

비동기적 갱신 의도 선언에 의한 트랜잭션 처리의 회피-기반 캐쉬 일관성 유지 기법

박용문*, 이찬섭, 최의인
한국전자통신연구원, 한남대학교 컴퓨터공학과
ympark@etri.re.kr, {cslee, eichoi}@dblab.hannam.ac.kr

Avoidance-based Cache Consistency Technique on Transaction Processing Using an Asynchronous Write Intention Declaration

Yong-Mun Park*, Chan-Sub Lee, Eui-In Choi
ETRI, Dept. of Computer Science, Hannam University

요 약

고속 LAN과 같은 통신 장비의 발달로 클라이언트/서버 시스템 환경이 일반화됨에 따라 데이터베이스 시스템도 클라이언트/서버 환경을 지원하는 데이터 서버로서의 역할이 요구되었다. 또한, 다양하고 복잡한 형태의 제어 시스템들이 필요한 각 응용 분야에서 클라이언트/서버 시스템이 이용되고 있다. 이러한 상황에서 클라이언트들은 통신 비용 절감과 서버의 부하를 줄이기 위해 클라이언트의 버퍼에 데이터의 사본을 캐쉬(cache) 함으로써 클라이언트 시스템의 확장성 및 독립성을 추구한다. 하지만, 캐쉬한 데이터의 사본에 의해 갱신 연산이 수행되기 때문에 캐쉬 데이터의 일관성 유지를 위한 효율적인 방법들이 필요하다. 본 논문에서 제안한 기법은 클라이언트/서버 간의 데이터 전송이 페이지 단위로 행해지는 페이지-서버 환경에서 적용되는 회피-기반(avoidance-based) 기법으로써, 클라이언트가 데이터를 갱신할 때 갱신 의도를 비동기적으로 선언하는 캐쉬 일관성 유지 기법을 제안하였다. 그리고 제안한 기법과 최근에 제안된 기법들을 비교 분석하였다.

1. 서론

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

네트워킹에 의한 거의 모든 컴퓨터 시스템의 경우 클라이언트/서버간에는 많은 메시지가 발생한다. 그래서 클라이언트는 통신비용의 절감과 서버에 대한 의존도를 낮추기 위하여 자신의 메모리에 데이터의 사본을 캐쉬 한다. 즉, 클라이언트/서버 데이터베이스 시스템 구조에서 성능 향상을 위하여 클라이언트가 데이터를 가지고 작업을 수행하기 때문에 클라이언트들에 캐쉬한 데이터의 일관성 유지는 중요한 문제가 된다. 이러한 일관성 유지 문제는 데이터의 일관성 유지 뿐만 아니라, 트랜잭션 처리량을 높이기 위하여, 서버의 부하와 클라이언트/서버 간의 통신 오버헤드 문제가 고려되어야 한다. 이를 위해, 클라이언트의 캐쉬 데이터의 일관성을 유지하기 위한 기법들이 많이 제안되고 있다[1,4].

기존의 회피-기반 기법들은 갱신 의도 선언을 동기적

(synchronous)으로 수행하는 CBL(Callback Locking) 기법과 지연(deferred) 하는 O2PL(Optimistic 2-Phase Locking) 기법을 기반으로 연구가 이루어졌다.

회피-기반 기법은 캐쉬 데이터의 정확성 검사를 하지 않기 때문에 네트워크의 단절 등으로 인해 캐쉬에 부정확한 데이터가 존재할 경우 심각한 문제가 발생할 수 있다.

본 논문에서 제안한 캐쉬 일관성 기법은 트랜잭션이 갱신 연산을 수행할 때 서버에게 록(lock)을 요청한 후, 서버의 응답을 기다리지 않고 계속 트랜잭션 처리를 수행하는 비동기적인 기법을 이용한다.

본 논문의 구성은 2장에서 기존의 캐쉬 일관성 기법들과 최근에 제안된 기법들에 대해 기술하고, 3장에서는 본 논문의 연구 환경과 클라이언트/서버 환경에서 제안한 일관성 유지 기법들의 수행과정을 기술하였다. 그리고 4장에서는 최근에 제안된 일관성 기법들과 비교 분석하고, 마지막으로 5장에서 결론과 향후 연구 방향을 제시하였다.

