

# 오픈소스 기반의 네트워크장비 관리시스템 구현

정상우\*, 신대민\*\*, 서정현\*\*, 이상현\*, 이미진\*, 김기일\*

\*충남대학교 컴퓨터공학과

\*\* (주)큐버

e-mail: [kikim@cnu.ac.kr](mailto:kikim@cnu.ac.kr)

## Development for Network Device Management System through Open Source

SangWoo Jung\*, DaeMin Shin\*\*, JeoungHyun Seo\*\*, SangHyun Lee\*,

MiJin Lee\*, Ki-Il Kim\*

\*Dept of Computer Science and Engineering, Chungnam National University

\*\*Quber Co., Ltd.

### 요 약

대부분의 관리 시스템은 서버를 통해 구현되며 클라이언트가 증가함에 따라 반드시 증설이 요구된다. 하지만, 물리적인 서버의 단순한 추가는 원하는 동작이 수행을 보장하지 못할 수도 있으며 여러 설정의 복잡성과 긴 서버구매시간동안 서비스 제공이 어렵다. 본 논문에서는 확장성 있는 장비관리 서버를 도커 컨테이너로 구성함으로써 서버증설이 필요한 경우 최소한의 설정으로 새로운 도커 컨테이너를 추가하는 함으로써 서버증설이 완료할 수 있는 오픈 소스 기반의 시스템의 인터페이스 구현 방안을 설명한다.

## 1. 서론

셋탑, 모뎀 등의 네트워크 장비들의 효율적인 관리를 위한 시스템이 반드시 필요하다. 이러한 관리시스템에서 가장 큰 기술적 요소 중에 하나는 네트워크 장비들과 연동하여 장비들로부터 정보를 수신하여 이를 처리하는 모니터링 프로토콜을 처리하는 모듈이다. 하지만, 서버와 연동되는 장비들이 늘어남에 따라 하나의 서버가 처리할 수 있는 최대 클라이언트수에 도달하게 된 경우 결국 서버증설을 필요로 하게 된다. 대부분의 이러한 증설의 경우 단순히 물리적인 서버를 몇 대 추가하는 것으로 끝나지 않고 많은 설정과 실제 물리적인 서버를 구매하기까지의 많은 시간이 요구된다. 따라서, 끊임 없는 서비스를 제공하기 위해서는 이러한 문제점을 해결하여야만 한다.

이를 위해 본 논문에서는 빠른 증설요구에 대응하기 위하여 핵심이 되는 장비관리서버의 서비스 모듈을 도커 컨테이너로 구현하는 방법을 제안한다. 이러한 도커 컨테이너들을 추가함에 있어 최소한의 설정을 통해 서비스 구현이 가능하며 이를 위한 각 컴포넌트간의 인터페이스와 구현 기술을 설명한다.

## 2. 네트워크장비 관리시스템

본 논문에서 제안하는 네트워크 장비 관리 시스템은 크게 장비 관리 엔진, 로드 밸런서, 관리자용 웹 포털, 데이터베이스, 컨테이너 관리시스템, 파일서버로 구성된다. 각각의 컴포넌트들에 대한 설명은 다음과 같다.

### 2.1 장비 관리 엔진

네트워크 장비를 관리하는 모듈을 통하여 구현된다. 현재 장비들을 관리하는 프로토콜은 대표적으로 TR-069, SNMP, LwM2M 등이 있다[1]. 따라서 각 장비에서는 하나의 관리 프로토콜을 사용하는 서버와 연동하게 된다. 본 논문에서의 장비 관리 엔진이란 언급된 관리프로토콜을 서비스하는 모듈을 지칭한다. 또한, 대상 프로토콜중 TR-069 프로토콜을 구현하기 위하여 오픈소스인 FreeACS를 사용한다.

### 2.2 로드 밸런서(Load Balancer)

제안 관리 시스템은 프로토콜에 따라 다양한 종류의 다수의 장비 관리 엔진을 연동함으로써 다수의 장비들이 연동된다. 따라서, 많은 수의 엔진들의 구동이 필요하다. 따라서, 부하의 집중을 막기 위하여 로드 밸런서가 필요하다. 고가의 L4 장비가 사용될 수도 있지만 본 논문에서는 오픈 소스인 HAProxy를 사용한다.

### 2.3 관리자용 웹 포털(Web Portal)

본 시스템은 일종의 SaaS(Service as a Service)처럼 동작하도록 설계된다[2]. 따라서, 새로운 장비 관리 엔진 추가 등의 시스템 작업을 관리자용 웹 포털에서 진행할 수 있도록 구성한다. 클라우드가 아니라 온-프레미스 타입의 서버를 구축하는 경우에는 물리서버를 관리 네트워크

크 안에 설치해 놓는 작업이 필요하겠지만 물리서버에 임의의 장비 관리 엔진을 웹 포탈을 통해 자유롭게 설치 및 삭제할 수 있도록 구성함으로써 편리성을 향상시킬 수 있다.

