[11]은 비공유 공간 데이터베이스 클러스터 시스템[14]에서 구현되었다. 평가에 사용되는 모든 릴레이션에 R트리 색인을 구축하였으며 공간 조인시 R트리를 사용하여 여과 단계를 수행하였다. 또한 성능 평가시 질의 처리 노드에서 해당 결과들을 사용자에게 전송하는 시간은 모든 기법에서 동일하므로 성능 평가 결과에서 제외시켰다. 평가에 사용된 시스템 환경은 아래의 <표 4>과 같다.

〈표 4〉 성능 평가에 사용된 시스템 환경

기 종	IBM PC
CPU	Pentium III 930MHz
Main Memory	768MB
Hard Disk	40GB
운영체제	Windows 2000 Professional
개발 언어	Visual C++ 6.0
LAN	100Mbps 허브, Category 5 케이블, 10/100Mbps NIC

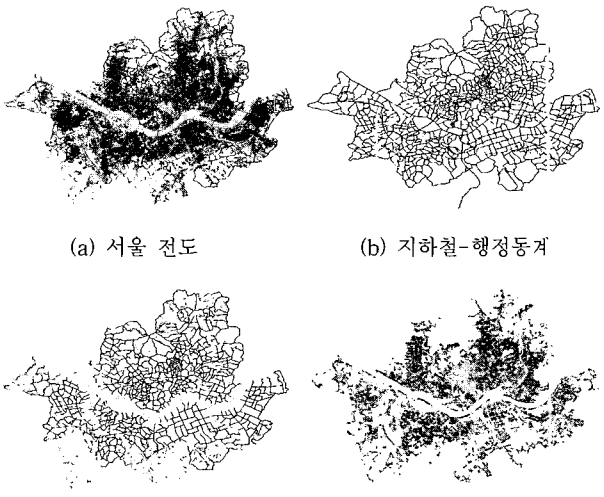
4.2 릴레이션 크기에 따른 병렬 공간 조인 성능의 평가

본 평가에서는 기존 기법과 제안 기법을 공간 조인 대상 릴레이션들의 크기를 증가시키며 질의 처리 시간을 비교하였다.

〈표 5〉 평가 릴레이션들의 상세 정보

	대상 릴레이션명	릴레이션 전체크기(Kb)	튜플 총개수(개)
CASE 1	지 하 철	42	112
	행정동계	80	100
CASE 2	강	498	1,220
	동 경 계	607	587
CASE 3	건 물	1,348	7,739
	도 로	1,394	6,482

<표 5>는 본 실험에 사용된 공간 릴레이션들에 대한 정보이며, (그림13)은 실험에 사용된 서울지역 공간 정보를 나타낸다. (그림 13)(a)는 서울 전체 지역, (그림 13)(b)는 지하철과 행정동계, (그림 13)(c)는 강과 동경계, (그림 13)(d)는 건물과 도로를 표시하고 있다.

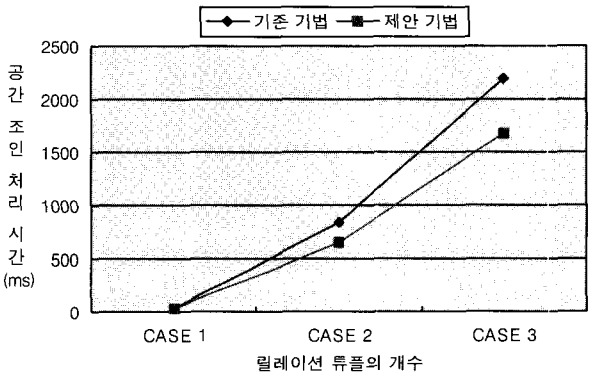


(a) 서울 전도 (b) 지하철-행정동계

(c) 강-동경계 (d) 건물-도로

(그림 13) 실험에 사용된 공간 정보

병렬 공간 조인 연산에 참여하는 클러스터 노드의 수는 최대 4대이고, 사용자는 1000명, 사용자별 질의 수는 10,000개이며, 각 사용자 질의는 모집단을 기반으로 임의로 샘플링된 공간 객체를 기반으로 생성된다. CASE 1은 지하철이 어느 행정동계와 교차하게 되는지를 처리하는 연산이고, CASE 2는 강과 교차하는 동을 검색하는 연산이며, CASE 3은 특정 도로와 인접한 건물을 검색하는 교차 연산이다.

