韓國 컴퓨터情報學會 論文誌
第11卷 第2號, 2006. 5.

2006-11-2-1-15

모바일 컴퓨팅 데이터베이스 환경에서의 낙관적 제어기법을 이용한 동시성제어기법

조 성 제*

A Concurrency Control Method using Optimistic Control in Mobile Computing DB Environment

Sung-Je Cho *

요 약

무선통신 기술의 급속한 발전으로 무선 인터넷 서비스가 점차 확대되고 있고 그 중 모바일 실시간 처리가 큰 비중을 차지하고 있다. 모바일 트랜잭션 처리는 낮은 대역폭과 핸드오버, 응답시간 지연 등으로 그것의 활성화를 저해하는 여러 가지 문제점을 지니고 있음에도 불구하고 모바일 컴퓨팅 분야에 다양하게 응용되고 있다. 그래서 모바일 컴퓨팅 환경에서 제한된 대역폭을 효율적으로 사용하고, 병목현상을 개선한 새로운 동시성 제어 기법이 요구된다. 본 논문에서는 모바일 컴퓨팅 환경에서의 동시성 제어 문제를 효과적으로 해결하고 동시에 여러 트랜잭션을 처리하여 병렬성을 증진시키는 낙관적 동시성 기법을 제안하였다. 기존기법과 달리, 제안하는 기법은 같은 세그먼트 내에 다른 데이터를 접근하는 트랜잭션에게 세그먼트를 허용함으로써 불필요한 대기시간을 최소화 할 수 있도록 하여 시스템 처리 율을 향상시켰다. 그리고 제안된 동시성 제어 기법의 알고리즘을 제안 하였다.

Abstract

The rapid growth of mobile communication technology has provided the expansion of mobile internet services, particularly mobile realtime transaction takes much weight among mobile fields. Current mobile transaction service has serious problems which check its development, such as low bandwidths, hand over, expensive charge system, and low response time. but, There is an increasing demand for various mobile applications to process transactions in a mobile computing fields. In mobile computing environments, A mobile host computing system demands for new concurrency control method to use the bandwidth efficiently, to improve the bottleneck and the response time of transactions. This study suggests about an efficient concurrency control in a mobile computing environment. Concurrency control method in existing method uses two phases locking method. In this method, Many clients can't use the same segment simultaneously, and so useless waiting time increases. The characteristic of this proposed method unlike existing one, Enable the transaction approaching different data in the same segment to minimize the useless waiting time by permitting segments, and therefore improves the coexistence of system. Also, It shows the algorithm of the proposed concurrence control method.

▸ Keyword : 모바일 컴퓨팅(Mobile Computing), 낙관적 기법(Optimistic Control), 세그먼트 완료 리스트(Segment Commit List)

• 제1저자 : 조성제
• 접수일 : 2006.04.07, 심사완료일 : 2006.05.19
* 성결대학교 e-비즈니스IT학부 교수

Ⅰ. 서 론

최근 무선통신 및 정보기술의 발전으로 모바일 컴퓨팅(Mobile Computing)환경에서의 서비스가 여러 분야에 폭넓게 제공되고 있다. 예를 들면, 교통정보, 기상정보, 증권거래 등의 분야에 활용하고 있다. 그 이유는 사용자들이 PDA나 휴대폰 등과 같은 모바일 장비를 사용하여 지리적 제약을 받지 않고 정보 서비스를 제공받을 수 있기 때문이다. 이러한 컴퓨터 환경을 모바일 시스템 이라고 한다.

모바일 시스템은 모바일 서버와 모바일 호스트로 구성되어 있다. 모바일 호스트와 통신할 수 있는 무선 인터페이스를 갖춘 서버를 방송서버 또는 모바일 기지국이라 한다[1].

방송서버의 기능은 모바일 호스트들이 트랜잭션을 처리하는 데 필요한 데이터를 제공하거나 다른 모바일 호스트들과 통신할 수 있는 통로를 제공한다. 노트북, PDA 등의 모바일 기기를 모바일 호스트라 정의 한다[16]. 모바일 호스트가 다른 모바일 호스트와 통신하기 위해서는 방송서버를 경유해야 한다. 모바일 호스트에서 수행되는 트랜잭션을 모바일 트랜잭션[2]이라 하는데, 모바일 호스트에서는 주로 판독 전용 트랜잭션이 수행되고, 반면에 방송서버에서는 기록연산으로 구성되는 갱신 트랜잭션이 수행된다.

최근 사용자들은 자신의 위치나 장소에 상관없이 모바일 시스템에 접근하여 은행업무, 주식정보, 공과금결재, 전자상거래 등을 수행하고자 한다. 이러한 형태의 요구는 클라이언트/서버 시스템의 환경과 많은 차이점이 있으므로, 새로운 형태의 모바일 시스템을 필요로 하게 되었다. 모바일 시스템에서 데이터를 전송하는 방법은 요구에 의한 방법과 방송에 의한 방법으로 분류할 수 있다[4]. 모바일 시스템에서는 방송서버에서 모바일 호스트로의 하향(down-stream) 대역폭은 매우 크고 반대 방향인 상향(up-stream) 대역폭은 매우 작은 비대칭적인 특징을 가지고 있다. 요구에 의한 방법은 병목현상으로 대기시간이 길어지는 문제점이 있다. 반면에 방송에 의한 방법은 방송서버가 모바일 호스트의 요청과 무관하게 주기적으로 데이터를 배포하는 방법이다. 모바일 호스트는 방송 채널을 통하여 필요한 데이터를 선택적으로 접근한다. 이 방법은 모바일 호스트의 수와 무관하게 제한된 대역폭을 통해 원하는 데이터를 접근할 수 있기 때문에 모바일 데이터베이스 환경에서는 요구에 의한 방법보다 효율적이다.

기존의 시스템에서와 같이 모바일 시스템에서도 여러 트랜잭션이 동시에 방송 채널을 통해 데이터를 접근하기 때문에 서로 다른 모바일 호스트의 트랜잭션이 같은 데이터를 동시에 접근하는 경우가 발생할 수 있다. 예를 들어, 동시에 두 모바일 호스트가 동일한 데이터에 접근하여 한 클라이언트는 읽기 연산을 다른 클라이언트에서는 쓰기 연산을 수행할 수 있다. 이러한 경우 데이터의 일관성은 보장하기 위하여 동시성제어 문제를 해결해야 한다. 동시성제어 기법에서 로킹(locking)기법은 필요한 데이터에 대한 록(lock) 획득하기 위해 방송서버에게 요청한다. 요청으로 인한 방송서버와 통신으로 대기시간과 록 해제(unlock)할 때 까지 다른 모바일 호스트에서 트랜잭션을 수행할 수 없는 문제점이 있다[10]. 이러한 이유로 낙관적 기법을 이용한 동시성 제어 기법을 제한하였다.

모바일 컴퓨팅 환경에서 방송서버와 모바일 호스트는 방송전파를 통하여 상호작용한다. 모바일 호스트가 캐싱(caching)을 사용할 때 캐시(cached)된 데이터의 일관성을 유지하기 위해 대부분의 기법에서는 주기적인 방송전파를 사용하였다. 캐싱을 사용할 때 캐시의 일관성이 위배될 수 있는 또 다른 경우는 모바일 호스트가 방송 서버와 단절되는 경우이다. 모바일 호스트가 단절되면 방송된 메시지를 수신하지 못하여 갱신된 데이터 값을 캐시에 반영하지 못하게 된다. 그리고 모바일 호스트가 셀의 경계를 넘어 이동하는 경우 방송 메시지를 수신하지 못하여 캐시 내 데이터의 일관성이 위배된다[5]. 따라서 많은 연구들이 데이터를 캐싱하는 방법이나 캐시 내 데이터의 일관성을 유지하기 위해 제안되어지고 있는데[6,7,8], 모바일 호스트에서 트랜잭션들을 수행 후 방송서버 데이터베이스에 반영할 때 완료 트랜잭션의 직렬 가능한 수행을 고려한 연구들은 많지 않다.

제안하는 기법은 기존의 문제점을 파악하여 다음과 같이 설계하였다.