2.4 데이터베이스

데이터베이스는 오픈소스인 MySQL을 사용하며 두 가지 종류가 사용된다. 첫 번째는 장비정보들이 기록되고 웹 포탈이 접근하는 작업용 데이터베이스이고 두 번째는 통계정보와 장비문제사항 분석 등의 작업을 위한 비즈니스 데이터베이스이다. 작업용 데이터베이스에서 레코드의 변경(추가/삭제/변경)이 발생하는 경우 이는 곧바로 비즈니스 데이터베이스로 동기화된다.

두 개의 데이터베이스로 분리함으로써 장비와 연동되는 데이터베이스에서 다른 작업(통계분석 작업 등)을 처리함으로 발생하는 지연을 최소화 할 수 있다.

2.5 컨테이너 관리 시스템

본 논문에서 주요 구성요소들을 관리하기 위하여 모든 요소들은 모두 도커 컨테이너화로 관리한다. 이러한 다수의 도커 컨테이너를 관리하기 위한 시스템으로는 Kubernetes를 사용한다.

2.6 파일서버

펌웨어나 앱들을 다수의 장비들에게 효율적으로 배포하기 위해서는 이러한 배포된 파일들을 보관하기 위한 파일서버가 필수적이다. 이 서버들도 역시 도커 컨테이너로 구성하였으며 파일서버는 HTTP(S)서버와 (S)FTP서버 두 종류를 지원한다.

3. 컴포넌트관리 시스템 구현

2절에서 설명한 것과 같이 본 시스템은 많은 컴포넌트들로 구성되어 있기 때문에 장비관리를 위해서는 관리자가 작업한 작업내용(펌웨어 배포, 장비 설정 변경, 장비 상태 조회 등)을 처리하기 위해서는 이 컴포넌트들 간의 긴밀한 연동이 필수적이다.

본 시스템에서 모듈들간 긴밀한 연동을 위하여 Apache Thrift[3]을 사용하여 모듈간 인터페이스를 정의하였으며 크게 다음과 같은 세 종류의 인터페이스가 존재한다.

3.1 웹포탈과 메인 서비스와의 인터페이스 (IF-1)

관리자가 웹 포탈에서 작업한 명령(도커 컨테이너 관리, 장비관리, 장비 상태조회 등)을 메인 서비스에게 전달하기 위한 인터페이스로 <표 1>에서 정의된 인터페이스들을 가진다.

3.2 컨테이너 관리시스템과 컨테이너들의 인터페

이스(IF-2)

제안된 시스템은 도커 컨테이너 기반으로 구성되므로 컨테이너 관리에 대한 요청 인터페이스를 <표 2>와 같이 정의함으로서 장비관리엔진과 파일서버의 인스턴스를 관

인터페이스 이름	수행내용
StartQdmsService	전체서비스를 시작
StopQdmsService	전체서비스를 중지
ReportQdmsStatus	서비스내의 구성요소등의 상태를 조회
AddDmEngine	장비관리엔진을 추가
RemoveDmEngine	장비관리엔진을 삭제
StartDmEngine	장비관리엔진을 시작
StopDmEngine	장비관리엔진을 중지
QueryDmEngineStatus	장비관리엔진의 상태를 조회

<표 2> IF-2 인터페이스 목록

인터페이스 이름	수행내용
StartContainer	특정 도커 컨테이너를 수행
StopContainer	특정 도커 컨테이너를 중지
QuerySts	특정 도커 컨테이너의 상태를 조회

리한다.

<표 1> IF-1 인터페이스 목록

3.3 파일서버와 메인서비스간 인터페이스(IF-3)

관리자가 웹포탈을 통해 펌웨어를 배포한 경우 배포한 펌웨어 파일을 배포프로토콜의 종류 등에 따라 적절한 파일서버에 업로드 및 삭제하기 위한 인터페이스가 <표 3>과 같이 정의되고 구현된다.

<표 3> IF-3 인터페이스 목록

인터페이스 이름	수행내용
RegisterFile	파일을 서버에 등록
UnRegisterFile	등록된 파일을 서버에서 삭제
QuerySts	파일서버의 상태를 조회

4. 결론

본 논문에서는 장비관리서버를 도커 컨테이너로 구성하고 이 컨테이너들을 연동하여 하나의 장비관리시스템으로 만들기 위한 시스템 구조와 인터페이스를 제안하였다. 서버 인스턴스를 도커 컨테이너로 구현하여 운용함으로써 서버 증설이 필요한 상황에서 빠른 시간 내에 서버증설을 제공함으로써 유연하고 확장 용이한 장비관리서비스를 제공할 수 있으며 모든 시스템은 오픈소스를 통하여 구현 가능하다.

본 논문과 관련하여 향후 제안된 시스템의 구현과 이에 따른 검증 및 성능 분석이 이루어질 것이며 시스템 성능 향상을 위한 방안을 계속 연구할 것이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터

의 SW중심대학사업의 연구결과로 수행되었음  
(2015-0-00930)

### 참고문헌

- [1] Enrico Rerrera, Davide Conzon, Paolo Brizzl, Lucas L. Gomes, Marc Jentsch, Peeter Kool, “XMPP-based network Management infrastructure for agile IoT application deployment and configuration,” International ICIN Conference, March, 2016
- [2] Friendly-Technology, <https://www.friendly-tech.com/>
- [3] Apach Thrift Project, <https://thrift.apache.org>