

웹사이트 유지보수를 위한 자동 업데이트 시스템의 설계 및 구현

Design and Implementation of an Automatic Update System for Website Maintenance

황대현*, 유재수**

충북대학교 전자정보공학과*, 충북대학교 정보통신공학과**

DaeHyeon Hang(hyeon_hd@eniasoft.com)*, JaeSoo Yoo(yjs@cbnu.ac.kr)**

요약

오늘날 우리는 인터넷을 통하여 웹사이트에서 다양한 활동과 많은 정보를 얻고 있다. 이러한 웹사이트들은 개인 또는 웹사이트 전문업체에 의해 유지보수되며, 기본적인 방법은 운영되고 있는 웹사이트를 구성하고 있는 파일들을 변경하는 것이다. 이 과정에서 전체 파일을 변경하는 것은 시간이 오래 걸리고, 변경하지 않아도 되는 파일들까지 변경하므로 효율성이 많이 떨어진다. 반대로 변경해야 하는 파일들만 변경하게 될 경우, 사람이 직접 각 경로를 탐색하여 파일들을 확인하고 하나씩 파일을 변경해야 하므로 손이 많이 간다. 이를 해결하고자 자동 배포 시스템들이 개발되었으나, 추가적인 자원과 학습이 필요하며 비용과 시간, 노동력이 추가로 발생한다. 이에 본 논문에서는 기존 웹사이트의 자원과 기술을 사용하여, 자원 소모를 최소화하고, 새로운 기술의 학습이 필요 없는 자동 업데이트 시스템을 제안하고자 한다. 이를 통해 사람의 작업 대비 신뢰성과 시간을 향상하는 것을 목적으로 한다.

■ 중심어 : | 전자정부 프레임워크 | 웹사이트 | 유지보수 | 자동 업데이트 | 자동 배포 |

Abstract

Today we are getting a lot of information and various activities on our website through the internet. These websites are maintained by individuals or by website specialists. The basic method is to change the files that make up the running website. Changing the entire file in this process takes a long time and changes the files that do not need to be changed, so the efficiency is greatly reduced. When only the files that need to be changed are changed, it takes a lot of effort as a person must manually search each path to check the files and change the files one by one. To solve this problem, automatic distribution systems were developed. Additional resources and learning are required, resulting in additional cost, time and labor. Therefore, in this paper, we propose an automatic update system to minimize resource consumption by using the resources and technologies of the existing website. The proposed system does not require learning new skills. This aims to improve reliability and reduce time compared to human work.

■ keyword : | eGovFrame | Website | Maintenance | Automatic Update | Automatic Distribution |

* 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. 2019R1A2C2084257), 과학기술정보통신부 및 정보통신기획평가원의 Grand ICT연구센터지원사업(IITP-2020-0-01462), 2020년 대한민국 교육부와 한국연구재단의 지원(NRF-2020S1A5B8103910), 그리고 중소벤처기업부 '산업전문인력역량강화사업'의 재원으로 한국산학연합회(AURI)의 지원(2020년 기업연계형연구개발인력양성사업, 과제번호 : S2929950)을 받아 수행된 연구임

접수일자 : 2021년 01월 11일

수정일자 : 2021년 02월 17일

심사완료일 : 2021년 02월 17일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

오늘날 우리는 인터넷을 통하여 웹사이트에서 다양한 활동과 많은 정보를 얻고 있으며, 기관과 기업은 홍보와 서비스 제공을 목적으로 활용하기도 한다.

이러한 웹사이트들은 개인 또는 웹사이트 전문업체에 의해 유지보수 된다. 웹사이트의 유지보수에서 업데이트 요소가 발생 시, 기본적인 방법은 현재 운영되고 있는 웹사이트를 구성하고 있는 파일들을 변경하는 것이다. 파일의 변경은 신규 파일의 추가, 기존 파일의 수정 또는 삭제로 나눌 수 있으며, 이 파일들은 보통 개인의 컴퓨터가 아닌 웹사이트를 운영하는 별도의 서버 컴퓨터에 존재한다. 개발자(개인)는 PC에서 서버로 접속하여 작업하는 것이 일반적이며 Windows 서버는 마이크로소프트사에서 제공하는 원격 접속 기능을 통하여 접속 가능하며, 리눅스 서버의 경우 별도의 FTP 클라이언트 또는 SSH로 접속한다. 접속한 이후에는 서버에서 에디터 프로그램을 통하여 기존 파일들을 바로 수정할 수도 있으나, 안전성의 문제로 많이 추천되지는 않는다. 대신, 기존에 PC에서 작업한 파일들을 서버에 복사하는 방식으로 변경한다. 이 과정에서 여러 개의 폴더나 파일을 모두 덮어쓰우는 것은 시간이 오래 걸리고, 변경하지 않아도 되는 파일들까지 변경하므로 효율적인 측면에서도 많이 떨어진다. 반대로 변경해야 하는 파일들만 변경하게 될 경우, 사람이 직접 각 경로를 탐색하여 파일들을 확인하고 하나씩 파일을 변경해야 하므로 손이 많이 간다. 더욱이 앞선 두 방법 모두 사람이 직접 작업하기 때문에 작업 중, 작업해야 할 내용을 혼동할 수도 있어서 신뢰성에 오류가 있을 수 있으며, 이를 보완하기 위해서는 확인하고, 또 확인하는 방법뿐이다. 신뢰성을 높이는 방법은 얼마나 더 꼼꼼하게 확인하였느냐이며, 신뢰성을 높일수록 작업 시간도 증가하게 된다.

