

<https://doi.org/10.7236/JIIBC.2022.22.1.83>

JIIBC 2022-1-13

엣지컴퓨팅을 활용한 분산처리 시스템의 가용성 향상에 관한 연구

A Study on the Improvement of Availability of Distributed Processing Systems Using Edge Computing

이건우*, 김영곤**

Kun-Woo Lee*, Young-Gon Kim**

요 약 최근 정보통신기술의 발전에 따라 사물인터넷(이하 IoT) 관련 기술이 지속적으로 발전하고 있다. IoT 시스템은 다양한 센서들을 바탕으로 센서마다 고유한 데이터를 네트워크를 통해 주고 받는다. IoT 시스템에서 발생하는 데이터는 실시간으로 발생한다는 특징과, 그 양이 설치된 센서의 양과 비례한다는 점에서 연속적으로 수집되는 데이터들은 빅 데이터로 정의할 수 있다. 현재까지의 IoT 시스템은 중앙 집중 처리 방식을 통한 데이터 저장, 처리 및 연산을 적용하였다. 하지만, 구축 규모가 커지고 다량의 센서를 사용하는 경우 기존의 중앙 집중 처리 방식의 서버는 병목 현상으로 인한 부하가 발생할 수 있다. 따라서, 본 논문에서는 IoT 환경에서 발생하는 실시간 센서 데이터를 효율적으로 처리하기 위하여 시스템의 고가용성을 목적으로 하는 데이터의 중요도 기반 알고리즘을 적용하기 위한 분산 처리 시스템에 대해 제안하였다.

Abstract Internet of Things (hereinafter referred to as IoT) related technologies are continuously developing in line with the recent development of information and communication technologies. IoT system sends and receives unique data through network based on various sensors. Data generated by IoT systems can be defined as big data in that they occur in real time, and that the amount is proportional to the amount of sensors installed. Until now, IoT systems have applied data storage, processing and computation through centralized processing methods. However, existing centralized processing servers can be under load due to bottlenecks if the deployment grows in size and a large amount of sensors are used. Therefore, in this paper, we propose a distributed processing system for applying a data importance-based algorithm aimed at the high availability of the system to efficiently handle real-time sensor data arising in IoT environments.

Key Words : Raspberry Pi3, Pi Camera, OpenCv

*정회원, 한국산업기술대학교 컴퓨터공학과

**정회원, 한국산업기술대학교 컴퓨터공학과

접수일자 2021년 7월 31일, 수정완료 2022년 1월 8일

게재확정일자 2022년 2월 4일

Received: 31 July, 2021 / Revised: 8 January, 2022 /

Accepted: 4 February, 2022

**Corresponding Author: ykim@kpu.ac.kr

Dept of Computer Engineering Korea Polytechnic University,
Korea

I. 서 론

최근 정보통신기술의 발전에 따라 사물인터넷(이하 IoT) 관련 기술이 지속적으로 발전하고 있다. IoT 시스템은 다양한 센서들을 바탕으로 센서마다 고유한 데이터를 네트워크를 통해 주고 받는다. IoT 시스템에서 발생하는 데이터는 실시간으로 발생한다는 특징과, 그 양이 설치된 센서의 양과 비례한다는 점에서 연속적으로 수집되는 데이터들은 빅 데이터로 정의할 수 있다. 현재까지의 IoT 시스템은 중앙 집중 처리 방식을 통한 데이터 저장, 처리 및 연산을 적용하였다. 하지만, 구축 규모가 커지고 다량의 센서를 사용하는 경우 기존의 중앙 집중 처리 방식의 서버는 병목 현상으로 인한 부하가 발생할 수 있다.

