

스마트 빌딩을 위한 IoT 클라우드 플랫폼의 성능 평가

박정규¹, 박은영^{2*}

¹창신대학교 컴퓨터소프트웨어공학과, ²신한대학교 임상병리학과

Performance Evaluation of IoT Cloud Platforms for Smart Buildings

Jung Kyu Park¹, Eun Young Park^{2*}

¹Department of Computer Software Engineering, Changshin University

²Department of Biomedical Laboratory Science, Shinhan University

요약 스마트 빌딩 (Smart Building)은 사물 인터넷 (IoT; Internet of Things) 장치와 클라우드 서비스에서 모두 사용될 수 있는 응용 프로그램을 의미한다. 최근 Amazon, Google 및 Microsoft와 같은 클라우드 서비스 제공 업체는 IoT 장치에서 클라우드 애플리케이션 서비스를 제공할 수 있는 IoT 클라우드 플랫폼을 제공하고 있다. Postscapes에 따르면 최근에 152 개의 IoT 클라우드 플랫폼이 존재하고 있으며 스마트 빌딩 구현을 위해 하나를 선택하는 것은 어려운 문제이다. 본 논문에서는 실험을 위해서 마이크로소프트 Azure IoT 허브와 아마존의 AWS(Amazon Web Services) IoT를 선택하였다. 두 개의 IoT 클라우드 플랫폼을 선택하고 스마트 빌딩 관점에서 평가하였다. IoT 클라우드 플랫폼을 평가하기 위해 두 가지 다른 IoT 클라우드 플랫폼을 활용하였고, 일반적인 스마트 빌딩 시나리오를 가정하여 프로토타입을 구현하였다. 실험은 IoT 클라우드 플랫폼을 사용하여 시스템을 개발하는 과정에서 얻은 정보와 경험을 기반으로 수행하였다. 이 평가 결과는 스마트 빌딩을 위해 IoT 클라우드 플랫폼을 선택할 때 활용될 수 있다.

Abstract A Smart Building, one that uses automated processes to control its operations, refers in this study to one that uses both Internet of Things (IoT) devices and cloud services software. Cloud service providers (e.g. Amazon, Google, and Microsoft) have recently provided IoT cloud platform application services on IoT devices. According to Postscapes, there are now 152 IoT cloud platforms. Choosing one for a smart building is challenging. We selected Microsoft Azure IoT Hub and Amazon's AWS (Amazon Web Services) IoT. The two platforms were evaluated and selected from a smart building perspective. Each prototype was evaluated on two different IoT platforms, assuming a typical smart building scenario. The selection was based on information and experience gained from developing the prototype system using the IoT cloud platform. The assessment made in this evaluation may be used to select an IoT cloud platform for smart buildings in the future.

Keywords : Cloud Service Provider, Embedded System, IoT, Raspberry Pi, Smart Building

1. 서론

스마트 빌딩(Smart Building)은 사물 인터넷 (IoT; Internet of Thing, 이하 IoT) 장치와 클라우드 서비스에서 공통적으로 사용되는 응용 프로그램 (Common

program)이다. 일반적으로 스마트 빌딩은 여러 대의 IoT 장비가 포함되며 이 장비는 스마트 빌딩을 이용해서 구성될 수 있다. 스마트 빌딩 시스템 데이터 처리를 수행할 수 있으며 기기를 원격으로 제어할 수 있도록 클라우드에 연결되어 있다 [1, 2].

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2018R1C1B5046282).

*Corresponding Author : Eun Young Park(Shinhan University)

email: 71eypark@gmail.com

Received February 24, 2020

Accepted May 8, 2020

Revised March 9, 2020

Published May 31, 2020

최근 Amazon, Google 및 Microsoft와 같은 클라우드 서비스 제공 업체(CSP; Cloud Service Provider)는 IoT 장치에서 클라우드 애플리케이션 서비스를 제공할 수 있는 IoT 클라우드 플랫폼을 제공하고 있다. IoT 클라우드 플랫폼은 장치, 네트워크 및 애플리케이션을 통합하고 있다 [2]. 이러한 플랫폼은 IoT 장치가 구축되는 에코 시스템을 제공하여 IoT 솔루션을 지원하고 활성화하기 때문에 사용자에게 구현 복잡성을 숨길 수 있다. IoT 클라우드 플랫폼이 지원할 수 있는 IoT 솔루션의 예로 스마트 라이트(Smart lights)를 들 수 있다. 스마트 라이트는 건물의 조명으로 사용되며 IoT 클라우드 플랫폼에 연결하여 원격 제어 및 스케줄링이 가능하다 [3]. 이 조명은 아침에 조명이 자동으로 켜지고 저녁에 꺼지는 데 활용할 수 있고 이를 통해 에너지를 절약할 수 있다.

