JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 11: 1463~1470, Nov. 2021

이기종 데이터베이스와 시각화 편의를 제공하는 통합 모니터링 시스템 구현

전세운[1] · 김민영[2] · 박유현[3*]

Implementation of an integrated monitoring system that support heterogeneous databases and convenient visualization

Seun Jeon[1] · Minyoung Kim[2] · Yoo-Hyun Park[3*]

[1]Full-time researcher, Department of Computer Software Engineering, Dong-Eui University, Busan, 47227 Korea
[2]Assistant professor, Research Institute of ICT Fusion and Convergence, Dong-Eui University, Busan, 47227 Korea
[3*]Professor, Department of Computer Software Engineering, Dong-Eui University, Busan, 47227 Korea

요 약

기존 모니터링 시스템들은 모니터링 대상에 따라 개별 시스템을 구현하였지만, 최근에서 Prometheus, Grafana와 같은 오픈소스를 활용하여 모니터링 시스템을 구현하고 있다. Prometheus와 Grafana를 활용하면 기존 모니터링 시스템 개발 방법에 비해서는 많은 부분이 편리해졌지만, 아직도 문제점이 남아 있다. 본 논문에서는 이러한 문제점을 해결하기 위해 Prometheus와 Grafana를 지원하는 통합형 모니터링 시스템을 제안한다. 제안하는 시스템은 모니터링 대상 데이터의 수집, 저장, 시각화, 관리를 수행하는 세부 모듈로 역할을 구분하고 기존의 문제점을 해결할 수 있도록 각 모듈을 구현하였다. 제안하는 시스템은 이기종 데이터베이스에 저장된 모니터링 대상을 편리하게 관리하여 모니터링 할 수 있으며 단순한 조작을 통하여 대시보드를 생성할 수 있다.

ABSTRACT

With the development of ICT technology, a monitoring system to check the status of an object to be managed in real time in various industrial fields is widely used. Existing monitoring systems implemented individual systems according to monitoring targets, but recently, monitoring systems have been implemented using open sources such as Prometheus and Grafana. When using Prometheus and Grafana, many parts become more convenient compared to the existing monitoring system development method, but there are still problems. In this paper, to solve this problem, we propose an integrated monitoring system that supports Prometheus and Grafana. The proposed system is a detailed module that collects, stores, visualizes, and manages data to be monitored, and each module is implemented so that roles can be divided and existing problems can be solved. The proposed system can conveniently manage and monitor monitoring targets stored in heterogeneous databases, and create dashboards through simple operation.

키워드 : 모니터링 시스템, 메트릭, 시각화, 프로메테우스, 그라파나
Keywords : Monitoring system, Metric, Visualization, Prometheus, Grafana

Received 5 October 2021, **Revised** 6 October 2021, **Accepted** 11 October 2021

*** Corresponding Author** Yoo-Hyun Park(E-mail:yhpark@deu.ac.kr, Tel:+82-51-890-1737)
Professor, Department of Computer Software Engineering, Dong-Eui University, Busan, 47227 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.11.1463 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

ICT(Information and Communications Technology)의 발전으로 다양한 산업 분야에서 관리해야 할 대상의 상태를 실시간으로 확인하기 위해 모니터링 시스템을 도입하고 있다. 모니터링 시스템을 통하여 관리자는 관리대상의 다양한 돌발상황에 대해 신속하게 대비할 수 있어 시간과 비용을 절약할 수 있다. 일반적으로 모니터링 시스템은 실시간으로 수집되는 데이터를 효과적으로 분석하고 즉시 의사결정을 할 수 있도록 해당 데이터를 직접 또는 가공하여 시각적으로 제공하는 대시보드(Dashboard)를 포함하고 있다[1-2].

모니터링 대상 데이터가 많아짐에 따라 여러 가지 문제점이 발생할 수 있다. 즉, 모니터링 대상 데이터가 많아 일부 데이터는 A 데이터베이스에, 다른 데이터는 B 데이터베이스에 저장한 경우, 모니터링 시스템에서 데이터를 가져오는 부분을 별도로 구축해야 한다.

따라서, 다양한 환경에 적용 가능한 통합 모니터링 시스템이 제공된다면 관리자는 효율적인 의사결정을 할 수 있게 될 것이다.[1-3].

