

컨텐츠 메타데이터 관리시스템의 이중화 관리기 설계

진기성^o, 이미영, 김준
한국전자통신연구원
{ksjin^o, mylee, jkim}@etri.re.kr

Design of Replication Manager for Content Metadata Management System

Ki Sung Jin^o, Mi Young Lee, June Kim
Electronics and Telecommunications Research Institute

요 약

데이터베이스 이중화는 데이터베이스의 복사본을 또 다른 서버로 복사하여 단일 서버에 집중되는 부하를 분산할 뿐만 아니라, 서버의 고장이 발생하더라도 정상 동작중인 다른 서버에서도 서비스가 가능하도록 하여 고성능 및 고가용성을 제공하기 위한 기술이다. 본 논문에서는 이러한 데이터베이스 이중화 기술을 CMM(Content Metadata Management)시스템에 적용하기 위한 방법에 대하여 기술한다. 제안하는 이중화 모델은 다중 위상구조의 환경에서 동작 가능하며 XML 문서와 같은 멀티미디어 객체에 대해서 복제를 지원한다. 또한 변경 로그를 이용하여 데이터베이스의 변경 사항을 복제 서버로 반영함으로써 데이터베이스의 동기화를 보장한다.

1. 서론

최근 몇 년간 인터넷의 광범위한 확산과 함께 인터넷을 이용한 서비스에 대한 요구가 계속해서 기하 급수적으로 증가하고 있는 추세이다. 이러한 인터넷 서비스는 시간적, 공간적 제약을 받지 않고 사용자의 거주지 또는 다양한 생활공간에서 요구되는 특성을 가지고 있다. 그러나 사용자가 요청을 할 때마다 집중된 중앙 서버에서 모든 요청을 처리하는 구조에서는 시스템 성능 감소를 초래할 뿐만 아니라, 결과적으로 사용자에 대한 서비스의 질적 저하를 가져오게 되어 다양한 사용자의 요구를 충족하기에 적합하지 않다.

이를 해결하기 위해 최근에 다수의 서버를 논리적으로 결합하여 서비스를 제공함으로써 고가용성을 제공하는 클러스터링 기술[1]이 연구되고 있다. 그러나 이 방법의 경우 클러스터링 환경을 구축하기 위해 고가의 장비를 요구할 뿐만 아니라 지역적인 제약사항으로 인해 단일점 오류(Single Point Failure)의 한계를 가지고 있어 다양한 형태의 인터넷 서비스에 적용하기 어려운 점이 있다.

본 논문에서는 이러한 문제를 해결하기 위해 데이터베이스 이중화 기법[2][3]을 적용한다. 데이터베이스 이중화는 중앙 서버에서 관리하는 데이터의 일부를 각 지역의 원격 서버들에 복사하여두고, 사용자의 요청이 발생할 경우 중앙 서버를 거치지 않고 각 원격 서버에서 독립적으로 처리하는 방식이다. 이렇게 함으로써 중앙 서버로 집중되던 사용자의 요청을 각 지역별로 분산하여 보다 빠른 서비스를 제공하고, 계획되지 않은 고장에 의해 중앙 서버가 다운이 되더라도 각 원격 서버에 저장된 데이터베이스를 이용하여 지속적인 서비스를 함으로써 중단 없는 서비스를 제공할 수 있다.

또한, 본 논문에서는 제안하는 이중화 기법을 기존에

개발된 컨텐츠 메타데이터 관리시스템인 CMM 시스템에 적용한다. CMM 시스템은 각종 멀티미디어 데이터들이 가지는 메타데이터 정보를 관리하기 위해 개발된 시스템으로서, 동영상, 오디오, 이미지와 같은 다양한 멀티미디어 컨텐츠들의 메타데이터를 효율적으로 저장 및 관리할 수 있는 시스템이다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구에 대해 간략히 살펴보고, 3 장에서는 본 논문에서 제시하는 이중화 관리기의 기본 프레임워크인 CMM 시스템의 특성과 구조를 기술한다. 4 장에서는 제안하는 데이터베이스 이중화 모델에 대해 설명하고, 5 장에서 결론을 맺는다.

2. 관련 연구

한 서버에 집중된 데이터를 다수의 서버로 분산하여 고가용성 및 고성능을 얻기 위한 데이터베이스 이중화 기술은 초기에 서버간 중복 데이터 기법을 통해 가용성을 높이는 데 초점이 맞추어져 있었으며, 중복을 허용하는 데이터의 타입이 숫자 또는 문자열 스트림 등으로 제한적이었다. 그러나, 인터넷의 발달과 함께 서비스의 형태가 다양해짐에 따라 데이터베이스 이중화에 대한 기술적 요구사항 또한 증가하고 있다. 이러한 요구사항으로는 다중 서버 이중화를 통해 서비스 운영의 효율성을 높이고, XML 데이터 타입과 같이 구조 및 속성 정보를 가진 데이터들에 대한 이중화 지원함으로써 다양한 멀티미디어 서비스를 가능하게 하는 방법 등을 들 수 있다.