이러한 문제를 해결하고자 프로그램을 통한 업데이트, 자동 배포화 시스템들이 개발되었으나, 이는 또 다른 한계를 가지고 있다. 각각의 시스템들은 많은 장·단점이 있으며 공통으로 해당 시스템을 사용하기 위해서는 추가적인 하드웨어 자원과 전문적인 기술의 학습이 필요하다. 이는 비용과 시간, 노동력의 추가로 직결되며

해당 시스템의 도입에 부정적인 영향으로 작용한다.

이에 본 논문에서는 기존 웹사이트의 자원과 기술을 사용하여, 자원 소모를 최소화하고, 새로운 기술의 학습이 필요 없는 자동 업데이트 시스템을 제안하고자 한다. 이를 통해 사람의 작업대비 신뢰성과 시간을 향상하는 것을 목적으로 한다.

본 연구의 구성은 다음과 같다. II장에서는 본 연구와 관련된 웹사이트의 개념과 기존의 시스템을 분석하고, III장에서는 제안하는 자동 업데이트 시스템을 설계한다. IV장에서는 제안하는 시스템의 구현 환경 및 결과, 실험 결과를 기술하고 마지막 V장에서는 연구의 결론에 대하여 기술한다.

II. 관련연구

1. 웹사이트의 개념

국내에서는 흔히 “홈페이지”라는 명칭으로 통용되며, 인터넷 프로토콜 기반의 네트워크에서 도메인이나 IP 주소(또는 루트 경로만으로 이루어진 URL)를 통하여 보이는 웹 페이지들의 그룹을 의미한다[1]. 웹사이트는 인터넷이나 랜(LAN)과 같은 네트워크를 통해 접속할 수 있어야 하며, 이를 위해 최소 1개 이상의 웹 서버를 통하여 호스팅 된다. 1개의 웹페이지는 일반적으로 순수 문자열의 문서이나 HTML/XHTML의 형식으로 표현할 경우, 웹 브라우저를 통하여 HTML 마크업 명령을 표시하는 그대로 페이지 내용을 표현한다[2]. 웹페이지는 HTTP로 접속하며, HTTPS를 사용하여 웹페이지 콘텐츠를 이용한 사람들에게 보안과 개인 정보 보호를 제공한다. 국내에는 관련법에 의거 개인정보를 입력받는 모든 웹사이트는 HTTPS로 통신하여 암호화된 통신을 하도록 한다.

2. 기존 업데이트 시스템

수정 또는 개선을 위해 컴퓨터 프로그램이나 지원 데이터를 변경하도록 설계된 일종의 행위로 간주하며, 일반적으로 이미 발표된 소프트웨어 제품에서 발견된 사소한 기능 개선 또는 버그나 오류 등을 수정하기 위해 개발자가 내놓는 업데이트를 지칭한다[3]. “배포”라는

명칭으로도 사용되기도 한다.

별도의 프로그램으로 배포되지 않는 웹사이트에서는 수정 또는 개선되는 대상이 현재 운영되고 있는 웹 서버(또는 WAS)의 파일들로 한정되며, 서버의 파일들을 교체함으로써 인해 웹사이트 페이지의 내용을 바꾸는 것이 목적이다. 리눅스 서버를 기준으로 대표적인 수동 배포 방법은 FTP 클라이언트를 통하여 서버에 접속하여 이미 수정된 파일을 올리는 방법과 시큐어 셸(Secure Shell, SSH)을 통하여, 서버에 원격으로 접속하여 실시간으로 파일을 수정하는 방법이 있다[4][5]. 간단한 예시로 기존에 자동차를 소개하는 웹사이트에 오토바이 항목을 추가하고자 한다면, 기존의 자동차 페이지와는 다른 페이지가 필요하다. 이에 오토바이에 대한 새로운 페이지를 만드는 것은 신규 추가에 해당하며, 기존의 자동차 페이지에 오토바이 페이지로 이동할 수 있는 링크를 추가하는 것은 수정에 해당한다. 이렇게 새로 만든 오토바이 페이지와 수정된 자동차 페이지를 운영 서버에 적용하려면, PC에서 운영 서버로 FTP 클라이언트를 통해 접속한 후, 운영 서버 내에 홈페이지를 구성하는 경로를 찾아간다.

이때 기존의 자동차 페이지가 /website/car/info.html 경로에 존재한다고 가정할 때, 기존의 info.html 파일은 언제든지 사용이 가능하도록 백업을 하고, 수정된 info.html 파일로 기존의 파일을 대체한다. 새로 만들어질 오토바이의 페이지는 /website/bike/의 경로를 새로 만들고, /website/bike/info.html로 새 파일로 만든다. 이렇게 운영 서버에 적용한 후에 웹브라우저로 해당 웹사이트에 접속하여, 각각 자동차, 오토바이 페이지에 접속하여 정상적으로 내용이 잘 나오는지 여부를 확인하고 작업을 종료하며, 이러한 과정을 작업하는 페이지의 수에 비례하여 반복한다.

