고성능 고가용성을 위한 ALTIBASE™ 자료저장 관리기의 설계

임화정° 이규웅,
상지대학교 컴퓨터공학과,
{hjlim°, kwlee}@sangji.ac.kr,

정광철
㈜ 알티베이스
jungkc@altibase.com

Design of ALTIBASE™ Storage Manager for High Performance and High Availability

Hwa-Jung Lim° Kyu-Woong Lee
Dept. of Computer Engineering, Sang-ji University

Kwang-Chal Jung
ALTIBASE.co.ctd

요 약

기존 디스크 기반 데이터베이스 관리 시스템은 디스크를 주저장장치로 활용하는 특징적 환경 때문에 주기억 상주 데이터베이스 관리 시스템의 메모리 관리, 인덱스 관리, 자료저장 관리 기능 등에 대한 설계 및 구현 기술이 기본적으로 다르다. 본 논문에서는 현재 상용 시스템으로 사용되고 있는 ALTIBASE™ 주기억 상주 DBMS의 설계 및 구현 내용을 기술한다

1. 서 론

통신산업 분야의 라우팅, 스위칭 응용분야나 실시간 산업 분야에서는 고성능 데이터 접근 및 효율적 데이터 관리를 요구하는 응용들이 증가하고 있다. 이러한 응용들은 개개의 트랜잭션에 대한 빠른 응답 시간 뿐만 아니라 트랜잭션의 영속성(durability)과 가용성(availability)을 요구하고 있어, 이를 만족하기 위해 기존 디스크 기반 데이터베이스 시스템들은 풍부한 메모리 공간을 활용하여 모든 데이터를 메모리에 적재하여 사용할 수 있는 대체방안을 제시하였다.

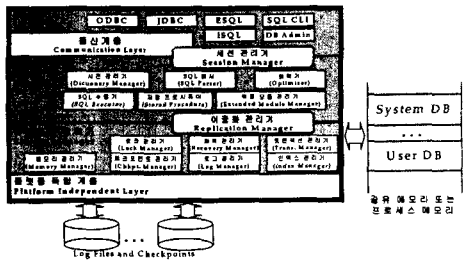
그러나, 디스크 공간을 주 저장장치로 설계된 시스템에서는 충분한 주기억장치 공간을 갖는다 하여도 순수 주기억 상주 데이터베이스 시스템에서의 성능 보다 우수하지 못함을 이미 다른 연구에서도 보였다[4, 5]. 또한, 기존 시스템에서 사용하던 기법들은 기본 저장장소가 기억장치로 변환되면서 직접적으로 적용하기 어려운 점이 많다[3]. 따라서 본 논문에서는 ALTIBASE™ 의 자료저장 관리자가 고성능 고가용성을 위해 사용한 여러 기법들을 소개한다.

본 논문은 2장에서 ALTIBASE™ 시스템의 기능 및 구조를 설명하고, 3장에서 트랜잭션 관리 기법, 회복 및 로깅 기법을 위한 설계 고려 사항, 데이터베이스 이중화 기법 및 충돌 상황에 대한 해결방법을 설명한다. 4장에서는 단위 트랜잭션 처리율을 시스템 부하의 변화에 따라 측정, 본 시스템이 실시간 응용 분야나 시간제약을 갖는 응용 분야에 적합한 성능을 제공함을 보이며, 5장에서 결론을 맺는다.

2. ALTIBASE™의 주요 기능 및 시스템 구조

2.1 ALTIBASE™ 시스템 구조도

ALTIBASE™는 주기억장치를 사용하는 관계형 데이터베이스 시스템으로서, 범용적 응용 뿐만 아니라 특정 실시간 응용에도 적합한 클라이언트-서버 구조와 응용 내장형 구조, 멀티 쓰레드 구조, 그리고 서버 연결 풀 구조를 제공한다. ALTIBASE™ 시스템의 구조는 <그림 1>과 같이 인터페이스 부분, 통신 계층, 질의 처리부, 그리고 자료저장 관리기로 분류할 수 있다.



