2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

모바일 GIS를 위한 양방향 동기화

차지태°, 안경환, 전봉기, 홍봉희
부산대학교 컴퓨터공학과
email : {jtcha, khan, bgjun, bhhong}@hyowon.pusan.ac.kr

Two-Way Synchronization For Mobile GIS

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

요 약

최근 모바일 환경을 기반으로 서버와 단절 상태에 있는 모바일 장치들과 수시로 변경되는 서버간의 데이타 동기화(Data Synchronization) 문제가 크게 부각되고 있다. 기존 객체 단위의 동기화 기법은 모바일 지리정보시스템(GIS : Geographic Information System)의 영역 단위 데이타 동기화에 적용하기에 적합하지 않다. 이 논문에서는 모바일 클라이언트와 서버간에 무선통신 기반으로 양방향 공간 데이타의 실시간 동기화 프로토콜을 제시한다. 또한 무선통신의 좁은 대역폭과 고비용을 고려하여 서버와 모바일 클라이언트간의 효율적인 양방향 동기화를 위한 로깅 기법을 도입하여, 모바일 클라이언트에 분산된 지역 공간 데이타를 실시간으로 서버와 동기화시키는 기법을 제시한다.

1. 서론

최근 모바일 환경이 급증함에 따라서 서버와 모바일 클라이언트(Mobile Client, MC)간의 데이타 동기화가 부각되고 있다. 현재 나와있는 동기화는 주로 서버의 데이타를 MC에게 보내주는 단방향 동기화로, 변경 내용을 반영하기 위해서 전체 비교 대상 데이타를 객체 단위로 비교하거나, 혹은 비교 작업 없이 단순히 영역 복사하기 때문에 GIS의 영역 단위 공간 데이타에 적용하기에는 비효율적이다. 양방향이란 MC에 저장된 지역 데이타에 대하여 단절상태(disconnect) 하에서 변경 작업이 발생함을 전제로 하며, 이는 동기화를 통해 이동 중에 서버의 데이타베이스에 대한 실시간 변경을 가정한다. 현재 모바일 GIS 분야에서 MC와 서버간 공간 데이타의 양방향 동기화 문제에 대한 해결 방법과 그에 대한 기법은 나와있지 않다. 고용량 공간 데이타의 양방향 동기화를 최적화 시키기 위해서는 서버에 동기화를 위한 변경 로그의 작성과 관리 기법이 요구되고, 효과적인 통신 프로토콜이 필요하다.

모바일 장치의 하나인 PDA(Personal Digital Assistant) 역시 제약적인 자원과 통신 환경을 가지고 있다. 이러한 PDA의 GIS 응용 프로그램이 전체 서버 공간 데이타(D_S)의 부분 영역(D_M)을 복사하여 자신의 저장장치에 지역 공간 데이타(D_{ML})로 저장한다고 가정하자. 이 때 MC가 D_{ML}을 서버와 양방향 동기화를 통해 최신으로 유지하고자 한다면, D_{ML}에 대한 시간과 영역 정보로 생성된 질의를 통해 최신 변경 정보(difference)만으로 동기화를 이룬다면 제약적인 모바일 환경에 효율적일 것이다. 이를 위해서 서버의 변경 로그를 저장하고 관리하는 기법과 변경 로그에서 MC의 요구 내용을 빠르게 검색하기 위해 시공간적 정보를 활용하는 기법을 제시한다.

2. 관련연구

모바일 GIS에서 무선을 통한 동기화에 대한 연구는 이루어지고 있지 않지만, 현재 SyncML이라는 동기화 프로토콜 표준이 정의되어 있고, 많은 분야에서 참여하고 있다[1].

