

다중 데이터베이스 미들웨어 시스템 MDATA에서의 웹 클라이언트 세션 관리

서정민^o, 홍영국, 진은숙, 윤수영, 송주원
한국통신 멀티미디어연구소
{jmseo, ykhong, ejin, syoon, jwsong}@kt.co.kr

Session Management of a Web Client in Multi-Database Middleware System MDATA

Jung-Min Seo^o, Young-Kuk Hong, Eun-Sook Jin, Su-Young Yoon, Ju-Won Song
Korea Telecom

요 약

데이터베이스 응용 시스템간에 공유되는 자료가 많아지고 웹 응용 시스템이 일반화됨에 따라서, 웹 상에서도 이기종 데이터베이스 시스템들에 접근하여 질의를 처리하는 방법이 요구되고 있다. 이에 본 논문에서는 이기종 데이터베이스의 자료를 쉽게 처리하는 다중 이기종 데이터베이스 미들웨어 시스템을 웹 환경에서 사용하게 하기 위한 웹 클라이언트를 제안하고, 이 웹 클라이언트가 다수의 전역 데이터베이스에 접근할 수 있도록 여러 사용자에 대한 세션을 관리하는 방법을 제안한다.

1. 서론

데이터베이스 응용 시스템이 방대해지고 각 시스템의 자료 간의 연관성이 복잡해지면서 이기종의 데이터베이스들을 서로 병합하거나 동시에 접근해서 일관성 있게 처리해야 하는 응용이 많아지고 있다[1,2,3,4]. 예를 들어서, 응용 시스템의 초기에는 한 종류의 데이터베이스를 구축하여 운용해 오다가 최근에 다른 시스템으로 확장을 하거나, 다른 시스템에 구축된 데이터베이스를 이용하여 새로운 시스템을 구축해야 하는 경우를 산업계나 금융계에서 쉽게 볼 수 있다. 이러한 요구를 해결하기 위하여 연구되고 있는 다중 데이터베이스 미들웨어 시스템은 이기종의 데이터베이스 시스템들과 일반 사용자의 중간 계층에 위치하여 다른 종류의 데이터베이스 시스템들 한 종류인 것처럼 처리할 수 있게 하는 시스템이다[1,4].

최근에는 웹 기술과 웹 산업이 발전함에 따라서 웹 응용에서도 이기종 데이터베이스들을 관리하고 처리하는 방법이 요구되고 있는데, 본 논문에서는 웹을 통해서 다중 데이터베이스 미들웨어 시스템에 접근하는 방법을 제공하는 웹 클라이언트를 제안한다. 기존의 웹 클라이언트는 한 종류의 데이터베이스만 처리해왔던 반면, 본 논문에서는 이기종의 데이터베이스들을 처리할 수 있도록 사용자의 세션을 관리하는 방법을 제시한다. 웹 클라이언트가 이용하는 HTTP는 세션 정보를 유지하지 않기 때문에, 한번 만들어진 연결을 통하여 계속 질의를 수행하는 일반 데이터베이스 클라이언트

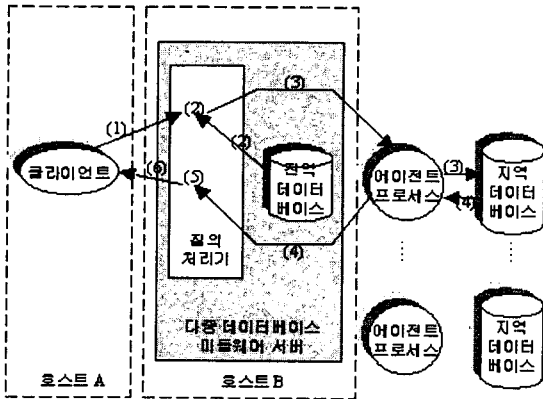
와는 차이가 있다. 그러나, 데이터베이스 응용 시스템의 특성상 이러한 연결은 관리가 되면서 여러 번의 질의를 수행할 수 있어야 한다. 즉, 하나의 세션으로 관리될 수 있어야 한다. 이에 본 논문은 웹 클라이언트에서 세션을 관리하는 방법에 대해서도 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 다중 데이터베이스 미들웨어 시스템에 대하여 설명하고 제 3장에서는 웹 클라이언트에서 세션을 관리하는 방법에 대하여 설명한다. 마지막으로 결론을 내린다.

