

다수 클라이언트의 동시 동기화 프로토콜 설계 및 구현

김흥기, 임창우, 이상신, 김동현

동서대학교

A Design and Implementation of Concurrent Synchronizations Protocol of Multiple Client

Hong-Ki Kim, Chang-Woo Lim, Sang-Sin Lee, Dong-Hyun Kim

Dongseo University

E-mail : inthestream@nate.com, discownt@nate.com, bennsera@nate.com, pusrover@dongseo.ac.kr

요 약

순차적으로 동기화를 처리하던 기존의 동기화 프로토콜은 다수의 클라이언트에 대해 뒤에 요청된 동기화 작업들이 대기하는 문제가 발생한다. 이 논문에서는 순차적 동기화로 발생하는 대기시간을 감소시킬 수 있는 동시 동기화 프로토콜을 구현 하였다. 서로 다른 동기화 작업 간 변경충돌이 없는 경우 동시 동기화 프로토콜은 다중 큐를 이용하여 라운드 로빈 방식으로 동기화작업을 진행하여 대기시간을 감소시킨다.

ABSTRACT

The existing protocol which processes sequentially synchronization has the problem where synchronization tasks which are requested subsequently from multiple client. Should be waited for a long time. In this paper, we propose a implementation of concurrent synchronization protocol that is capable of reducing the waiting time. If there are no update conflicts among different synchronization tasks, the proposed protocol synchronizes using the multi-queue based on the round-robin scheme, in order to reduce the waiting time.

키워드

동기화 프로토콜, Mobile, Multi-queue

1. 서 론

동기화는 컴퓨터와 컴퓨터 또는 컴퓨터와 단말기 사이에서 공유데이터 변경이 발생하면 상대 컴퓨터나 단말기로 갱신하는 작업이다. 유럽에서는 변경된 도로정보의 신속한 적용을 위해서 ActMAP[1]시스템을 개발하였다. 이 시스템은 서버의 지리데이터에 변경이 발생하면 무선통신을 이용하여 실시간으로 네비게이션 단말기로 변경 데이터를 전송하여, 단말기의 지리데이터를 최신의 데이터로 갱신하는 시스템이다.

[2]에서는 모바일환경에서 공간데이터를 양방향으로 동기화하는 프로토콜을 제안하였다. 이 프로토콜은 클라이언트에서 변경한 공간데이터를 현장에서 바로 서버로 갱신할 수 있다. 그러나 서버-클라이언트 시스템의 경우 하나의 서버에 다수의 클라이언트가 동기화 작업을 요청할 수 있다. 이러한 순차적 동기화 프로토콜은 먼저 동기화를

요청한 작업이 긴 경우에 뒤에 요청된 동기화 작업들이 긴 대기 시간을 가지는 문제가 발생한다.

예를 들어 우체국에서 모바일 기기를 이용하여 우체통 및 우편물 등을 관리하는 시스템을 사용한다고 하자. 우체통 및 우편물을 관리하는 모든 우편배달원들이 모바일 기기를 이용하여 동기화를 요청할 수 있다. 그러한 경우 먼저 동기화를 요청한 우편배달원의 동기화 작업이 끝날 때 까지 다른 배달원들은 동기화 작업을 대기하고 있어야한다.

이 논문에서는 긴 대기시간을 해결할 수 있는 동시 양방향 동기화 프로토콜을 설계 및 구현하였다. 동시 양방향 동기화 프로토콜은 다중 큐를 사용하여 변경충돌이 발생하지 않는 동기화 작업들 간에 동시 처리를 지원한다. 구현된 프로토콜을 사용하면, 처리시간이 긴 동기화를 처리 중에도 변경충돌이 발생하지 않는 다른 동기화 작업을 대기 없이 수행 할 수 있다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 이 논문에서 설계 및 구현한 동시 동기화 프로토콜의 전체 개념도에 대하여 설명한다. 4장에서는 구현 결과를 제시한다. 그리고 마지막으로 5장에서 결론 및 향후 연구에 대하여 기술한다.