2.1 동기화 정의

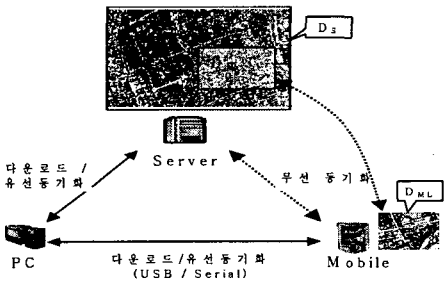
동기화(Synchronization)란 서로 다른 장치 및 서로 다른 응용 프로그램 간의 데이타를 일치시켜 주는 기술을 의미한다[2]. 동기화라는 개념이 나오게 된 주요한 배경은 모바일 환경에서 서버와의 단절

상태의 발생이다. 단절 상태 하에서 서로 다른 장치들 간에 중복되어 있는 데이타에 대한 변경이 발생하고 이 변경 내용들을 서로 일치시켜야 하는 필요성이 생겨나게 되었다.

2.2 SyncML

최근 동기화에 관한 SyncML 표준 프로토콜이 발표되었다[1]. 이것은 주로 다양한 모바일 장치와 서버간에 무선통신으로, 중복된 데이타베이스의 동기화를 이루는 기술이다. 그러나 이 기술은 MC와 서버간에 공통된 로그를 바탕으로 로그에 등록된 데이타의 변경을 field-by-field로 비교함으로써, 상대적으로 비교 대상이 많은 GIS 공간 데이타에는 비효율적이고, 공간 데이타의 추가, 수정, 삭제에 의한 변경이 있는 경우 적용하기 어렵다. SyncML의 프로토콜에서는 7가지의 서로 다른 동기화 타입을 정의하고 있다. 그 중에서도 이 논문에서 제시할 프로토콜과 유사한 양방향 동기화를 "MC의 변경을 먼저 전달함으로써 동기화를 시작하고, MC와 서버가 서로의 변경을 교환한다."라고 정의하고 있다.[2]

3. 대상 환경



[그림1] 대상환경

이 논문에서는 [그림1]에서와 같이 두 가지의 대상 환경을 전제로 한다. 첫번째는 MC와 서버의 무선 통신을 이용한 동기화 방법이고 두번째로 MC가 유선 연결된 PC를 통해 서버와의 동기화를 획득하는 방법이다. 대용량의 공간 데이타를 서버로부터 가지고 올 때는 PC와의 유선 연결을 이용하여 PC로부터 복사하는 방식을 선택한다. 서버와 MC는 1:다 관계를 가지고, MC들은 D_S 일부를 중복해서 저장하고 있다. MC는 단절상태로 작업을 수행하다가, 동기화 시점에서 서버에 연결된다. 서버의 공간 데이타을 변경하기 위해서는 권한을 획득해야 하며, 권한이 주어진 영역은 겹치지(Overlap) 않음을 전제한다. 권한 사용자들에 의한 변경은 동기화 시점에서 서버의 공간 데이타베이스에 즉시 반영된다. 일반 사용자들은 비교적 넓은 영역을 MC의 지역 저장 장치에 저장하고 있고, 이 D_{ML}의 동기화를 위해 통신 비용을 줄이고 시간을 단축하는 기법을 제시하는 것이 이 논문의 주요 대상 환경과 목적이다.

4. 동기화 기법

이 논문에서는 서버의 공간 데이타에 수시로 변경이 발생하고 이를 다수의 MC와 동기화 하는 양방향 동기화 기법을 제시하며, 동기화를 다음과 같이 정의한다.