따라서, 본 논문에서는 IoT 환경에서 발생하는 실시간 센서 데이터를 효율적으로 처리하기 위하여 시스템의 고가용성을 목적으로 하는 데이터의 중요도 기반 알고리즘을 적용하기 위한 분산 처리 시스템에 대해 제안하였다. 시스템의 구성은 MQTT 프로토콜을 바탕으로 하며 MQTT 프로토콜의 Publiiser에 해당하는 SensorNode와 Broker 역할을 하는 Gateway, Subscriber 역할의 일괄 처리, 실시간 처리 서버로 구성하였으며, Apache Jmeter를 이용한 서버 부하 테스트를 통해 시스템에 대한 성능 평가를 진행하였다.

II. 관련 연구

1. Edge Computing

Edge Computing은 응답 시간을 개선하고 대역폭을 절약하기 위해 필요한 곳에 연산과 데이터 스토리지를 도입하는 분산 컴퓨팅 패러다임 중 하나이다. 기존의 IoT 시스템에서는 센서와 디바이스들의 처리 능력의 부재로 대부분의 데이터를 수집하여 클라우드 서버 및 중앙 데이터 센터로 전송/전달 하는 중앙 집중 방식을 사용하였다. 데이터 센터로 전송/전달 된 데이터들은 서버에 의해 처리 및 분석되며 필요 여부를 판단 받게 된다. 서버에서 결정이 내려지면 이에 대한 명령을 네트워크를 통해 디바이스에게 전달되어 명령이 수행되게 된다. 엣지 컴퓨팅의 개념도는 그림 1과 같다^[1].

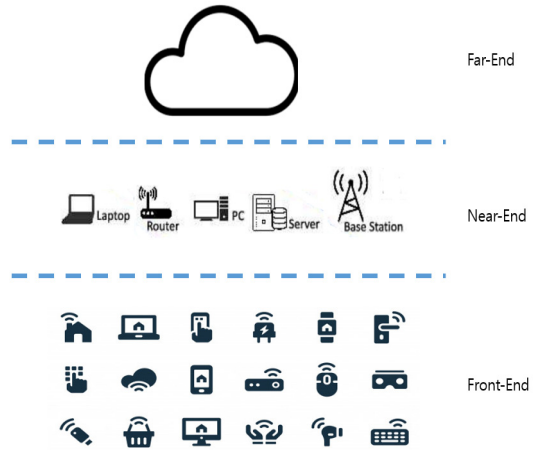


그림 1. 엣지 컴퓨팅 개념도
Fig. 1. Edge Computing Conceptual Diagram

2. MQTT 프로토콜

MQTT는 IoT와 같이 제한된 대역폭을 갖는 통신 환경에 적합한 메시지 전송 프로토콜로서 적은 코드로도 구현이 가능할 뿐 아니라 클라이언트/서버의 구조가 아닌 Publish(발행, 송신)/Subscribe(구독, 수신) 구조의 메시징 개념을 이용한 푸시 기술을 사용하고 있다. MQTT는 공간 분리성, 시간 분리성 및 동기화 분리성의 특징을 가지고 있다. 공간 분리성이란 메시지를 Publish하는 Publiiser나 메시지를 subscribe하는 Subscriber가 상대방의 IP 정보나 포트 번호 등을 모르더라도 Broker와 통신을 하면 된다는 개념이며, 시간 분리성이란 Publiiser와 Subscriber가 동시에 동작하지 않아도 데이터의 손실과 관련된 문제가 발생하지 않는다는 의미이다. 마지막으로 동기화 분리성은 송수신 작업이 핸드셰이킹 방식이 아니기 때문에 실시간적으로 동기화에 참여하지 않아도 된다. MQTT의 기본 개념도는 그림 2와 같다^[2].

