Architectural design of IoT-cloud computing integration platform

Adhitya Bhawiyuga[*1], Dany Primanita Kartikasari [2], Kasyful Amron [3], Ocki Bagus Pratama [4], Moch. Wildan Habibi [5]
Faculty of Computer Science, Brawijaya University, Kota Malang, Indonesia
[*]Corresponding author, e-mail: bhawiyuga@ub.ac.id[1], dany.jalin@ub.ac.id[2], kasyful@ub.ac.id[3],

Abstract

An integration between the Internet of Things (IoT) and cloud computing can potentially leverage the utilization of both sides. As the IoT based system is mostly composed by the interconnection of pervasive and constrained devices, it can take a benefit of virtually unlimited resources of cloud entity i.e storage and computation services to store and process its sensed data. On the other hand, the cloud computing system may get benefit from IoT by broadening its reach to real world environment applications. In order to incarnate this idea, a cloud software platform is needed to provide an integration layer between the IoT and cloud computing taking into account the heterogenity of network communication protocols as well as the security and data management issues. In this study, an architectural design of IoT-cloud platform for IoT and cloud computing integration is presented. The proposed software platform can be decomposed into five main components namely cloud-to-device interface, authentication, data management, and cloud-to-user interface component. In general, the cloud-to-device interface acts as a data transmission endpoint between the whole cloud platform system and its IoT devices counterpart. Before a session of data transmission established, the communication interface contact the authentication component to make sure that the corresponding IoT device is legitimate before it allowed for sending the sensor data to cloud environment. Notice that a valid IoT device can be registered to the cloud system through web console component. The received sensor data are then collected in data storage component. Any stored data can be further analyzed by data processing component. User or any developed applications can then retrieve collected data, either raw or processed data, through API data access and web console.

Keywords: *cloud computing, internet of things, software platform*

1. Introduction

The development of industrial revolution 4.0 have led to an emerging research issue in the area of Internet of Things (IoT). In IoT, the pervasive and ubiquitous devices equipped with microcontroller, sensors and actuators are expected to be interconnected through a transceiver module using various kind of communication protocols. Those kind of interconnection leads to the ability of an IoT based system to be uniquely identified, performing perception of surrounding environment and exchanging those perceived data through various kind of communication medium.As a result, any promising smart applications can be developed on top of that ecosystem such as:precision agriculture, building health monitoring and smart grid [1].

In general, IoT building block compose of six different parts. There are consist of identification, sensing, communication, computation, services and semantics part [2]. The computationpart located on microcontroller attached to IoT device, has the main role toprocess data acquired from sensing part. It has basic ability to process simple computationtaks such asanalog to digital conversion locally. However as a system running in a longer period, the datacollected might ex- ceeds the amount storage of local IoT devices to save and process it asrequired to be analyzed. Therefore, more reliable storage capacity and powerfulcomputationallyentity is highly demanded. In contrast with IoT devices,a cloud computing entityoffers arelatively unlimited capabilities to store and process huge data using virtualizationtechnologyapplied on it. [3] Not onlyCloud computing reliability of processing computation [4] butalso itmaximizing the resource as mention in [5].

Received June 7, 2018; Revised January 28, 2019; Accepted February 28, 2019

The integration of IoT devices and cloud computing environment has a chancetoescalate the leverage of system utilization on both sides. On the other hand, IoT system maygetbenefit from taking storage and computation resources provided by the cloud entity.Furthermore,cloud computing has more stable nature compared to IoT devices in the IoT system such that to ensure the availability of IoT sensor data that has been collected. The formation of IoT-Cloud areas receive serious attention within the field of IoT development as mention in [6]. In 2010, [7] has proposed an event-driven sensor virtualization approach for Internet of Things cloud which demonstrate the development of IoT application with reasoning capability using a green school motorcycle case study. In 2014, [8] has published a utility paradigm for IoT known as The sensingcloud.

Despite of its benefit, there exists several challenges arise from integrating the IoT to cloud computing including network communication, security and data management. [9] The first issue is network communication which originates from the diversity and variability of available networking protocols. To name a few, there are TCP based protocols such as HTTP, MQTT and AMQP while on the other hand there exists UDP based protocol such as CoAP [10] and [11]. The second issue is related to the ability of cloud system to confirm that its IoT device counterpart is a valid partner [12].

