

# 이동 컴퓨팅 환경에서 방송정보를 이용한 동시성제어 방법

고승민 김대인 임선모 황부현  
전남대학교 전산학과

{smgo<sup>o</sup>, dikim, bhhwang}@sunny.chonnam.ac.kr, smim@dongkang.ac.kr

## Concurrency Control Method using the Broadcast Information in Mobile Computing Environments

Seungmin Ko<sup>o</sup> Daein Kim Sunmo Im Buhyun Hwang  
Dept. of Computer Science, Chonnam National University  
Field of Computer & Internet, Dongkang College

### 요 약

이동 컴퓨팅 환경에서 이동 호스트는 가능한 한 제한된 대역폭의 사용을 줄이고 이동 트랜잭션의 응답 시간을 향상시키기 위하여 캐시를 사용한다. 그리고 지구국에서는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 주기적으로 갱신 정보로 구성된 무효화 보고서를 방송한다. 그러나 주기적인 무효화 보고서를 사용한 이동 호스트의 캐쉬 일관성 유지 방법은 이동 트랜잭션의 직렬가능한 수행을 보장하기 위하여 이동 트랜잭션의 완료 결정을 지연한다는 문제점이 있다. 본 논문에서는 캐싱된 데이터를 이용하여 이동 트랜잭션을 수행하는 경우 이동 트랜잭션의 직렬가능한 수행을 보장하는 CCM-DI 방법을 제안한다. 제안하는 CCM-DI 방법은 지구국에서 이동 호스트에게 방송한 데이터와 갱신된 데이터와의 교집합을 구하여 공통되는 데이터를 방송함으로써 이동 트랜잭션의 직렬가능한 수행을 보장한다. 또한 제안하는 CCM-DI 방법은 지구국에서 주기적으로 방송하는 무효화 보고서의 크기를 줄일 수 있다.

### 1. 서 론

이동 컴퓨팅 환경은 크게 이동 호스트와 서비스를 제공하는 서버로 구성된다. 그리고 이동 컴퓨팅 환경에서는 데이터를 전달하기 위하여 방송 기법을 이용한다[1]. 이동 컴퓨팅 환경은 기존의 분산 환경에 비하여 호스트의 이동성, 잦은 단절, 제한된 대역폭, 그리고 батери 용량에 제약을 받는다[8]. 그리고 이러한 이동 컴퓨팅 환경을 특성을 고려한 효율적인 이동 트랜잭션 처리 방법으로 캐싱 기법이 제안되었다. 그리고 제한된 대역폭과 같은 이동 컴퓨팅 환경의 특성을 고려한 이동 호스트의 캐쉬 일관성 유지 방법으로 주기적으로 무효화 보고서(invalidation report)를 방송하는 방법이 제안되었다[8]. 그러나 주기적인 무효화 보고서 방법에서 이동 호스트는 지구국에서 데이터 갱신 여부를 무효화 보고서를 수신할 때까지는 알지 못하며, 이로 인하여 이동 트랜잭션의 완료 결정이 지연된다는 문제점을 가지고 있다.

본 논문에서는 주기적인 무효화 보고서 방송을 통하여 이동 호스트의 캐쉬 일관성을 유지하는 경우 이동 트랜잭션의 직렬가능한 수행을 보장하고 빠른 응답 시간을 제공할 수 있는 방법으로 CCM-DI(Currency Control Method using the Data broadcast with Invalidation report) 방법을 제안한다. 제안하는 CCM-DI 방법은 지구국에서 현재 무효화 보고서 방송 구간 동안에 이동 호스트에게 방송한 데이터와 갱신된 데이터와의 교집합을 구하여 공통되는 데이터를 재방송함으로써 이동 트랜잭션의 직렬가능한 수행을 보장한다. 본 논문의 구성은 다

음과 같다. 2장에서는 동시성 제어 기법에 관한 기존의 연구 내용과 문제점을 설명하고 3장에서는 본 연구에서 제안하는 CCM-DI 방법을 설명한다. 마지막으로 4장에서는 결론 및 앞으로의 연구방향에 대하여 기술한다.

