

이동 컴퓨팅 환경에서 철회 단계를 감소시키는 갱신 빈도 캐쉬 일관성 기법

이찬섭^o 김동혁 백주현* 김영재 최의인
한남대학교 컴퓨터공학과
한국전산원*

{cslee^o, dhkim, yjkim, ehchoi}@dmlab.hannam.ac.kr
Bbaek@nca.or.kr*

Update Frequency Cache Consistency for Reducing Undo in Mobile Computing

Chan-Seob Lee^o Dong-Hyuk Kim Joo-Hyun Baek* Young-Jae Kim Eui-In Choi
Dept. of Computer Eng, Hannam University
NCA*

요 약

무선 네트워킹 기술과 통신 기기의 발달로 이동 컴퓨팅 환경이 보편화됨에 따라 제한된 대역폭의 절감과 빠른 응답시간, 그리고 확장성을 위해 이동 호스트는 지역 캐쉬를 이용한다. 이때, 이동 호스트와 지구국간에는 캐쉬의 일관성 유지가 필연적이며 이에 따라 많은 기법들이 제안되고 있다. 기존의 일관성 기법 중 탐지기반의 기법들은 간단하고 구현이 쉬운 반면 전송되는 메시지 횟수가 많고, 회피기반의 기법들은 빠른 처리가 가능하여 고정 네트워크에서 일반적으로 사용되고 있는 기법이다. 본 논문에서 제안한 기법은 페이지 요청 또는 완료시 갱신 연산이 일어난 객체에 대해 갱신 빈도를 고려함으로써 기존 기법에 비해 교환되는 메시지 횟수를 줄여 전송비용을 개선하였고, 또한 응답 시점을 옮겨 빠른 응답이 가능하게 하여 이동 컴퓨팅 환경에 적합하도록 하였다. 따라서, 본 논문에서 제안한 방법은 갱신 연산이 이루어지는 경우 비동기적으로 갱신 빈도에 따른 갱신의도 선언 또는 갱신을 선택적으로 수행하여 응답이 빠르고 철회 단계가 감소하는 장점을 갖는다.

1. 서 론

대량의 데이터를 무선으로 고속 전송할 수 있는 네트워킹 기술의 발전과 통신기기의 성능향상으로 사용자는 이동중에도 무선 통신망을 이용하여 데이터베이스에 쉽게 접근할 수 있게 되었다. 이러한 이동 컴퓨팅 환경에서 이동 호스트들은 빠른 응답과 확장성을 위해 이동 호스트내의 지역 캐쉬에 데이터의 사본을 유지하며 이때, 지구국(Mobile Support Station)과의 캐쉬 일관성 유지하기 위해 많은 기법이 연구되고 있다[1].

이동 컴퓨팅 환경은 고정 컴퓨팅 환경에 비해 좁은 대역폭과 잦은 단절이 발생된다. 즉, 빈번한 데이터 요청으로 인해 통신 채널에 대한 점유 경쟁이 발생하고, 셀을 벗어나거나 제한된 배터리의 수명으로 의도적 혹은 비의도적인 통신 단절이 일어나므로 상태 유지가 어렵다. 따라서, 기존의 일관성 유지 기법들에 대한 적절한 재구성이 필요하다[2].

이 논문에서는 좁은 대역폭내에서 지구국에 빈번히 일어나는 데이터 요청으로 대역폭의 점유 경쟁을 줄일 수 있는 메시지 전송 횟수를 줄이면서 빠른 응답을 받을 수 있는 캐쉬 일관성 유지 기법을 제안하였다. 즉, 갱신 빈도를 이용한 비동기적 갱신-의도 선언으로 전송횟수를 감소 시켰고, 다른 이동 호스트들로부터 CB(callback) 응답 전에 갱신-의도를 선언한 이동 호스트에 응답함으로써 신속한 응답이 가능하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 이야기 하고, 3장에서는 본 논문에서 제안한 갱신 빈도를 이용한 비동기적 갱신-의도 선언 및 응답 시점을 옮긴 효율적인 일관성 기법과 트랜잭션 처리를 위한 시나리오를 기술하며, 4장에서는 시나리오를 바탕으로 제안한 기법과 기존의 다른 일관성 기법들과 비교 분석한 후, 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

이동 컴퓨팅 환경은 고정 컴퓨팅 환경과 달리 유선과 무선 영역으로 구분된다. 지구국과 고정 호스트 사이는 유선으로, 지구국과 이동 호스트 사이는 무선으로 연결되어 있다. 이동 호스트와 지구국 사이는 NWL-MH와 같은 탐지기반의 기법이 사용하고, 고정 호스트들 사이는 O2PL 같은 회피-기반기법이 사용된다[1].

