

FTP 어댑터를 이용한 다중프로토콜 연동 및 모니터링 시스템 구현*

양준모, 박철형, 이수진, 백세인, 송양의, 이용규
동국대학교 컴퓨터공학과-서울
e-mail: j0238@naver.com

Implementation of a Multiprotocol Linkage and Monitoring System Using FTP Adapters

Joon Mo Yang, Cheol Hyung Park, Su Jin Lee, Se In Baek,
Yang-Eui Song, Yong Kyu Lee
Department of Computer Science and Engineering, Dongguk University-Seoul

요 약

다중 서버 환경에서 파일전송 프로토콜들이 혼용되어 사용되면서 프로토콜 간의 연동이 원활하지 않으며, 파일 송수신 로그 관리의 부재로 생기는 버전 관리의 문제점이 있다. 따라서 본 논문에서는 FTP(File Transfer Protocol) 어댑터를 이용하여 다중프로토콜 연동을 해결하고, 파일 송수신 로그를 관리하는 파일 버전 모니터링 시스템을 구현하고자 한다. 각 서버에서는 어댑터를 통해 중앙서버로부터 각 서버의 가용 프로토콜 정보를 받아 프로토콜 혼용 문제를 해결하고, 파일 송수신 로그를 중앙서버로 송신하여 파일 버전 관리의 문제를 해결한다. 본 시스템을 통해 프로토콜의 연동과 파일 송수신 로그를 관리함으로써 서버의 효율적인 관리를 할 수 있다.

1. 서론

인터넷이 발달하면서 많은 전송규약들이 생겨났지만 아직도 많은 서버에서 대용량파일 전송을 위해 FTP를 사용하고 있다[1]. 그러나 FTP는 보안의 취약점을 가지고 있어 보안이 강화된 전송을 보장하는 S_FTP 및 공인인증서로 신원을 보장하는 SSL_FTP 등이 대두 되었다[2]. 현재는 이러한 프로토콜들이 혼용되어 사용되면서 서버가 FTP, S_FTP, SSL_FTP를 모두 지원하거나 특정 프로토콜만을 지원하고 있다. FTP, S_FTP, SSL_FTP가 혼용되어 사용되는 다중 서버 환경에서는 각각의 서버가 어떤 프로토콜을 지원하는지 관리가 되지 않는다면 서버 간의 연동 문제가 발생할 수 있다.

현재 FTP를 이용한 서버 시스템에서는 파일의 송수신 정보를 따로 기록하지 않는 경우가 대부분이다. 이러한 서버 시스템에서는 각 서버에서 관리되는 파일에 대해 파일 송수신 완료 여부를 확인할 수 없다. 또한 다중 서버에서의 파일에 대한 최신 버전을 확인하기 어려워 파일의 버전을 관리하는데 문제가 생길 수 있다.

본 논문은 위와 같은 문제를 해결하기 위해 다중 서버환경에서 FTP 어댑터를 이용한 프로토콜 연동 및 모니터링 시스템을 제안하고자 한다. 본 논문에서 제안하는 시스템

에서는 파일 송수신을 위한 서버에 데몬 형태의 어댑터를 둔다. 또한 모니터링을 위하여 어댑터와 연결된 서버들의 정보(IP, PORT 등)를 가지고 있는 중앙서버로 구성한다. 각 서버의 어댑터들은 파일을 송수신할 다른 서버의 가용한 프로토콜 정보를 중앙서버로부터 받아온다. 각 서버가 다른 서버에 파일을 송수신하기 위해 프로토콜을 확인한 후 연동하게 된다.

또한 어댑터는 서버 내 파일의 송수신이 이루어질 때마다 로그를 중앙서버로 전송한다. 본 시스템에서 로그는 각 서버의 프로토콜 연동 정보 및 파일 버전 정보를 의미한다. 로그 정보를 관리하여 중앙서버는 각 서버의 파일 변경을 모니터링하면서 파일 버전을 관리할 수 있다.

2. 관련연구

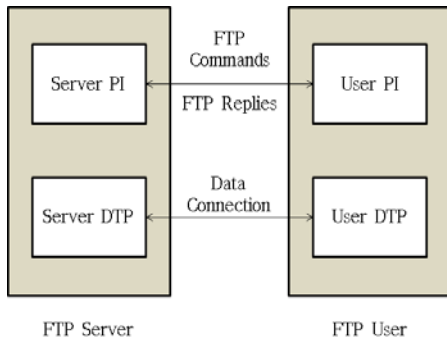
2.1 FTP

FTP란 파일 전송 규약의 약자로 서로 다른 컴퓨터 환경에서 파일을 빠르게 송수신할 수 있게 지원하는 규약이다[3].