2. 다중 데이터베이스 미들웨어 시스템

다중 데이터베이스 미들웨어 시스템은 실제로 자료를 저장하지는 않고, 이기종의 데이터베이스 시스템들의 상위 레벨에 위치하면서 그 데이터베이스들로부터 자료를 읽어와서 처리한다. 다중 데이터베이스 미들웨어 시스템 MDATA의 구조와 질의 처리 과정은(그림 1)과 같다.

다중 데이터베이스 미들웨어 시스템은 스키마로만 이루어진 여러 개의 전역 데이터베이스를 가진다. 각 전역 데이터베이스는 실제 자료를 유지하지 않으며, 스키마를 실행할 때마다 여러 지역 데이터베이스 시스템으로부터 질의 결과를 가져온다. 다중 데이터베이스 미들웨어 서버와 지역 데이터베이스 시스템 사이에는 에이전트 프로세스가 있어서 통신 및 지역 데이터베이스 질의 처리를 담당한다. 동시에 다수의 클라이언트가 이 서버에 접속이 가능하며, 또한 한 클라이언트는 여러 전역 데이터베이스에 대한 연결을 동시에 유지할 수 있다.



(그림 1) 다중 데이터베이스 미들웨어 시스템 구조.

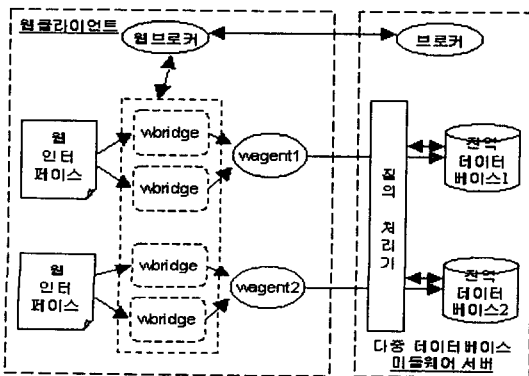
전역 질의문을 처리하는 과정은 다음과 같다. 웹 클라이언트는 공동된 인터페이스를 이용하여 이기종 데이터베이스 시스템들에 대한 전역 질의문을 미들웨어 서버에게 보낸다(과정 1). 전역 질의문은 미들웨어 서버 내부에서 컴파일되어서 여러 개의 지역 질의문으로 변환되고(과정 2), 이 질의문들은 지역 데이터베이스를 관리하는 이기종 데이터베이스 시스템에 전달된다(과정 3). 이기종 데이터베이스시스템의 처리 결과들은 미들웨어 서버에 전달되어(과정 4) 후처리 되고(과정 5), 마지막으로 통합된 질의 결과가 사용자에게 전달된다(과정 6).

3. 웹 클라이언트

웹 클라이언트는(그림 1)에 표시된 클라이언트의 일종으로 웹 브라우저를 통하여 다중 데이터베이스 미들웨어 시스템에 질의를 전달하고 결과를 받는 데이터베이스 클라이언트이다. 웹 클라이언트는 연속된 연결을 제공하지 않는 HTTP 환경에서 연속된 질의를 처리할 수 있도록 세션을 관리하는 기능을 포함하여야 한다. 또한, 스키마를 관리하거나 질의를 처리할 수 있는 인터페이스를 제공해야 한다.

3.1 웹 클라이언트구성

(그림 2)는 웹 클라이언트의 구성 요소를 나타낸다.



(그림 2) 웹 클라이언트 구성.

일반적으로 다중 데이터베이스 미들웨어 시스템과 전역 데이터베이스는 같은 호스트에 위치하고, 웹 클라이언트를 구성하는 요소들도 같은 호스트에 위치하게 된다.

(그림 2)에서 웹 인터페이스는 웹 사용자가 SQL 질의 형식을 모르더라도 데이터베이스를 관리하고 질의를 처리할 수 있게 제공하는 인터페이스이다. wbridge는 웹 인터페이스가 보내는 요청을 처리하는 cgi로서 한번 실행되고 나면 소멸된다. wbridge는 요청의 종류에 따라서 웹브라우저로부터 정보를 가져오거나 wagent에게 질의를 처리하기 위한 다른 요청을 보내기도 한다. wagent는 웹브라우저에 의해 생성되는데 웹 클라이언트와 다중 데이터베이스 미들웨어 서버를 실제로 연결한다. 같은 전역 데이터베이스에 대하여 여러 개의 wagent가 동시에 구동되어 있으면 wagent는 웹브라우저가 소멸시킬 때까지 구동되게 된다. wagent는 wbridge가 보내는 요청에 따라서 전역 데이터베이스로부터 질의 결과를 가져오거나 처리한다. 웹브라우저는 미리 구동되어 있는 데몬으로 다중 데이터베이스 미들웨어 서버의 브로커와 통신을 하여 전역 데이터베이스에 대한 정보를 전달받는다. 또한, wagent의 구동, 관리, 소멸을 담당한다.