<그림 1> ALTIBASE™ 시스템 구조

2.2 ALTIBASE™ 시스템의 주요 기능

ALTIBASE™ 시스템에서 트랜잭션을 처리하기 위한 기본적인 병행수행 제어 방법으로 다중 버전(multi- version)기법을 이용한 병행수행 제어 방법을 사용한다. 다중 버전 기법의 기본 전략에 따라 판독 전용(read-only)트랜잭션이 많고, 갱신 트랜잭션의 비율이 적은 응용 분야에 우수한 트랜잭션 처리율을 보인다[1, 2].

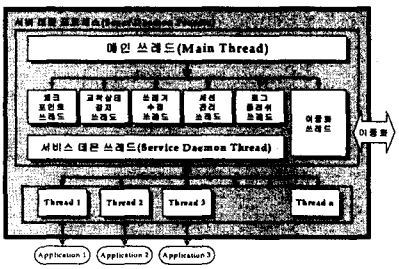
회복 관리 기능 중 트랜잭션의 영속성(durability)을 지원하기 위해 디스크와의 동기화 과정을 필수적으로 사용해야 한다. 트랜잭션의 로그 레코드 관리 기법으로 메모리 맵드 파일 또는 메모리 버퍼를 로그 버퍼로 선택적으로 사용할 수 있게 하고, "*log flush thread*" 를 구현하여 현재 수행 중인 트랜잭션에 간섭없이 디스크에 존재하는 로그 파일로 동기화 할 수 있도록 구현하였다. 메모리 역시 시스템 성능에 영향을 주기 때문에 실제 사용자 데이터와 메타 정보에 대한 내용을 저장하기 위한 영구공간(persistent space)과 인덱스 데이터나 질의 처리 시 필요한 임시공간(temporary space)으로 구분한다.

인덱스 관리 기능으로 캐쉬 계층의 최적화 기법과 캐쉬를 고려한 인덱스의 병행수행 제어 방법을 적용한 개선된 R*-Tree, T-Tree 및 B+-Tree를 제공하여 캐쉬 적중률을 높였으

며, 전체 트랜잭션의 수행 성능을 향상시키는 한편, 고가용성(high availability) 지원을 위해 물리적으로 분리되어 있는 여러 데이터베이스에 지역 데이터베이스의 변경내용을 원격지로 복제할 수 있는 이중화(replication) 기능을 지원한다. ALTIBASE™ 시스템에서는 데이터베이스의 변경 로그를 기반으로 하는 점대점(point-to-point) 이중화 기법을 적용한다..

3. 고성능 자료저장 관리기의 구조 및 설계

<그림 2>는 ALTIBASE™ 시스템의 핵심 기능을 제공하고 있는 자료저장 관리기의 서버 프로세스의 구성도 이다.



<그림 2> ALTIBASE™ 시스템의 서버 프로세스 구성도

3.1 트랜잭션 관리기의 설계

ALTIBASE™ 시스템의 트랜잭션 관리기는 트랜잭션 테이블(Transaction Table)과 트랜잭션 프리 리스트(Transaction Free List)를 관리하는 책임을 지고 있다.

ALTIBASE™ 시스템의 트랜잭션 관리기는 세가지 고립화 수준 즉, "*consistent*", "*repeatable*", "*no_phantom*" 을 제공한다.

- *Consistent* : 트랜잭션이 현재 읽은 데이터는 완료된 트랜잭션에 의한 것임을 보장. 갱신 일관성(update consistency)[2].
- *Repeatable* : 트랜잭션이 한 데이터에 대해 여러 번 읽기 연산을 수행하여도 언제나 같은 값을 판독할 수 있음을 보장. 약한 일관성(weak consistency)[2].
- *No_phantom* : 일반 데이터베이스 시스템에서 트랜잭션 수행의 기준으로 사용하는 직렬화 가능성(serializability) [7]을 보장.

이는 응용 시스템의 목적에 따라 다르게 설정하여 트랜잭션의 성능을 최대화 할 수 있도록 한다.

ALTIBASE™ 시스템은 부분 철회(partial rollback)를 지원하기 위하여 두 종류의 저장점을 제공한다. 명시적 저장점(explicit savepoint)은 사용자의 요구에 의해 설정되며 암시적 저장점(implicit savepoint)은 시스템 운영 목적에 따라 트랜잭션 관리기에 의해 설정된다.