D_S(서버 전체 데이타집합), D_{ML}(MC 데이타집합)
$D_S \supseteq D_M$, $D_M = D_{ML}$: 최초 상태
$D_M' \rightarrow D_M'$, $D_{ML}' \rightarrow D_{ML}'$: 단절상태에서 변경발생
$D_M' = D_{ML}'$: 동기화 완료 상태

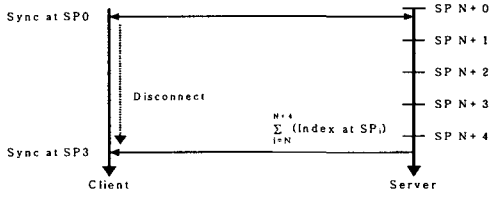
MC와 서버의 빠른 동기화를 위해서 변경 로그를 위한 인덱스를 유지하고 관리하는 기법을 제시한다. 또한 대용량의 공간 데이타베이스의 변경 로그 역시 크고 중복이 많이 발생하므로 이것을 해결하기 위해서 SyncPoint 개념을 도입하고 인덱스를 병합(Merge)하는 기법을 제시한다.

4.1 SyncPoint

MC는 서버와 동기화 이후 단절 상태를 거친 다음 동기화를 위해 다시 서버에 연결된다. 단절상태에서 서버는 계속해서 변경 로그를 저장하며, MC는 D_{ML}에 대한 변경 정보를 요청하는 질의를 한다. 이 때 서버의 MC 영역 전체의 변경 로그를 검색해야 하는데, 이것은 다음과 같은 이유로 비효율적이다.

- ▷ 해당 영역의 모든 변경 정보를 검색해야 한다.
- ▷ MC에서 질의한 영역에 대해 변경 정보가 중복해서 검색되고 전달된다.

이러한 문제를 해결하기 위해서 SyncPoint라는 개념을 도입한다. 서버는 일정 간격으로 SyncPoint를 생성하고, 각 SyncPoint 별로 변경 정보를 검색하기 위한 인덱스를 유지한다. MC는 자신의 최근 SyncPoint를 기억하고, 다음 동기화 시점에서 자신의 최근 SyncPoint를 이용해 질의하여 이후의 변경 정보를 전달 받을 수 있다. 즉, SyncPoint는 MC의 D_{ML}에 대하여 서버가 필요한 최신의 변경 정보만을 빠르게 검색할 수 있게 한다.



[그림2] SyncPoint

MC가 만약 SP_N에서 동기화를 했다가 단절상태 후에 다시 동기화를 요구한 시점이 SP_{N+4}이라면 서버는 MC 영역에 대하여 $SP_N \sim SP_{N+4}$에 해당하는 인덱스만을 검색하여 변경 로그를 찾아내면 된다. 여기서 SP_N가 포함되어 중복된 변경 정보를 찾아내는 문제가 있는데, 이후에 자세히 설명한다.

SyncPoint는 서버의 SyncPoint 매니저가 관리한다. 이는 SyncPoint를 생성하고, 생성 주기를 관리하며 다음에서 설명할 변경로그 인덱스 병합하는 주기 역시 관리한다.

4.1.1 SyncPoint 주기

서버측 SyncPoint 매니저는 SyncPoint를 생성하는 주기와 조건을 결정하게 되는데, 이를 결정하는

요소는 여러 가지가 있을 수 있으며 시스템 성능에 영향을 미친다. 예를 들어서 주기가 짧으면 변경 정보가 많고 수시로 변하는 환경에서 동기화를 요구하는 MC에게 최적의 변경 정보만을 검색하여 줄 수 있으나 시스템의 연산량이 증가하는 단점이 있다. 반면 긴 주기는 변경 정보의 양이 많지 않을 경우에 적합하나, 같은 주기에 두 번의 동기화를 요구하는 MC에게 중복된 변경 정보를 전달하는 문제점이 있다.

4.1.2 SyncPoint 생성 조건

SyncPoint의 주기는 일정 간격으로 또는 특정 조건이 발생할 때마다 생성될 수 있다.

▷ 일정 시간을 주기로 생성하는 방법

서버의 SyncPoint매니저가 일정 시간을 주기로 새로 생성하는 방법이다. 가장 일반적이며 이 논문에서 가정하는 방법이다. 앞에서 언급한 것처럼 SP_N에서 동기화한 MC가 SP_{N+1}에서 동기화를 요청하면 SP_N에서의 동기화 했던 중복 정보를 읽어오는 문제가 있다.