Finally, after the validity of IoT device is confirmed and the data from sensor is received, there must be a mechanism to effectively store the data so that it can be processed and accessed in the future. To deal with issues mention above, an integration layer between the IoT and cloud computing is required. Thus in this study propose an architectural design of IoT-cloud software platform for IoT and cloud computing integration. The proposed platform is composed by five main components namely cloud-to-device interface, authentication, data management, and cloud to user interface component. In general, the cloud to device interface acts as a data transmission endpoint between the whole cloud platform system and its IoT devices counterpart.

Before a session of data transmission is established, the communication interface contacts authentication component to make sure that the corresponding IoT device is legitimate be- fore it is granted to send the sensor data to cloud. Notice that a valid IoT device can be registeredto the cloud system through web console component. The received sensor data are then collected in data storage component. Any stored data can be further analyzed by data processing component. User or any developed applications can then retrieve collected data, either raw orprocessed data, through API data access and web console.

2. Preliminary Study on IoT-Cloud Platform

The IoT combined with Cloud computing functionalities leads to a new concept known as cloud of things [13, 14]. The term IoT Cloud was coined in [15] and [16] to explain the integration of IoT and cloud computing. Cloud computing and IoT basically developed in different way. From Botta's work, some complementary aspect of cloud and IoT is illustrated in Table 1. IoT can takea benefit of storage and computation resources provided by cloud entity. Furthermore, a more stable nature of cloud computing system compared to its IoT devices counterpart may give more assurance on the availability of sensor data. On the other hand, the cloud computing system may get benefit from IoT by broadening its reach to real world environment. Therefore, more pervasive services can be developed on top if this integration. Furthermore, the cloud can possibly take a role as middleware to bridge between the user or application to the IoT devices.

Table 1. Complementary aspects of IoT and cloud computing

Aspects	IoT	Cloud
Displacement	Pervasive	Centralized
Reachability	Limited	Ubiquitous
Components	Real world things	Virtual resources
Computational Capabilities	Limited	Virtually unlimited
Storage	Limited or none	Virtually unlimited
Role of the Internet	Point of convergence	Means for delivering services
Big data	Source	Means to manage

The cloud should then meet the requirement of IoT applications. It has to be able to serve different type of IoT application. All subject related to IoT application have to be offered the services provided by the IoT cloud. An IoT-Cloud system typically has three components. The first component is natural environment where the sensors, actuators and connecting devices located. Basically sensors as electronic devices that interact with the physical world, generate data from their surrounding environment. The next component is cloud computing system as part for storing, processing and analyzing generated data from the environment. IoT middleware collects data from sensors, then transfer it to Cloud environment. Virtual machines running on the physical server in the cloud will handle the data to be store in a storage system and do computation function. And the last component of proposed system is user applications. In user applications, user can do some actions such as visualizing result data from the sensor and controlling the environment remotely.

3. Research Method

The research methods conducted with several steps. For the first step is do literature study on the area of IoT, cloud computing and the issues related to the integration of both entities presented in Section 2. Next step is make a design of the system architecture of proposed IoT cloud platform which presented in Section 4. Based on the design, the next step is to implement each component composing the system in Section 5. After finish the implementation step, the system design is tested in term of its functionality and performance. Testing result is presented in Section 6. Finally, the last part is conclusion of the work mention in the conclusion in Section 7.

4. System Design

Figure 1 illustrates a general environment of the IoT based system using three actors namely IoT sensing device, IoT gateway device and IoT cloud platform. Notice that, to simplify the terminology IoT device is used to state both IoT sensing and gateway devices.