기존 “배포” 작업의 경우, 사람에 의해서 진행되는 작업이기 때문에 순서의 오류, 배포 대상의 오류등 다양한 실수가 발생할 소지가 있으며, 이 단순하고 반복적인 작업을 매번, 사람이 일일이 하는 것은 번거롭고, 비효율적이라는 의견이 지배적이다. 이러한 문제를 해결하고자 인간을 대신하여, 자동으로 “배포”하는 방법에 있어서 많은 연구가 이루어졌다. 그중 대표적으로 젠킨스를 활용하는 방법이 있다.

젠킨스는 지속적인 통합(CI)을 목적으로 여러 사용자의 작업 과정을 하나로 통합해주는 것이 크지만, 통합 이후에 자동으로 운영 서버에 배포하는 기능도 많이 사용되고 있다. 보통, 젠킨스 자체는 여러 개의 플러그인을 특정 상황에 맞추어 구동할 수 있는 틀을 제공하며, git, svn, maven, ant, ssh, shell 등 여러 플러그인과 명령 조합을 통하여 특정 임무를 수행하고자 하는 경우가 많다. 즉, 단순 war 배포를 제외하면 git, svn, maven, ant, ftp, ssh, shell 명령어 등 일정 수준을 하여야 하며, 이를 응용하여 자신이 원하는 상황에 맞게 조합할 수 있는 능력이 필요하다. 이는 초보자들에게 있어서 큰 벽과 다름없으며, 사용자의 목적에 맞게 설정하고 구성을 조합하였어도, 젠킨스의 원래 목적인 지속적인 통합은 단점이 될 수 있다. 예를 들어 누군가 젠킨스 도입을 하여 자동으로 운영 서버가 배포까지 설정되었다면, 중간에 누가 운영 서버에 작업을 하더라도 젠킨스가 이를 덮어쓰기 때문에 젠킨스를 통해서만 배포할 수밖에 없고, 프로젝트의 모든 인원이 젠킨스와 연동된 시스템/프로그램을 사용해야 한다.

III. 자동 업데이트 시스템 설계

1. 시스템 구조

웹사이트의 배포는 이미 개발이 완료된 파일들을 개발 스토리지에서 서버로 복사하고 적용하는 것이다. 이는 크게 3가지 단계로 나누어지는데, 서버에 배포할 파일을 준비하는 단계, 운영 서버의 교체될 파일들을 백업하는 단계, 서버에 파일을 적용하는 단계이다. 첫 번째로 서버에 배포할 파일을 준비하는 단계는 개발 스토리지에서 서버로 업데이트할 파일들을 분류하는 단계로 전체 업로드와 부분 업로드가 있다. 전체 업로드는 말 그대로 개발 스토리지에 있는 모든 파일을 서버로 올리는 것이다. 이는 주로 처음 서버로 업로드 또는 수정된 파일의 양이 많아서 일일이 분류하기 어려울 때 사용하는 방식으로 모든 파일을 올리기 때문에 시간이 오래 걸리며, 기존에 서버에 있는 파일 중 변경되어서는 안 되는 중요 파일들에 유의해야 한다. 부분 업로드는 개발자가 수정한 파일들만 업로드는 하는 방식으로

수정한 파일들이 적을 때 사용한다. 올리는 파일의 양이 적기 때문에 속도가 빠르나, 사전에 올리는 파일들을 분류하는 작업이 필요하다. 어떻게 파일을 올릴지, 어떠한 파일들을 올릴지 정했다면 두 번째는 백업단계이다. 기존 서버에 있는 파일 중, 앞서 분류한 결과에 따라 수정, 삭제될 파일들을 안전한 곳에 이동해놓는 작업이다. 모든 작업이 완료 후에 이전으로 되돌릴 필요가 없다면 좋겠지만, 간혹, 어떠한 이유로 이전의 파일로 되돌려야 할 때가 있다. 이럴 때 해당 파일들이 완전히 수정 또는 삭제되었을 경우, 다시 되돌리지 못하므로 다시 되돌릴 수 있도록 해당 파일들이 수정, 삭제 전에 안전한 장소에 보관하는 작업을 백업이라 한다.

이렇게 백업 작업까지 끝나면 이제 실제로 앞서 정한 업로드 방식과 적용 할 파일들을 실제로 서버의 문서 경로 (Document Root)로 올리는 작업을 진행한다. 업로드가 완료되면 PHP, JSP, ASP 등의 파일들은 서버에 업로드만으로도 작동하지만, Java의 Class 파일들은 WAS 서버를 재실행해야만 작동한다. 이에 Class 파일과 같이 재실행을 해주어야 하는 파일들이 있다면, 업로드 후에 재실행을 해주도록 한다. 본 연구에서는 이 과정을 전자정부 프레임워크를 통한 기본적인 웹사이트를 구현하는 방식으로 자동화하는 것을 목표로 한다. 이렇게 할 경우, 크게 2가지 이점을 가진다. 첫 번째는 기존 웹사이트를 만드는 방식과 같기 때문에 기존에 전자정부 프레임워크의 개발자라면 쉽게 개발 및 수정할 수 있다. 두 번째는 이미 운영 환경의 자원을 그대로 사용할 수가 있다. 이 시스템은 기존 운영 환경과 같은 WAS에 별도로 컨테이너를 하나만 추가하여 사용할 수 있으며, 데이터베이스도 기존 운영 환경에서 사용하는 것을 테이블만 추가하여 사용할 수 있다.

