

캐시 일관성 유지 알고리즘의 성능평가를 위한 Client 모델링 및 구현

신성철*

*해군사관학교 전산학과
e-mail:seochul@hanmail.net

A Modeling & Implementation of Client for Performance Evaluation of Cache Consistency Maintenance Algorithms

Seoung-Chul Shin*

*Dept of Computer Science, Naval Academy

요 약

CAD와 같이 계산위주의 응용을 지원하기 위해 개발된 초기의 객체지향 데이터베이스 시스템은 클라이언트에서 자료의 요구와 처리를 허용하는 자료 탑재방식이다. 이 방식은 클라이언트 워크스테이션에서 수행되는 작업의 대부분을 수행할 수 있는 지역 DBMS를 허용한다. 클라이언트 워크스테이션에서 자료와 잠금을 캐싱하는 것은 클라이언트-서버 데이터베이스 시스템의 성능 향상을 위한 중요한 기술이다. 이 논문에서는 클라이언트 워크스테이션이 자신의 지역 데이터베이스를 유지하면서 회피 또는 탐지 기반의 캐시 일관성 유지 알고리즘들의 성능을 평가할 수 있도록 하는 클라이언트 워크스테이션을 모델링하고 구현하였다. 이 시스템은 3개의 클라이언트로 구성되고, 각 지역 동시성 제어 기법으로는 2PL을 사용하였다. 이 모델은 멀티데이터베이스 시스템 환경으로 확장 가능하다.

1. 서론

분산 컴퓨팅과 객체지향의 장점들은 응답성과 분산 환경에서의 자료 공유와 같은 복잡성을 지원하기 위하여 클라이언트-서버 컴퓨팅 모델을 채택한 새로운 개념의 데이터베이스 시스템 개발을 위해 결합되었다. 현재의 관계 데이터베이스 시스템들은 대부분의 쿼리 처리를 서버에서 수행하고 클라이언트들은 사용자 인터페이스만을 관리하는 쿼리 탑재(query shipping) 방식인 반면, CAD와 같이 계산위주의 응용을 지원하기 위해 개발된 초기의 객체지향 데이터베이스 시스템은 전형적으로 클라이언트에서 자료의 요구와 처리를 허용하는 자료 탑재(data shipping) 방식이다[2]. 자료 탑재 시스템은 클라이언트 워크스테이션에서 수행되는 작업의 대부분을 수행할 수 있는 DBMS를 허용하기 때문에 워크스테이션 기반의 환경에 특별히 잘 적용된다.

객체 기반 DBMS를 위한 자료 탑재 방식의 장점

은 자료를 응용에 밀접하게 이동시킴으로써 객체 기반 DBMS들의 programmatic 인터페이스를 통해 영구적 자료 구조 내에서의 navigation을 가속화하고, DBMS의 많은 기능들을 서버에서 클라이언트로 이동시킴으로써 효율성 향상을 제공한다[2]. 반면에, 네트워크에 민감하고 클라이언트에 의해 요청된 자료량이 방대할 경우 서버의 병목현상이 나타날 수 있다. 이러한 병목현상을 회피하기 위해서는 자료 캐싱(data caching)을 이용하여 지역 클라이언트의 자원을 사용하는 것이 한 방법이 될 수 있다. 클라이언트 자료 캐싱은 클라이언트들이 서버로부터 자료들의 사본을 보유할 수 있게 하는 것으로써 국부성(locality)의 측면에서도 자료 캐싱은 클라이언트가 서버로부터 요청해야하는 자료량을 상당히 감소시킬 수 있다.

클라이언트 캐싱은 캐시된 자료에 대한 자료 유지기간에 따라 트랜잭션내(intra-transaction) 캐싱(트랜잭션이 수행되는 동안에만 캐시된 자료를 캐시에 유

지)과 트랜잭션간(inter-transaction) 캐싱(트랜잭션의 종료 여부에 관계없이 캐시된 자료가 캐시에 계속 유지)으로 분류될 수 있다. 트랜잭션내 캐싱은 비교적 간단하지만 새로운 트랜잭션이 수행될 때마다 필요한 자료를 서버로부터 새로 얻어야 하고, 트랜잭션간 캐싱은 서버의 자료와 캐시된 자료사이에 일관성 유지를 위한 프로토콜이 필요하다.

