

IoT 빅데이터 저장 및 모니터링 플랫폼

이재호, 최 영, 김용성, 송병권, 강선미

서경대학교

{wogh325, cy, wiz, bksong, smkang}@skuniv.ac.kr

IoT Big data Storage & Monitoring Platform

Jaeho Lee, Young Choi, Yongseong Kim, Byungkwen Song, Sunmee Kang

Seokyeong Univ

요 약

오늘날에는 다량의 IoT 장비 및 센서가 광범위한 지역 및 공간에 배포되어 있다. 이처럼 여러 곳에서 동시다발적으로 발생하는 대량의 데이터를 목적에 맞게 효과적으로 취합하고 처리하는 시스템은 필수적이다. 본 논문에서는 동시다발적으로 발생하는 빅데이터의 저장, 모니터링 및 데이터 분석을 위한 시각화, 이 두 목적으로 구축된 플랫폼을 소개한다. 또한, 구축된 플랫폼에서 스트레스 테스트를 통해 다량의 IoT 장비들이 동시에 데이터를 전송하는 가정된 상황에서도 데이터가 정상 취합됨을 검증하였다.

I. 서 론

오늘날은 대량의 데이터가 존재하는 빅데이터(Big data) 시대이다. 데이터를 가상의 서버인 클라우드(Cloud)에 저장하고 빅데이터를 기반으로 학습시켜 인공지능(Artificial Intelligence) 모델을 만들고 학습된 모델을 IoT(Internet of Things) 시스템에 적용하는 움직임이 커지고 있다. 이처럼 데이터 처리 기술의 발전과 활용도가 커짐에 따라 빅데이터의 저장, 관리 및 분석을 위한 모니터링의 필요성도 함께 증대되고 있다.[1]

산업생태계에서 사용하고 있는 다량의 IoT 장비는 주기적으로 측정되어 시계열의 특징을 가지며 누적된 데이터 그리고 앞으로 쌓일 데이터가 방대하게 많은 상황이다. 또한, 데이터 발생지가 물리적으로 거리가 있는 경우가 빈번하다. 위와 같은 빅데이터를 처리하고 활용하고자 한다면 무선 서비스를 통한 데이터의 저장과 실시간 모니터링은 산업생태계에서 필수라고 말할 수 있다.

본 논문에서는 구축된 빅데이터 저장 및 모니터링 플랫폼을 소개한다. HTTP(HyperText Transfer Protocol)와 MQTT(Message Queuing Telemetry Transport), 두 전송 프로토콜을 통해 IoT 장비 데이터를 Private Cloud내의 Go 기반 서버로 전송한다. 전송된 데이터는 Influx DB에 저장하고 Influx DB와 연동된 Grafana를 활용하여 데이터 시각화를 제공한다.

II. 채택된 IoT 장비

구현된 플랫폼에서 데이터를 전송할 IoT 장비로 채택한 미세먼지 측정기는 ㈜센트리에서 자체 개발한 광산란 방식의 미세먼지 센서를 이용하여 다양한 입경의 미세먼지농도(PM10, PM2.5, PM1)와 온도, 습도 등의 실시간 측정이 가능한 실내/외 설치용 공기질 측정기인 DUST-MON이다.

기존 측정기는 설치된 장소 주변 공기의 습도를 제어하여 포집하고 미세먼지 농도를 분석하여 1분 단위로 온도, 습도, 기기 식별번호 등을 포함한 데이터를 Raspberry Pi에 CSV 파일형태로 기록한다. 기록된 데이터는 USB 또는 LTE 라우터를 통한 근거리 무선 원격 제어로 수집할 수 있는 한계가 있다.