첫째, 기존의 대부분 기법들은 읽기전용 트랜잭션을 자체적으로 검증할 수 있도록 하는 기법을 제시했다. 그와 관련해 하나의 아이템에 대한 다양한 버전을 제공하거나 무효화 보고, 직렬화 그래프와 같은 제어정보를 이용하였다. 하지만 모바일 시스템 환경에서의 증권정보, 기상정보, 교통정보 등과 같은 읽기전용 트랜잭션뿐만 아니라 금융서비스 등과 같은 갱신 트랜잭션도 처리해야한다. 따라서 읽기전용 및 갱신 트랜잭션들을 효과적으로 처리할 수 있는 동시성제어 기법이 필요하다. 그러나 이에 대한 연구가 미흡한 상태이다.

그래서 제안하는 기법에서는 읽기전용 및 갱신 트랜잭션을 처리할 수 있는 기법을 제시 하였다.

둘째, 기존 기법은 데이터를 동시에 사용할 경우 철회와 재 수행으로 인한 성능저하가 발생된다. 제안하는 기법은 동시에 같은 세그먼트를 사용할 수 있는 모델을 설계하여 시스템 병렬성과 철회 및 재 수행으로 인한 문제점을 감소시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 모바일 시스템에서 동시성제어 기법에 대하여 기술하고, 3장에서는 본 논문에서 제안하는 모델을 설명하고, 4장에서는 제안된 기법의 내용을 기술하고 5,6장에서는 제안하는 알고리즘과 성능분석하며, 마지막 7장에서는 논문의 결론을 맺었다.

II. 관련 연구

2.1 모바일 캐싱 기법

Daniel기법[12]에서는 캐시를 사용하는 환경에서 주기적인 방송으로 캐시 내 일관성을 유지하는 세 가지 방법을 제안하였다. 첫 번째는 모바일 호스트의 상태나 캐시 내의 내용을 파악하고 있는 방송 서버와 모바일 호스트의 어떤 정보도 소유하고 있지 않은 방송 서버로 분류하였다. 두 번째는 모바일 호스트에 방송 무효화 전파 방법을 일정시간 동안 갱신정보를 모아서 정기적으로 전파하는 동기식과 갱신이 발생하는 즉시 전파하는 비동기식으로 분류하였다. 이 방법은 직렬성에 위배되는 경우가 발생한다.

방송 서버에서만 갱신작업을 하므로 모바일 호스트에서는 쓰기 연산하는 경우에는 적용이 어렵고, 방송전파에 따른 오버헤드가 존재한다.

2.2 모바일 동시성 제어 로킹 기법

Jin기법[11]은 모바일트랜잭션의 동시성제어를 위해 로킹 방법을 사용할 때 로그 해제를 위한 부수적인 메시지 교환을 줄일 수 있는 방법을 제안하였다. 읽기 전용 로그를 해제할 때 반드시 로그를 획득한 모바일 호스트가 아니라도 어떤 모바일 호스트에서도 임의로 해제할 수 있도록 하였다. 그래서 사용자의 이동성을 반영한 로그 해제로 통신의 병목현상을 줄일 수 있었다. 읽기전용 로그는 요청 즉시 부여하고, 갱신 로그는 2단계완료 프로토콜의 첫 번째 단계에서 얻을 수 있도록 하였다. 이 방법은 데이터베이스의 중복된 환경으로 가정하고, 중복 관리를 위해서는 과반수 프로토콜을 적용하고 있다. 또한 트랜잭션의 직렬 성을 위해서 다중버전을 사용하고 있다. 쓰기연산은 로그를 획득하지 못하면 트랜잭션 수행을 할 수 없다. 이 로그를 획득하기 위해 다른 모바일 호스트에 있는 로그가 해제하여야 사용할 수 있고, 임의 모바일 호스트에서 로그를 해제할 수 없어 부수적인 메시지 교환이 필요하다. 이 연구는 비 캐시 기법이므로 모바일 호스트가 데이터 항목을 접근하려고 할 때에는 방송서버에 요청하여야 한다. 그리고 방송 서버가 트랜잭션 철회 때 여러 모바일 호스트에 메시지 교환이 요구된다. 이 기법의 문제점은 모바일 호스트와 방송 서버간의 트랜잭션 수행 메시지 교환에 따른 병목현상이 발생한다는 점이다.

2.3 Callback Locking기법

Callback Locking기법[9]에서는 모바일 호스트가 데이터(lock mode)를 함께 캐시하고 있어, 트랜잭션이 종료 되더라도 로그를 소유하고 있어 그 데이터에 대한 유효성을 보장하는 기법이다[9]. 이 기법은 모바일 호스트가 데이터에 접근할 때 매번 서버에게 데이터 유효성을 검사할 필요가 없다. 그러나 읽기 전용로그(shared lock)를 가진 데이터에 트랜잭션을 변경하고자 할 때 전용로그(exclusive lock)를 요청해야 한다. 이 경우 서버는 해당 데이터가 캐시 되어 있는 다른 모든 모바일 호스트에 메시지를 보내어 캐시 된 로그를 회수(callback)해야 하는 문제점이 있다. [9]에서는 읽기 로그 회수기법(callback read locking)을 사용하였고, [9]에서는 읽기 및 변경용 로그를 캐시 하는 쓰기 로그 회수기법(callback write locking)을 사용하였다.

2.4 부분검증과 타임스탬프

Lee기법[13]에서는 트랜잭션의 타임스탬프를 구간으로 기록하고, 이를 동적으로 조정함으로써 불필요한 트랜잭션의 중단과 재실행을 줄이는 기법을 제안했다. 이 기법은 트랜잭션간의 충돌을 초기에 발견하고, 중단할 수 있도록 해서 모바일 호스트의 불필요한 자원낭비를 해결하였다. 그러나 모바일 컴퓨팅 환경에서는 대다수의 트랜잭션이 읽기전용 트랜잭션인데, 이 기법은 읽기전용 트랜잭션도 최종 검증을 위해 서버로 전송해야하므로 상향 통신 대역폭과 배터

리를 소모하게 되는 문제점이 있다. OCC_UTS[15]방법은 모바일 트랜잭션이 접근한 데이터의 타임스탬프가 모두 같거나 가장 최근에 수신한 무효화 메시지의 타임스탬프보다 더 작은 경우에 모바일 트랜잭션을 바로 완료함으로써 응답 시간을 향상 시켰다. 그러나 모바일 트랜잭션의 수행이 즉시 완료 조건을 만족하지 못하는 경우 접근한 모든 데이터가 가장 마지막으로 갱신된 데이터가 아니면 직렬가능하지 못한 것으로 간주하여 철회 결정을 내린다. 이 방법은 올바르게 수행된 트랜잭션도 철회시킬 수 있다는 문제가 있다. 불필요한 철회 결정은 재 수행으로 인한 많은 비용이 발생하며 제한된 대역폭의 사용이 증가한다는 문제점이 있다.

2.5 정방 향과 역방향 검증기법

EV기법[14]에서는 직렬화(Serialization) 검증을 일부 모바일 호스트에서 수행하도록 제안하였다. 실행이 완료된 트랜잭션에 의해 읽히거나 갱신된 데이터 항목들을 모바일 호스트에게 검증정보로서 제공한다. 이 검증정보를 이용하여 충돌이 발생한 트랜잭션을 점검하여 철회시킴으로써 방송서버와 모바일 호스트자원을 효과적으로 이용할 수 있다. 그러나 직렬화 순서의 고려 없이 이미 완료된 트랜잭션과 충돌이 일어나는 모든 실행중인 트랜잭션은 철회하므로 트랜잭션 완료 율이 현격히 낮아진다. Kam-Yiu[3]기법에서는 모바일 트랜잭션이 읽기 연산으로만 구성된 경우에 트랜잭션의 직렬 가능한 수행을 보장하는 방법으로 UFO(Update First Order)방법을 제안하였다. 이 기법은 이전에 방송한 데이터와 모바일 트랜잭션이 접근한 데이터 집합 사이의 교집합을 구하여 공집합인 트랜잭션은 완료하고 공집합이 아닌 것은 재방송을 요구한다. 그러나 잦은 데이터 재방송은 병목현상을 증가시키는 요인이 되며 데이터 방송 스케줄의 재구성으로 인한 부가적인 비용이 발생한다.