[그림 1]은 시스템 구조를 나타낸다. 사용자가 사용하는 웹사이트는 운영 서버의 소스 코드를 통하여 구동된다. 개발자가 개발 서버에서 테스트를 완료하면, 자동 업데이트 시스템은 개발자가 설정해놓은 폴더설정과 일정 관리에 의하여 개발 서버의 소스 코드를 운영 서버의 업데이트 준비 폴더로 이동하고, 파일 분류, 운영 서버 파일 백업, 운영 서버에 파일 적용, 운영 서버 재기동의 과정을 거치며, 운영 DB에 업데이트 명세를 기록한다.

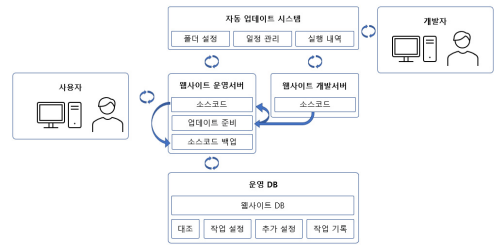


그림 1. 시스템 구조

자동 배포 시스템은 사용자의 명령 또는 기존에 사용자가 정해둔 시간에 따라 동작을 시작하며, 기존에 개발 서버(환경)에서 운영 서버로 접속하여 파일을 올리는 방식과 달리 서버가 개발 스토리지에 FTP로 접속하여 파일을 다운로드하는 형식으로 진행된다. 서버가 개발 스토리지에서 자신의 업데이트 준비 폴더에 파일들을 다운로드한 후, 해당 파일들을 분석하여 신규 추가 파일, 수정 파일, 삭제 파일 3가지로 구분하고 목록으로 만든다. 해당 목록을 통하여 시스템은 운영 서버의 폴더에서 수정되는 파일과 삭제될 파일들을 백업한다. 백업 과정이 끝나면 업데이트 준비 폴더의 파일들을 운영 서버의 폴더들로 이동하고, WAS를 재실행한다.

2. 시스템 기능 정의

자동 배포 시스템은 사람이 수동으로 하던 배포 작업을 자동으로 해주는 것을 목적으로 한다. 여기서 자동의 범위는 사용자가 자동 배포의 실행을 요청하였을 때, 시스템이 사용자 대신 파일을 다운로드, 적용할 파일의 분류, 수정/삭제될 파일의 백업, 파일의 적용, WAS의 재실행까지의 단계와 사전에 사용자가 지정해둔 시간에 시스템 스케줄에 의해 배포 실행까지를 의미한다.

[표 1]은 자동 업데이트 시스템의 기능 명세를 나타내며, 각 기능을 8개의 내용으로 구분하여 설명한다. 자동 배포 시스템은 서로 다른 컴퓨터에 있는 파일을 업로드 또는 다운로드하는 것이 핵심 요소이며, FTP 접속 기능은 가장 대중적인 파일전송 방식인 FTP를 이용하여 개발 환경에 접속하여 파일 다운로드를 실행한다.

파일 다운로드의 사전에 정의된 파일과 폴더를 제외한 모든 폴더와 파일을 서버의 업데이트 준비 폴더로

다운로드받는다. 파일 분류는 개발환경으로부터 다운로드받은 파일들의 Hash 값과 기존 서버의 파일들을 Hash 값을 통하여 비교한다. 모든 파일은 고유한 Hash 값을 가지고 있으며, 파일의 변경 시 hash값도 변한다[6]. 이를 이용하여 모든 파일 들의 hash 값을 추출하여 DB의 테이블에 파일 목록을 만들고, 이 값을 DBMS의 쿼리로 비교하여, 신규, 수정, 삭제 여부를 판단하여 결과를 DB의 테이블에 저장한다[7][8].

파일 분류가 끝난 후에는 백업 스토리지 내에 현재 날짜+시간(yyyymmddhh24miss)으로 이루어진 폴더를 생성한다. 만들어진 폴더를 백업 폴더라 명명하고, 서버 폴더에서 수정, 삭제로 지정된 파일들을 이동한다. 파일의 이동이 완료될 때마다 실행 명세를 DB에 로그로 저장한다. 이를 통해 파일을 복원해야 하는 경우 DB의 로그와 폴더를 통하여, 정확하고 안전하게 복원을 할 수 있다. 파일의 백업이 완료되면, 바로 다음 단계인 파일 적용을 시행한다. 업데이트 준비 폴더에서 "파일 목록"의 신규, 수정 대상 파일들을 서버 폴더로 복사하며, 이 과정을 DB에 기록하여, 모든 단계가 끝난 이후, 사용자가 진행 과정이 정상적으로 진행되었는지와 어떤 파일이 어떻게 적용되었는지 언제든지 확인 할 수 있다. 파일 적용이 완료되면, 운영 서버의 WAS를 재실행한다. 자바의 Class 파일들은 WAS를 재실행하여야만 운영 서버에 적용된다.

표 1. 시스템 기능 명세

기능명	내용 설명
FTP 접속	대상 스토리지로 FTP 접속
다운로드	개발 서버 -> 운영 서버 폴더로 다운로드
목록 생성	개발/운영 서버 파일들에서 Hash값을 추출하여, DB에 1차 목록을 생성
파일 분류	파일 목록을 Hash값을 대조하여 신규, 수정, 삭제로 분류하여 목록 생성
파일 백업	운영 서버의 파일들을 백업
파일 적용	분류된 파일들을 운영 서버에 적용
기록	운영 서버에 적용된 파일들을 DB에 기록
재실행	WAS를 재실행

3. DB설계

다음은 각 TABLE의 세부 구조를 정의한다. [표 2]는 작업 설정 테이블, [표 3]은 추가 설정 테이블을 나타낸

다. 작업 설정 테이블은 작업에 대한 전반적인 정보를 가진 테이블로 작업 대상 서버들과 폴더, 예정된 작업 일정 정보를 가지고 있다. 이를 바탕으로 시스템이 실행되면, 추가 설정 테이블을 통하여, 제외 대상이 되는 폴더들을 제외하고, 개발 서버에서 운영 서버로 파일들을 복사한다.

