이동 컴퓨팅 환경에서 데이타 그룹 정보를 이용한 동시성 제어 방법

(A Concurrency Control Method using Data Group Information in Mobile Computing Environments)

김 대 인 † 황 부 현 ††
(Daein Kim) (Buhyun Hwang)

요 약 이동 컴퓨팅 환경에서 이동 호스트는 제한된 대역폭을 효율적으로 사용하고 이동 트랜잭션의 응답 시간을 향상시키기 위하여 캐쉬를 사용한다. 그리고 이동 호스트에 캐쉥된 데이타가 서버에서 갱신되는 경우 서버는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 무효화 메시지를 방송한다. 그러나 주기적인 무효화 메시지 방송을 사용한 이동 호스트의 캐쉬 일관성 유지 방법은 이동 트랜잭션의 완료 결정을 무효화 메시지 수신 시점으로 지연함으로써 이동 트랜잭션의 응답 시간이 길어진다는 문제점이 있다. 본 연구에서는 캐쉥된 데이타를 사용하여 이동 트랜잭션을 수행하는 경우에 이동 트랜잭션의 응답 시간을 향상시킬 수 있는 UGR-MT 방법을 제안한다. 제안하는 UGR-MT 방법은 데이타 그룹 정보를 사용하여 무효화 메시지 수신 이전에 이동 트랜잭션의 완료 결정을 내림으로써 이동 트랜잭션의 응답 시간을 향상시킬 수 있다. 또한 제안하는 방법은 이동 호스트의 단절 시간이 무효화 메시지 방송 구간보다 긴 경우에 발생할 수 있는 이동 호스트의 전체 캐쉬 내용의 버림을 방지함으로써 캐쉬의 효율성을 높일 수 있다.
키워드 : 이동 컴퓨팅, 캐쉬, 동시성 제어, 무효화 메시지, 데이타 그룹

Abstract In mobile computing environments, a mobile host caches the data items to use the bandwidth efficiently and improve the response time of transactions. If the data items cached in mobile host are updated in the server, the server broadcasts an invalidation report for maintaining the cache consistency of mobile hosts. However, this method has a problem that the response time of mobile transactions can be long since their commit decision is delayed until receiving the invalidation report from the server. In this paper, we propose the UGR-MT method for improving the response time of mobile transactions. As the UGR-MT method can make a commit decision by using the data group information before receiving the invalidation report, the response time of mobile transactions can be improved. Also our method can improve the cache's efficiency since it prevents all the contents of a cache from being invalidated in the case that the disconnection of a mobile host is longer than the broadcast period of invalidation report.
Key words : Mobile Computing, Cache, Concurrency Control, Invalidation report, Data Group

1. 서 론

호스트의 이동성, 잦은 단절, 낮은 대역폭(bandwidth)과 제한된 배터리 용량 등의 이동 컴퓨팅 환경의 특성을 고려한 기법으로 서버에 있는 데이타베이스의 일부분을 이동 호스트로 가져오는 캐쉥 기법이 제안되었다[1-5]. 그러나 서버는 데이타가 갱신되는 경우 이동 호스트의 캐쉬 일관성(consistency)을 유지하기 위하여 이동 호스트에게 데이타 갱신 여부를 알려주어야 한다. 이동 컴퓨팅 환경에서 서버와 이동 호스트는 데이타 방송(broadcasting)을 통하여 정보를 교환하며, 서버와 이동 호스트 사이의 잦은 데이타 방송은 제한된 대역폭에 병목현상(bottleneck)을 발생시키는 요인이 된다. 그러므로 서버와 이동 호스트 사이에 가능한 한 통신을 줄이면서 이동 호스트에게 갱신 정보를 알려주는 방법에 대한 연구가 필요하다. 제한된 대역폭과 같은 이동 컴퓨팅 환경

· 본 연구는 2004년도 한국과학재단 지역우수과학자 육성지원연구(R05-2003-000-10532-0)의 지원으로 수행되었음
† 학생회원 : 전남대학교 전자컴퓨터정보통신공학부 교수
dikim@chonnam.ac.kr
†† 종신회원 : 전남대학교 전산학과 교수
bhhwang@chonnam.ac.kr
논문접수 : 2004년 9월 10일
심사완료 : 2005년 1월 28일

의 특성을 고려한 이동 호스트의 캐쉬 일관성 유지 방법으로 일정 기간 동안 서버에서 발생한 갱신 정보로 구성된 무효화 메시지(invalidation report)를 주기적으로 방송하는 방법이 제안되었다[1,2,6]. 그러나 이러한 방법에서 이동 호스트는 데이타 갱신 여부를 무효화 메시지를 수신할 때까지 알지 못하며, 이동 트랜잭션의 직렬가능한 수행을 보장하기 위하여 무효화 메시지를 수신할 때까지 이동 트랜잭션의 완료 결정이 지연된다는 문제점이 있다[6,7]. 본 연구에서는 이동 컴퓨팅 환경에서 주기적으로 무효화 메시지를 방송하여 이동 호스트의 캐쉬 일관성을 유지하는 경우 이동 트랜잭션의 직렬가능한 수행을 보장할 수 있는 방법을 제안한다. 제안하는 방법은 무효화 메시지 수신 이전에 이동 트랜잭션의 완료 결정을 내리기 위하여 데이타 그룹 정보를 사용하며, 이동 호스트의 단절로 인한 전체 캐쉬 내용의 무효화를 방지할 수 있다.

본 연구의 내용은 다음과 같다. 제2장에서는 이동 호스트의 캐쉬 일관성 유지 및 이동 트랜잭션의 직렬가능한 수행에 관한 기존의 연구 내용을 기술하고, 제3장에서는 이동 트랜잭션 수행의 직렬성 위배 예를 살펴본다. 제4장에서는 데이타 그룹 정보를 사용한 이동 트랜잭션의 직렬가능한 수행 보장과 단절시 이동 호스트의 캐쉬 일관성 유지에 대한 방법을 제시하고, 제5장에서는 해석적 모델을 이용하여 제안하는 방법의 성능을 분석한다. 끝으로 제6장에서는 본 연구의 결론과 향후 연구 방향을 기술한다.

2. 관련 연구

참고문헌 [1,2]에서는 서버에서 주기적으로 무효화 메시지를 방송하여 이동 호스트의 캐쉬 일관성을 유지하는 방법으로 TS(broadcasting TimeStamp), AT(Amnesic Terminals), SIG(SIGnatures) 방법을 제안하였다. 제안 방법에서 이동 호스트는 현재 무효화 메시지 방송 주기 동안에 제출된 질의(query)를 리스트 형태로 유지한다. 그리고 서버에서는 무효화 메시지를 주기적으로 방송하여 이동 호스트의 캐쉬 일관성을 유지하고 질의를 수행한다.

참고문헌 [6]에서는 서버에서 데이타 갱신 정보를 비트열로 구성하여 주기적으로 방송하는 BS(Bit Sequence) 방법을 제안하였다. BS 방법은 이동 호스트의 단절을 고려하여 한 구간 이상의 무효화 메시지 방송 주기 동안의 갱신 정보를 비트 열로 구성하여 방송함으로써 이동 호스트의 캐쉬 일관성을 유지한다. 그러나 참고문헌[1,2,6]에서 제안한 캐쉬 일관성 유지 방법은 이동 호스트의 캐쉬에 있는 데이타의 정확성을 검증하기 위하여 제출되는 질의를 무효화 메시지 수신 이후에 처리함으로써 질의 수행이 지연되는 문제가 있다[7,8].