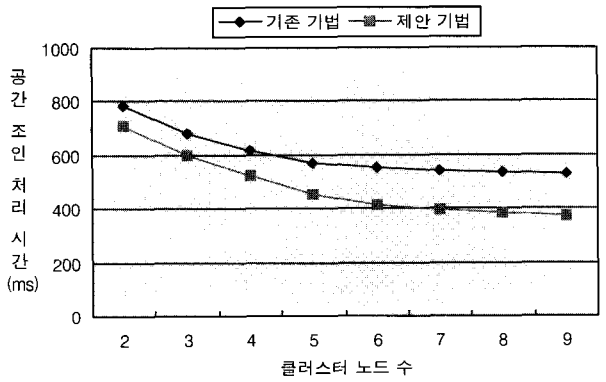


(그림 14) 릴레이션 크기에 따른 병렬 공간 조인 성능의 변화

(그림 14)은 릴레이션의 크기가 커짐에 따라 기존 기법보다 제안 기법이 약 23% 정도 더 빠른 질의 처리를 수행함을 알 수 있다. 이는 기존 기법에 비해 작업 생성 등의 과정없이 이미 각 노드에 할당된 논리적 분할 영역을 이용하여 모든 노드에서 병렬적으로 공간 조인 연산을 수행하는 제안 기법이 더 효율적임을 알 수 있다.

4.3 클러스터 노드 수에 따른 병렬 공간 조인 성능의 평가

본 실험에서는 공간 조인 연산을 클러스터 노드의 수를 증가시키며 기존 기법과 제안 기법의 질의 처리 시간을 비교하였다. 평가에 사용된 공간 릴레이션은 CASE 2의 강과 동경계 릴레이션이며 공간 조인 연산은 교차 연산이다.



(그림 15) 클러스터 노드 수에 따른 병렬 공간 조인 성능의 변화

(그림 15)는 클러스터 노드의 수가 2에서 9까지 증가할 때 두 기법 모두 공간 조인 처리 시간이 감소하고 있으며, 특히 제안 기법은 기존 기법보다 수행시간이 10%(클러스터 노드 수 : 2)에서 29%(클러스터 노드 수 : 9)정도의 더 큰 비율로 감소함을 알 수 있다. 이것은 클러스터 노드가 증가할수록 기존 기법에 비해 제안 기법은 노드간 메시지의 전송없이 공간 데이터의 지역접 인접성을 고려한 MBR 기반의 논리적 분할 영역을 사용하므로 디스크 입출력 횟수가 동일하기 때문이다.

5. 결 론

인터넷 환경에서 대규모의 사용자 질의를 처리하기 위해 비공유 구조의 공간 데이터베이스 클러스터 시스템에 대한 연구가 활발하게 이뤄지고 있으나, 높은 연산 비용이 요구되는 공간 조인에 대한 효율적인 처리에 관한 연구는 거의 이루어지지 않고 있다.

본 논문은 정적이며 지역적 인접성이 높고 고비용의 CPU 연산이 요구되는 대용량의 공간 데이터에 대해 지역 단위의 데이터 분할과 중복 방법을 이용하는 비공유 구조의 공간 데이터베이스 클러스터에서 고비용의 공간 조인 연산을 병렬적으로 수행하는 기법을 제안하였다. 제안 기법은 백업과 고성능을 목적으로 여러 클러스터 노드들에 중복하여 저장된 지역 단위의 공간 릴레이션들에 대해 영역 MBR 기반의 논리적 분할 영역을 설정하여 해당 노드들에 할당하도록 한다. 사용자로부터 요구된 고비용의 공간 조인 질의는 클러스터 노드들에게 전달되며, 해당 노드들은 자신에게 할당된 논리적 분할 영역에 포함되는 공간 객체들에 대해서 객체의 MBR 중점을 이용한 여과 연산을 수행하여 각 노드들간 중복이 없는 후보 객체 쌍을 산출한다.

본 기법은 조인에 참여하는 노드마다 자신에게 할당된 논리적 분할 영역 내의 공간 객체들에 대해서만 공간 조인을 수행하도록 함으로써 기존의 병렬 공간 데이터베이스 시스템에서 병렬 공간 조인 수행시 필요로 했던 각 노드간 작업 생성 및 할당 단계와 노드간 메시지 전송 등을 제거하여 고비용의 공간 조인 질의를 빠르게 처리한다. 또한, 공간 조인 연산의 여과 단계에서 논리적 분할 영역 경계상의 공간 객체들에 대해 객체 MBR 중점의 위치에 따른 여과 기법을 사용하여 해당 객체들에 대해 고비용이 요구되는 정제 연산을 여러 노드에서 중복 수행하는 것을 방지하였으며, 중복 결과 등을 제거하기 위한 병합 과정을 수행하지 않으므로 빠른 사용자 응답시간을 만족시킨다.

향후 연구 과제로는 지역 단위의 릴레이션에 대한 분할과 중복 방법이 아닌 레이어 단위의 분할과 중복 방법을 이용하는 비공유 공간 데이터베이스 클러스터에서의 병렬 공간 조인 수행과 이러한 클러스터 시스템을 지역 노드로 갖는 분산 환경에서의 효율적인 공간 조인 수행에 관한 연구가 필요하다.

참 고 문 헌

[1] Chan-Gu Li, Load Balancing Method using Proximity of Query Region in Web GIS Clustering System, Master Thesis, Inha Univ., 2001.

[2] H. J. Lee, Parallel Pipelined Spatial Join Method for Efficient Query Processing In Distributed Spatial Database Systems, Master Thesis, Inha Univ., 2002.