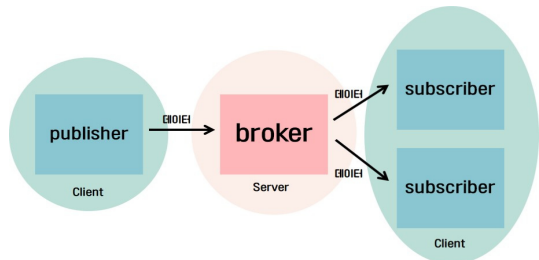


그림 2. MQTT 기본 개념도
Fig. 2. MQTT Basic Conceptual Plot

3. Raspberry Pi

라즈베리파이는 과학 교육의 증진을 목적으로 일반적인 컴퓨터의 기본적인 기능들을 갖춰 만든 싱글 보드 컴퓨터이지만, 최근에는 IoT기기로 취미와 업무에 널리 사용되고 있다. 2016년 출시한 Raspberry Pi3의 제원은 1.2GHz 쿼드코어의 CPU, 듀얼 코어 400MHz의 GPU, 1GB(DDR2) 메모리와 블루투스, 무선 랜 기능을 내장되어 있으며, Raspbian, Ubuntu MATE, Linux, Window 10 IoT Core 등 다양한 OS들을 지원한다.

또 Raspberry Pi는 일반적인 컴퓨터와 달리 하드웨어를 직접적으로 제어 가능한 다용도 입출력 포트(GPIO PIN; General Purpose Input Output)을 가지고 있으면서, 키보드와 모니터를 제외한 싱글 보드로 구성되어 있어 프로그래머의 의도대로 용도 변경 및 기능 확장이 가능하다는 것이 Raspberry Pi의 장점이다.

4. Ha Proxy

부하 분산을 위해 가상 IP를 통해 여러 서버에 접속하도록 분배하도록 하는 로드 밸런싱 기술의 일부로서 L4, L7 단계에서 로드 밸런싱 기능을 제공하는 소프트웨어 로드 밸런서이다. 서버로부터의 병목현상 제어 및 장애 발생시 다른 서버가 같은 일을 대응하여 처리해주는 방식으로서 공유기 분배의 역할과 유사한 개념이다.

로드 밸런싱의 개념도는 그림 3과 같다³⁾⁴⁾.

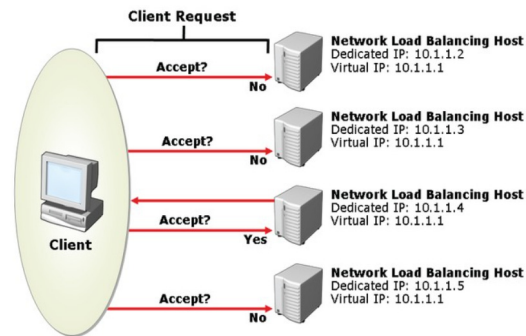


그림 3. 로드밸런싱 기본 개념도
 Fig. 3. MQTT Basic Conceptual Plot

III. 본 론

1. 시스템 구성도

비정형 데이터가 생성되는 센서 노드(IoT 환경)와 계

이트웨이 사이는 생성되는 데이터가 무분별하게 서버로 전송되는 것을 제한하기 위해 Publisher(Sensor), Broker(Gateway), Subscriber(Server) 구조의 MQTT 프로토콜을 적용하였다. Broker의 역할을 하는 게이트웨이는 가중치 기반의 우선순위 적용 알고리즘을 통해 수집 데이터에 대한 중요도를 설정하고, 데이터의 중요도에 따라 Real Time Processing Server와 Batch Processing Server의 특징을 고려하여 Task를 분배하도록 하였다. Real Time Processing Server는 Apache Storm을 적용하였으며, Batch Processing Server는 Apache Spark를 적용하였다. 제안 시스템의 개념도는 그림 4와 같다.