참고문헌 [8]에서는 이동 트랜잭션이 읽기 연산으로만 구성된 경우에 이동 트랜잭션의 직렬가능한 수행을 보장하는 방법으로 UFO(Update First Order) 방법을 제안하였다. UFO 방법은 방송 트랜잭션을 사용하여 이전에 방송한 데이타 집합과 이동 트랜잭션이 접근한 데이타 집합 사이의 교집합을 구하여 교집합이 공집합인 이동 트랜잭션은 완료하고 공집합이 아닌 이동 트랜잭션은 수행이 올바르지 않은 것으로 간주하여 데이타 재방송을 요구한다. 그러나 잦은 데이타 재방송은 대역폭 사용을 증가시키는 요인이 되며 데이타 방송 스케줄의 재구성으로 인한 부가적인 비용이 발생한다. 또한 직렬화 그래프에서 사이클을 발생시킨 이동 트랜잭션이 접근하는 데이타를 한 시점에 방송하지 않으면 데이타가 재방송되어도 유사한 형태의 사이클이 발생하여 이동 트랜잭션의 연속 철회가 발생할 수 있다.

참고문헌 [7]에서는 가능한 한 이동 트랜잭션 수행의 빠른 응답 시간을 제공하는 방법으로 OCC-UTS^2 (Optimistic Concurrency Control with Update Time-Stamp Span) 방법을 제안하였다. OCC-UTS^2 방법은 이동 트랜잭션이 접근한 데이타의 타임스탬프가 모두 같거나 가장 최근에 수신한 무효화 메시지의 타임스탬프보다 더 작은 경우에 이동 트랜잭션을 바로 완료함으로써 이동 트랜잭션의 응답 시간을 향상시켰다. 그러나 OCC-UTS^2 방법은 이동 트랜잭션의 수행이 즉시 완료 조건을 만족하지 못하는 경우 접근한 모든 데이타가 가장 마지막으로 갱신된 데이타가 아니면 이동 트랜잭션의 수행이 직렬가능하지 못한 것으로 간주하여 철회 결정을 내린다. 그러나 이러한 방법은 올바르게 수행된 이동 트랜잭션도 철회시킬 수 있다는 문제가 있으며, 불필요한 철회 결정은 이동 트랜잭션의 재수행으로 인한 많은 추가 비용이 발생하며 제한된 대역폭의 사용이 증가한다는 문제점이 있다.

3. 문제 제시

이동 컴퓨팅 환경에서 서버는 이동 호스트의 데이타 요청을 일정 기간 동안 수집한 후 방송 데이타 스케줄을 구성하여 데이타를 방송하며 이러한 역할을 수행하는 트랜잭션을 방송 트랜잭션(broadcast transaction)이라고 한다. 하나의 방송 트랜잭션은 이동 트랜잭션의 빠른 수행을 위하여 하나의 무효화 메시지 방송 주기 동안에 여러 번 수행될 수 있다[8]. 방송 트랜잭션은 기본적으로 이동 호스트가 요구한 데이타로 방송 스케줄을 구성하며, 대역폭에 여유가 있는 경우 이동 호스트가 자주 요구하거나 서버에서 자주 갱신되는 데이타를 방송 스케줄에 포함하여 구성할 수 있다. 이동 컴퓨팅 환경에

서 이동 호스트의 요청이 없더라도 이동 호스트의 요청 횟수나 서버에서의 갱신 횟수를 고려하여 이동 호스트에게 방송하는 데이타를 방송 기반 데이타(broadcast data)라고 하며 이동 호스트의 요청으로 인하여 방송하는 데이타를 요구 기반 데이타(on-demand data)라 한다[9,10]. 그러나 이동 컴퓨팅 환경에서 이동 호스트의 캐쉬 사용과 주기적인 무효화 메시지 방송을 통한 캐쉬 일관성 유지 방법은 제한된 대역폭을 효율적으로 사용하기 위하여 필수적이지만 다음 3.1절의 예와 같이 이동 트랜잭션의 직렬가능한 수행을 보장할 수 없다는 문제점이 있다.

3.1 이동 트랜잭션 수행의 직렬성 위배

그림 1은 이동 호스트가 캐쉬에 없는 데이타를 서버로 요청하여 이동 트랜잭션을 수행하는 경우에 발생할 수 있는 이동 트랜잭션 수행의 직렬성 위배 예를 보여준다.

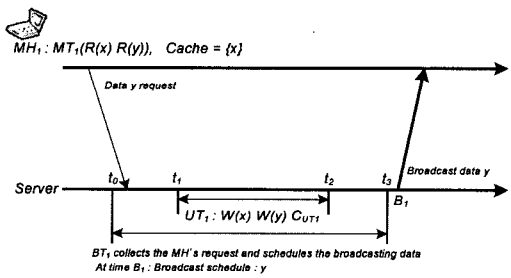

그림 1 이동 트랜잭션 수행의 직렬성 위배 예

이동 호스트 MH_1에 데이타 x와 y에 접근하는 이동 트랜잭션 MT_1이 제출되고 MH_1의 캐쉬에 x만 있는 경우 MH_1은 캐쉬에 없는 y를 서버로 요청한다. 그리고 서버에서는 방송 트랜잭션 BT_1이 t_0에서 t_3동안의 데이타 요청 메시지를 수집하여 t_3시점에 y를 포함하는 방송 스케줄을 구성한다. 동시에 서버에서는 x와 y를 갱신하는 갱신 트랜잭션 UT_1이 시점 t_1과 t_2사이에 수행된다. 그러나 서버는 이동 호스트의 캐쉬 상태를 알지 못하며 가능한 한 이동 트랜잭션이 최신의 데이타에 접근하도록 하기 위하여 구성된 방송 스케줄에 따라 시점 B_1에 방송되는 y는 UT_1의 갱신값을 방송한다. 그러므로 MT_1이 접근하는 x는 UT_1 이전의 값이지만($MT_1 \rightarrow UT_1$), y는 UT_1의 갱신값이며($UT_1 \rightarrow MT_1$) 직렬화 그래프에 사이클이 발생함으로써 MT_1의 수행은 직렬성에 위배된다. 그림 1과 같은 이동 트랜잭션 수행의 직렬성 위배를 막기 위하여 참고문헌 [1,2,4]에서는 현재 무효화 메시지 방송 주기 동안에 제출된 이동 트랜잭션을 즉시 수행하지 않고 지연한 후 무효화 메시지를 수신하여 데이타의 정확성을 검증한 후 수행한다. 그러므로 이동 트랜잭션의 응답 시간이 지연되며 무효화된 데이타 요청으로 인한 대역폭의 사용이 무효화 메시지 수신 시점으로 몰림으로써 병목현상이 발생할 수 있다는 문제점이 있다. 또한 참고문헌 [8]에서는 BT_1 수행 이전부터 MH_1의 캐쉬에 있는 x가 바로 이전에 수행된 방송 트랜잭션에 의하여 방송되지 않았으면 직렬성 위배를 검증할 수 없다. 참고문헌 [7]에서는 MT_1이 접근한 데이타의 타임스탬프가 서로 다르므로 즉시 완료할 수 없으며 완료 결정을 무효화 메시지 수신 시점으로 지연한다. 그리고 무효화 메시지 수신 후 MT_1이 무효화되는 y에 접근하였으므로 MT_1은 철회된다.

주기적인 무효화 메시지를 사용하여 이동 호스트의 캐쉬 일관성을 유지하는 경우, 이동 호스트는 무효화 메시지를 수신할 때까지 서버에서 발생한 갱신 정보를 알지 못한다. 그러므로 그림 1의 예와 같이 이동 트랜잭션과 서버에서 수행되는 갱신 트랜잭션 사이의 충돌로 인한 직렬화 그래프에 사이클이 발생하여 이동 트랜잭션의 수행이 직렬성에 위배될 수 있다. 그러므로 이동 호스트는 제출된 이동 트랜잭션이 현재 방송 주기 동안에 캐쉬된 데이타에 접근하는 경우 이동 트랜잭션의 직렬가능한 수행을 보장하기 위하여 이동 트랜잭션의 완료 결정 여부를 서버에서 방송하는 무효화 메시지를 수신할 때까지 지연시킨다. 그리고 이동 호스트는 무효화 메시지를 사용하여 캐쉬된 데이타의 유효성을 검사하여 무효화된 데이타에 접근하는 이동 트랜잭션들은 철회 결정을 내린다[1,2,7]. 그러나 이러한 방법은 그림 2의 예와 같이 잘못된 철회 결정이 내려질 수 있다는 문제점이 있다.