표 2. 작업 설정 Table 명세

필드	자료형	내용
IDX	NUMBER	구분번호
NAME1	VARCHAR	개발 서버 이름
NAME2	VARCHAR	운영 서버 이름
IP1	VARCHAR	개발 서버 아이피 주소
IP2	VARCHAR	운영 서버 아이피 주소
PATH1	VARCHAR	개발 서버 절대경로
PATH2	VARCHAR	운영 서버 절대경로
MEMO1	VARCHAR	개발 서버 간단한 설명
MEMO2	VARCHAR	운영 서버 간단한 설명
DATE	VARCHAR	날짜, YYYYMMDD (00 표기는 매월, 매일로 처리)
TIME	VARCHAR	시간, HH24MI

표 3. 추가 설정 Table 명세

필드	자료형	내용
IDX	NUMBER	구분번호
CATEGORY	NUMBER	카테고리, 0 : 제외 대상, 1 : 준비 폴더, 2 : 백업 폴더
NAME	VARCHAR	해당 항목의 이름
IP	VARCHAR	대상의 아이피 주소
PATH	VARCHAR	대상의 경로
MEMO	VARCHAR	해당 항목의 간단한 설명

[표 4]는 파일 임시 대조 테이블을 나타낸다. 복사한 파일들은 파일 임시 대조 테이블 기록한 후, SQL 쿼리로 운영 서버 파일들과 개발 서버 파일의 교집합, 차집합을 활용하여, 수정, 삭제, 추가 등 올릴 파일들의 목록을 추출하며, 각각 업데이트 준비 폴더와 백업 폴더에 파일들을 분류한다.

표 4. 파일 임시 대조 Table 명세

필드	자료형	내용
IDX	NUMBER	구분번호
FILE_PATH	VARCHAR	파일의 절대경로
FILE_NAME	VARCHAR	파일명
UPDATE	VARCHAR	업데이트 적용 대상 여부
EXT	VARCHAR	확장자명
HASH	VARCHAR	파일의 중복 여부 구분 값

[표 5]는 작업기록 테이블을 나타내며, 모든 작업이 끝난 후, 업데이트의 수행일, 성공 여부, 파일 명세 등을 작업 기록 테이블에 기록한다.

표 5. 작업 기록 Table 명세

필드	자료형	내용
IDX	NUMBER	구분번호
DATE	VARCHAR	업데이트 수행일
PATH	VARCHAR	업데이트 경로
MEMO	VARCHAR	해당 일정의 간단한 설명
LOG	CLOB	파일의 업데이트 기록 (모든 파일)
ACTION	VARCHAR	동작 방식 : 수동/자동
SUCCESS	VARCHAR	성공 여부 : Y/N

IV. 자동 업데이트 시스템 구현

1. 구현 환경

[표 6]은 본 연구에서 제안하는 구현 환경이다. 서버는 리눅스 계열의 Cent OS 로 구성하였으며, WAS는 아파치 재단의 Apache Tomcat 8.5 버전 DBMS는 Oracle 사의 Oracle 11g 의 사용한다. 개발 언어는 Java 이며, 별도로 전자정부 프레임워크를 사용했다.

표 6. 구현 환경

구분	환경 언어
OS	서버 : Cent OS (linux), 개발 : Window 10
WAS	Apache Tomcat 8.5
DBMS	Oracle 11g
Language	Java 8
Framework	전자정부 프레임워크 3.5.0

2. 적용 방법

제안하는 시스템은 사실상 하나의 웹사이트이며, 외부의 개발 소스 코드를 다른 웹사이트에 적용하는 역할을 한다. 보통 WAS는 다중 컨테이너를 지원하며, DBMS도 복수의 스키마를 지원한다. 즉, 기존의 WAS에 새로운 컨테이너를 추가하고, 새로운 웹사이트로 제안하는 시스템을 등록하면 된다. 하나의 WAS에 기존의 웹사이트와 기존의 웹사이트를 관리하는 웹사이트가 존재하는 구조이다.

간략하게 기존의 웹사이트를 A, 시스템을 B라고 가정하여 아래와 같이 설명한다.

- 1) 새로운 컨테이너를 추가하고, webdoc에 시스템 B의 소스 코드를 올린다.
- 2) DB에 시스템 B가 사용할 스키마와 테이블을 생성한다.
- 3) WAS의 B 컨테이너를 기동한다.
- 4) 시스템 B에 A 웹사이트의 소스 코드 경로와 시스템 설정을 등록한다.

보통, 하나의 WAS는 하나의 권한으로 실행되며, 제안하는 방법에서 WAS는 A와 B를 모두 실행하기 때문에 A와 B는 같은 권한을 가지고 있다. 이는 B의 권한으로 실행되는 명령들이 A의 소스 코드를 백업하거나 수정, 삭제, 추가할 수 있게 해준다.