II. 관련연구

ActMAP시스템은 유럽 자동차 회사들과 디지털 지도 제작 업체와의 프로젝트 팀에 의해 2007년도에 제안된 네비게이션 지원 시스템으로 자동차에 탑재된 네비게이션 기기에 무선 통신을 이용하여 지도의 최신 변경사항을 반영할 수 있도록 지원하는 시스템이다.

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

첫 번째는 무선 통신에서 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 데이터 모델에 기반하여 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 데이터가 변경됨에 따라 발생하는 변경 데이터의 품질 유지 문제이다. 이를 위하여 ActMAP은 변경을 위한 트랜잭션 개념을 제공하고 변경 데이터간의 계층·분할 의존성을 고려한다. 그러나 ActMAP 시스템은 서버에서 클라이언트로의 동기화는 가능하지만, 클라이언트에서 서버로 동기화할 수 없다.

분산되고 제약적인 모바일 센서 네트워크의 데이터동기화[3]는 통합 연산중 때때로 센서 노드간의 통신채널이 끊어져 발생하는 문제점을 해결하기 위한 알고리즘을 제공한다. 끊어진 통

신 채널이 복구되어도 노드들의 스냅 샷은 더 이상 일관되지 않는다. 이러한 문제를 해결하기 위해서 모바일 센서 네트워크의 데이터동기화 시스템에서는 2가지 알고리즘을 제공한다.

첫 번째 방법은 타임스탬프(timestamp)를 이용하여 노드 사이에 이동한 데이터레코드의 이동궤적을 저장하는 방법이다. 두 번째 방법은 사용한 해시(hash)정보와 인덱스 트리를 사용하여 두 센서노드 사이의 차이를 찾는 방법이다. 타임스탬프를 이용한 동기화 방법이 해시 기반의 동기화방법 보다 좀 더 효율적이다. 그러나 두 알고리즘 모두 다수 노드에 대해서 순차적인 방법으로 동기화를 진행한다.

모바일 E-Business 어플리케이션을 위한 트랜잭션 중심의 분할 동기화 방법[4]은 모바일 e-business 클라이언트 어플리케이션은 단절모드에서 계획적인 연산으로 모바일 장치의 통신비용과 전력소비를 감소시킬 수 있다.

모바일 트랜잭션과 서버데이터베이스의 동기화를 중간 동기화단계와 최종 동기화단계로 분할한다. 중간 동기화는 최종 동기화가 일반적인 낮은 비용의 유선LAN 같은 통신 수단을 이용하는 동안 높은 비용의 무선 수단을 이용한다. 사용자는 자신의 취향에 따라 동기화의 세부 기능을 정의 할 수 있다. 그러나 다수 클라이언트의 동기화에 대하여 고려하지 않았다.

III. 동시 동기화 프로토콜

이 논문에서 제안하는 동시 동기화 프로토콜을 수행하는 시스템 개념도는 그림 1과 같다.

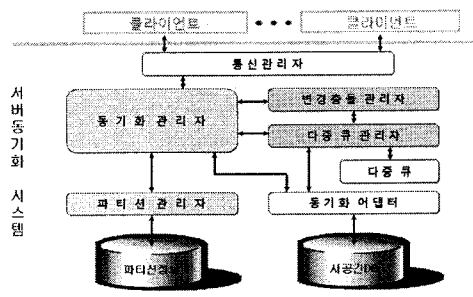


그림 1. 동시 동기화 시스템 개념도

동기화 관리자는 서버에서 처리되는 동기화 작업의 전체적인 흐름을 관리하며, 통신 관리자는 동기화를 요청하는 클라이언트들의 리스트를 관리하고 각 클라이언트들과 통신을 담당한다. 파티션 관리자는 시공간데이터베이스의 파티션 정보와 각 파티션의 최종갱신 시간정보를 가진 파티션정보를 관리한다. 동기화 요청 시 파티션정보의 최종갱신 시간을 이용하여 변경충돌검사를 한다.

변경충돌 관리자는 동기화를 요청한 클라이언트의 변경데이터와 서버의 시공간데이터베이스가 변경충돌이 발생하는지 검사하고, 처리중인 동기화 작업이 있는 경우 처리중인 변경데이터와의 변경충돌 발생여부를 검사한다. 다중 큐 관리자는 다중 큐에 변경데이터를 삽입하고, 다중 큐에 입력된 변경데이터를 라운드로빈 방식으로 삭제한다.

