

Design and Implementation of Data-Report Service for IoT Data Analysis

Raeyoung Jang
Hannam University
Department of Computer Engineering
Daejeon, Republic of Korea
renerae402@gmail.com

Wooyoung Soh
Hannam University
Department of Computer Engineering
Daejeon, Republic of Korea
wsoh@hnu.kr

Sung-Jae Jung
Hannam University
Department of Computer Engineering
Daejeon, Republic of Korea
posein@naver.com

Abstract—the number of various things began to connect to the Internet. Much more data than ever before has been recorded in the data system. Accordingly, the system that saves, analyze a data Measured from things is required. People try to predict the future through nature. IoT cloud platform that write an overflow data easily, and show a data analysis, give a warning when a situation occurs will be helpful. In this paper, reporting to the previous active service, we propose a new system in the other direction.

Keywords—IoT; Data-Report; Arduiono; Edison; Xively; ThingSpeak

I. INTRODUCTION

Internet of Things (IoT), Technology to connect the objects to the Internet, has become a new field of research and it has been a lot of technological developments. Such technology has enabled the observation of things, the management of things, and the transmission of data through network. To do this, two of infrastructure is needed [1]. One is Sensing Entity for gathering data, and the other is the Sensor Cloud to manage data. Sensing Entity has an important role that connects the sensor data to the server and it can be configured easily by a mini-board like Arduino. Sensor Cloud is a cloud computing to process the data measured from the object, provides a platform for the monitoring, analysis, control and notification system. Interest to an Internet of Things is not new. According to Gartner's top 10 strategic technology trends, IoT is constantly attracted attention from 2012 to the present. Network technology development is made possible at concept as Internet of Everything, it can be found easily nearby us [2].

In particular, the interest at the sensor network has constantly tendency of increase. Mini boards comparable to the performance of personal computer are leading the embedded market, Performance of a device like smartphones reached a level of Personal Computer. Mini Development Board like Arduino, Edison, Raspberry Pi, have made it possible to record all data using the sensor. However, just recording the data has no value by itself. The movement that makes information from data on

internet database server became active [3]. For this reason, system that analysis a data and report a result, became necessary. IoT Cloud Platform is used generally such as Xively, ThingSpeak, and Mnuvo. These cloud platform systems as service have limitations across each service, and notification service is also support narrowly [4].

In this paper, we propose an installation type analysis system that saves, analysis, report an observed sensing data values, DRSS (Data-Report Service System). The proposed system is recording the measured values, analysis and monitoring.

If the situation occurs, the system sends a message to the smart devices of administrator in real time. The proposal system performs two roles. One is data storage and Visualization, the other is a data analysis and notification. Unlike similar existing service, it plans to offer the service for free-standing scalability.

II. RELATION WORKS

A. Arduino

Arduino is an open source computing platform based on a simple microcontroller board, a tool that enables you to more easily detect and control the physical world. Arduino is easy to use for beginners with no experience in electronics or programming language, is programmed using an easy to understand code. Arduino receives input from various switches or analog sensors; it controls the output, such as lights, motors and connected actuator. The Arduino supports Windows, MAC, Linux development environment and it is an open-source platform that can be extended freely. Arduino is available while it is easy enough to create a high level control using an embedded device. By observing the physical nature through sensor-specific features, it may also be transmitted to the network using an Ethernet-related Shield. And the Arduino board design was opened, a variety of third-party boards are derived. On this account, it was able to access a lot of people to the Arduino. For programming, the Arduino platform provides an integrated development environment (IDE) based on the process, which includes support for C and C++ programming languages [6].

B. Edison

Edison is the second open source hardware platform of Intel since Galileo. Edison has received great attention from the announced. Small size, the unique interface that resembles as an SD card, support for low-power/Wi-Fi/Bluetooth4.0 was a specification that overwhelmed other development board. Edison is a high-end embedded board that uses a 500MHz dual-core Atom to the main core, 1GB main memory, 4GB storage. It includes Wi-Fi(802.11 a/b/g/n) and Bluetooth 4.0 to support communication with external.

The biggest difference with Arduino is a Yocto Linux as the OS is installed. Therefore, it is possible to use the Server machine, as well as Sensor device. We will understand a reason why Edison get strength, by consider the IoT characteristics. IoT is to connect sensor-able and computable things with the other object, the Internet. For that, Device must have a high level of CPU performance, low power, and communications capabilities. Edison has all of these features. It will need a breakout board or Arduino compatible boards, or other expansion boards because Edison is difficult to use alone. Of these, if we are using the Arduino compatible boards, we can develop as the same environment like Arduino while having compatibility. In this paper, we had attempted to study as use of Edison on Arduino compatible board.