일반적인 환경에서 캐쉬 일관성 유지를 위한 탐지기반(detection-based)과 회피기반(avoidance-based)의 기법은 트랜잭션이 캐쉬한 데이터에 접근할 때 데이터의 정확성에 대한 확인 여부에 따라 분류한다. 탐지기반의 기법으로는 C2PL [3,7], NWL[7], AOCC[6]등이 있으며 회피기반의 기법으로는 CB[7], ACBL[5,6], AACCC[4], O2PL[3]등이 있다.

또한, 이동 호스트의 접속 단절로 인해 셀 내의 존재 여부와 상태를 파악하기 어려우므로 회피-기반의 규약처

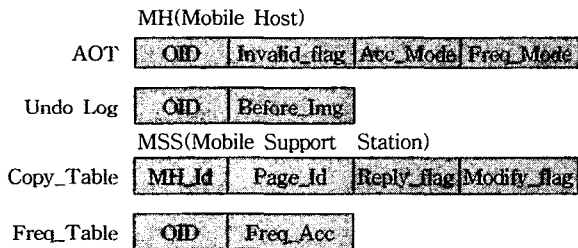
럼 ROWA (Read One Write All) 규칙을 적용하는 것은 적합하지 않아 일반적으로 데이터베이스의 갱신을 정기적으로 방송(broadcast)하는 TS, AT, SIG와 같은 방법이 사용된다[8].

3. 제안한 캐쉬 일관성 유지 기법

본 논문의 연구 환경은 하나의 지구국(Mobile Support Station)과 다수의 이동 호스트(Mobile Host)들이 고속의 무선 네트워크로 연결된 이동 컴퓨팅 환경으로 데이터는 데이터 전송(Data Shipping) 형태를 가진다.

3.1 데이터 구조

본 논문에서는 제안한 일관성 유지 기법을 위해 MH의 데이터 구조를 아래와 같이 제안하였다. [그림 1]과 같이 MH는 자신의 메모리에 캐쉬된 페이지에 속해있는 객체에 대한 정보(AOT)와 철회 시 이전 데이터로 되돌리기 위한 정보(Undo log)를 유지하고 MSS는 각 MH들에게 복사해준 페이지에 어떤 객체들이 속해있는지에 대한 정보(Copy_Table) 테이블과 갱신 빈도(Freq_Table) 테이블을 유지한다.



[그림 1] 데이터 구조

3.2 MH/MSS 처리 절차

아래의 알고리즘은 갱신 연산이 발생한 경우에 갱신 빈도와 빠른 응답을 위한 처리 부분만을 기술하였다.

- MH에서의 처리 알고리즘

```
while( 트랜잭션 연산 끝이 아닌 동안 ) {
    switch( 연산 ) {
        case "read" :
            ;
        case "write" :
            AOT의 Acc_Mode = 1 변경 ;
            Undo log에 OID, Before_Img를 추가 ;
    }
    if(Freq_Mode = 0) then
        오브젝트 갱신 ;
    else
        {MSS에게 갱신의도를 보냄 ;
        오브젝트 갱신 ;}
        // 트랜잭션 종료 처리
        종료 요구 메시지와 AOT와 갱신한 페이지를
        MSS에게 보냄 ;
        MSS의 응답을 기다림 ;
        break;
}
```

