Implementation of an Intelligent Indoor Environmental Monitoring and management system in cloud

Chao-Tung Yang, Shuo-Tsung Chen, Walter Den, Yun-Ting Wang, Endah Kristiani

PII: S0167-739X(18)30418-7
DOI: https://doi.org/10.1016/j.future.2018.02.041
Reference: FUTURE 4012

To appear in: *Future Generation Computer Systems*

Received date : 31 August 2016
Revised date : 18 February 2018
Accepted date : 25 February 2018

Please cite this article as: C.-T. Yang, S.-T. Chen, W. Den, Y.-T. Wang, E. Kristiani, Implementation of an Intelligent Indoor Environmental Monitoring and management system in cloud, *Future Generation Computer Systems* (2018), https://doi.org/10.1016/j.future.2018.02.041

Implementation of an Intelligent Indoor Environmental Monitoring and Management System in Cloud

Chao-Tung Yang*[1], Shuo-Tsung Chen[1], Walter Den[2],
Yun-Ting Wang[1] and Endah Kristiani[13]

[1]*Department of Computer Science, Tunghai University,*
Taichung City, Taiwan, (R.O.C.) 40704
[2]*Department of Environmental Science and Engineering,*
Tunghai University, Taichung City, Taiwan, (R.O.C.) 40704
[3]*Department of Industrial Engineering and Enterprise Information,*
Tunghai University, Taichung City, Taiwan, (R.O.C.) 40704
*Email: *ctyang@thu.edu.tw; shough33@yahoo.com.tw; wden0323@gmail.com;*
winnie3536@gmail.com; endahkristi@gmail.com

Abstract

Indoor environmental monitoring and management system has played an essential role in public health sustainability. By monitoring the indoor air quality in public areas such as schools, offices, home or other buildings, the authorities will be given a better picture of the indoor air quality to take the right steps to ensure the better air quality for people inside the buildings. This system also can give information about indoor air quality information for the society. Therefore, to achieve this goal, we need to develop a monitoring system by using Information Technology based on Big Data and Cloud Computing environment to give warning. In this paper, we propose Intelligent Indoor Environment Monitoring System (iDEMS) combined with ZigBee wireless sensor network technology to store and process environmental data in HBase. The mechanism of the proposed system is classified into three stages: data collection, data processing, and information monitoring. To understand an Intelligent Indoor Environmental Monitoring, first, we collected the gas from Intelligent Indoor Environmental Monitoring through the environmental sensors with ZigBee wireless sensor network technology. We build a platform for iDEMS to collect the data related to the indoor gases. Second, the environmental data collected in the first stage will be stored and processed in HBase which support massive data storage and free to increase storage capacity for the analysis and processing of Big Data. In this stage, we also compared several data-input methods to import data in HBase much more efficient. Third, the intelligent-control socket is integrated into iDEMS and give the warning if the air quality exceeds the absolute legal limit based on the air quality index rule from the authority. Finally, iDEMS presents the resulting information by a web-based Monitoring Platform so that users can use the Internet to monitor the environment and enable them to utilize these informed decisions on managing and improving the environments.

Keywords: **Sustainability, iDEMS, Environmental Sensors, Cloud Computing, Big Data, HBase**

1. Introduction

The importance of the indoor environmental monitoring system for human safety and health is growing, specifically intelligent monitoring and control system for air cleanliness. Most people tend to think that air pollution is only happening in outdoor environments, such as exhaust fumes from vehicles, the burning of fossil fuels in industries, or forest fires. In reality, the air inside schools, offices, homes, and other buildings can be more polluted than the air outside. Indoor air pollution might be polluted by volatile chemicals from building materials and cleaning products, cigarette smoke, the burning of fuels from stoves and ovens. Another significant source of indoor pollution is dust mites, mold spores, and pet dander. People with special needs, like children, elderly and people with the pulmonary disease might be especially sensitive to indoor pollutants. After repeated exposure, other effects might appear years later. These motivate us to develop the model and system to monitor the value of indoor air pollution level using the sensor in Big Data and Cloud Computing environment. We have reference to the scientific and commercial data grow exponentially, which focus on improving the resource utilization, reducing power consumption, and achieving the large-scale data analysis [1, 2, 3].

In the real-time application monitoring, we need to apply Internet of Things (IoT) consists of interrelated sensors, processors, and communication hardware that collect, send and act on data acquired from surrounding environments. However, this platform has some unprecedented characteristics such as have a vast number of connections simultaneously, needs to be highly scalable, reliable and secure environment. Thus, it is substantial to use Big Data and Cloud Computing to capture the large-scale data processing. Another critical point, we adopted OpenStack to deploy a distributed system that manages data from several sensors in given spots for the IoT back-end.