Postscapes에 따르면 최근에 152 개의 IoT 클라우드 플랫폼이 존재하고 있으며 스마트 빌딩 구현을 위해 하나를 선택하는 것은 어려운 현실이다 [4]. 또한 다양한 IoT 클라우드 플랫폼을 조사하고 평가하는 논문은 거의 없는 것으로 보인다.

본 논문에서는 대부분 플랫폼의 기능적 특성과 성능에 대해서 분석하고 논의한다 [6-7]. 논문의 목적은 IoT 클라우드 플랫폼을 사용하는 애플리케이션 개발 프로세스에 대한 비교 연구로 기존 논문을 보완하는 것이다. 본 논문에서는 현재 가장 많이 사용하는 IoT 클라우드 플랫폼 2개를 선택 후 평가를 진행하였다 [8].

논문 연구를 수행하기 위해 스마트 빌딩에 대한 일반적인 시나리오를 가정하였고 기기 프로토타입을 구현하였다. 실험 시나리오에 적용하는 일반적인 스마트 빌딩 기기는 난방, 환기 및 에어 컨디셔닝 (HVAC; Heating, Ventilation, Air Conditioning) 시스템이다. 시나리오 상에서 기기의 프로토타입은 IoT 클라우드 플랫폼 상의 애플리케이션과 연결된다. 실험을 수행하기 위해서 다음과 같은 조건이 필요하다.

- 조건 1. CSP는 기술 지원을 제공해야 함
- 조건 2. 플랫폼에 직접 접근이 가능해야 함
- 조건 3. 플랫폼은 사용 매뉴얼을 제공해야 함
- 조건 4. 플랫폼은 무료 체험이 가능해야 함
- 조건 5. 플랫폼은 하드웨어 종속성이 없어야 함
- 조건 6. 플랫폼은 예제 코드를 제공해야 함

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 설명한다. 3장에서는 실험을 위한 시스템 디자인에 대해

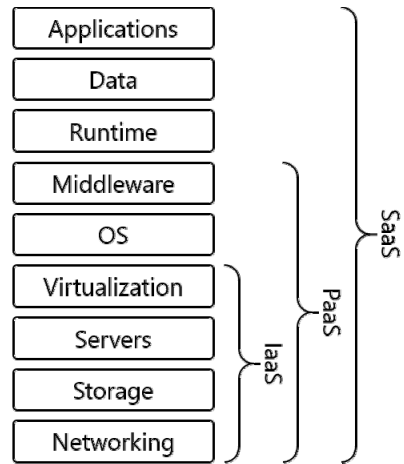


Fig. 1. Groups of Cloud Services

서 설명한다. 4장에서는 실험 환경을 설명하고, 5장에서는 실험 결과를 보여준다. 그리고 마지막 6장에서 결론을 맺는다.

2. 관련 연구

2.1 IoT 클라우드 플랫폼 (Cloud Platform)

사물 인터넷이 활성화 되면서 가정 뿐만 아니라 기업에서도 IoT 수요가 폭발적으로 증가하였다. 특히 기업에서는 사용하는 모든 IoT 장치를 연결하고 데이터를 수집하는 것을 필요로 하였다. 또한, 수집된 데이터를 이용하여 분석하는 것을 새로운 의사결정에 사용하고자 하였다. 컴퓨팅 시장에서는 이를 위해 IoT 클라우드 플랫폼을 제공하기 시작하였다. IoT 클라우드 플랫폼을 다음과 같이 3가지 기능을 제공해야 한다. 1) 관리자의 개입 없이 노드의 변경이 동적으로 가능, 2) IoT 시스템 전체를 확인할 수 있는 관리 도구 제공, 3) IoT 토폴로지 및 데이터 흐름의 실시간 표시 [9]

2.2 클라우드 서비스

클라우드 서비스 기능은 IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service)의 세 가지 그룹으로 나눌 수 있다 [10]. Fig. 1은 클라우드 공급자가 관리하는 서비스를 3가지로 나누어 표시하고 있다.