III. 제안하는 모바일 시스템 모델

모바일 컴퓨팅 환경에서의 데이터 전송방식은 모바일 호스트로 해당 데이터를 요구하는 방식과 방송 서버로부터 방송전파에 의한 방식이 있다. 모바일 컴퓨팅 환경의 대부분인 기상정보, 교통정보, 방송 등은 방송전파에 의한 읽기전용 방식을 이용하고 있다. 최근에는 쓰기연산인 공과금 결재, 은행결재 등에 대한 요구에 의한 모바일 컴퓨팅 환경이 대두되고 있다. 이런 모바일 컴퓨팅 환경에 적합한 데이터 모델과 동시성 제어 기법이 필요하다. 동시성 제어방법에는 로킹 기법과 낙관적 기법을 적용하고 있다. 모바일 컴퓨팅 환경일 경우는 데이터 충돌을 감지하기 위해 방송 서버와 모바일 사이트 간의 메시지 교환이 지속적인 요구로 인해 오버헤드가 발생한다. 그래서 모바일 컴퓨팅 환경에서는 낙관적 기법을 채택하는 것이 적합하다. 즉 공과금결재, 은행결재 등의 모바일 컴퓨팅 환경에서의 쓰기 연산은 데이터 충돌이 적기 때문에 낙관적 기법을 채택하였다. 본 논문에서는 모바일 데이터베이스 모델을 설계하고, 이에 따른 동시성제어 기법을 제시한다. 제안하는 모바일 컴퓨팅 시스템 모델은 방송서버와 여러 모바일 호스트 그리고 모발일 통신망으로 구성한다. 모바일 호스트는 방송서버로부터 세그먼트를 무조건 전송 받아 트랜잭션을 수행하고, 완료(commit) 후 방송 서버로 로그 정보만 전송한다.

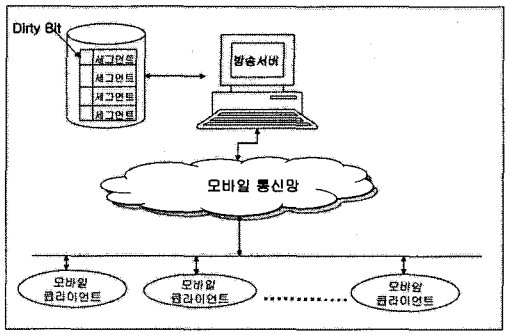

그림 1 시스템 모델
Fig 1 System Model

방송서버는 실시간 처리를 위하여 주 기억 데이터베이스를 유지한다. 갱신된 데이터는 최종적으로 방송서버의 데이터베이스에 반영한다.

여러 모바일 호스트에서 동시에 트랜잭션을 수행한 후 신속한 데이터 충돌 여부를 체크하기 위하여 모바일 호스트별 로그를 부여하고, 충돌을 해결한다. 모바일 호스트는 방송서버의 오버헤드를 감소하기위해 트랜잭션 처리와 철회(abort) 동작을 수행하도록 설계하였다. 시스템 모델은 [그림1]과 같다. [그림1]에서와 같이 방송서버, 주 기억 데이터베이스, 모바일 호스트, 그리고 모바일 통신망으로 구성하고 있다. 방송서버는 모바일 호스트에서 공급받은 데이터를 주 기억 데이터베이스에 반영하기 위해 트랜잭션간의 직렬성 검증을 담당한다. 그리고 방송서버는 일정한 주기로 모바일 호스트에게 방송되며, 완료(commit) 정보를 디렉터리 형태로 구성하여 신속한 전역검증을 담당한다. 모바일 호스

트는 하나의 트랜잭션 처리를 위하여 세그먼트 버퍼와 로그 버퍼 지니고 있다. 그 첫 번째 이유는 트랜잭션 수행 후 데이터의 캐시에 따른 일관성 문제가 발생하지 않는다. 두 번째 이유는 방송 서버와 모바일 호스트의 병목현상 감소를 위해 방송 서버의 간섭 없이 모바일 호스트에서 직접 트랜잭션을 수행하고, 완료 정보(commit log)만 방송 서버로 전송하므로 병목현상을 감소시켰다.

3.1 방송 서버의 구성

모바일 호스트/방송서버 환경에서 방송 서버의 역할은 데이터베이스 관리, 로그버퍼와 세그먼트 완료 리스트(Segment Commit List: SCL), 체크 포인트 기법, 그리고 백업처리기를 관리 담당한다. 기존 기법은 데이터베이스를 디스크에 저장하여 관리함으로서 디스크 입출력으로 빠른 응답을 할 수 없다는 문제점이 제기된다. 본 논문은 이런 문제점을 제거하기 위해 주 기억 장치에 전체 데이터베이스를 관리하고 디스크에는 백업용 데이터베이스를 저장 유지 하도록 구성하였다.

본 논문에서의 로킹 기법은 지연기법을 사용한다. 지연기법은 갱신된 연산을 가지므로 재 수행만을 실시하는 장점을 가지고 있다.

3.1.1 로그버퍼와 SCL

각 모바일 호스트들이 트랜잭션을 Commit한 후, 방송서버에게 재 수행 로그 레코드(Commit Transaction Record: CTR)를 전송한다. 이때 방송서버는 모바일 호스트로부터 전송받은 CTR을 기록하기 위해 세그먼트 버퍼가 필요하다. 이 세그먼트 버퍼는 트랜잭션 충돌여부 체크 수행 동작 때 필요하고, 또한 체크포인트에 의해 시스템 백업 후 제거된다.

그리고, 세그먼트 완료 리스트(Segment Commit List: SCL)는 어떤 트랜잭션이 하나 이상의 세그먼트에 접근 할 수 있는 환경에서 세그먼트별 충돌여부 동작을 위해 각 세그먼트별 로그정보를 유지 관리한다.

3.1.2 체크포인트 기법

SCL을 이용한 전역 검증 후 주 기억 데이터베이스에 즉시 반영하고, 그 후 안정된 장치인 디스크에 백업 동작을 위해 퍼지 체크포인트 기법을 이용한다.

3.1.3 백업 처리기

주기적으로 데이터베이스를 안정된 보조기억 저장 장치에 기록하여야 한다. 이때 백업 처리기는 버퍼크기와 SCL 크기를 고려하여 일정한 주기로 백업처리를 담당한다.

3.2 모바일 호스트의 구성

모바일 호스트는 로그버퍼와 세그먼트 버퍼, 그리고 동시성 처리기로 구성되어 있다. 로그버퍼는 하나의 철회수행 로그레코드(undo log record) 와 재 수행 로그 레코드(redo log record)를 유지 관리하며, 세그먼트 버퍼는 방송서버에서 전송 받는 세그먼트를 저장하기 위한 기억 공간이다. 철회수행 로그 레코드는 트랜잭션을 철회 때 로그 레코드를 이용하여 모바일 호스트가 철회(abort)를 직접 수행한다. 재 수행 로그 레코드는 트랜잭션 수행 완료시, 그 결과의 로그정보를 방송 서버에 전송하기 위한 레코드이다. 동시성 처리기는 모바일 호스트가 방송 서버의 간섭 없이 직접 트랜잭션을 수행할 때 사용되며, 그 결과 병목현상을 감소시킬 수 있다.

3.3 트랜잭션 모델

모바일 호스트는 자신의 버퍼를 지니고, 방송서버로부터 세그먼트를 이 버퍼로 전송받아서 직접적인 접근을 하고, 그에 따른 로그레코드를 만들어 방송서버로 전송하는 책임을 진다. 방송서버는 방송주기에 따라 전송하여 세그먼트를 모바일 호스트에게 할당해 주고, 모바일 호스트로부터 CTR을 전송받아 SCL에 할당된 엔트리에 저장한다.

3.3.1 트랜잭션 정의

본 논문에서 가정하는 모바일 호스트/방송서버 데이터베이스 컴퓨팅 시스템에서는 방송전송단위가 세그먼트이고, 트랜잭션은 모바일 호스트에서만 처리하는 것으로 가정한다.