한편, 기존 모니터링 시스템들은 모니터링 대상에 맞게 각자 시스템을 구현하였지만, 최근에서 오픈소스를 활용하여 모니터링 시스템을 구현하고 있다. 데이터 수집을 하기 위한 오픈소스로는 Prometheus[4]가 있으며 시각화를 위한 오픈소스로는 Grafana[5]가 많이 활용되고 있다.

Prometheus와 Grafana를 활용하면 기존 모니터링 시스템 개발 방법에 비해서는 많은 부분이 편리해졌지만, 주로 Prometheus와 함께 사용되는 Node Exporter의 데이터 종속성과 모니터링 대상의 변경을 반영하기 위해서 직접 설정파일을 변경해야 하는 불편함이 존재한다. 또한 대시보드를 구성할때에도 Grafana의 웹페이지에 접근하여 다양한 설정값을 모두 지정해야 하는 어려움이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 Prometheus와 Grafana를 지원하는 통합형 모니터링 시스템을 제안한다. 제안하는 시스템은 모니터링 대상 데이터의 수집, 저장, 시각화, 관리를 수행하는 세부 모듈로 역할을 구분하고 기존의 문제점을 해결할 수 있도록 각 모듈을 구현하였다. 제안하는 시스템은 이기종 데이터베이스에 저장된 모니터링 대상을 편리하게 관리하여 모니터링 할 수 있으며 단순한 조작을 통하여 대시보드를 생성할 수 있다.

Ⅱ. 관련 연구

현재 다양한 분야에서 데이터 기반 모니터링 시스템이 도입되어 사용되고 있다. 특히 센서 기반으로 하는 IoT(Internet of Things) 시스템은 본 논문의 서론에서 언급한 모니터링의 순기능을 사용자에게 제공하기 위해 대시보드를 필수적으로 제공하고 있다. 이런 모니터링 시스템을 사용하는 분야는 농작물 상태를 확인해 수확량을 높이기 위한 스마트 팜[6,7] 분야, 생산성 향상과 안전사고를 방지하기 위한 스마트 팩토리[8,9] 분야, 그리고 IDC(Internet data center)의 서버 컴퓨터와 통신망의 효율적인 상태 관리를 위한 ICT 인프라 분야[10] 등이 있다.

모니터링 시스템에서 데이터를 수집하는 오픈소스로 Prometheus가 있다. Prometheus의 Node Exporter 모듈은 Prometheus에게 메트릭을 가져갈 수 있게 특정 서비스에 메트릭을 노출시키고, Prometheus server에서 이 Node Exporter의 endpoint로 HTTP GET request를 보내 메트릭을 주기적으로 수집한다. Prometheus 특성상 모든 데이터를 수집하지 않고, 일정 주기 동안 발생하는 메트릭을 수집하기 때문에, 애플리케이션에 부담이 적다. 데이터 저장소가 시계열 데이터 저장소로 구성되어 있기 때문에 많은 양의 정보를 빠르게 검색할 수 있다. 그 외에도 다양한 메트릭 Node Exporter 를 제공한다는 점, 시각화 툴인 Grafana 와의 연동을 통한 운영이 쉽다는 장점이 있다.

또한, 모니터링 시스템의 대시보드를 구성하는 오픈소스로 Grafana가 있다. Grafana는 멀티 플랫폼 오픈 소스 분석 및 대화형 시각화 웹 애플리케이션으로, 지원되는 데이터 소스에 연결되었을 때 웹 기반의 차트, 그래프 및 알림(alert) 기능을 제공한다. 플러그인 시스템을 통해 여러 가지 확장기능을 사용할 수 있다. 사용자는 대화형 쿼리(query) 작성기를 사용하여 복잡한 모니터링 대시보드를 만들 수 있다[11].

Prometheus와 Grafana의 조합으로 모니터링 시스템을 구축한 기존 연구는 다수 존재한다[12]. 그러나 Prometheus 사용을 위해서는 데이터를 수집하는 Node Exporter라는 모듈을 사용해야 한다. Node Export 모듈

은 수집 데이터가 저장된 데이터베이스나 시스템에 종속적이기 때문에 수집할 데이터가 다수의 데이터베이스나 시스템에 저장되어 있는 경우에는 각각을 담당하는 Node Exporter를 별도로 사용해야 한다. 또한, 데이터 수집이 진행되고 있는 중에 수집 데이터를 추가해야 하는 경우에는 해당 Node Exporter가 메트릭 정의를 위해 사용하는 설정 파일 등을 직접 수정해야 한다. 아울러 모니터링 시스템에서 Grafana를 사용할 때, 직접 Grafana의 웹 페이지에 접근해서 다양한 설정값을 일일이 설정해야 하는데 이 부분에서 시간이 많이 소요되어 불편하다.