IaaS 서비스에서는 데이터 저장 스토리지, 서버, 네트

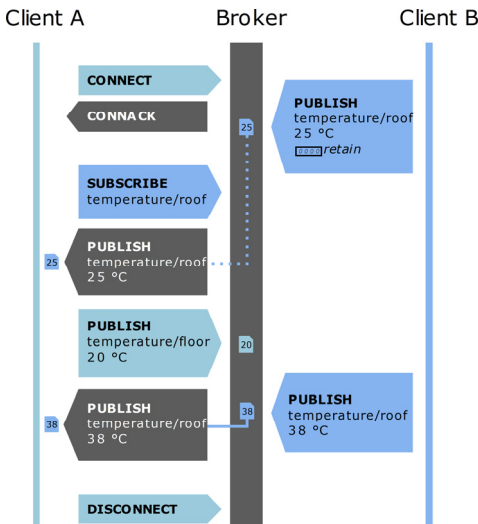


Fig. 2. Example of MQTT Connection

워킹을 제공한다. 서버는 일반적으로 리소스를 효율적으로 사용하고 비용을 절감할 수 있도록 가상화를 통해 접근이 가능하다. IaaS 서비스를 구입할 때는 일반적으로 가상 머신에 대한 액세스 권한이 부여된다. 데이터 저장은 데이터베이스에서 물리적 하드 드라이브 저장 공간에 이르기까지 모든 스토리지를 의미한다 [10].

PaaS는 소비자가 지원되는 프로그래밍 언어와 라이브러리를 이용하여 어플리케이션을 개발하고 클라우드에 바로 배포할 수 있게 하는 서비스를 제공한다. PaaS를 사용하면 클라우드 공급자가 제공하는 기본 애플리케이션 인프라 (IaaS)를 사용할 수 있는 이점이 있다. 클라우드 제공자는 기본 애플리케이션 인프라의 설치, 구성 및 운영을 책임진다. 이를 통해 소비자는 확장성, 하드웨어 유지 관리 및 하드웨어 비용에 대한 걱정 없이 응용 프로그램 개발에 집중할 수 있다 [10].

SaaS는 클라우드에서 애플리케이션 소프트웨어를 운영하는 서비스이다. SaaS 애플리케이션 사용자는 애플리케이션 또는 애플리케이션이 실행되는 플랫폼을 관리하지 않는다. SaaS의 예로 Google Docs가 있으며 여기서 Google은 애플리케이션을 관리하는 클라우드 서비스 제공 업체이다.

2.3 Message Queue Telemetry Transport (MQTT)

메시지 큐 텔레메트리 트랜스포트 (MQTT; Message Queue Telemetry Transport)는 ISO 표준으로 IoT에

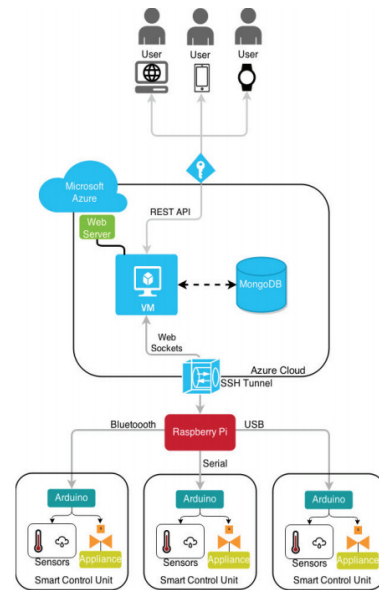


Fig. 3. Architecture of Smart Building using Cloud Computing

적합하고 가벼운 유연한 네트워크 프로토콜이다. MQTT는 네트워크에서 두 개의 기본 엔티티인 메시지 브로커와 여러 클라이언트를 정의한다. 메시지 브로커는 클라이언트로부터 모든 메시지를 수신하여 대상 클라이언트 라우트 한다. 클라이언트는 브로커와 상호 작용하여 수신하거나 송신한다 [10].

MQTT에서 메시지 송수신은 공개(Publish) 및 구독(Subscribe) 모델을 기반으로 한다. MQTT 브로커에 연결된 클라이언트는 브로커의 모든 메시지 주제를 구독할 수 있다. 클라이언트가 어떤 주제 (Topic)의 메시지를 공개하면 MQTT 브로커는 해당 주제를 구독하는 모든 클라이언트에게 메시지를 전달한다. Fig. 2는 MQTT를 이용한 메시지 전송을 그림으로 표현하고 있다. 클라이언트 A는 브로커에 연결되어 있고 클라이언트 B는 '온도/옥상 (temperature/roof)' 25 °C를 공개한다. 클라이언트 A는 '온도/옥상'를 구독하고 있기 때문에 25 °C를 수신할 수 있다. 그러나 클라이언트 A에서 공개한 '온도/마루 (temperature/floor)' 20 °C는 브로커에서 전달되지만 클라이언트 B는 '온도/마루'를 구독하고 있지 않기 때문에 온도를 수신하지 않는다.

