데이타베이스 공유 시스템에서 B-트리 인덱스를 위한 캐쉬 일관성 제어

온경오° 조행래
영남대학교 컴퓨터공학과
{ondal, hrcho}@cse.yeungnam.ac.kr

A Cache Coherency Control for B-Tree Indices in a Database Sharing System

Kyungoh Ohn° Haengrae Cho
Dept. of Computer Engineering, Yeungnam University

요 약

데이타베이스 공유 시스템(Database Sharing System: DSS)은 고성능 트랜잭션 처리를 위해 다수 개의 컴퓨터를 연동하는 방식으로, 각 노드들은 디스크 계층에서 데이타베이스를 공유한다. DSS에서 각 노드는 빈번한 디스크 액세스를 피하기 위해 최근에 액세스한 데이타 페이지와 인덱스 페이지들을 자신의 지역 메모리 버퍼에 캐싱한다. 이때 노드가 항상 최신의 페이지를 사용할 수 있기 위해서는 지역 버퍼에 캐싱된 페이지들의 일관성을 지원하여야 한다. 본 논문에서는 데이타 페이지에 비해 빈번히 액세스되는 인덱스 페이지의 정확성을 보장할 수 있는 캐쉬 일관성 제어 기법을 제안한다.

1. 서론

온라인 트랜잭션 처리를 필요로 하는 응용 분야들의 규모가 커지고 다양한 응용분야의 신설로 인해 트랜잭션 처리 시스템의 고성능화가 요구되고 있다. 고성능 트랜잭션 처리 시스템을 개발하는 방법으로는 방대한 예산이 소요되는 슈퍼 컴퓨터를 이용하는 것보다 저렴한 다수 개의 컴퓨터들을 연동하여 병렬 시스템을 구축하는 것이 효과적이다.

DSS는 다수 개의 컴퓨터 노드를 연동하여 병렬 데이타베이스 시스템을 구현하는 방식으로 각 노드들이 디스크 계층에서 전체 데이타베이스를 공유할 수 있다[3]. 따라서 분산 처리 작업이 단순해지며 노드들간의 부하 분산이 용이하다. 뿐만 아니라, 하나의 노드가 고장이 나더라도 다른 노드들은 기존 작업을 계속 수행할 수 있는 가용성과 새로운 노드의 추가로 인한 확장성 등의 장점을 가진다.

DSS에서 각 노드들은 최근에 액세스한 페이지를 자신의 지역 버퍼에 캐싱함으로써 디스크 액세스 수와 노드들간의 메시지 양을 줄여 성능을 높일 수 있다. 그러나 여러 노드에서 항상 최신의 페이지를 사용할 수 있기 위해서 각 노드에 캐싱되어 있는 페이지들 사이의 일관성을 유지해야 한다. 일반적으로 인덱스와 같이 빈번히 액세스되는 페이지들에 대해 데이타 페이지에 적용되는 캐쉬 일관성 제어 기법을 적용할 경우, 과도한 메시지 전송으로 인한 성능이 저하될 수 있다. 뿐만 아니라, B-트리 인덱스의 경우 트리 순회나 인덱스 스캔, 그리고 인덱스 페이지들의 분할/합병 등 데이타 페이지에 비해 복잡한 연산이 적용되므로 이러한 연산의 성격을 반영할 수 있는 별도의 캐쉬 일관성 제어 기법을 지원하는 것이 바람직하다[2].

이러한 관점에서 본 논문에서는 B-트리 인덱스를 위한 캐쉬 일관성 제어 기법을 제안한다. 제안한 기법은 캐쉬 일관성 제어를 위해 필요한 노드간 메시지 전송 빈도 수를 최소화하며, B-트리 순회 과정이 단순하다는 장점을 갖는다.

본 논문의 구성은 다음과 같다. 2절에서는 기존에 제안된 인덱스 페이지 캐쉬 일관성 기법에 대해 살펴보고, 3절에서는 본 논문에서 제안한 캐쉬 일관성 기법을 설명한다. 마지막으로 4절에서 결론 및 앞으로의 연구방향에 대해 논의하기로 한다.

2. 관련 연구

인덱스 페이지에 대한 기존 캐쉬 일관성 제어 기법으로 ARIES/IM[1]을 DSS환경으로 확장한 ARIES/IM-SD[2]와 낙관적 기법을 사용한 RIC[4]를 들 수 있다. ARIES/IM-SD는 DSS환경에서 제안되었으며, RIC는 의뢰자/서버 환경에서 제안되었다. 의뢰자/서버 환경에서 제안된 또다른 캐쉬 일관성 기법인 복합형 캐싱(Hybrid Caching) 일관성 관리 기법[5]은 의뢰자에서 인덱스 리프 페이지들만 캐싱하며, 중간 단계의 페이지들은 자체적으로 구성한다. 이 기법은 서버 노드가 갱신 및 캐쉬 미스를 처리하고, 그 결과 서버 노드에 부하가 집중될 수 있으므로 DSS에는 부적합하다.

