

비동기적 갱신 선언을 이용한 회피-기반 캐쉬 일관성 유지 기법

장창복, 조성훈, 강우석, 김동혁, 이찬섭, 박용문*, 최의인
한남대학교 컴퓨터공학과
*한국전자통신연구원
e-mail:chbjang@dblab.hannam.ac.kr

Avoidance-Based Cache Consistency Technique Using an Asynchronous Write Intension Declaration

Chang-Bok Jang, Sung-Hoon Cho, Woo-Suck Kang,
Dong-Hyuk Kim, Chan-Seob Lee, Yong-Moon Park*, Eui-In Choi
Dept of Computer Engineering, Han-nam University
*ETRI

요약

클라이언트/서버 데이터베이스 시스템 환경이 대중화됨에 따라 클라이언트 캐쉬 데이터의 일관성을 유지하기 위한 기법들이 많이 제안되고 있다. 기존의 회피-기반 기법들은 갱신의도(write intention) 선언을 동기적(synchronous)으로 수행하는 CB(Callback) 기법과 지연(deferred)하는 O2PL(Optimistic 2-Phase Locking) 기법을 기반으로 연구가 이루어졌다. 본 논문에서는 회피-기반(avoidance-based)에서 서버에게 갱신의도 선언을 비동기적으로 수행하는 캐쉬 일관성 유지 기법을 제안한다. 본 논문에서 제안한 기법은 갱신의도 선언을 비동기적으로 수행함으로써 서버의 응답을 기다리지 않고 트랜잭션 처리를 수행함으로써 좋은 성능을 보이고, 트랜잭션 철회(abort)율이 낮다는 장점을 갖는다.

1. 서론

워크스테이션의 성능향상과 네트워크의 속도가 빨라짐으로써 클라이언트/서버 시스템 환경이 보편화되었다. 데이터베이스 시스템도 기존의 중앙 집중적인 환경에서의 서비스보다는 여러 컴퓨터들이 LAN과 같이 고속으로 데이터를 전송할 수 있는 네트워크로 연결된 클라이언트/서버 환경에서 데이터 서버로서의 역할이 중요하게 되었다.

무기 및 방위 제어 시스템, 로봇 제어 시스템, 항공기에 대한 항로 제어 시스템, 공장 자동화 제어 시스템 등과 같은 많은 분야에서 컴퓨터 시스템이 이용되고 있다. 이들은 클라이언트/서버간에 많은 메시지가 발생한다. 그래서 클라이언트는 통신비용의 절감과 서버에 대한 의존도를 낮추기 위하여 자신의 메모리에 데이터의 사본을 캐

쉬한다. 이에 따라 캐쉬 데이터의 일관성을 유지하기 위한 연구가 많이 이루어지고 있다[2, 3, 4, 5, 6, 7, 8].

본 논문에서 제안한 기법은 트랜잭션이 갱신연산을 수행할 때 서버에게 록을 요청한 후, 서버의 응답을 기다리지 않고 계속 트랜잭션 처리를 수행하는 비동기적인 기법을 이용한다. 클라이언트로부터 록 요청을 받은 서버는 갱신되는 페이지를 캐쉬한 다른 클라이언트들에게 삭제 메시지를 보내고, 이 삭제 메시지에 대한 응답을 받는다. 본 논문에서 제안한 캐쉬 일관성 기법은 서버의 응답을 기다리는 시간을 줄이고 트랜잭션 철회율을 감소시키는 특징을 갖는다.

본 논문의 구성은 2장에서 기존의 캐쉬 일관성 기법들과 최근에 제안된 기법들에 대해 기술하고, 3장에서는 본 논문의 연구 환경과 클라이언트/서버에서 제안한 일관성 유지 알고리즘들의 수행과정을 기술하였다. 그리고 4장에

이 논문은 2001년도 한남대학교 학술 연구 조성비 지원에 의하여 지원되었음.

