

모바일 클라이언트에 캐쉬된 공간 데이터의 일관성 제어를 위한 로그 인덱싱

차지태⁰ 안경환 전봉기 홍봉희
부산대학교 컴퓨터공학과
(jtcha⁰, khan, bgjun, bhhong}@pusan.ac.kr

Log Indexing for Consistency Control of Cached Spatial Data of Mobile Clients

Ji-Tae Cha⁰, Kyoung-Hwan Ahn, Bong-Gee Jun, Bong-Hee Hong
Dept. of Computer Engineering, Pusan National University

요 약

모바일 클라이언트(Mobile Client, MC) - 서버 환경에서 캐쉬된 공간 데이터는 용량과 가용성이 크고 긴 단절을 가지는 특징이 있다. 이러한 긴 단절을 거친 MC의 일관성 제어시, 캐쉬된 데이터 전체가 무효화되어 재전송 하는 것은 비효율적이므로 서버는 변경된 객체 정보를 로그로 기록하여, 캐쉬된 영역에 속하는 최근 변경 정보만을 전송하는 것이 필요하다. 또한 일반적으로 로그의 순차 검색과 불필요한 로그의 누적으로 인해 일관성 제어시 통신 비용이 증가하게 되므로, 서버의 로그 검색 시간과 전송량을 최소화 하기 위한 기법이 필요하다. 이를 위해 본 논문에서는 로그를 이용한 공간 로그 인덱스(Spatial Log Index, SLI) 기법과 공간 로그 관리(Spatial Log Management, SLM) 기법을 제시하고 있다.

1. 서 론

최근 무선 통신이 가능한 휴대용 단말기(이동 전화, PDA 등)와 모바일 GIS의 발달로, 이를 응용한 서비스에 대한 관심이 증가되고 있다. 모바일 클라이언트(Mobile Client, MC) - 서버 환경에서는 무선 통신의 낮은 대역폭과 MC의 전원 문제로 인하여, MC는 빈번한 단절 상태를 가진다. MC는 단절 상태에서 성능 향상을 위해 서버 데이터 일부분을 캐싱하고 있으며, 캐쉬된 데이터는 재 접속시 일관성 유지가 필요하다. 특히 캐쉬된 공간 데이터는 다른 응용에 비해 그 크기가 크고 무선 환경에서의 긴 단절로 인하여, 일관성 제어시 접근 시간(access time)이 큰 특징을 가진다. 여기서 접근 시간이란 MC가 서버에 연결되어 요청을 시작하는 시점부터 서버의 처리 결과가 MC에 전송이 완료되는 시점까지의 시간을 의미하며[1], 이를 단축시키기 위해서는 서버에서의 처리 시간과 통신 시간의 최소화가 필요하다. 본 논문에서는 일관성 제어시 서버의 처리 시간을 변경 정보의 검색 시간으로 가정하고, 통신 시간은 일관성 유지를 위해 MC로 전달되는 전송량에 비례한다고 가정한다. 또한 통신 비용은 접근 시간에 비례한다고 가정한다.

긴 단절 이후 일관성 제어시, 캐쉬된 공간 데이터가 무효화되어 전체 데이터를 재 전송하는 문제를 해결하기 위해 서버는 변경 정보를 로그로 기록한다. 일반적인 로그는 발생 순서에 따라 순차적으로 기록하고, 저장된 전체 로그를 대상으로 순차 검색을 수행한다. 또한 저장된 로그는 시간이 지날수록 누적된다. 이러한 로그의 특징은 두 가지 문제를 일으킨다. 첫 번째로 검색 시간이 증가하고, 두 번째로 검색 결과에서 불필요한 로그의 비율이 높아져 전송량을 증가시킨다.