2. 관련 연구

2.1 캐쉬 일관성 유지 기법

캐쉬 일관성 유지 기법은 트랜잭션이 클라이언트에 캐쉬한 데이터에 접근할 때에 그 데이터의 정확성에 대한 확인 여부에

따라 회피-기반(avoidance-based) 기법과 탐지-기반(detection-based) 기법으로 구분된다. 그림 1은 본 논문에서 제안하는 캐쉬 일관성 기법과 비교하기 위해 기존의 기법들을 정확성 검사 시점에 따라 간략하게 분류하였다[5].

	동기 (Synchronous)	비동기 (Asynchronous)	지연 (Deferred)
회피기반	CBL ACBL	AACC	O2PL
탐지기반	C2PL	NWLN	AOCC

그림 1. 캐쉬 일관성 기법 분류

탐지-기반에 속하는 기법은 C2PL(Cache 2-Phase Locking), NWLN(No Wait Locking w/Notification), AOCC(Adaptive Optimistic Concurrency Control)[2,4], Cache Locks[1,4,5] 기법 등이 있다.

ROWA(Read One/Write All)에 따라 사본을 관리하는 회피 기반 기법은 트랜잭션들이 부정확한 데이터를 접근할 기회를 주지 않음으로써 일관성을 유지하는 기법으로 CBL[4], O2PL(Optimistic 2-Phase Locking)[2], Notify Locks[1], AACC (Asynchronous Avoidance-based Cache Consistency)[4,5] 기법 등이 있다.

가. 탐지-기반(detection-based) 기법

탐지-기반 기법은 클라이언트의 캐쉬에 부정확한 데이터(stale data)의 존재를 허용하기 때문에 트랜잭션이 데이터를 이용할 때 즉, 트랜잭션이 종료되기 전에 서버에게 캐쉬 데이터의 정확성 검사를 요청한다. 따라서, 탐지-기반 기법은 단지 한 개의 클라이언트와 서버를 포함하기 때문에 캐쉬 관리 방법이 간단하다. 그렇지만, 서버의 의존도가 높고 클라이언트와 서버의 통신량 증가로 오버헤드가 발생하는 문제점이 있다.

나. 회피-기반(avoidance-based) 기법

캐쉬 일관성 유지 기법에 대한 관련 연구 분야에서 최근 들어 회피-기반 기법에 대한 연구가 많이 진행되었다. 회피-기반 기법은 크게 CBL(Callback Locking)과 O2PL(Optimistic 2-Phase Locking)을 기반으로 한 기법들로 나누어질 수 있다. CBL을 기반으로 하는 기법들은 클라이언트의 갱신 의도를 동기적으로 선언함으로써 클라이언트가 서버의 응답을 기다리는 시간이 길어지고, O2PL을 기반으로 하는 기법들은 트랜잭션 종료 단계까지 갱신의도 선언을 하지 않기 때문에 트랜잭션 철회율이 높아지는 단점이 있다.

1) CBL(Callback Locking) 기법

Callback Locking 기법에서 클라이언트는 트랜잭션에게 지역 록을 허용하기 전에 즉시 서버에게 갱신 페이지에 대해 자신의 갱신 의도를 선언해야 한다. 서버는 갱신되는 페이지를 가지고 있는 모든 클라이언트에게 메시지를 보냄으로써 충돌이 발생한 사본을 "callback" 하고, 해당 클라이언트들로부터 응답을 받은 후 클라이언트에게 록을 허용한다.

Callback Locking 기법은 클라이언트가 서버의 응답을 받은 후 트랜잭션 처리를 진행할 수 있기 때문에 네트워크를 통한 많은 메시지 전송이 문제가 된다. 또한 서버의 callback 요청에 대한 클라이언트들의 응답이 늦어질 경우, 클라이언트의 대기 시간이 길어지는 단점이 있다[1,4,5].

2) O2PL(Optimistic 2-Phase Locking) 기법

O2PL 기법은 회피-기반 기법을 사용하는 기법 가운데 가장 낙관적인 기법이다. 트랜잭션은 클라이언트에서 지역적으로 록을 획득하고, 종료 단계에서 서버에게 갱신 사본(update-copy) 록을 획득한다. O2PL 기법은 클라이언트의 수가 많아질수록 성능이 저하되는 단점이 있다.

