

# 분산 공간 데이터베이스를 위한 동적 콜렉터

이재훈<sup>o</sup>, 박순영, 정원일, 배해영  
인하대학교 전자계산공학과

{neohacz<sup>o</sup>, sunny, wnchung}@dblab.inha.ac.kr, hybae@inha.ac.kr

Study on a Dynamic Collector for Distributed Spatial Databases

JaeHun Lee<sup>o</sup>, Soon-Young Park, Warnill Chung, Hae-Young Bae  
Dept. of Computer Science & Engineering, Inha University

## 요 약

분산된 공간 데이터를 효과적으로 저장, 관리하고 공유하기 위하여 분산 공간 데이터베이스 시스템의 필요성이 대두되었다. 분산 공간 데이터베이스 시스템은 많은 변화가 있는 환경이기 때문에 최적화된 질의 플랜을 작성하기가 어렵고 또한 고비용의 공간 연산 비용을 고려해야 하는 문제를 가지고 있다.

본 논문은 질의 실행 시간에 변화된 분산 데이터베이스 환경을 고려하여 질의를 수행하며 공간 조인을 병렬적으로 수행하는 동적 콜렉터를 제안한다. 동적 콜렉터는 분산 데이터베이스 환경의 변화에 적응할 수 있으며 분산 공간 조인을 효율적으로 처리할 수 있다.

## 1. 서 론

최근 독자적으로 구축된 데이터에 대해 통합하려는 요구를 만족시키기 위하여 분산 데이터베이스의 필요성이 대두되었다[2,4]. 기존의 분산 데이터베이스 질의 처리는 데이터에 대한 불확실한 통계 자료와 사용 가능한 리소스의 변화 때문에 동적 질의 연산자와 동적 콜렉터를 이용한 동적 질의 처리에 대한 많은 연구가 진행되고 있다.

분산 공간 데이터베이스 환경에서는 기존 분산 데이터베이스 환경이 가지고 있는 문제 이외에 공간 조인 연산이 고비용이라는 문제를 가지며 각 객체 쌍들에 대한 연산을 분리하여 수행할 수 있다는 특징을 가진다[3].

본 논문<sup>1)</sup>에서는 병렬 공간 조인을 고려한 동적 콜렉터를 제안한다. 제안하는 동적 콜렉터는 변수가 많은 분산 데이터베이스 환경에 적합한 질의 수행을 처리하며 고비용의 공간 조인 연산을 분할하여 독립적으로 병렬 수행되게 하여 효율적인 질의 처리를 수행한다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 살펴본다. 3장에서는 분산 공간 데이터베이스를 고려한 동적 콜렉터에 대해 기술한다. 4장에서는 제안하는 동적 콜렉터를 적용한 분산 공간 데이터베이스 시스템을 설계한다. 끝으로 5장에서 결론 및 향후 연구 방향을 논한다.

## 2. 관련 연구

기존의 분산 데이터베이스 환경에 적응하는 기법에는 동적 질의 연산자와 동적 콜렉터가 있다. 동적 질의 연산자에는 활용 가능한 메모리를 고려하는 정렬, 해쉬 조인과 빠른 응답 시간을 위한 조인이 있다[1]. 동적 콜렉터는 질의 플랜에 삽입되고 질의 실행 시간에 질의를 수행할 사이트를 그 시간의 분산 데이터베이스 환경에 맞

는 정책을 따라 선정하게 된다[5]. 또는 동적 콜렉터는 분산 데이터베이스 환경에 맞는 룰에 의해서 필요한 데이터를 가지고 와서 질의를 수행하거나 질의를 데이터가 있는 사이트로 이동시켜 질의를 수행하게 된다. 그리고 여러 사이트에서 이루어지는 질의 결과를 수집하여 다음 질의를 위한 사이트로 이동시킨다[3]. 그러나 사용자가 정의해야 하는 정책이나 룰은 그 자체를 정의하는 것이 복잡하다는 단점이 있다.