그림 2에서 이동 호스트 MH_1은 이동 트랜잭션 MT_1을 수행하기 위하여 캐쉬에 없는 데이타 x, y를 요청하고 서버에서는 방송 트랜잭션 BT_1이 시점 t_0에서 t_1까지 수행되어 이동 호스트의 요청 정보를 수집하여 x와 y를 포함하는 방송 스케줄을 구성한다. 그리고 BT_1은 B_1시점에 구성된 스케줄에 따라 데이타 x와 y를 이동 호스트에게 방송하며, 데이타 방송 후 서버에서 y와 z를 갱신하는 갱신 트랜잭션 UT_1이 t_2와 t_3사이에 수행된다. MT_1은 서버에서 B_1시점에 방송한 x와 y를 사용하여 수행되었으며 일관성 있는 데이타에 접근하였으므로 완료되어야 한다. 그러나 x와 y의 갱신 타임스탬프가 다르며 기존의 OCC-UTS[2] 방법과 같은 경우 이동 트랜잭션의 즉시 완료 조건을 만족시키지 못하므로 MT_1의 완료 결정은 서버에서 방송하는 무효화 보고서 수신 시점으로 지연된다. 그러나 서버에서 데이타 방송 후 UT_1에 의하여 y와 z는 갱신되며($B_1<TS(y),TS(z)$) B_2 시점에 방송되는 무효화 메시지를 수신한 이동 호스트는 y의 타임스탬프가 수신한 무효화 메시지에 포함된 y의 타임

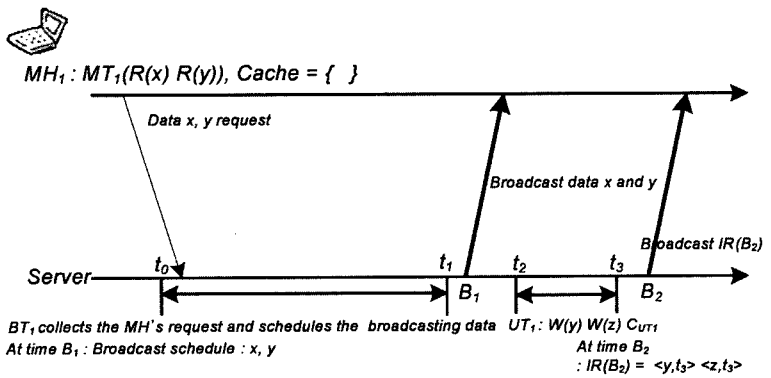

그림 2 이동 트랜잭션 수행의 잘못된 결정 예

스탬프보다 작으므로 y를 무효화한다. 그리고 이동 호스트는 무효화된 y에 접근하였으므로 MT_1을 철회시킨다. 그러나 MT_1의 철회는 잘못된 결정이며 더욱더 정확한 이동 트랜잭션 수행의 정확성을 검증할 수 있는 방법에 대한 연구가 필요하다.

본 연구에서는 데이타 그룹 정보를 사용하여 이동 호스트의 캐쉬 일관성을 유지하고 이동 트랜잭션의 직렬 가능한 수행을 보장하는 방법으로 UGR-MT(Using Group Report-for Mobile Transaction) 방법을 제안한다. UGR-MT 방법에서 이동 호스트는 캐쉬된 데이타를 사용하여 이동 트랜잭션을 수행하며, 이동 트랜잭션이 캐쉬에 없는 데이타를 요구하는 경우 서버로 데이타 요청 메시지를 보낸다. 서버에서는 방송 트랜잭션이 수행되어 이동 호스트의 데이타 요청 정보를 일정 기간 동안 수집하여 방송하며, 이동 트랜잭션의 빠른 수행을 위하여 하나의 무효화 메시지 방송 주기 동안에 방송 트랜잭션은 여러 번 수행된다고 가정한다.

제안하는 방법은 이동 트랜잭션의 완료 결정 여부를 가능한 한 무효화 메시지 수신 이전에 결정할 수 있도록 하기 위하여 데이타 그룹 개념을 사용한다. UGR-MT 방법에서 데이타는 갱신률 및 갱신 성향(pattern) 등을 고려하여 그룹화 되어있다고 가정한다. 또한 UGR-MT 방법에서 이동 트랜잭션은 읽기 전용 트랜잭션이며 모든 갱신 연산은 서버에서만 수행된다고 가정한다[1,2,6-8]. 그리고 서버는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 무효화 메시지와 윈도우 메시지를 방송하며 이동 호스트의 데이타 요청에 따른 데이타 메시지와 그룹 메시지를 방송한다. 무효화 메시지는 일정 주기 동안에 서버에서 발생한 갱신 정보로 구성되며, 윈도우 메시지는 이동 호스트의 단절에 대비하여 하나 이상의 무효화 메시지 내용으로 구성된다. 그리고 데이타 메시지는 방송 트랜잭션에 의하여 방송되는 이동 호스트의 요청 데이타이며 그룹 메시지는 각 그룹에 포함된 데이타 갱신에 대한 시간 정보로 구성된다.

4. UGR-MT 방법

4.1 데이타 그룹화

참고문헌 [11]에서는 데이타의 갱신율 및 이동 접근율에 따라 각각이 데이타들을 그림 3과 같이 *HH(Hot update Hot demand), HC(Hot update Cold demand), CH(Cold update Hot demand),* 그리고 *CC (Cold update Cold demand)* 그룹으로 분류하였다.

	Hot Demand	*Cold Demand*
Hot Update	*HH*	*HC*
Cold Update	*CH*	*CC*

그림 3 데이타 그룹의 분류

제안하는 UGR-MT 방법에서도 각각의 데이타들은 데이타의 갱신율과 접근율 등을 데이타 마이닝 기법으로 분석한 후 가능한 한 갱신율과 접근율이 유사한 데이타들이 같은 그룹으로 구성된다고 가정한다. 일반적으로 그룹에 포함되는 데이타 수가 많을수록 그룹을 이용한 알고리즘의 성능은 향상되지만 그룹 메시지 크기가 많아짐으로써 그룹 관리에 대한 비용이 커지는 반면에 그룹의 수가 작아지면 하나의 그룹에 많은 데이타가 포함됨으로써 무효화되는 데이타가 많아지는 특성이 있다[11]. 그러므로 제안하는 방법에서는 이러한 특성을 고려하여 적절한 수의 그룹으로 데이타는 그룹화 되어 있으며, 이동 호스트는 초기에 데이타의 그룹에 대한 정보를 알고 있으며 데이타 그룹 구성이 변경되는 경우 서버에서는 그룹 변경 정보를 다른 정보보다 최우선으로 하여 이동 호스트에게 방송한다고 가정한다.

4.2 UGR-MT

제안하는 방법에서 서버는 무효화 메시지(*IR*, Invalidation Report), 윈도우 메시지(*WR*, Window Report), 데이타 메시지(*DR*, Data Report), 그리고 그룹 메시지(*GR*, Group Report)를 방송한다. 이동 호스트는 캐쉉된 데이타를 사용하여 이동 트랜잭션을 수행하며, 서버는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 무효화 메시지 $IR(B_i)$, 또는 윈도우 메시지 $WR(B_i)$를 방송한다. 서버에서 시점 B_i에 방송하는 무효화 메시지 $IR(B_i)$와 윈도우 메시지 $WR(B_i)$는 다음과 같이 정의된다.

