
모바일 환경에서 양방향 동기화 시스템을 위한 회복 기법의 설계

김동현* · 김홍기** · 조대수***

Design of the Recovery Scheme for the Bidirectional Synchronization System
on Mobile Environments

Dong Hyun Kim* · Hong-Ki Kim** · Dae-Soo Cho***

본 연구는 국토해양부 첨단도시개발사업 - 지능형국토정보기술혁신 사업과제의
연구비지원(07국토정보C05)에 의해 수행되었음

요약

모바일 클라이언트에서 갱신된 시공간 데이터는 서버와 무선 네트워크를 이용하여 동기화되고 사용자에게 신속하게 서비스 될 수 있다. 그러나 모바일 단말기나 무선 네트워크는 오류에 매우 취약하다. 만약 동기화 도중에 오류가 발생하면 동기화 대상 데이터에 대하여 데이터 불일치가 발생할 수 있다. 이 논문에서는 동기화 도중의 오류로 인하여 발생하는 문제를 해결하기 위한 회복 기법을 제안한다. 제안한 회복 기법은 모바일 클라이언트와 서버의 지역저장장치에 로그 레코드를 기록하고 오류가 발생하면 저장된 로그 레코드를 이용하여 회복한다. 그리고 제안한 회복 기법을 위한 회복 시스템을 설계하고 구현하였다.

ABSTRACT

It is possible to synchronize the spatio-temporal data with a server using a wireless network. These data has the benefit to be up-to-date and able to service to the user efficiently. However, the mobile devices or the wireless network are very vulnerable to various hard or soft failures. If an failure occurs during a synchronization process, the spatio-temporal data which is synchronizing could be inconsistent state. In this paper, we propose the recovery scheme to solve the problem caused by the errors which occur during the synchronization on mobile environments. The proposed scheme writes the log records for the synchronization process into the both local storages of the server and the mobile client. When the errors arise, the data which are synchronizing are recovered using the written log records. Also, we design and implement the proposed recovery scheme on the synchronization system.

키워드

회복, 로그, 동기화, 모바일 클라이언트, 무선 네트워크

Key word

recovery, log, synchronization, mobile clients, wireless networks

* 정회원 : 동서대학교 컴퓨터정보공학부 부교수

** 정회원 : 동서대학교 유비쿼터스IT 대학원

*** 정회원 : 동서대학교 컴퓨터정보공학부 부교수

(교신저자, dscho@dongseo.ac.kr)

접수일자 : 2011. 03. 28

심사완료일자 : 2011. 04. 21

I. 서 론

과거의 공간 데이터 및 시공간 데이터의 변경 작업은 항공 사진과 같은 아날로그 데이터를 먼저 수집한 다음에 디지털라이저를 이용하여 사람이 수작업으로 디지털화하였다. 그러나 모바일 단말 기기에서의 시공간 데이터를 사용하는 응용 시스템이 가능해짐에 따라 데이터 변경 작업도 모바일 클라이언트를 활용하여 수행될 수 있다. 변경된 시공간 데이터는 동기화 시스템[1]을 이용하여 실시간으로 서버와 동기화되고 서비스 사용자에게 신속히 서비스되는 장점이 있다.

그러나 모바일 단말기기는 외부 충격에 취약하며 특히 무선 네트워크는 유선 네트워크에 비하여 불안정한 연결을 유지하는 단점이 있다. 따라서 변경된 시공간 데이터를 동기화시키는 도중에 모바일 단말기에 물리적인 장애가 발생하거나 또는 무선 네트워크가 갑자기 단절되면 변경된 시공간 데이터를 손실하게 되는 문제가 발생한다.

이 논문에서는 무선 네트워크를 이용하여 데이터를 동기화하는 과정 중에 오류가 발생할 때 시공간 데이터의 일관성을 보장하기 위한 회복 기법을 제안한다. 제안하는 회복 기법은 동기화 작업을 수행하는 도중에 동기화 로그 기록을 저장하며, 오류가 발생할 때 로그를 이용하여 일관된 상태로 회복할 수 있다. 또한 제안한 회복 프로토콜을 지원하기 위한 시스템을 동기화 시스템에서 설계하고 프로토타입을 구현하였다.