분산 공간 데이터베이스 환경에서의 병렬 공간 조인으로는 최하위 두 사이트간의 공간 조인 연산에 대해 대상이 되는 영역을 균등한 수의 공간 객체가 포함되도록 대상 릴레이션을 분할한 후 질의 수행에 참여하는 두 서버에 분배하여 병렬로 수행시키는 효율적인 질의 처리를 제안하고 있다[7]. 그러나 이 기법은 여러 분산 데이터베이스 환경의 변화에 대처하지 못하고 최하위 두 사이트에서만 병렬 공간 조인을 수행할 수 있으며 제안하는 기법을 수행하기 위해 로컬 사이트의 공간 데이터베이스가 수정되어야 한다는 단점을 가진다. 이런 단점을 해결하기 위해 데이터를 논리적으로 분할하여 각 노드에 할당하고 각 노드는 자신에게 할당된 논리적 분할 영역 내의 공간 객체들에 대해서만 공간 조인을 수행하여 각 노드간 작업 생성 및 할당 단계와 노드간 메시지 전송 등을 제거하는 병렬 공간 조인이 제안되었다[6]. 그러나 이 기법도 분산 데이터베이스 환경의 변화에 대처하지 못하며 질의 처리 전 데이터를 논리적으로 할당할 수 있는 환경을 가정하고 있는 단점을 지닌다.

## 3. 분산 공간 데이터베이스를 고려한 동적 콜렉터

제안하는 동적 콜렉터는 질의 플랜을 작성할 때 각 노드들의 데이터들을 조인하거나 각 데이터를 검색해서 합치는 부분에 삽입된다. 동적 콜렉터가 공간 조인을 수행해야 한다면 병렬 공간 조인을 할 수 있는지를 확인하도록 플래그를 설정하게 된다.

1) 본 연구는 대학 IT연구센터 육성, 지원사업의 연구 결과로 수행되었음.

```

[알고리즘 1] 사이트에 필요한 데이터의 유무 확인
Check_SiteHaveData(Data, Site)
Begin
  // 필요한 데이터가 있는지 확인
  If (!Check_ExistDataInSite(Data, Site)) Then
    // 내임 서버로부터 필요한 데이터가 있는 새로운
    // 사이트를 검색
    NewSite <- Check_NewSiteFromNameServer(Data)
    If (NewSite == null) Then
      Return false
    Else
      if (!Check_SiteHaveData(Data, NewSite)) Then
        Return false
  End

```

```

[알고리즘 2] 사이트의 리소스 사용 가능 확인
Check_SiteCanProcess(Resource, Site, Data)
Begin
  // 리소스 사용 가능 확인
  If (!Check_ResourceAvailable(Resource, Site)) Then
    NewSite <- Check_NewSiteFromNameServer(Data)
    If (NewSite == null) Then
      Return false
    Else
      if (!Check_SiteCanProcess(Resource, Site, Data))
      Then
        Return false
  End

```

```

[알고리즘 3] 사이트의 네트워크 상태 확인
Check_NetworkInSite(Site, Data)
Begin
  // 사이트 네트워크 상태 확인
  If (Check_NetworkStatus(Site) >= threshold) Then
    Return Site
  Else
    While
      (NewSite <- Check_NewSiteFromNameServer(Data))
      If (Check_NetworkStatus(NewSite) >= threshold) Then
        Return NewSite
    Return Site
  End

```

하여 각각의 결과들을 생성한다. 결과들은 각각이 독립적으로 생성되므로 콜렉터 역할을 하는 사이트에서 합쳐지게 된다. 데이터가 중복되어 있지 않다면 질의 플랜을 변경하지 않고 공간 조인을 수행하게 된다.

