

오픈 소스를 활용한 모니터링 시스템 자동 시각화에 관한 연구

전세운, 박유현
동의대학교

40798@deu.ac.kr, yhpark@deu.ac.kr

A Study on Automatic Visualization of Monitoring System Using Open Source

Jeon Se Un, Park Yoo Hyun
Dong-Eui Univ.

요 약

4 차 산업혁명을 맞이하여 다양한 분야에서 ICT 기술과의 융합이 이루어지고 있다. 더불어 ICT 와 융합된 시설 등의 모니터링 환경 구축 및 시각화가 큰 문제로 떠오르고 있는데 본 논문은 다양한 환경에서 적용할 수 있고 시각화를 자동으로 해주는 모니터링 시스템을 오픈 소스를 활용하여 구성하는 방법을 제안한다.

I. 서 론

4 차 산업 혁명시대를 맞이한 현대는 농업, 공업, 어업 등 다양한 분야에서 인력 절감과 생산효율 증대를 위하여 ICT 기술과 융합된 다양한 시설이 등장하고 있다. 이러한 시설들은 사람이 시설 내에서 직접 감시하지 않는 특징 때문에 원격에서 시설의 상태를 확인할 수 있는 모니터링 시스템이 필수적으로 요구된다.

그러나 모니터링 대상마다 데이터를 저장하는 형태, 저장된 데이터에 접근하는 방법 등이 다를 수 있기 때문에 모니터링 대상이 바뀌면 모니터링 시스템과 시각화 대시보드 등을 새롭게 구축해야 하는 단점이 있다.

이러한 단점을 해결하기 위해서 본 논문에서는 오픈 소스를 활용하여 데이터의 저장 형태가 다양한 시설에 반영할 수 있고 새로운 모니터링 대상에 대한 자동화된 시각화 기능을 탑재한 모니터링 시스템 자동 시각화를 구축하는 방법에 대하여 설명한다.

2 장에서는 모니터링 시스템 구축 연구에 관하여 소개하고 해당 연구와 본 논문이 어떠한 차별점이 있는지 다룬다. 3 장에서는 시스템의 구축을 위한 몇 가지 기술과 그 기술을 활용, 응용한 시스템의 설계 및 구현을 다루고 4 장에서 결론으로 마무리한다.

II. 관련 연구

시스템 모니터링은 다양한 분야에서 사용된다. 식물 재배를 위한 원격 식물 모니터링[1], 스마트 팩토리 내의 실시간 장비 모니터링[2]과 같은 생산, 제조업 분야, 클러스터 환경 모니터링[3] 같은 클러스터 환경 등이 있다. 그러나 연구에서 제안하는 시스템은 데이터의 수집과 저장, 시각화가 강하게 결합되어 있어서 모니터링 대상의 추가 및 변경이 어렵고 초기 구축 비용이 크다는 단점이 있다.

InfluxDB(TSDB)와 Grafana 를 이용한 플랫폼 모니터링 구축에 관한 연구[4]의 경우 데이터 저장, 시각화가 적절히 분리되어 있지만 여전히 대시보드의 구성에 수작업이 많이 필요하고 대상으로부터 수집된 데이터를 TSDB 에 저장하는 방법이 수집 대상마다 달라지므로 대상에 맞게 수집 방법을 커스터마이징 하는 작업이 필요하게 된다.

원격식물 모니터링[1], 스마트 팩토리 IIoT 엣지 디바이스[5] 이 두가지 연구는 시설에 발생한 데이터를 DB 에 저장한다. DB 접근 기술 부분만 추상화 하면 다양한 대상의 데이터를 같은 방식으로 접근할 수 있음을 말한다.

본 논문에서는 앞서 언급한 연구들을 이용, 개선하는 자동 시각화 모니터링 시스템을 제안한다. 다양한 모니터링 대상에 적용할 수 있고 안정적인 특징을 가진다.

III. 시스템 설계

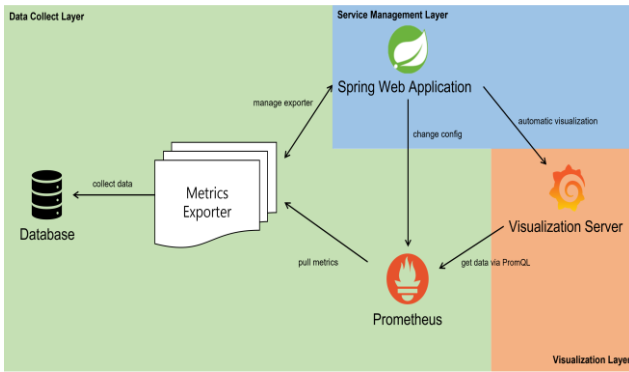
3.1 전체 시스템 구조

전체 시스템은 Metrics Exporter, Management Agent, Visualization Agent 로 나뉜다.

Metrics Exporter 는 모니터링 대상의 데이터를 수집해서 Metrics 으로 변환하여 발행하여 Prometheus TSDB 에서 저장 하도록 한다.

Visualization Agent 는 수집된 자료의 시각화를 담당한다.

Management Agent 는 Visualization Server 와 통신하여 자동 시각화를 수행하고 Metrics Exporter 관리 및 TSDB 설정파일 갱신을 수행한다. 더불어 각 구성은 모두 Docker Container 로 구성하여 사용하도록 한다. 전체적인 구조는 그림 1 과 같다.



