



IoT 디바이스 모니터링, 분석, 제어를 위한 포그컴퓨팅 플랫폼

Fog Computing Platform for Monitoring, Analyzing, and Controlling IoT Devices

저자 (Authors)	이한솔, 변기범, 김명, 홍지만 Hansol Lee, Gibeom Byeon, Jin Ming, Jiman Hong
출처 (Source)	한국정보과학회 학술발표논문집 , 2017.12, 131-133 (3 pages)
발행처 (Publisher)	한국정보과학회 KOREA INFORMATION SCIENCE SOCIETY
URL	http://www.dbpia.co.kr/Article/NODE07322074
APA Style	이한솔, 변기범, 김명, 홍지만 (2017). IoT 디바이스 모니터링, 분석, 제어를 위한 포그컴퓨팅 플랫폼. 한국정보과학회 학술발표논문집, 131-133.
이용정보 (Accessed)	성균관대학교 자연과학캠퍼스 115.***.238.194 2018/03/22 12:36 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

IoT 디바이스 모니터링, 분석, 제어를 위한 포그컴퓨팅 플랫폼

이한솔⁰, 변기범, 김명, 홍지만*

송실대학교, 컴퓨터학과

hslee.oslab@gmail.com, gibeom.ssu@gmail.com, jinming088@gmail.com,

jiman@ssu.ac.kr

Fog Computing Platform for Monitoring, Analyzing, and Controlling IoT Devices

Hansol Lee⁰, Gibeom Byeon, Jin Ming, Jiman Hong*

Department of Computer Science and Engineering,

Soongsil University

요 약

포그컴퓨팅은 IoT 디바이스가 생성하는 데이터를 효율적으로 처리하기 위해 제안된 개념이다. 포그컴퓨팅은 포그 서버와 디바이스간의 거리를 좁히고 각 서버가 관리하는 디바이스의 수를 줄임으로써 클라우드 컴퓨팅보다 고품질의 서비스를 제공한다. 본 논문에서는 포그컴퓨팅 환경에 포함된 IoT 디바이스를 효율적으로 모니터링·분석·제어하기 위한 포그컴퓨팅 플랫폼을 제안한다. 제안한 플랫폼은 IoT 디바이스의 활용을 극대화하고 기존의 다양한 포그컴퓨팅 프레임워크들을 쉽게 구현할 수 있게 한다.

1. 서 론

Internet of Things(IoT)의 발전으로 IoT 디바이스가 스스로 환경을 인지하거나 동작을 수행할 수 있게 되었다. 이에 따라 IoT 디바이스들을 관리하는 컴퓨팅 모델의 역할이 더욱 중요해졌다. 기존에는 클라우드 컴퓨팅이 IoT 디바이스 데이터 관리에 주로 사용되었다[1]. 그러나 IoT 디바이스가 생성하는 데이터가 많아지고 이를 처리하는 루틴이 복잡해지면서 새 컴퓨팅 모델인 포그컴퓨팅이 제안되었다. 포그컴퓨팅은 IoT 디바이스와 근거리 통신을 하며 데이터를 관리하는 모델이다. 포그 노드는 근거리 통신을 사용하기 때문에 반응속도가 클라우드 컴퓨팅보다 빠르고, 각 포그 노드에 할당되는 IoT 디바이스가 클라우드 컴퓨팅에 비해 적기 때문에 더 많은 작업을 빠르게 해낼 수 있다.