정의 1. 무효화 메시지 $IR(B_i)$와 윈도우 메시지 $WR(B_i)$

$$IR(B_i)=\{[j,\ TS(j)]\mid j\in D,\text{ and } B_i-L<TS(j)<B_i\},$$

$$WR(B_i)=\{[j,\ TS(j)]\mid j\in D,\text{ and } B_i-W<TS(j)<B_i\}$$

D는 데이타베이스에 있는 데이타의 집합이고, j는 무효화 메시지 방송 주기 L, 또는 W(W=N×L) 동안에 서버에서 갱신된 데이타 식별자이며, TS(j)는 데이타 j를 마지막으로 갱신한 트랜잭션의 완료 시간을 의미한다.

서버에서는 방송 트랜잭션을 수행하여 이동 호스트의 데이타 요청을 수집하여 다음에 방송할 데이타 방송 스케줄을 구성한다. 서버에서 시점 B_{di}에 방송하는 데이타 메시지 $DR(B_{di})$는 다음과 같이 정의된다.

정의 2. 데이타 메시지 $DR(B_{di})$

$$DR(B_{di})=\{[j,\ TS(j)]\mid j\in D\}$$

D는 데이타베이스에 있는 데이타의 집합이고, j는 이동 호스트의 요청 데이타 값이며, TS(j)는 데이타 j를 마지막으로 갱신한 트랜잭션의 완료 시간을 의미한다.

제안하는 방법에서 데이타는 갱신율 및 접근율에 따라 그룹화 되었으며, 서버에서 시점 B_i에 방송하는 그룹 메시지 $GR(B_i)$는 다음과 같이 정의된다.

정의 3. 그룹 메시지 $GR(B_i)$

$$GR(B_i)=\{[j,\ TS_f(j),\ TS_l(j)]\mid j\in G$$

$$\text{and } TS_f(j),\ TS_l(j)<B_i\}$$

G는 데이타 그룹의 집합이고, j는 갱신된 데이타가 속하는 그룹 식별자이며, $TS_f(j)$는 방송 주기 L 동안에 그룹 j에 포함된 데이타를 처음으로 갱신한 트랜잭션의 완료 시간이며, $TS_l(j)$는 그룹 j에 포함된 데이타를 마지막으로 갱신한 트랜잭션의 완료 시간을 의미한다.

제안하는 UGR-MT 방법에서 그룹 메시지는 그룹 j에 속하는 데이타는 $TS_f(j)$ 이전과 $TS_l(j)$ 이후에 갱신되지 않았음을 의미하며 이동 트랜잭션의 즉시 완료 결정 정보로 사용된다. 그리고 그룹 메시지는 서버에서 구성되며, 그룹 메시지는 다른 메시지 방송시 함께 방송된다. 그리고 그룹 j에 속하는 데이타 x를 갱신 트랜잭션 UT_1이 갱신하는 경우 그룹 j의 $TS_l(j)$에 UT_1의 완료 시간을 할당한다. 이 때 UT_1의 갱신이 현재 무효화 메시지 방송 구간 동안에 발생한 그룹 j에 속하는 데이타의 첫 갱신인 경우, 그룹 j의 $TS_f(j)$에도 UT_1의 완료 시간을 할당한다. 그리고 현재 무효화 메시지 방송 구간 동안에 갱신이 발생하지 않은 그룹 j의 $TS_f(j)$에는 그룹 메시지 방송 시간 B_i을 할당하며 $TS_l(j)$은 변경되지 않는다. UGR-MT 방법에서 그룹 j의 데이타 x를 UT_1이 갱신하는 경우 방송 시점에 B_i에 방송되는 그룹 메시지 $GR(B_i)$의 구성은 알고리즘 1과 같이 기술할 수 있다.

B_{IL} : The last broadcast time MH received an invalidation report
TS(x) : Timestamp of data item x in the server
1. A transaction updates data item x in group j
If *($TS_l(j) < B_{IL}$) then*
  $TS_f(j)$ = TS(x);
$TS_l(j)$ = $TS(UT_1)$;
2. When broadcasting the group report, at time B_i
For each *group index j of the group report*
  If *($TS_l(j) < B_{IL}$)* ***then***
    $TS_f(j)$ = Current group report broadcasting time B_i;
Broadcast *the group report $GR(B_i)$;*

알고리즘 1. UGR-MT에서의 그룹 메시지 구성 알고리즘

그림 4는 알고리즘 1을 사용한 그룹 메시지 구성의 예를 보여준다. 그림 4에서 데이타 x와 y는 그룹 G_1, 데이타 z는 그룹 G_2, 그리고 데이타 w는 그룹 G_3에 속하며 갱신 트랜잭션 $UT_1(W(x)W(y))$, $UT_2(W(y))$, $UT_3(W(z))$가 수행되어 각각 t_1, t_2, t_3 시점에 완료되었다. 그리고 그룹 메시지는 B_1과 B_2에 방송되며 z의 갱신은 현재 무효화 메시지 방송 구간 동안에 발생한 G_2의 첫 번째 갱신이라고 가정하자. 그림 4에서 G_1에 속하는 x와 y는 UT_1, UT2에 의해 갱신되었으며 알고리즘 1에 의하여 $TS_f(G_1)$와 $TS_l(G_1)$는 각각 10과 12가 된다. G_2에 속하는 z는 UT_3에 의해 갱신되었으며 z의 갱신은 G_2에 포함된 데이타의 무효화 메시지 방송 구간 동안의 첫 갱신이므로 $TS_f(G_2)$와 $TS_l(G_2)$ 모두 14가 된다. 그리고 방송 시점 B_2까지 G_3에 포함되는 데이타는 갱신되지 않았으며 G_3의 $TS_f(G_3)$는 그룹 메시지 방송 시간인 B_2(15)가 되며, $TS_l(G_3)$은 변경되지 않는다. 그러므로 B_2에 방송하는 최종적인 그룹 메시지 $GR(15)$는 그림 4와 같다.

제안하는 UGR-MT 방법은 가능한 한 이동 트랜잭션의 빠른 응답 시간을 제공하기 위하여 서버에서 주기적

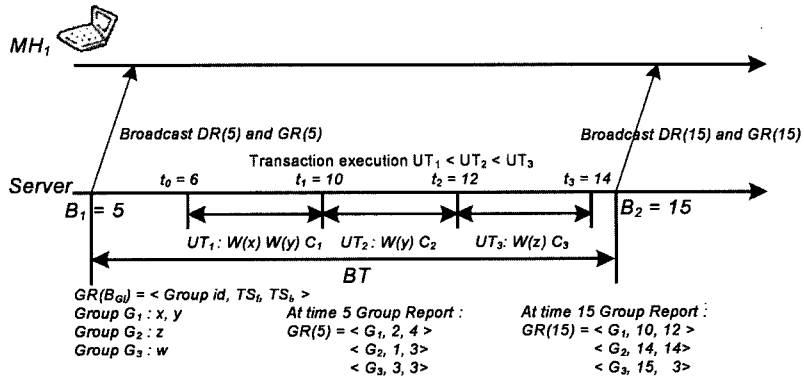

그림 4 그룹 메시지 구성

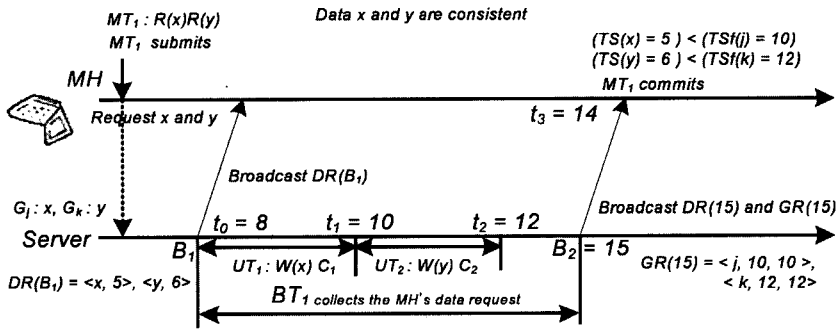

그림 5 이동 트랜잭션의 즉시 완료 결정의 예(*Type.2*)

으로 방송하는 무효화 메시지를 수신하기 이전에 그룹 메시지를 사용한다. 이동 트랜잭션이 접근한 데이타를 포함하는 그룹 리스트를 G_{MT}라고 하는 경우 다음과 같은 세 가지 조건을 만족하는 경우에 이동 트랜잭션은 즉시 완료할 수 있다.

