

서버 처리비용 분산을 위한 공간 뷰 클라이언트 실체화 기법

김태연*, 정보홍*, 이재동**, 배해영*

*인하대학교 전자계산공학과

**단국대학교 컴퓨터공학과

yeon211@hanmail.net

The Spatial View Client-Side Materialization Technique for Load-Balancing in Server-Side Computing Cost

Tae-Yeon Kim, Bo-Heung Jung, Jae-Dong Lee, Hae-Young Bae

*Dept. of Computer Science Engineering, Inha University

**Dept. of Computer Science Engineering, Dankook University

요약

공간 데이터베이스 시스템에서는 데이터의 보안과 사용자의 편의성을 제공하기 위해 사용자가 원하는 공간데이터만으로 구성 된 공간 뷰를 제공한다. 클라이언트/서버 환경의 공간 데이터베이스 시스템에서 다수의 클라이언트에 의해 공간 뷰에 대한 질의가 요청 될 시 대용량의 데이터를 처리하기 위한 서버의 I/O 연산의 수행비용과 질의처리 비용 및 결과 데이터의 전송을 위한 전송 비용이 서버의 부하를 일으키고 질의 처리속도의 저하를 야기시킨다. 본 논문에서는 클라이언트/서버 환경의 공간 데이터베이스 시스템에서 공간 뷰의 생성 과정을 서버와 클라이언트에 분산시킨 클라이언트 실체화 기법을 제안한다. 공간 뷰 생성의 질의처리를 서버와 클라이언트에 분산시켜 대용량의 데이터와 복잡한 공간 연산에 따른 공간 뷰 생성과정의 서버 부하를 감소시키고 클라이언트에 실체화 함으로 해서 공간뷰에 대한 질의처리 요구에 따른 서버의 병목현상과 서버 부하를 감소시켜 사용자 응답시간을 최소화 한다.

1. 서론

공간 데이터베이스 시스템의 지리데이터는 일부 영역 단위로 만들어진 것이 아니라 공통된 주제를 갖는 비공간 데이터별로 구분된 공간 데이터들의 집합인 레이어들의 단위로 관리된다.[8] 사용자의 편의성과 보안상 목적을 위하여 공간데이터베이스 시스템은 공간 뷰를 제공하는데, 공간 뷰는 공간데이터의 크기와 공간연산의 복잡도를 고려해서 실체화 되어 관리되어진다.[5] 서버/클라이언트 환경에서 공간 뷰의 실체화 관리를 위해서는 공간 뷰를 서버/클라이언트 중 어느 곳에 실체화 할 지의 문제와 실체화 된 공간 뷰와 기본 테이블간의 부결성 유지를 위한 공간 뷰 일관성 유지방법에 대한 고려가 필요하다. 기존의 서버에 실체화 시키는 방법은 다수의 클라이언트가 동시에 서버에 접속하여 질의의 요구를 할 시에는 복잡한 공간 연산 수행으로 인한 서버의 부하와 대번 네트워크를 통한 질의의 요구에 따른 질의처리 속도의 저하, 대용량 공간데이터의 반복적인 전송에 따른 전송 부하의 문제점이 있다.

본 논문¹에서는 질의처리의 향상과 서버부하와 공간 데이터의 전송 부하를 최소화하기 위해 공간 뷰를 클라이언트에 실체화 시키는 PSCMT(Preprocessing in Server and Client Materialization Technique) 기법을 제안한다. 이 기법은 다음과 같은 3단계로 나누어 처리된다. 첫째, 사용자의 뷰 정의의 요구가 있을 시 서버는 Where절의 프레디케트를 조사하여 각 테이블들에 대한 셀렉션 또는 프로젝션만을 취한 중간 결과 테이블을 생성하여 별도로 저장한다. 둘째, 생성된 중간 결과 테이블들을 클라이언트에 전송한다. 셋째, 클라이언트는 전송 받은 중간 결과 테이블들을 저장하고 뷰 정의문에 따라 공간 연산이나 조인등의 연산을 하여 공간뷰를 실체화 시킨다.