본 논문에서는 이러한 문제점을 개선한 통합 모니터링 시스템을 제안한다. 제안하는 시스템은 이기종 데이터베이스에 저장된 모니터링 대상의 데이터를 수집할 수 있으며, 수집 중에라도 대상 데이터를 동적으로 추가/삭제 할 수 있다. 또한, 기존 Grafana의 대시보드를 구성하는 작업의 많은 부분을 단순화 하였다.

Ⅲ. 시스템 설계

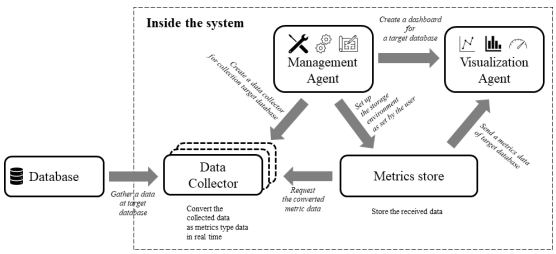

Fig. 1 The system configuration diagram

(그림 1)은 본 논문에서 제안하는 시스템의 전체 구성을 나타낸다. 본 논문의 시스템은 모니터링을 할 대상 시스템의 선택 데이터를 실시간으로 해당 데이터베이스에서 가져와 메트릭 형태로 변환해 '메트릭 저장소'에 전달하는 '데이터 수집기, 메트릭 저장소에 저장된 모니터링 대상의 변환된 메트릭 데이터를 불러와 시각화된 내용을 사용자에게 대시보드(Dashboard)로 보여주는 '시각화 에이전트', 모니터링 대상의 데이터를 수집하는데 필요한 데이터 수집기 생성하는 작업과 시각화 에이전트의 대시보드 생성 및 설정 작업 그리고 메트릭 저장소 설정 및 관리 작업을 할 수 있는 기능을 제공하는 '관리 에이전트'로 구성된다[3].

3.1. 관리 에이전트

관리 에이전트는 다음에서 설명할 데이터 수집기와 시각화 에이전트의 관리(생성/제거)와 이를 위한 환경설정을 할 수 있는 기능을 사용자에게 제공한다. 이 기능들은 별도의 프로그램 설치 없이 사용자가 웹 브라우저를 통해 관리 에이전트에 접속하여 이용할 수 있도록 웹사이트로 제공한다. 이를 위해 관리 에이전트는 HTTP 기반 서버에서 운영된다.

3.2. 데이터 수집기

데이터 수집기는 모니터링 할 대상의 데이터를 저장되고 있는 데이터베이스에서 해당 데이터를 수집해 메트릭 형태(Key, 시각화 에이전트)로 변환해 메트릭 저장소에 저장한다.

데이터 수집기는 관리 에이전트를 통해 생성되는데, 이때 (그림 1)와 같이 데이터 수집기는 대상 데이터베이스의 접근할 수 있는 접속 정보를 가진 Connector와 해당 데이터베이스 Connector에서 수집될 데이터를 수집하기 위한 질의문을 가져 메트릭 데이터 형태로 변환하는 메트릭 Convertor로 구성되어 생성된다. 데이터 수집기는 많이 사용하는 RDB(Relational Database) 뿐만 아니라 NoSQL 같은 비 관계형 데이터베이스와 같은 여러 형태의 데이터베이스를 지원할 수 있도록 설계하였다. 데이터 수집기는 생성된 이후 실시간으로 실행되어 메트릭 저장소에서 저장할 수 있도록 데이터베이스의 값을 메트릭 데이터 형태로 변환한다.

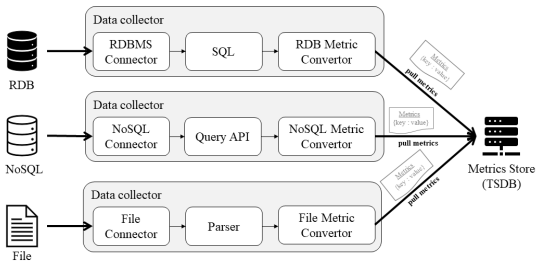

Fig. 2 The Data Collector schematic diagram

3.3. 메트릭 저장소

메트릭 저장소는 데이터 수집기에서 생성되는 메트릭 데이터를 실시간으로 수집해 자신이 가지고 있는 데

이터베이스에 저장한다. 메트릭 저장소는 자신이 수집한 데이터와 시간을 함께 저장하기 위해 시계열 데이터베이스(Time-series Database, TSDB)를 가진다.