(그림 1)은 FTP의 연결 구조를 나타낸다. PI(Protocol Interpreter)를 통해 사용자는 FTP Commands로 명령을 발생시킨다. 명령을 수신한 서버는 FTP Replies를 통해 명령에 대해 응답한다. DTP(Data Transfer Protocol)는 서버와 사용자간의 실제 파일 교환이 이루어질 수 있도록 한다[6].

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 서울어코드 활성화지원사업의 연구결과로 수행되었음(IITP-2016-R0613-16-1147)

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원 사업의 연구결과로 수행되었음(R7116-16-1014)

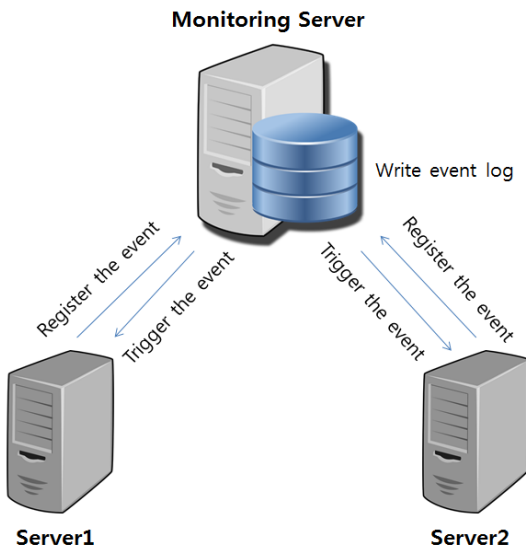


(그림 1) FTP 연결 구조

2.2 서버 모니터링 시스템

서버 모니터링 시스템은 서버의 CPU, Disk, Memory, 네트워크, 서버의 상태 등의 정보를 보여주고 관리할 수 있도록 한다[4]. 또한 파일 전송 예약, 서버 부하 균형, 상태 모니터링 등의 역할을 한다. 서버 모니터링 시스템은 서버의 정보를 항상 기록하고 분석한다[5].

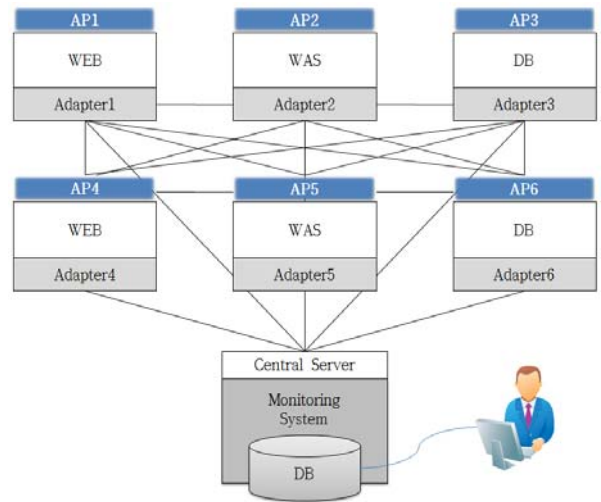
(그림 2)에서는 서버 모니터링 시스템의 일반적인 구조를 나타낸다. 각 서버는 특정 이벤트가 발생할 때(Trigger the event)마다 모니터링 서버로 로그를 전송(Register the event)하고 된다.



(그림 2) 서버 모니터링 시스템

3. 다중프로토콜 연동 및 모니터링 시스템

(그림 3)은 본 논문에서 제안하는 시스템의 구성 예시를 도식화한 것이다. 각 서버(AP)는 그 서버의 운용 목적에 따라 WEB, WAS, DB서버로 구분되지만, 모든 서버에는 동일하게 데몬 형태의 어댑터가 설치된다. 어댑터들은 각 서버를 모니터링하기 위해 중앙서버(Central Server)와 연동한다. 중앙서버에서는 각 서버의 IP, PORT, 가용 프로토콜, 파일 버전 정보 등을 관리한다. 중앙서버는 서버 간 프로토콜의 조정, 파일 버전에 대한 모니터링을 할 수 있다.