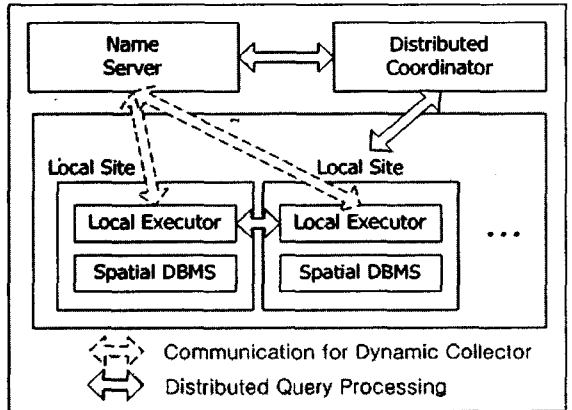
```

[알고리즘 4] 병렬 공간 조인이 가능한지 확인
Check_EnabledParallelSpatialJoin(SiteList, Region)
Begin
  // 데이터가 중복되어 있는가를 내임서버의 정보를
  // 가지고 확인
  If (!Check_ReplicateData(SiteList, Region)) Then
    Return false
  Else
    // 질의에 대해서 영역 분할이 가능한가를 확인
    If (!Check_EnabledSplitRegion(SiteList, Region)) Then
      Return false
    Else
      Execute_ParallelSpatialJoin(SiteList, Region)
  End

```

#### 4. 동적 콜렉터를 적용한 분산 공간 데이터베이스 시스템 설계

본 논문에서 제안하는 분산 공간 데이터베이스 시스템은 [그림 1]과 같이 내임 서버, 분산 코디네이터, 로컬 사이트로 구성된다.



[그림 1] 분산 공간 데이터베이스 시스템

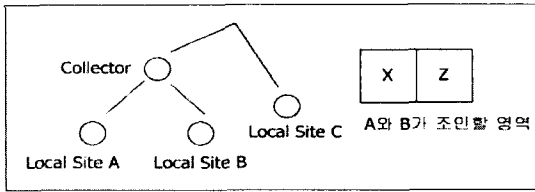
질의 실행 시간에 동적 콜렉터의 역할을 하게 되는 로컬 사이트는 조인되거나 합쳐지는 사이트들의 데이터의 유무, 리소스 사용 유무, 네트워크 상태를 위해 기술된 [알고리즘 1,2,3]에 의해 확인하여 현재 선정된 사이트가 알맞다면 현재 선정된 사이트를 가지고 질의를 수행하게 되고 알맞지 않다면 새로운 사이트를 재선정하여 질의를 수행하게 된다.

질의 실행할 때 동적 콜렉터에 병렬 공간 조인을 위한 플래그가 설정되어 있다면 다음 [알고리즘 4]에서와 같이 병렬 공간 조인이 가능한지를 확인한다.

병렬 공간 조인이 가능하면 영역을 분할하고 분할된 영역에 해당하는 데이터를 가진 사이트들 간에 공간 조인을 병렬적으로 수행되게 한다. 영역을 분할할 때는 논리적 분할을 한 다음 여과 단계에서의 중복 후보 객체를 제거한다. 그리고 각 분할된 영역에서 잠재연산을 수행

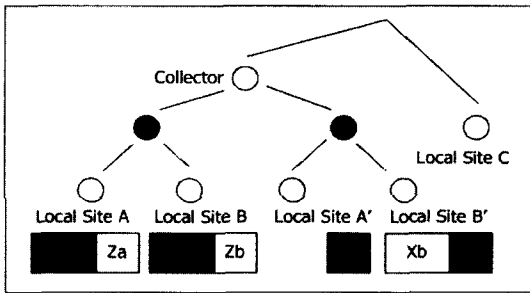
내임 서버는 분산 질의 처리를 위한 카탈로그를 관리한다. 관리하는 정보에는 로컬 사이트에서 가지고 있는 데이터들의 MBR이 포함되며 이 정보는 병렬 공간 조인을 위하여 사용된다. 분산 코디네이터는 분산 질의를 파싱하고 질의 플랜을 작성하며 질의 처리 사이트를 선정하고 질의 처리를 전체적으로 조정한다. 로컬 사이트는 로컬 실행자와 공간 데이터베이스 관리 시스템으로 구성된다. 로컬 실행자는 질의 플랜 중 자신의 로컬 사이트에 해당하는 연산을 처리한다. 해당하는 연산이 콜렉터라면 로컬 실행자는 콜렉터의 역할을 하게 된다.

코디네이터에서 작성되는 질의 플랜은 다음의 [그림 2]와 같이 로컬 사이트 A, B가 조인하는 부분에 콜렉터가 추가된다.



