제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

주기억장치 데이터베이스 중복 시스템 설계

최정현*, 최우영*, 진성일*, 염태진**
*충남대학교 컴퓨터과학과
**충남대학교 소프트웨어연구센터
e-mail : {leonardo, cutenc, sijin}@cs.cnu.ac.kr, tjyeom@sorec.cnu.ac.kr

Design of Main Memory Database Replication System

Jung-Hyun Choi*, Woo-Young Choi*, Seong-Il Jin*, Tai-Jin Yeom
*Dept. of Computer Science, Chungnam National University
**Software Research Center at Chungnam National University

요 약

본 논문은 많은 종류의 인터넷 정보시스템에서 데이터의 고속 검색과 저장 및 처리를 지원하는 주기억장치 데이터베이스 시스템을 중복하여 관리할 수 있는 중복 주기억장치 데이터베이스 시스템의 요구사항에 대해 알아본다. 아울러, 자료의 고속처리라는 측면을 주로 고려하여 중복되어 관리되는 주기억장치 데이터베이스 시스템의 아키텍쳐 및 트랜잭션 수행구조를 설계한다.

1. 서론

최근들어 이동통신을 위한 PCS(Personal Communuication System)에서의 HLR(Home Register Location), 이동컴퓨팅(Mobile Computing), 공장제어를 위한 CIM(Computer Integrated Manufacturing), 공장자동화(Factory Automation) 등 신속한 처리를 요구하는 응용분야가 점차 확대되고 있다. 현재 널리 사용되고 있는 디스크 기반 데이터베이스 시스템(Disk-Resident Database Systems: DRDBS)에서는 데이터 처리를 위한 디스크 입출력 시간 때문에 빠른 처리를 요구하는 응용분야에는 적합하지 않다. 그러나, 기억장치의 용량이 커지고 가격이 하락함에 따라서 대용량 메모리의 사용이 가능해졌다. 따라서 이러한 빠른 처리를 요구하는 응용을 효율적으로 지원하기 위해서는 주기억장치 데이터베이스 시스템(Main Memory Database Systems:MMDBS)이 필요하다.[1]

이와 같이 데이터의 고속 처리를 위해 쓰여지는 주기억장치 데이터베이스 시스템의 최근 추세는 응용의 대용량화로 인한 중복화, 민감한 데이터의 처리를 위한 고장대처(fault-tolerant) 능력을 요구한다. 따라서, 트랜잭션 처리율 향상을 위한 중복 데이터베이스 시스템에서의 중복데이터 관리 기술, 관련된 동시성 처리, 데이터 불일치 발생시의 충돌해결 등에 관한 연구가 필요하다.[4]

본 논문에서는 이러한 중복 데이터 관리를 주기억장치 데이터베이스 시스템에 적용하여 이것을 운영하기 위한 시스템 아키텍쳐, 중복 사이트에서의 트랜잭션 수행 구조, 고장 발생시의 비정상 동작 수행 방안 및 회복 구조 등에 관한 모듈을 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2 장에서는 중복 데이터베이스 시스템의 일반적인 개요 및 합리적인 설계방안을 검토해 본다. 3 장에서는 RLog(Replication Log)라는 중복 데이터베이스에서의 트랜잭션 수행을 위한 새로운 매커니즘을 제시한다. 4 장에서는 주기억장치 데이터베이스인 Kairos 에 기반하여 주기억장치 중복 데이터베이스 시스템의 전체적인 운영 구조를 설계한다. 5 장에서는 결론을 맺고 향후의 연구 방향에 대해 언급한다.

2. 중복 데이터베이스 시스템 개요

중복 데이터베이스에서는 일반적인 경우 데이터베이스 운영자(또는 사용자)의 명시적인 선언에 의해 두 개 이상의 데이터베이스 내에 존재하는 동일한 테이블간의 중복화 설정하여 동시에 트랜잭션을 수행할 수 있는 형태로 동작하게 된다. 만약 각 서버중 어느

※ 이 연구는 BK21 충남대학교 정보통신인력양성사업단의 지원을 받아 수행되었음.

하나에서 문제가 생겼거나 사용자에게 정상적으로 서비스할 수 없는 상황이 발생했을 경우 그 상황에 맞게 절체(Takeover)가 이루어져 서비스의 중단 없이 동작할 수 있다.