한 트랜잭션의 처리가 시작될 때 로그에 〈Ti, starts〉라는 로그레코드를 기록하고, 트랜잭션 수행할 때 데이터의 갱신을 기록하기 위해 〈Ti, data item, new value〉의 로그 레코드 형식을 로그에 추가 한다. 그 트랜잭션이 부분종료 시 〈Ti, commit〉라고 로그에 기록 한 뒤, 그 다음에 변경된 값은 모바일 호스트 로그버퍼에 기록된다. 이 과정을 하나의 트랜잭션이라 한다.

3.3.2 트랜잭션의 모델

[그림1]에서 보면 모바일 통신망을 통해서 모바일 호스트로 화살표가 있는데, 이것은 트랜잭션을 처리 하기위해 데이터 아이템이 속한 세그먼트를 방송 서버로부터 전송 받는 것을 나타낸다. 더불어 반대 방향의 화살표는 모바일 호스트에서 처리된 트랜잭션의 결과를 모바일 통신망을 통해 방송서버로 보내는 것을 의미한다.

3.3.3 시스템 가정

본 논문에서는 모바일 호스트/방송 서버 환경이고, 같은 세그먼트를 여러 모바일 호스트가 동시에 사용할 수 있도록 설계 하였다. 그리고 세그먼트 크기는 하나의 모바일 트랜잭션[2]에 필요한 자원으로 한정한다. 동시성 제어 모델에 대한 가정은 다음과 같다. 첫째, 먼저 완료한 트랜잭션이 우선 작업순위를 부여받는다. 둘째, 모바일 호스트들이 같은 세그먼트에 있는 동일한 데이터 아이템을 사용할 경우가 충돌 상태이고, 그 외 경우를 비 충돌 상태라고 정의한다. 셋째, 트랜잭션은 모바일 호스트에서만 처리하는 것으로 정의한다. 넷째, 모바일 호스트는 하나의 트랜잭션만 수행한다.

IV. 개선된 낙관적 동시성제어 모델

본 논문에서 기반으로 하는 모바일 컴퓨팅 동시성제어 기법은 SCL를 이용한 낙관적 동시성 제어기법을 설계하였다. 본 모델에서는 방송서버와 모바일 호스트들 사이는 제한된 대역폭을 가진 무선 통신망으로 연결되어 있다. 방송서버는 모바일 호스트에 방송할 데이터베이스를 유지하고, 각 모바일 호스트에게 읽기연산과 쓰기연산 환경을 제공한다. 방송 서버는 어떤 모바일 호스트에서 수행한 갱신 세그먼트를 데이터베이스에 반영하는 것을 담당한다. 갱신 세그먼트 간의 데이터 충돌은 낙관적인 동시성제어 프로토콜을 사용하여 해결한다. 왜냐하면, 모바일 호스트를 중심으로 한 무선망에서는 데이터에 대한 판독연산을 주로하고 있다. 예를 들면, 교통정보, 기상정보, 증권정보, 관광정보, 관공서 정보 등은 실시간 판독연산이고, 관공서 전자결재, 은행업무, 전자상거래 등은 대부분 하나의 모바일 호스트에서 수행하므로 충돌이 적다. 따라서 트랜잭션간의 충돌이 적게 발생하게 됨으로 로킹을 기반으로 한 기법보다 낙관적인 동시성제어 기법이 효율적이다. 모바일 컴퓨팅 시스템에서의 접속 단절과 핸드오버의 특성으로 인해 무조건 트랜잭션을 수행 후, SCL을 통해 록 모드의 충돌여부를 결정한다. 어떤 트랜잭션이든 모바일 호스트에서만 작업을 수행한다. 왜냐하면 여러 모바일 호스트에서 발생하는 트랜잭션을 방송서버에게 요청할 경우, 대역폭의 병목현상이 발생된다. 더불어 방송 서버의 트랜잭션 수행으로 인하여 오버헤드가 발생하여 연산시간이 길어진다. 방송서버는 단지 세그먼트의 트랜잭션 결과가 다른 트랜잭션과 충돌여부를 체크하고, 데이터베이스에 반영업무를 담당한다. 즉 갱신된 데이터는 최종적으로 방송서버의 데이터베이스에 반영되어야 한다. 각 모바일 트랜잭션은 쓰기 연산이 포함되어 있으므로 일관성을 위해 트랜잭션간의 데이터 충돌을 해결하여야 한다.

제안하는 동시성제어는 낙관적 방법이므로 읽기단계에서 아무런 간섭 없이 수행하다가 완료시점에서 검증단계를 통하여 트랜잭션의 충돌에 위배되지 않으면 완료(commit)를 수행하고, 위배될 경우 충돌 해결 작업을 통해 데이터 일관성을 유지하는 기법이다.

그리고, 방송 전송단위는 세그먼트로 한다. 왜냐하면 다양한 크기의 멀티미디어 데이터를 전송하기 위하여 데이터 아이템보다는 세그먼트단위를 전송단위로 한다.

4.1 모바일 호스트 모델

4.1.1 버퍼 모델

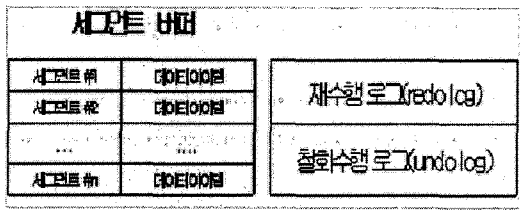

그림2 버퍼 모델
Fig2 Buffer Model

[그림2]와 같이 모바일 호스트는 자신의 버퍼를 지니고, 방송 서버로부터 세그먼트를 방송전송 받아서 세그먼트 버퍼에 저장하고, 이 버퍼를 통해서 직접적인 접근을 하고, 그에 따른 트랜잭션 수행 중에 발생하는 로그 레코드를 만들어 로그버퍼에 저장한다. 트랜잭션 수행 후 완료 트랜잭션 레코드(Commit Transaction Record: CTR)를 방송 서버로 무조건 전송한다. 세그먼트 버퍼는 하나의 트랜잭션을 위한 버퍼로 구성한다. 왜냐하면, 여러 완료된 세그먼트들은 모바일 호스트에서 일관성이 유지되나, 전역검증을 수행하지 않으면 방송 서버와 일관성이 파괴 되므로 연쇄 철회 율이 발생된다. 이런 이유로 한 트랜잭션 완료 즉시 방송 서버로 전송하여 검증한 후 데이터베이스 반영여부를 결정한다. 방송 서버는 주기적으로 수정된 세그먼트가 포함한 정보를 모바일 호스트들에게 방송 전송하고, 모바일 호스트로부터 로그레코드를 전송받아 세그먼트 당 할당된 엔트리에 저장한다.

4.1.2 완료 트랜잭션 레코드 모델

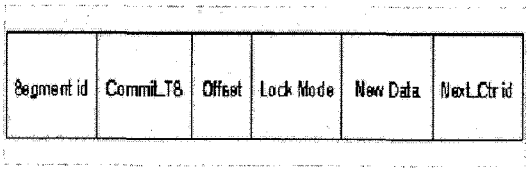

그림3 완료 트랜잭션 레코드
Fig3 Commit Transaction Record

〔그림3〕과 같이 본 논문은 모바일 호스트별로 완료 트랜잭션 레코드 공간을 따로 둔다. 모바일 호스트는 방송 서버로부터 세그먼트를 전송 받아 직접 접근 후, 동시성 처리기의해 트랜잭션을 수행한다. 트랜잭션 수행 중 새로운 Dirty 세그먼트가 전송받으면 작업을 철회 수행하여 취소(Abort)하고 재 수행 하므로 완료 후 검증단계에서 철회 율을 감소할 수 있는 장점이 있다. 그 결과 처리 율을 향상된다. 또, 본 논문에서는 트랜잭션간의 직렬 성을 신속히 검증하기 위해 SCL에서 세그먼트별로 디렉터리를 구성 한다.