[그림 2] 코디네이터에서 작성된 플랜

영역 X, Y는 사이트 A, B가 가지고 있는 공간 영역 중 서로 겹치는 부분이다. A와 B는 공간 조인을 하기 때문에 콜렉터에 병렬 공간 조인에 대한 플래그가 설정되게 된다. 이 질의 플랜은 질의 실행 시간에 콜렉터를 만나게 되면 A, B 사이트에 대해 앞서 기술된 알고리즘을 적용하여 상태를 확인하고 병렬 공간 조인이 가능한지를 조사하게 된다. 사이트 A에서 영역 Z에 해당하는 데이터가 다른 사이트 A'에 중복되어 저장되어 있고 사이트 B의 영역 X, Z에 속하는 데이터는 사이트 B'에 중복되어 저장되어 있다. 따라서 [그림 3]과 같이 플랜이 변경되어 병렬 공간 조인을 실행하게 된다.



[그림 3] 질의 실행 시간에 변경된 플랜

변경된 플랜을 따라 병렬 공간 조인은 두 개의 영역으로 분할되어 조인되게 된다. 영역 Xa와 Xb의 객체들이 조인되고 영역 Za와 Zb의 객체들이 조인되어 그 결과가 콜렉터의 역할을 맡고 있는 사이트로 보내지게 되고 모어진 결과들은 다른 작업이 필요 없이 합쳐져서 다음 플랜의 연산을 위한 사이트로 이동하게 되고 질의 플랜의 남은 연산들을 수행한 다음 생성된 결과를 사용자에게 보여지게 된다.

5. 결론 및 향후 연구 방향

본 논문의 동적 콜렉터는 질의 실행 시간에 플랜에 선정된 사이트에 대해서 데이터의 유무, 리소스 사용 가능성 유무, 네트워크 상태를 확인하며 고비용의 공간 연산 조인을 효율적으로 처리하기 위해 병렬 공간 조인을 수행한다. 이와 같이 분산 공간 데이터베이스 환경에서 동적 콜렉터를 사용함으로써 분산 데이터베이스 환경의 변화에 적응할 수 있고 분산 공간 조인을 효율적으로 처리할 수 있다.

향후 과제로는 동적 콜렉터가 병렬 공간 조인과 함께 다양한 분산 공간 조인 기법을 적용할 수 있도록 하는 것이다.

참고문헌

[1] Anastasios Gounaris, Norman W.Paton, Alvaro A.A. Fernandes, and Rizos Sakellariou, Adaptive Query Processing: A Survey, Lecture Notes in Computer Science 2405, July 2002  
 [2] Donald Kossmann, The State of the Art in Distributed Query Processing, ACM Computing Surveys, Vol. 32, pp. 422-469, December 2000  
 [3] Michael Stonebraker, Paul M. Aoki, Robert Devine, Witold Litwin and Michael Olson, Mariposa: A New Architecture for Distributed Data, Proceedings of 1994 IEEE 10th International Conference on Data Engineering, pp. 54-65, 1994  
 [4] Michael Stonebraker, Paul M. Aoki, Witold Litwin, Avi Pfeffer, Adam Sah, Jeff Sidell, Carl Staelin, Andrew Yu, Mariposa: a wide-area distributed database system, VLDB Journal 5, 1, pp. 48-63, 1996  
 [5] Zachary G. Ives, Daniela Florescu, Marc Friedman, Alon Levy, Daniel S.Weld, An adaptive query execution engine for data integration, In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 299-310, 1999  
 [6] 김종현, 비공유 공간 데이터베이스 클러스터에서의 효율적인 질의 처리를 위한 병렬 공간 조인 기법, 공학 석사 학위 논문, 인하대학교, 2003  
 [7] 서영덕, R-tree를 이용한 병렬 공간 조인을 위한 알고리즘의 구현 및 성능평가, 공학 석사 학위 논문, 부산대학교, 1999  
 [8] 이환재, 분산 공간 데이터베이스 시스템에서의 효율적인 질의 처리를 위한 병렬 연쇄 공간 조인 기법, 공학 석사 학위 논문, 인하대학교, 2002