3.2 회복 관리기의 설계

ALTIBASE™ 시스템의 회복 관리기는 각종 고장에 대한 회복을 위해 시스템의 정상 수행중의 모든 데이터베이스 변경 연산에 대하여 로그를 기록하며, 이를 활용하여 올바른 데이터베이스 상태로 복구하는 기능을 제공한다.

ALTIBASE™ 시스템에서는 로그 파일이 적재된 메모리 포인터만을 이용하여 로그 관련 I/O 작업을 수행하게 된다.

주기억 상주 데이터베이스 시스템은 로그 파일이 메모리에 위치하므로 모든 레코드들을 휘발성이 강한 저장장치에 보관할 수 밖에 없다. 예기치 못한 시스템 고장시 회복이 불가능한 상황을 유발할 수 있으므로, 비록 주기억 상주 데이터베이스 시스템이라 할지라도 메모리내의 로그화일을 주기적으로 영속성을 지닌 디스크에 반영시켜야만 한다[6].

ALTIBASE™시스템은 퍼지검사점(fuzzy checkpoint)과 핑퐁 검사점(ping-pong checkpoint)를 함께 제공하여 두 가지 백업 디스크를 교체적으로 번갈아 사용하여 현재 수행중인 트랜잭션에 부하를 주지 않고 수행할 수 있도록 설계하였다. ALTIBASE™ 시스템의 회복 기법에서는 분석 단계에서 갖는 시간과 비용을 절감하기 위하여 재수행 단계와 취소 단계 만으로 회복 작업을 수행한다.

ALTIBASE™ 시스템은 로그 버퍼를 주저장소인 메모리 내에 적재하여 사용하지만, 로그 버퍼의 내용은 회복시 필수적인 내용이므로 디스크의 동기화를 필수적으로 수행하여야 한다. 따라서 기본적인 로그 버퍼로서 메모리 맵드 파일(memory mapped file)을 제공하면서, 트랜잭션의 영속성 수준에 따라 메로리 버퍼와 메모리 맵드 파일 두 종류의 로그 버퍼를 선택적으로 사용할 수 있게 하여 관리자의 제어에 따라 조절할 수 있게 구현되었다. 트랜잭션의 영속성은 모두 5가지 수준으로 제공되며 각 수준에 따른 버퍼 및 기능은 <표 1>과 같다.

<표 1> ALTIBASE™ 시스템의 트랜잭션 영속성 수준

트랜잭션 영속성 수준	로그 버퍼 및 디스크 동기화	기능 설명
Level 1	메모리 버퍼	로그는 메모리 버퍼에만 반영
Level 2	메모리 버퍼 디스크 로그 파일 Log Sync Thread 동작	로그는 메모리 버퍼에 반영, Log Sync Thread에 의해 로그 파일에도 주기적으로 반영되나, 트랜잭션의 영속성 보장 안함.
Level 3	메모리 맵드 파일	모든 로그 디스크에 반영, 운영체제 파일 버퍼 적용, 트랜잭션의 영속성 보장.
Level 4	메모리 버퍼 메모리 맵드 파일 Log Sync Thread 동작	로그는 메모리 버퍼에 반영, Log Sync Thread에 의해 메모리 맵드 파일에도 주기적으로 반영된다. 단, 완료 트랜잭션의 영속성을 보장하지 않는다
Level 5	메모리 버퍼 디스크 로그 파일 Log Sync Thread 동작	로그는 메모리 버퍼에 반영되고 Log Sync Thread에 의해 로그 파일에도 주기적으로 반영된다. 트랜잭션의 완료는 완료 로그를 포함한 모든 로그가 디스크 로그 파일에 기록된 후에 선언된다.

또한, 오류에 대비하기 위한 불필요한 로그의 기록 때문에 전체적인 성능 저하 상황에 효과적으로 적용할 수 있도록 로그의 수준을 모두 세가지로 분류하여 트랜잭션의 영속성에 따라 적절히 적용할 수 있도록 구현하였다. 즉 트랜잭션의 영속성 수준이 0인 경우, 모든 로그를 기록하는 것은 불필요한 작업이 되므로 기록하는 로그의 수준에 따라 <표 2>와 같은 로깅 수준을 제공한다.