이 논문의 구성은 다음과 같다. 2장에서는 기존의 회복 기법에 대한 관련 연구를 소개한다. 3장은 동기화 작업 도중의 오류로 인한 문제점에 대하여 기술하고 4장에서는 발생 가능한 오류들에 대하여 분류한다. 5장에서는 로그를 이용한 동기화 회복 프로토콜을 제안하고 6장에서는 제안한 회복 프로토콜을 지원하는 회복 시스템 프로토타입의 설계와 구현 예를 제시한다. 그리고 마지막으로 7장에서 결론을 기술한다.

II. 관련연구

클라이언트-서버 환경에서 운용되는 대표적인 회복 기법으로 ARIES/CSA[2]가 있다. ARIES/CSA는 기존의

ARIES를 클라이언트-서버 환경에 운용될 수 있도록 확장한 연구이다. 그러나 이 기법은 유선 네트워크 환경의 클라이언트-서버간에 적용한 기법으로 네트워크 장애를 고려하고 있지 않다. 따라서 유선망에 비하여 네트워크 연결이 불안정한 무선 네트워크 환경의 동기화에 적용하기 어렵다.

모바일 분산 환경의 회복 기법[3]은 단절 상태에서 회복 정보를 클라이언트의 지역 저장 장치에 저장하여 네트워크를 이용한 메시지 전달 없이 검사점 작업을 수행한다. 그러나 단절 상태에 있는 모바일 클라이언트의 회복 데이터에는 접근할 수 없기 때문에 일관된 상태로 회복할 수 없는 경우가 발생하는 문제가 있다. [4]에서는 모바일 트랜잭션의 타이머를 동기화하여 다수로 있는 모바일 트랜잭션의 검사점 수행 시점을 동기화한다. 그러나 이 기법은 모바일 클라이언트 상에서 운용되고 있는 모바일 트랜잭션의 타이머를 동기화하기 위하여 메시지 전송을 요구하기 때문에 단절된 상태에서 운용되고 있는 모바일 클라이언트와는 타이머를 동기화할 수 없는 문제가 있다.

III. 문제정의

[1]에서는 다수의 모바일 클라이언트에서 동기화를 요청하였을 때 동기화 작업을 처리하면서 변경된 시공간 데이터의 일관성을 유지하기 위한 양방향 동기화 프로토콜을 제안하였다. 그러나 동기화 과정 중에 오류가 발생하면 변경된 공간 데이터에 대한 불안정한 상태가 발생할 수 있다.

특히 데이터베이스 관리 시스템에서 제공하는 기본적인 회복 프로토콜을 사용하면 서버에 저장되어 있는 시공간 데이터 집합과 모바일 클라이언트에 저장되어 있는 시공간 데이터 집합 간에 데이터 불일치 문제가 발생할 수 있다.

예를 들어 그림 1과 같이 모바일 클라이언트에서 동기화를 요청한다고 가정하자. 모바일 클라이언트는 동기화 프로토콜에 따라 자신이 변경한 데이터 B_1 을 서버에 전송한다. 서버는 B_1 을 저장한 후에 최신의 시공간 데이터인 B_2 를 모바일 클라이언트로 전송한다. 이 때 서버에 시스템 크래시가 발생하면 서버는 기본

적인 회복 프로토콜에 따라 이전의 안정된 상태로 되돌려 버린다(rollback). 따라서 B_1 은 모바일 클라이언트의 변경 데이터가 반영되기 이전의 상태로 되돌아간다. 그러나 모바일 클라이언트는 서버의 크래쉬 상황을 알 수 없기 때문에 전송된 B_5 를 자신의 저장장치에 반영한 후에 동기화 작업을 종료한다. 따라서 이 때 모바일 클라이언트의 B_1 과 서버의 B_1 간에 데이터 불일치 현상이 발생한다.