이동 트랜잭션의 즉시 완료 결정 조건

Type.1. *이동 트랜잭션이 접근한 모든 데이타의 타임스탬프가 같은 경우*

Type.2. *이동 트랜잭션이 접근한 모든 데이타의 타임스탬프가 그룹 메시지의 $MIN\{TS_f(j), j \in G_{MT}\}$보다 작은 경우*

Type.3. *이동 트랜잭션이 접근한 모든 데이타의 타임스탬프가 그룹 메시지의 $MAX\{TS_f(j), j \in G_{MT}\}$보다 큰 경우*

*Type.1*의 경우는 이동 트랜잭션이 접근한 모든 데이타가 가장 최근에 수신한 무효화 메시지 이전에 방송되어 최근에 수신한 무효화 메시지에 의하여 일관성이 검증된 데이타인 경우이다. 그러므로 이동 트랜잭션은 수행의 정확성을 보장받으며 즉시 완료할 수 있다. 또한 *Type.1*의 경우는 이동 트랜잭션이 하나의 갱신 트랜잭션이 갱신한 데이타에만 접근하는 경우도 포함되며, 이러한 경우 이동 트랜잭션은 갱신 트랜잭션의 완료 시점을 기준으로 일관성 있는 데이타에 접근한 것으로 간주하여 수행의 정확성을 보장할 수 있다.

그림 5는 이동 트랜잭션의 수행이 *Type.2*의 조건을 만족하는 예를 보여준다. 그림 5에서 이동 트랜잭션 MT_1은 그룹 j의 데이타 x와 그룹 k의 데이타 y에 접근한다. MT_1이 접근한 시점 B_1에 방송된 x와 y의 갱신 타임스탬프는 각각 5와 6으로 즉시 완료 조건 *Type.1*은 만족하지 못하며 MT_1의 완료 결정은 서버로부터 그룹 메시지를 수신할 때까지 지연된다. 그리고 서버에서는 방송 트랜잭션 BT_1이 수행되어 B_1에서 B_2동안의 데이타 요청 메시지를 수집하여 B_2시점의 데이타 방송 스케줄을 구성한다. 동시에 서버에서는 데이타 방송 후 x와 y를 갱신하는 갱신 트랜잭션 UT_1과 UT_2이 수행되어 각각 10과 12인 시점에 완료되었으며 이 갱신은 현재 무효화 메시지 방송 주기 동안에 발생한 첫 갱신이라고 가정한다. 그러면 그룹 j와 k의 $TS_f(j)$와 $TS_f(k)$는 각각 10과 12가 되며 시점 B_2(15)때 그룹 메시지는 데이타 메시지와 함께 이동 호스트에게 방송 된다. 그룹 메시지

를 수신한 이동 호스트는 MT_1이 접근한 모든 데이타의 타임스탬프가 수신한 그룹 메시지의 $TS_f(j)$와 $TS_f(k)$의 최소값인 10보다 작으므로 MT_1은 시점 10을 기준으로 일관성 있는 데이타에 접근한 것을 알 수 있으며 즉시 완료할 수 있다. *Type.3*의 경우는 이동 트랜잭션이 접근한 모든 데이타가 가장 최신의 데이타에 접근한 것을 의미하며 그룹 메시지의 방송 시점을 기준으로 일관성 있는 데이타에 접근한 것으로 간주하여 이동 트랜잭션 수행의 정확성을 보장한다.

제안하는 UGR-MT 방법에서 이동 호스트는 그룹 메시지를 사용하여 이동 트랜잭션이 접근한 데이타가 즉시 완료 조건을 만족하는 경우 즉시 완료하며 그렇지 않는 경우에는 서버에서 주기적으로 방송하는 무효화 메시지를 수신할 때까지 이동 트랜잭션의 완료 결정을 지연한다. 이동 호스트는 무효화 메시지를 사용하여 이동 호스트의 캐쉬 일관성을 유지하며, 서버는 이동 호스트가 주기적으로 방송되는 무효화 메시지를 수신하지 못하는 경우를 대비하여 여러 방송 구간의 무효화 메시지 정보로 구성되는 윈도우 정보를 유지한다. 그리고 이동 호스트는 무효화 메시지 수신 후 완료 결정을 기다리고 있는 이동 트랜잭션 중에서 무효화되는 데이타에 접근한 이동 트랜잭션은 철회한다. 또한 현재 수행되는 (active) 이동 트랜잭션 중에서 무효화된 데이타에 접근한 이동 트랜잭션도 철회한다. 무효화 메시지 수신 후 이동 트랜잭션 수행의 정확성 검증 방법은 알고리즘 2와 같이 기술할 수 있다.

B_i : current invalidation report broadcast time
B_{IL} : last broadcast time mobile host received an invalidation report
S_{MTi} : set of data items accessed by mobile transaction MT_i
L : the invalidation report broadcast period
$TS(x)$: Timestamp of data item x in a mobile host
$TS(IR(x))$: Timestamp of data item x in invalidation report
***Suppose** MH received $IR(B_i)$ and $GR(B_i)$ from the server*
***If** ((B_i - B_{IL}) > L) then*
  ***Request** the window report and **wait**;*
else
  *{ **For** (each data item x in the mobile host's cache)*
    If** ($TS(x) < TS(IR(x))$) **then
      Throw x out of the cache;
    else
      $TS(x) = B_i$;
  ***For** (each mobile transaction t that waited its verification)*
    If** (S_{MTt} contains any dropped data item) **then
      ***Abort** MT_t;*
    else
      ***Commit** MT_t;*
  ***For** (each mobile transaction t that is active)*
    If** (S_{MTt} contains any dropped data item) **then
      ***Abort** MT_t;*
  $B_{IL} = B_i$; }

알고리즘 2. UGR-MT에서 캐쉬 일관성 유지 및 이동 트랜잭션의 완료 결정 알고리즘

그림 2의 이동 트랜잭션 수행의 잘못된 결정의 예서도 제안하는 방법은 방송 시점 B_1에 데이타 메시지와 함께 방송되는 그룹 메시지 정보를 사용하여 이동 트랜잭션을 즉시 완료할 수 있다. 즉 그림 2에서 x와 y가 각각 다른 그룹의 데이타이며 각 그룹의 데이타는 B_1시점까지 갱신이 발생하지 않았다고 가정한다. 기존의 OCC-UTS[2] 방법에서는 x와 y의 타임스탬프가 다르므로 즉시 완료할 수 없으며 데이타 방송 후 발생한 갱신에 의하여 이동 트랜잭션은 철회된다. 그러나 제안한 UGR-MT 방법은 x와 y가 속하는 그룹의 데이타가 갱신되지 않았으므로 그룹의 TS_f(x의 그룹)와 TS_f(y의 그룹)는 방송 시점 B_1이 된다. 그리고 이동 트랜잭션이 접근한 x와 y의 타임스탬프가 B_1보다 작으므로 *Type.2*를 만족함으로써 이동 트랜잭션은 즉시 완료할 수 있다.