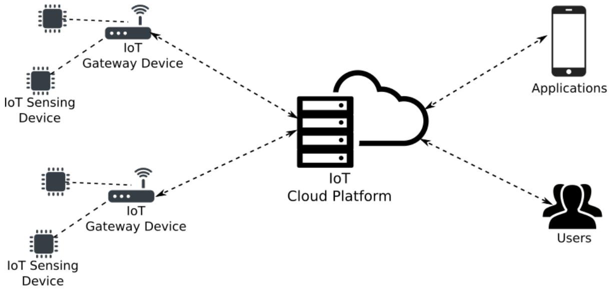


Figure 1. General environment of IoT based system

Not only IoT sensing device has a role to perceive contemporary circumstances of surrounding environment through involvement of sensors, but also has function to give a specific response to the environment by adding actuator part in IoT device. The collected sensor data will be transmitted to the IoT gateway through a wireless connection such as Wifi [17], Bluetooth Low Energy [18], zigbee [19] or low power wide area network (LPWAN) [20]. Once the data is received, IoT gateway device relays the data to IoT cloud platform through a cellular or backbone network connection. In this sense, the IoT gateway should have capability to communicate with both local IoT sensing device and global IoT cloud platform. Finally, the sensor data are received by IoT cloud platform and stored to its database systemfor a more complex data processing as well asfurther data access by users or developed applications. On reverse direction, the IoT cloud may receive a specific command from user through a predetermined application programming interface (API) and send it to IoT device through the assistance of IoT gateway.

In this study is focused on the design and implementation of IoT cloud platform part spesifically. The building block of proposed IoT cloud platform can be illustrated in Figure 2 There exists five main components in the system. They are cloud-to-device interface, authentication, data management, and cloud to-user interface component. In general, the cloud-to-device interface acts as a data transmission end- point between the whole cloud platform system and its IoT gateway devices counterpart. Before a session of data transmission established, the communication interface contacts the authentication component to ensure that the corresponding IoT gateway is legitimate before being granted to send acquired sensor data to cloud environment. Notice that a valid IoT gateway can be registered to the cloud system through web console component. The received sensor data that has arrived in cloud system is then collected in data storage component. Any stored data can be further analyzed by data processing component. User can retrieve the collected data through API data access and web console.

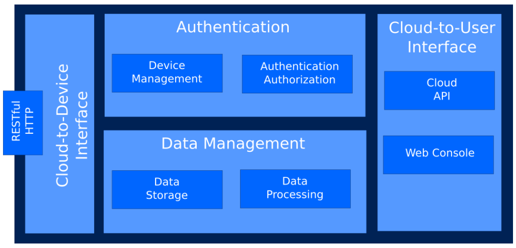


Figure 2. Building block of proposed IoT cloud platform

5. Component Implementation

In this section, the implementation of each component in proposed IoT cloud platform is explained.

5.1. Cloud-to-Device Interface

As stated in section 4., the cloud-to-device interface component has a vital role for providing a communication endpoint between an IoT device and the whole cloud platform system. In order to provide such functionality a messaging protocol named Restful HTTP [21] is utilized. This protocol is chosen due to their wide adoption in the area of machine-to-machine (M2M) communication.

5.1.1. Restful HTTP Interface

Figure 3 illustrates the design of RESTful HTTP in our proposed IoT cloud platform. RESTful HTTP interface is designed using three main actors. Namely IoT gateway/device acting as a HTTP client, IoT Gateway's corresponding HTTP server located in cloud platform and the database. First, the IoT devices send the acquired sensor data together with its authentication tokenin form of HTTP request. Upon reception, the HTTP server authenticates the device identity by comparing token credential with one in authentication database. Once it is clarified, the HTTP server then stores received sensor data to the database as storage component and send back a response indicating that the operation is successfully performed. In detail, the request and response format is presented in Table 2 while the data format of request message is presented in section 5.1.2.

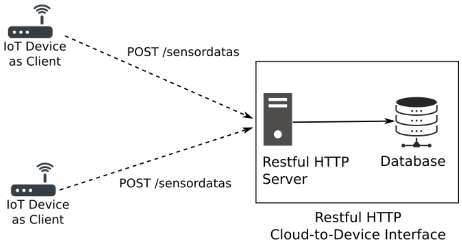


Figure 3. Design of RESTful HTTP interface

Table 2. Request and response format for sensor data transmission.

Request :
POST /sensordatas/ HTTP/1.1 Host: Cloud IP:Port Content-Type: application/json Authorization: JWT *token* *Sensor data in JSON format. Detail in section 5.1.2.*
Response
HTTP/1.1 201 Created {”results”:”Sensordatas has successfully added.”}

5.1.2. Data Format

A compact data format is required to represent various kind of sensor data coming from different IoT sensing devices. Therefore, this study is using a key value data in the form of Javascript Object Notation (JSON). The JSON offers a relatively lightweight data-interchange format which is easily readable both by human and machine [22]. The complete format used by IoT devices to send its sensor data has two main key-value. The first key value is ”label” key indicating the identifier of an IoT device registered in cloud platform. The second key value is ”sensors” key indicating all available sensor data. Inside the ”sensors” part there exists another key-value pairs, namely ”value” key indicating the list of sensor value with its corresponding timestamp and a sen- sor label to classify those list. By using this data format, an IoT gateway device can perform a data aggregation of sensor data coming from different attached IoT sensing devices in a certain time window.A compact data format is required to represent various kind of sensor data coming from different IoT sensing devices. Therefore, in this study, a key-value data in the form of Javascript Object Notation (JSON) is used. The JSON offers a relatively lightweight data-interchange format which is easily readable both by human and machine.