2. 캐싱 알고리즘의 특성 및 요구사항

자료의 캐싱은 트랜잭션이 처음으로 접근을 시도하는 자료가 캐시에 존재하지 않을 경우에 수행되는 것으로 만약 타당성 검사 초기화 시점이 트랜잭션을 시작하는 단계에서 수행된다면 캐시 내의 유효하지 않는 자료는 타당성 있는 자료로 대체되어 진다.

클라이언트에 캐싱된 자료가 캐시에 유지되어지는 기간은 캐시 일관성 유지 알고리즘이 어떤 트랜잭션적인 캐싱 방법을 채택하였느냐에 따라 달라지기 때문에 이 특성은 고려되어야 한다. 그러나 유효하지 않는 자료가 캐시에 존재한다 해도 트랜잭션의 타당성 검사 행동으로 자료의 타당성을 판단할 수 있기 때문에 캐시 내의 유효하지 않는 자료에 대한 별도의 고려는 필요 없다.

<표 1> 캐시 일관성 유지 알고리즘들의 특성

기능	행동	분류
Validation Check Initiation	Synchronous Asynchronous Deferred	탐지
Change Notification Hints	Asynchronous	회피
Remote Update Action	Propagation Invalidation Dynamic	탐지 회피
Write Intention Declaration	Synchronous Asynchronous Deferred	회피
Write Permission Duration	Transaction Client	회피
Remote Conflict priority	Wait Preempt	회피

탐지 기반의 알고리즘 특성에서 타당성 검사 초기화 시점은 시뮬레이션 대상 알고리즘에 따르는 것이기 때문에 고려할 필요가 없다. 그러나 클라이언트가 서버로 타당성 검사를 요청하는 기능은 구현되어야 한다. 갱신 통지 힌트에 대한 구현은 시뮬레이션 대상 알고리즘에서 이 기능을 요청한 시점까지의 변경된 자료와 함께 원격 갱신 행동을 포함하여(동적인 원격 갱

신 행동을 위해) 서버로 전달할 수 있도록 구현되어야 한다.

회피 기반의 알고리즘 특성에서 일관성 행동의 초기화 시점에 대한 구현은 시뮬레이션 대상 알고리즘에 따라 다르지만 그 행동으로 인한 서버의 행동이 뒤따르기 때문에, 클라이언트에서 일관성 행동의 초기화 사건이 발생했음을 서버에 알리는 기능 구현은 필요하다. 클라이언트의 갱신 의도 선언은 시뮬레이션 대상 알고리즘이 이 기능을 요청한 시점의 자료에 대한 갱신 의도를 서버에게 알리는 기능이 구현되어야 하며, 갱신 허락을 요구한 자료에 대해 갱신 허락 소유 여부를 기록하는 기능과 어떤 자료에 대해 갱신 허락을 소유하고 있는 지를 조사하는 기능도 구현되어야 한다. 갱신 허락 보유 기간에 대한 구현 역시 시뮬레이션 대상 알고리즘에 따라 다르므로 특정 자료에 대한 갱신 허락 소유를 반환시키는 기능이 구현되어야 한다. 원격 충돌 우선순위에 대한 구현에서 대기 방법은 클라이언트의 지역 DBMS의 정책에 따라 수행되도록 하면 되지만, 선점 방법은 클라이언트의 지역 DBMS의 수정이 필요하게 되어 지역 자치성을 보장할 수 없다(2PL의 경우 먼저 읽기 잠금을 가진 트랜잭션이 갱신 트랜잭션으로 인하여 철회될 수 있도록 지역 DBMS가 수정되어야 한다).

원격 갱신 행동은 탐지 기반과 회피 기반의 알고리즘에서의 행동이 동일하다. 이 행동은 전파와 무효화 기능만을 구현한다. 동적인 방법은 이 두 방법의 혼용으로 시뮬레이션 대상 알고리즘에서 선택되어 사용되어지기 때문이다.