그리고 이동 호스트의 단절 기간이 서버에서 유지하는 윈도우 메시지의 구간보다 큰 경우에는 이동 호스트의 캐쉬에 있는 데이타의 일관성을 검증할 수 없으며 이동 호스트는 캐쉬에 있는 데이타를 모두 무효화한다 [1,2]. 그러나 무효화된 데이타 중에는 실제로 갱신 이 발행하지 않아 유효한 데이타가 존재할 수 있으며, 잘못된 데이타 무효화 결정은 데이타 재캐싱으로 인한 비효율적인 대역폭 사용이 빈번해진다는 문제점이 있다. UGR-MT 방법은 이동 호스트의 단절 기간이 윈도우 구간보다 길더라도 서버에서 방송하는 그룹 메시지의 데이타 그룹에 대한 시간 정보를 사용하여 전체 데이타의 무효화를 막을 수 있다. 일반적으로 갱신율이 낮은 데이타일수록 이동 호스트의 단절 기간이 길더라도 유효한 데이타일 가능성이 높다. 만약 이동 호스트의 캐쉬에 그룹 j의 데이타 x가 있고 그룹 j의 타임스탬프 $TS_f(j)$이 가장 최근에 수신한 무효화 메시지 방송 시점 B_{IL}보다 작은 경우 이동 호스트는 단절 기간에 상관없

이 그룹 j의 데이타는 유효한 데이타임을 보장받을 수 있다. 즉 그룹 메시지의 타임스탬프 $TS_l(j)$의 시간 정보에 의하여 그룹 j의 데이타는 B_{IL} 이후 갱신되지 않았음을 알 수 있다. UGR-MT 방법에서 그룹 메시지와 윈도우 메시지를 사용한 캐쉬 일관성 유지 방법은 알고리즘 3과 같이 기술할 수 있다.

B_{IL} : last time mobile host received an invalidation report broadcast time
B_{Wi} : current window report broadcast time
$TS_l(GR(j))$: Last update TS of the group j
***Suppose** MH received the window report and the group report from the server*
***If** ((B_{Wi} - B_{IL}) < window report broadcast period W) then*
Replace an IR(B_i) by WR(B_{Wi}) in Algorithm. 2 and execute Algorithm. 2;
else
For (each group identifier j in group report)
*If ($TS_l(GR(j)) > B_{IL}$) **then***
***Throw** Group j's all data out of the cache;*
else
***For** (each data item x of group j in mobile host's cache)*
$TS(x) = B_{Wi}$;

알고리즘 3. UGR-MT에서의 단절 후 이동 호스트의 캐쉬 일관성 유지 알고리즘

그리고 제안하는 UGR-MT 방법이 이동 트랜잭션의 직렬가능한 수행을 보장할 수 있다는 것은 정리 1과 같다.

정리 1. UGR-MT 방법은 이동 트랜잭션의 직렬가능한 수행을 보장한다.

(증명) 증명을 위하여 본 연구에서 제안한 UGR-MT 방법은 이동 트랜잭션의 직렬가능한 수행을 보장하지 못한다고 가정한다. 이동 트랜잭션의 직렬가능한 수행을 보장하지 못한다는 것은 이동 트랜잭션이 현재 방송 주기에 갱신된 데이타에 접근하여 이동 트랜잭션과 서버에서 수행되는 갱신 트랜잭션 사이에 사이클이 발생한다는 것을 의미한다. 일반적으로 읽기 연산으로만 구성된 이동 트랜잭션의 수행에서 사이클 발생은 이동 트랜잭션이 서로 다른 시점의 데이타에 접근함으로써 발생하며 이동 트랜잭션의 모든 연산이 어느 한 시점을 기준으로 일관성 있는 데이타에 접근하는 경우 이동 트랜잭션의 직렬가능한 수행을 보장할 수 있다. 제안한 UGR-MT 방법에서 서버는 현재 무효화 메시지 방송 주기 동안에 발생한 데이타들의 첫 번째 갱신 시간과 마지막 갱신 시간에 대한 시간 정보인 그룹 메시지를 방송한다. 그리고 이동 호스트는 이동 트랜잭션이 접근한 데이타의 타임스탬프와 그룹 메시지의 시간 정보를 이용하여 이동 트랜잭션 수행의 완료 및 지연 여부를 결정한다. UGR-MT 방법에서 이동 호스트는 이동 트랜잭션 수행 후 즉시 완료 조건을 만족하는 경우 즉시 완료하며 만족하지 않는 경우에는 무효화 메시지를 수신할 때까지 완료 결정을 지연한다. 이동 호스트가 가장 최근에 수신한 무효화 메시지 방송 시점을 B_1, 이동 트랜잭션 수행 후 수신한 그룹 메시지의 수신 시점을 B_3, 이동 트랜잭션이 접근한 데이타가 포함되는 그룹 리스트를 G_{MT}라고 하는 경우 $MIN\{TS_l(j),\ j \in G_{MT}\}$을 B_2, 그리고 서버에서 방송하는 다음 무효화 메시지 방송 시점을 $B_4(B_1 < B_2 < B_3 < B_4)$로 정의하자.

본 연구에서 제안한 *Type.1*의 경우, 이동 트랜잭션이 B_1시점 이후에 제출되더라도 B_1시점 이전에 캐쉬된 데이타에 접근한 경우이며 이동 트랜잭션이 접근한 데이타는 B_1시점에 서버에서 방송한 무효화 메시지로 인하여 일관성 있는 데이타임을 보장받으며 이동 트랜잭션 수행의 정확성을 보장한다. *Type.2*의 경우, 이동 트랜잭션이 B_2시점 이후에 제출되어도 그룹 메시지의 $\{TS_l(j),\ j \in G_{MT}\}$ 정보에 의하여 이동 트랜잭션이 접근한 데이타가 최소한 B_2시점까지는 일관성 있는 데이타임을 보장한다. *Type.3*의 경우는 이동 트랜잭션의 제출 시점에 상관없이 이동 트랜잭션은 그룹 메시지 수신 시점 B_3을 기준으로 일관성 있는 데이타에 접근하였으며 이동 트랜잭션 수행의 정확성을 보장한다.

제안한 UGR-MT 방법은 이동 트랜잭션의 수행이 세 가지 완료 조건을 만족하지 못하는 경우 무효화 메시지를 사용하여 이동 호스트의 캐쉬에 있는 데이타 유효성을 검사한 후 무효화되는 데이타에 접근하는 이동 트랜잭션은 철회시키고, 그렇지 않은 이동 트랜잭션은 완료한다. 그러므로 무효화 메시지 수신 후 완료되는 이동 트랜잭션은 무효화 메시지 방송 시점인 B_4시점에 일관성 있는 데이타에 접근한 것으로 간주하여 정확성을 보장받을 수 있다. 제안한 UGR-MT 방법은 실제적인 이동 트랜잭션의 제출 시점이 아닌 접근한 데이타의 갱신 시점을 기준으로 일관성 있는 데이타에 접근하는 경우 이동 트랜잭션을 완료시킴으로써 이동 트랜잭션의 직렬가능한 수행을 보장할 수 있다. □

5. 성능 평가

본 장에서는 제안한 UGR-MT 방법의 성능 평가를 위한 매개변수를 기술하고 이동 트랜잭션의 응답 시간 및 완료율에 따른 성능을 분석한다. 그리고 제안한

UGR-MT 방법의 성능을 분석하기 위하여 참고문헌 [8]의 UFO 방법과, 참고문헌 [7]의 OCC-UTS^2 방법을 비교 대상으로 한다. UFO 방법과 OCC-UTS^2 방법은 이동 호스트가 캐쉬를 사용하여 이동 트랜잭션을 수행하며 무효화 메시지를 사용하여 이동 트랜잭션의 직렬 가능한 수행을 보장하기 때문에 제안한 UGR-MT 방법의 비교 대상으로 적합하다.