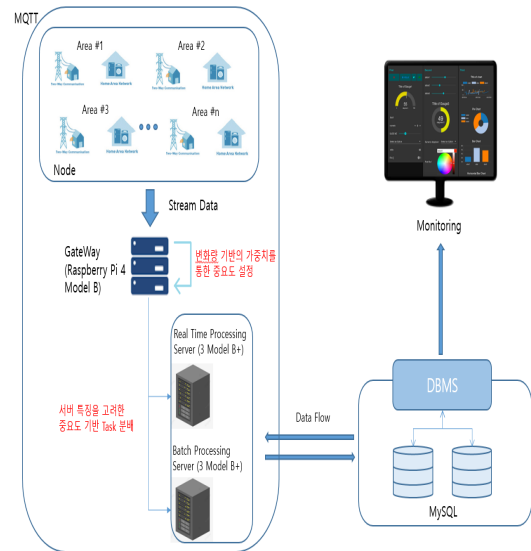


그림 4. 전체적인 관리 시스템 구성도
 Fig. 4. Overall Management System Diagram

2. 시스템 흐름도

센서 노드(Publisher)가 토픽을 바탕으로 전송하는 센서 데이터들을 게이트웨이(Broker)에서 받아들인다. 수집된 데이터에 데이터의 중요도를 나타내는 우선순위가 있는지 확인하고 우선순위가 존재하지 않는다면 중요도를 기반으로 한 우선순위 알고리즘을 적용시켜 데이터에 중요도를 부여한다. 본 논문에서 부여한 우선순위의 기준에 따라 데이터는 각각 일괄 처리 서버, 실시간 처리 서버로 전송되어 효율적인 처리 과정을 거쳐 DB로 저장하며 이후 빅 데이터 분석 툴인 R Language, Kibana 등을 통해 분석 및 시각화를 진행한다. 본 논문에서 제안한 시스템의 처리 흐름도는 그림 5와 같다.

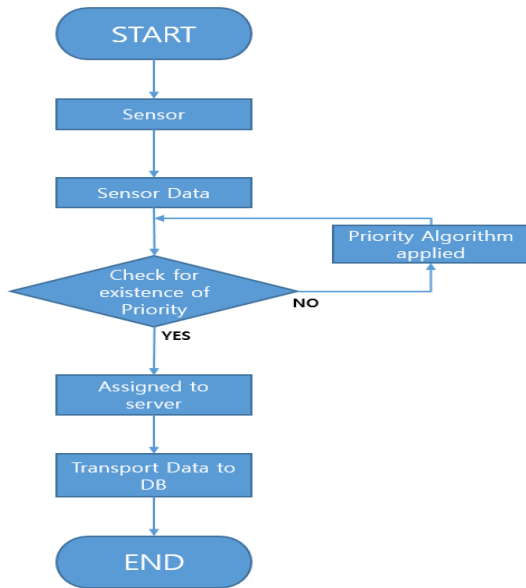


그림 5. 제안 시스템 흐름도
Fig. 5. Proposed System Flowchart

3. 가중치 기반의 우선순위 적용 알고리즘

수집 데이터는 센서에 따라 다양하게 적용 가능하며, 본 논문에서는 온도 데이터와 습도 데이터를 기준으로 작성하였다. 센서로부터 수집된 데이터를 저장 위한 각각의 변수에 해당하는 Prev.xData, Curr.xData를 선언하였으며, 해당 데이터의 중요도를 따지기 위한 Weight 변수, 비교하고자 하는 데이터의 변화량을 비율로 환산하기 위한 r 변수, 변화량의 수치를 담기 위한 fin 변수를 선언하였다. 변수 선언 이후 센서로부터 데이터를 읽어들이고 Curr.Data에 저장한다. 이후 이전 데이터를 저장하고 있는 Prev.Data 변수의 값을 확인하고, 만약 값이 존재하지 않는다면 Curr.Data의 값을 Prev.Data에 저장한다. 이후 새롭게 생성되는 데이터를 Curr.Data 변수에 저장하고 Prev.Data와 Curr.Data의 차이를 절대값으로 변환하여 Prev로 나누고 기존 대비 변화량을 비율로 환산한 값을 fin 변수에 저장한다. fin 변수와 가중치를 주기 위한 기준에 해당하는 r의 값을 비교하고 기준에 충족하는 경우 weight 값을 증가시키고 r값도 증가시켜 반복적으로 시행한다. 조건을 만족하지 못하는 경우에는 weight 값과 함께 수신된 데이터를 Server로 보내고 그 값을 DB에 저장한다. 알고리즘에 대한 순서도는 그림 6과 같다.