3.2 웹 클라이언트 인터페이스

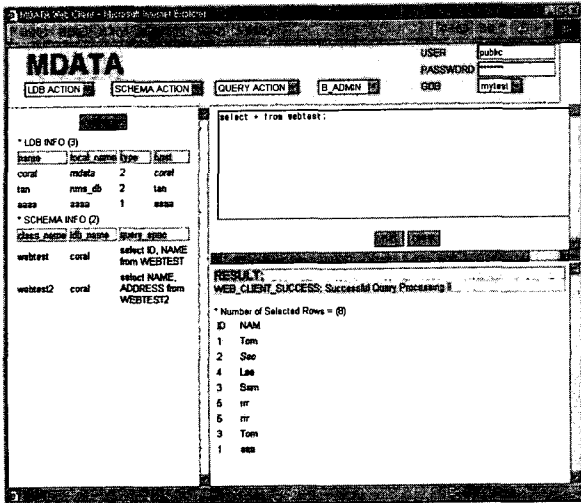
웹 클라이언트가 가지는 기능은 다음과 같다.

- ✓ 다중 데이터베이스 미들웨어 서버 관리 및 운용
 - 다중 데이터베이스 미들웨어의 서버 구동, 다중 데이터베이스 미들웨어의 서버 소멸, 서버/에이전트 프로세스/wagent의 현재 상태 점검
- ✓ 지역 데이터베이스 관리
 - 지역 데이터베이스 등록, 지역 데이터베이스 정보 삭제, 지역 데이터베이스 정보 수정, 등록된 지역 데이터베이스 정보 리스팅
- ✓ 스키마 관리
 - 클래스 생성, 클래스 삭제, 클래스 수정, 클래스 정보 리스팅
- ✓ 질의 처리
 - 질의 실행, 외부 파일로부터 질의 읽기, 외부 파일에 질의문 저장

(그림 3)은 이러한 기능들을 포함하는 웹 클라이언트의 인터페이스를 보여준다. 이 인터페이스는 사용자 ID와 패스워드를 입력받고, 웹브라우저로부터 받은 전역 데이터베이스 이름들을 제공한다. 주 메뉴에는 지역 데이터베이스 관리, 스키마 관리, 질의 관리, 데이터베이스 운용 관리가 있다. 왼쪽 프레임에는 작업을 하는 사용자가 항상 참조할 수 있도록 지역 데이터베이스 정보와 스키마 정보를 제공한다. 오른쪽 프레임은 작업 공간으로서 주메뉴에 따라서 다른 인터페이스를 보여주고 그 결과를 보여준다.

3.3 사용자 세션 관리

HTTP에서는 지속적인 연결을 제공하지 않으므로 매번 데이터베이스 연결을 열어야만 하는데, 이를 해결하기 위해서 데이터베이스와의 연결을 유지하는 프로세스가 구동되어 있어야 한다. wagent는 특정 전역 데이터베이스에 대한 연결을 가지는 cgi이다.



(그림 3) MDATA 웹 인터페이스.

3.3.1 사용자 로그인

(그림 3)의 웹 인터페이스가 처음 구동되면 웹브라우저만 구동되어 있는 상태이다. 처음에 사용자가 사용자 ID, 패스워드, 전역 데이터베이스 이름을 입력하면 웹브라우저는 그 데이터베이스에 대한 연결을 가지는 wagent를 구동하여 사용자 ID에 대하여 전역 데이터베이스에 연결할 수 있도록 한다. 그리고, (전역 데이터베이스, 사용자 ID, wagent ID) 정보를 등록한다. 사용자는 wagent를 통하여 전역 데이터베이스에 로그인이 된다.