ARIES/IM-SD는 갱신에 의해 중간 단계 인덱스 페이지가 변화될 때 그 페이지를 캐싱하고 있는 다른 노드들의 페이지를 무효화시킴으로써 캐싱된 페이지가 지역 노드에

서 유효하다면 항상 정확한 페이지임을 보장한다. 그리고 리프 페이지에 대한 캐쉬 일관성을 위해 물리적 로크(P-로크)를 이용한 소유자 개념을 지원한다. 단, ARIES/IM을 기반으로 하였으므로 리프 페이지에 대한 P-로크 요청은 다음 키 로킹 개념에 따라 데이타 레코드에 대한 로크 요청 메시지에 포함된다. 그리고, 트리 순회 과정에서 부모 페이지가 다른 노드에 의해 무효화되는 것을 감시하기 위해 자식 페이지에 대한 지역 래치를 획득한 후 부모 페이지의 무효화 여부를 검사한다. 부모 페이지가 무효화되었을 경우, 최신 버전을 디스크나 다른 노드로부터 액세스한 후 트리를 재 순회한다.

RIC는 의뢰자 노드에 캐싱된 인덱스 페이지를 순회하는 동안 이전 버전 인덱스 페이지를 액세스하는 것을 허용하되 동기화 시점(Synchronization Point: SP)에서 액세스한 인덱스 페이지가 최신의 것인지를 검사한다. 일반적으로 SP는 데이타에 대한 로크를 서버에 요청하는 시점을 의미한다. SP에서 의뢰자 노드는 로크 요청 정보와 함께 일관성 제어를 위한 정보들을 서버에게 전송한다. 일관성 제어 정보는 해당 인덱스를 포함하는 인덱스 세그먼트의 식별자와 세그먼트의 버전 번호로 구성된다. 서버는 세그먼트 버전 번호를 이용하여 세그먼트가 최신 버전인지를 판단하고, 이전 버전일 경우 서버 노드는 인덱스 세그먼트에 포함된 모든 페이지들의 page_LSN 벡터(Segment Coherence Information Structure: SCIS)를 로크 승인 메시지에 포함해서 전송한다. 의뢰자는 SCIS를 이용하여 변경된 페이지들을 무효화하고, 만약 현재 액세스 경로의 페이지들이 무효화되었을 경우 다시 인덱스를 순회한다.

ARIES/IM-SD는 캐쉬 무효화를 위한 통신 오버헤드가 존재하며, 부모 페이지가 무효화될 경우 다시 인덱스를 순회해야 한다는 단점이 있다. 이에 대해 RIC는 인덱스 순회상에서의 통신은 최소화할 수 있지만, 인덱스가 자주 갱신될 경우 SCIS 전송으로 인한 통신 오버헤드가 증가하며 실제 데이타와 관계없는 인덱스의 변화에 의해 여러 번 인덱스를 수정해야 하는 오버헤드가 발생한다.

3. 리프 페이지의 일관성 검사

RIC와 ARIES/IM-SD에서는 불필요한 인덱스 재 순회가 발생한다. 그림 1과 같이 소유자 노드에서 리프 페이지 P1의 분할로 인해 P2의 부모 페이지(P0)의 page_LSN이 변경될 수 있고, 그 결과 P2에 대한 캐쉬 일관성 검사를 요청한 다른 노드들은 변경된 P0에 의해 다시 인덱스 순회를 하여야 한다. 이때 P0의 최신 버전을 소유자 노드로부터 전송받아야 하므로, 인덱스 순회를 위한 지연이 발생한다. 그러나, P0의 최신 버전을 이용한 새로운 인덱스 순회의 결과 리프 페이지도 역시 P2이므로, P2의 관점에서는 인덱스 순회 및 P0의 전송이 필요없다.