이러한 문제점들을 해결하기 위해서 본 논문에서는 공간 로그 인덱스(Spatial Log Index, SLI) 기법과 공간 로그 관리 기법(Spatial Log Management, SLM)을 제시한다. 공간 로그

인덱스는 일관성 제어시 공간 객체의 변경 정보에 대한 검색 시간을 단축시킨다. 공간 로그 관리 기법은 불필요한 로그를 정의하여 삭제 및 여파 알고리즘을 적용함으로써 서버에서 유지되는 로그의 크기와 일관성 제어시 전달되는 전송량을 최소화한다. 본 논문에서는 공간 로그 인덱스와 공간 로그 관리 기법을 통해 통신 비용을 최소화 함으로써 모바일 환경에서의 효율적인 일관성 제어를 달성하고 있다.

2. 관련연구

모바일 환경은 불특정 다수의 모바일 클라이언트를 대상으로 하기 때문에 일관성에 관한 연구는 주로 브로드캐스팅(broadcasting) 기반으로 활발한 연구가 이루어져 왔다. 브로드캐스팅은 크게 주기적 방식과 비주기적 방식으로 나누어 질 수 있으나[1], 이러한 기법들은 일정 시간 동안 서버의 변경 리포트(update report)를 수신하지 못한 경우, MC에 있는 데이터 전체를 무효화시켜야 하는 문제가 있다[2]. 특히, 대용량의 캐쉬된 공간 데이터는 긴 단절 시간을 가지는 특징이 있으므로, 모바일 GIS와 같은 응용에서는 서버의 브로드캐스팅을 수신할 수 없는 경우가 빈번하므로 부적절하다.

로그는 트랜잭션, 동시성 제어, 회복 기법 등 다양한 분야에서 각기 다른 기능으로 사용되고 있다. 또한 분산 환경에서 데이터의 높은 지역성(locality)과 가용성(availability)을 위해 데이터를 중복(replication) 하는데, 이때 중복 데이터의 일관성 유지를 위해 로그를 사용하기도 한다[3]. 일반적으로 로그는 대용량의 순차 구조로 정의된다. 이러한 로그 구조는 회복 과정에서 비효율적인 작업을 초래하므로, 빠른 회복을 위해 로그에 대한 함수를 재정의하고 대용량 로그의 불필요한 부분을 압축하는 연구가 진행되기도 하였다[4]. 그러나 로그의 빠른 검색을 위한 구조와 동적인 로그 관리에 관한 연구는 거의 이루어지지 않고 있다.

3. 대상 환경

본 논문은 무선 통신 기반의 모바일 클라이언트(MC) - 서버 환경(그림1)을 가정한다. MC는 서버의 공간 데이터 일부를 캐싱하고 있으며 이를 단절 기간 동안 유지하고 사용한다. 본 논문에서는 이러한 MC를 잠재적 MC(Potential MC)라고 정의한다. MC의 일관성 제어는 서버에 재 접속할 때 시작되며, 효율적인 제어를 위해서 서버는 MC에 캐쉬된 데이터에 대한 정보를 유지하고 있다.

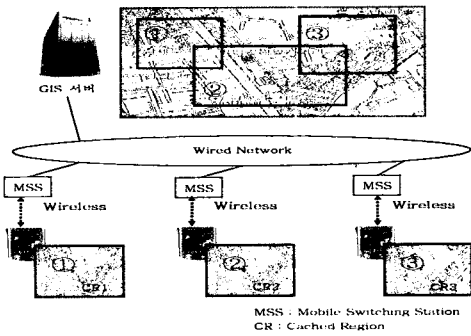


그림 1. 대상 환경

4. 공간 로그 인덱스 (Spatial Log Index, SLI)

서버에 유지되는 로그의 양은 시간이 지남에 따라 크게 증가한다. MC의 일관성 제어시 필요한 변경 정보를 검색하기 위해서 일반적으로 저장된 전체 로그에 대한 순차 검색을 수행한다면 검색 시간이 크게 증가할 것이다. 특히 MC에 캐쉬된 공간 데이터는 2차원의 영역 단위로 저장되어 있기 때문에, 서버에서 공간 로그 검색을 위한 공간 인덱스 구조가 반드시 필요하다. 본 논문에서는 공간 로그의 빠른 검색을 위하여 공간 로그 인덱스(Spatial Log Index, SLI) 기법을 제시한다. SLI는 로그가 반영한 변경 객체의 영역 정보인 MBR로 공간 인덱싱 한다.