공간 뷰 생성 과정의 임시테이블들과 뷰에 관한 정보를 저장하고

관리하기 위해 본 논문에서는 먼저 서버와 클라이언트에서 사용될 자료구조를 정의하고 중간 결과들의 효율적인 전송을 위한 전송 패킷 구조를 설계하고, 기본 테이블에 대한 서버의 변경이 있을 경우에 클라이언트에 실체화 함으로 해서 발생할 수 있는 데이터의 불일치성을 방지하기 위해 공간 뷰의 일관성 유지에 관한 내용을 서술한다.

본 논문에서 제안하는 기법은 공간 뷰를 클라이언트에 실체화 함으로써 공간 뷰에 대한 서버의 처리비용을 줄이고, 대용량의 복잡한 연산을 요구하는 다수의 클라이언트에 의한 서버의 병목현상과 대용량 데이터의 전송에 따른 네트워크 전송부하를 감소시킨다. 이는 사용자의 질의 처리에 대한 응답시간을 최소화 하고, 클라이언트들의 가용한 처리 성능 및 자원을 활용할 수 있다는 장점을 갖는다.

본 논문의 구성은 다음과 같다. 2장에서는 공간 뷰 실체화 방법에 대해 기술하고, 3장에서는 서버/ 클라이언트 환경에서 서버처리 비용을 분산시킨 공간 뷰 실체화 기법에 대해서 제안한다. 마지막으로 4장에서 결론을 맺는다.

2. 공간 뷰 실체화 방법

뷰는 서로 다른 사용자들에게 효율적인 질의 처리와 보안 등의 이유로 전체 데이터 중 일부분만을 제공하는 개념의 데이터이다 [1,7,8] 일반적으로 뷰의 관리 기법으로는 크게 가상 뷰와 실체화된 뷰로 구분 할 수 있다. 전통적인 뷰의 개념으로는 가상 뷰를 다루지만, 점점 대용량의 데이터를 다루고 뷰에 대해 복잡한 질의물 요구하는 경우가 많아짐에 따라 실체화 하는 기법을 사용한다.[2] 실체화 하는 기법에는 언제 어느 환경에서 어디에 위치시키는가의 문제가 있는데 [5]에서는 공간 뷰의 클라이언트 실체화 방안을 제시하고 있다. 실체화 기법에는 뷰의 생성 시점과 클라이언트에 실체화 하는 과정에 따라 다음과 같이 구분할 수 있다.

2.1.1 뷰의 생성 시점에 따라 실체화 하는 방법

클라이언트에 의해 뷰에 대한 정의가 요구될 때 서버에서 즉시 생성 하여 실체화 하는 뷰 정의 시간 실체화(View Definition Time Materialization) 방법과 클라이언트로부터 실제 뷰에 대한 질의가 있

¹ 본 연구는 정보통신부의 대학 S/W 연구센터 지원사업의 연구 결과임

을 때 뷰를 생성 한 후 실체화하는 뷰 수행시간 실체화 (View Invoke Time Materialization) 방법을 생각 할 수 있다.

2.1.2 클라이언트에 뷰를 실체화 하는 방법

서버에서 모든 질의 처리를 하여 생성 된 공간 뷰를 클라이언트로 전송하면 클라이언트에서 실체화하는 방법과 서버에서 공간 뷰 생성에 필요한 기본 테이블 전체를 클라이언트로 전송하면 클라이언트에서 공간 뷰를 생성하여 실체화하는 방법, 서버에서 각 기본 테이블에 대한 선택선이나 프로젝션에 대한 연산만 수행 후 중간 결과 테이블들을 클라이언트로 전송하고 클라이언트는 전송 받은 테이블들로 최종 연산을 수행하여 실체화하는 방법을 고려 할 수 있다.