C. Xively

Pachube was acquired by LogMeIn, remote solutions company in 2012, and it was newly opened. After introducing the prototype in 2008, it was open to the public in 2010; the acquisition in 2012 was renamed as Xively. Xively is a suitable platform for IoT Big Data. Pachube idea began in purpose to manage the real-time data from the sensors in an interactive environment, by Usman Haque that is an architect and artist based in the United Kingdom. It was developed for the purpose of connecting the people and things, the application, it can manage data in real-time from around the world via a web-based service, it provides an environment to registered users for share a gathered data and collaborate.

Xively is providing HTTP-based and Restful Open API, allows anyone to use the data. Xively focuses on platform services through the Open API rather than focuses a web site service, as a PaaS (Platform as a Service). The product through the Open API, it is possible to easily and quickly develop by registered sensor device as real-time information, history, such as to enable the administration for the environment. [7]

D. ThingSpeak

ThingSpeak is IoT platform that designed to enable meaningful connections between people and things. Features of ThingSpeak is real-time data collection, data analysis, data processing of the position information, data visualization, message transmission, etc. using a connected SNS, via an open source API to support a various platforms. It helps you easily transfer data from embedded devices such as Arduino, Raspberry Pi. It supports various languages and environments such as C, Node.js, and Python [8].

III. PROPOSAL SYSTEM

A. Data-Report Service System

The proposed system, DRSS (Data-Report Service System), uses the Apache, PHP, MySQL, curl. System flow chart is shown in Figure 1. The measuring device is transmitted using the HTTP Request how the data collected by the sensor to a server over a network and is saved. The default role is to write to the entire values input database and analyzed. After the server have two operating modes, save the value in the database.

One is a User Report Interface to show a web page when the client connected to the server directly over the network, the other is a function that throws the real-time notifications to a predefined smart device if the value of the specified value or a specified amount of change for each sensor channel are detected.

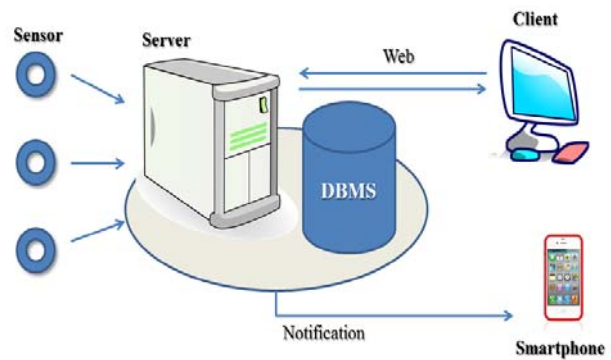


Figure 1. System Flow Chart

B. Setting for System

Prior to actual use, after the user has connected to management page, must have two settings. The first operation is creation of the sensor channel ID. When you create a sensor channel ID, the server is in a state that can receive the data.

The second is a token registration of smart devices for user notifications. The proposed system uses an application called PushBullet for user notifications. Accordingly, personal token registration is required. Google or Facebook account is needed to use the PushBullet. Users should download and install the PushBullet through Google Play or App Store. After logging in PushBullet Webpage, you can obtain the individual Token of "Access Token" page. Copy it and, write a token value at the proposed system's settings, "Connect a PushBullet by Token". If token is not saved in the system, alerts are not enforced.

C. Principle

Usage that transfers the value to the server is as follows: Figure 2.

```
http://ServerIP/DRSS/insert?sensor_id=aaa&sensor_type=bbb&value1=ccc&value2=ddd
```

Figure 2. Send a data to server by HTTP Request

Variable sensor_id that sent to the server is the generated sensor channel ID. If it is not a registered channel ID, the system automatically generates the channel

ID as written and stores the transmitted value. In general, because the data on the IoT is an insert type, not an update type, it was considered to only handle Insert task in the proposed system. When the actual value is transmitted via the sensor network, the server parses the value to be stored in the database. And storing a value, System that exceeds a specified value for each channel, or it is determined whether more than a given amount of change compared to the value stored immediately before. When an event occurs, a server save the status in the event table and get the end of the situation. Event table is periodically determined whether the non-operation status, if there is a non-operating status, process sends a notification using the registered PushBullet token. In this case, the administrator can be checked in real-time event by the smart device or PC.