3) AACC(Asynchronous Avoidance-based Cache Consistency) 기법

AACC 기법은 회피-기반 캐쉬 일관성 기법으로, 갱신 의도를 비동기적으로 서버에게 선언하는 기법이다. AACC는 AOCC (Avoidance-based Optimistic Cache Consistency)의 문제점인 높은 철회율과 ACBL(avoidance-based Callback Locking)의 문제점인 많은 메시지 전송과 대기 시간에 대한 해결책을 제안한 기법이다. 이 기법은 CB 기법을 기반으로 하고, 클라이언트/서버 모두 록 관리를 수행하며 페이지와 객체 단위의 록을 관리한다.

3. 비동기적 갱신의도 선언에 의한 캐쉬 일관성 유지 기법

3.1 클라이언트/서버 구조

본 논문의 연구 환경은 그림 2와 같이 하나의 서버와 다수의 클라이언트들이 고속 LAN을 기반으로 하는 네트워크를 통해 연결된 클라이언트/서버 데이터베이스 시스템 구조이다.

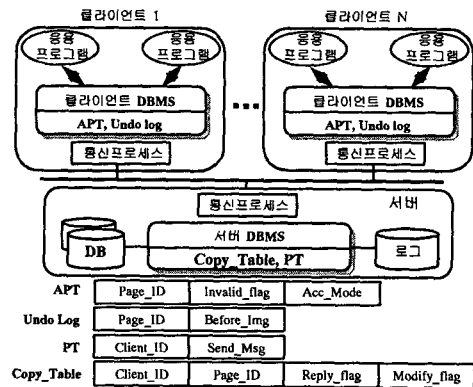


그림 2. 데이터베이스 시스템 구조 및 데이터 구조

서버는 원본 데이터를 갖고, 클라이언트들의 요구를 지원하며 안전한 데이터베이스 디스크와 로그 디스크를 갖는다. 서버는 각 클라이언트 캐쉬 데이터의 정보를 유지하고, 데이터가 갱신되었을 때 갱신된 페이지를 캐쉬한 클라이언트들에게 갱신을 통보하는 권한을 가진다

클라이언트는 트랜잭션을 생성하는 응용프로그램, 서버와의 통신을 담당하는 통신프로세스, 그리고 클라이언트 DBMS로 구성된다. 클라이언트 DBMS는 지역동시성제어기(local concurrency control manager), 클라이언트 버퍼 관리기(client buffer manager), 클라이언트관리기(client manager), 자원관리기(resource manager)로 구성된다. 클라이언트 버퍼 관리기는 현재 클라이언트 버퍼에 캐쉬한 페이지들의 정보를 유지한다.

3.2 클라이언트/서버 처리 알고리즘

가. 클라이언트 처리

클라이언트 DBMS는 자신의 버퍼에 캐쉬한 페이지 정보를 유지하며, 지역적인 록 관리 기능을 수행한다. 일관성 유지 기술을

용이하게 적용하기 위하여 트랜잭션은 클라이언트에서만 생성되고, 어느 순간에는 하나의 트랜잭션만 수행한다고 가정한다. APT는 트랜잭션이 이용한 페이지 정보를 유지하며, Undo log는 갱신되기 이전 페이지가 저장된다.

① 페이지 요청

클라이언트에 해당 페이지가 캐쉬 되지 않았을 경우에 페이지 식별자(Page_ID)와 접근 모드(Acc_Mode)를 가지고 서버에게 페이지를 요청한다.

② 클라이언트에서 트랜잭션 처리 절차(정상 상태)

- APT(Page_ID, Invalid_flag, Acc_Mode) 정보갱신
- 페이지를 갱신하기 전에 Undo log에 페이지 저장
- 서버에게 갱신의도를 등록(갱신 목록 요구)

③ 서버로부터 갱신 통보를 받은 경우, 서버에게 페이지 삭제 정보를 전송한다.

④ 트랜잭션 종료처리

- APT와 갱신된 페이지를 서버에게 전송한 후 서버의 응답을 기다린다.
- 서버의 응답에 따라 APT.Invalid_flag == 1인 페이지 삭제하고 Undo log를 이용해 갱신 이전 상태로 되돌린다.

나. 서버 처리

서버 DBMS는 클라이언트에 캐쉬 된 사본의 정보를 유지하는 Copy_Table과 클라이언트에 보낼 갱신통보를 저장하는 PT(Propagate Table)를 관리한다. 서버는 클라이언트의 페이지 요구에 대해 자신의 버퍼에 있는 페이지의 사본을 전송하고, 갱신 목록 요청에 대해 목록을 설정한 후, 갱신되는 페이지를 캐쉬한 클라이언트들에게 전송할 갱신통보 메시지를 PT에 저장한다. PT의 내용은 피기-백(piggy-back) 기법으로 전송한다. 트랜잭션의 종료 요청에 대해서 APT.Acc_Mode == 0인 것에 대해서 판독 목록을 설정하고, 갱신통보 메시지에 대한 응답을 모두 받았을 경우(Copy.Reply_flag == 0), 클라이언트에게 종료 허가 메시지를 보낸다.