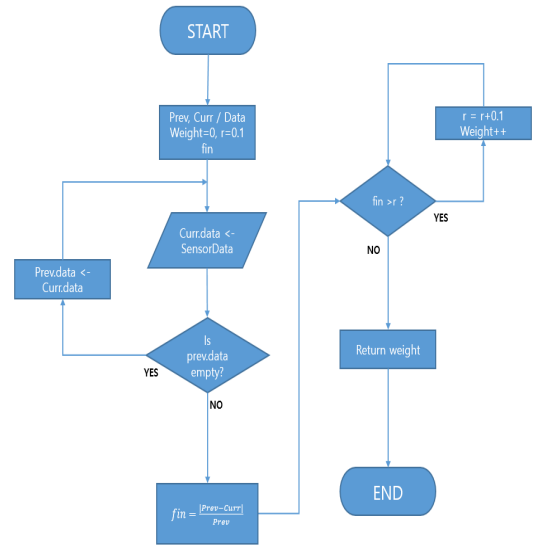


그림 6. 가중치 기반의 우선순위 적용 알고리즘
Fig. 6. weight-based priority application algorithm

IV. 시뮬레이션

수집된 데이터를 Broker 역할을 하는 Gateway에서 1차적으로 중요도에 따른 우선순위를 선정하는 데이터 정제 과정을 거쳐, 중요도에 따라 일괄 처리 및 실시간 처리 서버로 데이터를 분산 전송 및 처리하여 중앙 집중 처리 방식의 서버의 부하와 관련된 단점을 보완하기 위한 시스템을 제안하였다. 중요도 알고리즘의 적합성을 판단하기 위해 이상 데이터를 온풍기를 사용하여 임의로 발생시킨 것은 그림 7과 같다.



그림 7. 임의의 이상 데이터 발생
Fig. 7. Random Outlier Data Occurrence

센서 노드를 통해 15초 마다 데이터를 수집하였다. 1번 센서 노드에서 처음으로 수집되어 저장된 데이터는 온도 22℃, 습도는 61%였으며, 기존에 존재하는 데이터가 없으므로, Priority는 0이 저장되었다. 13시 38분 30초까지는 온풍기를 가동하지 않았으며, 4번째 데이터가 수집되는 시점에서 온풍기를 일정거리를 유지한 상태로 가동시켜 데이터를 발생시켰다.

우선순위의 값이 변동된 8~9번 데이터의 차이를 보았을 때 기존 데이터에 해당하는 8번 레코드의 온도, 습도 값은 각각 25℃, 56%이며, 현재 데이터에 해당하는 9번 레코드의 온도, 습도 값은 28℃, 48%로 각각 12% 상승, 14% 하락함을 확인할 수 있었다. 이에 따라 우선순위를 의미하는 Priority 값이 증가하여 저장되었으며, 제안한 알고리즘이 정상적으로 적용되어 나타남을 확인할 수 있었다. 센서를 통해 수집된 데이터를 우선순위 알고리즘을 적용하여 DB에 저장한 결과 화면은 그림 8과 같다.

기존 알고리즘의 초당 트랜잭션 수와 제안 알고리즘의 초당 트랜잭션 수는 그림 4-14와 같다. 본 연구에서 제안한 변화량 기반의 가중치를 통한 중요도 알고리즘은 최고 초당 2700개의 트랜잭션을 처리할 수 있었던 반면, 기존 방식은 최고 초당 2300개로 최고 성능의 경우 약 17% 정도의 차이를 보였으며, 평균 트랜잭션 수는 기존 방식의 경우 1800 transactions/sec 이었으며, 제안 알고리즘의 경우 2050 transactions/sec으로 평균 트랜잭션 수는 약 13% 정도 향상되었음을 확인할 수 있었다. 적용 알고리즘 별 초당 트랜잭션 수는 그림 9와 같다.