데이터 수집기는 데이터 베이스 당 하나씩 생성된다. 만약 모니터링 시스템에서 10개의 데이터베이스를 참조하고자 하면 10개의 데이터 수집기를 생성해야 한다.

이는 본 논문의 시스템에서 독립적인 개체로 생성하고 이후 제거가 쉽게 할 수 있도록 관리적인 측면을 고려한 것이다.

3.4. 시각화 에이전트

시각화 에이전트는 메트릭 저장소에 저장된 모니터링 대상의 메트릭 데이터를 시각화하여 사용자에게 제공할 대시보드를 생성한다. 이와 관련된 데이터 선정, 시각화 방법, 환경설정은 관리 에이전트에서 수행한다. 또한, 사용자는 관리 에이전트를 통해 시각화 에이전트에서 사용될 여러 개의 대시보드를 만들 수 있다. 시각화 에이전트의 설정 내용은 JSON(JavaScript Object Notation) 형태로 저장한다.

Ⅳ. 시스템 구현 및 결과

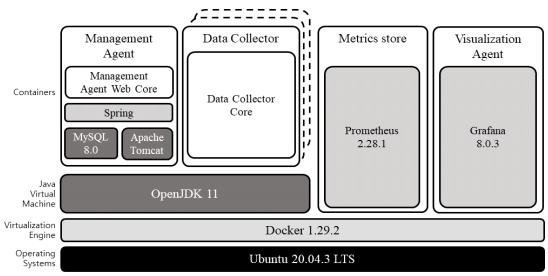

Fig. 3 Implementation environment hierarchy

(그림 3)은 실제 본 논문의 시스템을 구현하기 위해 구축한 시스템 환경을 계층별로 구분하여 보여준다. 본 논문의 시스템의 각 요소는 Docker 용 컨테이너 인스턴스로 구현하였다. 본 논문의 요소별로 요구되는 소프트웨어 환경이 달라 가상화 기반 프로세스 실행 환경을 만들어 운영할 수 있는 Docker 컨테이너로 구현하였다.

(그림 4)는 본 논문의 시스템이 실행되는 순서 내용을 나타내는 절차도이며, 해당 내용을 바탕을 시스템 절차를 구현하였다.

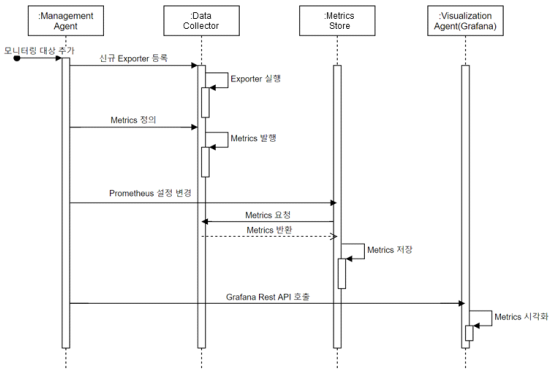

Fig. 4 Implementation sequence diagram

4.1. 관리 에이전트

관리 에이전트는 데이터 수집기 Docker 컨테이너 생성 및 구동, Grafana HTTP API 호출, Prometheus 설정 파일 관리 역할을 하며 이 기능을 웹 페이지를 통해 쉽게 사용할 수 있도록 구현하였다. 실제 Spring boot web application 형식으로 구현되었으며 추가로 데이터 수집기 컨테이너 생성을 위해 Docker client API를 추가로 활용하여 구현하였다.

데이터 수집기는 모니터링 데이터 수집대상 마다 각각의 Docker 컨테이너로 생성하여 관리 에이전트에서 관리한다. Docker를 이용하여 데이터 수집기를 동적으로 생성, 실행, 제거 해서 모니터링 시스템이 실행 중일 때에도 동적으로 신규 모니터링 대상의 추가, 삭제가 가능하다. 이때 각 데이터 수집기의 메트릭을 관리하기 위해 API를 웹 페이지를 통해 관리할 수 있는 기능을 구현하였다.

