

웹상에서의 효율적인 데이터 관리 방안

The Efficient Data Management Method on Web

최신형, 한군희, 진광윤**
삼척대학교, 천안대학교, 삼척대학교**

Choi Shin-Hyeong*, Han Kun-Hee*,
Jin kwang-yun**
Samcheok Univ., Cheonan Univ.*,
Samcheok Univ.**

요약

인터넷이 발달하면서 기존의 문서들이 웹을 통해 제공된다. 시간이 흐를수록 이들 정보들은 수정 및 추가 등을 통해 데이터의 양은 증가한다. 웹이 광범위하게 사용됨에 따라 정보를 획득하기 위해 대다수의 사용자들이 웹에 의존하고 있으며, 이로 인해 웹을 통한 데이터 관리의 필요성이 증가하고 있다. 네트워크로 연결되어 있는 환경에서는 갑작스런 시스템 장애가 자주 발생하며, 이런 위험으로부터 데이터를 안전하게 관리·보호하는 것이 중요한 업무 중 하나이다. 본 논문에서는 웹상에서 백업과 복구를 통해 데이터 관리를 체계적이고 효율적이며 안정적으로 처리할 수 있는 시스템을 제안한다.

Abstract

The Internet advances and the documents of existing is provided on web. An amount of data is increased through amendment and addition of information. The web is wide and is used and to follow the users of the majority which acquires information depend in web, the necessity of the data management by web is increasing. The sudden system failure frequently occurs from network environment, we must protect data from this danger. In this paper, we present data management system, which is composed of backup and restoration. This system provides systematic, efficient and stable data management on web.

I. 서론

최근 컴퓨터와 인터넷의 발전으로 웹 기반 시스템 기술이 눈부시게 발전하였다. 그리고 그 발전 속도는 앞으로 더 가속화되어 웹 기반 기술이 우리 생활의 거의 모든 분야에 실제 영향을 미치게 될 것이다.

인터넷의 발전으로 사용자들은 컴퓨터 네트워크를 이용하여 자신이 원하는 정보를 수집한다. 즉, 인터넷은 정보전달과 서비스 이용의 수단으로써 큰 역할을 하고 있다[1, 2, 3, 6].

이와 같은 인터넷의 장점과 사용자들의 자료수집

의식의 변화로 인해 기존의 오프라인 상에서 서적이 라는 문서로서 존재하던 데이터가 대부분 데이터베이스 형태로 저장되며, 앞으로도 많은 양의 데이터가 저장될 것이다. 이를 통해 사용자들은 보다 빨리 정확한 정보를 수집할 수 있다.

그러나 기하급수적으로 증가하는 정보로 인해 웹에서 제공되는 정보는 오프라인 상의 문서보다는 최근의 데이터로 구성되어야 하며, 사용자들 또한 최신의 정보를 구하기 위해 인터넷을 이용한다. 또한 많은 웹 응용프로그램들이 사용자가 필요한 정보들을 찾고 이용하는 것을 돕는다.

웹을 통해 정보를 제공하고자 하면 지속적이며 안정적으로 데이터를 갱신하여야 하며, 주기적인 백업을 통해 귀중한 데이터가 갑작스런 문제로 인해 삭제되지 않도록 관리하여야 하고, 데이터를 안전하게 보호하기 위한 보안문제 또한 중요하다.

그러므로 본 논문에서는 웹상에서 데이터의 백업과 업로드 측면에서 데이터 관리를 효율적이며 안정적으로 처리할 수 있는 방법을 제안한다.

II. 데이터 관리

일반적인 단일 시스템에서도 갑작스런 시스템 장애, 자연적으로 발생하는 재난 그리고 우발적으로 일어난 손실로부터 데이터를 안전하게 보호하는 것이 중요한 업무 중 하나이다. 네트워크를 통해 연결되어 있는 환경에서는 이와 같은 문제가 더욱 더 다양하며 언제 발생할 지 예측 또한 불투명하다. 그러므로 이에 대한 대비는 필수적이며, 특히 웹을 통해 데이터를 제공하는 측에서는 데이터 관리에 대한 방안이 마련되어야만 한다.