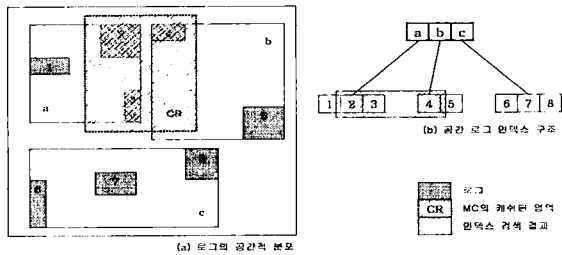


그림 2. 로그 인덱싱

위 그림 2의 (a)는 저장된 로그의 공간적 분포의 예를 보여주고 있고, 그림 2의 (b)는 (a)의 로그들이 공간 로그 인덱스를 구성하고 있는 예를 보여주고 있다. SLI는 변경된 공간 로그가 지역적으로 집중된 분포를 가질 수 있기 때문에 균형 트리인 R-tree 계열의 공간 인덱스를 사용한다.

4.1 공간 로그 검색

로그를 이용한 일관성 제어시 서버의 처리 시간은 대부분 대용량 로그 검색 시간에 의해 좌우된다. 로그 검색은 다음 두 가지 경우에 발생할 수 있다. MC의 일관성 제어시 캐쉬된 영역 정보를 이용한 영역 질의와 로그 관리를 위해 로그 매니저에 의해 수행되는 로그 검색이 있다. 로그는 객체의

영역 정보인 MBR에 의해서 2차원 공간 인덱싱 되므로 공간 질의시 보다 빠른 결과를 얻을 수 있다.

단절 이후 재 접속한 MC의 일관성 제어를 위해 서버는 해당 MC에 캐쉬된 영역 정보로 SLI에 영역 질의하여 로그 정보를 검색한 후, 불필요한 로그를 여과 과정을 통해 제거한다. 이러한 일련의 과정을 통해 MC에게 전달된 정보에는, MC의 최근 단절 시간 동안 캐쉬된 영역 내에서 발생한 변경 정보만을 포함하게 된다.

4.2 로그 구조

서버는 변경된 공간 데이터에 대한 정보를 로그로 기록한다. 로그는 변경 객체의 메타 정보와 변경된 값으로 구성되며, 빠른 검색을 지원할 수 있는 구조(그림3)로 정의된다.

LID	OP	OID	TS	REF	VALUE
LID	Log Identifier				
OP	Operation of Log				
OID	Updated Object Identifier				
TS	Creation Time of Log				
REF	Number of MCs that will refer this log				
VALUE	Updated Object Value				

그림 3. 로그 구조

5. 공간 로그 관리(Spatial Log Management, SLM)

SLM의 목적은 필요 없는 로그를 제거하여 검색 시간을 줄이고, MC로의 불필요한 정보의 전달을 방지하는 것이다. 본 논문에서는 아래 표 1과 같이 불필요한 로그를 정의한다.

로그	정의	처리 기법
낡은 로그 (Obsolete Log)	모든 참조 가능성이 있는 MC(잠재적 MC)에 의해서 참조된 로그.	Log Purging
무관심 로그 (Indifferent Log)	MC에 의해서 참조된 가능성이 없는 로그. 즉, 모든 잠재적 MC의 캐쉬된 영역과 겹치지(overlap) 않는 로그.	Log Avoiding
중복된 로그 (Redundant Log)	삭제 로그 삽입시 같은 OID를 가지는 기존에 저장된 삽입 로그	Log Replacing
	MC의 로그 검색시 같은 OID를 가지는 로그가 3개 이상인 경우, TS가 가장 작은 로그와 가장 큰 로그를 제외한 모든 로그	Log Filtering

표 1. 불필요한 로그

5.1 로그 생성

서버측 데이터에 변경이 발생하는 경우, 참조 가능성을 고려해 변경 정보를 로그로 생성해야 한다. 참조 가능성이란 해당 로그를 참조할 잠재적 MC의 존재를 의미한다. 로그의 이름은 OP 필드의 값에 따라 결정되는데, OP가 INSERT인 것을 삽입 로그, DELETE인 것을 삭제 로그라고 한다.