2.4 IoT 프레임워크를 위한 스마트 빌딩

Enrique는 스마트 빌딩이 클라우드 컴퓨팅을 활용할 수 있도록 IoT 프레임워크를 개발했다 [11]. 연구에서는

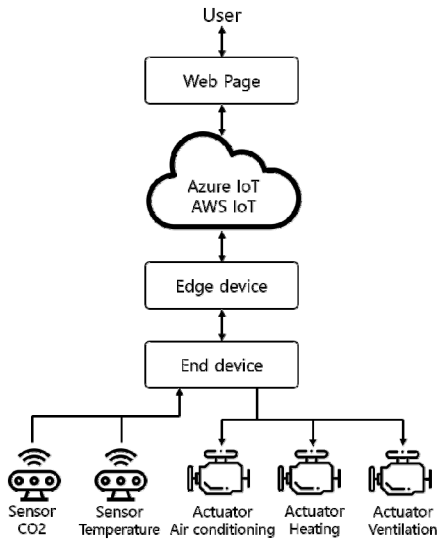


Fig. 4. Architecture of IoT Sensing System

스마트 빌딩의 장점과 구현해야 하는 이유에 대해 설명하였다. 또한 스마트 빌딩을 구현하고 시스템의 다양한 구성 요소를 실험하였다. 관련 연구에서는 통신 및 클라우드 컴퓨팅을 조사하였다. 조사를 통하여 클라우드 컴퓨팅의 장점을 논의하였고 통신 표준이 얼마나 중요한지를 설명하였다. 이 논문의 핵심은 클라우드 컴퓨팅을 활용하면서 스마트 빌딩을 위한 IoT 프레임 워크를 제시한 것이다. 이 프레임 워크는 서로 다른 통신 프로토콜을 통해 게이트웨이에 연결된 여러 개의 장치를 사용한다. 그리고 게이트웨이는 차례로 클라우드 서비스에 연결된다. Fig. 3은 스마트 빌딩의 구조를 표시하고 있다.

3. 시스템 디자인

논문에서 제안하는 시스템은 단말기(end device), 에지 장치(edge device), IoT 클라우드 플랫폼 및 웹 페이지(web page)의 4 부분으로 구성된다. Fig. 4는 4개의 부분이 연결된 시스템을 보여주고 있다.

3.1 단말기 (End Device)

단말기는 주변 정보를 측정하는 센서와 외부 기기를 동작시켜주는 액추에이터와 연결하고 있으며 실험에서는 아두이노 우노 (Arduino Uno)를 사용하였다. 단말기는 에지 장치에서 받은 명령을 기반으로 액추에이터를 제어

한다. 또한 단말기의 마이크로 컨트롤러는 센서에서 측정 값을 가져오고 에지 장치의 요청이 있을 때 측정 값을 에지 장치로 전송한다.

3.2 에지 장치 (Edge Device)

에지 장치로 Raspberry Pi B+를 사용했으며 리눅스 운영체제 (Raspbian)로 동작한다. 에지 장치의 목적은 IoT 클라우드 플랫폼과 단말기 사이의 게이트웨이 역할을 하는 것이다. 이 구조는 최근 많은 IoT 프레임워크에서 사용되고 있으며 앞서 소개한 Enrique의 연구에서도 사용되었다.

보통 단말기의 프로세서는 보안 및 인증을 구현하기에 성능이 부족하기 때문에 에지 장치는 이 문제를 나누어 처리해야 한다. 또한, 스마트 빌딩과 관련하여 여러 개의 단말기를 사용하는 것이 일반적이다. 에지 장비를 사용하여 더 많은 단말기를 시스템에 통합할 수 있다. Fig. 3과 같이 하나의 에지 장치 (Raspberry Pi)에 여러 개의 단말기를 연결할 수 있다.

3.3 IoT 클라우드 플랫폼

IoT 클라우드 플랫폼은 실험하고자 하는 시스템의 핵심 요소이다. 시스템의 모든 논리는 클라우드에 배치되고 IoT 클라우드 플랫폼은 시스템의 다른 부분의 규제, 메시지 라우팅, 통합을 담당한다. 본 논문에서는 2개의 IoT 클라우드 플랫폼을 선택하였으며 4장 실험 및 결과에서 설명한다.

3.4 웹 페이지

Fig. 4의 시스템 상단에 있는 웹 페이지의 목적은 센서 데이터와 단말기에 연결된 외부 장치의 상태를 시각화하고 원격 제어를 가능하게 하는 것이다. 웹 페이지는 최종 사용자가 시스템과 상호 작용하는 유일한 부분이다. 따라서 사용자가 원하는 온도 범위와 같은 설정을 변경할 수 있도록 하는 역할도 한다.