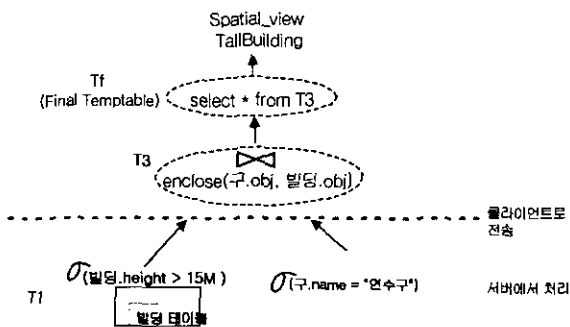
이와 같은 경우를 바탕으로 클라이언트에 실체화 시킬 수 있는 방법은 6가지가 가능한데, 대용량의 데이터로 복잡한 공간 연산을 수행해야 하는 공간 데이터베이스 시스템에서는 질의 요구가 있을 때 뷰를 실체화 하여 질의 요구를 처리하기까지는 많은 시간이 소요되므로, 뷰 생성 질의가 있을 때 미리 실체화 시키는 방법이 효율적이다. 그리고 질의처리 능력이 있는 클라이언트이므로, 서버에서 직접 뷰를 생성하는 것보다는 클라이언트에서 처리하도록 한다. 위에서 언급한 뷰 생성에 필요한 기본 테이블 전체를 클라이언트로 전송하여 처리하는 방법은 서버처리 비용은 감소하지만, 전송비용이 너무 커서 다수의 클라이언트가 동시에 뷰 정의에 대한 질의를 요구할 시 전송부하가 크고, 클라이언트의 뷰 생성 처리비용이 증가하므로 클라이언트의 부하를 야기 시키다. 또한, 전체 테이블을 전송하므로 클라이언트의 요구와는 상관 없는 정보까지 전송하게 된다. 따라서, 본 논문에서 제안하는 PSCMT 기법은 뷰 정의 시간에 실체화(View Definition Time Materialization) 질의처리를 시작하여 서버와 클라이언트에 실체화 비용을 분산시켜 클라이언트에 실체화 시키는 방안이다.

3.서버처리 비용을 분산시킨 클라이언트 실체화 기법

본 장에서는 공간 뷰를 실체화 하는 과정에서 실체화 처리 과정의 서버비용을 분산시키는 PSCMT 기법에 대해 자세히 서술한다. 이를 위해 먼저 사용되는 자료구조를 설명하고, 세부적으로 질의 처리전략에 대해 설명한다.

PSCMT 기법에서 질의 처리는 3단계로 구성되며 각 단계는 첫번째, 클라이언트에서 공간 뷰 생성 요청을 하면 서버는 기본 테이블들에 대한 각각의 프레디킷을 조사하여 각각의 테이블에 대해서 선택선 또는 프로젝션을 하여 조건에 맞는 데이터만을 추출한 후 이 결과 데이터를 별도 저장해서 관리한다. 두번째, 서버에서 생성 한 중간 결과 테이블과 클라이언트에서 처리 할 프레디킷을 클라이언트에 전송한다 세번째, 클라이언트는 전송받은 테이블과 프레디킷을 저장하고 최종연산을 수행하여 실체화 시킨다.

다음 [그림2]는 본 논문의 효율적인 이해를 위해 일반적인 공간 뷰 생성 정의문인 [예1]을 파싱(Parsing)을 거쳐 플랜트리플 작성 한 것이다.