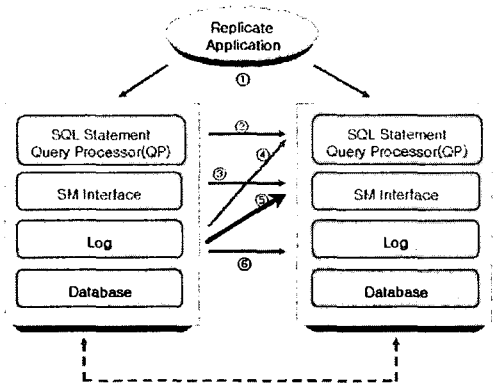

그림 1. 데이터베이스 중복화 방법 [3]

중복 데이터베이스를 구현하기 위해 [그림 1]과 같은 6 가지 방법을 생각해 볼 수 있다. ①은 응용 프로그램 단계에서 중복화를 처리하는 것으로써, 중복화로 인한 트랜잭션 처리율의 향상을 기대할 수 없으며, 또한 트랜잭션 수행 전후의 중복 데이터에 대한 일치성을 보장할 수도 없다. ②의 방법은 한 사이트로 전달된 질의어 자체를 다른 사이트로 직접 전송하여 데이터 중복을 실행하는 것으로써, 전송되는 데이터의 양을 줄일수 있는 장점이 있다. 하지만, 보통 이러한 방법에서 전송되는 데이터가 SQL 문장임을 감안하면 SQL 문의 처리를 중복하여 수행하게 됨으로써 시스템 전체의 성능을 저하시켜 부하 분산의 효과를 반감시키는 단점이 있다. ③의 방법은 SQL 문장의 처리 결과(이하 실행계획)을 전송하여 중복화 사이트에서 수행하는 방법으로써 전송량 증가로 인한 통신 부하가 가중된다는 단점이 있다. ④의 방법은 상대적으로 전송량이 작은 로그를 이용하여 SQL 문장의 형태로 재정의하여 전송하는 방법으로 마찬가지 맥락으로 치환 및 질의 처리 비용이 증가한다는 단점이 있다. ⑥의 경우 로그를 직접 전송하여 상대편 서버에서 REDO 혹은 UNDO 연산을 수행하여 중복하고자 하는 데이터의 이미지를 일치시키는 방법이다. 이 방법은 전송량 등으로 인한 통신부하는 아주 작으나 A-S 방법에만 적용할 수 있다는 단점이 있다.[2]

따라서, 중복 주기억장치 데이터베이스 시스템 구현을 위해서는 SQL 문장의 질의 처리 비용과 중복화하고 있는 상대편 서버로 전송되는 자료의 양을 최소화할 수 있는 방법인 ⑤를 채택하는 것이 가장 합리적이라고 할 수 있다. 즉, 중복되어 있는 데이터의 일치성을 보장하기 위해 중복이 설정되어 있는 각각의 사이트에서 처리되는 사용자 트랜잭션을 수행하고, 그 과정에서 발생되는 실행계획(질의어 처리를 수행한 후 발생하는 트랜잭션에 관한 각종 정보)을 참조하여 일종의 로그와 같은 형태로 정의하여 중복화하고 있는 상대 서버에 전송한 후 해당 사이트에서 수행중인 사용자 트랜잭션과 같은 형태로 실행하게 함으로써 데이터의 일치성을 보장할 수 있다. 또한 이 방법은 통신 네트워크를 이용한 데이터베이스시스템에서 가장 문제가 되는 통신 비용면에 있어 상대적으로 작으므로 전체적인 측면에서 성능향상을 기대할 수 있다.

3. RLog(Replication Log)

본 논문에서는 중복 데이터베이스 시스템의 구현을 위해 주기억장치 상주형 데이터베이스 시스템인 Kairos 를 사용한다. Kairos 중복화서는 중복된 데이터의 일치성 유지를 위한 트랜잭션 수행 매커니즘으로 RLog(Replication Log)라는 새로운 개념을 사용한다. RLog 는 일반적인 개념의 데이터베이스 로그와는 상이한 구조를 가지고 있다.

중복되지 않는 Kairos 시스템에서 사용되는 데이터베이스 로그는 로그 레코드의 종류, 해당되는 트랜잭션 ID, 데이터베이스 이미지의 실제 위치(OID), 등을 명시하는 로그 헤더와 변경전후의 데이터 이미지인 실제 로그 레코드로 이루어져 있다. 이와 같은 데이터베이스 로그를 사용하여 중복 데이터베이스의 이미지를 일치시키는 방법을 사용하면, 중복시에 상대편 사이트로 전송되는 데이터의 양을 줄여 통신비용을 절감할 수 있고, 기존 시스템의 구성요소를 변경하는 것 없이 중복화 구현할 수 있는 장점이 있다. 하지만, 변경된 이미지를 반영하는 방식이 로그 레코드를 가지고 REDO 나 UNDO 연산을 수행하는 것이므로 이에 따르는 부정적인 영향(전송된 상대편 사이트의 지역 트랜잭션과의 충돌이 발생 가능성)을 일으킬 수 있다.