M. Rönkkö et al. [2] studied automation of the selection and sequencing of preprocessing methods based on the user requirements. The authors propose the use of characterizations and a reachability algorithm to solve the selection and sequencing problem. In their paper, they present the algorithm and argue for its correctness, how the algorithm is implemented as a cloud service, and illustrate the use of the service with simple case studies. S. Trilles et al. [3] presented how to embed an open sensory platform for both hardware and software in the context of a smart city, more specifically in a university campus. For this integration, GIScience comes into play, where it offers different open standards that allow full control over ”smart things” as an agile and interoperable way to achieve this. To test their system, the authors have deployed a network of different sensory platforms inside the university campus, to monitor environmental phenomena. D. Meana-Llorian et al. [4] proposed a new approach to control the temperature using the Internet of Things together its platforms and fuzzy logic regarding not only the indoor temperature but also the outdoor temperature and humidity to save energy and to set a more comfortable environment for their users. Although these works have developed the system using algorithm and sensors as IoT, there is yet no study focusing on improving how to manage large data captured by sensors that have unprecedented characteristics such as have a vast number of connections simultaneously. In our work, we elevate the capability of our system using methods of high performance and high throughput computing environment to give efficient, reliable and quick access.

This paper is an extended version of our previous paper by C.-T. Yang, et al. [5], which demonstrated the applicability of air quality monitoring prototype using Carbon Dioxide (CO2) concentration as an essential indicator of the indoor air quality in Taichung Veterans General Hospital. This real-time air quality monitoring system provides the authority for monitoring air quality in the designated area and notifying medical staff via mobile phone when the air quality deteriorates below a threshold level. We improve this paper by comparing Wireless Sensor Network (WSN) data with the Temperature and Humidity Index (THI) value to categorize the level of comfort value of the environmental index. THI is a combined index of temperature, dew point, and relative humidity. If THI is low, human body feels increasingly cold. Conversely, human will suffer heat exhaustion. Therefore, the values should be between 20 and 26 to let human body feel comfortable. Besides, we provide some other results from the sensors namely the Temperature, Humidity, Carbon Monoxide (CO), Formaldehyde, and Volatile Organic Compounds (VOCs). In the previous paper, the data from WSN is stored in MySQL Database in short time based from the index value of CO2 captured by sensors; then the program will determine whether it exceeds the limit or not to give warning. In our extended system, we store data from ZigBee WSN technology into HBase database system; then we will do the further operation from this database. In this work, we use Zigbee WSN, which has significant advantages in the term of low-power consumption and low data rate. To give a warning system, we embed an intelligent socket that will automatically give notification when the air quality exceeds the certain legal limit. Additionally, we enhance the system performance with several technical like index tuning and some adjustment to provide our data can be read and written with the high-performance computing. We designed and implemented a distributed data flow management system based on Hadoop platform. In this case, MapReduce is used to process user requests, and The Hadoop Distributed File System (HDFS) is used to manage data flows. It can efficiently improve the processing time for data collection and data analysis.

The implementation and applications of Hadoop open source are widespread in the past few years. T. Kun-Fu et al. [6] showed how to store large amounts of data transferred from MySQL to HBase, and arrange with Thrift that was initially developed by Facebook. Thrift is an Apache project contains a set code-generation tools that allows developers to build Remote Procedure Call (RPC) clients and servers by just defining the data types and service interfaces in a simple definition file. This given file is used as an input. Then, the code is generated by the compiler to build RPC clients and servers for communicating across programming languages. We used Java code to create graph and chart from the analyzed data to display resulting information on the user interface in our system.

In summary, this research aims to develop a prototype of an intelligent indoor environment monitoring system using sensors in the Cloud Computing and Big Data environment based on the comfortable THI value. The prototype will provide real-time access with high-performance computing to give efficient, reliable and quick access to the user. Moreover, the prototype will address the following goals:

- Developing iDEMS in OpenStack as a Cloud Computing Application.
- Developing an architecture of IoT for environmental monitoring by combining with ZigBee WSN technology.
- Developing a distributed computing environment based on Hadoop.
- Processing the environmental data in HBase.
- Comparing WSN data and the THI value to categorize the level of comfort value of the environmental index.
- Embedding an intelligent socket to automatically give notification if the air quality exceeds the certain legal limit.
- Enhancing the system performance such as index tuning and some adjustment to provide high-performance computing.
- Developing the service to connect back-end and front-end HBase data using Thrifts.
- Displaying the resulting information in graph and chart based on the web application using Java programming as a client.