5.2. Authentication

The authentication component plays an important roles to ensure the validity of IoT de vices connected to the cloud platform. In order to do that, the cloud platform software should have the ability to perform an authentication process. However, instead of putting the already sensitive username-password credential on the device, the proposed system is using the token based in form of JSON Web Token (JWT). The JWT is chosen since it has an expiration mechanism which is safer from a wireless tampering and sniffing than that of the username-password. Figure 4 illustrates the flow of authentication for both RESTful HTTP. Since the RESTful HTTP is a stateless protocol, therefore, for each sensor data transmission, it should always contains the token credential located in the HTTP header part.

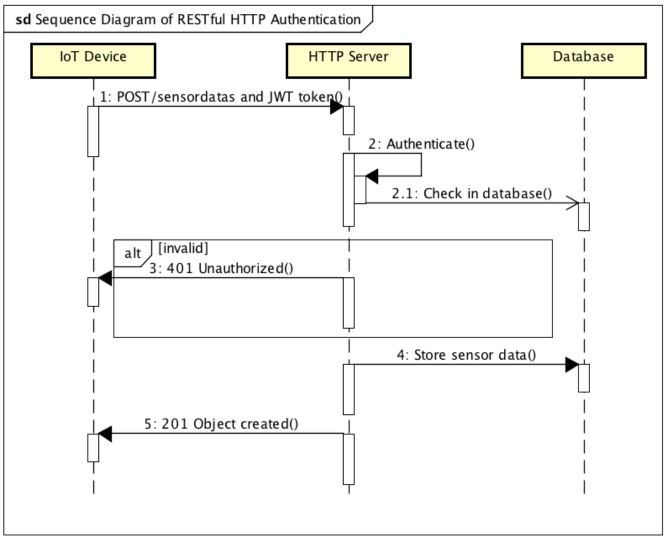


Figure 4. Authentication mechanism

5.3. Data Management

The sensor data received by cloud platform should then be stored on a database management system (DBMS) for further processing and access. There exists several characteristics of IoT sensor data that should be considered in selecting the right DBMS. First, the variety of data can be expanded as additional sensor attached on IoT devices. Furthermore, a sensor data can be transmitted in the form of streaming ranging from daily period to a near real time fashion which may leads to a huge volume of sensor data storage. Therefore, DBMS with a less strict schema rules combined with fast writing performance is suitable for IoT data storage purposes. Taking into account aforementioned requirement, in this study, a NoSQL DBMS called MongoDB is utilized. In contrast with SQL based DBMS, the NoSQL offers a schema-less feature in which doesn't require to specify a predefined structure of the table. It means, the table property can be elastically defined during the insertion of new data. Furthermore, MongoDB can be horizontally scaled thanks to its sharding and replication capabilities. For using MongoDB, the system need to define a big picture of data storage in term of document. Figure 5 illustrates the design of proposed data storage using MongoDB.

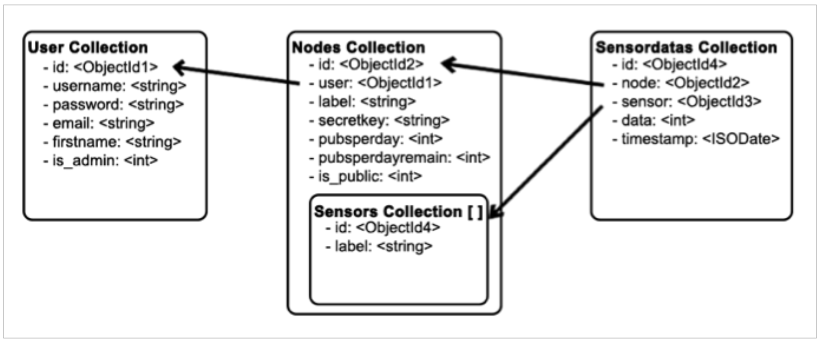


Figure 5. Design of proposed data storage using mongoDB document

In the design, MongoDB is consists with four documents: Users, Nodes, Sensors and Sensor data. The User Document is utilized to store user information which is used mainly for authentication purposes. A user may have several IoT device nodes under his/her supervision while a node can possible have more than one sensor. At last, there exists Sensordata's document to store the received sensor data. It has relation with document nodes and sensors to give status about the data originated information.