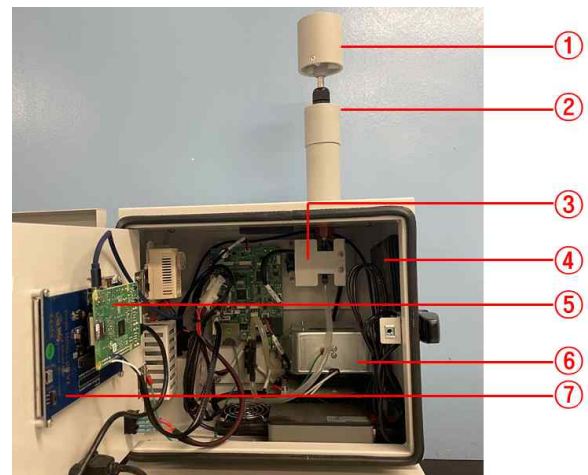


그림 1. 측정기 구성도

번호	이름
1	공기 흡입구
2	공기 전처리 장치
3	미세먼지 센서
4	LTE 라우터
5	Raspberry Pi 3 Model B+
6	펌프
7	5 Inch HDMI LCD

표 1. 측정기 구성 상세

III. 모니터링 플랫폼

3.3. 구현된 아키텍처

시스템 아키텍처는 IoT 장비에서 측정되는 데이터의 범주 및 타입과 마이크로 서비스 연동 효율성을 고려하여 설계되었다.

IoT 장비의 센서 데이터는 CSV 파일 형태로 기록된다. 시간 기반 잡 스

케줄러인 Cron을 통해 1분 단위로 실행되는 Python 기반 전송 모듈은 CSV 파일을 검사한 후 송신되지 않은 데이터 열부터 Json 형태로 변환하여 Private Cloud Domain에 순서대로 일괄적 전송을 한다. 이는 IoT 장비 결함이나 네트워크 문제로 발생할 수 있는 데이터 누락을 해결해준다.

Private Cloud는 확장성과 효율적인 리소스 관리를 고려한 VMware Hypervisor 기반 플랫폼으로 서버 성능 및 자원에 대한 스케일업과 스케일아웃이 용이하도록 구축하였다. Private Cloud 내의 데이터 수신 및 처리를 위한 서버로는 Channel을 통해 동시성을 제공하고 비동기 처리가 우수한 Go 언어를 채택하여 구축하였으며 이를 통해 동시다발적으로 수집되는 데이터를 효율적으로 처리할 수 있다.[2] 구축된 서버가 수집한 데이터는 시간에 종속적인 특성을 가지는 시계열 데이터이므로 본 플랫폼에서는 시계열 데이터 저장에 최적화된 시계열 데이터베이스인 Influx DB에 기기 식별번호로 구분하여 저장하였다. Influx DB에 저장된 데이터는 시각화 툴인 Grafana와 연동을 통해 그림 4와 같이 실시간으로 모니터링이 가능하다.[3] 이때 Influx DB, Grafana 등 주요 컴포넌트들은 Infrastructure에 대한 효율적인 버전 관리 및 구성 환경 통일을 제공하는 Docker Container 기반으로 구축되었다.

IoT 장비와 Private Cloud 간의 데이터 전송을 위한 프로토콜로는 HTTP와 MQTT를 사용하였다. HTTP는 짧은 기간의 연결로도 데이터를 송신할 수 있고 방대한 양의 데이터 트래픽을 처리할 수 있는 요청-응답 프로토콜이다. 반대로 MQTT는 요청없이 데이터를 송신할 수 있는 발행-구독 프로토콜로 M2M과 IoT를 위해 최소한의 전력 및 패킷량으로 통신할 수 있게 설계되었으며 수신된 데이터를 처리하지 않고 이벤트 기반으로 사용할 수 있다. 본 플랫폼에서는 각 프로토콜의 특징과 이점을 고려하여 네트워크 환경 변경에 대응할 수 있도록 두 프로토콜을 이중화하여 사용하였다.[4]