그리고, Commit_TS는 모바일 사이트에서 트랜잭션 완료 후 방송 서버에 전송 직전 부여한다. 모바일 호스트와 방송 서버는 일정주기로 동기화 과정을 거치고 TS를 LSN으로 사용함으로서 LSN의 비중복성과 증가 성을 보장한다. 방송서버 사이트에서 직렬성 검증 때 Commit_TS 번호가 사용된다. 그 결과 정확한 일관성을 보장할 수 있다.

4.1.3 모바일 알고리즘

모바일 컴퓨팅 환경에서의 모바일 호스트의 알고리즘을 살펴보면 다음과 같다.

1. 모바일 호스트는 트랜잭션을 처리하기 위해 필요한 데이터 아이템이 속한 세그먼트를 방송서버로부터 전송 받아 세그먼트 버퍼에 저장한다.
2. 트랜잭션 처리기에 의해 세그먼트 버퍼에 있는 세그먼트를 이용하여 트랜잭션 수행한 후, 로그 레코드를 생성하여 로그 버퍼에 저장한다.
3. 트랜잭션이 완료(commit)직전에 CTR을 생성하여 방송 서버로 보냄
4. 방송서버는 SCL, 록 모드 양립성함수를 이용하여 록 모드(lock mode)의 충돌 여부 조사하여 모바일 호스트에게 완료(Commit) 또는 취소(Abort) 메시지를 전송한다.
5. 완료(commit)일 때 방송서버는 데이터베이스에 반영하여 트랜잭션 완료하고, 취소(Abort)일 때 모바일 사이트는 로그 레코드를 모두 제거하고 방송서버로부터 세그먼트를 전송받아 재작업을 시작한다.
6. 방송서버는 완료(commit)일 때 CTR을 이용하여 SCL의 해당위치에 로그를 저장한다.

4.2 방송 서버 모델

4.2.1 세그먼트 완료 리스트 모델

방송서버는 모바일 호스트로부터 트랜잭션 수행에 따른 CTR을 전송받아 해당 세그먼트를 위한 로그버퍼에 순차적으로 저장시킨다. 직렬성 검증 후 완료(Commit)일 경우는 방송 서버는 〔그림4〕와 같이 해당 위치에 레코드를 저장한다.

즉, 한 트랜잭션 수행이 완료될 때 마다 방송 서버는 CTR(Commit - Transaction - Record)을 해당 엔트리의 맨 뒤에 삽입한다.

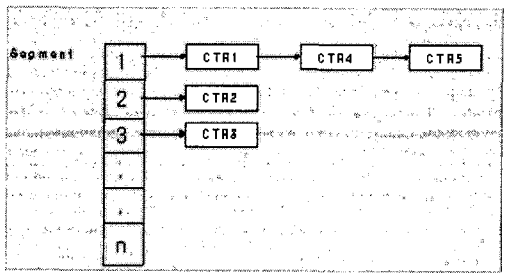

그림 4 세그먼트 완료 리스트
Fig 4 Segment Commit List

여기서 엔트리는 하나의 세그먼트에 대응한다. 이 저장구조를 세그먼트 완료 리스트(Segment-Commit-List: SCL)라 한다.

CTR 구성을 살펴보면, 첫 번째는 세그먼트 일련번호이고, 두 번째는 모바일 호스트에서 부여하는 타임스탬프를 부여하고, 세 번째는 세그먼트의 데이터아이템의 주소이고, 넷째는 록 모드를 구성하고 있다. 이 모드를 통해서 트랜잭션 충돌여부를 결정한다. 다섯째는 NEW DATA를 통해 DB 반영 여부에 사용한다. 마지막은 다음 CTR 주소를 지니고 있다. SCL는 세그먼트별 디렉터리 구성으로 신속한 충돌 검증과 DB 반영을 할 수 있다.

V. 개선된 낙관적 동시성제어 기법

제안하는 동시성제어 모델은 병목현상 감소와 시스템 병렬성 향상을 위해 전송단위를 세그먼트로 설계하였고, 충돌여부를 신속하게 처리하기 위해 dirty된 세그먼트를 디렉터리로 구성하는 기법을 이용하였다. 그리고 동시에 여러 모바일 호스트가 같은 세그먼트를 수행한 할 수 있는 낙관적 동시성제어 기법을 제안하였다.

5.1 SCL을 기반으로 한 낙관적 동시성제어 기법

낙관적 기법을 기반으로 하는 환경은 데이터 충돌이 적을 때 좋은 성능을 제공한다. 즉 모바일 사이트에서는 데이터에 대한 판독 연산을 주로하고 있다. 예를 들면, 교통정보, 기상정보, 증권정보, 관광정보, 관공서 정보 등은 실시간 판독 연산이고, 관공서 전자결재, 은행업무, 전자상거래 등은 대부분 하나의 모바일 호스트에서 갱신연산을 수행하므로 충돌이 적다. 그리고 모바일 컴퓨팅 환경에서는 낮은 대역폭과 핸드오버 현상으로 해당 데이터에 록(Lock)의 획득과 해제에 따른 비용과 시간이 많이 소요된다.

따라서 모바일 사이트는 충돌이 적게 발생하므로 로킹 기법을 기반으로 하는 것 보다는 낙관적인 기법이 효율적이다. 모바일 컴퓨팅 시스템에서의 접속 단절과 핸드오버의 특성으로 인해 무조건 트랜잭션을 수행 후, SCL을 통해 록 모드의 충돌여부를 결정한다.

기존의 로킹 기법은 여러 모바일 호스트가 같은 세그먼트를 동시에 사용하지 못하는 단점이 있다. 즉 같은 세그먼트 내에 다른 데이터 아이템을 접근하고자 할 때는 트랜잭션의 불필요한 대기시간이 발생하는 문제점이 있다. 또한, 데이터 변경할 때 모든 모바일 호스트 사이트에 메시지를 보내어 캐시 된 로그를 회수해야 하는 문제점이 있고, 트랜잭션 철회 율이 많아지는 단점이 있다. 본 논문에서는 모바일 사이트와 방송서버 사이의 전송단위는 기존의 기법과 달리 세그먼트 단위이고, 록 단위는 데이터 아이템단위로 하여 같은 세그먼트를 여러 모바일 호스트가 사용함으로써 시스템 병렬성과 트랜잭션 처리 시간을 단축하는 새로운 동시성 제어 기법을 제안하였다. 그리고 제안된 기법을 이용하여 방송서버와 모바일 호스트 사이 동작을 알고리즘으로 표현하였다.

SCL에서 사용하는 록 양립성 테이블은 표1과 같다.

표1 록 모드 양립성 함수
Table 1 lock mode compatibility Function

server / client	Read Lock	Write Lock
Read Lock	Commit	Abort
Write Lock	Commit	Abort

모든 트랜잭션은 방송서버로부터 전송받은 세그먼트를 가지고, 트랜잭션을 수행하면서 데이터 아이템의 록 모드를 재수행 로그와 철회 수행로그에 기록한다. 이때 록 모드는 판독연산을 수행하는 데이터애 대한 판독 록(READ LOCK : RL)과 쓰기연산을 수행하는 데이터에 대한 쓰기 록(WRITE LOCK : WL)을 사용한다. 제안하는 기법은 모든 연산을 모바일 호스트가 수행하고, 로그 양립성 여부는 방송서버에서 수행한다. 그래서 방송서버 전역 검증을 위해 SCL에 이전 수행한 세그먼트와 그 세그먼트 내에 록을 수행한 데이터 아이템을 기록한다. 이 때 여러 모바일 사이트에서 수행한 CTR을 전송 받으면, 첫 번째는 세그먼트 록 모드의 충돌여부를 체크한다. 비 충돌이면 MMDB에 즉시 반영한다. 두 번째는 충돌이 발생할 경우 SCL의 세그먼트 내에 데이터 아이템에 대한 록 모드와 CTR 내의 데이터 아이템의 록 모드를 표1의 양립성 함수에 의해서 결정한다. 또한 표1의 록 모드 양립성 함수는 모바일 호스트에서도 사용한다. 즉 모바일 호스트에서 수행하는 트랜잭션의 세그먼트가 방송서버에서 전송하는 세그먼트와 불일치 할 때 철회수행 로그와 재 수행 로그를 이용하여 트랜잭션 충돌여부를 결정한다.