5.1.1 객체 생성시 로그

새로운 객체가 생성되는 경우 삽입 로그를 생성하게 된다. 동일 객체의 이전 로그가 없으므로 참조 가능성만 검사하여 무관심 로그인 경우 로그를 생성하지 않는다.

```

Create_Insert_Log(Obj) {
    // 참조 가능성 있는 MC의 개수를 파악
    Ref = CountReferenceMC(Obj, MBR);
    // 참조 가능성을 검사하여, 무관심 로그면 생성 방지
    if ( Ref == 0 ) return;
    Log = CreateLog(Ref, INSERT, Obj)
    InsertLogToSLI(Log);
}
    
```

그림 4. 삽입 로그 생성 알고리즘

5.1.2 객체 삭제시 로그

기존 객체가 삭제되는 경우, 동일 객체에 대한 삽입 로그가 존재하는지 검사하여 있으면 이전 삽입 로그를 삭제 로그로 갱신하고, 없으면 참조 가능성을 검사하여 삭제 로그를 생성하여 삽입하게 된다. 이 과정에서 중복 로그가 존재하는 경우 제거된다(Log Replacing).

```

Create_Delete_Log(Obj) {
    // 동일 객체에 대한 이전 로그 검사
    Prv_Log = FindLogAtSLI(Obj.MBR);
    // 참조 가능성 있는 MC의 개수를 파악
    Ref = CountReferenceMC(Obj.MBR);
    // 이전 로그가 없는 경우
    if(Prv_Log is NULL) {
        // 참조 가능성을 검사하여, 무관심 로그면 생성 방지
        if ( Ref == 0) return;
        Log = CreateLog(Ref, DELETE, Obj);
        // 로그 인덱스에 삽입한다.
        InsertLogToSLI(Log);
    } else {
        // 이전 로그가 삽입로그면 삭제로그로 갱신하고,
        // 삭제로그인 경우, 그대로 둔다. (Log Replacing)
        if(Prv_Log is INSERT_LOG)
            ReplaceToDeleteLog(Prv_Log, Ref, Obj);
    }
}
    
```

그림 5. 삭제 로그 생성 알고리즘

5.1.3 객체 수정시 로그

객체가 수정되는 경우, 변경 이전 값에 대한 로그와 변경 이후 값에 대한 로그를 각각 기록해야 한다. 이전 값은 삭제 로그로, 이후 값은 삽입 로그로 만들며 각 과정은 객체의 삭제 및 삽입 과정과 동일하다. 그림 6에서 T1 이전에 일관성 유지된 MC3가 단절 상태를 거쳐 T5 이후 다시 일관성 유지할 때, 로그 L1이 없다면 객체 O1에 대한 변경 로그를 영역 질의를 통해 검색할 수 없게 된다. 그러므로 수정이 발생한 경우, 변경 전 값과 변경 후 값을 모두 인덱싱 하고 있어야 한다. MC마다 캐쉬된 영역과 일관성 유지된 시간이 다르므로, 동일 객체가 여러 번 수정된 경우, 같은 OID를 가진 로그가 한 개 이상 저장될 수 있다. 그러나, 그림 6과 같이 동일 객체(O1)가 여러 번 변경되는 경우, 기존 객체의 중복 로그인 L2는 삭제되어야 한다(Log Replacing).

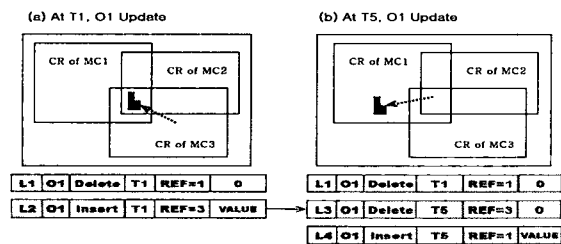


그림 6. 로그 생성의 예

5.2 로그 삭제

본 논문에서는 서버에서 유지해야 될 로그의 양을 줄이기 위한 세 가지 기법을 제시하고 있다. 첫 번째로, 로그 생성 전에 무관심 로그(Indifferent Log)인 경우 생성하지 않는다(Log Avoiding). 그림 7의 (a)에서 객체가 수정되는 경우 삭제 로그는 무관심 로그이므로 생성하지 않는다. 두 번째로, 참조 가능성이 있는 모든 MC에서 참조한 낡은 로그(Obsolete

