모바일 환경에서 시공간 데이터의 동시 양방향 동기화 기법

김홍기, 김동현
동서대학교

A Scheme of Concurrent Two-Way Synchronizations for Spatio-Temporal Data on a Mobile Environments

Hong-Ki Kim, Dong-Hyun Kim
Dongseo University
E-mail : inthestream@nate.com, pusrover@dongseo.ac.kr

요 약

모바일 기기와 무선 통신 기술이 발달함에 따라 모바일 기기에서 수집 또는 변경되는 대용량 시공간 데이터를 서버와 현장에서 동기화하는 서비스의 제공이 가능해지고 있다. 이러한 현장 동기화 서비스를 제공하기 위하여 다수의 모바일 기기에서 변경된 대용량 시공간 데이터를 서버와 효율적으로 동기화하는 양방향 동기화 프로토콜이 필요하다. 그러나 기존의 프로토콜은 순차적으로 동기화를 처리하기 때문에 다수의 모바일 기기에 대하여 수행할 때 처리 시간이 길어지는 문제가 있다. 이 논문에서는 다수의 양방향 동기화 작업간에 변경 충돌이 발생하지 않는 경우에 서버에서 동기화 작업들을 동시 수행하는 처리기법에 대하여 제안한다.

ABSTRACT

As the mobile devices and the wireless networks have high-performance capabilities, it is possible to synchronize the spatio-temporal data of a server with the spatio-temporal data of a mobile device which are collected at a field. However, since the server process the synchronization which the model device requests, the whole synchronizations of mass mobile devices take long time. In this paper, we propose the scheme to process concurrently the synchronizations of mobile devices which does not conflict with others using the scheme of a multi-queue.

키워드
동기화 프로토콜, ActiveSync, ActMAP, u-GIS

Ⅰ. 서 론

모바일 기기들이 다양화되고 고성능화되었으며 무선 네트워크도 기존에 비하여 대용량의 디지털 데이터를 고속으로 전송할 수 있게 되었다. 따라서 고성능화된 모바일 기기와 무선 네트워크 기술을 기반으로 다양한 형태의 지도 또는 위치 데이터를 수집할 수 있게 되었으며, 특히 기존에 제공하기 어려웠던 대용량의 시공간 데이터를 현장에서 바로 수집하여 가공하고 서비스하는 것이 가능하게 되었다. 이러한 시공간 데이터 서비스는 원격관리, 응급경계 서비스 같은 공공 서비스부터 네비게이션, 교통감시, 물류추적 같은 다양한 서비스 분야에서 활용될 수 있다[1]. 그러나 기존의 서버와 모바일 기기간의 동기화 프로토콜은 대용량의 시공간 데이터의 양방향 동기화에 비효율적이다.

기존의 양방향 동기화 프로토콜은 동기화를 요청한 순서에 따라 순차적으로 동기화하는 프로토콜이다. 순차적 동기화 프로토콜은 동기화 처리시간의 길이에 상관없이 동기화 요청 순서대로 동기화한다. 처리시간의 길이를 고려하지 않은 이 프로토콜은 처리시간이 긴 동기화작업 중에 처리시간이 짧은 동기화 요청이 들어와도 처리중인 동기화가 끝날 때까지 대기하는 문제가 있다. 또 모바일 기기들로부터 특정 시간대에 동기화가 요청이 집중될 경우 동기화 처리작업이 쌓이는 문제가 있다.

이 논문에서는 기존의 양방향 동기화 프로토콜을 기반으로, 동시 양방향 동기화 프로토콜을 제안한다. 동시 양방향 동기화 프로토콜은 다중

큐를 사용하여 변경충돌이 발생하지 않는 동기화 작업들 간에 동시 처리를 지원한다. 제안한 프로토콜을 사용하면, 처리시간이 긴 동기화를 처리중 이더라도 변경충돌이 발생하지 않는 동기화 작업을 대기 없이 동기화 할 수 있다.

Ⅱ. 관련연구

2.1 Exchange ActiveSync

Exchange ActiveSync[2]는 대기 시간이 길고 대역폭이 낮은 네트워크에서 작동하도록 최적화된 Microsoft Exchange 동기화 프로토콜이다.

HTTP와 XML을 기반으로 한 이 프로토콜은 브라우저 기능이 있는 휴대폰이나 Microsoft Windows Mobile® 탑재 장치 같은 모바일 장치에서 Microsoft Exchange를 실행하는 서버에 있는 조직의 정보에 액세스하는 데 사용된다. 모바일 장치 사용자는 Exchange ActiveSync 를 통해 전자 메일, 일정, 연락처, 작업 등에 액세스 할 수 있으며 오프라인 작업 시에도 이 정보를 사용할 수 있다.