서는 다른 일관성 기법들과 비교 분석하였다

2. 관련 연구

캐쉬 일관성 유지 기법은 트랜잭션이 클라이언트에 캐쉬된 데이터에 접근 시 그 데이터의 정확성에 대한 확인 여부에 따라 탐지-기반(detection-based) 기법과 회피-기반(avoidance-based) 기법으로 구분된다[3, 4]. 탐지-기반에 속하는 기법으로는 C2PL(Cache 2-Phase Locking)[3, 7], NWLN(No Wait Locking w/Notification)[7], Cache Locks[2] 등이 있다. ROWA(Read One/ Write All)에 따라 사본 관리를 하는 회피-기반 기법으로는 CB[7], O2PL(Optimistic 2 Phase Locking)[3], Notify Locks[2] 등이 있다.

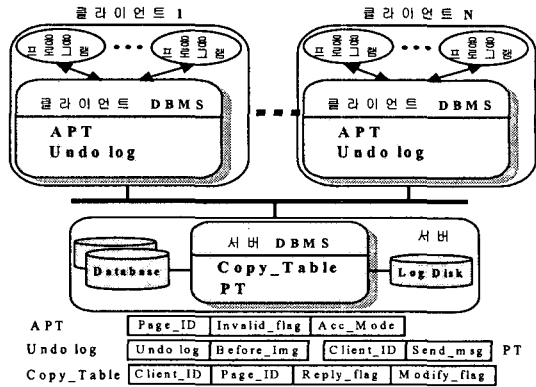
캐쉬 일관성 유지 기법에 대한 관련 연구 중에서 최근 들어 회피-기반 기법에 대한 연구가 많이 진행되었다[5, 6, 8]. 회피-기반 기법은 크게 CB과 O2PL을 기반으로 하는 기법들로 나누어질 수 있다. CB을 기반으로 하는 기법들은 클라이언트의 갱신의도를 동기적으로 선언함으로써 클라이언트가 서버의 응답을 기다리는 시간이 길어지고, O2PL을 기반으로 하는 기법들은 트랜잭션 종료 단계까지 갱신의도 선언을 하지 않음으로써 트랜잭션 철회율이 높아진다는 단점이 있다.

최근에 제안된 캐쉬 일관성 유지 기법으로는 ACBL(Adaptive-granularity CallBack Locking)과 AOCC(Adaptive Optimistic Concurrency Control) 그리고 AACC(Asynchronous Avoidance-based Cache Consistency) 기법 등이 있다[5, 6, 8]. ACBL 기법은 CB 기법을 기반으로 동시성 제어 및 사본관리의 단위를 융통성 있게 수용함으로써 트랜잭션 철회율이 낮다는 특징이 있다. AOCC 기법은 트랜잭션 종료 시점까지 서버와 통신을 하지 않음으로써, 전송 메시지 수가 낮고 트랜잭션 처리율이 높다는 특징을 갖는다. AACC 기법은 공유 관독(shared-read) 록과 개인 관독(private-read) 록으로 구분하여 록을 관리를 함으로써 위 두 기법의 단점을 보완했다는 특징을 갖는다.

3. 제안한 캐쉬 일관성 기법

3.1 클라이언트/서버 구조

본 논문의 연구 환경은 [그림 1]과 같이 하나의 서버와 다수의 클라이언트들이 네트워크를 통해 연결된 클라이언트/서버 데이터베이스 시스템 구조이다.



[그림 1] 클라이언트/서버 데이터베이스 시스템 구조

가. 클라이언트에서의 처리

클라이언트는 완전한 기능을 갖는 클라이언트 DBMS와 트랜잭션을 생성하는 응용프로그램으로 구성된다. 클라이언트 DBMS는 자신의 버퍼에 캐쉬한 페이지의 정보를 유지하며, 지역적인 록 관리를 수행한다. 일관성 유지 기법의 기술을 용이하게 하기 위하여 트랜잭션은 클라이언트에서만 생성되고, 어느 한순간에는 하나의 트랜잭션만 수행한다고 가정한다. APT는 트랜잭션이 이용한 페이지의 정보를 유지하며, Undo log는 갱신되기 이전 페이지가 저장된다.