5.2 모바일 호스트 수행기능

모바일 호스트는 연산을 위하여 방송주기 때 트랜잭션 수행에 필요한 세그먼트들을 액세스하여 자신의 세그먼트 버퍼에 저장한다. 그리고 자신의 동시성처리기를 이용하여 먼저 세그먼트 내의 데이터 아이템에 대한 록 모드를 CTR에 기록한다. 그 다음 재 수행 로그와 철회수행 로그를 생성하면서 트랜잭션 수행한다. 모바일 호스트는 읽기연산과 쓰기연산의 두 형태의 트랜잭션을 갖는다. 읽기 연산은 방송되는 데이터 아이템에 대해 읽기 연산을 수행한 후 자신이 읽은 데이터 항목의 로그 모드를 CTR에 기록한다. 쓰기 연산은 연산을 수행한 후 록 모드와 갱신 데이터를 CTR에 기록한다. 이 정보는 최종적으로 방송 서버로 전송하여 전역 검증 정보로 사용된다. 트랜잭션의 완료와 취소는 방송

서버에서 전송되는 전역검증 정보(COMMIT/ABORT) 메시지에 의해서 결정한다. 취소일 경우는 다시 방송서버로부터 세그먼트를 전송받아 재 작업한다.

5.2.1 읽기 단계

제안하는 기법은 제3장 가정과 같이 모바일 호스트는 하나의 트랜잭션을 수행하므로 판독단계나 검증단계가 필요가 없다. 즉 모바일 컴퓨팅 환경에서는 병목현상과 시스템 병렬성 향상을 위해 낙관적 기법을 제안하였다. 그래서 방송서버로부터 무조건 세그먼트를 전송받아 읽기 트랜잭션을 수행한다. 왜냐하면 방송서버로부터 최신 정보를 수신하여 트랜잭션 읽기단계를 수행하므로 판독단계나 검증단계가 불필요하다.

5.2.2 쓰기단계

읽기단계와 마찬가지로 모바일 사이트에서 판독단계나 검증단계가 불필요하다. 그러나 읽기단계와 달리 쓰기단계에서는 수행 도중에 CTR에 로그모드와 갱신정보를 기록하여야 한다. 그 이유는 트랜잭션 완료 후 갱신작업의 일관성 검증을 방송 서버에서 수행하므로, 방송 서버에 검증정보를 전송해야한다. 트랜잭션별로 직렬 화 순서를 결정하기 위해 모바일 호스트별로 Commit_TS을 기록한다.

5.2.3 완료 후 수행 단계

트랜잭션은 완료 후 전역 검증을 위하여 CTR 정보를 방송 서버로 전송한다. 방송 서버에서 SCL을 이용하여 충돌해결기법인 검증단계를 거쳐 완료(Commit)/취소(Abort)를 결정한다. 방송 서버는 완료(Commit) 결정일 때는 즉시 데이터베이스에 반영하고, 취소(Abort) 결정일 때는 데이터베이스에 반영하지 않는다. 그리고 완료(Commit)/취소(Abort) 정보를 해당 모바일 사이트에 전송한다. 해당 사이트는 완료(Commit) 메시지를 받을 때는 트랜잭션을 완료하고, 취소(Abort) 메시지를 받을 때 트랜잭션을 중지하고, 다시 다음 방송주기에 세그먼트를 전송하여 재작업을 한다.

5.3 방송서버 수행기능

제안하는 기법에서는 트랜잭션들의 수행완료 후 직렬 화 검증을 위해 모바일 호스트별로 부여한 Commit_TS를 사용한다. 데이터 아이템의 충돌여부는 방송 서버에 존재하는 SCL을 사용함으로써 신속정확하게 직렬 화를 검증할 수 있다. 그 이유는 다음과 같다.

첫째, 완료 세그먼트별로 디렉터리를 구성함으로써 세그먼트 번호를 통해 신속히 충돌여부를 결정할 수 있다. 둘째, 세그먼트 충돌이 발생하면 즉시 트랜잭션을 취소 결정하는 것이 아니다. 왜냐하면 같은 세그먼트 내의 다른 데이터 아이템을 수행하는 경우가 있어, SCL내의 CTR 정보를 이용하여 직렬 화를 검증한다. 직렬화 위배 때는 표1의 로그모드 양립성 함수를 이용하여 직렬화 위배를 결정한다. 그 결과를 해당 모바일 사이트로 전송하여 완료(Commit)/취소(Abort)를 결정한다. 이 때 발생하는 충돌의 형태는 표1의 로그모드 양립성 함수와 같이 네 가지로 분류할 수 있다.

그리고 방송서버의 구성은 다음과 같다. 첫째, SCL을 세그먼트별로 데이터 아이템을 사용한 내역을 기록하여 두어 직렬 성을 검사할 때 사용한다. 그리고 로그 모드의 충돌 발생시 표1의 로그모드 양립성 함수를 이용하여 트랜잭션을 완료(Commit)/취소(Abort)할 것인가를 결정한다. 둘째, SCL은 각 세그먼트별로 디렉터리를 구성하고, 그 세그먼트를 어떤 모바일 호스트가 어떤 작업 순으로 사용하고 있는지를 기록하여 관리하고 있다. 이것은 여러 모바일 호스트들이 같은 데이터 세그먼트를 사용하고 있을 때 충돌여부를 조사하기 위해 사용한다.

셋째, 로그모드 양립성 함수는 CTR에서 데이터 아이템의 충돌이 발생할 경우 이 함수에 의해 완료(Commit)/취소(Abort) 여부를 결정한다.

예를 들면, 모바일 호스트1에서 데이터 아이템 100을 read한 후, 작업을 완료하고, CTR에 lock을 기록하였다. 그 다음, 모바일 호스트3 에서 데이터 아이템100을 이용하여 갱신 연산 후 commit직전에 SCL을 조사 했을 때, 같은 세그먼트를 사용하여 충돌 발생한 경우라도 이 트랜잭션을 정상 완료작업으로 처리한다. 그러나 모바일 호스트3 작업이 먼저 완료된 후 모바일 호스트1이 완료 되었을 때는 비정상 작업으로 간주하여 abort한다.

5.3.1 방송서버의 알고리즘

모바일 호스트와 방송서버 사이에 같은 세그먼트를 동시에 처리 할 수 있도록 제안된 기법의 알고리즘은 [그림5]와 같다.

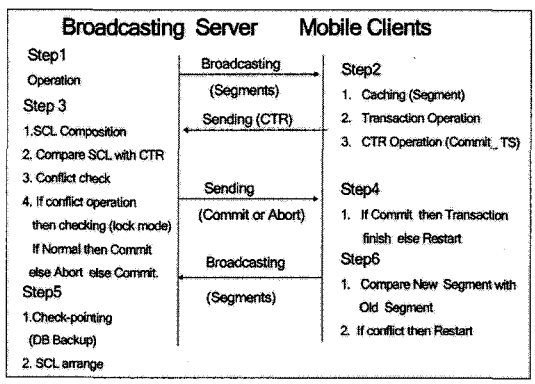

그림5 방송서버 알고리즘
Figure 5 Broadcasting Server Algorithm

VI 실험 및 성능평가

이 장에서는 제안된 기법의 성능평가에 사용된 실험환경을 설명하고 비슷한 트랜잭션 처리환경을 가진 [14]EV기법과의 성능특성을 비교 분석하기 위해 모의 실험한 결과를 기술한다. [14]EV기법은 제안된 기법과 마찬가지로 낙관적 동시성 제어 기법에 기반을 두어 방송 환경의 요구사항을 만족시키도록 설계된 것으로 데이터 충돌을 감지하는 즉시 실행중인 트랜잭션을 취소시키는 것이 다른 점이다. 성능평가에 사용된 인수들은 [표2]와 같다. 실험 모델에 대하여 기본 매개변수를 설정하고, 설정된 매개 변수를 사용하여 다음과 같은 순서로 진행한다. 먼저 EV기법을 본 연구 환경에 맞도록 변화시켜 로킹 기법, 동시성 제어 기법 등에 대해서 모델링을 위한 분석을 한다. 그리고 제안한 시스템 모델링을 기초로 하여 각 시뮬레이션을 실행하였다.