### 2. 관련연구

이동 컴퓨팅 환경에서 이동 호스트에게 데이터를 방송하는 방법은 풀 기반(pull based) 방송, 푸시 기반(push based) 방송, 그리고 두 기법을 혼용한 하이브리드 방송으로 분류된다[1,2,6,9]. 풀 기반 방송은 이동 호스트가 요구하는 데이터를 방송하는 방법이고 푸시 기반 방송은 서버에서 방송 스케줄에 따라 데이터를 방송하는 방법이다. 그리고 하이브리드 방송 방법은 이동 호스트의 데이터 요청과 지구국에서의 방송 스케줄을 혼용한 방송 방법이다. 그러나 푸시 기반 방송은 이동 트랜잭션 수행을 위하여 이동 호스트 요청한 데이터를 얻는데 대기 시간이 길어질 수 있다는 단점이 있다.[7]

[5]에서는 풀 기반 방송 환경에서 이동 트랜잭션의 직렬가능한 수행을 보장할 수 있는 방법으로 OCC-UTS<sup>2</sup>(Optimistic Concurrency Control - Update Timestamp Span)를 제안하였다. OCC-UTS<sup>2</sup> 방법은 이동 트랜잭션이 접근한 데이터의 타임스탬프가 모두 같거나 가장 최근에 수신한 무효화 보고서의 타임스탬프보다 작은 경우에 이동 트랜잭션을 바로 완료함으로써 이동 트랜잭션의 응답 시간을 향상시켰다. 그러나 OCC-UTS<sup>2</sup> 방법은 이동 트랜잭션이 접근한 모든 데이터가 가장 최신의 데이터가 아닌 경우에는 직렬가능한 수행이 아닌 것으로 간주하여 철회 결정을 함으로써 잘못된 철회 결정을 내릴 수 있다는 문제점이 있다. 또한 대부분의 경우 무효화

• 본 논문은 2004년도 한국과학기술연구원 지역우수과학자 육성지원연구(R05-2003-000-10532-0)의 지원으로 수행되었음.

보고서 수신 후 이동 트랜잭션의 완료 결정을 내림으로써 이동 트랜잭션의 응답 시간이 길어지게 된다. 그러므로 가능한 한 이동 트랜잭션의 응답 시간을 향상시키고 보다 정확한 이동 트랜잭션의 완료 결정을 내릴 수 있는 방법에 대한 연구가 필요하다.

### 3. CCM-DI 기법

제안하는 CCM-DI 방법의 환경은 다음과 같다.

- 모든 갱신은 서버에서 발생하고 이동 호스트는 읽기 전용 트랜잭션만 수행한다.
- 서버에서 이동 호스트로의 데이터 전송은 풀 기반 방송을 사용한다.
- 이동 호스트는 이동 트랜잭션 수행에 필요한 데이터를 수행 전에 모두 얻어 사용하는 Cache-ahead 방법을 사용한다[3,4].
- 서버는 이동 호스트가 요청하는 데이터를 일정 기간 동안 수집하여 버킷 단위로 방송한다

#### 3.1 서버 알고리즘

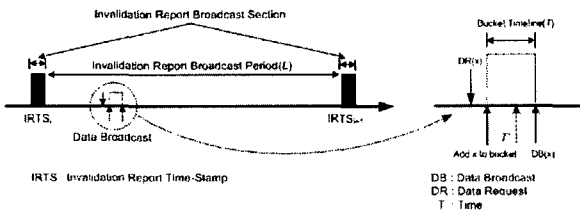


그림1. 데이터 방송과 무효화보고서 방송

제안하는 CCM-DI 방법에서 서버는 그림 1과 같이 현재 무효화 보고서 방송 주기 동안에 서버에서 갱신된 데이터 집합인  $U\_DS$ (Update Data Set)와 이동 호스트에게 방송한 데이터 집합인  $B\_DS$ (Broadcast Data Set)를 유지한다. 또한 두 집합에 공통적으로 포함된 데이터 식별자와 타임스탬프의 쌍으로 이루어진 충돌 데이터 정보인 리스트  $K$ 를 계산한다. 그리고 서버는 이동 호스트가 데이터를 요구하는 경우 충돌 데이터 리스트  $K$ 를 요청 데이터와 함께 전송하며, 리스트  $K$ 를 수신한 이동 호스트는 리스트  $K$ 에 포함된 데이터가 캐쉬에 있는 경우 지구국으로 데이터 재전송을 요청한다. 또한 서버는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 무효화 보고서를 주기적으로 방송한다. 제안하는 CCM-DI 방법에서는 현재 무효화 보고서 방송 주기 동안에 갱신된 데이터 집합 중에서 이동 호스트의 데이터 요구시 함께 전송한 리스트  $K$ 에 포함된 데이터의 갱신 정보를 제거함으로써 무효화 메시지의 크기를 줄일 수 있다. 즉 이동 호스트가 요청한 데이터 전송시 함께 방송된 리스트  $K$ 에 포함된 데이터가 이동 호스트의 캐쉬에 있는 경우에 이동 호스트는 데이터 재전송을 요청하기 때문에 이동 호스트는 최신의 데이터를 캐쉬에 유지하게 되기 때문이다. 그러므로 무효화 보고서는 갱신된 데이터 집합인  $U\_DS$ 에서 리스트  $K$ 에 포함된 데이터 정보를 제거한 데이터 정보로