▷ 특정 조건이 발생하면 생성하는 방법

특정 조건이란 예를 들어서 변경 정보가 정해진 임계치 이상으로 발생하게 되면 새로운 SyncPoint를 생성하는 것이다. 앞에서 말한 중복해서 변경 정보를 읽어오는 문제를 해결할 수 있으나 변경 정보가 다수에 의해서 자주 발생하는 경우는 비효율적이다.

4.1.3 SyncPoint의 생성 주기와 조건

주기 \ 조건	일정 시간	특정 조건
짧은 주기	V	
긴 주기		V

[표 1] SyncPoint 생성 조건과 주기

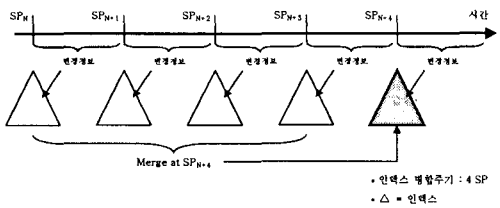
[표1]을 보면, 변경 정보가 빈번히 발생하여 주기가 짧은 경우는 일정 시간 간격으로 SyncPoint를 생성하며, 변경 정보가 빈번히 발생하지 않아서 주기가 긴 경우는 특정 조건이 발생할 때 SyncPoint를 생성하는 것이 좋다는 것을 설명하고 있다.

4.2 변경로그 인덱스(Update Log Index, ULI)

모바일 환경에서 무선통신은 고비용이 소요되므로, MC가 질의하는 영역에 대하여 응답 시간과 전송 정보의 크기를 최소화해야 한다. 이를 달성하기 위해서 서버의 변경 정보를 인덱싱한다. ULI는 매 SyncPoint마다 새로 생성된다. 해당 SyncPoint에서의 변경 정보가 발생하면 새로운 ULI를 만들고 해당 SyncPoint 주기안의 변경 정보만을 인덱싱한다. ULI는 SyncPoint마다 생성되기 때문에 MC가 기억하는 최근 SyncPoint 이후에 생성된 인덱스들만 검색하면 된다. MC는 영역 질의를 하게 되므로 공간 인덱스인 Grid File이나 R-Tree 계열을 사용할 수 있으나, 변경 정보는 도메인 공간이 가변적이고 서버의 전체 공간 데이타(D_S)에 넓게 분포됨을 가정하기 때문에 해쉬 기반의 인덱스보다 트리 기반의 인덱스인 R-Tree 계열이 유리하다[3]. 그러나 본 논문에서는 변경 로그를 위한 인덱스의 종류는 제한을 두지 않는다. MC의 변경 정보가 서버의 데이타베이스에 즉시 반영되고 나서 인덱스에 로그를 삽입하게 되는데, 변경 로그는 변경 객체의 영역(MBR)으로 인덱싱되며, 디스크에 저장된 변경 객체의 주소(address)인 포인터로 구성된다.

4.3 ULI 병합(Update Log Index Merge, ULIM)

서버에서 유지되는 변경 로그는 시간이 지날수록 점점 증가하게 된다. 특히 변경이 빈번한 환경일 경우 그 정도가 더 심하기 때문에 로그 검색 속도가 현저히 감소하게 된다. 이를 해결하기 위해서는 같은 객체에 대하여 중복되는 변경 로그를 인덱스에서 제거하거나 참조되지 않을 영역의 변경 로그는 삭제해야 할 것이다. 같은 객체에 다수의 변경 정보가 있는 경우, 가장 최신의 변경 정보만 의미가 있다. 중복된 변경 로그를 제거하는 방법은 동기화 시점에서 변경 로그를 검색 시간을 줄일 수는 있으나, 변경 로그가 많을수록 서버측의 연산량이 늘어나 부하가 많아지게 된다. 참조되지 않을 영역의 변경 로그 삭제를 위해서는 동기화의 가능성이 있는 MC의 영역을 관리하는 기법이 필요하므로 복잡하고, 다수의 MC가 존재하는 경우 비효율적이다.