The rest of this work is organized as follows. In Section 2, we review background and preliminaries. In Section 3, we introduce the proposed system design and implementation. Section 4 shows our experiments and results. In Section 5, conclusions and future work are given.

2. Background and Preliminaries

In this section, background knowledge and related work are reviewed. First, we discuss the Cloud Computing, OpenStack and IoT. Then we discuss Big Data, NoSQL, Hadoop, HDFS and HBase about Big Data analysis and processing applications. Finally, we discuss how to categorize the level of comfort value of environmental index by comparing iDEMS with THI Formula.

2.1. Cloud Computing

Cloud computing is an operation mode based on the Internet. In this way, the resources of hardware and software can be provided to computers and other devices on demand. Users no longer need to know the details of the cloud infrastructure, or have the appropriate expertise and direct control [7, 8, 9].

2.2. *OpenStack*

OpenStack [10] is one of Infrastructure as a Service (IaaS) cloud computing projects. It was launched by NASA and Rackspace in 2010, written with Python and used Apache license, thus being massively scalable and feature-rich. Cloud files platform of Rackspace with the company's code to build OpenStack project in cloud object storage foundation. It means that NASA use Nebula to build cloud computing in OpenStack. The technology includes a series of interrelated components that can provide multiple solutions for cloud infrastructure. Its mission is to help companies build systems running on standard hardware to provide cloud computing services. OpenStack includes the operations of the compute module, networking module and storage module, as shown in Figure 1. Above three modules, the categories dashboard module uses a centralized management of the three modules. Finally, four modules combined in OpenStack as shared services that can provide external computing resources with elastic expansion or scheduling of VMs approaches [11, 12].

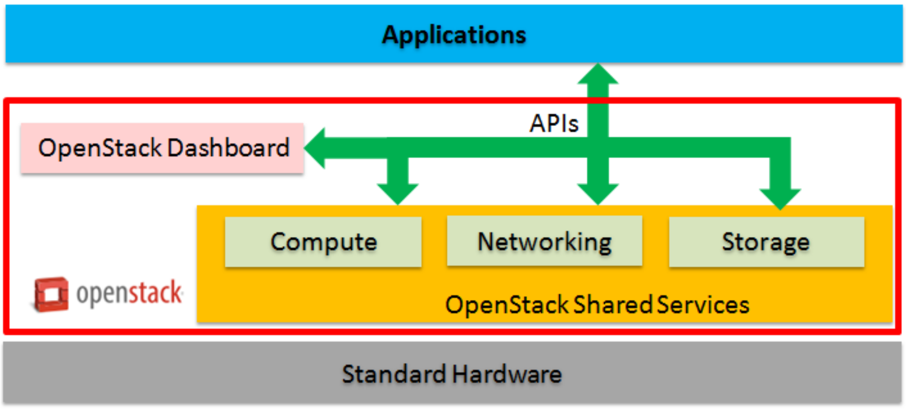


Figure 1: OpenStack Architecture

2.3. *Big Data*

Big Data refers to data in such a huge scale that, within a reasonable time, cannot be manually captured, managed, processed, and organized to become information comprehensible by human E. Feller et al.[13]. Compared with the individual analysis of small independent data sets with the same total amount of data, after combining the various small data sets as a Big Data set, additional information and data relevance can be retrieved and used to detect trends, determine product quality and real-time messages, etc. Such use is the major reason for the prevalence of Big Data.

2.4. NoSQL

NoSQL appeared in 1998. It is developed by Carlo Strozzi as a lightweight, open source, relationship database without SQL function. In 2009, Eric Evans from Rackspace's put forward the concept of NoSQL once again. In this time, NoSQL mainly refers to non-relational, distributed, and not provide ACID repository model. The slogan of NoSQL East conference held in Atlanta is "select fun, profit from real world where relational= false."Therefore, the most common explanation is "non-associated type", which emphasizes the advantages of key-value stores and document repository, rather than simply opposes Relational Database Management System (RDBMS). The full name of NoSQL is Not Only SQL. It is different from relational database management system design [14, 15].