- MSS에서의 처리 알고리즘

```
MH_MSG = MH로부터 받은 메시지 ;
If Freq_Mode = 0 then MH_MSG = "W_intention"
switch (MH_MSG) { // 갱신의도 처리
    case "W_intention" :
        if (OID가 갱신 목적이
        일러있다) then
            MH에게 트랜잭션 철회 응답을 함
        else
            OID에 갱신 목적 설정 ;
            while ( Copy_Table의 끝이 아닌 동안 ) {
                if (Copy_Table의 OID 포함된 페이지 존재) then {
                    Copy_Table의 Reply_flag = 1로 설정 ;
                    Freq_Acc = Freq_Acc + 1 ;
                    Freq_Mode = MSS에서 정해놓은 초기 ;
                    Freq_Mode = 1로 설정 ;
                }
                if (Freq_Mode = 0) then
                    요청 MH에게
                    Freq_Mode 갱신결과를 보냄 ; // 캐쉬된 다른
                    페이지에 대한 메시지 보냄 ;
                }
                If Freq_Mode = 0 then
                    MH에게 "commit"
                    else break;
            }
            Freq_Mode = 0로 설정 ;
            종료 요구 메시지 보냄 ;
            종료 응답을 받지 않고 Commit 요청한
            페이지에 대한 메시지 보냄 ;
        }
}
```

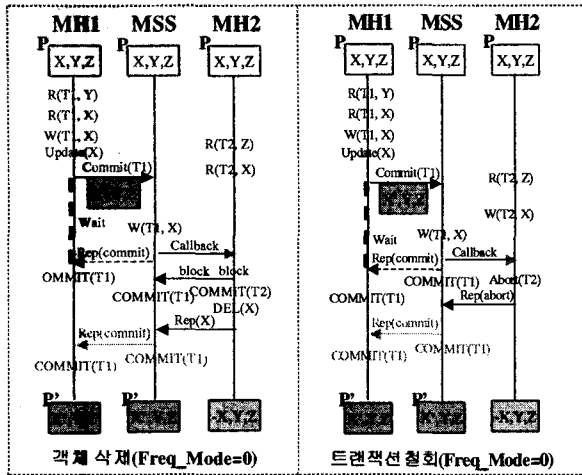
MH에서는 Freq_Mode에 따라 갱신-의도 선언 또는 갱신이 선택적으로 이루어지며, MSS에서는 Freq_Acc의 처리와 갱신 통보 응답 없이 Commit을 승인한다.

3.3 일관성 유지를 위한 시나리오

MH 1과 MH 2에서 다음과 같은 연산 순서로 트랜잭션(T1, T2)이 수행된다고 가정하며 제안한 기법은 갱신이 드물게 일어나는 경우(Freq_Mode = 0)에만 비교하였다.

- T1 : read(Y) → read(X) → write(X)
- T2 : read(Z) → read(X)

트랜잭션 수행 과정에서 발생할 수 있는 시나리오는 아래 [그림 2]와 같이 객체 삭제와 트랜잭션 철회의 두 가지 경우이다. 페이지 삭제의 경우는 객체 삭제와 유사하다. 시나리오 1은 각 MH에 캐쉬된 페이지 중 MH1에서 갱신하고자 하는 객체를 다른 MH2에서 일부 객체에 대한 판독 연산만 수행한 경우로 해당 객체를 삭제한다. 시나리오 2은 각 MH에 캐쉬된 페이지 중 MH1에서 갱신하고자 하는 객체를 다른 MH2에서 갱신-의도를 선언하거나 갱신 연산을 수행하고 있는 경우로 MH1이 MH2보다 Commit 선언을 먼저 했으므로 MH2의 트랜잭션 T2의 갱신은 철회한다.