[3] T. Brinkhoff, H. P. Kriegel, R. Schneider and B. Seeger, Multi-Step Processing of Spatial Joins, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.197-208, 1994.

[4] T. Brinkhoff, H. P. Kriegel and B. Seeger, Parallel Processing of Spatial Joins using R-trees, Proceedings of the 12th International Conference on Data Engineering, New Orleans, Louisiana, USA, pp.284-292, 1996.

[5] K. Tamura, Y. Nakano, K. Kaneko and A. Makinouchi, The Parallel Processing of Spatial Selection for Very Large Geo-Spatial Databases, ICPADS 2001, pp.26-30, 2001.

[6] Y. I. Jang, C. H. Lee, J. D. Lee and H. Y. Bae, Web GIS Cluster Design with Extended Workload-Aware Request Distribution (WARD) Strategy, Proceedings of KISS, Vol. 28, No.2, pp.304-306, 2001.

[7] B. Kemme, Database Replication for Clusters of Workstations, Ph.D thesis, Department of Computer Science, ETH Zurich, Switzerland, 2000.

[8] T. Brinkhoff and H. P. Kriegel, Efficient Processing of

Spatial Joins Using R-trees, Proc. ACM SIGMOD International Conference On Management of Data, Washington, DC, pp.237-246, 1993.

[9] Jin-Deog Kim, et. al, A Study on Task Allocation of Parallel Spatial Joins using Fixed Grids, KIPS Journal, Vol.8-D, No. 4, pp.347-360, 2001.

[10] E. G. Hoel and H. Samet, Data-Parallel Spatial Join Algorithms, Proceedings of International Conference on Parallel Processing, pp.227-234, 1994.

[11] L. Mutenda, M. Kitsuregawa, Parallel R-tree Spatial Join for a Shared-Nothing Architecture, 1999 Int'l Symposium on Database Applications, pp.429-436, 1999.

[12] Y. D. Seo, Implementation and Performance Evaluation of Parallel Spatial Join Algorithm using R-tree, Master Thesis, Pusan National Univ., 1999.

[13] H. P. Kriegel, T. Brinkhoff and Ralf Schneider, Efficient Spatial Query Processing in Geographic Database System, Data Engineering Bulletin 16(3), pp.10-15, 1993.

[14] Chung-Ho Lee, A Partial Replication Protocol and a Dynamically Scaling Method for Database Cluster Systems, Ph.D Thesis, Inha Univ., 2003.

[15] Gunter von Bultzingsloewen, Optimizing SQL Queries for Parallel Execution, SIGMOD RECORD, Vol.18, No.4, Dec., 1989.

[16] D. Ries and R. Epstein, Evaluation of distribution criteria for distributed database system. UBC/ERL Technical Report M78/22, UC Berkeley, May, 1978.

[17] M. L. Lee, M. Kitsuregawa, B. C. Ooi, K. Tan and A. Mondal, Towards Self-Tuning Data Placement in Parallel Database Systems, Proceedings of the ACM SIGMOD Int'l. Conf. on Management of Data, Dallas, Texas, pp.225-236, 2000.

[18] L. Arge, O. Procepiue, B. S. Ramaswamy, T. Suel and J. S. Vitter, Scalable Sweeping-Based Spatial Join, Proc. of VLDG conf., 1998.

[19] J. M. Patel and D. J. DeWitt, Partition Based Spatial-Merge Join, Proc. of ACM SIGMOD, 1996.

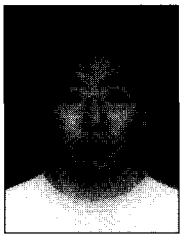
정 원 일

e-mail : wnchung@dblab.inha.ac.kr

1998년 인하대학교 전자계산공학과 (공학사)

1998년~현재 인하대학교 대학원 전자계산공학과 박사과정

관심분야 : 공간 데이터베이스 클러스터, 이동체 데이터베이스, GML, LBS 등



이 충 호

e-mail : chlee@dblab.inha.ac.kr

1997년 인하대학교 전자계산공학과(공학사)

1999년 인하대학교 대학원 전자계산공학과 (공학석사)

2003년 인하대학교 대학원 전자계산공학과 (공학박사)

2003년~현재 인하대학교 지능형 GIS 연구센터 연구원

관심분야 : 데이터베이스 클러스터, 분산 데이터베이스, 이동객체 데이터베이스 등

배 해 영

e-mail : hybae@inha.ac.kr

1974년 인하대학교 응용물리학과(공학사)

1978년 연세대학교 대학원 전자계산학과 (공학석사)

1989년 숭실대학교 대학원 전자계산학과 (공학박사)

1985년 Univ. of Houston 객원 교수

1992년~1994년 인하대학교 전자계산소 소장

1982년~현재 인하대학교 전자계산공학과 교수

1999년~현재 지능형 GIS 연구센터 소장

2000년~현재 중국 중경우전대학교 대학원 명예 교수

관심분야 : 분산 데이터베이스, 공간 데이터베이스, 지리정보 시스템, 멀티미디어 데이터베이스 등