2.5. Internet of Things (IoT)

The IoT is based on the Internet, traditional telecommunication network and other information carriers to enable all ordinary physical objects, which can be independently addressed, achieve interoperability of networks . IoT is machine to machine (MTM) with the Internet. It covers everything in the world by using RFID and wireless data communication technology [16]. IoT generally uses a wireless network; since the number of devices around everyone can reach 1000 to 5000, so IoT might include more than 500 Trillion objects. By implementing the IoT, everyone can use electronic tags to find real objects on the Internet and find out their specific locations. Users can use a central computer to manage and control machines, equipment and personnel; they can even remote control house devices and cars, and search locations to prevent goods from stealing. By implementing IoT, systems with GPS can communicate with each other and share information.

2.6. WSN

A WSN consists of numerous automatic devices distributed in space and composes a wireless a communications network. In the WSN architecture, sensors are designed to have features of a small size, low cost, low power consumption, easy to build, and with a distributed environmental sensing capability [17]. In the beginning, "Smart Dust", a system of many tiny microelectromechanical systems (MEMS) was originally proposed by University of California, Berkeley in 1990s. Smart Dust not only controls physical state of the environment by using sensors in different locations, but also has applications in military-related intelligence gathering. Now, because of advancements of MEMS and nanotechnology, the sensors are constantly shrinking in volume, lightweight, and carrying positioning sensing nodes. Various types of micro-sensor elements are combined with wireless transmission communication networks, and large number of sensors can be spread in the environment to collect and provide useful data for people any time. Their applications today can be used for fulfillment of a Lifestyles of Health and Sustainability (LOHAS) environment to have convenience, safety, comfort, and energy saving. Besides, WSN has been widely used in various fields such as environmental sensing and ecological health detection, and traffic control. ZigBee [18, 19] is a low-speed, short-range wireless network protocol; its underlying layer uses IEEE 802.15.4 standard for media access control and the physical layer. ZigBee is developed by ZigBee Alliance of Honeywell Company [20]; the idea of ZigBee, a self-organized wireless ad-hoc network standard, was conceived in 1998. The main features of ZigBee are low-speed, low power, low cost, supporting a large number of network nodes and variety of network topologies, and its application is simple, fast, reliable and secure.

2.7. Hadoop

Hadoop [21, 15] is an open-source software for reliable, scalable, distributed computing under the Apache Software Foundation. The initial prototype of Hadoop-Nutch was developed for web searching by Doug Cutting and Mike Cafarella. In 2006, Doug Cutting joined Yahoo and set up a professional team to continue research and development of this technology, officially named as Hadoop. Hadoop is written in Java; it can provide a distributed computing environment for huge data. The concept of Hadoop architecture is based on the BigTable and Google File System papers published by Google. Currently, Yahoo! and other companies have teams for Hadoop development; and more and more companies and organizations publicly express the intention to use Hadoop as cloud computing platform.

Hadoop includes a number of sub-projects. Hadoop MapReduce provides a distributed computing environment; HDFS provides a lot of storage space; and HBase provides a BigTable-like distributed database [22]. There are other parts that can be used to link together these three main parts, providing easy integration of cloud services, as shown in Figure 2. The following section will introduce HDFS and HBase.