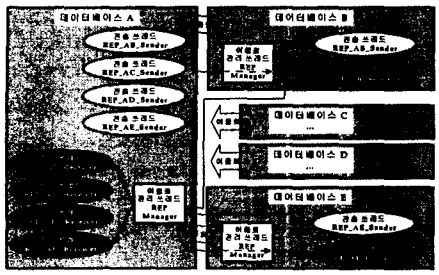
<표 2> ALTIBASE™ 시스템의 로깅 수준

로깅수준	로그 내용	기능 설명
Level 0	로그를 기록하지 않음	변경을 유도 하는 트랜잭션에 대한 회복은 불가능하며, 트랜잭션의 영속성은 마지막 검사점까지만 보장.
Level 1	DML(Insert, Update, Delete) 연산에 대한 로그	변경연산에 대한 취소 연산을 수행하기 위한 로그를 기록하며, 검사점 이후에 완료한 트랜잭션은 데이터베이스에 반영되지 않는다.
Level 2	모든 로그 기록	모든 로그를 기록하며 모든 오류 상황에 대해서 회복이 가능하다

3.3 고가용성을 위한 데이터베이스 이중화 설계

ALTIBASE™ 시스템은 고가용성과 안정성 제공을 위해 데이터베이스 이중화(replication) 기능을 지원한다. ALTIBASE™ 이중화 기능은 최소비용, 신뢰성, 독립성, 다중 복제와 같은 특징을 지닌다.

ALTIBASE™ 시스템의 이중화 기능을 위한 쓰레드 구조는 <그림 3>과 같이 도식화 할 수 있다.



<그림 3> ALTIBASE™ 시스템의 이중화 쓰레드 구조

4. 성능 평가

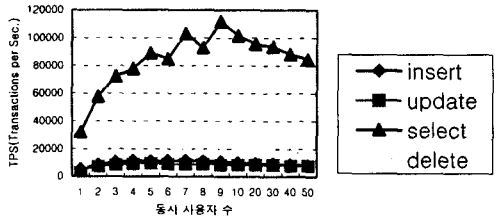
본 절에서는 성능평가를 통해 ALTIBASE™ 시스템이 여러 종류의 시스템 부하에 따른 TPS(Transaction per Second) 값을 집중적으로 평가하여 시간제약적인 응용 분야에 적합한 시스템임을 보인다. 모든 실험의 운영체제는 400MHz CPU 4개와 4G 바이트 메모리를 보유한 " *Sun Enterprise 3500*" 플랫폼과 " *Solaris 2.5.8*" 이며, 실험에 따라 동시 사용자 수 를 단일 사용자부터 50명의 사용자로 레코드의 개수를 총 10,000개부터 500,000개로 구성하였다. 실험에 사용된 트랜잭션은 모두 4종류, 검색, 삽입, 변경 그리고 삭제 트랜잭션이며, 대상 테이블은 " *number*" , " *real*" , " *varchar*" 등의 여러 가지 속성들로 구성되는 총 20개의 속성을 갖는 단일 테이블로써, ALTIBASE™ 시스템에서 제공하는 저장 프로시듀어 인터페이스를 사용하여 구현하였다. 또한, 모든 속성에 대해 검색 트랜잭션 및 변경 트랜잭션의 모든 작업을 수행 수행하도록 구성하였다. 모든 트랜잭션의 조건식은 인덱스를 갖는 속성에 대해 조건식을 작성하였다. 실험에 사용된 트랜잭션 영속성은 4, 로깅 레벨은 2로서 가장 일반적인 응용 환경과 동일한 조건에서 수행되었다. <표 3>에서 측정한 결과가 다른 상용 주기억 상주 데이터베이스 시스템 보다 우수함을 알 수 있다.