5.4. Cloud-to-User Interface

A cloud-to-user interface is developed to provide an access to stored sensor data for both user and other application using RESTful webservice based application programming interface. The specification of API for request format are presented in Table **??**. In addition, a web console to provide a graphical user interface (GUI) for both managing allowed IoT devices and accessing the sensor data is developed.

Table 3. API Spesification request format for sensor data access

Request	Description
POST /auth/	Authenticate the user using JWT token
GET /sensordatas/	Get all sensor data from all nodes
GET /sensordatas/node/*node-id*	Get all sensor data a node with specific *node-id*
GET /sensordatas/node/*node-id*/ sensor/*sensor-id*	Get all sensor data a node with specific *node-id* and *sensor-id*

6. Result and Analysis

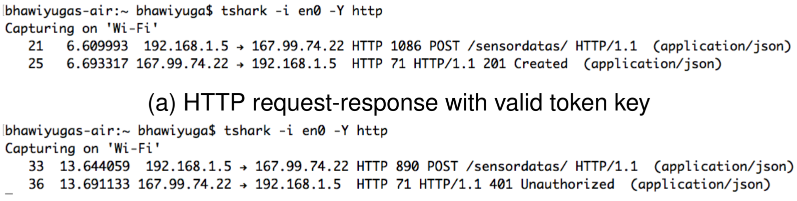
In this section is presenting the result and analysis of the testing performed to the proposed IoT cloud platform in term of functional and performance testing. All components are deployed on a virtual private server (VPS) with specifications 1.6 GHz Single Core CPU with 1 GB RAM, 30 GB SSD drive and public IP address running Ubuntu 14.04 as server operating system. We use Apache version 2.4.7 [23] as webserver and Django Framework [24] for developing RESTful HTTP service as well as JWT authenticationserver. For helping our work in database management system, MongoDB [25] version 3.4.14 is chosen.

6.1. Functional Testing

To perform the functional testing, a set of hardware prototype is developed to run IoT sensing and gateway device function. IoT device prototype is built using Arduino Nano with AT-Mega 328 microprocessor. It also equipped with a 433 MHz RFM95 LoRa communication module. The IoT device prototye is connected to sensor devices to measure soil moisture, air temperature-humidity and rain level. The IoT devices gather data from surrounding and transmit the acquired data to IoT gateway. IoT gateway prototype is built using Raspberry Pi B series with 1.2 GHz ARM Processor, 1 GB RAM and 8 GB MicroSD card. As communication interface, IoT gateway prototype is engaged with 433 MHz RFM95 LoRa communication module and IEEE 802.11 b/g/n module. Once the IoT gateway device receives data from its connected sensing devices, IoT gateway device will send the data to cloud platform by using RESTful HTTP protocol. First, We tested the authentication mechanism by sending a sensor data request to cloud system using both valid and invalid token. Figure 6a and 6b show the HTTP request and response with valid and invalid token, respectively. From the result, we observe that the authentication component successfully perform a validation by giving different responses to the both valid and invalid tokens. The sensor data is then stored by management component on a MongoDB instance. Figure 7 and 8 show the stored sensor data and its visualization in web console, respectively.

6.2. Performance Testing

This study performed a test to measure the performance of proposed IoT cloud platform in term of: response time of sensor data reception and throughput of RESTful HTTP server.



