

내장형 DBMS를 위한 동기화 서버 시스템

최우영^{○*}, 이경아*, 염태진**, 진성일*

충남대학교 컴퓨터학과

{cutenc[○], kalee, sijin}@cs.cnu.ac.kr*, haru@sorec.cnu.ac.kr**

The Design of Synchronization Server for Embedded DBMS

Woo-Yunng Choi^{○*}, Kyung-A Lee*, Tae-Jin Yum**, Seong-Il Jin*

Dept. of Computer Science, Chungnam National University*

Software Research Center, Chungnam National University **

요 약

이동 컴퓨팅이 활성화 되면서 데이터의 관리 및 저장을 위하여 내장형 DBMS의 필요성이 증대되고 있다. 클라이언트에서 운용되는 내장형 DBMS는 서버용 DBMS와 동기화 기능이 필요하다. 본 논문에서는 통신 비용 절감과 빠른 동기화를 위하여 SQL로 이루어진 로그 정보로 클라이언트와 서버간의 데이터를 양방향 동기화 하며 Mirror Table을 가지는 Mid-tier 구조로 안정성을 유지하는 새로운 동기화 서버 구조를 제안한다. 제안된 동기화 서버는 저장 공간과 안정성이 부족한 내장형 DBMS의 데이터를 일관성과 안정성이 있도록 보장하고 통신비용이 최소화될 수 있도록 함으로써 효율적으로 데이터를 관리할 수 있도록 지원한다.

1. 서 론

최근 정보통신 기술의 발전에 기인하여 편리성과 신속성을 추구하기 위한 휴대형 정보 기기를 통한 업무처리의 활용이 전산업에 걸쳐 많이 요구되고 있다. 특히, 이동 정보 기기를 이용한 전자 비즈니스(e-business), 이동 전자 상거래(Mobile Commerce), 이동 사무실(Mobile Office) 이라는 새로운 시장이 각광 받고 있다.

이러한 추세에 발 맞추어 노트북, 개인정보 단말기(PDA), 인터넷 접속이 가능한 스마트 폰 등의 기술과 이동 통신 기술이 비약적으로 발전하여 휴대형 이동 기기가 널리 보급되었다. 이들 휴대용 기기들의 발전은 언제 어디서나 원하는 정보를 액세스 할 수 있는 이동 컴퓨팅 시대를 이끌었다[1]. 이에 따라 더욱 많은 정보들을 다루게 되었고 작은 크기로 휴대형 이동 기기에 내장 되는 내장형 DBMS가 각광을 받게 되었다.

하지만 이러한 내장형 DBMS를 이용한 데이터의 관리는 이동 환경에서 독립적으로 휴대형 이동기기에서 작동한다. 이 때문에 이동 컴퓨팅은 사용자에게 이동성과 휴대성을 만족 시켜주지만 데이터를 유지, 관리 하는데 있어서 많은 제약이 따른다. 따라서 내장형 DBMS는 데이터를 유지, 관리 하는데 있어서 최소한의 비용으로 최대한의 효과를 이끌어 내는 동기화 기술이 필수적이다. 기존에 이동 DBMS를 위한 동기화 시스템은 Sybase, IBM, Point Base, Oracle 등의 전문 업체에서도 많이 발표되고 상용화되었다.

* 본 논문은 한국과학재단이 지정한 지역협력연구센터(RRC)인 충남대학교 소프트웨어연구센터에 의해 지원으로 수행된 과제의 결과입니다.

본 논문은 기존 업체들이 발표하였던 이동 DBMS 동기화 시스템들의 단점을 보완하고 내장형 DBMS와 서버 DBMS의 일관성을 보장하는 Mid-tier 구조의 새로운 동기화 시스템을 제안한다.

2. 관련 연구

2.1 내장형 DBMS 동기화

내장형 DBMS 동기화 서버란 이동성을 가지는 내장형 DBMS를 탑재한 클라이언트의 데이터 관리에 대한 단점을 보완하기 위하여 높은 성능을 가진 서버이다. client는 단순한 로그 정보만을 가지고 지역적으로 작동하게 되며 통신을 통하여 서버에 접근하게 되고 서버는 이러한 여러 클라이언트들의 작업에 대한 저장, 관리 및 조율을 하게 된다.