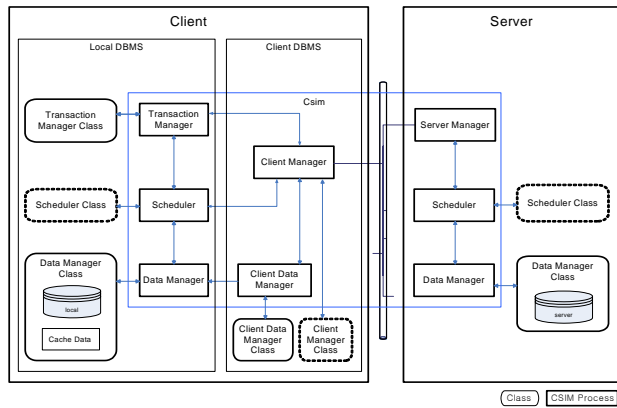
클라이언트들이 자신의 독자적인 지역 데이터 베이스를 운용하기 때문에 클라이언트들의 지역 자치성을 보장하여야 한다. 그러나 일부 알고리즘은 지역 자치성을 보장하지 않을 수 있기 때문에 이러한 경우에는 시뮬레이션 시 별도의 조작을 필요로 한다.

3. 클라이언트 워크스테이션의 모델링 및 구현

이 논문에서는 지금까지 제안된 많은 캐시 일관성 유지 알고리즘들의 성능 평가를 효율적으로 지원할 수 있도록 하는 시뮬레이션을 위한 클라이언트 워크스테이션 모델 구조에 대한 프로토타입을 제안한다. 제안한 클라이언트 워크스테이션 모델의 구현은 Visual studio 6.0 C++와 Microelectronics and Computer Technology Corporation에서 개발한 CSIM18 시뮬레이션 library를 이용하여 객체지향으로 구현하였다.

이 논문에서 제안하는 모델의 각 클라이언트에는

상호 이질형(heterogeneous)일 수 있는 지역 DBMS와 자료 탑재 시스템에서 서버 DBMS의 부하를 감소시키기 위해 사용될 수 있는 클라이언트 DBMS가 동시에 존재하며 그 구조는 그림 1과 같다. 구현은 클라이언트 3개, 각 클라이언트들의 지역 동시성 제어 기법은 모두 2PL을 사용하는 것으로 하였다.



(그림 1) 시스템 모델

지역 DBMS는 트랜잭션의 수행 전반에 대한 관리 기능을 수행하며, 클라이언트 DBMS는 서버와 관련된 제반 사항에 대한 기능을 수행한다. 이 두 모듈의 상호작용에 의해 자료가 캐싱되고 트랜잭션들이 수행되어야 하기 때문에 지역 DBMS에 약간의 수정을 필요로 한다. 클라이언트에서는 트랜잭션을 생성하고, 생성된 트랜잭션은 다른 클라이언트의 자료를 접근하지 않는다. 서버는 일반적인 클라이언트-서버 구조에서의 서버 기능을 수행한다.

클라이언트의 지역 DBMS는 트랜잭션 관리자, 스케줄러, 자료 관리자로 구성되며 클라이언트 DBMS와의 인터페이스를 제외하면 일반적인 DBMS와 동일하다. 트랜잭션의 모든 연산들은 지역 DBMS의 제어 하에 수행된다.

트랜잭션 관리자는 지역 데이터베이스 자료만을 접근하는 트랜잭션을 지역 트랜잭션으로, 지역 데이터베이스 자료와 같이 서버 자료까지를 접근하는 트랜잭션 또는 서버 자료만을 접근하는 트랜잭션을 전역 트랜잭션으로 구분하여 관리한다. 지역 트랜잭션의 경우는 일반적인 데이터베이스에서의 관리 방법과 동일하다. 전역 트랜잭션이 수행을 시작하면 접근하려는 서버 자료 항목에 대한 리스트를 클라이언트 관리자(client manager)에게 통보하고, 각 전역 연산에 대한 수행 여부를 클라이언트 관리자로부터 허락받는다(갱신 전과 트랜잭션의 경우는 제외).