Oracle 9i Advanced Replication[4]에서는 주어진 시점에 원본 서버의 테이블을 다수의 복제 서버에 카피하여 중복된 데이터를 처리하는 방법을 제공한다. 원본 서버에서 변경된 데이터는 Trigger 또는 Stored Procedure 에 의해 복제 서버에 전송되어 변경된 데이터의 동기화를 유지하며, 주어진 시점에 하나의 서버에서만 변경이 가능한 Dynamic

Ownership 과 다중 서버에서 동시에 변경이 가능한 Shared Ownership 을 제공한다. 한편, IBM DB2[5]에서는 데이터베이스 이중화를 위해 Controlled Redundancy Solution 을 제공한다. 원본 서버에서 변경이 발생하면 Capture 프로세스가 로그정보를 참조하여 CD(Changed Data) 테이블과 CCD(Consistent Changed Data) 테이블을 생성해 내며, DB2 는 이 데이터를 수집하여 복제 서버에 전송하고, 복제 서버에서는 CD/CCD 테이블을 주기적으로 접근하여 변경된 정보를 감지한다. 그러나 Oracle 9i 와 IBM DB2 의 경우 다중 서버 이중화 구조를 지원하지 않고 있으며, XML 데이터에 대한 이중화를 지원하지 않아 다양한 사용자의 응용에 적용하기에 어려운 문제점이 있다.

3. CMM 시스템 특성 및 구조

본 장에서는 제안하고자 하는 이중화 구조의 기본 프레임워크인 CMM 시스템[6]에 대하여 설명한다.

3.1 CMM 주요 특징

CMM 은 이미지 데이터, 전문 정보 및 XML 문서와 같은 다양한 멀티미디어 문서를 효율적으로 저장, 검색, 관리할 수 있는 객체 관계형 데이터베이스 관리 시스템이다.

CMM 은 객체 지향 자료 모델에 따라 객체 식별자, 클래스 및 상속, 그리고 멀티미디어 자료 관리를 위한 데이터 타입으로 BLOB, CLOB, Collection, Vector, Image, XML 등을 지원하며, 사용자 인터페이스를 위해 ODBC, JDBC, C++ binding 을 제공한다. 또한, 트랜잭션 처리 기능으로 ACID 특성을 만족시키기 위해 동시성 제어 및 회복 기능을 제공하며, 신속한 자료 접근을 위해 B+-인덱스 및 멀티미디어 데이터인 전문 정보 검색을 위한 역 파일 인덱스와 이미지 정보 검색을 위한 CIR(Content based Image Retrieval) 다차원 인덱스를 지원한다.

3.2 CMM 구조

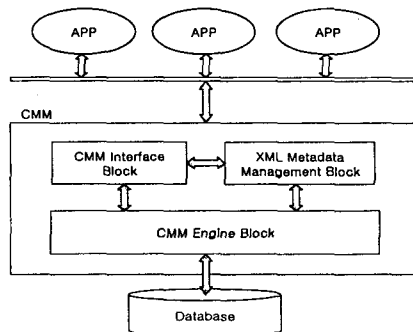


그림 1. CMM 시스템 구조

CMM 은 그림 1 과 같은 구조로 이루어져 있으며 엔진 내부에서 다중 쓰레드로 동작하여 클라이언트의 요청에 대해 하나 이상의 쓰레드들이 동시에 처리한다. Interface 블록에서는 클라이언트들이 데이터베이스에 접근하기 위한 API(ODBC, JDBC, C++ binding)를 제공하고, XML 블록에서는 XML 문서를 효율적으로 저장 및 관리하기 위해 DOM 인터페이스와 XPath 를 이용한 구조검색을 지원한다. 엔진블록은 질의 처리 및 트랜잭션 관리와 같은 데이터베이스 관리의 핵심 기능을 제공한다.

4. CMM 에서의 이중화 관리기 설계

본 장에서는 CMM 시스템을 기반으로 고가용성 및 고성능을 제공하기 위한 데이터베이스 이중화 관리기를 제안한다. 제안하는 이중화 모델은 CMM 시스템에 밀결합(tightly-coupled)되어 기존의 객체 관계형 데이터베이스 특징들을 지원할 수 있도록 설계한다.

4.1 주요 특징

본 논문에서 제안하는 데이터베이스 이중화 관리기의 주요 특징은 아래와 같다.

