

알티베이스 데이터베이스 서버를 위한 PHP 확장 모듈 구현¹⁾

홍대용, 강운학, 이상원
성균관대학교 컴퓨터공학과
e-mail:hongdy07@gmail.com

Implementation of PHP extension for Altibase database server

Dae-Yong Hong, Woon-Hak Kang, Sang-Won Lee
Dept of Computer Engineering, Sung Kyun Kwan University

요 약

PHP는 널리 쓰이는 웹 개발 언어이고, 다양한 데이터베이스 관리 시스템(DBMS)을 위하여 C로 작성된 인터페이스들을 갖고 있다. 특히 ODBC라는 표준화된 인터페이스를 통해 DBMS를 이용할 수 있지만, 이는 DBMS 고유의 인터페이스를 구현하여 사용하는 것에 비해 시스템을 활용하는 면에서 한계를 가진다. 국산 DBMS인 알티베이스는 고유의 C 인터페이스를 제공하고 있으며 본 논문에서는 이를 이용한 전용 PHP 인터페이스 구현에 대해 다루고, 구현의 결과 PHP스크립트에서 데이터베이스 연결 및 데이터 운용을 할 수 있었다.

1. 서론

인터넷의 급격한 발달로 인해 오늘날에는 웹 애플리케이션 개발을 위한 다양한 기술이 존재하게 되었다. 그중 PHP는 웹 개발 언어로써 직관적이고, C나 JAVA와 유사한 문법 구조로 인하여 사용자가 배우기 쉬우며 개발 시간이 짧은 장점을 가지고 있다. 이런 이유로 인해 PHP는 세계적으로 가장 널리 쓰이는 웹 개발 언어로 자리 잡고 있다[1, 2]. Facebook, YouTube, Yahoo! Flickr와 같은 주요 소셜 콘텐츠 사이트도 PHP를 이용하여 개발되었으며 [3], 웹 기술 관련 조사를 실시하는 W3Techs에 따르면 실로 오늘날 웹 사이트를 위한 서버측 프로그래밍 언어의 78%가 PHP인 것으로 나타났다.

웹 사이트는 일반적으로 데이터베이스 관리 시스템(DBMS)과 함께 구성되며 이를 웹 애플리케이션에서 쉽게 이용하기 위해 JDBC(Java Database Connectivity)같이 인터페이스를 제공하는 라이브러리나 확장 모듈을 사용한다. PHP는 기본적으로 다양한 DBMS의 인터페이스를 지원하며, 이를 C로 작성하는 PHP 확장 모듈(PHP Extension, 이하 Extension)의 형태로 제공하고 있다. Extension이 항상 PHP와 제3의 라이브러리를 중개할 필요는 없지만, 일반적으로 C로 작성된 외부 라이브러리를 PHP 스크립트에서 사용하기 위한 수단으로 사용된다[4].

PHP는 Extension의 한 가지로 범용 인터페이스인 ODBC(Open Database Connectivity)를 제공한다. PHP용 ODBC를 이용하기 위해서는 표준 C 인터페이스인 ODBC 드라이버가 있어야 한다. PHP전용 인터페이스가 없는 DBMS일지라도 ODBC 표준 드라이버가 있다면 PHP용 ODBC를 이용할 수 있다. 하지만 일반화된 ODBC는 특정 DBMS가 제공하는 고유 기능을 지원하기에는 한계가 있다. 국산 DBMS솔루션인 알티베이스의 경우, PHP와의 연동을 위한 고유 인터페이스 없이 ODBC를 이용해왔다. 하지만 알티베이스 고유의 인터페이스인 ACI(Altibase C Interface)가 가용하기 때문에 이를 사용하여 Extension을 구현하면 PHP와 알티베이스간 전용 인터페이스를 구성할 수 있다. 또한 ACI는 표준 ODBC를 이용하는 것보다 적은 함수 호출을 하면서 동일한 기능을 수행할 수 있기 때문에 [5], 이를 이용하여 구현된 PHP Extension은 ODBC를 이용한 것보다 성능 면에서 앞설 수 있다. 이는 앞서 언급한 PHP의 시장성과 결합되어 새로운 알티베이스의 장점으로 부각될 수 있는 부분이다.