① 페이지 요구

- 클라이언트에 페이지가 캐쉬되지 않았을 경우, 페이지 식별자(Page_ID)와 접근 모드(Acc_Mode)를 가지고 서버에게 페이지를 요구한다.

② 클라이언트에서 트랜잭션 처리 절차

- 만일 판독 연산이라면
 - APT(Page_ID, Invalid_flag=0, Acc_Mode = 0) 정보 갱신
- 만일 갱신 연산이라면
 - APT(Page_ID, Invalid_flag=0, Acc_Mode = 1) 정보 갱신
 - 페이지를 갱신하기 전에 Undo log에 페이지 저장
 - 서버에게 갱신의도를 등록(갱신 록 요구)

③ 서버로부터 갱신통보를 받은 경우, 다음과 같이 처리를 하고 서버에게 페이지 삭제 정보를 전송한다.

- 페이지가 갱신 중(APT.Acc_Mode == 1)인 경우, 트랜잭션 철회
- 페이지가 판독 모드로 사용 중(APT.Acc_Mode == 0)인 경우, APT.Invalid_flag = 1로 갱신한다. 이 페이지는 트랜잭션 종료 시 삭제 됨

- 페이지가 이용되고 있지 않다면, 페이지 삭제
- ④ 트랜잭션 종료처리
 - APT와 갱신된 페이지를 서버에게 전송 후, 서버의 응답을 기다림
 - 서버의 응답이 종료허가라면
 - APT.Invalid_flag == 1인 페이지 삭제
 - 록 해제, APT, Undo log를 삭제하고 트랜잭션을 종료한다.
 - 트랜잭션 철회 메시지를 받았다면
 - Undo log를 이용해 갱신되기 이전 상태로 되돌린다.

나. 서버에서 처리

서버 구조는 서버 DBMS, 로그 디스크와 데이터베이스 디스크로 구성되며 가장 최근의 원본 데이터를 갖고 클라이언트들의 요구를 처리한다. 클라이언트에게 데이터의 사본을 복사해주며 이 사본 데이터를 캐쉬한 클라이언트의 정보를 유지한다. 이것은 전역적인 록 관리의 역할을 하며, 데이터가 갱신되었을 때 갱신된 페이지를 캐쉬한 클라이언트들에게 갱신을 통보하는 역할을 한다.

서버 DBMS는 클라이언트에 캐쉬된 사본의 정보를 유지하는 Copy_Table과 클라이언트에 보낼 갱신통보를 저장하는 PT(Propagate Table)를 관리한다. Copy_Table과 PT의 구성은 [그림 1]과 같다. Client_ID와 Page_ID는 각각 클라이언트 식별자와 페이지 식별자를 의미하고, Reply_flag는 갱신통보를 보낸 클라이언트들로부터 응답 메시지의 수신 여부를 나타내며, Modify_flag는 안전한 데이터베이스 디스크에 갱신된 페이지의 반영여부를 나타낸다.

서버는 클라이언트의 페이지 요구에 대해 자신의 버퍼에 있는 페이지의 사본을 전송하고, 갱신 록 요구에 대해 록을 설정하고 갱신되는 페이지를 캐쉬한 클라이언트들에게 보낼 갱신통보 메시지를 PT에 저장한다. PT의 내용은 피기-백(piggy-back) 기법으로 전송한다. 트랜잭션의 종료 요구에 대해서 APT.Acc_Mode == 0인 것에 대해서 판독 록을 설정하고 갱신통보 메시지에 대한 응답을 모두 받았을 경우(Copy.Reply_flag == 0), 클라이언트에게 종료 허가 메시지를 보낸다.