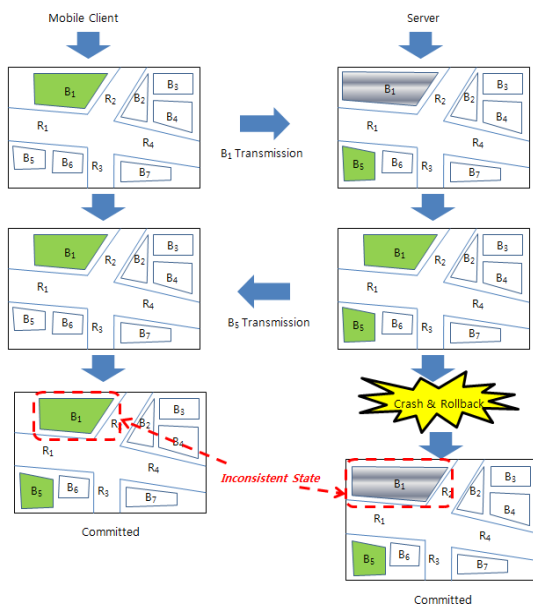


그림 1. 개체 B_1 의 데이터 불일치 상태
Fig 1. Inconsistency State of Entity B_1

IV. 시스템 오류

모바일 클라이언트에서 양방향 동기화를 요청할 때 발생할 수 있는 오류는 크게 5가지로 분류될 수 있다. 표 1은 발생 가능한 오류를 분류한 표이다. 표 1의 오류가 동기화 수행 도중에 그림 2의 오류 중요 구간에 발생하면 데이터 불일치 현상이 발생한다.

표 1. 발생 가능한 오류
Table 1. Possible System Error

심각도	발생 위치	오류 내용	분류
soft failure	server	fail-stop assumption	경우 1
	client	fail-stop assumption	경우 2
hard failure	server	시스템 크래쉬, 강제 종료, H/W 손상 등	경우 3
	client	시스템 크래쉬, 배터리 등	경우 4
	network	무선 네트워크 영역 이탈, 네트워크 단절	경우 5

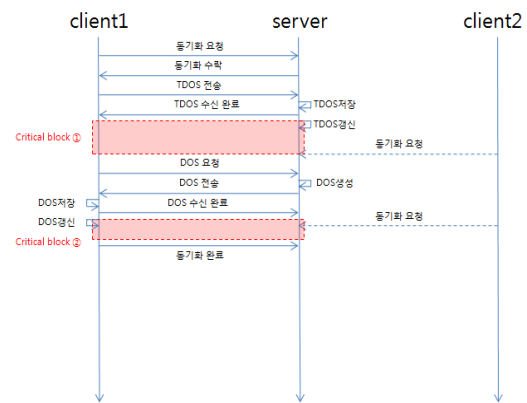


그림 2. 동기화 프로토콜의 오류 중요 구간
Fig 2. Error Critical Section of the Synchronization Protocol

첫 번째 오류 중요 구간은 모바일 클라이언트에 변경된 데이터의 집합인 임시델타객체집합(Temporary Delta Object Set, TDOS)의 수신 여부를 동기화 서버에서 모바일 클라이언트로 알려준 후에 TDOS를 서버 데이터베이스에 갱신하는 작업의 구간이다. 두 번째 오류 중요 구간은 서버에서 변경된 데이터의 집합인 델타객체집합(Delta Object Set, DOS)의 수신 여부를 모바일 클라이언트에서 동기화 서버로 알려준 후에 모바일 클라이언트의 저장 장치에 갱신하는 작업의 구간이다. 이 오류 중요 구간에서 경우 3에서 경우 5까지의 오류가 발생하면 데이터 불일치 현상이 발생한다.

V. 동기화 회복 프로토콜

5.1. 로그

모바일 클라이언트는 동기화 작업을 수행하는 도중에 그림 3과 같이 동기화 로그를 저장한다. 로그는 동기화 서버와 모바일 클라이언트에서 각각 작성되어 지역 저장 장치에 저장된다.

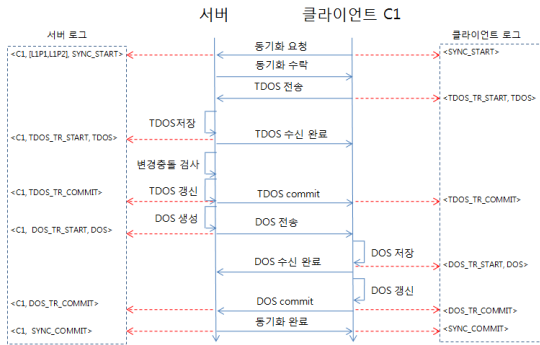


그림 3. 각 단계별 로그
Fig 3. Log on Each Section of Synchronization Protocol

서버의 로그 구조는 표 2와 같다.

표 2. 서버 로그 구조
Table 2. Log Structure of the Server

CID	ClientInfo	State	Data
CID	IP, CR, SDBName	SYNC_START	
CID		TDOS_TR_START	TDOS
CID		TDOS_TR_COMMIT	
CID		DOS_TR_START	DOS
CID		DOS_TR_COMMIT	
CID		SYNC_COMMIT	
CID		SYNC_CANCEL	

클라이언트의 로그 구조는 표 3과 같다.