본 논문에서는 알티베이스를 PHP 애플리케이션에서 사용하기 위한 전용 Extension(이하 알티베이스 Extension)의 구현과 그 결과에 대해 다룬다. 2장에서는 Extension의 제작 기반인 PHP와 PHP Extension, 알티베이스에 대해 설명하고, 3장에서는 데이터베이스 연결 및 질의 명령 수행의 기본적인 주요 함수 위주로 알티베이스 Extension의 구현 방식을 설명한다. 4장을 끝으로 결론을 맺는다.

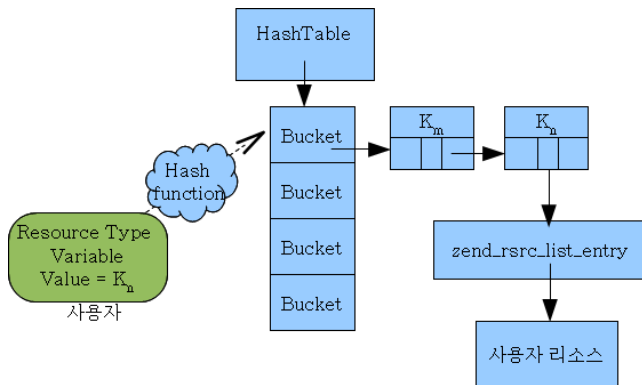
1) 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원사업의 연구결과로 수행되었음 (NIPA-2012-(H0301-12-3001))

2. 관련 연구

2.1. PHP(Professional Hypertext Preprocessor)

PHP는 1995년 6월에 Rasmus Lerdorf에 의해 발표된 서버측 스크립트 언어이다. 주된 목적으로는 동적 웹 애플리케이션을 개발하는데 사용되며, 인터프리터 방식을 취하여 PHP 스크립트로부터 HTML 콘텐츠를 생성한다. PHP 인터프리터는 C언어를 이용하여 개발되었고 오픈소스라는 특징이 있다. PHP 인터프리터는 일반적으로 사용되는 대부분의 운영체제에서의 사용이 가능하며, 아파치 등의 주요 웹서버의 모듈 형식으로 같이 사용될 수 있다[6].

PHP는 C를 이용하여 만들어졌지만 사용하는 자료형은 다소 차이가 있다. Long, String, Boolean, Double, Array, Object, Resource, NULL 등의 자료형이 사용되며, Extension을 구현할 때 이러한 점이 특히 고려되어야 한다. 특히 Resource 형은 PHP가 표현할 수 없는 포인터나 구조체등을 이용하기 위한 중요 수단이다. 포인터나 사용자 정의 자료형의 데이터는 PHP 엔진 내부의 해시테이블에 등록하고 (그림 1)과 같이 그 키(Key)값을 사용자가 Resource형으로 지니고 있으면서 해당 자원에 접근할 때 키값을 사용하는 방식을 취하는데, 해시테이블에 데이터를 등록하는 데에 어떠한 자료형 제한도 없으므로 실제로는 Resource형을 이용하면 C의 모든 자료형을 사용할 수 있다.



(그림 1) Resource 자료형

PHP는 2012년 9월 현재 최신버전으로 PHP 5.4.7이 존재한다.

2.2. PHP Extension

PHP는 그 자체에서 제공되는 기능 외에 확장 모듈(Extension)을 제공하며 PHP의 기능적 한계를 넓힐 수 있다. 또한 동일한 기능을 구현하더라도 인터프리터 언어인 PHP스크립트를 이용하는 것보다 컴파일된 C 라이브러리를 내부적으로 사용하는 Extension을 사용하는 것이 성능 향상에 유리하다[4]. 이러한 Extension들은 공유 라이브러리 형태로 컴파일 되어 PHP의 일부가 된다.