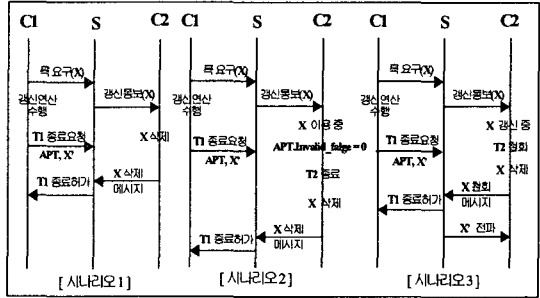
3.2 캐쉬 일관성 유지 기법의 실행 예

클라이언트1과 클라이언트2에 다음과 같은 트랜잭션들(T1, T2)이 각각 수행되고 있다고 가정하자.

T1 : read(Y) -> read(X) -> write(X)

T2 : read(Z) -> read(X) -> write(X)

[그림 2]는 트랜잭션 수행 과정에서 발생할 수 있는 시나리오들이다.



[그림 2] 캐쉬 일관성 시나리오

가. 시나리오1

T1이 X를 갱신하기 위하여 서버에게 X에 대한 갱신 록을 요구한다. 서버는 현재 X에 록이 걸려 있지 않기 때문에 갱신 록을 설정하고 갱신되는 X를 캐쉬한 클라이언트2에게 갱신통보 메시지를 보낸다. 클라이언트2에서 X를 사용하고 있지 않음으로 X를 삭제하고 서버에게 삭제 메시지에 대한 응답을 보낸다. 클라이언트1로부터 종료 요청을 받고 있던 서버는 클라이언트2로부터 갱신통보에 대한 응답을 받은 후 클라이언트1에게 종료 허가 메시지를 전송한다.

나. 시나리오2

클라이언트1에서 갱신되는 페이지 X를 클라이언트2의 T2에서 판독 모드로 사용하고 있는 경우다. 이때는 T2의 APT.Invalid_flag = 1로 설정한다. T2의 종료시 X는 삭제되고, 이에 대한 정보를 갱신통보에 대한 응답 메시지와 함께 서버에게 전송한다.

서버는 갱신통보를 보낸 클라이언트2로부터 응답 메시지를 받은 후 클라이언트1의 종료 요청을 허락한다.

다. 시나리오3

클라이언트1에서 갱신되는 페이지 X를 클라이언트2에서 갱신하려고 하는 시나리오이다. 이와 같은 상황은 서버에게 클라이언트2에게 갱신통보를 보내고 클라이언트2에서는 T2에 의해서 지역적이 갱신 록이 설정되고 아직 서버에게 갱신에 대한 록을 요청하지 않은 경우 발생할 수 있다. 이때는 T2는 철회되고, 이에 대한 정보를 갱신통보에 대한 응답 메시지와 함께 서버에게 전송한다.

클라이언트2로부터 응답 메시지를 받은 서버는 클라이언트2로부터

엔트2에게 새로운 페이지 X를 전파(propagation)한다. 이렇게 함으로써 전파와 삭제를 유동적으로 수행한다.

4. 다른 일관성 유지 기법들과의 비교

[표 1]은 기존의 캐쉬 일관성 기법인 ACBL, AACC, AOCC와 본 논문에서 제안한 캐쉬 일관성 기법과 비교한 결과이다.

[표 1] 기존 기법과의 비교

| | | ACBL | AACC | AOCC | 제안한 기법 |
|-----------------------|-------|---|--|-------------------------------|------------------|
| 서버로부터의 록 획득방법 | | 동기 | 비동기 | 지연 | 갱신 록에 대해서 비동기 |
| 클라이언트 캐쉬 일관성 행위 | | 삭제 | 삭제 | 삭제 | 삭제 및 전파 |
| 록 보유 시간 | | 길다 | 짧다 | 짧다 | 짧다 |
| 록 단위 | | 페이지 오브젝트 | 페이지 오브젝트 | 오브젝트 | 페이지 |
| 데이터 구조 | 클라이언트 | ROS, PROMISS MOS, PLOCKS, Undo log. | | ROS, MOS, Undo log. | APT, Undo log |
| | 서버 | MOS, Dir(P) O-LOCK(P) | | MOS, Invalid(C), Dir(P) | Copy_Table PT |
| 기타 | | 페이지 머지와 갱신 토른 | private-read shared-read lock 추가 | 갱신 통보 | 갱신통보 |