ins_date	num	Identity	Temperature	Humidity	Dust	Vibration	Priority
2021-04-20 13:38:00	1	1	22	61	NULL	NULL	0
2021-04-20 13:38:15	2	1	22	61	NULL	NULL	0
2021-04-20 13:38:30	3	1	22	61	NULL	NULL	0
2021-04-20 13:38:45	4	1	24	59	NULL	NULL	0
2021-04-20 13:39:00	5	1	25	54	NULL	NULL	0
2021-04-20 13:39:15	6	1	25	55	NULL	NULL	0
2021-04-20 13:39:30	7	1	25	55	NULL	NULL	0
2021-04-20 13:39:45	8	1	25	56	NULL	NULL	0
2021-04-20 13:40:00	9	1	28	48	NULL	NULL	4
2021-04-20 13:40:15	10	1	31	42	NULL	NULL	4
2021-04-20 13:40:30	11	1	31	42	NULL	NULL	4
2021-04-20 13:40:45	12	1	31	42	NULL	NULL	4
2021-04-20 13:41:00	13	1	31	42	NULL	NULL	4
2021-04-20 13:41:15	14	1	37	29	NULL	NULL	8
2021-04-20 13:41:30	15	1	37	29	NULL	NULL	8
2021-04-20 13:41:45	16	1	37	29	NULL	NULL	8
2021-04-20 13:42:00	17	1	37	29	NULL	NULL	8
2021-04-20 13:42:15	18	1	37	29	NULL	NULL	8
2021-04-20 13:42:30	19	1	43	19	NULL	NULL	12
2021-04-20 13:42:45	20	1	43	19	NULL	NULL	12
2021-04-20 13:43:00	21	1	43	19	NULL	NULL	12
2021-04-20 13:43:15	22	1	46	16	NULL	NULL	13
2021-04-20 13:43:30	23	1	47	17	NULL	NULL	13
2021-04-20 13:43:45	24	1	47	18	NULL	NULL	13
2021-04-20 13:44:00	25	1	49	19	NULL	NULL	13
2021-04-20 13:44:15	26	1	49	19	NULL	NULL	13
2021-04-20 13:44:30	27	1	49	19	NULL	NULL	13
2021-04-20 13:44:45	28	1	49	19	NULL	NULL	13
2021-04-20 13:45:00	29	1	49	19	NULL	NULL	13
2021-04-20 13:45:15	30	1	49	19	NULL	NULL	13
2021-04-20 14:45:30	31	1	46	16	NULL	NULL	14
2021-04-20 14:45:45	32	1	46	16	NULL	NULL	14
2021-04-20 14:46:00	33	1	45	16	NULL	NULL	14
2021-04-20 14:46:15	34	1	45	16	NULL	NULL	14
2021-04-20 14:46:30	35	1	44	16	NULL	NULL	14