참고문헌 [1,2,7]에서의 분석 방법에 기초하여 이동 트랜잭션이 특정 데이타에 λ의 비율로 접근하며 λ가 지수분포(exponential distribution)의 형태를 따르는 경우 무효화 메시지 방송 주기 L동안에 질의가 있을 확률 P_Q는 다음과 같다.

$$q0 = Prob[active\ and\ No\ Query\ during\ L] = (1-Prob[Disconnection]) \times e^{-\lambda L} \quad (1)$$

$$p_0 = Prob[No\ query\ during\ L] = Prob[Disconnection] + q_0 \quad (2)$$

$$P_Q = 1-P_0 = (1-Prob[Disconnection])(1- e^{-\lambda L}) \quad (3)$$

본 성능평가에서는 비교하는 모든 방법에 대하여 이동 호스트는 단절되지 않는다고 가정하며, 식 (3)에도 이를 적용하여 방송 주기 L동안에 질의가 있을 확률 P_Q는 $(1- e^{-\lambda L})$로 다시 표현할 수 있다. 그리고 서버에서의 데이타 갱신율이 u이고 지수분포의 형태를 따르는 경우 무효화 메시지 방송 주기 L동안에 데이타가 갱신될 확률 P_U는 다음과 같다.

$$u0 = Prob[No\ update\ during\ L] = e^{-uL} \quad (4)$$

$$P_U = 1-u_0 = 1-e^{-uL} \quad (5)$$

또한 참고문헌 [1,2]에서와 같이 식 (1), (2), (4)의 값을 이용하여 무효화 메시지 방송 주기 L동안에 평균 캐쉬 적중률 h는 다음과 같이 계산할 수 있다.

$$h = \frac{(1-p_0)u_0}{1-q_0u_0} \quad (6)$$

5.1 이동 트랜잭션의 응답 시간 분석

이동 트랜잭션의 응답 시간은 데이타 캐쉉 시간, 이동 트랜잭션의 수행 시간, 그리고 이동 트랜잭션의 완료 결정 시간으로 구성된다. 이동 호스트는 이동 트랜잭션이 캐쉬에 없는 데이타를 요구하는 경우 데이타 요청 메시지를 보내며 서버는 방송 트랜잭션을 수행하여 이동 호스트가 요청한 데이타를 전송한다. 따라서 n개의 데이타에 접근하는 이동 트랜잭션을 수행하기 위하여 필요한 평균 캐쉉 시간 $MT_{caching}$은 다음과 같다.

$$MT_{caching} = n \times (1-h) \times (transfer\ time) \quad (7)$$

UGR-MT 방법, UFO 방법, 그리고 OCC-UTS^2 방법에서 이동 트랜잭션의 처리 시간은 데이타 캐쉉 및 완료 결정 시간에 비하여 매우 작다고 볼 수 있다. 그러므로 응답 시간의 측정에 있어서 이동 트랜잭션의 처리 시간은 생략한다. UFO 방법에서 이동 트랜잭션의 완료 결정은 수행 후 서버에서 방송하는 무효화 메시지 수신 시점으로 지연된다. 무효화 메시지 방송 주기 L동안에 이동 트랜잭션의 평균 지연 시간을 $L/2$이라고 하는 경우 UFO 방법에서 이동 트랜잭션의 평균 응답 시간 RT_{UFO}는 다음과 같다.

$$RT_{UFO} = MT_{caching} + \frac{L}{2} \quad (8)$$

OCC-UTS^2 방법은 이동 트랜잭션이 접근한 데이타의 타임스탬프가 모두 같거나 가장 최근에 수신한 무효화 메시지의 타임스탬프보다 작은 경우에는 즉시 완료하며, 그렇지 않은 경우에는 서버에서 주기적으로 방송하는 무효화 메시지 수신 시점까지 완료 결정이 지연된다. 그러므로 이동 트랜잭션의 수행이 즉시 완료 조건을 만족할 확률 P_{OCC}와 이동 트랜잭션의 평균 응답 시간 RT_{OCC}는 다음과 같다.

$$P_{OCC} = \left(\frac{(1-h)}{e^{uL}}\right)^n \quad (9)$$

$$RT_{OCC} = MT_{caching} + \frac{L}{2} \times n \times (1-P_{OCC}) \quad (10)$$

UGR-MT 방법에서 이동 트랜잭션은 제안한 완료 조건 중 하나를 만족하는 경우 즉시 완료할 수 있으며, 그렇지 않은 경우에는 무효화 메시지 수신 시점까지 완료 결정이 지연된다. UGR-MT 방법에서 즉시 완료 조건 *Type.1*은 OCC-UTS^2 방법에서 제안한 즉시 완료 조건과 같다. 그리고 즉시 완료 조건 *Type.2*는 이동 트랜잭션 접근한 데이타가 현재 무효화 메시지 방송 주기 동안에 방송되었지만 그 데이타가 방송 주기 동안에 갱신되지 않는 경우이며, 즉시 완료 조건 *Type.3*은 데이타 방송 후 갱신이 발생하지 않는 경우이다. 그러므로 무효화 메시지 방송 주기 동안에 평균적으로 갱신이 발생하지 않는 구간을 E라고 할 때 UGR-MT 방법에서 이동 트랜잭션의 수행이 즉시 완료 조건을 만족할 확률과 이동 트랜잭션의 평균 응답 시간 RT_{UGR}은 다음과 같다.

$$P_{Type.1} = \left(\frac{(1-h)}{e^{uL}}\right)^n, \quad P_{Type.2} = \left(\frac{(1-h)}{e^{uE}}\right)^n \quad (11)$$

$$P_{Type.3} = \left((1-h) \times \left(1-\frac{1}{e^{u(L-E)}}\right) \times \left(1-\frac{1}{e^{\lambda E}}\right)\right)^n \quad (12)$$

$$RT_{UGR} = MT_{caching} + \frac{L}{2} \times (1-(P_{Type1} + P_{Type2} + P_{Type3})) \quad (13)$$

분석 결과 UGR-MT 방법은 이동 트랜잭션의 응답 시간 면에서 기존의 UFO 방법과 OCC-UTS^2 방법에 비하여 최소 $\frac{L}{2} \times (P_{CD2} + P_{CD3})$만큼의 성능향상이 있음을 알 수 있다.

5.2 이동 트랜잭션의 완료율

UFO 방법은 이동 트랜잭션 수행의 정확성을 보장하

기 위하여 이동 트랜잭션의 완료 결정을 서버로부터 무효화 메시지를 수신할 때까지 지연한다. 그리고 이동 트랜잭션이 무효화되는 데이타에 접근한 경우에 이동 호스트는 철회 결정을 내린다. 그러므로 UFO 방법에서 이동 트랜잭션의 완료율 CR_{UFO}는 다음과 같다.

$$CR_{UFO} = \frac{1}{e^{uL}} \times (1 - \frac{1}{e^{\lambda L}}) \quad (14)$$

OCC-UTS[2] 방법은 무효화 메시지를 수신하기 이전에 이동 트랜잭션이 사용한 데이타의 타임스탬프가 모두 같거나 가장 최근에 수신한 무효화 메시지의 방송 시점보다 작은 경우에 즉시 완료한다. 그리고 즉시 완료 조건을 만족하지 못하는 경우에는 무효화 메시지 수신 후 무효화되는 데이타에 접근한 이동 트랜잭션은 철회 결정을 내린다. 그러므로 식 (9)를 이용한 OCC-UTS[2] 방법에서 이동 트랜잭션의 완료율 CR_{OCC}는 다음과 같다.

$$CR_{OCC} = P_{OCC} + (1 - P_{OCC}) \times \frac{1}{e^{uL}} \times (1 - \frac{1}{e^{\lambda L}}) \quad (15)$$

UGR-MT 방법은 OCC-UTS[2] 방법과 같이 이동 트랜잭션의 수행이 제안한 즉시 완료 조건을 만족하는 경우 완료한다. 그리고 조건을 만족하지 못하는 경우에는 무효화 메시지 수신 후 무효화되는 데이타에 접근한 모든 이동 트랜잭션은 철회 결정을 내린다. 그러므로 식 (11), (12)의 즉시 완료 조건을 만족할 확률을 P_{UGR}로 정의하면 UGR-MT 방법에서 이동 트랜잭션의 완료율 CR_{UGR}은 다음과 같다.

$$CR_{OCC} = P_{UGR} + (1 - P_{UGR}) \times \frac{1}{e^{uL}} \times (1 - \frac{1}{e^{\lambda L}}) \quad (16)$$

그리고 성능 분석을 위하여 해석적 모델에 적용할 매개변수는 표 1과 같다. 각 매개변수의 값들은 참고문헌 [7]에서의 값을 사용한다.