첫째, 다중 서버 데이터베이스 이중화를 지원한다. 다중 서버 데이터베이스 이중화는 그림 2 에서와 같이 1:1(one primary-one slave), 1:N(one primary-multi slave), M:N(multi primary-multi slave) 모델을 지원한다. 1:1 구조는 두 개의 서버간 이중화 관계가 이루어지는 기본 모델로서 원본 서버의 테이블 TA 가 원격 서버의 테이블 TA' 로 복제되는 모습을 보여준다. 1:N 구조에서는 원본 서버의 테이블을 다수의 원격 서버로 복제가 가능한 구조로서 사용자 요청을 다수의 서버로 분산하여 응답시간을 줄이기 위한 응용에 사용될 수 있다. M:N 구조는 이전 모델들을 개선한 구조로서 동일한 스키마 구조를 가진 두 개 이상의 원본 서버로부터 데이터를 복제 받을 수 있는 특징을 가진다.

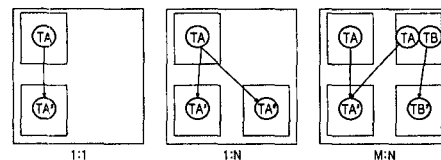


그림 2. 이중화 위상 구조

둘째, 변경 로그에 기반한 데이터복제를 수행한다. 원본 서버에서 데이터베이스의 변경이 발생하면 변경된 정보가 별도의 장소에 로깅 되고, 일정한 규칙에 의해 원격 서버에 전송되어 변경된 데이터의 동기화를 보장한다. 또한 반영이 된 최신 로그의 정보를 시스템 관리 영역에 유지함으로써 데이터 불일치, 중복 등의 오류를 방지한다. 변경 로그에는 자료저장기 수준의 인터페이스가 수행된 내용을 가지고 있다. 사용자가 요청한 SQL 수준의 모든 변경 연산이 데이터베이스에 반영되기 위해서는 최종적으로 자료저장 관리기의 인터페이스를 호출하게 된다. 따라서, 주 서버에서 변경이 이루어진 자료저장기 연산들을 트랜잭션의 수행 순서에 따라 원격 서버에서 재수행 하여 변경된 데이터에 대해 동일한 결과를 보장한다. 이러한 방식의 경우 원격 서버에서 변경 내용을 반영하기 위해 SQL 구문 분석 및 정당성 검사와 같은 절차를 거치지 않아 빠른 이중화의 장점이 있다.

셋째, 데이터베이스 일반 데이터 타입 및 멀티미디어 서비스를 위한 XML 데이터 타입에 대해서 복제를 지원한다. 특히 차세대 문서 표준으로 주목 받고 있는 XML 문서의 경우 대용량, 다 계층화된 구조적인 특징을 가지고 있다. 따라서 이러한 문서들에 대해 구조 정보가 손실되지 않도록 데이터 복제를 지원한다.

4.2 이중화 관리기 구조

데이터베이스 이중화 관리기는 CMM 엔진블록의 내부 Unit 으로서 동작하며, 그림 3 과 같이 세부 구성요소로 Service 관리기, Object 관리기, Log 관리기, Communication 관리기로 이루어진다.

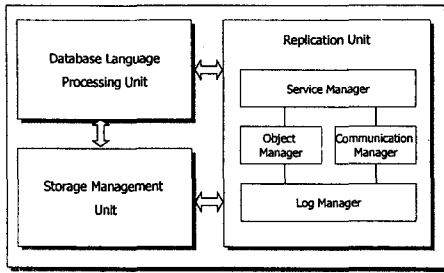


그림 3. 이중화 모듈 구조

Service 관리기는 사용자로부터의 이중화 관련 요청을 네트워크를 통해 수신하고 이를 해석하여 해당 함수를 호출한 후, 수행 결과를 사용자에게 전송하는 기능을 한다. Object 관리기는 이중화 대상 객체들을 관리한다. 이중화 객체 관리 기능은 이중화 객체 등록, 이중화 객체 삭제, 이중화 서비스 시작, 이중화 서비스 중지 등이 있다. 특히, 다중 서버 이중화를 지원하기 위해서는 각각의 이중화 객체들이 속한 서버들의 정보를 필수적으로 유지함으로써, 복제 시 데이터 충돌을 방지한다. 이중화 객체는 사용자 유틸리티에 의해 정의되며 테이블 또는 XML 스키마가 될 수 있다. 일반 테이블의 경우는 다수의 레코드들의 집합으로 이루어지기 때문에 레코드의 변경 사항에 대한 추적을 통해 데이터의 복제가 가능하다. 그러나 XML 문서의 경우 내부의 속성 및 구조 정보를 표현하기 위해 다수의 테이블들로 이루어지는 특성이 있다. 따라서 이러한 XML 문서를 원격 서버로 복제하기 위해서는 XML 문서를 표현하기 위한 각각의 테이블들 관계를 함께 고려하여 원격 서버로 전달하기 위한 방법이 필요하다. Log 관리기는 이중화 객체에 대한 변경 사항을 관리한다. 이중화 객체의 변경정보를 관리하기 위한 주된 기능은 변경 정보의 로깅, 로그의 추출, 로그의 재수행 등이다. 마지막으로 Communication 관리기는 주 서버와 원격 서버간 메시지 송수신을 담당한다.