2.2 동기화 서버 요구사항

동기화 서버는 서버와 클라이언트의 동기화를 담당하여 일반적인 서버 - 클라이언트의 구조와는 다른 구조를 가지게 된다. 특히 내장형 DBMS를 위한 동기화 서버는 다음과 같은 조건을 만족해야만 한다[3].

첫째, 동기화 모듈의 사이즈가 작아야 한다. 이동 기기는 앞에서 살펴본 바와 같이 저장 공간 및 전원 등에 있어서 많은 제약을 가지고 있다. 이러한 환경에서 사용자에게 효율적인 정보 서비스를 제공하기 위해서는 DBMS에 탑재되는 동기화 모듈의 사이즈가 작아야만 한다. 또 이에 이용 되는 로그의 사이즈를 효율적으로 관리 하여야 한다[3][4].

둘째, 데이터베이스 서버와 클라이언트 서버 양방향 동기화를 지원하여야 한다. 이동 기기와 같은 정보 기기들은 특성상 연결이 단절된 상태에서의 작업이 많기 때문에 독립적으로 작업을 수행할 수 있어야 한다. 이러한 상황에서 필요한 것이 서버측 자료를

이동 클라이언트에서 복제하여 사용하고, 이후의 연결 상태에서 자료의 일관성을 보장할 수 있도록 양방향 동기화는 필수적이다.

셋째 충돌 탐지, 충돌 해결 능력이 있어야 한다. 동기화 서버는 여러 대의 클라이언트로부터 동기화 메시지를 받아서 처리하기 때문에 서로 동일한 튜플을 각기 다르게 삽입, 삭제, 변경을 할 수 있기에 삽입 충돌, 삭제 충돌, 갱신 충돌 등이 일어나게 된다 [5]. 충돌을 탐지하고 해결을 하는 작업은 서버에서 일어나게 되며, 버전 정보 및 상태 정보를 이용 하여 이를 해결한다. 이러한 정보들을 이용하여 충돌의 발생시 동기화 정책에 따라 변경 연산을 거부하거나 반영한다.

2.3 동기화 서버 시스템 비교 분석

대표적인 상용제품에 대해 특징 및 동기화 기술에 대하여 살펴 보았고 [표 1]에서 각 제품마다 특징을 볼 수 있다[2]

분류	SQL Remote	Sync Server	Cloud Sync	iConnect	제안 시스템
안정된 동기화	○	○	○	○	○
시스템 구조	DBMS 통합	Mid-tire	Mid-tire	DBMS 통합	Mid-tire
Embedded DBMS 연동	X	X	○	X	○
동기화 할 내용 파악	Transaction 로그	변경역사 자동 추적	Business Logic	Before-image이용	SQL 로그
충돌탐지	DB 서버	동기화 서버	DB 서버	DB 서버	동기화 서버
충돌해결	Trigger로 정의	서버우선	개발자 구현	우선 규칙 정의	서버우선

표 1 : 동기화 서버 시스템 비교 분석

3. 동기화 서버 시스템

3.1 동기화 서버 모델

본 논문은 내장형 DBMS 와 서버 DBMS의 동기화를 위하여 동기화 모델을 제안 한다 제안하는 동기화 모델의 구조는 [그림1]과 같으며 동기화 모델은 다음과 같은 조건을 갖는다.

1. Embedded DBMS와 Source Server 중간에 동기화 서버를 두는 Mid-tire구조를 갖는다.
2. 각 모바일 디바이스에 존재하는 클라이언트는 독립적이며 데이터는 일단 처리된 후, 사용자의 요청에 의한 동기화 시점에 동기화 하는 지연된 동기화(delayed synchronization) 방식.
3. 동기화 서버에서 사용되는 Mirror Table은 각 동기화 대상 디바이스별로 생성.
4. 로우 아이디(RowID)를 사용하여 각 레코드를 구별하며, 각 디바이스마다 RowID 풀을 사용하여 ID 충돌을 피한다.
5. 데이터 변경 상태를 표시하기 위하여 컬럼과 동기화 버전 정

보를 표시하기 위해 동기화 버전 정보를 가지는 컬럼을 추가.