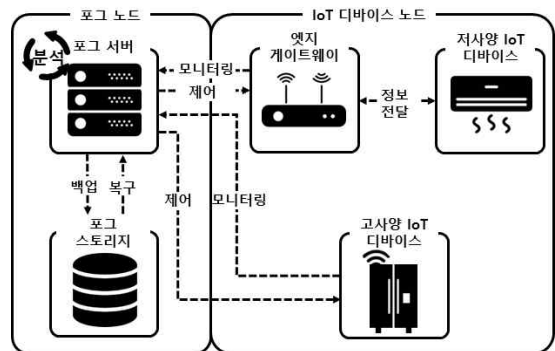
포그컴퓨팅의 포그 노드는 클라우드 서버와 IoT 디바이스 사이에서 자료 백업 및 네트워킹 서비스를 제공한다. 또한 노드와 디바이스가 근거리 통신을 하기 때문에 실시간 응용 프로그램 경험을 제공할 수 있다[2]. 현재 스마트 홈, 스마트 오피스, 스마트 팩토리 등 다양한 곳에 포그컴퓨팅이 활용되고 있다. 그러나 이들 각각에 사용되는 포그컴퓨팅 프레임워크들을 포함하는 솔루션은 존재하지 않는다.

본 논문에서는 IoT 디바이스를 위한 모니터링·분석·제어하기 위한 플랫폼을 제안한다. 제안한 플랫폼은 IoT 디바이스의 상태 및 동작을 모니터링하고, 이를 분석하여 IoT 디바이스의 새로운 실행 환경을 구축할 수 있게 한다. 또한 제안한 플랫폼은 프레임워크 및 디바이스 간 효율적인 API를 정의한다.

2. 포그컴퓨팅 플랫폼

본 논문에서 제안하는 플랫폼은 [그림 2-1]과 같이 포그 노드와 IoT 디바이스 노드로 구성된다. 포그 노드는

포그 서버와 저장용 스토리지로 나눌 수 있다. 포그 서버는 IoT 디바이스의 정보를 모니터링하며 분석한 정보를 토대로 IoT 디바이스를 제어한다.

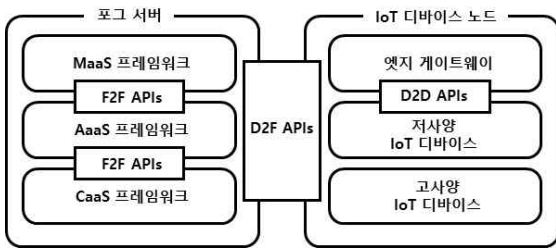


[그림 2-1] 제안한 포그컴퓨팅 플랫폼

IoT 디바이스 노드는 연결된 IoT 디바이스의 사양에 따라 연결 방식이 다르다. 고사양 IoT 디바이스와 달리, 저사양 IoT 디바이스는 단순한 센서와 액추에이터로만 구성되어 있어 포그 서버와 통신하기가 힘들다. 이를 해결하기 위해 엣지 게이트웨이를 연결시켜 포그 서버와 대신 통신하게 한다.

3. 포그컴퓨팅 프레임워크

[그림 3-1]은 본 논문에서 정의하는 포그컴퓨팅의 모니터링·분석·제어 프레임워크 구조를 보여준다. 각 프레임워크는 세가지 모듈로 구성된다. 모듈의 이름과 기능은 [표 3-1]과 같다.



[그림 3-1] 모니터링·분석·제어 프레임워크

[표 3-1] 각 프레임워크 모듈 설명

구분	모듈	역할
MaaS	Device Connector	IoT 기기의 접근 제어
	Data Manager	데이터 분류 및 저장
	Device Monitor	IoT 디바이스 모니터링
AaaS	Change Tracker	생성 및 획득한 데이터의 변화 추적
	Feature Extractor	Change Tracker가 생성한 변화정보로부터 특징 추출
	Integrated Analyzer	Feature Extractor에서 추출된 특징을 분석하여 새 정보 생성
CaaS	Trigger Updater	Integrated Analyzer가 생성한 지식을 기반으로 트리거 갱신
	Module Updater	Device Monitor가 알아낸 디바이스 결함 정보와 벤더가 제공한 소프트웨어 모듈을 통해 각 디바이스 모듈을 갱신
	Device Controller	IoT 디바이스 결함의 원인이 MACaaS에 있는지, 벤더가 제공한 모듈에 있는지 확인