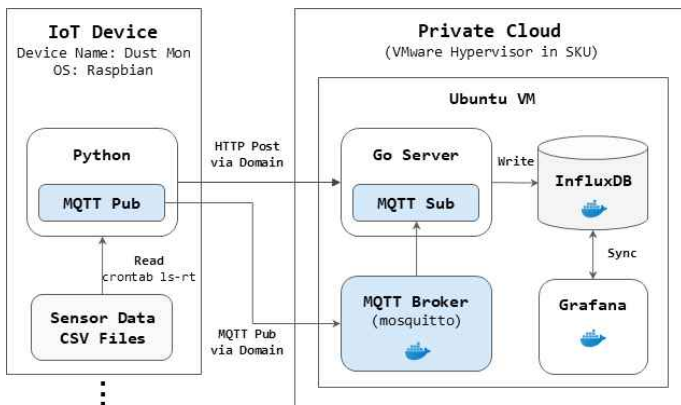


그림 2. System Architecture

3.4 플랫폼 검증

플랫폼 검증을 위하여 스트레스 테스트를 진행한다. IoT 장비들이 동시다발적으로 데이터를 전송하는 가정된 상황을 위해 데이터의 범주 및 빈도를 고려하여 10ms 단위로 데이터를 전송하였으며 이를 통해 프로토콜 별 데이터 취합을 확인하였다. 테스트 결과, 그림 3과 같이 극한의 상황에서도 Influx DB에 데이터가 정상 취합되는 것을 확인하였다.

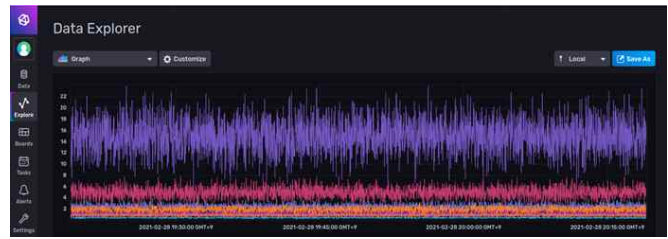


그림 3. 스트레스 테스트 결과

3.5 플랫폼 운용

Grafana Dashboard Pannel에서 연동된 Influx DB의 데이터를 Flux 데이터 스크립팅 언어로 쿼리하여 시각화하였다. 여러 Pannel들로 구성된 Dashboard에서 특정 시간 범위 내의 기기별 데이터를 확인할 수 있고 CSV형태로 다운로드 할 수 있다.

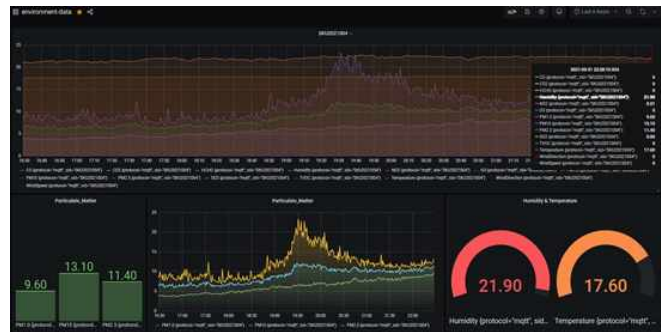


그림 4. 실시간 모니터링 및 시각화용 대시보드 구성

IV. 결론

기존 IoT 장비에서는 직접 수동 조작하여 데이터를 취합하였다. 본 논문에서는 원격 데이터 취합 및 실시간 모니터링이 용이하도록 장비 내의 LTE 라우터를 활용하여 무선통신 및 연계로 IoT 빅데이터 저장 및 모니터링 플랫폼을 구축하였다. 이를 이용하여 기기 고장탐지 및 이상감지를 실시간으로 확인할 수 있으며 시각화된 데이터를 통해 빅데이터 분석 및 예측에 활용할 수 있다. 향후에는 본 플랫폼과 AI API를 제공하는 퍼블릭 클라우드와의 연동을 통해 하이브리드 클라우드를 구축하여 IoT 빅데이터 기반의 데이터 분석 및 예측 플랫폼에 관한 연구를 진행할 계획이다.

ACKNOWLEDGMENT

본 연구는 환경부의 재원으로 한국환경산업기술원의 생활화학제품 안전관리기술개발사업의 지원을 받아 연구되었습니다(2020002970006).

참 고 문 헌

- [1] 김재현. (n.d.). 산업현장에서 요구하는 빅데이터 필요성과 데이터의 중요성. TTA JOURNAL, 통(182), pp. 40-46.
- [2] Documentation . (n.d.). <https://golang.org/>.
- [3] InfluxDB OSS 2.0 Documentation . (n.d.). <https://www.influxdata.com/>.(<http://www.nist.gov/aes>).
- [4] HTTP vs MQTT - Which Communication Protocol Should You Use in Your IoT Application? . (n.d.). <https://www.electromaker.io/blog/article/http-vs-mqtt>.