ACBL은 동시성 제어와 사본관리의 단위를 페이지와 오브젝트로 실시함으로써 트랜잭션의 철회율이 매우 낮다는 장점은 갖지만 다른 기법들에 비해 유지하는 정보가 많아 서버의 작업 부하가 높다는 단점을 가지며, AOCC는 ACBL 기법보다 판독 연산이 많은 환경에서는 높은 성능을 발휘하나 갱신 연산이 많아지면 철회율이 높다는 단점을 갖는다. AACC 기법은 공유 판독(shared-read) 록과 개인 판독(private-read) 록으로 구분함으로써 AOCC 기법보다는 철회율이 낮고 ACBL 기법보다는 성능이 좋다는 특징을 갖는다.

본 논문에서 제안한 기법은 클라이언트들이 갱신 록에 대해서 비동기적으로 록을 획득함으로써 트랜잭션의 처리율을 높이고, 클라이언트의 사본 관리에 있어 삭제를 기반으로 전파를 선택적으로 수행하는 특징을 갖는다. 그리고 AOCC 기법에서는 ROS와 MOS라는 자료구조를 유지하는데 비해, 본 논문에서는 APT에 트랜잭션이 이용한 페이지 식별자(Page_ID)와 이에 대한 접근 모드(Acc_Mode)를 뮌으로써 하나의 자료구조만 이용한다.

5. 결론 및 향후 연구

본 논문에서는 클라이언트/서버 환경에서 캐쉬의 일관

성을 유지하기 위한 알고리즘을 제안했다. 본 논문에서 제안한 알고리즘의 특징은 다음과 같다.

- ① 트랜잭션의 갱신 록을 서버로부터 비동기적으로 획득함으로써 트랜잭션 처리율을 높일 수 있다.
 - ② 갱신되는 페이지를 캐쉬한 클라이언트들에게 갱신통보 즉 삭제 정보를 보냄으로써, 부정확한 데이터를 가지고 연산을 수행함으로써 철회되는 트랜잭션을 줄일 수 있다.
 - ③ 갱신통보에 대한 응답으로 트랜잭션 철회 메시지를 보낸 클라이언트에 대해서만 갱신된 페이지를 전파함으로써 캐쉬 일관성 유지를 동적으로 수행한다.
- 향후 연구 과제로는 알고리즘의 성능 평가에 대한 연구가 필요하며, 제안한 캐쉬 일관성 알고리즘을 기반으로 동시성 제어에 대한 연구가 필요하다.

참고 문헌

- [1] A. Delis and N. Roussopoulos, "Modern Client-Server DBMS Architectures," Proc., ACM SIGMOD RECORD, pp.52-61, 1991.
- [2] K. Wilkison and Marie-Anne Neimat, "Maintaining Consistency of Client Cached Data," Proc., VLDB, pp.122-133, 1990.
- [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. Franklin, M. Carey, M. Livny, "Transactional Client-Server Cache Consistency: Alternatives and Performance," Proc., ACM TODS, pp.315-363, 1997.
- [5] M. Tamer, and Kaladhar, "An Asynchronous-Based Cache Consistency Algorithm for Client Caching DBMSs," Proc., VLDB, pp.440-451, 1998.
- [6] M. Zaharioudakis, M. Carey, M. Franklin, "Adaptive, Fine-Grained Sharing in a Client-Server OODBMS: A Callback-Based Approach," to appeared ACM TODS 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] R. Gruber, "Optimism VS. Locking: A Study of Concurrency Control for Client-Server Object-Oriented Databases," PhD thesis, MIT, 1997.