[그림3] ULIM (Update Log Index Merge)

새로운 SyncPoint마다 변경 정보가 발생하는 경우, 생성된 인덱스의 수도 SyncPoint의 수에 비례하여 증가할 것이다. 이것은 단절상태 기간이 긴 MC가 동기화를 위해 재 접속하였을 경우 검색해야 할 ULI의 수가 증가함을 의미한다. 위에서 언급한 두 가지 문제를 해결하기 위해서 [그림3]에서와 같이 인덱스를 병합하는 기법을 이용한다. ULIM은 서버가 일정한 시간(N개의 SyncPoint)이 지나면 자동으로 수행할 수 있어야 한다. ULIM 주기는 SyncPoint 생성 주기와 관련이 있고, 변경 로그의 수와도 관련이 있다.

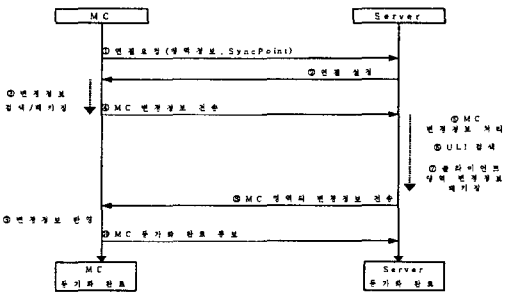
인덱스 병합은 인덱스 매니저가 수행하게 된다. 인덱스가 병합될 때, SyncPoint가 순서가 빠른 것부터 병합에 참여해서 SyncPoint가 느린 순서대로 병합을 진행하게 된다. 이 과정에서 공간 객체의 OID를 검사해서 중복된 객체에 대한 변경 로그는 필요한 최신의 정보로 덮어쓰게(Overwrite) 된다. 이 방법을 통해서 같은 객체에 대한 중복된 변경 로그를 제거하는데, 별도의 연산 없이 덮어쓰기를 하므로 신속하고 부하를 줄이게 된다. 병합에 참여했던 인덱스는 삭제되어 이후 동기화에서 더 이상 참조되지 않는다.

4.3.1 ULIM 주기

앞에서 살펴본 SyncPoint 생성 주기 및 조건과 마찬가지로 ULIM 주기 역시 일정 간격과 특정 조건에 의해서 발생하는 경우로 나눌 수 있다. 일정 간격으로 주기를 결정하는 경우는 SyncPoint가 생성되는 주기에 의해서 결정된다. 그러나 ULIM의 목적이 중복된 변경 정보를 제거하는 것과 검색해야 하는 인덱스의 수를 줄이는 것이므로 특정 조건에 의해서 병합할 경우는 그 목적에 최적화시킬 경우에 선택할 수 있다. SyncPoint 주기와 ULIM 주기는 서로 밀접한 관계를 가지고 서로 영향을 미치게 된다.

5. 동기화 프로토콜

5.1 무선통신을 이용한 서버 연결

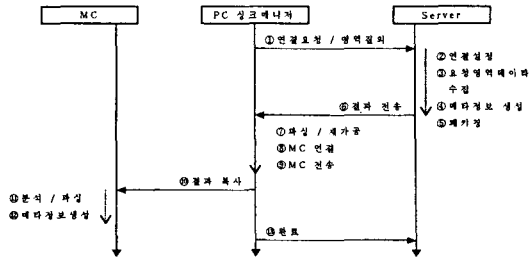


[그림4] 동기화 프로토콜 (Server-MC)