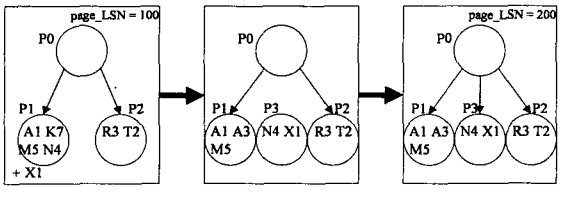

그림 1. P2와 관련없는 삽입연산

3.1 지역 노드에서 인덱스 트리의 순회

본 논문의 기본 알고리즘은 일관성 검사를 위해 지역 노드 N_i에서 전역 로크 관리자(Global Lock Manager: GLM)로 데이타 로크를 요청할 때 순회된 인덱스 페이지들의 페이지 식별자 리스트와 리프 페이지의 page_LSN을 데이타 로크 요청 메시지에 포함하는 것이다. N_i의 리프 페이지가 이전 버전일 경우 페이지 식별자 리스트를 이용하여 GLM은 루트 페이지부터 리프 페이지까지의 page_LSN과 소유자 정보를 N_i에게 전송하고, N_i는 최신 페이지를 전송받은 후 인덱스를 재 순회한다. 그러나, 리프 페이지가 최신 버전일 경우에는 중간 단계의 인덱스 페이지들이 이전 버전이라도 무효화를 하지 않으므로 무효화 오버헤드를 줄일 수 있다. 단, 모든 데이타 로크 요청 메시지에 순회된 인덱스 페이지들의 식별자 리스트를 포함함으로써 메시지 길이가 길어지는 단점이 있다.

로크 요청 메시지의 길이를 줄이기 위해 중간 단계 페이지들의 식별자를 포함하지 않는 것도 가능하다. 이 경우 리프 페이지가 이전 버전임을 GLM으로부터 통보받은 후, N_i는 중간 단계 페이지들의 식별자 리스트를 GLM 노드에 재 전송하여 소유자 정보와 page_LSN을 가지고 최신 페이지를 액세스할 수 있다. 그러나 이 방법은 리프 페이지가 이전 버전일 경우 메시지 전송 횟수가 증가하므로 인덱스가 빈번히 갱신되는 환경에서는 오히려 성능이 떨어진다.

노드 N_i의 인덱스 트리 순회 시 페이지 Pi에 대해 캐쉬 미스가 발생했을 경우, N_i는 GLM에게 Pi의 요청 메시지를 전송한다. 이때 요청 메시지에 Pi에 대한 정보만 포함될 경우 문제가 발생할 수 있다. 그 이유는 Pi가 다른 노드에서 삭제된 후 재 사용될 수 있고, N_i에서 Pi의 부모 노드에는 이러한 사실이 반영되지 않을 수 있기 때문이다. 예를 들면, 그림 1에서 P3의 삭제되기 전의 버전을 요청한 노드에게 그림 1의 P3을 전송하는 것은 잘못

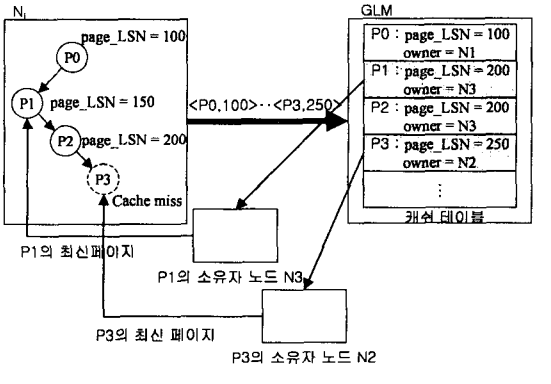

그림 2. 캐쉬 미스 시 GLM과 소유자 노드와의 동작

된 연산 결과를 초래한다. 이를 해결하기 위하여 Pi의 요청 메시지에는 Pi의 식별자 외에 순회 과정에서 현재까지 액세스한 조상 페이지들(즉, 루트 페이지부터 부모 페이지까지)의 [페이지 식별자, page_LSN] 리스트도 포함된다. GLM은 Pi의 조상 페이지들에 대한 page_LSN을 각 페이지의 최신 page_LSN과 비교하여 작을 경우 각 페이지의 소유자 노드로부터 N_i에게 최신 버전을 전송하도록 한다. N_i는 전송받은 페이지들 중에서 가장 상위 페이지부터 트리를 재 순회한다. 예를 들면, 그림 2에서 N_i는 P1과 P3의 최신 버전을 전송받은 후, P1부터 트리를 재 순회한다.

3.2 인덱스 연산

인덱스 트리는 Fetch Next, 삽입, 그리고 삭제 연산을 위해 액세스된다. 그리고 삽입에 의해 페이지가 분할되거나 삭제에 의해 페이지가 삭제되는 경우 인덱스 구조 변경(Structure Modification Operation: SMO)연산이 필요하다. 모든 연산에 대해 앞절에서 설명한 트리 순회 알고리즘을 거쳐 리프 페이지까지 액세스하였다고 가정한다.

3.2.1 Fetch Next

조건을 만족하는 다음 키를 찾기 위해 Fetch Next를 수행한다. Fetch Next는 추가적인 인덱스 순회를 하지 않고, 리프 페이지의 식별자와 page_LSN을 다음 키에 대한 데이타 로크 요청 메시지에 포함하여 GLM에게 전송한다. 이때 리프 페이지가 이전 버전이면 소유자 노드로부터 최신 리프 페이지를 전송 받는다. 리프 페이지에서 모든 키를 액세스한 후, 다음 혹은 이전 페이지가 있으면 NextPage, PrevPage를 참조하여 조건을 만족하는 모든 데이타들을 액세스할 수 있다.