Fig. 4에서 에지 장치의 요청에 의해 단말기를 현재 온도와 CO₂ 값을 측정하고, 값을 수신 후 클라우드 플랫폼으로 전달된다. 전달된 데이터를 웹 페이지를 통해서 실시간으로 모니터링이 가능하다. Fig. 5에서는 수신된 데이터를 웹으로 표시하고 있다.

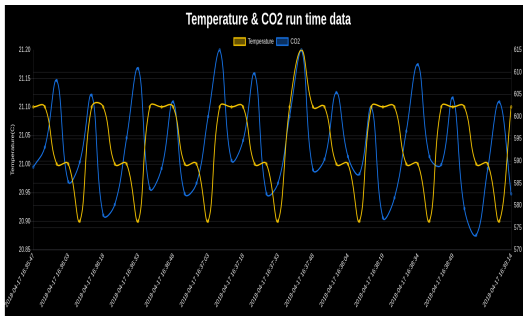


Fig. 5. Web page for temperature and Co2 sensor data

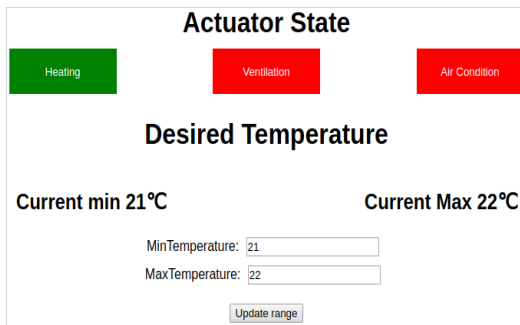


Fig. 6. Web page for actuator status display and remote control

Fig. 6의 웹 페이지에서는 현재 액추에이터 상태를 표시하고 온도 범위 설정이 가능하다.

4. 실험 환경

4.1 실험 시나리오

아래에 제시된 시나리오는 시스템의 요구사항을 수집하는 데 사용될 수 있는 시나리오다. 또한 시나리오는 시스템의 기능도 설명한다. IoT 클라우드 플랫폼을 이용하여 시나리오에 기술된 시스템을 구현함으로써 비교 연구를 위한 정보와 경험을 수집할 수 있다.

실험 시나리오: 회사 동료들은 회의를 하기 위해 스마트 빌딩의 내의 회의실로 들어간다. 실내 온도와 CO₂ 레벨은 온도 및 CO₂ 센서에 의해 모니터링 된다. 회의실의 온도가 빌딩 관리자가 정한 범위 내에 있지 않으면 난방이나 에어컨이 자동으로 가동된다. CO₂수치가 정상치(600ppm) 이상으로 상승하면 실내 공기를 정화하기 위해 자동으로 환기가 시작된다. 관리자는 언제든지 난방,

환기, 에어컨을 원격으로 제어할 수 있다.

위 시나리오에서는 클라우드로 센서 데이터 전송, 장치의 원격 제어, 의사 결정 등과 같은 IoT 클라우드 플랫폼의 기능들이 사용된다.

4.2 IoT 클라우드 플랫폼 선택

많은 IoT 클라우드 플랫폼 중에서 실험에 맞는 플랫폼을 선택하기 위해서 Postscapes에서 정리한 문서를 참고하였다. Postscapes 에서는 약 152개의 IoT 클라우드 플랫폼을 분석해 놓았다. 이 목록을 기반으로 1장에서 설명한 6개의 조건에 맞는 플랫폼을 18개 선택하였다. 선택된 플랫폼은 Table 1에서 표시하고 있다.

Table 1. List of IoT Cloud Platforms

Platform	Platform	Platform
Amazon AWS IoT	AT&T M2X	Cloudplugs
Cloudthing.io	ForgeRock	GE Predix
Google Cloud IoT	GroveStreams	IBM Watson IoT Platform
Intel RIoT Platform	Microsoft Azure IoT	Oracle Internet of Things Cloud
PTC ThingWorx	PubNub	scriptr.io
Splunk	thethings.io	Thing+

실험에서 비교 대상으로 선정된 플랫폼은 마이크로소프트 Azure IoT 허브와 아마존의 AWS(Amazon Web Services) IoT로, 둘 다 PaaS 서비스이다 [12, 13]. Azure IoT 허브와 AWS IoT는 마이크로소프트와 아마존 양대 클라우드 업체에서 제공하고 있다. 이 두 업체를 선택한 첫 번째 이유는 최근 클라우드 플랫폼 개발에서 많이 사용되고 있기 때문이며, 두 번째 이유는 실험 조건에 맞고 두 IoT 클라우드 플랫폼을 평가하는 연구가 없는 것으로 조사되었기 때문이다 [8].

