

IoT 소프트웨어 배포 및 관리를 위한 Kubernetes 적용 기법

*금승우 김영기 유미선 문재원

전자부품연구원

*swkum@keti.re.kr

Methods on Kubernetes-based deployment of IoT software

*Seungwoo Kum, Youngkee Kim, Miseon Yu, Jaewon Moon

Korea Electronics Technology Institute

요약

IoT 표준은 미디어 관련 분석 기술에 대한 IoT 기술의 적용을 통하여 IoT 기반 프로토콜로 미디어 분석 기술에 대한 제어 및 IoT 기기간 연동을 제어할 수 있는 기술로써 ISO/IEC SC29 산하에서 이루어지고 있는 표준 기술이다. 2019년 국제 표준의 도출을 통하여 각종 미디어 관련 제어 기술을 분석 (Analyze), 입력 (Sensor), 표현 (Actuator), 저장 (Storage) 등의 기기 분류를 통하여 표준화하고 있으며, 다양한 유즈케이스에 대응하는 영상 분석 기술에 대한 인터페이스가 도출된 바 있다. 본 논문은 이러한 인터페이스를 가지는 IoT 기술에 대하여 서비스를 구성 및 배포 관리하기 위한 기법을 제안한다. IoT 기기 혹은 서비스를 조합하여 영상 입력으로부터 최종 결과까지의 서비스를 구성하기 위해서는 각각의 IoT 기기에 대한 설정 및 기기, 서비스간 연결의 설정이 요구된다. 다만 IoT 표준은 각 IoT 서비스 기기에 대한 인터페이스만을 정의하고 있어 이러한 기기 및 서비스간 연결에 대한 구성이 포함되어 있지 않으며, 본 논문에서 제시하는 방법은 클라우드에서 사용되고 있는 자원 관리 및 배포 기술인 Kubernetes를 통하여 IoT 기기들간의 연결을 설정하고 최종 서비스를 구성하는 방법을 포함한다.

1. 서론

IoT (Internet of Media Things)는 ISO/IEC SC29 WG11에서 진행되고 있는 국제 표준으로[1] 써 미디어에 대한 제어를 위하여 IoT 기술을 적용하는 방법을 제시한다. 이를 통하여 다양한 미디어 관련 기술들을 IoT를 통하여 손쉽게 제어할 수 있으며, 또한 IoT 기기들과의 호환성도 보장할 수 있는 장점을 가진다. IoT는 본 논문이 작성되는 시점 기준으로 IS (International Standard)가 도출되었으며, 이는 ISO/IEC 23093 문서에서 확인할 수 있다. IoT는 미디어 관련 IoT 제어 및 서비스 연계를 위하여 크게 네 가지의 디바이스 타입을 정의하고 있는데 영상 및 데이터를 수집할 수 있는 MSensor, 영상의 분석 기능을 제공할 수 있는 MAnalyzer, 결과를 표현 및 표출할 수 있는 MActuator 그리고 영상, 데이터 및 분석 결과를 저장할 수 있는 MStorage가 그것이다. 해당 기기들의 정의는 최상위 객체인 MThing의 하위 클래스로 정의된다. 각 IoT 개발자는 자신이 원하는 기능의 구현을 위하여 적절한 디바이스 타입을 선정하여 해당 기기를 구현할 수 있으며, 또한 필요에 따라 적절히 기능을 확장 정의할 수 있도록 설계되어 있다.

이러한 IoT 서비스 기기를 통하여 센서로부터 최종단계까지의 서비스를 구현가능하다. 예를 들어 카메라로부터 입력을 받아 영상에 포함되어 있는 오브젝트를 검출하고 이를 다시 영상으로 표현하는 경우, MCamera로부터 입력된 영상을 MAnalyzer에 전달하고, 이 MAnalyzer의 결과를 다시 MActuator에 전달하는 형태로 최종 서비스를 구성할 수 있다. 각각에 대한 인터페이스는 IoT에 포함되어 있는 표준 인터페이스를 활용하고 필요에 따라 확장 정의할 수 있다. 이와 같

이 서비스를 구성하는 경우 각 IoT 서비스간의 연결에 대한 설정은 자동으로 구성되지 않으며 이에 대한 연결 설정에 대한 정의 및 적용 방법이 요구된다. 현재로써는 각 IoT 서비스에 대해서 별도의 설정을 제어하는 방법을 활용할 수 있으나 그에 대한 구체적인 구현에는 아직까지 확인되지 않고 있다.