일반적으로 데이터베이스를 사용하여 데이터를 관리하는 환경 하에서는 "db dump"라는 명령어를 통해 주기적으로 백업을 한다. 즉, 데이터베이스에 저장되어 있는 데이터를 보존 혹은 백업용으로 만들어 놓는다. 백업을 하는 목적은 시스템 해킹 혹은 천재지변으로 인해 발생한 서버가 다운되었을 때 데이터 분실을 막기 위해서 실시한다.

하지만 "db dump"를 통해 백업되는 파일은 외부 시스템이 아닌 서버 자체에 저장되므로 추후 FTP 접속을 통해 해당 파일을 다운받아야 하는 불편함이 존재한다.

III. 데이터 관리 방안

일반적으로 "db dump"를 사용하여 백업을 실시하였다 하더라도 서버 자체에 문제가 있을 경우 복구

가 불가능할 수도 있다. 또한, 관리자가 원격지에 떨어져 있는 경우 telnet 서비스를 통해 백업을 수행 후 ftp 접속을 통해 다운로드 받아야 한다.

그러므로 본 논문에서는 웹을 통해 데이터를 제공하는 경우 장소에 상관없이 인터넷이 가능한 곳에서 웹 브라우저를 통해 백업과 업로드를 할 수 있는 방안을 제시한다.

1. 백업

웹상에서 데이터 서버를 관리하기 위해서는 인증과정을 통한 데이터 서버에 대한 접근과 조작이 필요하다. 이를 위해 본 논문에서는 웹상에서 유효한 인증과정을 통해 데이터 서버에 접근하여 데이터에 대한 백업을 수행하기 위해 두 가지 방식을 사용한다. 즉, Data 방식과 SQL 방식으로 나누어서 백업을 수행한다.

Data 방식은 별도의 SQL 문장은 생략하고 순수 데이터만을 백업한다. 이 방식은 SQL 방식에 비해 순수한 데이터만을 추출함으로써 백업 데이터 크기의 축소와 테이블 구조가 문제없을 경우 신속한 업로드가 가능하다.

표 1은 웹상에서 데이터 서버에 접속하여 해당 테이블을 Data 방식으로 백업하는 코드이다.

[표 1] Data 방식 백업

```
function DB_data_down($dbname)
{
    $connect=db_connect();
    $db_name=db_info('db');
    mysql_select_db($db_name,$connect);
    $result = mysql_query("show table status from
$db_name like '$dbname' ",$connect)
    while($dbData=mysql_fetch_array($result))
    {
        $TBLName = $dbData[Name];
        DB_getDataList($TBLName);
    }
}
```

SQL 방식은 테이블 생성(create table 문)과 데이

터 삽입(Insert 문)을 포함한 것으로 데이터 서버에 문제가 발생할 경우 MySQL을 재설치한 후 쿼리를 수행함으로써 쉽게 데이터를 복구할 수 있게끔 한다. 그림 2는 웹상에서 데이터 서버에 접속하여 해당 테이블을 SQL 방식으로 백업하는 코드이다.

[표 2] SQL 방식 백업

```
function DB_schema_down($dbname)
{
    $connect=db_connect();
    $db_name=db_info('db');
    mysql_select_db($db_name,$connect);
    $result = mysql_query("show table status from
$db_name like '$dbname' ",$connect)
    while($dbData=mysql_fetch_array($result))
    {
        $TBLName = $dbData[Name];
        DB_getSchema($TBLName);
    }
}
function DATABACKUP_down($tableName)
{
    DATABACKUP_getSchema($tableName);
    DATABACKUP_getDataList($tableName);
}
```

2. 복구

데이터 서버의 갑작스런 고장이나 DBMS의 손상으로 인해 해당 서버에 저장된 데이터의 복구가 불가능할 경우에는 데이터 서버에 MySQL을 재설치 후 데이터 서버 관리시스템의 백업기능을 사용하여 백업받은 DB파일을 복구 시킬 수 있다.