표 3. 모바일 클라이언트 로그 구조
Table 3. Log Structure of the Mobile Client

ServerInfo	State	Data
IP, SDBName	SYNC_START	
	TDOS_TR_START	TDOS
	TDOS_TR_COMMIT	
	DOS_TR_START	DOS
	DOS_TR_COMMIT	
	SYNC_COMMIT	
	SYNC_CANCEL	

5.2 회복 프로토콜

```

Procedure Network_Recovery of Server
IF Client(CID) does not respond to Server THEN
Check the NETWORK Socket
IF NETWORK Dead THEN // Network Error
Waiting Network recovery
IF NETWORK Alive THEN
Waiting Client connection
IF receive alive message from CID THEN
send alive_ok message to CID;
scan the LOG last_log_record
startsync(last_log_record);
END IF
END IF
END IF
END IF

Procedure Network_Recovery of Mobile Client
IF Server does not respond to the client THEN
Check the NETWORK Socket
IF NETWORK Dead THEN // Network Error
Waiting Network recovery
IF NETWORK Alive THEN
send alive message to SERVER;
IF receive alive_ok message from SERVER THEN
scan the LOG last_log_record
startsync(last_log_record);
END IF
END IF
END IF
END IF
    
```

그림 4. 네트워크 오류 회복 알고리즘
Fig 4. Recovery Algorithm for Network Error

회복 프로토콜은 크게 세 가지로 구성된다. 첫 번째는 그림 4와 같이 네트워크 연결 오류 상황인 경우 5의 오류가 발생할 때 회복하기 위한 프로토콜이다.

```

Procedure Server_Recovery of Server side
BEGIN
  scan the LOG backward UNTIL log record = <SYNC_COMMIT>
  IF last_log_record ≠ <CID, SYNC_COMMIT> || <CID, SYNC_CANCEL> THEN
    send alive message to CID;
    IF receive alive_ok message from CID THEN
      startsync(last_log_record);
    END IF
  ELSE
    Waiting Client connection
  END IF
END

Procedure Server_Recovery of Client side
IF Server does not THEN
  Check the NETWORK
  IF NETWORK Alive THEN // Server Error
    Waiting SERVER System recovery
    IF receive alive_ok message from SERVER THEN
      scan the LOG last_log_record
      startsync(last_log_record);
    END IF
  END IF
END IF

```

그림 5. 서버 오류 회복 알고리즘
Fig 5. Recovery Algorithm for Server Failure

두 번째는 그림 5와 같이 서버에서 경우 3과 경우 4의 오류가 발생하였을 때 회복하기 위한 프로토콜이다. 먼저 서버에서 경우 3과 경우 4가 발생하였을 때 먼저 로그 기록을 역방향으로 검색한다. 그리고 검색된 최종 로그 기록에 따라 알맞은 회복 작업을 수행한다.

마지막으로 모바일 클라이언트에서 오류 3 또는 오류 4가 발생하였을 때 그림 6과 같이 먼저 모바일 클라이언트의 로그 기록을 역방향으로 검색하고 최종 로그 레코드에 따라 회복 알고리즘을 수행한다.

```

Procedure Client_Recovery of Client side
BEGIN
  scan the LOG backward UNTIL log record = <SYNC_COMMIT>
  IF last_log_record ≠ <SYNC_COMMIT> || <SYNC_CANCEL> THEN
    send alive message to SERVER;
    IF receive alive_ok message from SERVER THEN
      startsync(last_log_record);
    END IF
  END IF
END

Procedure Client_Recovery of Server side
IF Client does not respond to the server THEN
  Check the NETWORK
  IF NETWORK Alive THEN // Client Error
    Waiting Client connection
    IF receive alive message from CID THEN
      send alive_ok message to CID;
      scan the LOG last_log_record
      startsync(last_log_record);
    END IF
  END IF
END IF

```

그림 6. 클라이언트 오류 회복 알고리즘
Fig 6. Recovery Algorithm for Client Failure

VI. 구현

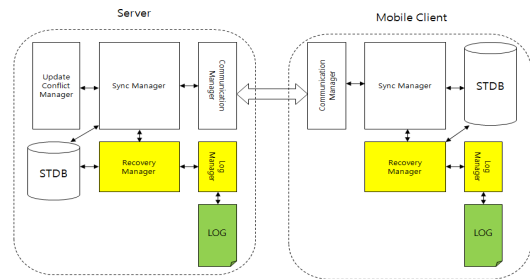
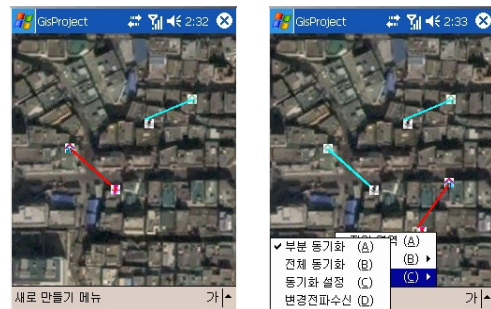


