

Open Stack 기반의 IoT 시스템에 대한 활용 방안

A Study on the Application of Open Stack on IoT systems

201011348 윤장웅, 201311269 김제현, 201311274 박기원

건국대학교 공과대학 소프트웨어융합학부 컴퓨터공학과

Yoon Jang Woong, Kim Je heon, Park Keewon
Konkuk University

지도교수 : 한선영

< 요약 >

본 논문은 최근 대한민국에서 가장 큰 문제점으로 떠오르고 있는 미세먼지를 측정하는 인프라를 만드는데 있어서, 외부 농도 외에 미세먼지 센서를 이용해서 내부 농도도 측정하여 비교를 하도록 하였고, 대규모 데이터를 처리하는 대부분의 IoT 서비스들이 클라우드 환경에서 돌아간다는 점을 감안해서 현재 대기업들이 많이 사용하기 시작한 Open Stack (IaaS 형태의 클라우드 컴퓨팅 오픈 소스 프로젝트)(1)을 사용한 프로젝트를 바탕으로 기술하였다.

I. 서론

사회적 관심도를 알 수 있는 구글 트렌드에 따르면 3월 중순 이후 미세먼지에 대한 관심도가 크게 증가했다고 한다. 3월초 구글 내 미세먼지 검색은 10포인트 수준이었지만 27일 현재 83포인트로 8배 이상 관심도가 높아졌다. 게다가 미세먼지 주의보가 상시화 되면서 최근 미세먼지 관련 제품들이 불티나게 팔리고 있다.

세계적으로 IoT 시장이 크게 성장하고 있는 상황에서 어떤 식으로 IoT가 이루어지고 개발되는지 한번 경험해보고자 해서 주제를 IoT로 정해보았다. IoT 중에서도 최근 폭발적으로 각광받고 있는 Home IoT에 주목했다.

서비스적인 측면 외에도, 요즘 대규모 데이터를 처리하는 대부분의 IoT 서비스들이 클라우드 환경에서 돌아간다는 점에 감안해서 기술적으로 OpenStack을 사용해 보기로 하였다.

II. 본론

1. 오픈스택

“ 오픈스택(OpenStack)은 IaaS 형태의 클라우드 컴퓨팅 오픈 소스 프로젝트이다. 2012년 창설된 비영리 단체인 OpenStack Foundation에서 유지, 보수하고 있으며 아파치 라이선스하에 배포된다.

오픈스택은 열린 설계와 개발을 지향한다. 커뮤니티는 6개월의 릴리즈 사이클로 개발을 진행하고 있다. 매 사이클의 기획단계에서는 오픈스택 디자인 서밋(OpenStack Design Summit)을 개최하여, 개발자 작업을 지원하고, 로드맵을 설정하고 있다.” (1)

기존 클라우드 시스템을 구축하기 위해서는 많은 비용이 드는 반면, 오픈스택은 오픈소스를 제공하기 때문에 저렴한 비용으로 클라우드 시스템을 구축할 수 있다. 그 장점을 살려 이번 프로젝트에 효율적으로 사용하였다.

본 프로젝트를 진행함에 있어 우분투 운영체제의 인스턴스를 2 개로 나누어 구축하였다. 하나는 서버용 인스턴스, 그리고 나머지 하나는 데이터베이스용 인스턴스이다. 차후 시스템의 규모가 커질 수 있는 점을 감안하여 두 가지 인스턴스로 분리하여 구축하였다. 그리고 얼마든지 인스턴스의 개수를 늘려 유연하게 시스템을 운영할 수 있다.

2. IoT

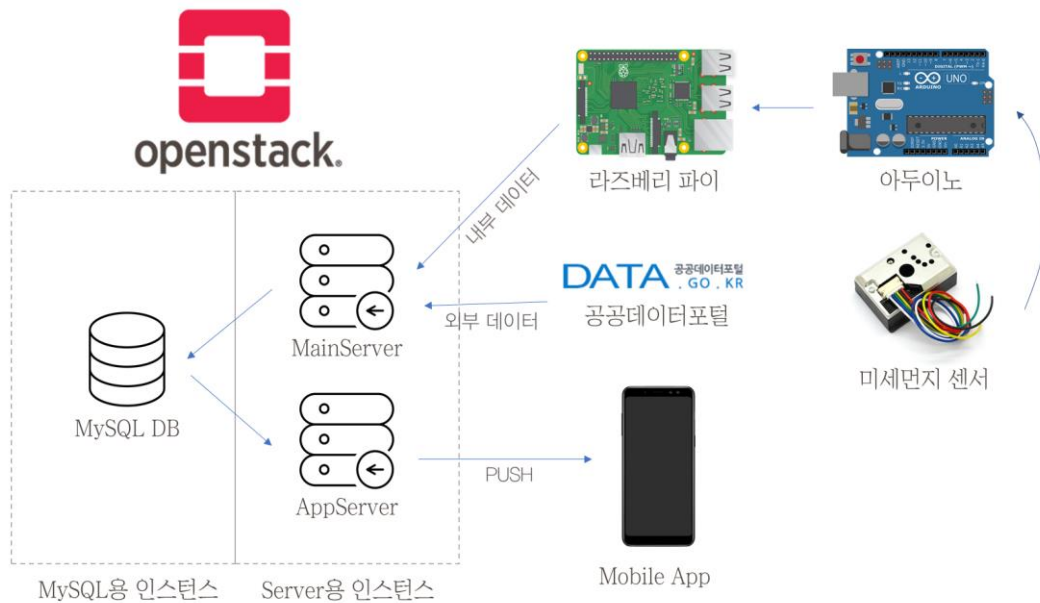
“ 사물인터넷(Internet of Things, 약어로 IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술. 즉, 무선 통신을 통해 각종 사물을 연결하는 기술을 의미한다. 인터넷으로 연결된 사물들이 데이터를 주고받아 스스로 분석하고 학습한 정보를 사용자에게 제공하거나 사용자가 이를 원격 조정할 수 있는 인공지능 기술이다. 여기서 사물이란 가전제품, 모바일 장비, 웨어러블 디바이스 등 다양한 임베디드 시스템이 된다. 사물인터넷에 연결되는 사물들은 자신을 구별할 수 있는 유일한 아이피를 가지고 인터넷으로 연결되어야 하며, 외부 환경으로부터의 데이터 취득을 위해 센서를 내장할 수 있다. 모든 사물이 해킹의 대상이 될 수 있어 사물인터넷의 발달과 보안의 발달은 함께 갈 수밖에 없는 구조이다.” (2)