6.1 실험 성능 측정

본 실험은 CSIM 라이브러리를 이용하여 데이터 충돌이 발생할 수 있는 파라미터를 변경하면서 그것이 성능에 미치는 영향을 측정하였다.

표2 시뮬레이션 매개 변수
Table2 Simulation parameter Variable

매개 변수	
데이터베이스 크기	256Mbyte
레코드 크기	32 words
세그먼트 크기	4096 words
워드 당 바이트	4 bytes
세그먼트 요청	EXPON(10)
세그먼트를 기억 장치로 load하는데 걸리는 시간	87.8ms
트랜잭션 처리시간 + 로그 생성 시간	2ms
로그 디스크에 쓰는 시간	87.8ms
로그 분석 + 재 수행	2ms

시뮬레이션 모델에서 세 가지 환경에 대해 각각 시뮬레이션을 수행한다.

(가) 20/80 : 충돌이 일어날 확률이 20%, 비 충돌의 확률이 80%인 경우

(나) 50/50 : 충돌이 일어날 확률이 50%, 비 충돌의 확률이 50%인 경우

(다) 80/20 : 충돌이 일어날 확률이 80%, 비 충돌의 확률이 20%인 경우

6.2 성능 비교 분석

본 시뮬레이션의 결과는 아래 [표3] [표4]와 같다. [표3]은 제안된 기법을 이용해 시뮬레이션 하여 나타낸 결과의 값들이다. [표3] 테이블에서는 세그먼트 수가 증가됨에 따라 CPU의 로드가 증가됨을 알 수 있다. 세그먼트 사용 범위는 1000에서 4000으로 제안하였다.

표3 제안된 모델의 시뮬레이션 결과
Table 3 Simulation Result for proposed model

세그먼트 수	1000	1500	2000	2500	3000	3500	4000
transaction time 20/80	42200	82600	108300	124000	150000	175400	204100
transaction time 50/50	42200	82500	108200	123900	149800	175200	203900
transaction time 80/20	42100	82400	108000	123600	149500	174800	203400

그리고 세부분으로 나누어 각각에 대해 모델을 작성하여 시뮬레이션을 실시하였다. 그 결과를 살펴보면 평균적으로 균등하게 나타났다. 즉 트랜잭션 처리 시간이 균등한 분포로 나타났다.

표4 EV기법 모델의 시뮬레이션 결과
Table 4 Simulation Result for EV model

세그먼트 수	1000	1500	2000	2500	3000	3500	4000
transaction time 20/80	39500	81300	105600	124500	144400	168300	199200
transaction time 50/50	49400	101700	132600	154500	181400	211300	234200
transaction time 80/20	57500	119700	156600	183500	214400	251300	294200

〔표4〕는 EV기법을 이용해 시뮬레이션 하여 나타낸 결과 값들이다. 〔표4〕의 테이블에서는 〔표3〕과 마찬가지로 세그먼트 수가 증가됨에 따라 CPU의 로드가 증가됨을 알 수 있다. 세그먼트 사용 범위는 1000에서 4000으로 하였고, 그리고 세부분으로 나누어 각각에 대해 시뮬레이션을 실시하였다. 그 결과를 살펴보면 전체적으로 제안된 기법보다 높은 수치로 나타났다. 즉 트랜잭션 처리 시간이 많이 소요되었다.

위의 시뮬레이션 결과를 종합적으로 살펴보면, 대부분 세그먼트 수 증가에 따라 시간이 증가됨을 알 수 있다. 그리고 세그먼트 수가 증가됨에도 불구하고 균등 분포를 이루고 있다. 이 표에 나타난 결과의 값을 이용하여 각 단계를 그래프로 나타내어 비교 분석하고자 한다.

본 제안된 기법과 EV기법과 비교하기 위해 세그먼트 충돌 율이 20:80, 50:50, 80:20으로 구분하여 시뮬레이션을 실시하였다.

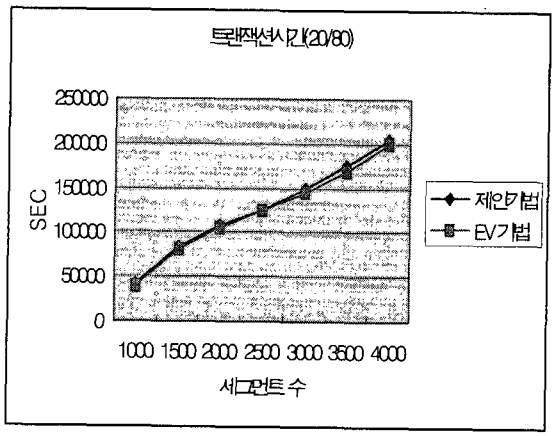

그림6 전체 트랜잭션 처리시간(20:80%)
Figure 6 Total Transaction Processing Time

〔그림6〕은 세그먼트의 크기가 변화됨에 따라 부가되는 CPU로드 측면에서 시뮬레이션을 수행한 결과는 증가됨을 알 수 있다. 또한 각 기법을 비교하면 제안된 기법이 EV기법보다는 시간이 많이 소요되는 것으로 나타났다. 그 이유는 모바일 호스트에서 지수 분포로 발생하는 트랜잭션이 사용하고자 하는 세그먼트 80% 정도가 다른 모바일 호스트에서 사용하지 않으므로 록 충돌 여부를 조사하는 오버헤드가 없어진다. 즉 EV기법은 데이터 충돌여부만 조사하지만 제안된 기법은 충돌여부를 조사하기 위하여 SCL과 록 모드 양립성함수를 사용한다. 그래서 세그먼트 충돌이 적을 경우는 EV기법이 약간 우수하다.

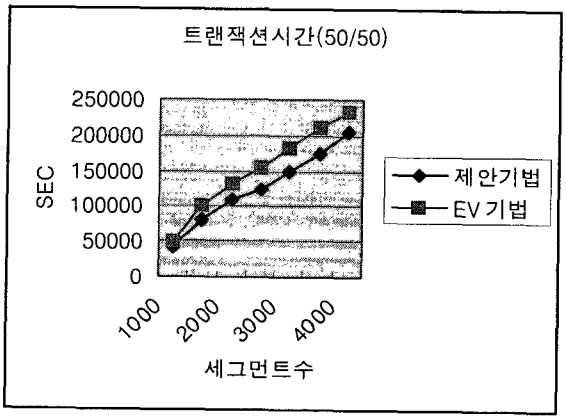

그림7 전체 트랜잭션 처리시간(50:50%)
Figure 7 Total Transaction Processing Time

〔그림7〕은 〔그림6〕과 같이 세그먼트의 크기가 변화됨에 따라 CPU로드 측면에서 시뮬레이션을 수행한 결과는 증가됨을 알 수 있다. 또한 각 기법을 비교하면 제안된 기법이 EV기법보다는 많이 우수한 것으로 나타났다. 그 이유는 모바일 호스트에서 지수 분포로 발생하는 트랜잭션이 사용하고자 하는 세그먼트 50% 정도가 다른 모바일 호스트에서 사용하므로 세그먼트를 철회 및 재 수행에 따른 오버헤드가 발생한다. 즉, EV기법은 같은 데이터를 사용하면 즉시 철회 및 재 수행에 따른 대기 시간이 야기되지만, 제안된 기법은 무조건 세그먼트를 사용하고 그 세그먼트가 충돌하여도 데이터 아이템이 충돌이 하지 않으면 트랜잭션을 수행 할 수 있다. 그 결과 대기 시간을 줄일 수 있다. 〔그림7〕은 제안된 기법이 Exodus기법 보다 우수함을 보여 주고 있다.