따라서, 중복된 데이터에 대한 트랜잭션 수행 결과를 상대편 사이트에 존재하는 데이터에도 트랜잭션의 형태로 반영해야 데이터베이스의 일관성을 유지할 수 있다. 이를 위해 2 에서 설명한 것과 같은 이유에 근거하여 통신 비용과 질의어 재수행 비용을 고려한 새로운 형태의 중복 트랜잭션(Transaction for Replication)에 관한 매커니즘을 정의한 것이 RLog(Replication Log)이다.

RLog 는 사용자가 발생시킨 트랜잭션 중 쓰기연산과 관련된 것들을 중복된 사이트에서 재수행하기에 적당한 구조로 실행계획을 재정의한 것이다. 이것을 전송함으로써 중복화의 상대편 사이트는 질의어 처리의 오버헤드를 줄일 수 있으므로 전체적인 성능향상을 가져올 수 있다. 또한 실행계획 중 트랜잭션의 재수행에 필요한 최소한의 정보만을 가려내어 구조화하여 상대편 서버에 전송함으로써 통신에 소모되는 비용을 줄일 수 있도록 한다.

RLog 는 크게 header 와 body 로 이루어진다. RLog header 의 구성요소는 다음의 [표 1]과 같다.

Attribute	Type	Description
type	Short	Log type - insert, delete, deleteAll, Update
rlog_id	Int	RLog ID - 순차적 증가값
length	long	The size of body in Log Record
site	binary	지역 서버의 위치

표 1. RLog 의 header

RLog 의 body 부분은 트랜잭션의 종류에 따라 다음과 같이 4 가지로 구분한다.

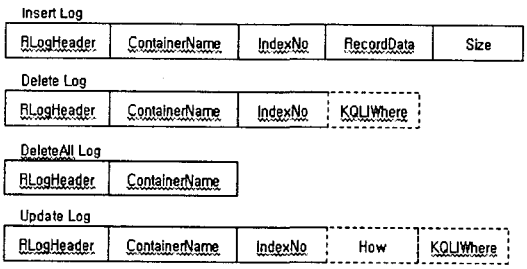

그림 2. RLog 의 body

중복된 데이터가 존재하는 사이트에서 수행되는 데이터에 대한 트랜잭션은 쓰기 연산과 관련되어 있는 것만 수행하면 된다. 따라서, Kairos 에서의 트랜잭션 수행의 주체라고 할 수 있는 KQLIBase 클래스에 메소드로 정의되어 있는 것들 중 쓰기연산에 관련 있는 것들만 선택하여 [그림 2]와 같은 형태의 RLog 구조를 정의하였다.

4. 중복 데이터베이스 시스템 설계

4.1 시스템 내부 아키텍처

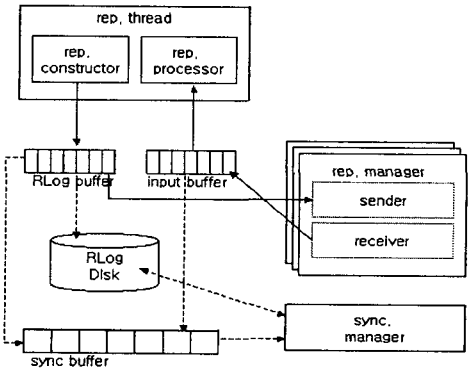

그림 3. 중복 데이터베이스 내부 모듈

Kairos 를 위한 중복 데이터베이스 시스템 아키텍쳐는 위의 [그림 3]과 같다. 중복화 트랜잭션 수행의 핵심 메커니즘인 RLog 를 생성하는 RLog Constructor 와 전송받은 RLog 를 수행하는 Replication Processor 가 쓰레드 형태로 생성되어 기존의 Kairos 시스템에 추가되어 핵심 커널을 형성한다. 이외에 n-way 중복화를 지원하기 위한 Replication Manager 및 RLog 의 전송과 오류 발생시의 동기화를 위한 내부구조인 RLog buffer, sync buffer, input buffer, RLog disk, sync manager 등의 모듈로 구성된다. 각각에 대한 자세한 설명은 다음과 같다.