Extension의 형태로 제공되는 DBMS 인터페이스의 대표적인 예로, MySQL을 위한 MySQL 및 MySQLi, ORACLE의 OCI8, SQLite의 SQLite3 등이 있다.

2.3. 알티베이스와 ACI(Altibase C Interface)

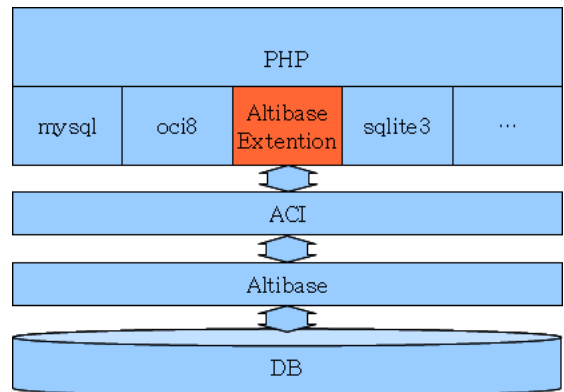
알티베이스는 초기 MMDB(Main Memory Database)로 시작하여, 현재는 MMDB와 디스크 기반 데이터베이스 개념의 혼합 형태로 발전한 세계 최초의 하이브리드 DBMS 솔루션이다. 알티베이스는 국방부, 검찰, 대법원등의 국내 기관 및 SK, China Mobile등 국내외 통신 사업체와 증권사, 제조업체에서 주로 사용되고 있다. 알티베이스는 C, C++를 이용하여 개발되었고 현재 Altibase HDB 6.1.1이 최신 버전이다.

ACI는 C로 작성하는 애플리케이션에서 알티베이스 HDB를 사용하기 위한 전용 인터페이스이다. 데이터베이스에 접근하여 데이터를 조회하거나 조작 할 수 있으며, 알티베이스 이용을 위한 60여개의 C 함수를 제공한다.[5]

3. 알티베이스 PHP Extension

알티베이스 PHP Extension은 (그림 2)와 같이 ACI와 PHP의 중간층의 역할을 수행한다. 사용자 입장에서는 PHP 함수를 호출하지만 내부적으로 ACI 함수가 호출되는 형태를 띠게 된다. 따라서 기본적으로 PHP 함수에서 입력된 매개변수 값이 ACI 함수 호출시 전달되고 ACI 함수의 호출 결과 반환 값이 PHP 함수의 반환 값으로 전달되어야 한다.

ACI는 데이터베이스 연결, 및 질의 실행, 결과 집합으로부터 행 반환, 결과 집합 및 연결 소멸 등의 기본기능을 제공하며 이 부분에 대한 9개 주요함수 외, 기타 기능을 제공하는 19개 함수가 Extension으로 구현되었다.