시각화 에이전트에 대시보드를 생성하고 해당 대시보드에 그래프를 추가하는 작업을 설정할 수 있는 기능을 웹 페이지를 통해 구현하여 제공한다. 이때 사용자가 웹페이지를 통해 설정한 해당 설정 내용을 JSON 형태의 데이터를 생성해 HTTP를 이용해 시각화 에이전트에게 전달하도록 구현하기 위해 Grafana HTTP API를 호출하는 라이브러리를 구현하였고 관리 에이전트의 요청에 맞게 해당 라이브러리를 사용하여 Grafana에서 시각화 작업을 처리할 수 있도록 하였다.

메트릭 저장소에서 데이터 수집기가 발행하는 메트릭을 저장하기 위해서는 데이터 수집기가 메트릭을 발행하는 주소(Http Enpoint)를 JSON 형식의 설정 파일에

기술해 주어야 한다. 이를 위해서 Prometheus 설정 파일을 갱신하는 라이브러리를 구현하였으며 해당 라이브러리를 이용하여 Prometheus 설정 JSON 파일을 갱신하는 방식으로 메트릭 저장소와 데이터 수집기가 연결되도록 구현하였다.

마지막으로 관리 에이전트의 웹페이지에 접근할 수 있는 사용자를 추가하고 관리할 수 있는 기능을 구현하였다. 또한, 관리 에이전트의 환경설정 내용을 저장하기 위해 MySQL로 전용 데이터베이스를 구축하였다.

4.2. 데이터 수집기

데이터 수집기는 실행 시점에 환경변수에서 데이터베이스 접속 정보를 읽어와 DataSource 객체를 생성하고 해당 객체를 기반으로 데이터베이스에 연결하여 대상 데이터를 수집해 메트릭 형태로 변형하도록 구현하였다.

데이터 수집기는 메트릭의 생성, 삭제, 수정, 조회 등의 기능을 제공하기 위하여 Spring boot 웹 application 기반으로 구현하여 Rest API를 제공한다. 관리 에이전트에서는 데이터 수집기에서 제공하는 API를 이용해 데이터 수집기가 발행하는 메트릭을 관리한다. 그리고 데이터 수집 후 메트릭 데이터로 변환하기 위하여 Prometheus client API 이용해 구현하였다.

4.3. 메트릭 저장소

메트릭 저장소는 데이터 수집기에서 변환되는 메트릭 데이터를 실시간으로 수집하기 위해 관련 오픈소스 플랫폼인 Prometheus로 구현하였다. 이때 Prometheus는 시계열 데이터베이스(TSDB)를 기본적으로 포함하고 있다. 또한, 최초의 환경설정을 하는 스크립트도 구현하였으며, 데이터 수집을 위한 데이터 수집기의 주소를 설정 파일에서 읽어와 메트릭 데이터를 저장하며 해당 설정 파일은 Docker Volume을 이용하여 관리 에이전트와 메트릭 저장소가 공유한다.

4.4. 시각화 에이전트

시각화 에이전트는 Grafana를 이용하여 구축하였으며 메트릭 저장소에 있는 메트릭 정보를 읽어와 시각화한다. Grafana HTTP API를 제공하는 것을 이용하여 관리 에이전트에서 해당 API를 호출하는 방식으로 시각화 에이전트를 조작하게 된다. 관리 에이전트에서 API를 호출하면 대시보드 생성 또는 대시보드 내의 그래프를 생성하고 사용자에게 제공할 수 있도록 구현하였다.

4.5. 동작과정

사용자가 해당 시스템을 이용하여 모니터링 시스템을 구성하는 절차는 다음의 총 8단계의 과정을 거친다. (1) 시각화 에이전트 데이터 소스 추가 (2) 데이터 수집기 생성 (3) 메트릭 저장소 설정 변경 (4) 데이터 수집기 실행 (5) 대시보드 생성 (6) 시각화 규칙 생성 (7) 메트릭 정의 (8) 시각화 그래프 추가

(1) 시각화 에이전트 데이터 소스 추가 단계 : 모든 과정에 앞서서 최종 결과물인 대시보드 생성을 위해서는 시각화 에이전트에서 특정 데이터를 가져올 수 있도록 데이터 소스를 추가하는 작업이 필요하다. 본 논문에서는 데이터 소스로 메트릭 저장소(Prometheus)를 사용하므로 메트릭 저장소를 데이터 소스로 추가한다. 해당 작업은 서비스 최초 실행 시 한번 수행하며 사용자가 별도로 조작하지 않아도 수행된다.