Log)는 삭제한다(Log Purging). 그림 7의 (b)에서 MC3이 T15에서 로그 L1을 읽은 후, 더 이상 참조할 MC가 없는 로그 L1은 삭제된다. 마지막으로, 5.1.2와 5.1.3에서 설명한 Log Replacing 기법이 있다. 이와 같은 불필요한 로그 삭제를 위해서 서버는 참조 가능한 MC들의 캐쉬된 데이터에 대한 정보를 관리하고 있어야 한다. 첫 번째 방법(Log Avoiding)은 로그의 공간적 참조 가능성을 검사하여 삭제하는 과정으로, 이를 위해 서버는 MC에 캐쉬된 영역 정보가 필요하다. 두 번째 방법(Log Purging)은 로그의 시간적 참조 가능성을 검사하여 삭제하는 과정으로, 서버는 MC의 가장 최근 일관성 유지된 시간 정보가 필요하다.

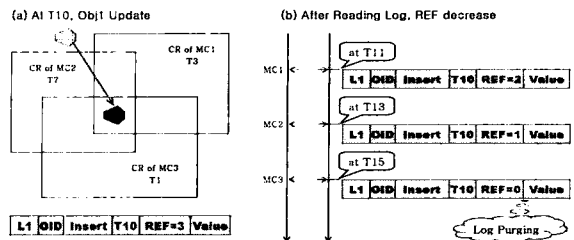


그림 7. Log Purging

5.3 로그 여과 (Log Filtering)

로그 여과란 영역 질의로 검색된 결과에서 요청한 MC에게 필요 없는 로그를 제외시키는 과정이다. 먼저 최신 정보만 추출하기 위해 로그의 TS로 여과 한 후, 동일 객체의 중복 로그에 대하여 여과 한다. 로그 여과 기법에 의해 무선 통신을 통해 MC로 전달되는 전송량을 줄여 통신 시간을 최소화한다.

6. 결론 및 향후 연구

본 논문은 긴 단절 시간을 거치는 MC의 캐쉬된 공간 데이터의 무선 통신을 통한 효율적인 일관성 제어를 위해 공간 로그 인덱스와 공간 로그 관리 기법을 제시하고 있다. 공간 로그 인덱스를 통해 로그의 검색 시간을 최소화하고 공간 로그 관리 기법을 통해 저장되는 로그의 양과 MC로 전송되는 전송량을 최소화하여 MC의 통신 비용을 최소화한다.

향후 연구로 방대한 MC들의 캐쉬 정보를 서버에서 유지하는 좀 더 정교한 기법에 대한 연구가 필요하다. 또한 실제로 무선 통신으로 접속한 다수의 MC에 대한 일관성 제어가 서버에 의해 효율적으로 처리되고, 최소한의 통신 비용으로 이루어지는지 실험을 통해 평가하고 검증해야 할 것이다.

참고 문헌

[1] Kiar-Lee, Jun Cai, Beng Chin Ooi, "An Evaluation of Cache Invalidation Strategies in Wireless Environments," IEEE Transaction On Parallel and Distributed Systems, vol. 12, no. 8, AUGUST 2001
 [2] A. Kahol, S. Khurana, Sandeep S.S. Gupta, Pradip K. Srimani, "A Strategy to Manage Cache Consistency in a Disconnected Distributed Environments," IEEE Transaction On Parallel and Distributed Systems, vol. 12, no. 7, JULY 2001
 [3] M. Ruffin, "A Survey of Logging Uses" Proceedings of the 11th SRDS, Oct, 1992
 [4] B. Weiler, "SpeedLog : A Generic Log Service Supporting Efficient Node-Crash Recovery", IEEE, 1994