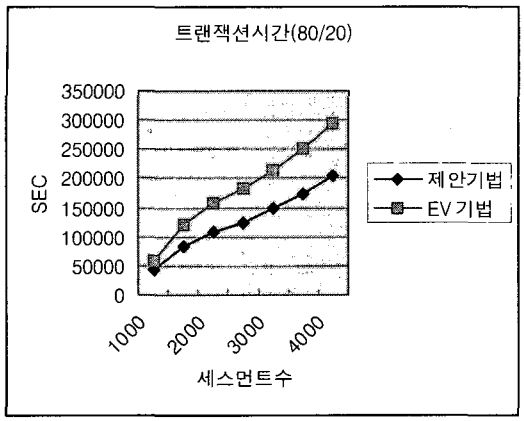

그림8 전체 트랜잭션 처리시간(80:20%)
Figure 8 Total Transaction Processing Time

〔그림8〕은 〔그림7〕과 같이 세그먼트의 크기가 변화됨에 따라 CPU로드 측면에서 시뮬레이션을 수행한 결과는 증가됨을 알 수 있다. 또한 각 기법을 비교하면 제안된 기법이 EV 기법보다는 많이 우수한 것으로 나타났다. 그 이유는 모바일 호스트에서 지수 분포로 발생하는 트랜잭션이 사용하고자 하는 세그먼트 데이터 80% 정도가 다른 모바일 호스트에서 사용하므로 세그먼트를 철회 및 재수행 따른 대기 시간(waiting time)이 발생한다. 즉, EV기법은 같은 데이터를 동시에 수행할 수 없어 대기 시간이 야기되지만, 제안된 기법은 무조건 세그먼트를 사용하고 그 세그먼트가 충돌을 하여도 데이터 아이템이 충돌하지 않으면 트랜잭션을 수행 할 수 있다. 그 결과 대기 시간을 줄일 수 있다. 〔그림8〕은 제안된 기법이 EV 기법 보다 우수함을 보여 주고 있다. 본 기법의 성능이 훨씬 뛰어난 이유는 크게 두 가지로 생각할 수 있다.

첫째, EV기법의 동시성 제어기법은 데이터를 무조건 수행하다가 방송전파에 의한 철회 및 재 수행을 한다. 그래서 여러 모바일 호스트들이 동시에 사용하지 못하므로 대기 시간이 발생한다. 본 기법은 같은 세그먼트에 다른 아이템을 사용할 때 동시에 세그먼트를 사용 할 수 있어 대기 시간을 줄일 수 있었다. 둘째, EV기법에 비해 시스템병렬성과 연산시간이 단축되었다. 즉 같은 데이터를 동시 사용하고, SCL에 의해 신속히 충돌여부를 결정 할 수 있었다. 그 결과 연산시간을 단축함을 시뮬레이션의 의해 확인할 수 있었다.

6.3 실험 결과 분석

비교 분석한 결과를 토대로 기존 기법과 본 기법을 분석 하고자 한다. 기존 기법은 EV기법을 비교 대상으로 제시되었고, 비교 분석에는 동시성 제어 기법을 세 가지로 분류하여 트랜잭션 처리시간을 시뮬레이션 하였다.

표5 시뮬레이션 결과
Table 5 Simulation Result

세그먼트 수	실험 결과
Total Transaction Response Time 20/80	-2.7%
Total Transaction Response Time 50/50	16.8%
Total Transaction Response Time 80/20	30.8%

실험 결과 값은 EV기법과 제안한 모델과 비교했을 때 본 모델의 성능을 백분율로 나타낸 것이다. 〔표5〕를 살펴보면 트랜잭션 처리시간은 54.9%정도 개선되어 본 기법이 상당히 우수함 보여주고 있다. 결과 값에서 볼 수 것 같이 모바일 호스트간의 세그먼트 충돌 율이 작을 경우에는 별 효과를 나타내지 않지만 충돌 율이 커질수록 본 논문에서 제안한 모델이 EV기법보다 더 우수하다는 것을 알 수 있다.

VII. 결 론

본 논문에서는 모바일 호스트/방송서버 데이터베이스 환경에서 발생하는 여러 가지 문제점들을 제시하고, 이 문제점 해결을 위한 데이터 모델과 병목현상을 개선한 낙관적 동시성 제어 기법을 제안하였다. 제안하는 동시성제어는 낙관적 방법이므로 읽기단계에서 아무런 간섭 없이 수행하다가 완료시점에서 검증단계를 통하여 트랜잭션의 충돌 여부를 결정함으로써 데이터 일관성을 유지할 수 있는 기법이다. 또한 방송 전송단위는 세그먼트로 한다. 왜냐하면 다양한 크기의 멀티미디어 데이터를 전송하기 위하여 데이터 아이템보다는 세그먼트단위를 전송단위로 하였다. 시스템 병렬처리를 하기 위하여 여러 모바일 호스트들이 동시에 세그먼트를 사용할 수 있도록 SCL방법을 제안하였다. 그리고 시뮬레이션 모델을 이용한 성능 평가를 통하여 제안 기법이 기존의 기법에 비하여 빠른 트랜잭션의 처리시간을 보임을 알 수 있었다.

참고문헌

[1] Kam-Yiu Lam,Tei-Wei Kuo, Wai-Hung Tsang and Gray c.k. Law "Concurrency Control in Mobile Distributed Real-Time Database Systems",Information Systems, Vol25, No4, pp261-322 ,2000

[2] M. Franklin et. al., "Local Disk Caching for Client-Server Database Systems," Proc. VLDB pp 641-654. 1993

[3] Kam-Yiu Lam, Mei-Wai Au, "Broadcasting Consistent Data to Read-only Transaction from Mobile Clients", The Computer Journal, Vol.45 No.2 pp129-146 2002

[4] Acharya, s., Alonso, R., Franklim, M. and Zdonik, "Broadcast disks: data management for asymmetric communication environment.". Proc. ACM SIGMOD 1993, pp199-210,1993

[5] Bernstein, P.,Hadzilacos, V., and Goodman, N. "Concurrency control and Recovery in Database Systems", Addition -Wesley, 1987

[6] Carey, M., Dewitt, D., Richardson J., Schekita, E. "Storage Management for Objects in EXODUS", in Object-Oriented Concepts, Databases, and Applications, W. Kim F. Lochovsky, eds., Addition-Wesley, 1989.

[7] Daniels, D., Spector A., Thompson, D., "Distributed Logging for Transaction Processing", Proc. ACM SIGMOD Conf. San Francisco, May, 1987.

[8] Franklin, M., Zwilling, M., Tan, C., Carey, M., Dewitt, D. "Crash Recovery in Client-Server EXODUS", TR #1081, Comp Sci Dept., Univ. of Wisconsin-Madison, Mar. 1992.

[9] The Committee for Advanced DBMS Function, "Third Generation Data Base System Maifesto", SIGMOD Record, Vol. 19, No. 3, Sept. 1990

[10] Shanmugasundaram. J., Nithrakashyap, A., Sivasankararn "Efficient Concurrency for Bdisk environment", ACM SIGMOD International conf. on Management of data, pp85-86, 1999

[11] Jin Jing,Omran Bukhres, Ahmed Elmagarmid, "Distributed lock management for mobile transaction", proceedings of the 15th IEEE international conference on Distributed computing systems, May ,1995

[12] Daniel Barbara, Tomasz Imielinski, "Sleepers and Workaholics: caching Strategies in Mobile Environments", VLDB, 1995.

[13] Lee, V.C.S., Kwok-wa Lam, Son, S.H.,"On Transaction processing with partial validation and Time stamp ordering in mobile broadcast environment," IEEE pp 1196-1211, 2000

[14] Victor C.S., Kwok Wa Lam "Efficient Validation of mobile transaction in wireless environments", The Journal of Systems and Software. pp183-194, 2004

[15] SanKeum Lee, "Caching and Concurrency Control in a Wireless Mobile Computing Environments",IEICE Trans. VolE85-D No.8 Aug, 2002

[16] 김대인,황부현 "이동컴퓨팅 환경에서의 데이터 그룹정보를 이용한 동시성제어방법", 정보과학회논문지 제32권3호, pp315-325,2005

저 자 소 개

조 성 제

1997년 2월 : 홍익대학교 전자계산학과 이학박사

2005년 ~ 현재 : 성결대학교 e-비즈니스IT학부 교수

〈관심분야〉모바일 컴퓨팅, 실시간처리, 이동데이터베이스, 전자상거래