<표 3> 단일 사용자 환경에서의 TPS 측정값

삽입 트랜잭션	변경 트랜잭션	검색 트랜잭션	삭제 트랜잭션
6,134.97	4,405.29	29,411.76	12,345.68

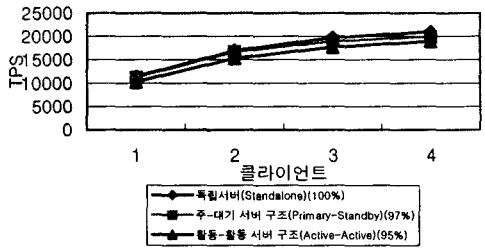
단위 : TPS(Transactions Per Second)

<그림 5>에서 동시 사용자 수가 10명 이상이 되는 지점에서부터 CPU 처리능력이 부족하여 TPS 수치가 점차 약간 감소하거나 일정 수준을 유지함을 알 수 있다. 이 결과를 통해 ALTIBASE™의 성능 확장성과 시스템 가용성을 보장함을 알 수 있다.



<그림 5> 동시 사용자 수의 변화에 따른 TPS

<그림 6>에서 모두 활동 서버로 서비스하는 이중화 구조에서도 트랜잭션 처리율의 95%이상의 성능을 보임을 알 수 있다.



<그림 6> 데이터베이스 이중화 기능에 따른 TPS

실험을 통해 측정된 결과 다른 상용 주기억 상주 DBMS 들과 유사하거나 다소 우세한 실험 결과임을 알 수 있으며, 데이터베이스 이중화 기능 또한 부가적인 오버헤드 없이 지원가능 함을 알 수 있다. 따라서 본 성능 평가를 통하여 ALTIBASE™ 시스템이 시간제약 사항을 지닌 실시간 응용 분야나 통신 산업 분야의 응용 업무에 적용 가능함을 알 수 있다.

5. 결론

본 논문에서는 주기억 상주 데이터베이스 시스템인 ALTIBASE™ 시스템에 대한 설계에 대한 고려사항 및 자료저장 관리기의 세부 구성요소와 구조적 특징과 기능에 대하여 설명하였다.

현재 ALTIBASE™ 3.0이 상용 시스템으로 발표되었으며, 이 시스템은 성능 및 안정성 그리고 고가용성을 요구하는 여러 응용 분야에 범용적으로 활용되고 있다. ALTIBASE™의 향후 연구 및 개발 방향은 대용량 데이터베이스관리의 한계 극복을 위한 디스크 기반 데이터베이스 시스템과 주기억 상주 데이터베이스 시스템을 혼합하여 사용할 수 있는 다중 저장장치 데이터베이스 시스템을 연구하는 것이다.

참고문헌

[1] D. Agrawal and V. Krishnaswamy, " Using Multiversion Data for Non-Interfering Execution of Wirte-Only Transactions" , Proc. of the ACM SIGMOD International Conference on Management of Data, 1991.

[2] P. M. Bober and M. J. Carey, " Multiversion Query Locking" , Proc. of the 18th Conference on Very Large Database, 1992.

[3] P. Bohannon, D. F. Lieuwen, R. Rastogi, A. Silberschatz, S. Seshadri, and S. Sudarshan, " The Architecture of the Dali Main-Memory Storage Manager" , Multimedia Tools and Applications, 4(2), 1997.

[4] P. Bohannon, J. Parker, R. Rastogi, S. Seshadri, A. Silberschatz, and S. Sudarshan, " Distributed Multi-Level Recovery in Main-Memory Databases" , Proc. of the International Conference on Parallel and Distributed Information Systems, 1996.

[5] H. Garcia-Molina and K. Salem, " Main Memory Database Systems : An Overview" , IEEE Transactions on Knowledge and Data Engineering, 4(6), 1993.

[6] H. V. Jagadish, A. Silberschatz, and S. Sudarshan, " Recovering Main Memory Lapses" , Proc. of the 19th Conference on Very Large Databases, 1993.

[7] K. Ramamritham and P. K. Chrysanthis, " A Taxonomy of Correctness Criteria in Database Applications" , VLDB Journal, 5(1), 1996.

[8] R. Rastogi, S. Seshadri, P. Bohannon, D. W. Leinbaugh, A. Silberschatz, and S. Sudarshan, " Logical and Physical Versioning in Main Memory Databases" , Proc. of the 23rd International Conference on Very Large Databases, 1997.