- Replication Manager
  1:N 중복화를 지원하기 위해 중복하려는 사이트마다 1 개씩 생성되는 것으로써, 중복된 데이터베이스가 존재하는 사이트와의 자료 송수신, 오류 탐지 및 해결을 위한 동기화를 담당한다.
- Sync Mgr
  네트워크 오류나 시스템 오류 등에 의해 중복화가 단절된후 회복될 때 데이터 동기화를 실행하는 모듈로써 오류 발생 후 회복시에 마스터 사이트가 상대편 서버로 RLog 를 전달한다.
- Storage Manager
  중복 트랜잭션을 위한 RLog 생성시에 주요 참조 정보가 되는 SQL 문장의 파싱 및 분석결과를 제공하는 모듈로서 Kairos 의 내부에 이미 존재하고 있는 모듈이다. 여기에 RLog Constructor 와 Replication Processor 모듈을 추가하여 새로운 중복 트랜잭션을 수행할 수 있도록 한다.
- RLog Constructor
  사용자로부터 전달된 SQL 문장을 파싱하고 분석한 후 생성된 실행계획과 관련된 매개변수를 RLog 의 형태로 변환하는 일을 담당한다.
- Replication Processor
  다수의 중복 데이터베이스 사이트에서 전달된 RLog 를 input buffer 로 받아들여 트랜잭션의 형태로 실행하는 역할을 담당한다.
- RLog Disk : 오류 발생시 문제 해결(동기화)을 위해 RLog 를 저장하는 파일 시스템에서의 논리적 영역으로 시스템 마다 1 개씩 존재
- Sync buffer : 오류 발생 후 회복시에 RLog 를 전송하는 마스터서버에서의 사용자 요구 처리의 효율성을 높이기 위해 RLog buffer 의 RLog 를 임시저장하는 버퍼로 시스템마다 1 개씩 존재

4.2 트랜잭션 수행구조

Kairos 를 위한 중복 데이터베이스 시스템에서 가장 중요하게 생각해야 할 점은 통신 비용을 줄이면서 자신의 사이트에서 수행된 트랜잭션이 중복 사이트에서도 트랜잭션의 형태로 수행될 수 있는가 하는 점이다. 이를 위해 3 에서 설명하고 있는 RLog(Replication Log) 라는 매커니즘을 위해서 전체적인 중복 사이트에서의 트랜잭션 수행구조 또한 정립되어야 한다.

먼저 기존의 Kairos 에서의 트랜잭션 수행과정은 다음과 같다. 1 단계로 Transaction Manager 로부터 transaction id 를 생성받고 트랜잭션 리스트에 등록한다. 2 단계로 입력된 트랜잭션에 대해 parse tree 를 생성한다. 여기까지 완료되면 실제로 트랜잭션이 시작되는

transaction begin 단계와 parsing 된 질의의 분석 후 액션을 취하는데 필요한 오브젝트를 생성하는 분석(analysis)단계, 마지막으로 분석단계에서 생성된 객체를 매개변수로 하여 실제적인 연산을 수행하는 액션(action)단계로 이루어진다.

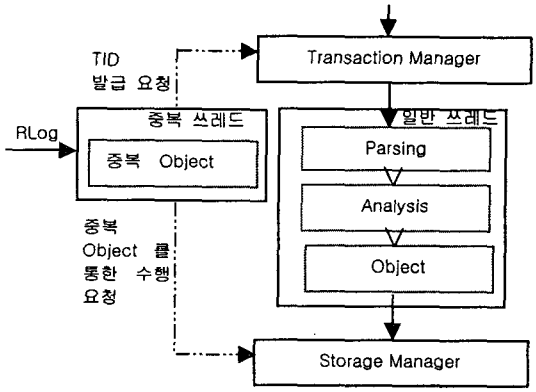

그림 4. 중복 트랜잭션의 수행 구조

Kairos 시스템에서 사용자가 트랜잭션을 생성하여 Transaction Manager 에 전달하고 parsing 과 분석단계를 지날 때까지의 비용은 트랜잭션 수행 비용 전체의 약 75%정도로 추정된다. 이러한 비용을 줄이기 위해 지역 사이트의 사용자 트랜잭션 수행 단계에서 분석이 끝난 질의어를 앞에서 정의한 RLog 형태로 만들어 중복 사이트에 전달하고, 전달받은 사이트는 이것을 트랜잭션 객체로 생성하여 트랜잭션화하고 액션을 취할 수 있도록 하였다. 이것은 대부분의 분석단계 및 Parsing 과정이 생략되므로 전체적인 성능은 많이 향상된다고 할 수 있다. 중복된 데이터베이스 시스템에 전달되어 트랜잭션으로 수행되는 구조는 [그림 4]와 같다.