4.3 실험 환경

제안하는 시스템을 테스트하기 위해 아두이노 우노, 라즈베리 파이, 센서를 이용하여 Fig. 7와 같이 구성하였다. 실험에서 CO₂ 값이 자주 변하기 않기 때문에 Fig. 7에 버튼 센서를 이용하여 CO₂ 값을 증가/감소 시킬 수 있도록 코드를 작성하였다. 또한 버튼을 이용하여 난방/냉방 기능을 수동으로 제어할 수 있도록 하였다. 난방과 냉방을 수동으로 제어할 수 있게 됨으로써 온도 센서가 측정하는 값을 제어하면서 실험하였다.

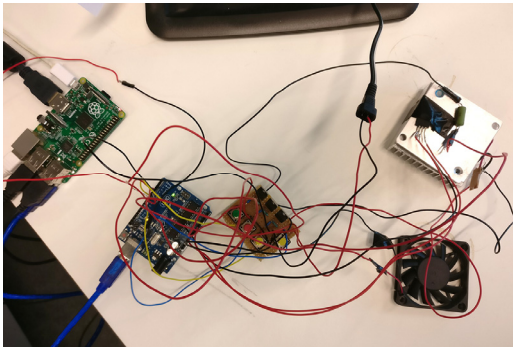


Fig. 7. Architecture of IoT Sensing System

5. 실험 결과

실험은 Azure IoT 허브와 AWS IoT 두 IoT 클라우드 플랫폼에서 진행하였으며 통신, 장치, 데이터 처리, 개발 부분으로 나누어 평가하였다. 논문에서는 통신, 장치 관리 부분의 결과를 중점적으로 살펴보았다. 실험 결과 점수는 시나리오를 만족하고 기능을 만족할 경우 5점 만점을 기준으로 하였다.

5.1 통신 관리

5.1.1 단말기와 클라우드 통신

Azure IoT 허브와 AWS IoT 모두 단말기에서 클라우드로 데이터를 전송할 때 JSON 메시지 데이터 형식을 사용하였다. 또한 제공하는 샘플 코드, 문서 및 장치 간 통신에 대한 문서화 구현 부분에서 Azure IoT 허브와 AWS IoT 사이에 큰 차이점이 없었다.

Azure IoT Hub는 다양한 코드 샘플을 제공하였고, 단말기에서 클라우드로 메시지를 전송할 때 표준화된 방법을 사용하여 클라우드 통신 부분에서 5점 만점을 부여하였다. AWS IoT는 Azure IoT 허브와 같은 이유로 5점 만점을 부여하였다.

5.1.2 클라우드와 단말기 통신

Azure IoT 허브에서는 장치가 수신할 데이터 항목의 사전 정의가 가능하다. 이에 비해 Azure IoT 허브는 전송할 항목을 더 자유롭게 선택할 수 있고 AWS IoT는 전송되는 데이터를 가능한 한 적게 제한하고 있다. 두 방법 모두 구현 방법은 이해가 되지만 실험 시나리오에서는 Azure IoT 허브가 더 적합하다.

Azure IoT 허브는 구현이 쉽고 장치간의 콜백은 시나리오에서 잘 동작하여 클라우드와 단말기 간 통신에서 5점 만점을 부여하였다. AWS IoT도 클라우드와 단말기 간 통신 기준에서 5점 만점을 부여하였다. 델타 콜백을 사용한 AWS IoT의 구현은 잘 작동했으며 설계가 잘 되어 있었다.

5.1.3 메시지 라우팅

Azure IoT 허브의 라우팅 규칙과 비교하여 AWS의 규칙은 보다 직관적으로 생성 및 구성이 가능하다. AWS의 서비스 통합은 규칙에서 직접 수행되는 반면, Azure IoT 허브의 라우팅 규칙은 엔드 포인트를 구성하고 Azure 서비스와 별도로 통합해야 하는 방식이라 더 복잡하다. 또한 라우팅되는 엔드 포인트는 Azure 서비스와 통합되어야 한다.

Azure IoT 허브는 메시지 라우팅 항목에서 4점 (5점 만점)을 받았다. 그 이유는 다른 Azure 서비스와의 통합이 복잡하고 이해하기 어렵기 때문이다. AWS IoT는 메시지 라우팅 항목에서 5점을 받았다. AWS IoT의 메시지 라우팅이 이해하기 쉽고 AWS IoT와 다른 AWS 서비스 간 라우팅을 위한 직관적인 인터페이스를 제공하기 때문이다.