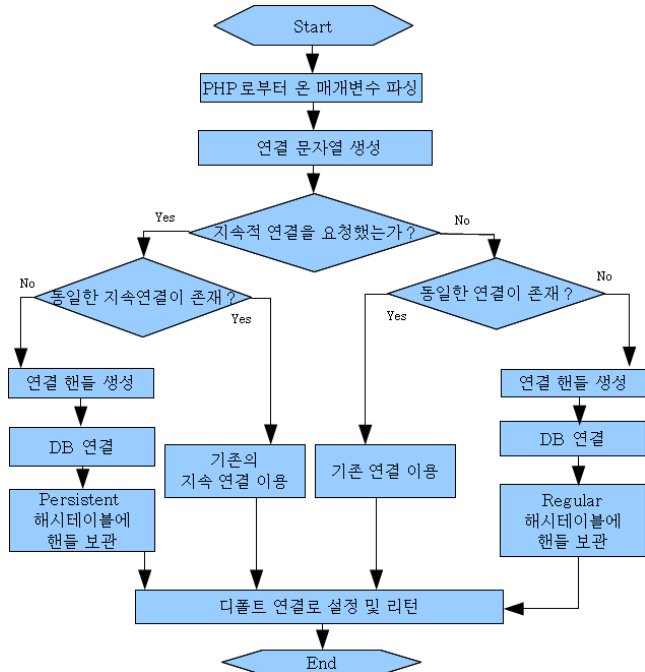


(그림 2) PHP, Extension, ACI의 관계

3.1. 데이터베이스 연결 함수

PHP는 자체적으로 메모리를 관리하는 시스템이 존재하며, 클라이언트로부터 오는 페이지 요청이 한번 종료되면 사용했던 자원을 반환하는 것이 일반적이다. 데이터베이스 연결 시에 생성되는 연결 핸들 또한 한 번의 페이지 요청이 종료되면 소멸되어 더 이상 사용할 수 없게 된다. 클라이언트의 매 요청마다 반복되는 연결 핸들생성의 오버헤드를 피하기 위하여, DBMS Extension들은 페이지 요청 종료에 관계없이 연결이 유지되는 지속적인 연결(Persistent Connection)을 제공한다. 따라서 알티베이스

PHP Extension의 데이터베이스 연결 함수를 구현할 때에도 이 부분을 고려하여 구현하였다. 또한 포인터형 알티베이스 연결 핸들을 PHP의 Resource 자료형으로 대체하여 사용할 수 있도록 하였다. 데이터베이스 연결함수의 전체적인 동작은 (그림 3)과 같다.



(그림 3) 알티베이스 Extension의 DB 연결함수 로직

ACI는 데이터베이스 연결을 위하여 연결 문자열을 필요로 한다. 연결문자열은 연결을 원하는 사용자가 지정한 호스트, 포트, 사용자 및 패스워드, 옵션 등을 정해진 형식의 단일 문자열에 나타낸 것이다. Extension에서는 PHP 함수가 이러한 선택사항들을 하나의 문자열로 받지 않고 사용자가 원하는 사항을 각각 분리된 매개변수에 입력받게 한 후, 함수 내부에서 하나의 연결문자열로 재구성하여 ACI에 전달하는 방식을 택하였다.

앞서 언급한 지속적인 연결에 관하여 사용자는 함수 호출시에 지속적인 연결을 바라는지 여부를 선택할 수 있으며, 지속적인 연결을 선택한 경우에는 실제로 연결 핸들이 소멸되지 않도록 Persistent List라는 PHP 엔진 내의 해시 테이블에 저장하도록 하였다. 이러한 지속적인 연결이 생성되면 페이지 요청이 종료되거나 새로운 요청이 발생하는 것에 무관하게 연결 핸들이 소멸되지 않게 된다. 해당 자원들을 PHP 모듈의 수명이 종료될 때까지 계속 메모리에 있으면서, 동일한 연결의 요청(연결문자열이 동일하게 되는 연결 요청)에 대하여 새로운 연결 핸들의 생성 없이 서비스를 하게 된다.

비 지속적인 연결은 연결이 성립되면 비지속적인 자원을 위한 Regular List 해시 테이블에 저장된다. 일반적으로 한 번의 페이지 요청시 연결이 소멸되지만, 페이지 요청의 종료 없이 두 번이상의 동일한 연결을 생성할 경우

가 있을 수 있으므로, 해시테이블에서 동일한 연결이 있는지 검색하여, 기존의 연결을 재활용 할 수 있도록 하였다.

연결함수의 호출은 해당 연결을 디폴트 연결로 설정하며 디폴트 연결은 가장 최근에 요청된 연결 핸들을 의미하며, 연결 핸들을 매개변수로 하는 다른 함수들의 디폴트 매개변수로써 사용된다.