(2) 데이터 수집기 생성 단계 : 모니터링 대상의 정보를 수집하기 위하여 데이터 수집기를 생성한다. (그림 5) 데이터 수집기 생성에는 해당 수집기의 식별을 위한 이름과 데이터베이스 연결 정보를 입력한다. 특히 Auto Create Dashboard 옵션을 체크하고 데이터 수집기를 생성하는 경우 시각화 에이전트에 대시보드 하나를 자동으로 생성한다. 이 기능을 이용하면 (5)를 생략할 수 있다.

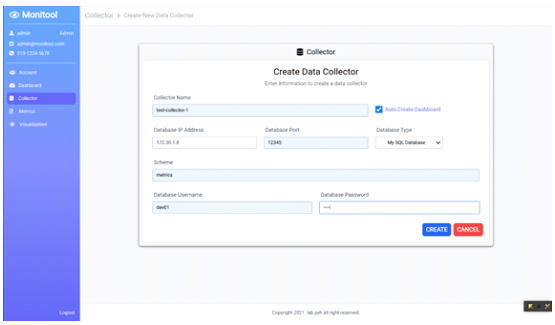

Fig. 5 Management Agent – Create new Data Collector

(3) 메트릭 저장소 설정 변경 단계 : 데이터 수집기가 생성되었을 때 메트릭 저장소가 데이터 수집기에서 메트릭을 수집할 수 있도록 메트릭 저장소의 설정 파일을 갱신해주어야 한다. 이를 위해 관리 에이전트에서 새로운 데이터 수집기가 추가되면 해당 데이터 수집기의 메트릭을 수집하는 IP 정보와 고유 식별자를 메트릭 저장

소의 설정 파일에 추가한다. 설정 파일은 JSON 형식으로 되어 있으며 해당 파일은 Docker의 Volume을 이용하여 관리 에이전트와 메트릭 저장소가 공유한다.

(4) 데이터 수집기 실행 단계 : 데이터 수집기가 생성한 직후에 데이터 수집기는 Docker의 Container로 존재할 뿐이며 실행 중이 아니다. 데이터 수집기를 사용하기 위해서는 데이터 수집기를 실행하는 작업이 필요한데 이는 데이터 수집기 관리 콘솔(그림 6)에서 수행할 수 있다. 관리 콘솔 우측 하단의 실행 버튼을 선택하면 잠시 후 데이터 수집기가 실행되고 상태가 RUNNING으로 바뀐다.

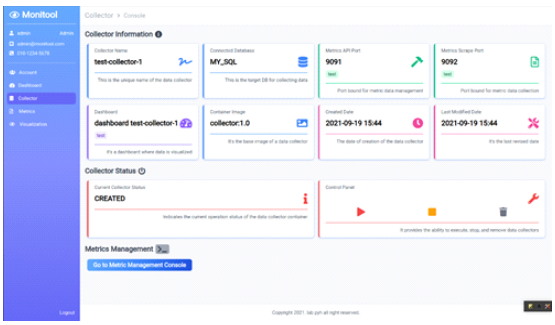

Fig. 6 Management Agent function – Data Collector state monitoring and configuration set-up

(5) 대시보드 생성 단계 : 데이터 수집기에 특정 메트릭을 수집하도록 명령하기 전 시각화 작업을 위한 대시보드 생성이 필요하다. 데이터 수집기 생성시의 Auto Create Dashboard 옵션을 선택하면 별도의 작업이 필요하지 않으며 데이터 수집기 생성 시에 Auto Create Dashboard 옵션을 체크 하지 않았다면 (그림 7) 대시보드 생성 기능을 사용하도록 한다.

(6) 시각화 규칙 생성 단계 : 시각화 작업 전으로 필요한 것은 대시보드 내의 그래프의 배치를 어떻게 할 것인가에 대한 규칙 정의다. Visualization 메뉴로 가면 생성된 시각화 규칙을 확인하거나 신규 시각화 규칙을 생성할 수 있는데 여기서 자신이 사용할 새로운 규칙을 하나 정의한다.