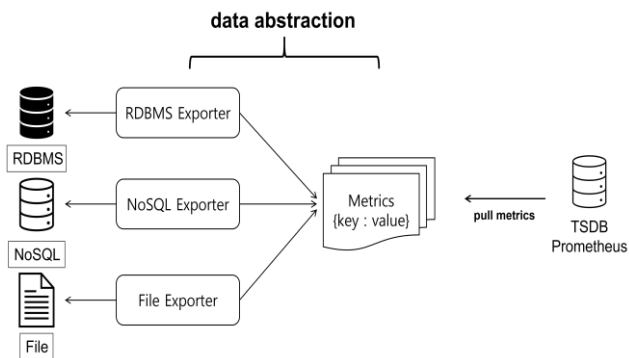
<그림 1 시스템 전체 구성>

3.2 Metrics Exporter (ME)

Metrics란 키, 값의 쌍으로 된 자료를 말하며 Metrics Exporter란 특정 데이터를 읽어 Metrics으로 변환하여 발행하는 소프트웨어를 말한다.

Prometheus[6] Client API와 스프링 부트를 이용하여 제작하고 DB/File 등 다양한 데이터 저장 환경에 대응하기 위해 DB 연결, 파일 입출력이 가능하도록 구현한다. Docker Container로 만들어 구동하며 다음절에서 설명하는 Management Agent(MA)의 요청에 의해 Metrics 정보를 정의하고 발행한다. 이때 MA가 쿼리나, 파일 이름과 같은 수집 대상에 대한 정보를 ME로 전송하면 ME는 Java의 Supplier로 번역하여 데이터 수집에 사용한다.

ME를 사용하면 수집 데이터를 Metrics 형태로 추상화하기 때문에 Prometheus(TSDB)에서 모든 대상을 같은 방식으로 저장할 수 있게 된다.



<그림 2 Metrics Exporter의 Metrics 발행 구조>

3.3 Visualization Agent (VA)

Visualization Agent는 Metrics 시각화를 담당하는데 Metrics 시각화에 최적화된 Grafana[7]를 사용한다. 기본적으로 다양한 그래프를 지원하고 대시보드 구성을 위한 Rest API를 제공한다. 이 API를 MA에서 호출할 수 있도록 하여 모니터링 시스템을 자동 시각화하는데 사용한다.

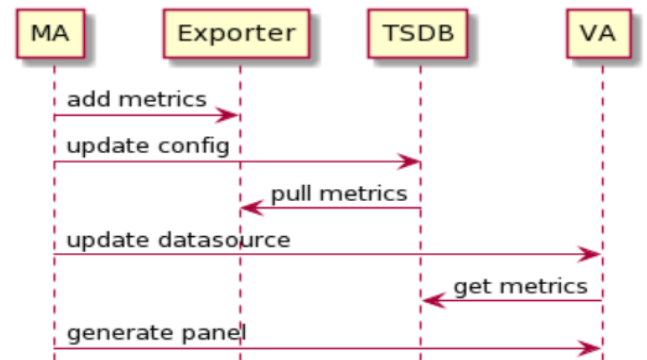
3.4 Management Agent (MA)

MA는 Exporter 관리(생성, Metrics 정의) 및 VA가 대시보드를 생성하도록 API를 호출하는 역할을 한다.

모니터링 대상으로부터 자료를 수집할 ME를 등록하고 ME가 수집할 Metrics을 추가하면 새로운 Metrics을 Prometheus가 수집하도록 Docker volumes으로 연결된 Prometheus의 설정 파일을 업데이트 한다. Prometheus 재시작 없이 설정이 반영되도록 prometheus.yml의 file_sd_configs 속성을 이용하였다. 이는 설정 변경을

위한 서버 재부팅 과정 중에 모니터링이 중단되는 것을 막기 위함이다.

새로운 Metrics을 Prometheus가 수집하면 VA가 대시보드를 생성하도록 API를 호출한다. Metrics에 따른 대시보드 생성 규칙은 미리 정의된 JSON 형태로 만들어 전송한다.



<그림 3 모니터링 자동 시각화 시퀀스>

IV. 결론

본 논문에서는 Prometheus와 Grafana 두가지 오픈소스를 활용하여 다양한 환경에 적용 가능하고 자동화된 시각화 서비스를 제공하는 시스템을 제안하였다. 이러한 구조를 이용하면 모니터링 환경 구축에 드는 비용 절감할 수 있고 한번 제작한 모니터링 시스템의 재사용성을 높일 수 있으리라 생각한다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역 지능화혁신인재양성(Grand ICT 연구센터) 사업의 연구결과로 수행되었음 (IITP-2021-2020-0-01791)

참고 문헌

- [1] 나성주, 송주환. (2021). 라즈베리파이 기반 원격 식물 모니터링 시스템 개발. 한국디지털콘텐츠학회 논문지, 22(4), 737-742.
- [2] 김동현, 이재민, 김종덕. (2021). 스마트팩토리에서 다중장치기반 실시간 장비 모니터링 및 이력관리 시스템 설계 및 구현. 한국정보통신학회논문지, 25(1), 124-133.
- [3] 김재환, 김경훈, 노재춘, 박성순. (2020). 쿠버네티스 클러스터 환경에서 분산 AI 애플리케이션의 SLO를 효율적으로 지원하는 모니터링시스템. 한국방송미디어공학회 학술발표대회 논문집, (), 32-33.
- [4] 김한기. (2020). InfluxDB와 Grafana를 이용한 EDISON 플랫폼 모니터링 대시보드 설계 및 구현. 한국정보과학회 학술발표논문집, (), 25-26.
- [5] 유홍식, 박유현. (2021). 이기종 공작기계 데이터 수집을 위한 IIoT 엣지 디바이스 설계 및 구현. 인터넷전자상거래연구, 21(1), 23-32.
- [6] <https://prometheus.io/>
- [7] <https://grafana.com/grafana/>