그림 2는 서버로 클라이언트가 접속하면 처리되는 동기화 처리 흐름도이다. 클라이언트가 접속하여 동기화가 시작되면 먼저 클라이언트의 변경데이터에 대한 변경충돌검사를 한다. 충돌이 없으면 동기화를 진행하고, 충돌이 발생하면 클라이언트로 변경데이터의 충돌을 알리고 대기 상태가 된다. 변경충돌이 없는 경우에는 다중 큐에 변경데이터를 삽입하고, 삽입된 데이터는 큐 삭제를 통해서 연산자에 따라 변경데이터를 처리한다.

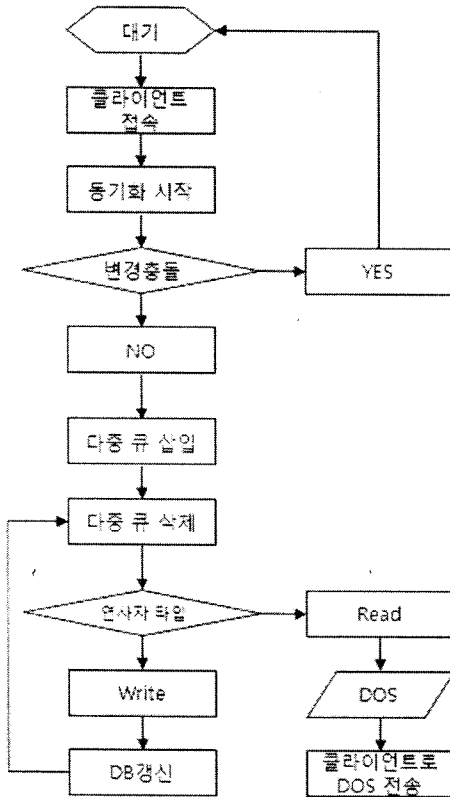


그림 2. 동기화 처리 흐름도

그림 3은 다중 큐의 노드정보이다. 다중 큐의 노드는 클라이언트식별자, 연산자타입, 변경데이터, 복사영역(CR)으로 구성되어있다. 클라이언트식별자를 이용하여 다수의 클라이언트가 동기화할 때 처리된 변경데이터와 복사영역이 어떤 클라이언트의 작업이였는지 구분한다. 연산자타입

은 다중 큐 삭제 시 처리될 연산의 타입이다. 연산자타입이 Write인 경우는 변경데이터를 서버데이터베이스에 갱신하고, Read인 경우 복사영역에 포함되는 서버의 변경데이터를 클라이언트로 갱신한다.

다중 큐 노드

CID	OpType	변경데이터	CR영역
역할	CID		OpType
타입	클라이언트ID		Operation 형태
내용	SOCKET		int
내용	동기화를 요청한 클라이언트의 ID		1:Write 2:Read

변경데이터

OID	객체길이	X	Y	X'	Y'
역할	변경객체 CID	변경객체의 길이		X, Y, X', Y'	
타입	int	int		int	
내용					

CR영역

PID	MBR	LST	
역할	PID	MBR	LST
타입	파티션 PID	파티션의 영역	최종동기화 시간
내용	int	RECT	Char[13]
내용		파티션이 최종으로 동기화 했던 시간	

그림 3. 다중 큐 노드 자료구조

IV. 구현 결과

구현된 시스템의 서버환경은 Dual Core 3.00GHz, 2.00GB RAM 200GB S-ATA Hard Disk 이고, 클라이언트환경은 Intel PXA270 624MHZ, 256MB RAM으로 되어 있다. 그림 4는 클라이언트들이 서버로부터 최초로 다운받아 실험에 사용한 10만개의 도로데이터이다.