스케줄러는 스케줄링 테이블과 지역 동시성 제어

(2-Phase Locking, Serialization Graph Test, Time-Stamp 등)를 위한 자료 구조를 유지하면서 지역/전역 트랜잭션 구분 없이 지역 스케줄링 정책에 따라 행동한다. 클라이언트 관리자로부터 캐시 자료에 관련된 정보를 받아 지역 스케줄링과 관련된 자료 구조를 갱신한다. 또한 클라이언트 관리자에게 필요시 전역적 동시성 제어를 위해 필요한 지역 동시성 정보(전역 트랜잭션들의 대기 그래프, 직렬화 그래프 등)를 제공한다. 원격 갱신 행동에 따른 갱신 트랜잭션의 연산 수행을 위한 기능(우선순위에 의한 선점과 같은 원격 갱신 행동을 위해 갱신 트랜잭션의 우선순위를 조정 또는 부여)을 제공한다.

자료 관리자는 캐시 자료 관리자(cache data manager)로부터 캐시 자료의 생성 및 삭제에 대한 요구를 받아 처리하고, 그 결과를 캐시 자료 관리자에게 통보한다. 지역 데이터베이스 자료 접근과 캐시 자료 접근을 구분하여 지역/전역 연산을 수행하고 그 결과를 지역 스케줄러에게 돌려주고, 캐시 자료에 대한 연산의 경우에는 캐시 자료 관리자에게 통보한다.

클라이언트 DBMS는 클라이언트-서버 시스템에서 캐싱과 캐시 일관성 유지와 관련된 기능을 수행한다. 이 모듈은 서버와의 인터페이스를 담당하는 클라이언트 관리자와 클라이언트 캐시의 자료들을 관리하는 캐시 자료 관리자로 구성된다.

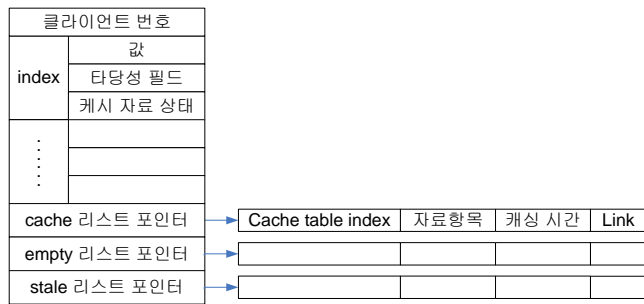
클라이언트 관리자는 시뮬레이션 대상 알고리즘에서 채택한 특성(표1)들에 따른 기능을 수행하고, 서버와의 통신을 담당하며, 원활한 기능 수행을 위해 필요한 자료 구조를 가지며, 캐시 일관성 유지 알고리즘 자신이 가질 수 있는 동시성 제어 기법과 트랜잭션 완료 프로토콜을 수행한다. 서버로부터 원격 갱신 행동을 수신했을 때는 갱신 전과 트랜잭션을 생성하여 지역 DBMS의 트랜잭션 관리자에게 제출한다. 자료가 캐시 되었거나 캐시된 자료가 삭제된 경우 지역 DBMS의 스케줄러에게 알려 지역 스케줄링 관련 자료를 수정할 수 있게 하고, 필요시 전역 교착상태를 해결하기 위한 대기 그래프 등과 같은 정보를 스케줄러에게 요구할 수 있다.

캐시 자료 관리자는 캐시 자료와 관련된 정보를 유지 관리하며, 자료 관리자에게 캐시 자료의 생성 및 삭제를 요청한다. 이 결과를 클라이언트 관리자에게 통보한다. 캐시 자료를 관리하는 캐시 테이블 구조는 그림 2와 같으며, 자료 캐싱에 대한 처리 과정은 그림 3과 같다. 캐시 자료에 대한 희생자 선정 알고리즘은 FIFO를 사용하였다.

각 모듈간 통신은 CSIM의 event나 mailbox 기능 및 패키지(그림 4)를 구성하여 이용하였고, 캐시 일

관성 유지 알고리즘들의 특성에 대한 행동들을 메소드로 제공함으로써 재사용이 가능하도록 하였다.

캐시 일관성 유지 알고리즘에 대한 성능 평가는 성능 평가 대상 알고리즘과 그와 유사한 알고리즘들의 성능이 상호 비교되어야 하기 때문에, 캐시 일관성 유지 알고리즘에 대한 선정에 어려움이 있어 캐시 일관성 유지 알고리즘에 대한 세부 구현은 향후 과제로 남겨둔다.