2.2 ActMAP

ActMAP 시스템[3]은 유럽 자동차 회사들과 디지털 지도 제작 업체와의 프로젝트 팀에 의해 2007년도에 제안된 네비게이션 지원 시스템으로 자동차에 탑재된 네비게이션 기기에 무선 네트워크를 이용하여 지도의 최신 변경사항을 반영할 수 있도록 지원하는 시스템이다.

ActMAP은 오프라인 방식과 온라인 방식의 변경을 모두 지원한다. 오프라인 방식의 변경은 기존의 방식을 따르지만 무선 네트워크를 이용한 온라인 방식은 여러 제약조건을 고려한다.

첫 번째는 무선 네트워크에서 야기되는 제한된 대역폭이다. ActMAP은 제한된 대역폭 환경에서 전달되는 변경 지도 데이터의 양을 줄이기 위하여 지도 데이터를 계층(Layer)과 분할(Partition)로 구분하여 관리한다. 그리고 변경사항이 발생한 계층의 분할에 포함된 지도 데이터만을 전송하여 변경하는 점진적 부분 변경 방식을 사용한다.

두 번째는 다수의 변경 데이터 제공자와 수요자가 사용하는 고유 데이터 형식이다. 각 제공자와 수요자가 고유 데이터 형식을 사용하기 때문에 호환성이 떨어지고 모든 형식에 대하여 다수의 변환기가 필요한 문제가 있다. ActMAP은 이를 해결하기 위하여 GDF[4] 데이터 모델에 기반 하여 XML을 이용한 표준 데이터 형식을 제공한다.

세 번째는 변경이 반영되기까지 소요되는 지연으로 인하여 변경이 발생한 현장과 변경 내용들이 반영되는 지도 제공자들, 그리고 최종 네비게이션 시스템의 데이터 시간이 각각 다른 문제가 있다. 이를 위하여 ActMAP은 변경 전략을 제시하고 각각의 경우에 대한 시간 모델과 이를 기반으로 한 변경 방법을 제공한다.

마지막으로 지도 데이터 변경됨에 따라 발생하는 변경 데이터의 품질 유지 문제이다. 이를 위하여 ActMAP은 변경을 위한 트랜잭션 개념을 제공하고 변경 데이터간의 레이어·분할 의존성을 고려한다.

Ⅲ. 기존 동기화 프로토콜의 문제점

기존의 모바일 GIS DB를 위한 양방향 동기화 프로토콜은 모바일 기기를 통하여 수집·변경된 GIS 데이터를 서버에 갱신할 수 있는 동기화 프로토콜이다. 이 프로토콜은 현장에서 수집하거나 변경한 GIS 데이터를 모바일 기기에서 유·무선으로 서버와 동기화를 지원한다. 현장에서 수집·변경 GIS 데이터는 모바일 GIS DB에 바로 적용시킬 수 없다. 모바일 GIS DB의 갱신은 반드시 서버를 통해서만 가능하다. 다수의 모바일 기기에서 동기화를 요청하였을 경우에는 동기화 요청 순서에 따라 순차적으로 동기화한다. 이 방식은 긴 동기화가 처리중일 경우 처리작업이 짧은 동기화 요청이 들어와도 처리중인 동기화가 끝날 때까지 대기해야한다.

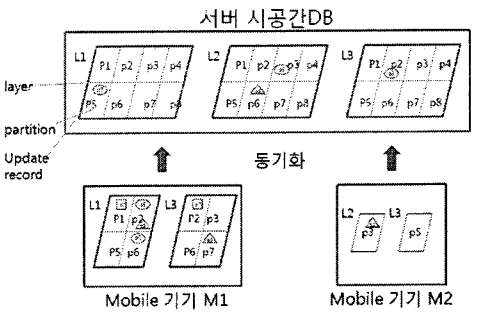

그림 1) M1, M2 동기화 요청

그림 1)에서 M1은 6개의 GIS 데이터를 동기화 하고, M2는 1개의 GIS 데이터만을 동기화한다. 그림 2)는 그림 1)의 동기화 요청을 기존의 동기화 프로토콜을 이용하여 순차적으로 동기화 하였을 때의 프로세스 처리의 시간 모델이다. M1이 t1시간에 동기화를 요청하여 동기화 중이다. 이 때 M2가 t2시간에 동기화를 요청 하였지만, M1의 동기화가 끝난 t3시간까지 대기하였다가 동기화를 시작한다. M2는 t3시간에서 t4 사이의 시간이면 동기화가 완료되지만, M1의 동