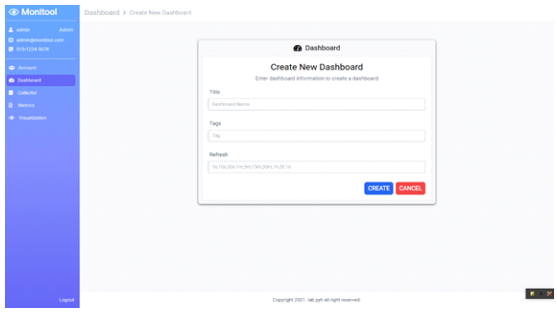

Fig. 7 Management Agent function – Create new Dashboard

(7) 메트릭 정의 단계 : 데이터 수집기가 수집할 메트릭을 정의하기 위해서 데이터 수집기가 실행된 상태에서 (그림 6)의 좌측 최하단에 있는 Go To Manage Console 버튼을 선택한다. 데이터 수집기가 정상적으로 실행된 경우 화면은 메트릭 관리 콘솔로 이동하는데 (그림 8) 해당 화면에서 메트릭을 정의할 수 있다. 메트릭은 고유한 이름과 도움말 그리고 모니터링 대상에서 데이터를 가져올 쿼리 문자열, 그리고 메트릭의 타입으로 구성되며 본 논문에서는 RDB메트릭 저장소의 데이터를 수집하는 데이터 수집기를 구현하고 사용하였기 때문에 쿼리 문자열은 SQL이 된다. 메트릭의 타입은 수집하려는 값의 특성에 맞게 선택하면 되는데 자세한 내용은 본 논문의 범위를 벗어나므로 Prometheus 공식문서[9]를 참조한다. 주의할 점은 쿼리 문자열의 수행 결과는 반드시 하나의 Double 형식 값이어야 한다. 하나를 초과하는 값이거나 Double 형식 또는 Double 형식으로 변환이 불가능한 값인 경우 메트릭으로 생성하지 못한다.

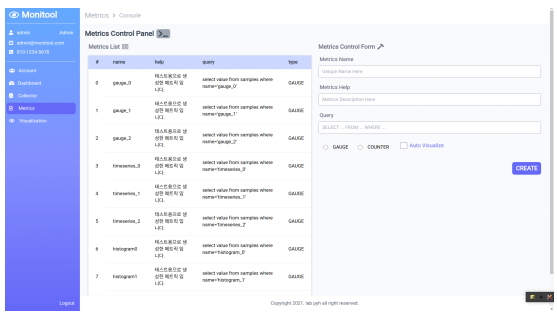

Fig. 8 Management Agent function – Create and modify new Metric rule item

(8) 시각화 그래프 추가 단계 : 최종적으로 그래프를

생성하기 위해서는 (7)에서 메트릭을 정의할 때 Auto Visualize 옵션을 체크해야한다. 해당 옵션을 체크하고 메트릭을 정의하면 어떤 대시보드에 그래프를 생성할지 선택하는 화면으로 이동하고 선택 이후 시각화 규칙 선택화면 마지막으로 그래프의 종류를 선택하는 화면으로 이동하며 시각화 에이전트에 그래프가 생성된다. 대시보드 선택과 시각화 규칙 선택은 한 번 수행하고 나면 데이터 수집기와 대시보드, 시각화 규칙이 연결되기 때문에 매번 수행하지 않아도 된다.

(그림 9)는 해당 기능을 이용하여 생성한 시각화 에이전트의 대시보드를 나타낸다.

Fig. 9 Visualization agent function – Show a dashboard

V. 결 론

ICT 기술 발전으로 다양한 산업 분야에서 관리해야 할 대상의 상태를 실시간으로 확인하기 위한 모니터링 시스템이 많이 사용되고 있다. 모니터링 시스템을 통하여 관리자는 관리대상의 다양한 돌발상황에 대해 신속하게 대비할 수 있어 시간과 비용을 절약할 수 있다.

기존 모니터링 시스템들은 모니터링 대상에 맞게 각자 시스템을 구현하였지만, 최근에서 오픈소스를 활용하여 모니터링 시스템을 구현하고 있다. 데이터 수집을 하기 위한 오픈소스로는 Prometheus가 있으며 시각화를 위한 오픈소스로는 Grafana가 많이 활용되고 있다.

Prometheus와 Grafana를 활용하면 기존 모니터링 시스템 개발 방법에 비해서는 많은 부분이 편리해졌지만, 주로 Prometheus와 함께 사용되는 Node Exporter의 데이터 종속성과 모니터링 대상의 변경을 반영하기 위해서 직접 설정파일을 변경해야 하는 불편함이 존재한다. 또한 대시보드를 구성할때에도 Grafana의 웹페이지에 접근하여 다양한 설정값을 모두 지정해야 하는 어려움이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 Prometheus와 Grafana를 지원하는 통합형 모니터링 시스템을 제안하였다. 제안하는 시스템은 관리 에이전트, 데이터 수집기, 메트릭저장소, 시각화 에이전트로 구성된다. 제안하는 시스템을 통하여 이기종 데이터베이스에 저장된 모니터링 대상을 편리하게 모니터링 할 수 있으며 단순한 조작을 통하여 대시보드를 생성할 수 있다.