또한, 본 논문에서 채택하고 있는 지연중복(Lazy Replication) 방식에서의 가장 큰 문제점인 데이터 충돌로 인한 중복 데이터베이스의 비일관성 문제를 해결하기 위해서 운영되는 전체 시스템의 입장에서 다음의 [그림 5]와 같은 구조를 취한다.

기본적인 아이디어는 중복 데이터베이스를 가지고 있는 모든 사이트 중 주서버(Main Server)의 역할을 하는 사이트를 정하고 사용자 연결, 부하 분산 등의 역할을 하도록 한다. 또한, 사용자가 보내는 트랜잭션의 종류를 파악하여 변경(write, update, delete)에 관한 트랜잭션일 경우 자신의 데이터베이스에 반영하고 사용자에게 응답한 후, 다른 모든 사이트에 만들어진 RLog 를 방송(broadcasting)한다. 사용자가 질의한 트랜잭션이 읽기에 관한 것일때에는 적절한 부하 분산 알고리즘을 통해 자신을 포함한 모든 사이트에서 응답하도록 한다.

이러한 제한된 구조를 가지도록 하는 것은 주기억장치 데이터베이스 시스템의 응용분야가 쓰기 연산이 자주 발생되는 응용보다 읽기 연산이 보다 많은 응용에 적합하며, 데이터의 불일치에 매우 민감한 응용을 대상으로 했다. 따라서, 일반적인 경우의 데이터베이스 응용에 포괄적으로 적용될 때에는 적합하지 않을 수도 있다..

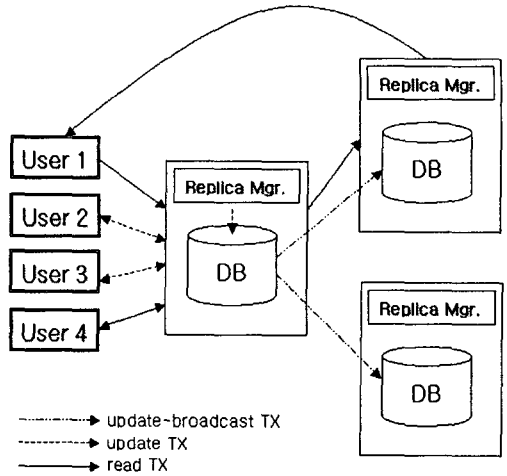

그림 5. 트랜잭션 종류에 따른 수행 구조

5. 결론 및 향후 연구과제

주기억장치 데이터베이스 시스템의 응용분야는 정보시스템의 처리 용량이 많아지고 실시간으로 처리해야하는 요구사항이 발생하면서 갈수록 증가하는 추세에 있다. 따라서, 기존의 단일 데이터베이스 시스템의 주기억장치 상주형 모델로는 이러한 요구에 한계점이 있으므로 중복 데이터베이스 시스템을 지원해야만 한다.

본 논문에서 제시한 RLog 매커니즘 및 중복 데이터베이스 시스템의 전체적인 구성은 앞에서 언급한 것과 같은 요구사항을 만족시키는 동시에 기존의 중복 데이터베이스 시스템에서 진행되었던 연구와 구현 사례보다 응용 시스템에서의 활용도를 중점적으로 고려하여 설계하였다.

따라서, 앞으로의 연구 방향은 확장된 의미에서의 주기억장치 중복 데이터베이스 시스템의 구조에 관해 연구하고, 필연적으로 발생하게 될 데이터 충돌을 효과적으로 해결하는 동기화 기법을 고안하여 적용하는데 목표를 두고 진행할 예정이다.

참고문헌

[1] 송은미, "주기억장치 상주형 DBMS 에서 그림자 갱신을 이용한 회복기법", 석사학위논문, 충남대학교, 2000.

[2] "Altibase 이중화 기술 소개 및 시연", http://www.altibase.com, 2002.

[3] "실시간 이중화를 활용한 응용 구축", 프로그램세계, 2002.

[4] J.Gray, P.Helland, P. O'Neil, and D. Shasha. The Dangers of Replication and a Solution. In Proceedings of the ACM SIGMOD Conference, 1996