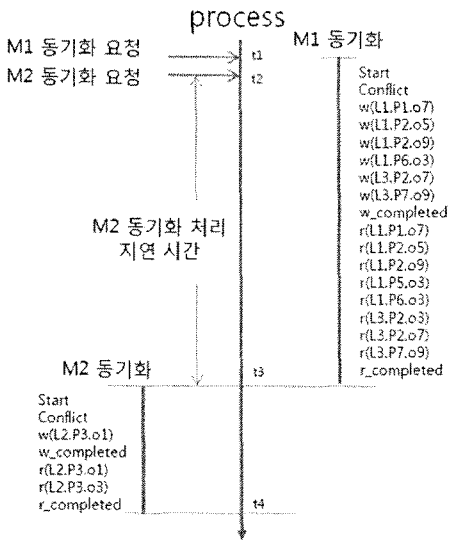

그림 2) 순차적 동기화 처리 Time모델

기화가 끝날 때 까지 대기한 t2시간에서 t3시간까지의 지연시간으로 동기화완료까지 t2시간에서 t4시간까지의 시간이 소요되었다. 순차적인 동기화 처리는 M2와 같이 자신의 동기화 처리 시간보다 긴 시간을 대기하는 문제가 있다.

Ⅳ. 제안하는 양방향 동기화 프로토콜

이 논문에서 제안한 동시 양방향 동기화 프로토콜은 둘 이상의 동기화 요청작업에 대하여 변경충돌[4]이 발생하지 않는 경우 동시 동기화를 지원하는 기법이다. 다중 큐를 이용하여 변경충돌이 발생하지 않는 동기화 작업을 서로 다른 큐에 삽입하여 라운드 로빈 방식으로 동기화 처리하는 프로토콜이다.

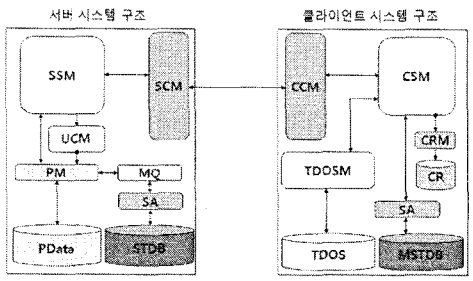

그림 3) 전체 구조도

동시 양방향 동기화 프로토콜은 그림 3)과 같이 구성된다.

4.1 동시 동기화 시나리오

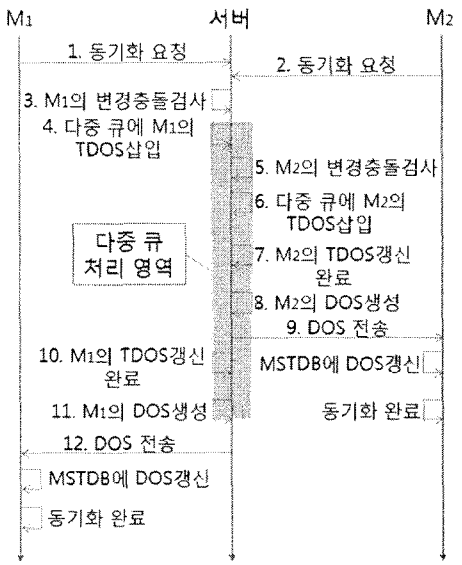

그림 4) M1과 M2 의 동시 동기화 시나리오

그림 4)는 그림 1)의 동기화 요청을 제안한 동기화 프로토콜로 동기화하는 시나리오이다.

1. M1에서 동기화를 요청한다. 동기화할 데이터는 임시 델타 레코드 셋(Temporary Delta recOrd Set, TDOS)이라 정의한다.
2. M2에서 동기화를 요청한다.
3. 먼저 동기화를 요청한 M1을 변경충돌검사를 한다. 변경충돌검사는 변경충돌검사 매니저(Update Conflict Manager, UCM)에서 한다.
4. 변경충돌이 없는 경우 다중 큐에 삽입, 변경충돌이 있을 경우 동기화를 철회한다.
5. 먼저 동기화를 요청한 M1의 TDOS가 큐에 삽입되었음으로, M2의 변경충돌검사를 한다. M2의 변경충돌검사는 큐에서 동기화 처리중인 M1과도 변경충돌검사를 한다.
6. 서버와 M1과의 변경충돌이 없을 경우, 다중 큐에 M2의 TDOS를 삽입한다.
7. 처리작업이 짧은 M2의 TDOS의 갱신완료.
8. M2에 동기화할 델타 레코드 셋(Delta recOrd Set, DOS)을 생성한다.
9. DOS를 M2로 전송한다. M2는 전송받은 DOS를 모바일 시공간 데이터베이스(Mobile Spatio-Temporal DataBase, MSTDB)에 갱신하고 동기화를 완료한다.
10. M1의 TDOS갱신완료.
11. M1의 DOS를 생성한다.
12. DOS를 M1로 전송한다. M2는 전송받은

DOS를 MSTDB에 갱신하고 동기화를 완료한다.