6. 동기화 서버의 시스템 상태 정보를 입력하기 위하여 property 파일을 이용.

7. property 파일에는 동기화 가능한 디바이스의 정보, 각 디바이스별로 이용되는 테이블의 이름과 Primary Key에 대한 정보를 가지고 있다.

8. 동기화 서버 시스템은 소켓을 통한 TCP/IP 기반의 통신을 한다.

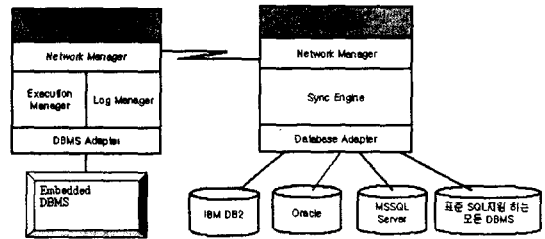


그림 1 : 동기화 모델

3.2 동기화 서버 구조

동기화 서버의 구조를 모듈 별로 나누어 보면 [그림2]과 같다. 동기화 모델을 모듈별로 살펴보면 Network Manager 역할을 하는 Network Module과 Database Adaptor 역할을 하는 Acceptor Module, Sync Engine 역할을 하는 Data Store Module, Sync Module, Log Module로 이루어진다.

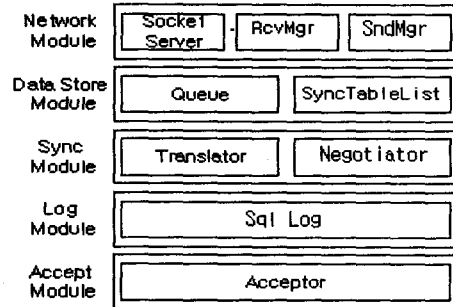


그림2 : 동기화서버모듈구성

이러한 모듈들은 클래스로 구성되어 캡슐화 되며 메시지 기반의 구조로 연결 된다.

3.2 동기화 서버 구조

동기화 서버는 SQL 형태의 로그를 이용하여 서버 측의 Mid-tire 구조에서 작동한다. 동기화 메시지는 로그 정보를 이용하여 [그림3]과 같이 컴퍼넌트들에서 수행이 된다.

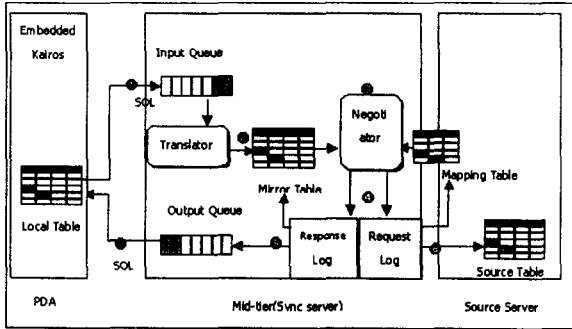


그림 3 : 동기화 서버 구조

3.3 동기화 정책

동기화는 클라이언트에 있는 Local Table과 서버에 있는 Source Table을 일관성 있게 유지 시켜주는 작업이다. 이를 위해서는 동기화에 필요한 동기화 정책이 필요하며 충돌의 발생시 이를 해결 해 줄 수 있는 충돌 해결 정책이 필요하다.

3.3.1 데이터 변경 정책

Negotiator에서 Mirror Table과 Mapping Table의 비교를 통하여 충돌이 없을 시에 반영한다. 충돌 발견 시 충돌 해결 정책에 의하여 반영이 된다.