3.3 캐쉬 일관성 유지 기법의 트랜잭션 시나리오

클라이언트1과 클라이언트2에 다음과 같은 연산 순서로 트랜잭션들(T1, T2)이 각각 수행되고 있다고 가정한다.

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

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

그림 3은 트랜잭션 수행 과정에서 발생할 수 있는 시나리오를 나타내었다.

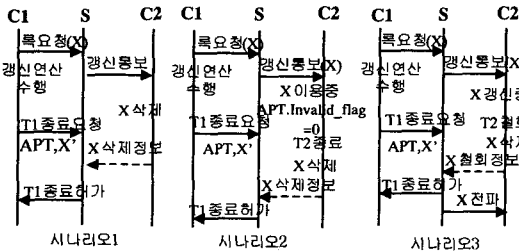


그림 3. 캐쉬 일관성 수행 시나리오

4. 비교 분석

본 논문에서 제안한 기법은 클라이언트의 사본 관리에 있어 삭제를 기반으로 전파를 선택적으로 수행한다. 그리고 AOCC 기법에서는 ROS와 MOS라는 자료구조를 유지하는데

비해, 본 논문에서는 일관성유지 알고리즘을 적용하기 위해 APT에 트랜잭션이 이용한 페이지 식별자(Page_ID)와 이에 대한 접근 모드(Acc_Mode)를 등록해서 한 개의 자료 구조만을 이용한다.

표1은 최근에 제안된 캐쉬 일관성 기법들과 본 논문에서 제안한 캐쉬 일관성 기법을 비교한 결과이다.

표1. 기존 기법과의 비교표

		ACBL	AACC	AOCC	제안한 기법
특	획득	동기	비동기	자연	갱신 목록에 대해 비동기
사본	관리	삭제	삭제	삭제	삭제, 전파
메시지	수	많다	적다	적다	없다
특	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
기타		페이지 머지 갱신 토른	private-read shared-read lock 추가	갱신통보	갱신통보

5. 결론 및 향후 연구

본 논문에서는 클라이언트/서버 환경에서 캐쉬 일관성을 유지하기 위한 기법을 제안하였다. 본 논문에서 제안한 캐쉬 일관성 기법은 클라이언트와 서버 간의 통신 비용을 줄이기 위해 갱신 연산에 대해서만 갱신의도를 등록하는 방법을 적용하였고, 트랜잭션이 서버의 응답 대기 시간을 줄이기 위해 비동기적인 갱신의도선언 기법을 이용하였다. 본 논문에서 제안한 알고리즘의 특징은 다음과 같다.

- ① 트랜잭션의 갱신 목록을 서버로부터 비동기적으로 획득함으로써 트랜잭션의 처리율을 높일 수 있다.
 - ② 갱신되는 페이지를 캐쉬한 클라이언트들에게 갱신통보 즉, 삭제 정보를 전송한다. 이것은 부정확한 데이터에 의해 철회되는 트랜잭션을 줄일 수 있는 하나의 방법이 된다.
 - ③ 트랜잭션 철회 메시지를 보낸 클라이언트에 대해서만 갱신된 페이지를 전파함으로써 캐쉬일관성유지를 동적으로 수행한다.
- 향후 연구 과제로는 본 논문에서 제안한 기법의 성능 평가에 대한 연구가 필요하며, 제안한 캐쉬 일관성 기법을 기반으로 동시성 제어에 대한 연구가 필요하다.

참고문헌

- [1] K. Wilkson and Marie-Anne Neimat, "Maintaining Consistency of Client Cached Data," Proc., VLDB, pp.122-133, 1990.
- [2] M. Carey, M. Franklin, M. Livny, E. Shekita, "Data Caching Tradeoffs in Client-Server DBMS Architectures," Proc., ACM SIGMOD, pp.357-366, 1991.
- [3] M. Franklin, M. Carey, M. Livny, "Transactional Client-Server Cache Consistency: Alternatives and Performance," Proc., ACM TODS, pp.315-363, 1997.
- [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 1997.