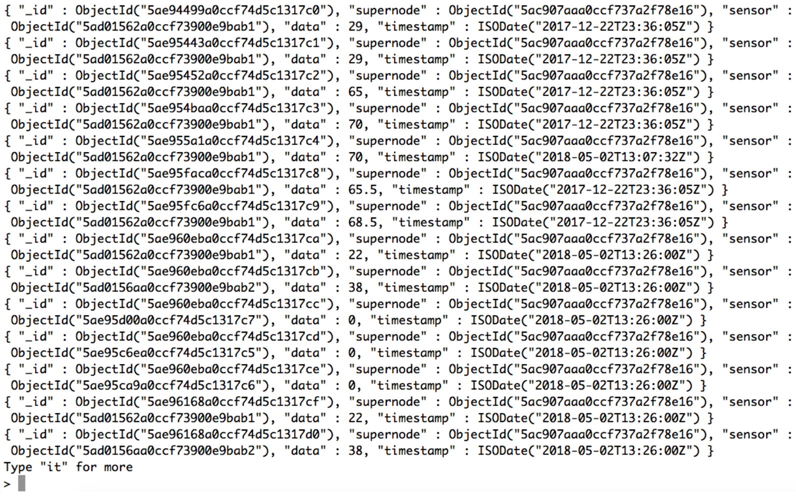
```
bhawiyugas-air:~ bhawiyuga$ tshark -i en0 -Y http
Capturing on 'Wi-Fi'
   21   6.609993  192.168.1.5 → 167.99.74.22 HTTP 1086 POST /sensordatas/ HTTP/1.1  (application/json)
   25   6.693317 167.99.74.22 → 192.168.1.5  HTTP 71 HTTP/1.1 201 Created  (application/json)
```

(a) HTTP request-response with valid token key

```
bhawiyugas-air:~ bhawiyuga$ tshark -i en0 -Y http
Capturing on 'Wi-Fi'
   33  13.644059  192.168.1.5 → 167.99.74.22 HTTP 890 POST /sensordatas/ HTTP/1.1  (application/json)
   36  13.691133 167.99.74.22 → 192.168.1.5  HTTP 71 HTTP/1.1 401 Unauthorized  (application/json)
```

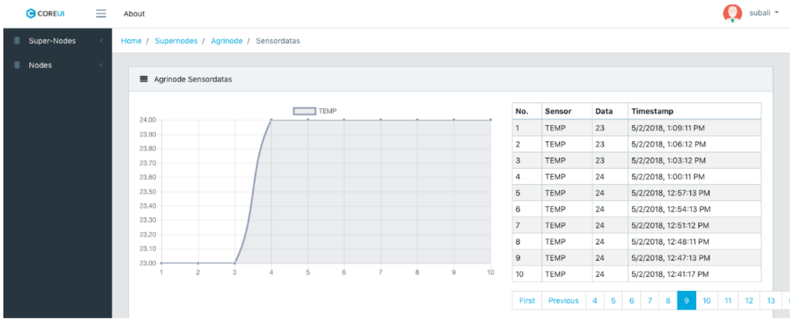
(b) HTTP request-response with valid token key

Figure 6. Authentication mechanism testing using valid and invalid token key



```
{ "_id" : ObjectId("5ae94499a0ccf74d5c1317c0"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 29, "timestamp" : ISODate("2017-12-22T23:36:05Z") }
{ "_id" : ObjectId("5ae95443a0ccf74d5c1317c1"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 29, "timestamp" : ISODate("2017-12-22T23:36:05Z") }
{ "_id" : ObjectId("5ae95452a0ccf74d5c1317c2"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 65, "timestamp" : ISODate("2017-12-22T23:36:05Z") }
{ "_id" : ObjectId("5ae954baa0ccf74d5c1317c3"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 70, "timestamp" : ISODate("2017-12-22T23:36:05Z") }
{ "_id" : ObjectId("5ae955a1a0ccf74d5c1317c4"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 70, "timestamp" : ISODate("2018-05-02T13:07:32Z") }
{ "_id" : ObjectId("5ae95faca0ccf74d5c1317c8"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 65.5, "timestamp" : ISODate("2017-12-22T23:36:05Z") }
{ "_id" : ObjectId("5ae95fc6a0ccf74d5c1317c9"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 68.5, "timestamp" : ISODate("2017-12-22T23:36:05Z") }
{ "_id" : ObjectId("5ae960eba0ccf74d5c1317ca"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 22, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
{ "_id" : ObjectId("5ae960eba0ccf74d5c1317cb"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad0156aa0ccf73900e9bab2"), "data" : 38, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
{ "_id" : ObjectId("5ae960eba0ccf74d5c1317cc"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ae95d00a0ccf74d5c1317c7"), "data" : 0, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
{ "_id" : ObjectId("5ae960eba0ccf74d5c1317cd"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ae95c6ea0ccf74d5c1317c5"), "data" : 0, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
{ "_id" : ObjectId("5ae960eba0ccf74d5c1317ce"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ae95ca9a0ccf74d5c1317c6"), "data" : 0, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
{ "_id" : ObjectId("5ae96168a0ccf74d5c1317cf"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad01562a0ccf73900e9bab1"), "data" : 22, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
{ "_id" : ObjectId("5ae96168a0ccf74d5c1317d0"), "supernode" : ObjectId("5ac907aaa0ccf737a2f78e16"), "sensor" :
 ObjectId("5ad0156aa0ccf73900e9bab2"), "data" : 38, "timestamp" : ISODate("2018-05-02T13:26:00Z") }
Type "it" for more
>
```