[그림2] 프레디킷에 따른 플랜 트리

```

Create Spatial_View TallBuilding As
Select * from 빌딩, 구
Where 빌딩.height > 15M and 구.name="연수구"
and encloses (구.obj, 빌딩.obj);
    
```

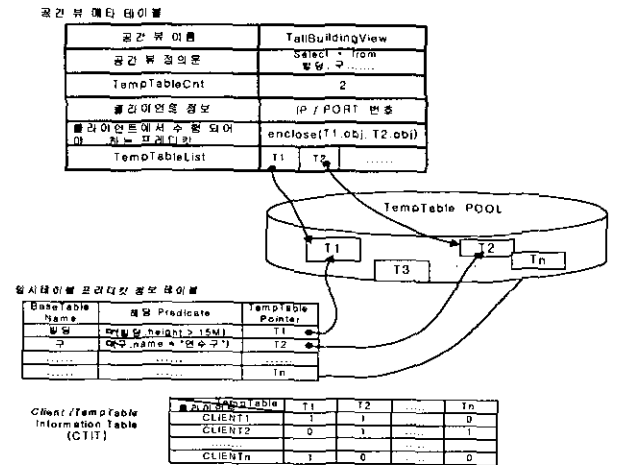
[예1] TallBuilding뷰를 정의하는 공간 SQL 정의문

3.1 PSCMT 기법을 위한 자료구조

기존의 공간 데이터 시스템에서 이 기법을 지원하기 위해서는 서버와 클라이언트에 뷰에 대한 메타 테이블을 위한 자료구조, 서버와 클라이언트 사이의 효율적인 전송을 위한 패킷 구조의 재 정의가 필요하다.

3.1.1 실체화를 위해 서버에서 관리해야 할 자료구조

서버는 생성 된 플랜트리의 토큰을 조사해서 one-table에 대한 연산만 수행하고 생성 된 결과는 다수의 클라이언트가 동일 뷰 생성 또는 질의요구가 있을 때 효율적인 처리를 하기 위해 저장한다. [그림 3-1]에서 공간 뷰 메타테이블은 공간 뷰에 대한 정보를 관리하기 위해 공간 뷰 이름, 공간 뷰 정의문, 임시테이블 수, 뷰 정의를 한 클라이언트, 클라이언트에서 처리되어야 할 부분, 임시테이블 위치정보를 저장한다. [임시테이블 프레디킷 정보 테이블]은 클라이언트가 새로운 뷰 생성 요구를 할 경우나 다른 클라이언트에 실체화 된 뷰에 대해 질의처리 요구 시 동일하게 처리되는 임시테이블에 대해서는 반복적인 처리를 하지 않기 위해 기본테이블이름, 해당 프레디킷, 임시테이블 위치정보를 별도로 관리한다.



[그림 3-1] 서버측 자료구조

3.1.2 실체화를 위한 전송 패킷 구조

클라이언트와 서버 사이의 효율적인 전송을 위한 패킷 구조는 다음과 같다.

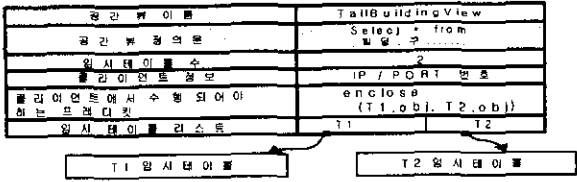


[그림 3-2] 전송패킷 구조

대용량의 공간 데이터의 전송은 한 패킷으로 처리 될 수 없는 경우가 많으므로, 위의 구조에서 Packet Tag는 모든 패킷이 전송되었는지 여부를 알기 위한 변수이다.

3.1.3 실체화를 위해 클라이언트에서 관리해야 할 자료구조

최종 연산을 수행하여 실체화 하기 위해 클라이언트는 자신이 정의한 공간 뷰 이름, 공간 뷰 정의문, 서버에서 전송 받은 임시테이블의 수, 클라이언트에서 처리되어야 하는 프레디킷, 임시테이블의 위치정보를 저장한다.