본 논문은 이러한 IoT 기술 기반의 미디어 서비스 제공에 있어 현재 클라우드에서 사용되고 있는 자원 관리 및 배포 기법을 적용한 소프트웨어 배포 및 관리 기술을 제안한다. 클라우드 기술은 최근 마이크로 서비스 아키텍처의 적용이 빠른 속도로 이루어지고 있으며, 이로 인하여 각 단위 마이크로서비스간의 연동을 설정 및 관리하기 위한 기술들이 다방면에서 연구 개발되고 있다. 이러한 서비스간 연동 기법은 IoT에서 요구되는 서비스 연계와 그 성격이 매우 유사하며 동일 기술의 활용을 통하여 보다 손쉽게 서비스를 구성할 수 있다.

2. Kubernetes 특징점 (IoT 측면에서)

Kubernetes는 클라우드에서 자원 관리 및 서비스 배포를 위하여 제시되고 있는 오픈 소스 프로젝트로써, 이를 위한 다양한 기술을 제공하고 있다. IoT의 측면에서 볼 때 Kubernetes는 다음과 같은 특징을 가진다.

소프트웨어 배포 기술. Kubernetes는 Docker 기술을 활용하여 특정 소프트웨어를 특정 하드웨어 자원에 배포할 수 있는 기능을 포함한다. 본 소프트웨어 기술을 IoT에 적용하게 될 경우 한 개의 하드웨어 상에 다양한 IoT 소프트웨어 배포를 통하여 유연한 서비스 구성을 가져올 수 있는 장점을 제공할 수 있다. 예를 들어 GPU가 설치되어 있는

하드웨어가 있을 경우, object detection을 위한 소프트웨어의 배포를 통하여 해당 하드웨어를 Object Detector로 활용하거나 혹은 이미지 분석기를 배포하여 다른 형태의 분석기로 활용할 수 있다.

서비스 구성 기술. Kubernetes는 YAML 형식으로 서비스와 서비스간의 관계를 기술하는 방법을 제공하며, 또한 동일 클러스터 내에 local DNS를 제공하여 서비스간의 discovery를 보다 손쉽게 제공할 수 있다. 이러한 기능의 활용을 통하여 IoMT에서 서비스간 연계를 보다 손쉽게 제공할 수 있다.

GPU 자원 관리 기술. Kubernetes는 일반적인 컴퓨팅 자원 (CPU, 메모리, 스토리지 등)의 특정 하드웨어 자원의 지원을 통하여 device-plug-in을 제공하고 있으며, AMD, Nvidia등 GPU 관련 업체는 각 사의 device-plug-in을 개발 및 제공하고 있다. 이러한 GPU 자원 관리 기술을 통하여 클러스터 내에 영상 처리를 위한 가속기의 유무를 손쉽게 파악할 수 있으며 이에 대한 자원 관리 및 서비스 배포를 활용할 수 있다.

서비스 구성 기술. Kubernetes는 YAML 형식으로 서비스와 서비스간의 관계를 기술하는 방법을 제공하며, 또한 동일 클러스터 내에 local DNS를 제공하여 서비스간의 discovery를 보다 손쉽게 제공할 수 있다. 이러한 기능의 활용을 통하여 IoMT에서 서비스간 연계를 보다 손쉽게 제공할 수 있다.

3. Kubernetes 기반 IoMT배포 기법

IoMT 서비스를 활용하기 위한 Kubernetes 기반의 서비스 운용을 위해서는 아래와 같은 절차가 필요하다.

1) Docker Container 기반 IoMT 서비스 구현

Kubernetes를 통하여 IoMT 서비스의 제어를 위하여 우선 각 IoMT 서비스는 Kubernetes를 통하여 배포 가능한 형태로 구성되어야 한다. 이는 Docker, Container 등 Kubernetes에 호환될 수 있는 다양한 형상을 지원한다. 저자는 지난 논문에서 IoMT를 Docker Container 형상으로 구성할 수 있는 방법을 제안한 바 있다.