D. Performance Comparison

In this paper, we measure the value using the Edison Board and ultrasonic distance sensor (HC-SR04), by transmitting the same data to the sensor Xively, ThingSpeak, DRSS and It was compared an each service performance and functionality. Edison-Arduino boards can easily be developed and uses the Arduino IDE. In addition, this development is possible using JavaScript and C/C++ with Intel XDE and Eclipse. Features of Edison development environment is shown in the following table. The test was performed using the Arduino IDE (for Edison), Sensor (HC-SR04) can be the detection range from nearest 2cm to farthest 4m. Figure 5 is a board used in the experiments.

TABLE 1. COMPARISON OF EDISON IDE

IDE	Language	Features
Arduino IDE (Edison)	C/C++	Same as Arduino IDE Code limited is 10MB
Intel XDE	JavaScript	Node.js Debugging
Eclipse	C/C++	CDT Available Pin Control using MRAA, UPM

E. Code for Xively

Figure 3 is a code for transferring the value to Xively. Xively provides a number of the Library for Arduino and other languages. Including Xively and HttpClient Library, device can communicate with the Xively server. The main routine is to pass values with a function of xivelyclient object by the factor using the API key, FEED that can be obtained from the Xively website.

```
void loop() {
  WiFiClient client; XivelyClient xivelyclient(client);
  //read sensor values
  digitalWrite(trig, HIGH); delayMicroseconds(10);
  digitalWrite(trig, LOW);
  int distance = pulseIn(echo, HIGH) * 17 / 1000;
  //print the sensor value
  Serial.println(distance); datastreams[0].setFloat(distance);
  //send value to Xively
  Serial.println("Uploading it to Xively");
  xivelyclient.put(feed, xivelyKey);
}
```

Figure 3. Send a Value to Xivley

F. Code for ThingSpeak

ThingSpeak supports the Library for Arduino, however, because it is also possible the GET method of data input via the HTTP URL. In the experiment, it was written in code via the direct HTTP Request. Edison is capable of HTTP communication with other networks using the client object created by Wi-Fi. ThingSpeak can transmit data by using only the API key for each channel unlike Xively. The following Figure 4 shows the code to transfer the value to ThingSpeak server.

```
// Send Value to ThingSpeak
if (client.connect(server, 80)) {
  // Make a HTTP request:
  client.print("GET /update?key=");
  client.print(thingspeak_channel_key);
  client.print("&field1=");
  client.print(distance);
  client.println("HTTP/1.1");
  client.println("Host: api.thingspeak.com");
  client.println("Connection: close");
  client.println();
}
```

Figure 4. Send a Value to ThingSpeak

G. Code for Proposal Analysis System

The proposed system, (DRSS: Data-Report Service System), passes the value through the HTTP Request as ThingSpeak. Edison only serves to transfer a value, and, exactly, the server is written an entered value to the database through the sensor_id in a web page. The difference between the server and ThingSpeak is the number and role of the parameters passed by GET Method.

IV. RESULT BY EXPERIMENT

TABLE 2. DIFFENCE BETWEEN IOT PLATFORM

	Xively	ThingSpeak	DRSS
Type	Serviced	Serviced	Installed
Transmission Mode	GET, API	GET, API	GET
Delay Time	5sec	15sec	free
Visualization	Yes	Yes	Yes
Sensor Channel	Multiple	Multiple	Multiple
Notification	LIMITED	LIMITED	Instantly

The following Table 2 shows a comparison of differences between the Xively, ThingSpeak, and DRSS according to the result. The DRSS, proposal system is an installation package unlike Xively, ThingSpeak that existing service type IoT platform. This package is available at a server that installed an Apache, PHP, MySQL, curl. In the experiments, Edison sends the same data value to the three without delay. In Xively, after about

five minutes, it generates a Status Code 403 ERROR and record status was irregular. Xively must set a delay time longer than 5 seconds for stable record. ThingSpeak is different than Xively. ThingSpeak received input values themselves every 15 seconds regardless delay time. However, for stable data record, you must specify the delay time for at least 15 seconds. The DRSS was to write data without limitation unlike conventional platforms.