그림 7. 회복 시스템 구조도
Fig 7. Recovery System Architecture

그림 7은 제안된 회복 프로토콜을 구현한 시스템 구조도를 보여준다. 서버는 윈도우즈 XP 환경에서 운용되는 카이로스 시공간 데이터베이스를 이용하였고 모바일 클라이언트는 윈도우즈 CE 환경에서 구현하였다.



(a) (b)

그림 8. 갱신 및 동기화
Fig 8. Updating & Sync

실험은 케이블 네트워크 작업을 대상으로 수행하였다. 먼저 그림 8의 (a)와 같이 갱신 작업을 수행한 후에 (b)와 같이 동기화를 수행한다.

만약 동기화 수행 도중에 모바일 클라이언트의 시스템에서 오류가 발생하면 그림 9의 (a)와 같이 제안한 회복 알고리즘이 수행되고 (b)와 같이 안정된 상태로 회복된다.

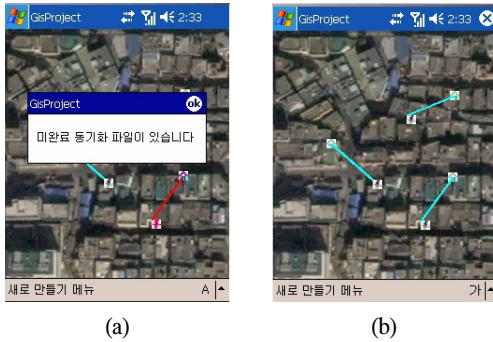


그림 9. 오류 회복
Fig 9. Error Recovery

VII. 결론

시공간 데이터 변경이 모바일 클라이언트를 활용하여 수행할 수 있게 됨에 따라 현장의 정보가 최단시간에 반영된 시공간 데이터를 서비스하는 것이 가능해졌다. 그러나 모바일 단말기나 무선 네트워크는 오류에 취약한 단점을 가지고 있으며 특히 동기화하는 도중에 오류가 발생하면 데이터 불일치 현상이 발생하는 문제가 있다. 이 논문에서는 모바일 클라이언트 시스템에서 무선 네트워크를 이용하여 동기화할 때 오류가 발생하면 시공간 데이터의 일관성을 보장하기 위한 회복 기법을 제안하였다. 그리고 제안한 회복 프로토콜을 지원하는 회복 시스템을 설계하고 프로토타입을 구현하였다. 제안한 회복 프로토콜의 장점은 동기화 작업 도중에 어떠한 오류가 발생하더라도 데이터 일관성이 보장되는 것이다.

참고문헌

- [1] 김흥기, 김동현, 조대수, “모바일 u-GIS를 위한 동기화 시스템 설계 및 구현,” 2009년도 춘계종합학술대회 논문집, Vol. 13, No. 1, 2009.
- [2] C. Mohan, Inderpal Narang, “ARIES/CSA: A Method for Database Recovery in Client-Server Architectures,” ACM SIGMOD, 1994.
- [3] R. E. Strom, S. Yemini, “Optimistic Recovery in Distributed Systems,” ACM Transaction on Computer Systems, vol 3, no 3, 1985.
- [4] N. Neves, W. K. Fuchs, “Adaptive Recovery for Mobile Environments,” Communications of the ACM, vol 40, no 1, 1997
- [5] RICHARD KOO, SAM TOUEG, “Checkpointing and Rollback-Recovery for Distributed Systems,” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. SE-13, No. 1, 1987.
- [6] Shuai Hao, Hock Beng Lim, “Data Synchronization in Distributed and constrained Mobile Sensor Networks,” UIC 2007, LNCS 4611, pp.673-683, 2007.

저자소개

김동현(Dong Hyuni Kim)

한국해양정보통신학회논문지
제 15권 제2호 참조

김흥기(Hong-Ki Kim)

한국해양정보통신학회논문지
제 15권 제2호 참조

조대수(Dae-Soo Cho)

한국해양정보통신학회논문지
제 14권 제4호 참조