2) 마이크로서비스 아키텍처 기반 서비스 구성

최종적으로 제시될 end-to-end service의 아키텍처를 정의한다. 이는 어떠한 형태의 아키텍처도 사용될 수 있으나 호환성 및 서비스 관리의 용이성을 위하여 클라우드 기술에서 활용되는 마이크로서비스의 구성을 따를 수 있다. 마이크로 서비스 아키텍처는 전체 서비스를 구성함에 있어 각각의 요소 기술을 단위 서비스 (마이크로서비스)로 정의하고, 각 단위 서비스를 loosely coupled 형태로 연계하는 기법을 의미한다.

3) 서비스 연결 구성

상기와 같은 아키텍처 및 컨테이너가 준비되면 각 서비스간의 연결을 위한 서비스를 구성한다. 이는 Kubernetes에서 제공하는 설정 문법에 따라 JSON 혹은 YAML 형태로 정의된다. Kubernetes는 각 클러스터 내에 존재하는 단위 서비스에 대한 local DNS를 제공하고 있어 이를 통한 손쉬운 설정의 구성이 가능하다. 각 마이크로서비스는 로컬 클러스터 내에서 unique한 이름을 가지며, 서비스 제공을 위한 포트 번호를 본 설정을 통하여 제공한다.

4) 클러스터 구성 및 배포

각 서비스를 실제 하드웨어에 배포하기 위하여 배포를 위한 클러스터를 구성한다. 클러스터는 클라우드에 설치되어 있는 가상 머신 혹은

로컬에 존재하는 하드웨어에 Kubernetes를 설치하고 단일 클러스터로 설정함으로써 이루어진다. 일단 클러스터가 생성되고 나면 클러스터에 존재하는 컴퓨팅 자원에 대하여 특정 서비스의 배포 및 관리를 Kubernetes에 요청할 수 있다.

서비스 구성 기술. Kubernetes는 YAML 형식으로 서비스와 서비스간의 관계를 기술하는 방법을 제공하며, 또한 동일 클러스터 내에 local DNS를 제공하여 서비스간의 discovery를 보다 손쉽게 제공할 수 있다. 이러한 기능의 활용을 통하여 IoMT에서 서비스간 연계를 보다 손쉽게 제공할 수 있다.

4. 실험

상기의 구성에 대한 검증을 위하여, 영상 처리 기반 서비스 구성을 위한 마이크로서비스 아키텍처 서비스의 Kubernetes 클러스터에 대한 배포 실험이 진행되었다. 본 실험은 RESTful 인터페이스를 가지는 영상 처리 마이크로서비스를 통하여 구성되었으며, 이는 향후 손쉽게 IoMT 마이크로 서비스로의 대체 및 구현이 가능하다. 구성은 IP Camera, 영상 처리 서비스 (Object Detection) 및 영상 모니터링 단말로 이루어졌으며 각각 마이크로서비스로 구현되었다. 또한 이를 위한 클러스터는 Nvidia GPU를 가지는 PC 4대의 조합으로 구성되었다.

구성된 마이크로 서비스는 3.4GB의 크기를 가지는 Docker container로 구성되었으며, 실제 Kubernetes를 통한 서비스의 배포에는 약 3분의 배포 시간이 소요되었다. 각 마이크로서비스는 local DNS를 통하여 object_detect 및 monitoring의 hostname을 가지고 통신할 수 있도록 구성되었으며, 설정에 따라 분석된 영상이 단말에 표현되는 것을 확인할 수 있다.

5. 결론

본 논문은 IoMT 표준에 대하여 IoMT 서비스 단말들간의 조합을 통한 end-to-end service를 Kubernetes를 활용하여 구성하는 방법을 제시한다. 본 논문에서 제시한 방법은 클라우드에서 활용되고 있는 Docker 및 Kubernetes를 활용한 IoMT 소프트웨어의 배포 및 설정 기법으로써 영상 처리 기반 소프트웨어 모듈을 활용하여 제시된 구성의 활용성을 확인하였다. 향후 IoMT 소프트웨어 모듈 대체 및 인터페이스의 검증이 추가로 진행될 예정이다.

감사의 글

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 1711075689, AI 어플리케이션을 지원하는 IoT 연동 분산 Edge 클라우드 기술 개발)

참고문헌

- [1] ISO/IEC 23093, Information Technology - Internet of Media Things
- [2] Kubernetes, <http://kubernetes.org>
- [3] 금승우, 문재원, 김영기, "IoMT 표준 기반 Object Detection 서비스 제공을 위한 미디어 분석 서비스 운용 기술," 2019 방송미디어공학회 하계학술대회.