(그림 2) Cache Table 구조

이 논문에서는 이러한 캐시 일관성 유지 알고리즘들의 성능 평가를 위해 시뮬레이션을 위한 클라이언트 워크스테이션 모델을 제안하고 구현하였다. 제안된 모델은 클라이언트가 독자적인 지역 데이터베이스 시스템을 운용하면서 클라이언트-서버 데이터베이스를 이용하는 환경에 그 기반을 두었으며, 지금까지 제안된 캐시 일관성 유지 알고리즘들을 분석하여 요구되는 기능들을 도출하였다. 이렇게 도출된 요구들은 지역 자치성을 최대한 유지하면서 구현될 수 있도록 설계하고 구현하였다.

제안된 클라이언트 워크스테이션 모델은 객체 지향 기법으로 구현하여 소프트웨어의 재사용성을 제공함으로써 다양한 형태의 캐시 일관성 유지 알고리즘들의 성능평가를 위한 시뮬레이션을 용이하게 할 수 있도록 하였다. 또한 이 모델은 멀티데이터베이스(multidatabase) 시스템 환경에서 캐시 일관성 유지 알고리즘들의 성능 평가에도 활용할 수 있도록 확장 가능하다.

참고문헌

[1] M. J. Franklin, M. Carey, "Client-Server Caching Revisited", Proceeding International Workshop in Distributed Object Management, pp. 57-78, MAY 1992.

[2] M. J. Franklin, M. J. Cary, and M. Livny, "Transactional Client-Server Cache Consistency : Alternatives and Performance", ACM Transactions on Database Systems, Vol. 22, Num. 3, pp 315-363, 1997.

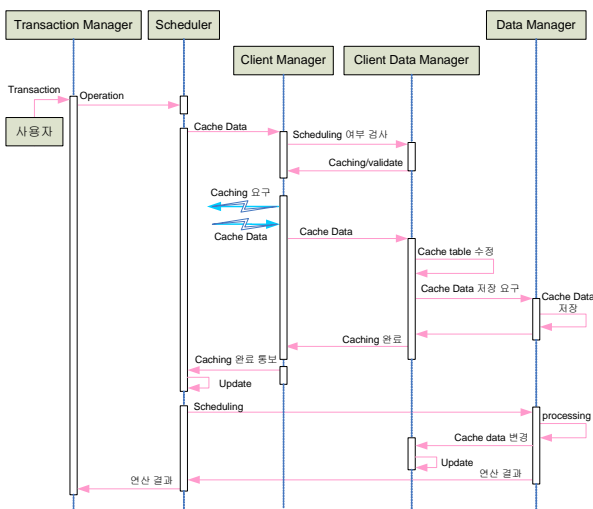
[3] M. T. Ozsu, K. Voruganti, and R. C. Unrau, "An Asynchronous Avoidance-based Cache Consistency Algorithm for Client Caching DBMSs", Proceedings of the 24th VLDB Conference, pp. 440-451, 1998.

[4] P. A. Bernstein, V. Hadzilacos, and N. Goodman, "Concurrency Control and Recovery in Database Systems", Addison-wesley, 1991.

[5] 『User's Guide CSIM18 Simulation Engine』, Mesquite Software, Inc., 1994.

[6] 신성철, "Client-Server 데이터베이스의 시뮬레이션을 위한 Client 모델", 해사논문집, 제48집, pp. 141-150, 2005.

[7] 엄기현, "클라이언트/서버 구조", 이한출판사, 1995.



(그림 3) Caching 처리도

id	Client No	data	value	command	op	Mode	TM-idx	Cache-idx
* id:packet id, op:세부 행동, Mode:명령/응답 구분								

(그림 4) Packet 구조

3. 결론

클라이언트의 자료 캐싱은 자료 중복의 동적인 형태로서 자료의 정확성에 대한 개념은 데이터베이스 시스템에서 트랜잭션 개념과 일치한다. 클라이언트 캐싱에서는 서버와 클라이언트에 캐시된 자료사이의 일관성 유지를 위한 프로토콜이 필요하다.