[표 3-2] APIs

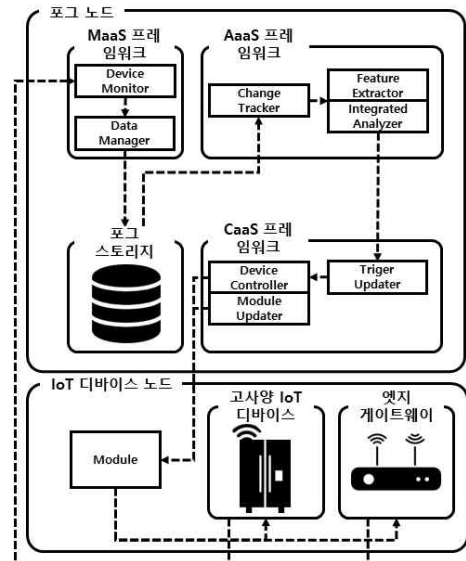
구분	적용범위	API
D2F	Common	send(framework, data) receive(device, data) requestConnection(framework)
	Device ↔ Device Connector	checkConnection(device) accessControl(device) connect(device)
	Device ↔ Data Monitor	monitor(device)
	Device ↔ Module Updater	updateModule(device, module)
	Device ↔ Device Controller	control(device, fault)
F2F	Common	send(framework, data) receive(framework, data)
D2D	Common	send(device, data) receive(device, data)
MaaS	Data Manager	classify(data) save(storage, data)
	Change Tracker	trackChange(data) load(storage, data)
AaaS	Feature Extractor	extractFeature(changeInfo)
	Integrated Analyzer	analyze(feature)
CaaS	Trigger Updater	updateTrigger(knowledge)
	Module Updater	checkUpdate(vendor, module)
	Device Controller	checkFault(device, fault)

제안한 포그컴퓨팅 프레임워크에서 사용하는 함수와 연산은 [표 3-2]에서 정의한 API에 의해 이루어진다. 정

의한 API에는 디바이스 간 통신(D2D), 프레임워크 간 통신(F2F), 디바이스와 프레임워크 간 통신(D2F) 그리고 내부 연산 등이 있다.

프레임워크의 내부 구조를 몰라도 정의된 API를 통해 모니터링·분석·제어를 위한 새 프레임워크와 디바이스를 추가할 수 있다. API 함수들의 실행 흐름은 [그림 3-2]와 같다.

IoT 디바이스 노드에서 생성된 데이터는 모니터링되어 모니터링 프레임워크인 MaaS로 전송된다. 데이터를 받은 MaaS 내 Data Manager는 이 정보를 포그 스토리지에 저



[그림 3-2] 실행 흐름

장한다. 이 저장정보는 분석 프레임워크인 AaaS 내 Change Tracker에게 감지되어 다시 Feature Extractor, Integrated Analyzer로 전송된 뒤 분석된다. 분석 데이터는 제어 프레임워크인 CaaS에서 IoT 디바이스 소프트웨어를 갱신하고 이벤트 트리거를 설정하기 위한 정보로 사용된다. CaaS는 갱신한 정보를 IoT 디바이스 노드에 적용시켜 IoT 디바이스를 제어한다.

4. 결론

본 논문에서 새 프레임워크와 디바이스를 추가할 수 있는 포그컴퓨팅 플랫폼을 제안하였다. 제안한 플랫폼에서는 IoT 디바이스 데이터에 대한 모니터링·분석·제어를 기본 동작으로 하며 관련 프레임워크를 위한 공통 API를 정의하였다.

5. Acknowledgement

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No.2017-0-00487, 포그 컴퓨팅을 위한 자가적응형 MACaaS 프레임워크 개발)과 2017년 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2016R1D1A1B01016073, 가상화 기반 오픈 게이트웨이 플랫폼과 오케스트레이션 보안 서비스 프레임워크 설계 및 구현)

참 고 문 헌

[1] Doukas, Charalampos., and Maglogiannis, Ilias. "Bringing IoT and cloud computing towards pervasive healthcare." Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. IEEE. 2012.

[2] Stojmenovic, Ivan. "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks." Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian. IEEE. 2014.