Fig. 8에서 통신 관리의 실험 3가지를 하나의 그래프로 요약하였다.

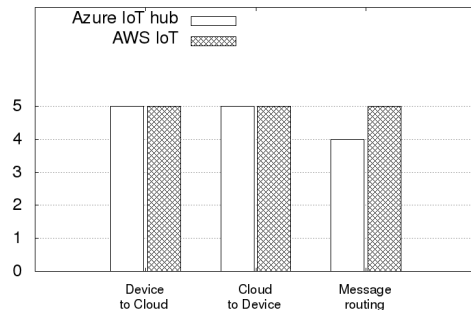


Fig. 8. Result of Communication Management

5.2 장치 관리

5.2.1 장치 모니터링

장치 모니터링 기준에서 Azure IoT Hub와 AWS IoT의 차이점은 그렇게 크지 않지만 Azure IoT의 장치 관리 샘플이 AWS IoT 비해 이해가 잘 되게 작성되어 있다는 것이다. 장치 모니터링 기준으로 평가할 때 장치 간

모니터링을 쉽게 구현할 수 있게 좋은 샘플과 다양한 문서를 제공하고 있어 5점 만점을 부여하였다. AWS IoT는 Azure IoT 허브와 같은 5점 만점을 부여하였다. 제공되는 샘플이 다양하지 않았지만, 시스템 구현하기에 충분한 정보를 제공하였다.

5.2.2 장치 프로비저닝

장치 프로비저닝 측면에서 Azure IoT 허브에 비해 AWS IoT가 더 복잡했다. AWS에서는 보안 인증서를 생성하고 다운로드 하여 장치에 설치하면 되지만 Azure에는 연결 키만 필요하기 때문이다. 또한, AWS에는 사용자 정의 정책이 필요하지만 Azure의 권한은 기본 용도로 사전 구성되어 있었다.

Azure IoT 허브는 장치 프로비저닝 기준 5점 만점을 부여하였다. 그 이유는 장치 프로비저닝 단계가 매우 간단하고 Azure에서 수행하는 방법을 단계별 지침으로 쉽게 제공하기 때문이다. AWS IoT는 디바이스 프로비저닝 기준에서 4점을 기록했다. 그 이유는 프로비저닝 프로세스에 인증서를 생성, 다운로드 및 연결하려는 장치로 이동하는 등의 많은 단계가 포함되어 있기 때문이다.

Fig. 9에서 장치 관리의 실험을 하나의 그래프로 정리하였다.

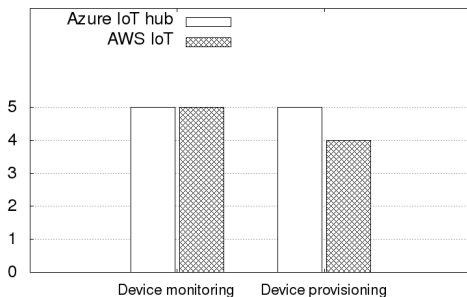


Fig. 9. Result of Device Management

6. 결론

본 논문에서는 스마트 빌딩의 관점에서 IoT 클라우드 플랫폼을 평가하는 것이다. 실험 결과는 Azure와 AWS 모두 IoT 프레임워크 최고의 입지를 보이기 때문에 예상한 것과 같이 높은 점수를 받을 수 있었다. 또한, Azure IoT Hub 및 AWS IoT가 제공하는 많은 기능이 유사한

방식으로 구현되어 있었다. 예를 들어 Azure의 라우팅 규칙 및 AWS의 규칙을 들 수 있다. 두 플랫폼 간의 가장 큰 차이점은 문서화, 사용 편의성 및 이해도를 들 수 있다. 또한, 클라우드와 장치 간 통신이 구현되는 방식과 같은 차이점도 있다.

두 플랫폼 간의 차이점은 구현될 시스템에 따라 선택할 수 있다. 예를 들어 스토리지 또는 데이터 처리와 같은 CSP 서비스를 사용하여 시스템을 구성할 경우 AWS IoT가 더 나은 선택이다. AWS IoT는 AWS가 제공하는 여러 서비스 간 메시지 라우팅할 때 직관적인 방법을 제공한다. 그러나 웹 페이지와 같은 외부 응용 프로그램과 통합되어야 하는 시스템의 경우 Azure IoT 허브가 더 적합한 선택이다.