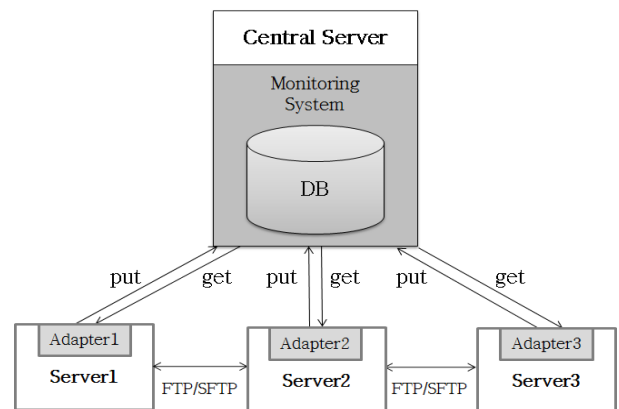


(그림 3) 시스템 구성 예시

3.1 어댑터를 이용한 다중서버와 중앙서버의 연결

여러 서버가 운용되는 다중 서버 시스템에는 하나의 중앙 관리시스템인 중앙서버가 있다. 각 서버에는 시스템 백그라운드에서 돌아가는 데몬 형태의 어댑터를 설치하고, 어댑터는 각 서버와 중앙서버를 연결한다.

(그림 4)에서는 각 서버와 각 서버 내에 설치된 어댑터들, 어댑터들과 연결된 중앙서버 간의 연결 구조를 나타내고 있다. 어댑터는 각 서버의 파일 송수신 완료여부를 중앙서버에 전송한다(put). 또한 어댑터는 서버 간 프로토콜 연동 시 중앙서버에서 다른 서버의 가용한 프로토콜 정보를 받아온다(get).

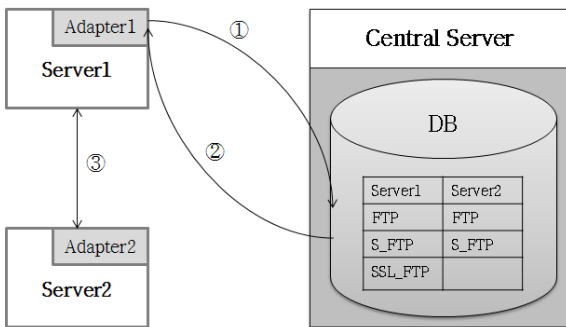


(그림 4) 중앙서버와 어댑터 간 연결구조

3.2 두 서버 간 전송 프로토콜 변경

각 서버 내에 설치된 어댑터들은 중앙서버의 데이터베이스와 연결되어 다른 서버의 가용 프로토콜 정보를 수신하게 된다. 두 서버가 연동하는데 프로토콜이 변경될 경우, 해당 서버의 어댑터는 중앙서버의 데이터베이스에 다른 서버의 가용프로토콜 정보를 받아와 다른 서버와의 프로토콜 연동을 유지할 수 있다.

(그림 5)에서는 두 서버 간의 전송 프로토콜 변경 과정을 도식화한 것이다. Server1과 Server2가 기존에 FTP방식의 연동을 유지하다가 S_FTP로의 변환이 될 때의 과정은 다음과 같다. ①에서는 Server1이 Server2와의 프로토콜을 FTP에서 S_FTP로 변경하기 위해 Server2의 가용 프로토콜 목록을 요청한다. ②에서 중앙서버는 Adapter1에 Server2의 가용 프로토콜 목록을 송신한다. ③에서 Server2의 가용 프로토콜 목록 중 S_FTP를 확인한 Adapter1은 Server1과 Server2의 연동 프로토콜을 FTP에서 S_FTP로 변경할 수 있다.



(그림 5) 두 서버 간의 전송 프로토콜 변경 과정

3.3 다중서버 파일 버전 모니터링 방법

다중서버 관리 차원에서 각 서버의 연동 현황 및 파일 전송 스케줄을 중앙서버에서 관리한다. 각 서버 내 어댑터들은 연결된 서버 정보와 파일 송수신 정보 등을 중앙서버 내 데이터베이스에 업데이트 하게 된다.