본 프로젝트를 진행하면서 IoT 장비로는 라즈베리파이와 아두이노를 사용하였다. 두 가지 장비를 함께 사용한 이유는 라즈베리파이에서는 시중에서 미세먼지 센서를 구할 수 없었고, 아두이노는 무선통신을 지원하지 않기 때문에 이 두 가지 문제점을 함께 보완하기 위해 두 장비를 함께 사용하였다.

3. Mobile Application

IoT의 실용성을 높이기 위해 결과값을 모바일 애플리케이션으로 전송해서 현재 위치와 상관없이 실시간으로 확인할 수 있도록 구축하였다. 파이썬으로 앱 서버를 구축하고 안드로이드 모바일 애플리케이션과 통신한다.

4. 흐름도



-

- 오픈 스택의 경우, Server 용 인스턴스와 MySQL 용 인스턴스 2 개를 생성하였다.
- 기본적으로 OpenStack 에서 Server 용 인스턴스의 MainServer 및 AppServer, MySQL 용 인스턴스의 MySQL DB 가 돌아가고 있다.
- 미세먼지 센서의 경우 라즈베리 파이용 센서가 존재하지 않아서 아두이노용 센서를 구매하였고, 따라서 아두이노도 이용할 수 밖에 없었다.
- 라즈베리파이에서 'Arduino IDE' 를 통해서 아두이노를 제어하여, 미세먼지 센서로부터 집안 내부의 미세먼지 측정 값을 읽어오면, RaspberryPi 가 이를 OpenStack 에서 Server 용 인스턴스의 MainServer 로 보내게 된다.
- 이와는 별개로 MainServer 에서 공공데이터포털로부터 외부 미세먼지 측정 값을 읽어온다.
- MainServer 는 1 분마다 이 둘의 값을 읽어오게 되며, 이 값들을 MySQL 용 인스턴스의 MySQL DB 에 저장하게 된다.

- Mobile App 을 실행 시, Server 용 인스턴스의 AppServer 에 접속해서 데이터 요청을 받고, MySQL 용 인스턴스의 MySQL DB 에 저장된 가장 최근 측정값을 받아서 반환하게 된다.

III. 결 론

본 프로젝트에서는 테스트용으로 하나의 센서와 각각 하나의 라즈베리파이, 아두이노를 사용하고, 오픈스택 클라우드 환경에서 두 개의 인스턴스만을 구축하였다. 이를 발전한다면 가정에 맞게 디자인하여 서울특별시 취약계층(노인가구)이나 학교와 같은 어린이, 청소년들이 밀집해 있는 공공기관에 보급하고 실내 미세먼지를 측정하여 건강에 유의할 수 있도록 할 수 있다. 추가로 미세먼지 농도만 측정하는 것이 아닌 위치 정보를 수집하여 여러 요소를 분석해 미세먼지 발생의 원인을 파악하는 용도로 사용이 가능하다. 수집된 데이터는 기존에 존재하지 않은 통계로 다양한 분야에 사용이 가능하다.

만약 한 도시 전체로 확장되고 클라우드 시스템의 규모가 커지게 되어 서버의 부하가 폭주할 때 서버 인스턴스를 여러 개 늘리고 부하가 줄어들면 인스턴스를 줄이는 기능인 오토 스케일링 기능 및 이러한 서버가 여러 대일 때 접속 부하를 적절하게 분배하는 로드 밸런싱 기능을 구현하면 좋을 것이다.

IV. 참고문헌

- (1) 위키백과 오픈스택 :

<https://ko.wikipedia.org/wiki/%EC%98%A4%ED%94%88%EC%8A%A4%ED%83%9D>

- (2) 위키백과 IoT :

<https://ko.wikipedia.org/wiki/%EC%82%AC%EB%AC%BC%EC%9D%B8%ED%84%B0%EB%84%B7>