Figure 7. Sensor data stored in mongoDB.



No.	Sensor	Data	Timestamp
1	TEMP	23	5/2/2018, 1:09:11 PM
2	TEMP	23	5/2/2018, 1:06:12 PM
3	TEMP	23	5/2/2018, 1:03:12 PM
4	TEMP	24	5/2/2018, 1:00:11 PM
5	TEMP	24	5/2/2018, 12:57:13 PM
6	TEMP	24	5/2/2018, 12:54:13 PM
7	TEMP	24	5/2/2018, 12:51:12 PM
8	TEMP	24	5/2/2018, 12:48:11 PM
9	TEMP	24	5/2/2018, 12:47:13 PM
10	TEMP	24	5/2/2018, 12:41:17 PM

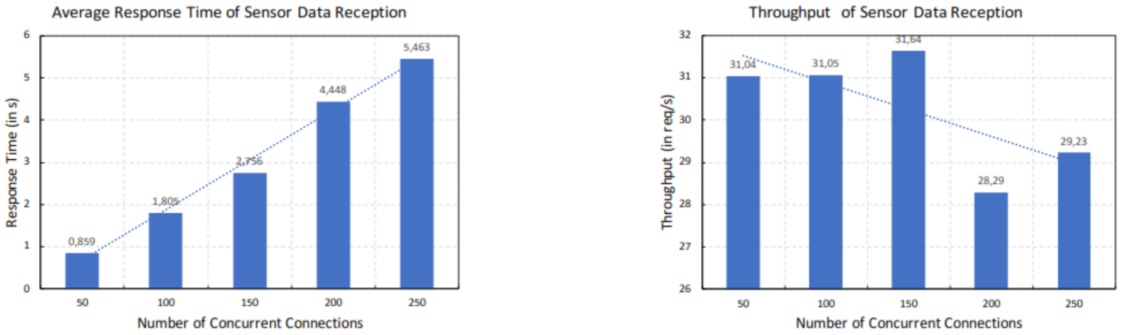
Figure 8. Data Visualization in Web Console.

6.2.1. Performance of RESTful HTTP Interface

For measuring the performance of RESTful HTTP interface, this study utilizes JMeter tool to simulate concurrent sensor data request from 50, 100, 150 IoT devices. Figure 9 (a) and (b) shows the result of measurement in term of response time (in seconds) and the throughput of HTTP server (in requests per second), respectively. From the result can be observe that there is a conflicting trend between the response time and the throughput of sensor data reception. As the number of device increases, the response time tends to increase while the throughput decrease. This can be happened since the HTTP server should handle and maintain more HTTP request as the number of devices increases. However, from the result has shown that at worst the average response is about 3 seconds while the server can handle the 29.23 requests/second for 250 devices concurrent connections which are still acceptable considering the specification of the server machine.

7. Conclusion

The architectural design of IoT-cloud platform for IoT and cloud computing integration has been presented. The proposed software platform can be decomposed into five main components: cloud-to-device interface, authentication, data management, and cloud-to-user interface component. In general, the cloud-to-device interface acts as a data transmission endpoint between the whole cloud platform system and its IoT devices counterpart. Before a session of data transmis- sion establish, the communication interface contact the authentication component to make sure that the corresponding IoT device is legitimate before it is granted to send the sensor data to cloud. Notice that a valid IoT device can be registered to the cloud system through web console component. The received sensor data are then collected in data storage component. Any stored data can be further analyzed by data processing component. User or any developed applications can then retrieve collected data, either raw or processed data, through API data access and web console. From functional testing result shown that the proposed system has been able to provide the communication, security and storage functionalities. Furthermore, the performance result shows that there exists an impact of the increase of concurrent device connections on the delay and throughput of sensor data reception from IoT devices to the cloud system.