[그림 2] 시나리오 1, 2

4. 비교 분석

아래 <표 1>은 기존의 AOCC, AACC 기법을 본 연구 환경에 맞게 변환한 뒤 본 논문에서 제안한 캐쉬 일관성 기법을 비교한 결과이다. AOCC 기법은 AACC 기법 보다 낮은 철회율과 높은 성능을 가지며 충돌이 적은 상황에서 좋은 성능을 보이고 있다. 갱신 연산을 수행하기 전에 먼저 갱신의도를 선언하기 때문에 충돌을 일찍 발견하여 철회 동작을 줄인다. 또한 판독 록에 대해 Private-Read or Shared-Read로 페이지를 구분함으로써 P-Read시에는 갱신 연산을 그냥 수행하는 방법으로 성능을 높였다. AOCC와 제안한 기법(Freq_Mode = 0)은 갱신 연산을 먼저 수행한 후 Commit 단계에서 정확성 검사를 수행하므로 높은 성능을 가진다. 대신 Commit 단계에서 정확성 검사를 수행하므로 충돌을 늦게 발견할 수 있어 철회 단계가 길어지는 단점이 있을 수 있다. 제안한 기법은 AOCC 기법과 달리 무조건 갱신 연산을 먼저 수행하지 않고 갱신 빈도에 따라 수행하므로 충돌을 훨씬 더 줄일 수 있다. 또한 Commit 시점도 앞당김으로써 빠른 응답이 가능하도록 하였다.

<표 1> 기존 캐쉬 일관성 기법과의 비교

		AOCC	AACC	제안한 기법
기반		탐지기반	회피기반	탐지/회피 기법
록 획득방법		지연	비동기	지연/비동기적
응답속도		보통	보통	빠름
메시지 수		적음	적음	적음
록 단위		객체	페이지, 객체	페이지, 객체
데이터 구조	MH(C)	ROS, MOS, Undo log		AOT, Undo log
	MSS(S)	MOS, Dir(P) Invalid(C),		Copy_Table Freq_Table

5. 결론

본 논문에서는 이동 컴퓨팅 환경지원을 위한 갱신 빈도에 따른 캐쉬 일관성 유지 기법을 제안했다. 본 논문의 특징은 데이터의 갱신에 대해 선택적으로 갱신 의도 선언 시나 갱신 후에 MSS에게 갱신 의도를 선언할 수 있도록 하여 충돌이 자주 일어나거나 자주 일어나지 않는 일반적인 경우에 모두 효과적으로 적용될 수 있다. Freq_Mode = 0 인 경우 갱신을 먼저 한 후 완료 시점에서 갱신 의도를 선언하므로 MSS의 응답을 기다리는 대기시간을 줄일 수 있으며, Freq_Mode = 1 인 경우 갱신 의도를 먼저 하도록 하여 철회율을 줄일 수 있다. 또한 Callback에 대한 응답 전에 Commit을 승인함으로써 빠른 결과를 얻을 수 있다. 따라서, 트랜잭션의 갱신 정보를 종료(commit) 시점까지 지연시키는 기법(O2PL, AOCC)보다 트랜잭션 철회를 줄이고 있다.

향후 연구 과제로는 알고리즘에 대한 CPU 사용시간, MH와 MSS간 메시지 전송 시간, 디스크 I/O시간, 네트워크 통신 비용 등을 매개 변수로 하여 정확한 성능평가 필요하다.

참고문헌

- [1] Jin Jing et al., "Distributed Lock Management for Mobile Transaction," Proc. IEEE Distributed Computing System, 1995.
- [2] Michael J. Carey et al., "Conflict Detection Tradeoffs for Replicated Data," ACM Transactions on Database Systems, Vol. 16, No. 4, Dec. 1991.
- [3] M. Carey, M. Franklin, M. Livny, E. Shekita, "Data Caching Tradeoffs in Client-Server DBMS Architectures," Proc., ACM SIGMOD, pp.357-366, 1991.
- [4] M. Tamer., and Kaladhar., "An Asynchronous-Based Cache Consistency Algorithm for Client Caching DBMSs," Proc., VLDB, pp.440-451, 1998.
- [5] M. Zaharioudakis, M. Carey, M. Franklin, "Adaptive, Fine-Grained Sharing in a Client-Server OODBMS: A Callback-Based Approach," to appeared ACM TODS.
- [6] R. Gruber, "Optimism VS. Locking: A Study of Concurrency Control for Client-Server Object-Oriented Databases," PhD thesis, MIT, 1997.
- [7] Y. Wang and L. Rowe, "Cache Consistency and Concurrency Control in a Client/Server DBMS Architecture," Proc., ACM SIGMOD, pp.367-376, 1991.
- [8] Daniel Barbara, Tomasz Imielinski, "Sleepers and Workaholics : Caching Strategies in Mobile Environments," Proc. ACM SIGMOD, pp1-12, 1994.