표 1 성능 분석을 위한 매개변수

변수명	설 명	변수값
L	무효화 메시지 방송 주기	10s
n	이동 트랜잭션이 접근하는 평균 데이타 수	동적
λ	특정 데이타에 대한 평균 접근율(hot spot)	0.01/s
u	서버에서의 데이타 평균 갱신율	동적
E	서버에서 갱신이 일어나지 않는 평균 주기(시간)	0.5s

표 1의 매개변수를 식 (8)-(13)에 적용하면 이동 트랜잭션의 응답 시간을 구할 수 있다. 식 (7)에 표 1의 값을 적용하고 데이타 갱신율 u를 참고문헌 [6]에서 적용한 0.005로 적용한 경우 캐쉬 적중률 h는 약 0.68이 되며, 하나의 데이타를 전송하는데 걸리는 시간은 0.16이 된다. 그리고 참고문헌 [6]에서의 해석적 모델과 같이 이동 트랜잭션이 접근하는 데이타 개수를 기준으로 한 각 방법의 응답 시간은 그림 6과 같다.

그리고 표 1의 매개변수를 식 (14)-(16)에 적용한 데이타 갱신율에 따른 이동 트랜잭션의 완료율은 그림 7과 같다.

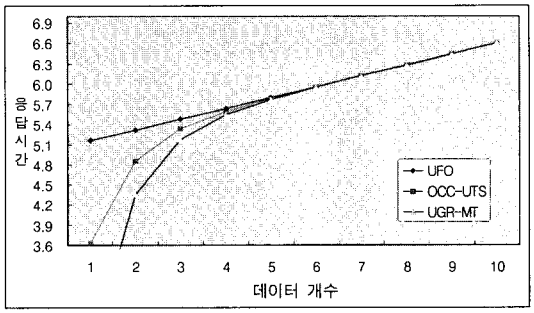

그림 6 이동 트랜잭션의 응답 시간

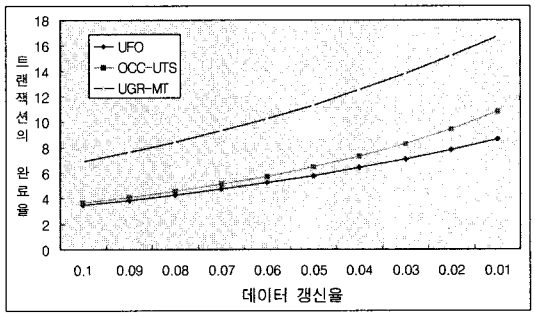

그림 7 이동 트랜잭션의 완료율

그림 6과 7에서 알 수 있듯이 본 연구에서 제안한 UGR-MT 방법은 기존의 방법들에 비하여 그룹 정보가 추가됨으로써 방송되는 메시지의 크기가 커지지만 이동 트랜잭션이 접근하는 데이타 개수가 적거나 데이타 갱신율이 작을수록 이동 트랜잭션의 응답 시간이 빨라지고 완료율이 높아짐을 알 수 있다.

6. 결론 및 향후 연구 방향

이동 컴퓨팅 환경에서 이동 호스트는 가능한 한 제한된 대역폭의 사용을 줄이고 이동 트랜잭션의 빠른 응답 시간을 제공하기 위하여 캐싱 기법을 사용한다. 그리고 서버에서 데이타가 갱신되는 경우 갱신된 데이타와 이동 호스트에 캐싱된 데이타 사이의 일관성을 유지하기 위하여 주기적으로 무효화 메시지를 방송한다. 그러나 무효화 메시지 방송을 사용한 이동 호스트의 캐쉬 일관성 유지 방법은 이동 트랜잭션의 직렬가능한 수행을 보장하기 위하여 불필요한 지연이 발생한다는 문제점이 있다.

본 연구에서는 서버에서 주기적인 무효화 메시지를

방송하여 이동 호스트의 캐쉬 일관성을 유지하는 경우 데이타 그룹 정보를 사용하여 이동 트랜잭션의 직렬가능한 수행을 보장하는 UGR-MT 방법을 제안하였다. 제안한 UGR-MT 방법은 데이타 그룹 개념을 사용하여 가능한 한 이동 트랜잭션의 완료 결정 여부를 무효화 메시지 수신 이전에 결정할 수 있도록 함으로써 이동 트랜잭션의 응답 시간을 향상시킬 수 있다. 또한 이동 호스트의 단절이 오래 지속되는 경우에 발생하는 이동 호스트의 전체 캐쉬 내용의 버림도 방지할 수 있다. 그리고 해석적 모델을 이용한 성능 평가를 통하여 제안한 UGR-MT 방법이 기존의 방법에 비하여 빠른 이동 트랜잭션의 응답 시간과 완료율을 보임을 알 수 있었다. 앞으로의 연구 방향은 데이타 그룹을 효율적으로 구성함으로써 이동 트랜잭션의 응답 시간과 대역폭의 사용을 줄일 수 있는 방법에 대하여 연구하고자 한다.

참 고 문 헌

[1] D. Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," Proceeding of the ACM SIGMOD Intl. Conference on Management of Data(SIGMOD 94), Page 1-12, 1994.

[2] D. Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments(Extended Version)," VLDB Journal Vol.4, 1995.

[3] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing: Solutions and Challenges in Data Management," Communication of ACM, Vol.37, No.10, 1994.

[4] R. Alonso and H. Korth, "Database System Issues in Nomadic Computing," In Proceedings of the ACM SIGMOD Conference on Management of Data, Page 388-392, 1993.

[5] T. Imielinski and B. R. Badrinath, "Data Management for Mobile Computing", ACM SIGMOD RECORD, Vol.22, No.1, Page 34-39, Mar, 1993.

[6] J. Jing, A. Elmagarmid, A. Helal and R. Alonso, "Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," Mobile Networks and Applications Vol.2, No.2, Page 115-127, 1997.

[7] SangKeun Lee, "Caching and Concurrency Control in a Wireless Mobile Computing Environments," IEICE Trans. INF. & SYST., Vol.E85-D, No.8 Aug, 2002.

[8] Kam-Yiu Lam, Mei-Wai Au and Edward Chan, "Broadcasting Consistent Data to Read-Only Transactions from Mobile Clients," The Computer Journal, Vol.45 No.2, Page 129-146, 2002.

[9] S. Acharya, M. Franklin and S. Zdonik, "Balancing Push and Pull for Data Broadcast," In Proceedings of the ACM SIGMOD Conference on Management of Data, Page 183-194, May, 1997.

[10] Chi-Wai Lin, H. Hu, and Dik-Lun Lee, "Adaptive Data Delivery in Wireless Communication Environment," Wireless Networks, Vol.10, Page 103-120, March, 2004.

[11] K. L. Tan and J. Cai, "Broadcast-Based Group Invalidation : An Energy-Efficient Cache Invalidation Strategy," Information Sciences, Vol.100, Page 229-253, Aug, 1997.

김 대 인

1996년 동신대학교 컴퓨터학과 졸업(이학사). 1998년 전남대학교 대학원 전산통계학과 졸업(이학석사). 2000년 전남대학교 대학원 전산학과 박사수료. 2004년~현재 전남대학교 전자컴퓨터정보통신공학부 강사. 관심분야는 분산시스템, 이동 컴퓨팅, 전자상거래, XML

황 부 현

1978년 숭실대학교 전산학과(학사). 1980년 한국과학기술원 전산학과(공학석사) 1994년 한국과학기술원 전산학과(공학박사). 1980년~현재 전남대학교 전자컴퓨터정보통신공학부 교수. 관심분야는 분산시스템, 이동 데이타베이스, 데이타마이닝, 멀티미디어 데이타베이스, 객체지향 시스템, XML