3.2. 질의문 수행 함수

데이터베이스에 질의문을 전송하면 ACI는 그 결과를 가져오기 위해 두 가지 방식을 제시한다. 사용자는 결과 집합의 행을 데이터베이스 서버로부터 한 개씩 가져오거나, 아니면 결과 집합 전체를 한 번에 가져올 수 있다. 이 경우 사용자는 질의 결과를 얻기 위해 2회의 함수를 호출해야 한다. Extension에서는 이 과정을 하나로 통합하여 한 번의 함수 호출로 앞서 언급한 과정을 마치도록 하며, 결과 집합을 가져오는 방식에 따라 두 개의 함수 중에 선택하여 사용할 수 있게 하였다.

3.3. 행 반환 함수

사용자는 질의문 수행 후에 행 반환 함수를 이용하여 결과 집합으로부터 행을 한 개씩 가져온 후 사용할 수 있다. 행 반환 함수를 통하여 반환된 한 개의 행은 열로 구성된 Array 형으로 반환된다. 이후 PHP의 Array 접근 방식을 이용하여 각 열에 접근 할 수 있다.

Array형은 PHP 내에서 실제로는 해시테이블이다. 따라서 사용자는 반환된 행에 속한 각 열에 접근할 때에 문자열 형식의 해시키(Hash Key)를 사용할 수 있다. 알티베이스 Extension에서는 사용자가 열에 접근할 때 배열 오프셋 이외에, 열의 이름을 해시키로 사용할 수도 있도록 행 반환 함수를 구성하였다. 예를 들어, NAME과 AGE를 열로 갖는 행을 \$row에 반환받았다면, NAME열에는 \$row[0] 또는 \$row[NAME]과 같이 접근 할 수 있다.

4. 결론

본 논문에서는 PHP에서 사용하기 위한 알티베이스 전용 인터페이스 구현 및 방식에 대하여 다루었다. ACI가 제공하는 C 라이브러리를 이용하여 PHP용 Extension을 구현하였고 그 결과 PHP 스크립트에서 알티베이스 DB에 접속, 질의문 수행 및 데이터 운용이 가능하였다. 특히 DB에 연결 하는 부분에 있어 지속적 연결과 비 지속적 연결을 구분하여 사용할 수 있었다.

ACI는 1회의 구문 분석 이후 질의문을 여러 번 재사용 할 수 있도록 하는 Prepare-Execution 방식의 인터페이스를 제공하고 있다. 차후에는 이 Prepare-Execution 방식의 인터페이스를 Extension에 구현 할 예정이다. 현재 Extension에 구현되어있는 인터페이스는 질의문을 실행할 때마다 항상 새롭게 구문 분석을 해야 한다. 하지만 동일한 구조의 질의문을 반복적으로 실행할 경우 처음 한 번의 구문 분석만 의미가 있으며, 그 이후 동일한 구조의 질

의에 구문분석을 하는 것은 불필요한 동작이다. Prepare-Execution 방식의 인터페이스는 반복적 질의 실행시 처음 1회 이후의 구문 분석은 생략되므로, 기존의 질의 실행 방식보다 웹 애플리케이션의 성능 향상을 꾀할 수 있을 것이다.

참고문헌

- [1] 정인근, 이명무, 김용진, “Perl/CGI와 PHP의 비교를 통한 웹 어플리케이션 개발성장에 미치는 영향에 관한 연구”, 대한산업공학회/한국경영과학회 춘계공동학술대회, 2002.
- [2] Wei Cui, Lin Huang, LiJing Liang, and Jing Li, “The Research of PHP Development Framework Based on MVC Pattern”, 4th International Conference on Computer Sciences and Convergence Information Technology, Nov, 2009.
- [3] Won Kim, Ok-Ran Jeong, Sang-Won Lee, “On social Web sites”, Information Systems, vol. 35, April, 2010, pp. 215-236.
- [4] Rasmus Lerdorf, Kevin Tatroe, Bob Kaehms and Ric McGredy, “Programming PHP”, O’Reilly, 2002, Chapter 14.
- [5] “Altibase C Interface Manual”, ALTIBASE co., April, 2012.
- [6] Rasmus Lerdorf, Kevin Tatroe, Bob Kaehms and Ric McGredy, “Programming PHP”, O’Reilly, 2002, Chapter 1.