<표 1> 각 서버에서 중앙서버로 전송하는 로그 속성

로그 속성	설명
Server_id	해당 서버의 구분 ID
Adapter_id	해당 서버의 어댑터 ID
Port_num	해당 서버의 포트 번호
Server_ip	해당 서버의 IP 주소
Client_id	해당 접속 클라이언트 ID
Schedule_time	서버 파일 전송 시간
Protocol_type	서버의 프로토콜 유형

<표 1>에서는 각 서버의 어댑터들이 중앙서버로 전송하여 관리되는 로그 정보를 나타내고 있다. 로그 속성은 각 구분 id값, IP, PORT, 자원 사용량, 파일 전송 시간, 프로토콜이 있다. 어댑터는 해당 서버에서 파일 송수신이 발생할 경우, 로그를 중앙서버로 송신하게 된다. 중앙서버에서는 수집한 로그를 활용하여 서버 간 프로토콜 연동 현황을 관리할 수 있다. 또한 이벤트가 발생할 경우뿐만 아니라 일정간격의 시간동안 서버의 상태 정보가 중앙서버에 업데이트된다. 각 서버의 파일 송수신 정보들은 전송 시 중앙

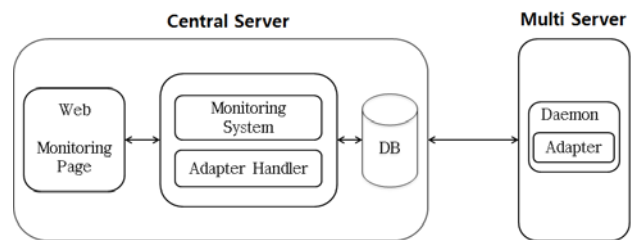
서버에 업데이트 된다. 이를 통해 중앙서버의 데이터베이스에 기록된 파일의 마지막 파일 송수신 정보와 최신 파일 송수신 정보를 비교 할 수 있다. 이 정보들을 비교하여 파일의 버전 현황을 실시간으로 확인할 수 있다.

4. 설계 및 구현 결과

4.1 시스템 구성도

본 논문에서 제안하는 전체 시스템의 구성도는 (그림 6)과 같다. 관리 대상의 다중 서버들에는 JAVA 데몬 형태의 Adapter가 설치한다. 다중 서버를 관리하는 중앙서버에는 각 서버 내 어댑터들을 관리하는 Adapter Handler 부분과 실질적인 모니터링 시스템 부분으로 나누어져 있다.

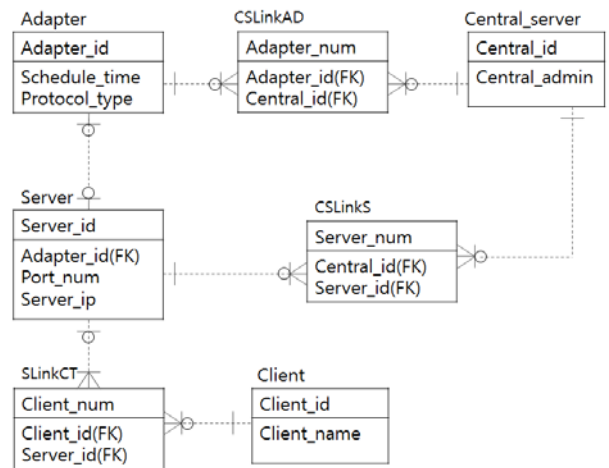
중앙서버 관리자는 모니터링 시스템에 접속하여 서버 관리 현황을 실시간으로 확인할 수 있다.



(그림 6) 전체 시스템 구성도

4.2 데이터베이스 설계

본 논문의 시스템에서는 다중서버에서 각 서버들의 정보를 관리하고 로그를 수집하기 위한 데이터베이스 설계가 필요하다. 또한 각 서버의 기본적인 모니터링 속성 및 프로토콜 속성을 관리한다.



(그림 7) 서버 관리 논리적 모델

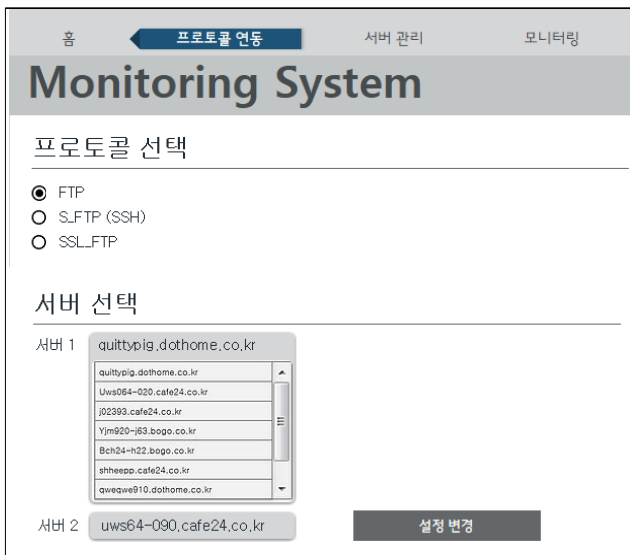
본 시스템의 서버 관리 논리적 모델은 (그림 7)과 같다. 중앙 서버에서 관리하는 서버와 어댑터 정보를 가지는 Central_server, 어댑터의 정보를 저장하는 Adapter, 각 서버의 상태 정보를 저장하는 Server, 서버에 접속한 사용자

를 나타내는 Client 개체(Entity)가 있다.