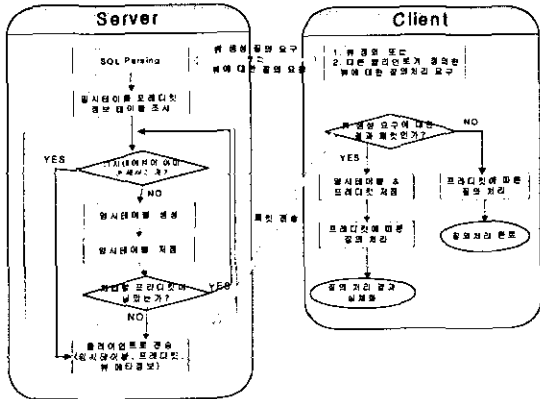


[그림 3-3] 클라이언트측 자료구조

3.2 PSCMT의 질의처리 전략

본 논문에서 제안하는 PSCMT에서 공간 뷰에 대한 서버의 질의 처리는 클라이언트의 새로운 공간 뷰 생성 요구와, 다른 클라이언트에 실체화 된 공간 뷰에 대해 질의처리 요청이 있을 때만 수행된다.

뷰 생성 질의에 대한 요청 또는 다른 클라이언트에 실체화 된 뷰에 대한 질의요청 시 처리과정은 다음과 같이 흐름도로 표현할 수 있다.



[그림 3-4] 공간 뷰 생성 질의요구 처리 흐름도

3.2.1 뷰 생성 질의요구

클라이언트로부터의 뷰 생성 질의 요구를 받았을 때 서버는 파싱한 결과에 따라 현재 [임시테이블 프래디킷 정보 테이블]에 같은 조건에 의해 이미 생성된 임시테이블이 존재하는지 검색하여 임시테이블이 존재하면 기존의 임시테이블을 사용하고 그렇지 않은 경우에만 임시테이블을 생성하기 위해 파싱된 결과의 Expression에 따라 해당 테이블에 대한 질의처리를 수행한다.

서버는 생성된 플랜트의 토권을 조사해서 one-table에 대한 연산만 수행하고 생성된 결과는 다수의 클라이언트가 동일 뷰 생성이나 뷰에 대한 질의처리 요구가 있을 때 재사용하기 위해 저장한다. 클라이언트는 전송된 임시테이블과 프래디킷에 따라 최종 연산까지 수행하여 실체화 시킨다.

[예1]에서는 T1, T2 까지만 서버에서 처리하고 나머지 처리 사항에 대해서는 다음과 같이 Query Reformulation 해서 T1, T2 와 함께 클라이언트로 전송하면 클라이언트에서 T3, Tf 에 대해 연산하여 실체화 시킨다.

T1 = select * from 빌딩 where 빌딩.height > 15M ;	1 단계
T2 = select * from 구 where 구.name = '연수구' ;	2 단계
T3 = enclose(T1.obj, T2.obj) ;	3 단계
Tf = select * from T3 ;	

3.2.2 다른 클라이언트에 실체화 된 뷰에 대한 질의 요청

서버의 뷰에 대한 정 책은 뷰를 정의한 클라이언트에만 실체화 시킨다. 클라이언트가 다른 클라이언트에 실체화 된 뷰 즉, 자신이 정의하지 않은 뷰에 대해서 질의 요청 시 서버는 공간 뷰 메타데이터를 조사하여 어떤 임시테이블이 필요한지 검색한다. CTIT테이블을

검색하여 필요한 테이블이 클라이언트에 저장되어 있지 않은 테이블만 전송한다. 전송 받은 테이블과 현재 저장되어 있는 임시테이블을 이용하여 클라이언트는 데이터를 구성하고 질의 처리를 한다. 이때 클라이언트는 전송된 테이블에 대해서는 뷰의 일관성 유지를 용이하게 하기 위해 저장하지 않는다.