3. 구현 결과

아래는 본 연구의 설계를 바탕으로 구현된 결과이며, [그림 2]는 자동 업데이트 시스템의 관리를 위해 제공되는 화면이다. 관리자 페이지의 메인 화면으로 관리자 시스템에 접속 시 보이는 화면이다. 이 화면에서는 파일 업데이트에 대한 전반적인 설정을 확인할 수 있으며, 신규 설정의 추가 및 기존 설정을 수정할 수 있다.



그림 2. 시스템 설정

메인화면의 최상단 영역은 개발 서버와 운영 서버가 어떻게 연결되어 있는지를 직관적으로 보여준다. 개발 서버의 구분1 과 운영 서버의 구분1이 1:1 로 연결되어 있으며, 서비스가 실행되면 최종적으로 개발 서버의 설정된 폴더에서 운영 서버의 설정된 폴더로 업데이트가 진행된다. 이처럼 개발 서버와 운영 서버의 폴더를 등록하기 위해서는 하단의 등록 버튼을 통하여 신규화면 등록 페이지로 이동한다.

[그림 3]은 신규 등록 화면을 나타내며, 개발 서버와 운영 서버의 폴더가 각각 1:1로 연결되기 때문에 개발 서버와 운영 서버의 값을 같이 설정한다. 이는 수정, 삭제 시에도 똑같이 적용된다.

신규 환경 등록

대상	항목	아이피	경로	내용
개발서버		0.0.0.0		
운영서버		192.168.0.12		

그림 3. 신규 등록 설정

[그림 4]는 운영 서버에 적용되면 안 되는 폴더 및 파일들을 설정한다. 개발 서버와 운영 서버는 운영체제, IP 주소, 개발 DB와 운영 DB 등 서로 다르게 설정된 파일들이 존재한다. 이러한 파일들은 변경될 경우, 서비스에 심각한 오류를 발생시킬 수 있기 때문에 자동 업데이트 시, 적용되면 안 되는 항목들을 설정한다.

제외 파일 설정

구분	대상	아이피	경로	내용
1	운영서버	192.168.0.12	/home/jpr/www/WEB-INF/classes/com/properties/	
2	운영서버	192.168.0.12	/home/jpr/www/WEB-INF/tst/cert.cer	
3	운영서버	192.168.0.12	/home/jpr/www/tst/ils/	
4	운영서버	192.168.0.12	/home/jpr/www/keyboard/test.exe	

그림 4. 제외 파일 설정

해당 항목들은 자동 업데이트 실행 시, 파일들이 변경되지 않으며, 이 외에 업데이트 준비 폴더와 백업 폴더도 설정 가능하다. 다만, 보안 측면인 이유로 셸 스크립트와 같은 실행 명령어들은 설정할 수 없다.

모든 설정이 완료되면, 자동 업데이트를 수행할 일정

을 등록한다. 앞서 설명한 개발 서버와 운영 서버 간 설정을 마치면, [그림 5]와 같이 자동으로 일정 관리 화면에 해당 일정들이 나타난다. 기본 상태는 자동 업데이트를 사용하지 않음 상태이며, 이 상태에서는 자동 업데이트가 수행되지 않는다. 실행 버튼을 클릭하면, 즉시 업데이트를 실행한다. 사용자가 특정 날짜 또는 시간으로 자동 업데이트의 일정을 설정하고자 한다면, 수정 버튼을 클릭하여 일정 설정 화면으로 이동한다.

일정

구분	경로	내용	일정	실행	수입/삭제
1	D:/project/www/ -> /home/jpr/www/	* 소스코드 업데이트	-	<input type="button" value="실행"/> <input type="button" value="삭제"/> <input type="button" value="추가"/>	
2	D:/project/www/DATA/ -> /home/jpr/www2/ksunboard/		매월 0200	<input type="button" value="실행"/> <input type="button" value="삭제"/> <input type="button" value="추가"/>	

그림 5. 작업 일정

작업 일정의 설정 화면에서는 매월, 매주, 매일의 시간을 1시간 단위로 등록이 가능하다. 다만, 같은 일정에 대하여 중복 등록은 불가능하며, 특정 날짜의 시간으로도 수행 시간을 등록할 수 있다. 설정한 일정을 취소하고 싶을 때는 일정을 “사용 안 함” 상태로 변경하거나 삭제 버튼을 클릭하면, “사용 안 함” 상태로 변경된다.

서비스가 실행되고, 모든 작업이 완료되면, 실행 명세에서 결과를 확인할 수 있다. 결과에는 각 업데이트의 수행일과 어떤 업데이트를 어떻게 진행했는지, 업데이트의 성공 여부를 확인할 수 있다. 실행 명세의 경로를 클릭하면, 상세 명세로 이동한다. [그림 6]은 실행 명세의 상세설명을 나타내며, 업데이트 1건에 대한 상세 명세가 표기된다. 어떤 파일과 폴더들이 적용되었는지 기록을 보여주며, 업데이트가 실패할 경우, 어떤 파일까지 성공했고, 어떤 파일을 적용하다가 실패했는지 보여준다.