본 논문에서 구현된 시스템은 현재 RDBMS 중 MySQL 및 Maria DB만 지원하고 있지만, 추후 다양한 모니터링 시스템의 데이터베이스를 지원하기 위해 NoSQL와 일반 데이터 파일을 추가로 지원할 수 있는 기능을 구현할 예정이다.

ACKNOWLEDGEMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2021-2020-0-01791) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation).

References

[1] KOREA Data Agency. The choice for reliable service: Knowing the monitoring system [Internet]. Available: https://www.kdata.or.kr/.

[2] WhaTap. 'What is monitoring' for beginners in monitoring [Internet]. Available: https://www.whatap.io/ko/blog/35/.

[3] S. U. Jeon and Y. H. Park, “A Study on Automatic Visualization of Monitoring System Using Open Source,” in *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 543-544, 2021.

[4] Prometheus Authors, Prometheus official web site [Internet]. Available: https://prometheus.io/.

[5] Grafana Labs, Grafana official web site [Internet]. Available: https://grafana.com/.

[6] M. S. Choi. “A Study on the Efficient Implementation Method of Cloud-based Smart Farm Control System,” *Journal of Digital Convergence*, vol. 18, no. 3, pp. 171-177, 2020.
[7] J. D. Seo, Y. S. Park, Y. H. Jang, and G. B. Kim, “Air House Smart Farm Growth Environment Control System based on IoT and Big Data,” in *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 358-359, 2017.
[8] J. H. Yoon, J. M. Jung, and B. J. Ko, “Implementation of Monitoring System for Smart Factory,” *Journal of Advanced Navigation Technology*, vol. 22, no. 5, pp. 485-489, 2018.
[9] H. R. Choung, K. H. Bae, M. K. Lee, H. M. Kwon, and S. H. Hong, “Quality Strategy for Building a Smart Factory in the Fourth Industrial Revolution,” *Journal of Korean Society for Quality Management*, vol. 48, no. 1, pp. 87-105, 2020.
[10] Secure Point, ICT infrastructure integrated monitoring - ‘SolarWinds’ [Internet]. Available: http://www.securepoint.co.kr/business/monitoring_01_01.php.
[11] H. G. Kim, “Design and implementation of EDISON Platform monitoring dashboards using InfluxDB and Grafana,” in *Proceedings of Korean Institute of Information Scientists and Engineers*, pp. 25-26, 2020.
[12] S. Y. An, Y. S. Cha, E. J. Jeon, G. Y. Gwon, B. C. Shin, and B. R. Cha, “A Pre-Study on the Open Source Prometheus Monitoring System,” *Smart Media Journal*, vol. 10, no. 2, pp. 110-118, 2021.

전세운(Seun Jeon)

2021년 동의대학교 컴퓨터소프트웨어공학과 공학사
2021년 ~ 현재 동의대학교 소프트웨어공학과 연구원
※관심분야 : 인터넷 시스템, 소프트웨어공학, 네트워크

김민영(Minyoung Kim)

2011년 동의대학교 컴퓨터공학과 공학사
2013년 동의대학교 컴퓨터공학과 공학석사
2020년 동의대학교 컴퓨터공학과 공학박사
2020년 ~ 현재 동의대학교 ICT융복합연구소 조교수
※관심분야 : IoT, LPWA 통신, 블록체인, 딥러닝, 인터넷 시스템

박유현(Yoo-Hyun Park)

2008년 부산대학교 전자계산학과 이학박사
2000년 한국국방연구원(KIDA) 연구원
2001년 ~ 2009년 한국전자통신연구원(ETRI) 선임연구원
2012년 ~ 2014년 동의대학교 부산IT융합부품연구소 부소장
2009년 ~ 현재 동의대학교 컴퓨터소프트웨어공학과 교수
2018년 ~ 현재 동의대학교 융합소프트웨어센터장
2021년 ~ 현재 동의대학교 ICT융복합연구소장
※관심분야 : 클라우드, 빅데이터, 인터넷 시스템