Adapter의 Protocol을 통해 각 어댑터의 전송규약을 알 수 있고, Schedule_time를 서버 파일 전송 시간을 알 수 있다. 어댑터와 연결되어진 서버를 Server_id를 통하여 알 수 있고 해당 서버의 정보를 수집한다.

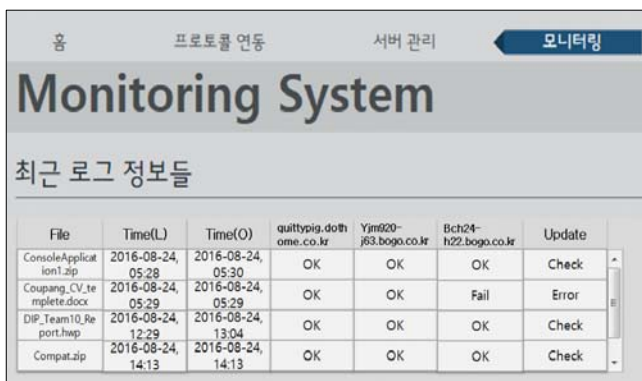
4.3 구현 결과

중앙서버에서 관리되는 모니터링 시스템을 구현하였다. (그림 8)은 본문에서 설명된 두 서버 간 프로토콜 연동 설정 화면을 보여주고 있다. 중앙서버에서 두 서버를 선택하여 두 서버 간 이용할 프로토콜 연동을 수동으로도 설정할 수 있다.



(그림 8) 서버 간 연동 프로토콜 수동 설정 화면

(그림 9)는 최근 수집된 로그 정보의 모니터링 화면이다. 마지막 파일 전송 시간과 최신 파일 전송 시간을 확인 할 수 있고 이를 통하여 업데이트 여부를 확인할 수 있다.



(그림 9) 서버별 파일 버전 정보 모니터링 화면

5. 결론

기존의 다중 서버 환경에서 파일전송 프로토콜들이 혼용되어 사용되면서 각 서버 간 연동 문제가 발생하였다. 또한 파일 송수신 정보에 대한 기록이 제대로 되지 않으면서 서버 내 파일 버전에 대한 관리가 효율적으로 이루어지지 않는다는 문제점이 있었다.

본 논문에서는 다중 서버 환경에서 각 서버가 가지는 프로토콜 정보 및 서버 정보를 데이터베이스에 저장하여 관리한다. 또한 각 서버 내에 어댑터를 설치하여 실시간으로 서버 정보와 파일 송수신 로그를 수집하는 시스템을 구현하였다. 다중 서버 관리를 위하여 중앙서버를 두고 각 서버마다 설치된 어댑터로 연동한다. 중앙서버 관리자는 중앙서버에서 모니터링 시스템을 통해 실시간으로 상황과 각 서버에서 시각화 된 데이터를 파악하여 효율적인 관리를 할 수 있다.

향후 서버 모니터링 시스템을 개선하기 위하여 서버 간의 파일 전송 예약 기능 추가에 관한 연구를 진행할 것이다.

참고문헌

- [1] 안재원, “보안 기능을 지원하는 파일 전송 프로토콜의 설계 및 구현”, 한국산학기술학회논문지, 제 15권, 제 5호, pp.3086-3092, 2014.
- [2] Rolf Oppliger, “SSL and TLS: Theory and Practice”, ARTECH HOUSE, pp.75-79, 2009.
- [3] Jon Postel, Joyce Reynolds, “File Transfer Protocol (FTP)”, IETF, RFC 959, 1985.
- [4] 박규식, “클라우드 시스템에서 효과적인 통합모니터링 도구의 연구”, 단국대학교 정보미디어대학원, 석사학위논문, 2012.
- [5] 김상준, “오픈소스를 활용한 서버 모니터링 기법”, 숭실대학교 대학원, 석사학위논문, 2014.
- [6] IBM Knowledge center, <https://www.ibm.com/support/knowledgecenter>