실행 내역 (상세)

구분	일자	경로	내용	행동	성공여부
2	2020.11.27 19:23	D:/project/www/ -> /home/jpr/www/	* 소스코드 업데이트	수동	성공

그림 6. 실행 내역 (상세)

2. 실험 및 비교

본 연구는 기존에 사람이 작업하던 과정을 프로그램이 대신 작업을 하도록 구현한 시스템으로 사람이 작업하는 것과의 차이를 비교하고자 총 10회 걸쳐서 비교 실험을 진행하였다[9]. [표 7]과 [그림 7]은 실험 결과를 각각 표와 차트로 표현한 결과이다. 실험에는 FTP로 접속하여 20개의 파일이 들어있는 폴더를 운영 서버의 특정 폴더로 업로드하고, 서버를 재실행하는 데 걸리는 시간을 측정하였으며, 매 실험의 횟수를 증가할 때마다 20개의 파일이 들어있는 폴더의 개수를 1개씩 추가하였다. 그 결과 1회부터 시스템이 사람보다 작업을 하는 속도가 빠른 것을 확인할 수 있었으며, 사람은 평균 52.6의 결과값이 나왔으며, 시스템은 평균 23.9의 결과값이 나왔다, 이를 통해 사람이 하는 작업이 시스템보다 2.2배 느리다는 것을 알 수 있다.

특히, 파일의 수가 많을수록 사람과의 차이가 더욱더 명확하게 대조된다. 사람과 시스템 각각의 표준편차가 23 대 11로 시스템의 변동 폭이 사람보다 1/2배다 적음을 의미하며, 파일의 수가 배로 증가하기 때문에 1/2이라는 수치는 증가하는 폭이 고르며 안정성이 높다는 것을 나타낸다. 이는 작업을 하는 동안 시스템은 파일의 증가로 인한 작업 속도의 증가 외에는 다른 영향을 보이지 않는다고 볼 수 있으며, 실제로 사람의 경우 실험 8 ~ 9회차 작업 중 마우스 클릭을 잘못하는 실수를 범하였고, 결과에 그대로 반영이 되었다.

표 7. 작업 속도 비교

회차	1	2	3	4	5	6	7	8	9	10
사람	7	11	14	19	22	27	30	34	36	39
시스템	14	22	33	46	53	62	69	71	74	82

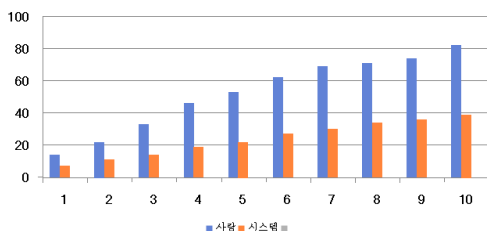


그림 7. 작업 속도 비교

V. 결론

본 논문에서는 기존에 사람에 의해 수동으로 진행되었던 웹사이트의 업데이트를 자동으로 구현하는 시스템을 제안했다. 기존에 사람이 업데이트의 경우, 올려야 할 파일의 체크부터 운영 서버의 파일 백업, WAS의 중지부터 재실행까지 각 과정을 사람이 일일이 작업해야 했다. 이로 인하여 중간에 올려야 할 파일을 잘못 올리거나, 반대로 누락하는 경우, 또는 변경되면 안 되는 파일을 변경하는 경우 등 사람에 의해 발생 가능한 실수가 다양하게 존재했고, 이를 알아차리거나 새로 작업하는 데까지 추가적인 시간이 소요되었다. 또한, 전반적으로 단순한 작업대비 많은 노동력을 소모하기도 한다. 단순히 파일을 올리고 서버를 재실행하는 작업이지만, FTP 프로그램을 실행하고, 서버의 각 폴더를 찾아다니며, 파일을 백업하고, 업데이트할 파일을 올리고, 수정한 파일들을 기록하고, SSH를 실행하여 서버를 재실행하는 과정은 항상 작업자가 실수하지 않기 위한 긴장감 속에서 작업이 진행되기 때문에 전체적인 작업 속도가 늦어지며, 긴장으로 인한 피로감도 발생한다[10]. 반면 제안하는 시스템은 지정된 시간 또는 버튼 클릭 한 번으로 이러한 일련의 과정을 자동으로 처리해주며, 시스템이 교체할 파일들을 정확한 알고리즘에 의해 자동으로 지정하고, 지정한 파일들만 수정하기 때문에 클릭 실수 또는 파일 누락 등의 실수에 대한 위험 요소도 없다. 특히, 앞서 보여준 실험 결과와 같이 같은 조건에서 사람이 하는 작업보다 빠른 속도를 보여준다. 덧붙여 작업 과정을 모두 기록하기 때문에 시간이 지나도 언제든지 이전 작업의 확인이 가능하며, 이전 파일들을 백업도 지원하므로 시간이 지난 후 문제가 발견된 경우에도 쉽게 복원을 할 수 있다.

이에 본 연구가 기존과 차별되는 부분은 젠킨스와 같은 솔루션들이 외부의 솔루션, 별도의 서버, 외부의 여러 플러그인을 사용하는 것과 달리 기존 웹사이트를 만들 때 사용한 전자정부 프레임워크로 만들어진 시스템이기 때문에 기존의 웹사이트와 이질성이 없으며, 전자정부 프레임워크에 하나의 기능으로 통합할 수 있다.

지난 10여 년간 정부에서 공공사업의 통일성과 공정성을 위하여 전자정부 프레임워크를 도입하였다. 이는

전자정부 프레임워크만 사용할 줄 알면, 어느 기관의 홈페이지든지 어떤 사람, 어떤 업체가 와서 작업을 해도 같은 결과를 낼 수 있게 되었고, 이는 곧 기존 업체와 인원에 대한 종속성을 제거하여 인원과 업체 선정에 대하여 공정한 심사를 할 수 있는 기반이 되었다[11].