업로드 과정은 데이터 서버 고장뿐 아니라 데이터에 문제점이 발생할 경우에도 이전에 백업받은 DB파일을 복잡한 절차 없이 데이터 서버 관리 시스템의 Restore 기능을 통해 빠르게 복구할 수 있다.

그러므로 복구과정은 두 가지로 나눌 수 있다.

첫째, 데이터 구조에는 문제가 없지만 입력 데이터에 문제가 발생한 경우에는 Data 방식을 통해 다운

로드 받은 순수한 데이터만을 업로드 함으로써 복구 시킨다.

그림 3은 데이터 구조는 정상인 경우 순수한 데이터를 업로드 시키는 코드를 나타낸다.

[표 3] Data 방식 복구

```
function DB_data_upload($dbname)
{
    $connect=db_connect();
    $db_name=db_info('db');
    mysql_select_db($db_name, $connect);
    $update_file = $upload_path.$data.".dat";
    $file_name=$data;
    if(file_exists($update_file)){
        unset($dbData);
        $dbData = file("$update_file");
        $sql = "INSERT INTO $table_name VALUES (*:
$colum_data = trim($data[$field_cnt]);
        $sql .= "$colum_data)";
    }
    while(!mysql_close($connect));
}
```

둘째, 데이터 구조에 문제가 발생한 경우에는 SQL 방식을 통해 다운로드 받은 데이터 구조와 데이터 모두를 업로드 함으로써 복구시킬 수 있다.

[표 4] SQL 방식 복구

```
function DB_upload($dbname)
{
    if (strlen(trim($data))>0){
        $update_file = $upload_path.$data.".tar";
        $file_name = $upload_path."down_data.tar";
        if(file_exists($update_file)){
            exec("cp -u $update_file $file_name");
        }
        if(file_exists($file_name)){
            exec($upload_path."tar.sh");
        }
    }
}
```

IV. 결 론

현재는 많은 사람들이 정보를 구하기 위해 인터넷을 이용하고 있다. 이런 흐름에 따라 기존에 오프라인을 통한 정보 제공자들도 웹에 중요한 정보를 업로드 시키고 있으며, 이들 온라인 정보는 기하급수적으로 증가하고 있다. 네트워크를 통해 연결되어 있는 환경에서는 갑작스런 시스템 장애와 같은 손실이 단일 시스템에 비해 발생비율이 높으며, 이런 위험으로부터 데이터를 안전하게 보호하는 것이 중요한 업무 중 하나이다. 현재의 데이터 서버 관리에서는 직접 접근을 통한 서버 관리 방법이 주를 이루며 서버의 일부를 사용하는 사용자들에게는 접근하기에는 어려운 부분이 많다. 이를 개선하기 위해서 본 논문에서는 웹상에서 데이터의 백업과 복구 측면에서 데이터 관리를 효율적이며 안정적으로 처리할 수 있는 방법을 제시하였으며, 이를 이용하면 데이터 서버가 아닌 원격지에서 백업과 업로드가 가능하다.

■ 참고 문헌 ■

- [1] Abiteboul, Buneman, Suciu, "Data on the Web from relations to Semistructured Data and XML", Morgan Kaufmann Publishers, 2000.
- [2] Boldyeff C., Kewish R., "Reverse engineering to achieve maintainable WWW site", Reverse Engineering, 2001. 8th Working Conf. on, 2001.
- [3] Jun Hirai 외 3인, "Webbase: A repository of web pages. In Proceedings of the International World-Wide Web Conference, 2000.
- [4] 루크 웰링 외, PHP와 MYSQL, 인포북, 2003.
- [5] 김종환 외 3인 "웹 사이트의 효율적인 구조 관리와 평가를 위한 시스템의 설계 및 구현", 한국정보처리학회 추계학술대회 논문집, 제11권, 제2호, 2004.
- [6] <http://www.di.fc.ul.pt/tech-reports/03-8.pdf>
- [7] <http://www.mysql.com>
- [8] <http://www.php.net>