3.2.2 삽입, 삭제

인덱스 트리를 순회하여 삽입, 삭제를 위한 정확한 리프 페이지를 찾고 연산을 위해 다음 키에 대한 갱신 로크를 요청한다. 이때 다른 트랜잭션에 의해 리프 페이지가 동시에 갱신되는 것을 방지하기 위해 리프 페이지에 대한 소유자 권한을 획득해야 하며, 이를 위해 리프 페이지의 식별자, page_LSN과 P-로크 요청 메시지를 데이타 로크 메시지에 포함시켜서 GLM에 전송한다. GLM은 리프 페이지의 현재 노유자 노드에게 소유권 이전을 요청하며, 현재 소유자 노드는 필요할 경우 리프 페이지의 최신 버전을 로크 요청 노드에게 전송한다. 이후 GLM은 리프 페이지에 배타적인 P-로크를 로크 요청 노드에게 할당함으로써 로크 요청 노드를 새로운 소유자 노드로 등록한다.

3.2.3 SMO

삽입, 삭제 연산에 의해 리프 페이지가 분할되거나 삭제될 경우 부모 페이지가 변경되어야 한다. 분할의 경우 분할이 필요한 페이지에 P-로크를 유지하고, 새로운 페이지를 할당받아 P-로크를 요청한 후 키들을 새로운 페이지로 옮긴다. 그리고 부모 페이지가 P-로크를 가지면 두 분할된 리프 페이지들에서 로크를 해제하고 부모 페이지를 변경한다. 이때 부모 페이지에 새로운 키가 삽입됨으로써 분할이 필요할 경우 리프 페이지와 같은 방식으로 수행한다. 삭제의 경우에도 리프 페이지의 삭제로 인해 부모 페이지가 삭제될 경우 더 이상 부모 페이지에 영향을 주지 않을 때까지 상향식으로 전파된다.

4. 결론 및 향후과제

DSS에서 인덱스와 같이 빈번히 액세스되는 페이지들에 대해 데이타 페이지에 적용되는 캐쉬 일관성 제어 기법을 적용할 경우, 과도한 메시지 전송으로 인한 성능이 저하될 수 있다. 본 논문에서 제안한 인덱스 페이지를 위한 캐쉬 일관성 제어 기법의 장점은 다음과 같다. 첫째, 인덱스의 루트 페이지에서 리프 페이지까지 순회하는 동안 캐쉬 일관성을 위한 다른 노드와의 추가적인 메시지 전송이 발생하지 않는다. 둘째, 인덱스 페이지에 대한 캐쉬 일관성 요청을 데이타 로크 요청 메시지에 통합함으로써 GLM으로의 메시지 전송 빈도 수를 줄일 수 있으며, 필요한 경우 실제로 액세스한 인덱스 페이지들의 캐쉬 정보만을 GLM으로부터 전송 받음으로써 불필요한 메시지 전송을 최소화한다. 마지막으로, 리프 페이지가 변경되었을 경우에만 중간 페이지들의 캐쉬 일관성을 검사하므로, ARIES/IM-SD에서 발생하는 중간 페이지에 대한 캐쉬 무효화 메시지 전송 오버헤드와 RIC에서 발생하는 빈번한 인덱스 재 순회 과정을 줄일 수 있다. 본 논문의 향후과제는 제안한 알고리즘을 구현하고 성능을 분석하여 기존의 알고리즘과 비교하는 것이다.

5. 참고문헌

[1] C. Mohan and F. Levine, "ARIES/IM: An Efficient and High Concurrency Index Management Method Using Write-Ahead Logging," in: *Proc. ACM SIGMOD* (1992) 371-380.

[2] C. Mohan and I. Narang, "Locking and Latching Techniques for Transaction Processing Systems Supporting the Shared Disks Architecture," Unpublished Manuscript (1995).

[3] C. Mohan and I. Narang, "Recovery and Coherency Control Protocols for Fast Intersystem page Transfer and Fine-Granularity Locking in a Shared Disks Transaction Environment," in: *Proc. 17th VLDB Conf.* (1991) 193-207.

[4] V. Gottemukkala, E. Omiecinski and U. Ramachandran, "Relaxed Index Consistency for a Client-Server Database," in: *Proc. Intl. Conf. on Data Engineering* (1996) 352-361.

[5] M. Zaharioudakis and M. Carey, "Highly Concurrent Cache Consistency for Indices in Client-Server Database Systems," in: *Proc. ACM SIGMOD* (1997) 50-61.