3.3.2 사용자 세션 관리

사용자의 세션은 웹브라우저가 관리한다. 웹브라우저는 (전역 데이터베이스, 사용자 ID, wagent ID) 정보를 관리하면서 어느 사용자가 어떤 전역 데이터베이스에 접속해 있는지를 확인한다. 한 사용자는 여러 전역 데이터베이스에 동시에 접속할 수 있기 때문에 (사용자 ID, 전역 데이터베이스) 단위로 관리해야 한다. 각 (사용자 ID, 전역 데이터베이스) 별로 하나 이상의 wagent가 구동될 수 있는데, 웹브라우저는 이러한 정보를 유지하고 wagent의 상태를 파악한다.

wagent는 활동 상태이거나 휴지 상태에 있을 수 있다. wagent는 처음에 웹브라우저로부터 구동이 되고 나면 활동 상태로서 wbridge가 보내는 요청을 처리한다. wbridge는 한 요청만 처리하고 나면 소멸되므로 이 때 wagent와의 연결이 끊어지고 wagent는 휴지 상태로 변환된다. 웹브라우저는 wagent들의 상태가 변경될 때마다 기록을 한다. 웹브라우저는 (사용자 ID, 전역 데이터베이스)에 대한 요청이 오면 휴지 상태의 wagent가 있는지 확인하여 휴지 상태의 wagent가 있다면 이를 활용한다. 반면 모두 활동 상태이면 새로운 wagent를 구동하여 wbridge가 보내는 요청을 처리하도록 한다. 이와 같이 웹브라우저는 (사용자 ID, 전역 데이터베이스) 별로 wagent를 구동시켜서 사용자 세션을 유지한다. 또한, 타임아웃을 제공하여 일정 시간이 지나도

사용되지 않는 wagent는 자동으로 소멸시킨다.

wagent는 항상 auto-commit 모드로 트랜잭션을 관리한다. 이는 여러 개의 다른 웹 인터페이스에서 같은 사용자에 대하여 동시에 같은 전역 데이터베이스에 접근할 수 있기 때문이다.

3.3.3 질의 처리

웹 클라이언트는 일반적인 데이터베이스 언어와 마찬가지로 자료 정의 언어와 자료 처리 언어를 제공한다. 자료 정의 언어에는 지역 데이터베이스를 관리하는 문들과 스키마를 관리하는 문들로 구성된다. 각각의 자료 정의 문들에 해당하는 웹 인터페이스를 제공하고 그 결과를 보여준다. 자료 처리 언어 또한 질의의 종류별로 다른 인터페이스를 제공하고 그 결과를 바로 보여준다.

모든 질의에 대하여 별개의 웹 인터페이스를 제공하므로 사용자는 질의문의 문법을 알 필요가 없다. 웹 인터페이스에서 입력된 질의 정보는 wbridge에 전달되고 wbridge는 질의문을 생성하여 wagent에 전달한다. wagent는 다중 데이터베이스 미들웨어 시스템에 질의를 전달하고 그 결과를 웹 인터페이스에 출력한다.

3.4 개발 환경

웹 클라이언트는 Solaris 2.x, Netscape Enterprise Web Server 개발 환경에서 개발되었다. 또한, cgi를 이용하여 서버와 연결한다. 이는 MDATA가 현재 서블릿이나 php를 제공하지 않기 때문이다.

4. 결론 및 향후 계획

지금까지 MDATA에서 웹 클라이언트 구성과 사용자 세션을 관리하는 방법을 설명하였다. 이 웹 클라이언트는 사용자가 SQL 질의문을 모르더라도 여러 종류의 데이터베이스에 접근하여 자료를 처리할 수 있다는 장점이 있다. 또한, 전역 데이터베이스와 사용자의 세션을 관리함으로써 구동된 프로세스의 재사용을 가능하게 한다.

향후에는 MDATA에 대한 PHP 또는 서블릿 모듈 개발을 통하여 일반화가 이루어져야 하고, auto-commit 모드가 아닌 트랜잭션 관리에 대한 연구가 더 필요하다. 또한, 전역 데이터베이스에 대한 뷰 기능을 제공하여야 한다.

[참고문헌]

- [1]. Won Kim, Modern Database Systems, Addison Wesley, 1995.
- [2]. Amit P.S. and James A. L, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", ACM Computing Surveys, Vol.22, No.3, 1990.
- [3]. Witold L. and Leo M, "Interoperability of Multiple Autonomous Databases", ACM Computing Surveys, Vol.22, No.3, 1990.
- [4]. 서정민 외, "다중연결을 지원하는 JDBC의 구조," 한국정보과학회 '98 추계 학술 발표 논문집(25, 2), 1998.