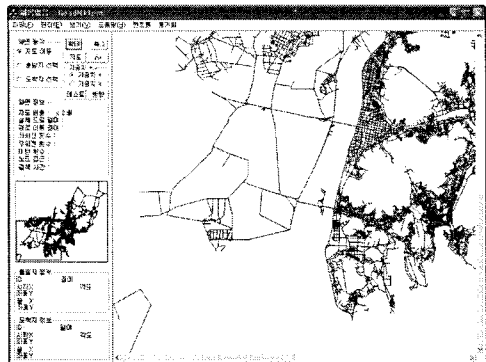


그림 4. 초기 도로데이터

그림 5와 6은 각각 클라이언트들이 변경한 데이터를 보여준다. 변경된 도로는 각 그림의 사각

형 안에서 볼 수 있다. 이렇게 변경된 데이터는 메뉴에 있는 동기화버튼을 클릭하면 서버로 변경된 데이터가 전송되고, 서버의 변경데이터를 다운 받는다.

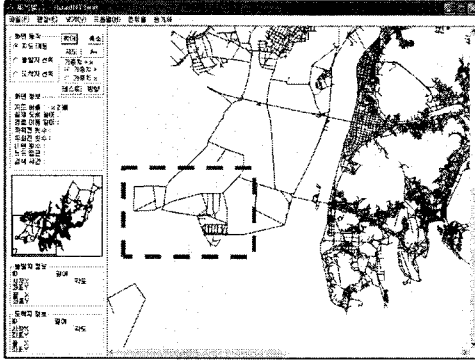


그림 5. 클라이언트1의 변경데이터

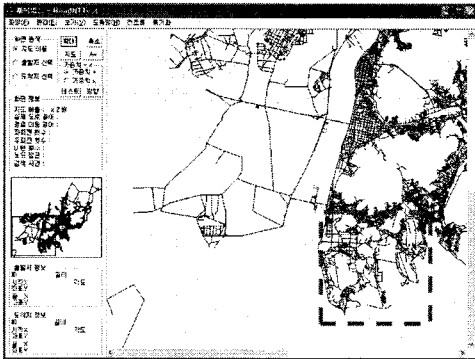


그림 6. 클라이언트2의 변경데이터

그림 7은 클라이언트1이 클라이언트2 보다 먼저 서버와 동기화하고 클라이언트2가 서버와 동기화한 결과이다.

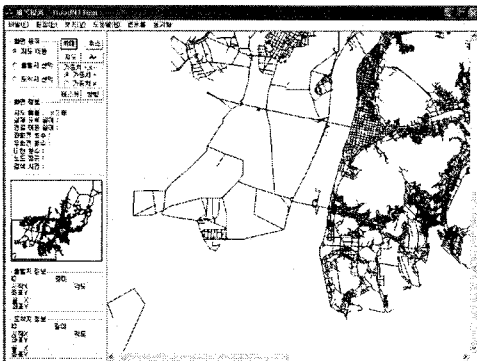


그림 7. 동기화 후 클라이언트2의 도로데이터

V. 결 론

구현된 시스템은 처리중인 동기화 작업이 있어도 동기화를 요청한 클라이언트와 처리중인 동기화 작업이 변경충돌이 없으면 동시에 동기화를 한다. 변경충돌검사로 인한 처리연산이 증가 하였지만 대기시간 없이 동기화를 할 수 있다. 그러나 처리시간이 긴 동기화 작업들이 동시에 동기화를 수행할 경우에 시스템의 효율성을 장담할 수 없다. 실험을 통하여 시스템의 효율성 검사가 필요하고, 다양한 조건에서 효율적인 동시 동기화프로토콜에 관한 연구가 필요하다.

참고문헌

- [1] "ActMAP White Paper and Interfaces to the FeedMAP framework," white paper, 2007
- [2] 김홍기, 임창우, 이상신, 조대수, 김동현, "모바일 GIS DB를 위한 양방향 동기화 프로토콜 설계," 한국해양정보통신학회 2008 추계종합학술대회, Vol.12, No.1, pp.183-186, 2008
- [3] Shuai Hao, Hock Beng Lim, "Data Synchronization in Distributed and onstrained Mobile Sensor Networks," UIC 2007, LNCS 4611, pp.673-683, 2007
- [4] Mi-Seon Choi, Young-Kuk Kim, Juno Chang, "Transaction-Centric Split Synchronization Mechanism for Mobile E-Business Applications," IEEE, 2005