그림 8. 알고리즘이 적용된 데이터 저장 결과 화면
 Fig. 8. Save data with algorithm screen

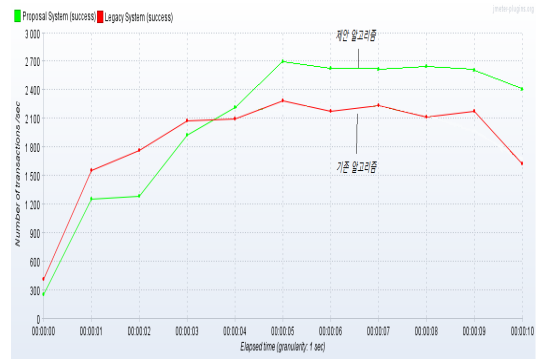


그림 9. 알고리즘이 적용된 데이터 저장 결과 화면
 Fig. 9. Overall Management System Diagram

V. 결 론

본 논문은 IoT 환경에서 발생하는 대량의 데이터를 서버의 부하를 최소화하고, 가용성을 유지하기 위하여 중요도 기반의 알고리즘을 적용한 분산처리 시스템을 제안하였다. 기존의 시스템은 서버에 데이터를 전달함에 있어서, 중앙 집중 처리 방식과, 분산처리의 두 가지 방식이 있었다. 그러나 중앙 집중 처리 방식은 대량의 데이터가 발생하였을 때, 시스템의 가용성이 유지되지 못하는 단점이 있었으며, 기존의 분산처리 방식 또한 시스템의 구조적인 복잡도와 관련된 단점이 있었다. 이러한 문제를 해결하기 위하여 중요도 알고리즘과 분산처리 시스템 구조를 제안하였다. 중요도 알고리즘은 데이터의 변화량의 크기를 바탕으로 가중치를 산출하여 중요도를 선정하였으며, 선정된 중요도를 기준으로 하여 각각의 서버에서 데이터를 받아 처리하는 형태로 구성하였다. 각 서버는 Apache Storm, Apache Spark를 활용하여 Real Time Processing Server와 Batch Processing Server로 구분하고, 이러한 환경에서 시뮬레이션을 진행하였고, 평균 15%, 최소 13%, 최대 17%의 향상된 트랜잭션 수를 확인할 수 있었다.

본 연구는 서버의 고가용성을 유지하며 대량의 데이터를 처리하기 위한 시스템의 구조와 알고리즘에 대하여 진행하였으나, 사용자의 요청에 따른 데이터의 효율적 처리 및 가공 방법에 대한 연구는 아쉬움으로 남는다. 추후 저장된 데이터를 R, Kibana 등 시각화 툴을 사용하여 말단 노드를 제어하기 위한 연구를 진행할 예정이다.

References

- [1] S. H. Kim, Y. G. Kim, "A Study on Light Weight Authentication Method of Distributed Cluster-based IoT Devices", The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 19, No. 2, pp. 103-109, 2019.
DOI: <https://doi.org/10.7236/JIIBC.2019.19.2.103>
- [2] D. J. Shin, J. H. Park, J. H. Kim, K. J. Kwak, J. M. Park, J. J. Kim, "Big Data-based Sensor Data Processing and Analysis for IoT Environment", The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 19, No. 1, pp. 117-126, 2019.
DOI: <https://doi.org/10.7236/JIIBC.2019.19.1.117>
- [3] S. H. Lee, D. J. Shin, J. H. Kim, D. Y. Shin, J. M. Park, J. J. Kim, "Mobile-based Big Data Processing and Monitoring Technology in IoT Environment". The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 18, No. 6, pp. 1-9, 2018.
DOI: <https://doi.org/10.7236/JIIBC.2018.18.6.1>
- [4] Y. G. Hong, "Trend of edge computing technology for industrial IoT", The Proceedings of the Korea Electromagnetic Engineering Society, Vol. 30, No. 3, pp. 41-50, 2019.
- [5] D. H. Kong, S. J. Shin, "Analysis of Arduino Timer Callback for IoT Devices", The journal of the institute of internet, broadcasting and communication, Vol. 18, No. 6, pp. 139-143, 2018.
DOI: <https://doi.org/10.7236/JIIBC.2018.18.6.139>
- [6] T. H. Jang, S. H., Jo, Y. G. Kim, "Smart queue management system based on IoT for efficient customer management", The journal of the institute of internet, broadcasting and communication, Vol. 18, No. 5, pp. 141-146, 2018.
DOI : <https://doi.org/10.7236/JIIBC.2018.18.5.141>

저 자 소 개

이 건 우(정회원)



- Kun woo Lee received his BS in low at Korea Polytechnic University in 2018. He is currently Master in degree department Computer engineering at Korea Polytechnic University.

김 영 곤(정회원)



- Young Gon Kim received his BS in Electronic Engineering at Kyungpook University in 1983 and MS in Electronic Engineering at Yonsei University in 1985, respectively. In 2000, he received his PhD in at KAIST. He is currently a professor at the department of Computer Science at Korea Polytechnic University. His research interests include Software Engineering, nformation communication system, object-oriented analysis and design, etc.