MC는 D_{ML}에 대한 변경을 수행하여 서버와의 동기화를 통해서 서버에 변경 정보를 반영할 수도 있다. MC가 동기화를 위해서 최근 SyncPoint와 영역 정보를 서버에 보내 동기화를 요청하면 서버는 연결을 생성한다. MC가 권한을 가지고 D_{ML}에 변경을 수행한 경우라면 먼저 MC의 D_{ML}의 변경 정보를 서버로 전달하여 서버의 데이타베이스를 즉시 변경하고, 변경 로그를 최신의 SyncPoint에 해당하는 ULI에 추가하게 된다. 서버 측의 동기화가 끝나면, 서버는 MC가 질의한 영역에 대한 변경 정보를 ULI에서 검색한다. 변경 정보 검색은 MC의 영역과 최근 SyncPoint 정보에 의해서 이루어진다. 만약 MC의 최근 SyncPoint가 SP_N이고 서버측의 최신 SyncPoint가 SP_{N+5}라면 서버의 인덱스 매니저는 SP_N에서 SP_{N+5}의 각 인덱스를 순서대로 검색하여 변경 정보를 추출하게 된다. ULI가 그 사이에 병합된 경우라면 병합되기 전의 인덱스는 삭제 되었으므로 검색할 인덱스 수는 줄어들게 된다. 변경 정보 추출 과정에서 중복 변경 정보는 제거된다. 여기서의 중복 변경 정보는 앞에서 말한 인덱스에서 제거가 아니라, MC로 보낼 정보에서 제거된다는 의미이다. 추출된 변경 정보와 서버의 현재 SyncPoint가 MC로 전달되면 MC는 이를 자신의 D_{ML}에 반영하는 작업을 한 후에 동기화를 종료하게 된다

5.2 PC를 통한 서버 연결

MC가 PDA인 경우, PC와 USB(혹은 Serial 포트)를 통해 연결된다. 이 환경에서 MC는 서버로부터 특정 영역 복사와 동기화 두 가지 작업을 수행할 수 있다. 특정 영역의 공간 데이타는 그 용량이 수메가



[그림5] 동기화 프로토콜 (Server-PC-MC, 영역복사)

에서 수십 메가에 이르기 때문에 현재로서는 무선 연결을 이용한 복사는 현실성이 없다. PC의 싱크 매니저는 서버와 MC간의 통신을 중개하여 서버의 공간 데이타를 MC에 이식시키거나, MC의 D_{ML}에 대하여 서버와 동기화하는 작업을 수행한다.

6. 결론 및 향후 연구

이 논문에서 MC와 서버간의 실시간 양방향 동기화 기법을 제시하고 있다. MC와 서버간의 변경 로그를 객체가 아닌 영역 단위로 관리하고, MC는 SyncPoint를 기준으로 최신 변경 정보만을 획득하게 된다. 또한 동기화 속도를 높이고, 로그의 크기를 줄이기 위해서 서버에서 ULI를 운영한다. ULI는 주기적인 병합을 통해서 중복 정보를 제거하고 검색 속도를 높이게 된다. 이 기법을 통해 좁은 대역폭과 고비용이 존재하는 무선 통신 환경에서 분산되어 있는 MC의 D_{ML}을 서버와 효과적으로 동기화 시킬 수 있다. 또한 서버측 대용량 공간 데이타의 일부를 복사하기 위해서 PC에 직접 유선 연결하여 비용을 최소화 하는 기법을 제시하고 있다.

향후 연구에서는 무선 통신을 통하여 서버로부터 영역을 복사하고 동기화 하는 방법이 요구되고, 동기화시 공간 관련성을 유지하는 연구도 이루어져야 할 것이다.

참고 문헌

[1] http://www.syncml.org, SyncML White Page

[2] http://www.syncml.org, SyncML Sync Protocol version 1.1

[3] Volker Graede and Oliver Gunther, "Multidimensional Access Methods", pp21-24, pp43-46

[4] ABRAHAM SILBERSCHATS and HENRY F. KORTH and S. SUDARSHAN, "Database System Concepts", pp511-542, McGraw-Hill