3.3.2 충돌 해결 정책

충돌은 Negotiator에서 Mirror Table과 Mapping Table의 비교로 이루어지며 선동기화 우선-충돌 레코드 반영 정책을 가진다. 선동기화 우선-충돌레코드 반영 정책은 서버에서 먼저 일어난 동기화가 우선권을 가지며 클라이언트는 서버의 데이터를 우선으로 하는 정책이다. 충돌 유형은 [표2]와 같다

Local	Source	설명
Insert		Insert시 Local과 Source에 같은 primary key를 가진 Record가 있을 때
Delete	Delete	Delete하려는 레코드가 이미 Delete 되어 있을 때
	Update	Delete하려는 레코드가 이미 Update 되어 있을 때
Update	Delete	Update하려는 레코드가 이미 Delete 되어 있을 때
	Update	Update하려는 레코드가 이미 Update 되어 있을 때

표2 : 동기화 충돌 유형

이러한 충돌이 발생하게 되면 Negotiator는 Source Table의 변경 값을 Response Log를 통해 전달한다. 충돌 예제는 다음과 같다.

예제) Insert conflict

- Insert 하려는 레코드의 PK와 같은 Primary Key가 소스 테이블에 존재하는 경우 Insert 쿼리를 거부.
- Mirror Table과 Local Table에서 삽입 하려는 레코드를 삭제
- Mirror Table과 Local Table에 이미 Source Table에 존재하는 레코드 추가

Insert Conflict에 대하여 실제로 수행 되는 쿼리는 다음과 같다.

(L)Local Query, (N)Negotiator Query

L=Local Table, M=Mirror Table, S=Source Table

(L) Insert into S (id ,name) values (5,'전지현')

(N) Delete from M where id =5

Insert into M (id, name, Status, Vsn) values(6, '전지현','Insert', 0)

Delete from L where id =5

Insert into L values(6, '전지현')

4. 결론

최근 이동 컴퓨팅이 활성화 되어 휴대성과 효율성을 만족시켜주는 내장형 DBMS의 이용이 크게 증가하였다. 이에 따라 더욱 많은 데이터들을 효율적으로 관리하고 이용하기 위하여 내장형 DBMS의 사용이 크게 증가 되고 있다. 그러나 이동 컴퓨팅 기기는 휴대성을 전제로 하기에 컴퓨팅 파워 및 자원의 크기 등의 많은 제약 조건이 발생하였다. 이러한 제약과 단점들을 보완하기 위해서는 서버 DBMS와 클라이언트 DBMS를 동기화 하는 기술이 중요사항 이다.

본 논문은 기존의 이동 DBMS 동기화 서버의 단점을 보완하기 위해 SQL 형태의 로그 정보를 이용하여 서버-클라이언트 간의 동기화를 제안 하였다. 이 구조는 양방향 동기화로서 Mid-tire 방식으로 안정성이 뛰어나며 SQL형태의 로그를 이용하여 통신 비용의 절감과 자원의 효율적 관리가 가능하다. 또, 사용자의 요청에 의한 동기화 시점에 동기화 하는 지연된 동기화(delayed synchronization) 방식을 사용하여서 접속 능력면에서 효과적이다.

앞으로의 연구 과제는 내장형 DBMS 동기화 서버가 업무용으로 사용되어 많은 디바이스를 대상으로 동기화 할 때의 안정성 확보와 효율 향상을 위해 광범위하게 성능을 테스트 해 보는 것과 SQL형태의 로그 정보의 암호화를 통한 보안성 확보가 있다.

참고 문헌

[1] 최미선, 김영국, "이동(Mobile) 데이터베이스 개요 및 연구 현황", 데이터 베이스 연구회지 17권 3호, pp3-16, 2001.9.
 [2] 이상윤, 박순영, 이미영, 김명준, "이동 DBMS의 데이터 동기화 기술 분석", 데이터 베이스 연구회지 17권 3호, pp29-31, 2001.9.
 [3] Sixto Ortiz Jr., "Embedded Databases Come out of Hiding", IEEE Computer Magazine, Mar. 2000, pp.16-19
 [4] 최정현, 최미선, 김영국, 진성일, 박중기, 김중배, "무선 인터넷 환경에서의 내장형 DBMS를 위한 자료 동기화 기술" 제1회 정보 가전과 실시간 시스템 응용 워크샵, pp203-207, 2001.11
 [5] Bauer, Jonathan Andover, Bodge, Andrew, "System for synchronizing share data between computers", United States Patent Number 5,884,325 Mar.19