References

- [1] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, V. H. C. de Albuquerque, "A Reference Model for Internet of Things Middleware", *IEEE Internet of Things Journal*, Vol.5, No.2, pp.971-883, Jan. 2018.
DOI: <https://doi.org/10.1109/IJOT.2018.2796561>
- [2] S. Sidid, S. Gaur, "Smart grid building automation based on Internet of Things", *Proceedings of 2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, IEEE, Vellore, India, pp.1-4, April 2017.
DOI: <https://doi.org/10.1109/IPACT.2017.8245201>
- [3] W. Anani, A. Ouda, A. Hamou, "A Survey Of Wireless Communications for IoT Echo-Systems", *Proceedings of 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, IEEE, Edmonton, AB, Canada, pp. 1-6, May 2019.
DOI: <https://doi.org/10.1109/CCECE.2019.8861764>
- [4] T. Harwood. IoT Cloud Platform Landscape [Internet]. Postscapes, 2019 [cited 2019 November 01], Available From: <https://www.postscapes.com/internet-of-things-platforms/> (accessed Feb. 22, 2020)
- [5] T. Pflanzner, A. Kertesz, "A survey of IoT cloud providers", *Proceedings of 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, Opatija, Croatia, pp.730-735, June 2016.
DOI: <https://doi.org/10.1109/MIPRO.2016.7522237>
- [6] P. Ganguly, "Selecting the right IoT cloud platform", *Proceedings of 2016 International Conference on Internet of Things and Applications (IOTA)*, IEEE, Pune, India, pp.316-320, 2016.
DOI: <https://doi.org/10.1109/IOTA.2016.7562744>

[7] P. P. Ray, "A survey of IoT cloud platforms", *Future Computing and Informatics Journal*, Vol.1, No.1-2, pp.35-46, Dec. 2016.
DOI: <https://doi.org/10.1016/j.fcij.2017.02.001>

[8] L. Dignan. Top cloud providers in 2020: AWS, Microsoft Azure, and Google Cloud, hybrid, SaaS players [Internet]. ZDNet, 2020 [cited 2020 March 07]. Available From: <https://www.zdnet.com/article/the-top-cloud-providers-of-2020-aws-microsoft-azure-google-cloud-hybrid-saas/> (accessed Mar. 07, 2020)

[9] M. A. L. Peña, I. M. Fernández, "SAT-IoT: An Architectural Model for a High-Performance Fog/Edge/Cloud IoT Platform", *Proceedings of 2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, IEEE, Limerick, Ireland, pp.633-638, 2019.
DOI: <https://doi.org/10.1109/WF-IoT.2019.8767282>

[10] M. Alrashoud, E. Hazza, F. Alqahtani, M. Al-Hammadi, A. Abhari, A. Ghoneim, "Cognitive and Hierarchical Fuzzy Inference System for Generating Next Release Planning in SaaS Applications", *IEEE Access*, Vol.7, pp.102966-102974, Jul. 2019.
DOI: <https://doi.org/10.1109/ACCESS.2019.2929214>

[11] E. Carrillo, V. Benitez, C. Mendoza, J. Pacheco, "IoT framework for smart buildings with cloud computing", *Proceedings of 2015 IEEE First International Smart Cities Conference (ISC2)*, IEEE, Guadalajara, Mexico, pp.1-6, 2015.
DOI: <https://doi.org/10.1109/ISC2.2015.7366197>

[12] Microsoft. Azure IoT Hub [Internet]. Microsoft, 2019 [cited 2019 October 01], Available From: <https://azure.microsoft.com/en-us/services/iot-hub> (accessed Feb. 22, 2020)

[13] Amazon. AWS IoT [Internet]. Amazon, 2019 [cited 2019 October 01], Available From: <https://aws.amazon.com/iot> (accessed Feb. 22, 2020)

박 은 영(Eun Young Park)

[정회원]



- 2006년 : 경희대학교 교육학과(화학교육) 교육학석사
- 2014년 : 서울시립대학교 에너지 환경시스템공학과 박사
- 2017년 ~ 현재 : 신한대학교 임상 병리학과 연구교수

<관심분야>

Nonlinear computer modeling, 폐기물 자원화, 환경화학

박 정 규(Jung Kyu Park)

[정회원]



- 2002년 : 홍익대학교 컴퓨터공학과 석사
- 2013년 : 홍익대학교 컴퓨터공학과 박사
- 2014년 ~ 2016년 : 단국대학교 연구교수
- 2018년 ~ 현재 : 창신대학교 컴퓨터소프트웨어공학과 교수

<관심분야>

운영체제, 임베디드시스템, 로보틱스