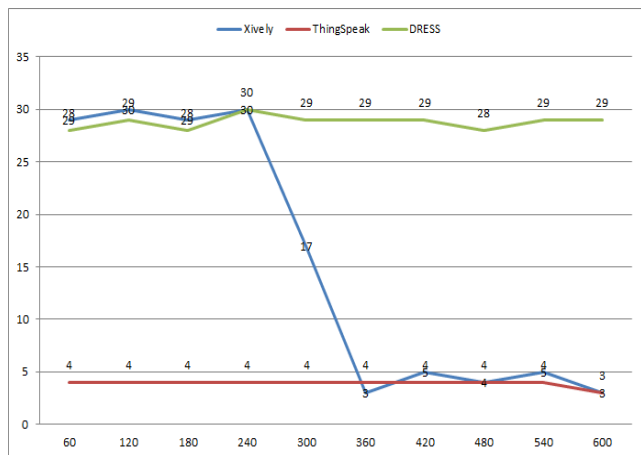


Figure 5. Data Processing number for 10 minutes

Figure 5 shows the results by a sequential line chart showing the average, measured three times repeated when Edison transmit data to Xively, ThingSpeak, DRSS for 10 minutes, while it is a 1000ms delay time, a single channel. With this result, we can get the difference between the IoT platforms under such conditions.

The reason for this result is a service platform, consider a distributed system resources, however, on the other hand, the DRSS is designed for the best use of system resources because it is for single stand-alone server. Typically, because sensor data are measured a nature phenomenon, it rarely changes the value quickly. So, there is no problem even if the delay is longer than 15 seconds as ThingSpeak. For example, temperature and humidity variation is not as fast. This is useful services such as installation type when the measured value change in a short time is required for a new attempt using a complex sensor in IoT sensor networks.

In addition, there are differences in the notification that send a status to users. Xively sends a notification through the HTTP POST method when the sensor values fit the criteria specified value. However, it is not possible to work alone. It helps deliver situation to Twitter, Facebook, Email and more, in conjunction with services that support the Trigger Function, such as Zapier. For example, when a current value is more than 30 degrees by a temperature sensor, it is possible to activate the Trigger. But, It has limited functionality that must be the correct value.

ThingSpeak using Twitter typically is more restrictive than Xively. Not it throw a notification from server by a change of value, but It posted a message on the Twitter Timeline authenticated users via HTTP POST Request when a specified situation occurs from machine like a Arduino. Not a function of the platform, only function has to set at the sensor. After setup, it is difficult to modify. On the other hand, the proposed system can set the amount

of change as a value or as a percentage at a web page. The server is designed to deliver immediately that this situation occurred through the curl daemon towards smart device when the value for a particular value or a specified amount of change. Apps support for Android, iPhone, and supports PC (Chrome web browser extension), connect to users in real time. Through this, notification is passed easily without limitation, and we see the IoT allows real-time sensor status from anywhere, anytime.

The feature of proposed system in this paper can be summarized as follows:

First, it is the type of the installation package, it can be used an unlimited server resources. Thus, it is free to record more data. Second, you can easily find out the accumulated data through the graph depending on the time or date change. Third, when any of the conditions occur, it delivers instant alerts to your smart device.

V. CONCLUSIONS

Existing service is less freedom for the user although the advantages described in the paper. Xively, ThingSpeak has a disadvantage that uses the data platform according to the policy of the service providers. Xively offers a reminder service using email, twitter, but real-time alerts to the user's smartphone is difficult to use. ThingSpeak has provided other types of alerts not a notification of the change in value. When situation, according to a given value or a percentage change occurs, it was to provide a notification service to your PC, smart device through the app. In addition, the system suggests a structure for receiving the data processing freely, for using server resources to the maximum because it is assumed a single-server user. Through this, more details analysis for Big Data, using advantage of the IoT data, will be possible.

Since then, the system tries to reflect a more efficient thread management and big data environments with additional features by modifying the system as JavaScript, NoSQL. Furthermore, we have plans not merely analyze an IoT data to receive but also be able to control the IoT device over the web, conversely.

REFERENCES

- [1] X.Y. Chen & Z.G. Jin, "Research on key technology and applications for Internet of Things," *Physics Procedia*, vol. 33, pp. 561-566, 2012
- [2] Feng, X. & Laurence, T.Y.. 2012. *Internet of Things*.
- [3] J. Staten. (2013). *Cloud Computing Predictions for 2014: Cloud Joins the Formal IT Portfolio*. Available: http://blogs.forrester.com/james_staten/13-12-04-
- [4] James,G & David, R, "Extracting Value from chaos", *IDCIVIEW*, 2011
- [5] Weber, R.H., 2010. *Internet of Things – New security and privacy challenges*. *Computer Law & Security Review*, 26(1), pp.23–30. Available, <http://linkinghub.elsevier.com/retrieve/pii/S0267364909001939>
- [6] Arduino, Available, <http://www.arduino.cc/en/Tutorial>
- [7] Xively, Available, <http://personal.xively.com/dev/tutorial>
- [8] ThingSpeak, Available, <http://thingspeak.com/docs/tutorial>