구성된다. 제안하는 CCM-DI 방법에서의 서버 알고리즘은 그림 2와 같이 기술할 수 있다.

```

1. MSS received the data request from MH
   add request data to bucket;

2. If it is time to broadcast the data bucket or data
   bucket is full
    $K = U\_DS \cap B\_DS$ 
   broadcast data bucket and K
   remove K both  $U\_DS$  and  $B\_DS$ ;
   add K to  $K_w$  ;
   // 이동 호스트의 단절을 위해 K의 사본을 유지
   add requested data to  $B\_DS$ 
    
```

그림2. 서버 알고리즘

#### 3.2 이동 호스트 알고리즘

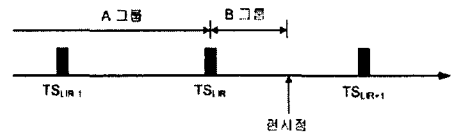


그림3. 이동 호스트의 데이터 그룹

이동 호스트는 가장 최근에 수신한 무효화 보고서의 타임스탬프  $TS_{LIR}$ 를 유지한다. 그리고 그림 3과 같이 이동 호스트에 캐싱된 데이터는 두 개의 그룹으로 분류할 수 있다. 그림 3에서 A그룹은  $TS_{LIR}$  이전에 캐싱한 데이터 그룹이고 B그룹은  $TS_{LIR}$  이후에 캐싱한 데이터 그룹이다. 즉 A그룹의 데이터는  $TS_{LIR}$  시점의 무효화 보고서를 수신함으로써 데이터 일관성을 검증받은 데이터 그룹이다. 그러므로 이동 트랜잭션이 접근한 데이터가 모두 A그룹의 데이터이면 이동 호스트는 서버와의 별도의 통신 없이 즉시 완료할 수 있다. 그러나 이동 트랜잭션이 B그룹의 데이터에 접근했을 경우 데이터의 정확성 여부를 알 수 없으며 OCC-UTS<sup>2</sup>나 다른 방법들과 마찬가지로 무효화 보고서를 수신할 때까지 이동 트랜잭션의 완료 결정은 지연한다. 그러나 제안하는 CCM-DI 방법에서는 리스트  $K$ 를 이용하여 B그룹의 데이터에 대하여 데이터의 정확성을 보장함으로써 이동 트랜잭션을 즉시 완료할 수 있다.

그림 4에서 이동 호스트에 이동 트랜잭션  $MT_1$ 이 제출되었으며 캐쉬에는 데이터  $x$ 와  $y$ 가 존재한다. 그리고 이동 트랜잭션  $MT_1$  수행 후 이동 트랜잭션  $MT_2$ 가 제출되었으며  $MT_2$ 를 수행하기 위하여 캐쉬에 없는 데이터  $z$ 를 서버로 요청한다. 서버는 이동 호스트가 요청한 데이터  $z$ 를 데이터 버킷에 넣고 DB<sub>2</sub>시점에 요청 데이터  $z$ 와 리스트  $K$ 를 함께 전송한다. 그리고 데이터와 리스트  $K$ 를 수신한 이동 호스트는 리스트  $K$ 에 데이터  $x$ 가 포함되어 있으므로 데이터  $x$ 를 지구국으로 재요청하여 이동 트랜잭션  $MT_2$ 를 수행한다. 즉 다음에 방송되는 무효화 보고서 수신 시점까지 대기하지 않고 바로 데이터를 재요청

하여 이동 트랜잭션을 수행할 수 있다. 만일 리스트  $K$ 에 이동 트랜잭션이 접근한 데이터가 포함되어 있지 않으면 이동 트랜잭션은 즉시 완료한다. 이처럼 리스트  $K$ 는 무효화 보고서 방송 구간에 서버에서 갱신된 데이터 집합과 이동 호스트가 요청한 데이터 집합간의 충돌을 탐지하여 이동 호스트의 데이터 요청시 함께 전송함으로써 이동 트랜잭션의 직렬가능한 수행을 보장하고 응답 시간을 향상 시킬 수 있다. 그러나 기존의 OCC-UTS<sup>2</sup> 방법은 다음에 방송되는 무효화 보고서 수신시점  $IRTS_{i+1}$ 까지 이동 트랜잭션을 지연시킨 후 철회시킨다.

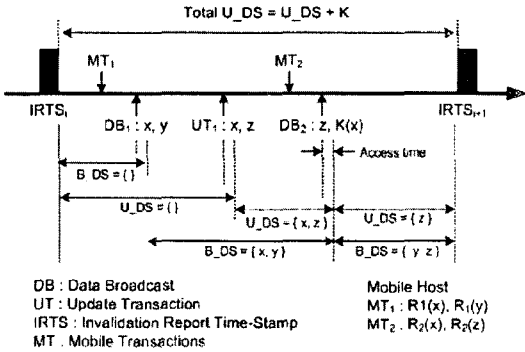


그림 4. 이동 호스트와 서버 알고리즘 예제

그리고 본 연구에서 제안하는 CCM-DI 방법에서의 이동 호스트 알고리즘은 그림 5와 같이 기술할 수 있다.