(a) Response Time

(b) Throughput

Figure 9. (a) Response time and (b) Throughput of IoT Cloud Platform Sensor Data Reception for Various Amount of Concurrent Connections

References

[1] Lee, In, and Kyoochun Lee. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*. 2015; 58(4): 431-440.

[2] Al-Fuqaha, Ala, et al. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*. 2015;17(4): 2347-2376.

[3] Botta, Alessio, et al. Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*. 2016; 56: 684-700.

[4] Kobusinska, A, et al. Towards increasing realibility of clouds environment with RESTful web services. *Future Generation Computer Systems*. 2018; 87: 502-513.

[5] Biswas, A.R, et al. IoT and cloud Convergence: Opportunities and challenges. *IEEE World Forum on Internet of Things (WF-IOT)* (2014): 375-376.

[6] Villari, M, et al. Leveraging the Internet of Things Integration of Sensors and cloud computing. *International Journal Distributed Sensor Network*. 2016; 12: 9764287.

[7] Alam, S, et al. *SenaaS: An event-driven sensor virtualization approach for Internet of Things cloud*. in 2010 IEEE International Conference on Networked Embedded System for Enterprose Application. 2016: 1-6.

[8] Distefano, S, et al. A Utility paradigm for IoT: The Sensing Cloud. *Pervasive Mobile Computing*. 2015; 20: 127-144.

[9] Díaz, Manuel, Cristian Martín, and Bartolomé Rubio. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications*. 2016; 67: 99-117.

[10] Hedi, I, et al. IoT network protocols comparison for the purpose of IoT constrained networks. in 2017 *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2016: 501-505.

[11] Al-Fuqaha, Ala, et al. Toward better horizontal integration among IoT services. *IEEE Communications Magazine*. 2015; 53(9): 72-79.

[12] Bhawiyuga, Adhitya, Mahendra Data, and Andri Warda. *Architectural design of token based authentication of MQTT protocol in constrained IoT device.* Telecommunication Systems Services and Applications (TSSA), 2017 11th International Conference on. IEEE, 2017.

[13] M. Aazam and E. N. Huh, *Fog Computing and Smart Gateway Based Communication for Cloud of Things,* in 2014 International Conference on Future Internet of Things and Cloud. 2014: 464–470.

[14] A. Farahzadi, P. Shams, J. Rezazadeh, and R. Farahbakhsh, Middleware technologies for cloud of things-a survey, *Digit. Commun. Netw.*. 2017.

[15] L. Hou et al., Internet of Things Cloud: Architecture and Implementation, IEEE Commun. Mag., 2016; 54(12): :32–39.

[16] G. J. L. Paulraj, S. A. J. Francis, J. D. Peter, and I. J. Jebadurai, Resource-aware virtual machine migration in IoT cloud, Future Gener. Comput. Syst.,2018; 85: 173–183.

[17] Ahmed, N., H. Rahman, and Md I. Hussain. A comparison of 802.11 ah and 802.15. 4 for IoT. *ICT Express*. 2016; 2(3): 100-102.

[18] Ryu, Dae-Hyun. Development of BLE sensor module based on open source for IoT applications. *The Journal of the Korea institute of electronic communication sciences*. 2015; 10(3): 419-424.

[19] Han, Dae-Man, and Jae-Hyun Lim. Smart home energy management system using IEEE 802.15. 4 and zigbee. *IEEE Transactions on Consumer Electronics*. 2010; 56(3).

[20] Bardyn, Jean-Paul, et al. *IoT: The era of LPWAN is starting now.* European Solid-State *Circuits* Conference, ESSCIRC Conference 2016: 42nd. IEEE, 2016.

[21] Keranen, A, et al. RESTful Design for Internet of Things Systems. (2015).

[22] Bray, Tim. The javascript object notation (json) data interchange format. (2017).

[23] Apache Web Server. The Apache HTTP Server Project. (2018).

[24] Django. The web framework for perfectionists with deadlines. (2018).

[25] MongoDB, C. R. U. D. Introduction-MongoDB Manual 3.4. (2017).