그러나, 운영 서버를 관리하고 소스 코드를 배포하는 것에 있어서는 가이드 라인만 있을 뿐, 모든 재량이 담당자에게 있다. 이는 각 기관과 업무자에 따라 결과물의 차이가 발생한다. 만약, 조금 더 많은 연구가 진행되어 전자정부 프레임워크 안에 이러한 배포 기능이 내장된다면, 정부 차원에서도 통일된 매뉴얼과 복구 대응 방법 등 현직에 종사하는 사람들에게 좀 더 많은 편의가 되돌아올 것으로 생각한다. 산업혁명 이후, 우리는 사람보다 기계가 일을 하는 게 빠르고 효율적이며 편하다는 것을 상식으로 알고 있다. 이러한 시대에 대부분의 사용자가 시스템에 의한 배포보다 수작업으로 배포하는 것은 분명 문제라고 볼 수 있다. 현재 공공기관에서 가장 많이 사용되는 전자정부 프레임워크의 일부 기능으로 이러한 배포 기능이 나온다면 충분히 작업자에게 있어서 효율적이고, 편한 노동 환경을 제공할 수 있을 것이다.

하지만, 아직 안정성에 많은 한계가 있으므로 상용되기까지는 많은 연구가 필요하다. 먼저 백업된 파일을 복구하는 방법이 부족하다. 모든 업로드에 백업을 병행하는 이유는 지금 또는 언젠가 문제가 발생하였을 때 복구하기 위해서이다. 그러므로 자동으로 파일을 백업하는 것은 물론 문제가 생겼을 때 백업된 파일을 복구하는 기능도 자동화가 필요하다. 다음은 서버에서 자동으로 생성되거나 누군가 급하게 수정한 파일들이 있을 때, 이름 감지하고 적절한 대응할 수 있도록 알고리즘이 필요하다. 일부 웹사이트의 경우, 외부에 자료 제공하기 위하여, 자체적으로 파일을 만든다. 이렇게 생성된 파일들을 시스템이 기존의 개발 서버와 비교하여 잘못된 명령을 실행할 수 있다. 이와는 반대로 상황에 따라 항상 정해진 규칙으로 업데이트가 진행된다고 장담할 수 없다. 작업자가 여럿일 경우, 운영 서버에서 바로 작업을 하고, 작업 내용이 다른 작업자들에게 전달이 안 될 경우도 있다. 이러한 상황에 대하여, 어떻게 처리를 진행할지 시나리오를 구성하고 이에 대응하는 알고리

즘을 개발하는 것이 필요하며, 향후 연구 방안은 이러한 안정성의 한계를 극복하는 것을 목표로 한다.

참 고 문 헌

- [1] 이한식, *웹플랫폼의 진화와 개인화 서비스에 관한 연구*, 서원대학교, 석사학위논문, 2009.
- [2] 김은경, *웹을 기반으로 한 장난감 도서관 대여 및 반납 시스템*, 충북대학교, 석사학위논문, 2019.
- [3] Wiki, "패치"- [https://ko.wikipedia.org/wiki/패치_\(컴퓨팅\)](https://ko.wikipedia.org/wiki/패치_(컴퓨팅))
- [4] 가비아 라이브러리, "FTP를 이용하는 간단한 웹페이지 업로드"- <https://library.gabia.com/contents/main/2560/>
- [5] Wiki, "시큐어_셸"- https://ko.wikipedia.org/wiki/시큐어_셸
- [6] 양눈, "[LINUX] 파일의 해시값 확인하기 (MD5, SHA 1)"- <https://yangnoon.tistory.com/36>
- [7] 뭐하라, "테이블 비교 후 같지 않은 값 출력,"- <https://2ssoosike.tistory.com/262>
- [8] 알짜배기 프로그래머, "ORACLE 테이블 비교, 교집합 차집합 데이터 추출,"- <https://aljjabaegi.tistory.com/475>
- [9] 김주성, *웹사이트 구조, 웹사이트 정보량, 파일 복잡도가 사용성에 미치는 영향*, 충북대학교, 석사학위논문, 2004.
- [10] 임윤수, *웹 기반 장난감 거래 시스템 설계 및 구현*, 충북대학교, 석사학위논문, 2019.
- [11] 임현정, *효율적인 전산 장비 관리 시스템 설계 및 구현*, 충북대학교, 석사학위논문, 2019.

저 자 소 개

황 대 현(DaeHyeon Hwang)

정회원



- 2014년 2월 : 배재대학교 게임공학과(공학사)
- 2019년 3월 ~ 현재 : 충북대학교 전자정보공학과 석사과정

〈관심분야〉 : 안드로이드, 네트워크 통신, 웹 사이트, 데이터베이스 시스템 등

유 재 수(JaeSoo Yoo)

종신회원



- 1995년 2월 : KAIST 전산학과(공학 박사)
- 1995년 2월~1996년 8월 : 목포대학교 전산통계학과 전임강사
- 1996년 8월 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 정교수

〈관심분야〉 : 데이터베이스 시스템, 멀티미디어 데이터베이스, 센서 네트워크, 바이오인포매틱스, 빅데이터 등