3.2.3 공간 뷰 일관성 유지

공간 데이터의 특성 상 기본 테이블에 대한 변경이 빈번하지는 않지만 서버에서 기본 테이블에 대한 변경 시 파생된 뷰의 데이터 무결성을 유지하기 위해 일관성 유지를 고려해야 하는데, 본 논문에서는 기본 테이블에 대한 변경이 있을 때 우선 파생된 임시테이블을 [임시테이블 프래디킷 정보 테이블]에서 조사하여 갱신된 부분에 영향을 받는 임시테이블만 재 생성한다. CTIT테이블에서 임시테이블에 대한 비트 연산을 하여 변경된 기본 테이블에서 파생된 임시테이블을 갖고 있는 클라이언트를 조사하고, 검색된 클라이언트에 재 생성된 임시테이블을 전송하면 각 클라이언트에서 뷰에 대한 직접적인 갱신을 한다.

5. 결론

본 논문에서는 공간 데이터에 대한 서버의 부하와 병목현상을 줄여 질의처리 속도를 향상시키기 위한 방법으로 서버의 처리비용을 분산시켜 클라이언트에 공간뷰를 실체화하는 PSCMT 기법을 제안하였다. 이를 위하여 서버와 클라이언트에서 사용되는 자료구조와 전송패킷 구조를 정의하고, 처리과정을 서술하였다. 제안된 기법은 뷰를 생성하기 위한 모든 처리를 서버에서 하지 않고 기본 테이블에 대한 선택선 또는 프로젝션만 수행한 후 결과 데이터를 클라이언트로 전송시켜 클라이언트에서 최종 연산을 수행 후 실체화 시킨다. 따라서 다음과 같은 장점을 갖고 있다. 첫째, 1차적으로 one-table에 대한 연산을 서버에서 처리하여 필요한 데이터만 전송하므로 대부분이 두 테이블 이상에 대한 공간연산에 따른 대용량의 데이터를 전송하지 않으므로 전송부하를 줄일 수 있다. 둘째, 이러한 대용량의 공간 뷰를 클라이언트에 실체화 시킴으로써 서버의 저장공간을 절약한다. 셋째, CTIT의 관리는 서버에 저장된 중간 결과 테이블을 재사용할 수 있으므로 동일 처리에 대해서 서버의 처리 비용을 줄이고, 뷰의 일관성 유지를 용이하게 한다. 마지막으로 클라이언트에 실체화하기 때문에 질의 처리를 각 클라이언트에서 해결할 수 있으므로 서버의 부하와 병목현상을 방지하여 사용자에게 빠른 질의처리를 제공하는 장점을 갖고 있다.

향후 연구과제로는 공간 뷰의 일관성 유지를 위한 임시테이블 관리 전략과 뷰의 갱신 반영을 위한 멀티캐스팅에 따른 부하를 최소화 하는 연구가 이루어져야 한다.

참고논문

- [1] N. Roussopoulos An Incremental Access Method for ViewCache : Concept, Algorithms, and Cost Analysis ACM transactions on Database Systems, Vol.16, No.3 September 1991, pp 535-563
- [2] ADIBA, M, AND LINDSAY, B Database snapshots. In Proceedings of the 6th International conference on VLDB(Monterial,1980), pp 86-91.
- [3] J.A Blakeley, P.Larson and F.W. Tompa Efficiently Updating Materialized Views In SIGMOD 1986 pp 61-71
- [4] A. Gupta, I.S.Mumcik and V.S.Subrahmanian Maintaining Views Incrementally In SIGMOD 1993
- [5] 임덕성, 반재훈, 문상호, 홍봉희 "클라이언트 공간뷰의 실체화 방법" 한국정보과학회 99 봄 학술발표논문집(B) 1999, 04 v.26, pp.101-103
- [6] P.Sevenson "Geo-SAL : A Query Language for Spatial Data Analysis, Proc. of 2nd Sysposium SSD 91, Spring-Verlag, pp.119-140
- [7] C.J.Date, An introduction to Database System, 6th Ed., Addison-Wesley Pub.
- [8] 오영환, 조영섭, 배해영 "지리적 특성에 의한 공간 외부스키마의 설계 및 구현" 한국정보과학회 96 가을 학술발표논문집 v.23, pp283-286