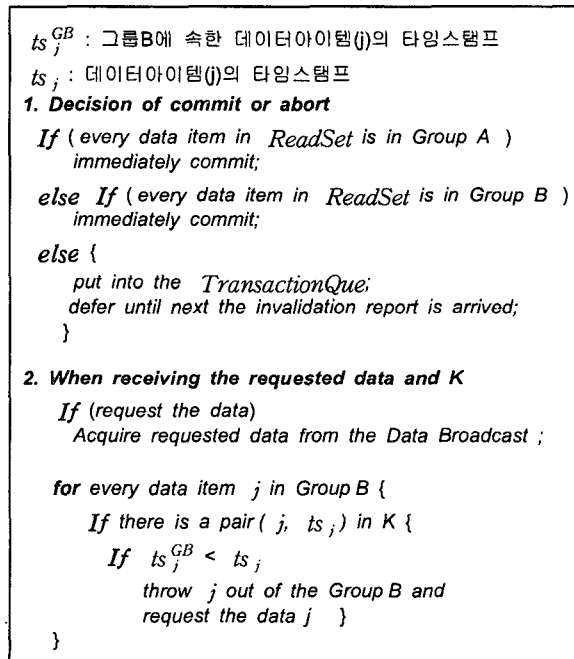


그림 5. 이동 호스트 알고리즘

본 연구에서는 이동 컴퓨팅 환경에서의 데이터 방송을 이용하여 이동 트랜잭션의 직렬가능한 수행을 보장하는 CCM-DI 방법을 제안하였다. 제안한 CCM-DI 방법은 기존의 OCC-UTS<sup>2</sup> 방법에 비하여 보다 많은 경우에 이동 트랜잭션을 즉시 완료함으로써 빠른 응답 시간을 제공할 수 있다. 즉 제안한 CCM-DI 방법은 무효화 보고서 방송 주기 동안에 이동 호스트에서 캐싱한 데이터와 이동 호스트가 요구한 데이터들간의 충돌을 분석하여 요청 데이터 방송시 데이터 충돌에 관련된 리스트  $K$ 를 함께 방송함으로써 이동 트랜잭션 수행의 정확성 여부를 즉시 판단할 수 있다. 그러므로 제안한 CCM-DI 방법은 이동 트랜잭션의 빠른 응답 시간을 제공할 수 있으며 이동 호스트의 캐쉬 일관성을 유지하기 위하여 방송되는 무효화 보고서의 크기도 줄일 수 있다는 장점을 가지고 있다.

향후 연구 방향으로 이동 트랜잭션이 두 그룹의 데이터에 각각 접근하는 경우 이동 호스트에서 자치적으로 충돌을 탐지함으로써 가능한 한 서버와의 통신량을 줄이는 방법에 대하여 연구하고자 한다.

참고 문헌

- [1] S. Acharya, M. Franklin, and S. Zdonik. "Balancing push and pull for data broadcast." In Proceedings of ACM SIGMOD Conference on Management of Data, pages 183-194, Tucson, AZ, USA, May 1997
- [2] T. Imielinski and S. Viswanathan. "Adaptive wireless information systems." In Proceedings of the Special Interest Group in DataBase Systems (SIGDBS) Conference, pages 19-41, Tokyo, Japan, October 1994
- [3] 김치연, 황부현, "이동 컴퓨팅 환경에서 캐쉬 록을 이용한 동시성 제어 기법", 한국정보처리학회 논문지, Vol. 5, No. 2, 1998
- [4] 김치연, "이동 컴퓨팅 환경에서 트랜잭션의 동시성 제어 방법", 전남대학교 박사학위논문, August, 1999
- [5] SangKeun LEE "Caching and Concurrency Control in a Wireless Mobile Computing Environment" *IEICE TRANS. INF. & SYST.*, Vol.E85-D, No. 8 August 2002
- [6] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Data Broadcast." in Proceedings of ACM SIGMOD, Tucson, Arizona, May 1997.
- [7] W. -C. Lee, Q. L. Hu, and D. L. Lee " Channel Allocation Methods for Data Dissemination in Mobile Computing Environments." 6th International Symposium in High Performance Distributed Computing(HPDC '97) Portland, USA, August 1997
- [8] D.Barbara and T.Imielinski, "Sleepers and Workaholics : Caching in Mobile Distributed Environments," Proc. 1994 ACM-SIGMOD Int'l Conf. Management of Data, pp.1-12, June 1994
- [9] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks : Data Management for Asymmetric Communication Environments" Proc. of ACM SIGMOD, pp. 199-210, 1995.