4.2 변경충돌검사

이 절에서 사용되는 표기는 다음과 같다.

- SP_k : 서버의 분할P_k
- CP_k: 클라이언트의 분할P_k
- $LST(P_k)$: 클라이언트P_k의 최종동기화시간
- $LUT(P_k)$: 서버P_k의 최종갱신시간
- $OS(SP_k)$: SP_k의 레코드 집합
- $WS(CP_k)$: CP_k의 DB에 갱신하기위한 레코드의 집합
- ST : 동기화를 요청한 트랜잭션
- QT : 큐에서 처리중인 트랜잭션

갱신을 위한 동기화 요청이 들어오면, UCM에서 복사영역(Copy Region, CR)과 분할 데이터(Partition Data, PData)의 같은 분할에 대해서 최종동기화시간(List Sync Time, LST)와 최종갱신시간(List Update Time, LUT)를 비교한다.

·LUT(Pk) <= LST(Pk) 이면, 변경충돌 없음.
·LUT(Pk) > LST(Pk) 이면, 변경충돌 발생가능.

변경충돌이 발생가능일 때, LST(Pk)에서 LUT(Pk)사이의 WS(CPk)와 OS(SPk)을 검사한다.

·WS(CPk) ∩ OS(SPk) ≠ 0 이면, 변경충돌 없음.
·WS(CPk) ∩ OS(SPk) = 0 이면, 변경충돌 발생, 동기화를 철회한다.

큐에 처리중인 트랜잭션이 있을 때 아래의 조건도 추가로 검사한다.

·LST(QT) <= LST(ST) 이면, 변경충돌 없음.
·LST(QT) > LST(ST) 이면, 변경충돌 발생가능.

변경충돌이 없으면, 큐의 삽입하고, 변경충돌이 발생가능이면 아래 조건을 비교한다.

·WS(QT) ∩ WS(ST) ≠ 0 이면, 변경충돌 없음.
·WS(QT) ∩ WS(ST) = 0 이면, 변경충돌 발생. 동기화를 철회한다.

4.3 다중 큐

3개의 큐에 동기화 작업을 분산하여 라운드 로빈 방식으로 처리하는 것이 다중 큐의 기본개념이다. 3개의 큐 각각을 A, B, C라 정의한다.

갱신을 위한 동기화의 규 삽입 알고리즘
①3개의 큐가 모두 비어 있을 때, A에 동기화 작업을 삽입한다.
②1개의 규에 작업이 있을 때, 알파벳 순서에 따라 비어있는 큐에 동기화 작업을 삽입한다.
③1개의 규가 비어 있을 때, 비어있는 큐에 동기화 작업을 삽입한다.
④모든 큐에 작업이 있을 때, 길이가 가장 짧은 큐에 동기화 작업을 삽입한다.

읽기를 위한 동기화의 규 삽입 알고리즘
①②③은 갱신을 위한 동기화의 규 삽입과 같다.
④의 경우 먼저 삽입된 갱신동기화 작업들 중 같은 객체에 대한 읽기가 있을 경우, 큐의 길이에 상관없이 같은 객체의 갱신동기화 작업이 있는 큐에 삽입한다.

큐 삭제 알고리즘
①1개의 큐에만 작업이 있을 경우, 작업이 있는 큐에서만 삭제한다.
②2개의 큐에 작업이 있을 경우, 2개의 큐에 대해서 라운드 로빈 방식으로 삭제한다.
③3개의 큐에 작업이 있을 경우, 3개의 큐에 대해서 라운드 로빈 방식으로 삭제한다.

V. 결 론

이 논문에서는 기존의 양방향 동기화 프로토콜에 다중 큐를 적용하여 동기 동기화 프로토콜을 제안하였다. 제안한 프로토콜은 다중 큐에 들어 있는 작업들을 라운드로빈 방식으로 병행 처리하여 변경충돌이 없는 동기화간 동시 동기화하는 것이다. 향후 연구로는 시스템 구현 및 실험을 통하여 실제 처리시간의 효율성에 관한 연구가 필요할 것이다.

참고문헌

[1] 류석상, "디지털 컨버전스로 나타나는 유비쿼터스 사회,"한국전산원 u-전략팀, 2005.9
[2] Exchange ActiveSync , "http://technet.microsoft.com/ko-kr/library/aa998357(EXCHG.80).aspx"
[3] "ActMAP White Paper and Interfaces to the FeedMAP framework," white paper, 2007
[4] 최우영, 이경아, 염태진, 진성일,"내장형 DBMS를 위한 동기화 서버 시스템" 한국정보과학회, 한국정보과학회 학술발표논문집 한국정보과학회 2004년도 봄 학술발표논문집 제31권 제1호(B), 2004. 4, pp. 127 ~ 129