4.3 데이터 복제 알고리즘

원본 데이터베이스에 변경이 이루어지면 원격 서버와의 데이터 동기화를 유지하기 위해 변경이 이루어진 내용을 원격 서버의 데이터베이스에 반영하여야 한다. 따라서, 주 서버에서 데이터베이스에 대한 변경이 발생한 경우 그 내용을 원격 서버에 전송한 후, 주 서버에서 변경한 내용과 동일한 결과를 유지할 수 있도록 적용하는 방법이 필요하다. 이를 위해 CMM 이중화 관리기는 변경 로그를 이용한 이중화를 사용한다. 변경 로그를 이용하여 데이터 복제가 이루어지는 과정은 그림 4 와 같다.

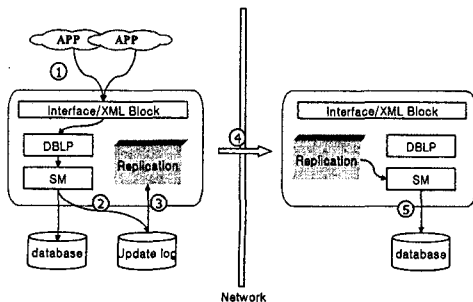


그림 4. 데이터 복제 알고리즘

그림 4 에서는 두 서버간 이중화 관계가 설정되어 현재 구축된 데이터에 대한 동기화가 이루어져 있다고 가정하고, 변경질의가 발생했을 때 원격 서버로 반영하는 모습을 단계별로 보여준다.

첫째, 클라이언트가 주 서버에게 변경 질의(INSERT, UPDATE, DELETE)를 요청하는 단계이다. 변경 질의는 주 서버의 Interface 블록 또는 XML 블록을 통해서 질의처리기와 자료저장기를 호출한다.

둘째, 사용자 요청에 대한 데이터베이스 연산을 수행 후, 변경된 연산의 정보를 변경 로그에 기록하는 단계이다. 이를 위해, 요청된 테이블이 이중화 대상 테이블인지를 검증하기 위한 내부적인 단계가 필요하다. 이중화 대상 테이블이 아니거나, 검색 질의인 경우 로그에 기록하지 않는다.

셋째, 이중화 관리기에 의해 원격 서버로 전송할 로그를 수집하는 단계이다. 이중화 관리기는 특정 시점마다 이중화 로그를 관찰하여 주 서버에서 변경이 되었으나 원격 서버에 반영이 되지 않은 로그를 수집한다. 만일, 반영되지 않은 로그 정보가 있을 경우 전송 버퍼를 구성하여 원격 서버의 이중화 관리자에게 전송한다.

넷째, 수집된 이중화 로그를 원격 서버로 전송하는 단계이다. 로그의 전송은 적당한 패킷 단위로 분리되어 원격 서버의 이중화 관리자에게 전달한다.

다섯째, 주 서버에서 변경이 이루어진 데이터를 원격 서버의 데이터베이스에 반영하는 단계이다. 전송 받은 이중화 로그는 단일 트랜잭션으로 처리되어 데이터 무결성을 보장한다.

5. 결론

본 논문에서는 기존의 객체 관계형 DBMS 인 CMM 시스템을 기반으로 데이터베이스 이중화를 설계하기 위한 방법에 대하여 기술하였다. 제안하는 이중화 방법은 다중 서버 이중화를 지원하여 서버 운영의 활용도를 증가시킬 수 있고, XML 과 같은 멀티미디어 객체에 대해서도 복제를 지원함으로써 멀티미디어 인터넷 응용 서비스의 구축에 효과적으로 활용될 수 있다. 또한 변경 로그에 기반하여 복제를 수행함으로써 이중화 서버간 데이터 동기화를 보장할 수 있도록 하였다.

참고문헌

- [1] Rajkumar Buyya, "High Performance Cluster Computing" : Architectures and Systems, Prentice Hall, 1999.
- [2] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, "Understanding Replication in Databases and Distributed Systems", ICDCS, 2000.
- [3] J. Gray, P. Helland, P. O'Neil, D. Shasha, "The Dangers of Replication and a Solution", SIGMOD, 1996
- [4] Advanced Replication in Oracle 9i : Technical white paper, www.oracle.com
- [5] IBM DB2 Universal Database : Technical white paper, www.ibm.com
- [6] 김영균, 이미영, 김준, 원종호, 김명준, "인터넷 멀티미디어 문서 DBMS : 바다-IV", 데이터베이스 연구회지, 2001