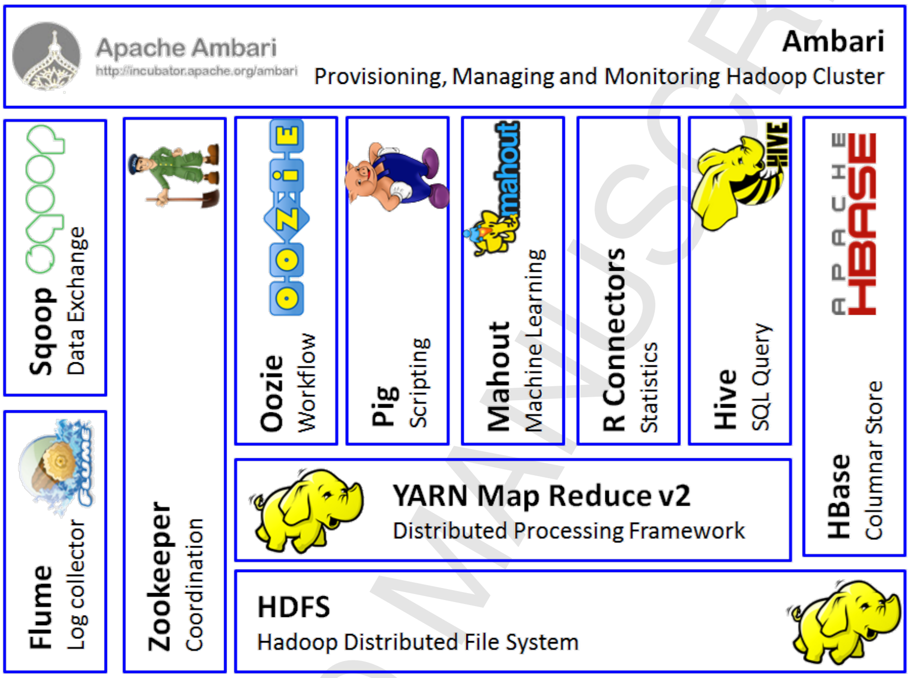


Figure 2: Apache Hadoop Ecosystem

2.8. HDFS

Hadoop is a cluster system, which is an integrated super computer expanded from a single server to thousands of machines. In this cluster the information is stored in HDFS, which integrates dispersed storage resources into a fault-tolerant, high efficiency, large capacity, and remote backup storage environment. In Hadoop systems, the large amount of data and temporary files generated during computation are stored on this distributed file system. Through HDFS, Hadoop can store tera bytes (TB) or peta bytes (PB) of Big Data [23]. It does not need to worry about the size of a single file exceeding the size of a disk sector, or data lost caused by damaged machines. HDFS has not been integrated into the Linux kernel, and it only can operate files via dfs shell command of Hadoop, or use FUSE to be treated as a file system under the user space. All systems under Hadoop are integrated with

HDFS as a data storage, backup, and sharing medium. As mentioned earlier, when the system is assigning computing tasks, MapReduce will assign computing task to the nodes stored with the data for operation, thus reducing the time to transmit the large amount of data via networks [14, 15].

2.9. HBase

Apache HBase [14, 24] is a project undertaken by Powerset to deal with the huge amount of data generated by natural language searching. But now it is already a top-level project of the Apache Foundation. HBase runs on HDFS and has attracted widespread attention. Facebook chose HBase to implement it's messaging platform in November 2010. HBase is distributed database on HDFS architecture and it is different from general relational database. It is modelled with reference of Google's BigTable and is programmed in Java. It is fault-tolerant to store massive sparse data. The table from HBase can be used as inputs and outputs in MapReduce tasks. It can be accessed through the Java API, and it also can be accessed by REST, Avro or the Thrift API. Today, it has been used in a number of data-driven sites, including Facebook's messaging platform. In order to conveniently disperse data and operation work, the entire data table is divided into many regions. One region is composed of one or more columns, which can be stored in different hosts called as the region servers; master server is used to record a region corresponding to each region server; besides, there is the master server to record every region server corresponding to every region. The master server will automatically reassign regions on the region server that cannot provide services to another region server.

3. System Design and Implementation

This section introduces the design of iDEMS and its implementation. To test the system in a real environment, we have implemented iDEMS in buildings of our school, i.e., Tunghai University, Taiwan. The concept of proposed system and architecture of the system prototype for data transfer are introduced in 3.1. Next, the proposed system architecture is presented in 3.2. The flowchart of the environmental information monitoring system with intelligent sockets is shown in 3.3 and the cloud architecture of iDEMS is proposed and described in 3.4. The implementation of the proposed system is presented in 3.5.

3.1. Platform Concept

In order to efficiently store, process, and analyze massive data obtained from environmental sensors, many services are implemented in iDEMS. In the back-end of these services, various kinds of environmental sensors are installed inside the experimental space, and the environmental data captured by the various sensors are transmitted to a data storage unit by a WSN. Finally, via Ethernet, these data are stored in a cloud distributed database. We adopt OpenStack to build a cloud cluster with multiple virtual machines. As shown in Figure 3, the proposed services are introduced as follows:

- The original sensor data are uploaded to the cloud platform and stored in a distributed database.
- After processing and analyzing data in real time, the useful information will be displayed to the user.
- The data stored in the distributed databases can be handled; via the cloud cluster data processing services, the user can search, filter, and analyze the stored data.

Figure 3: Diagram of the concept of services

3.2. System Architecture

Figure 4 and Figure 5 show our system structure and architecture, respectively. The sensor data are transferred to the cloud platform; upon receiving the data, the data collection service is invoked to store the data in the distributed database system in the cloud. The data in the cloud storage can be accessed by data searching services and data processing services via the cloud computing platform. After data processing operations, the computed results can be sent to update information used in the web application service and user interface. Finally, iDEMS uses web technologies such as HTML5, JavaScript and CSS 3 to visually display the results. In order to achieve the goal of creating a dynamic web, jQuery is used to implement the user interface. Users can view real time environmental information via the user interface. The flow chart of iDEMS is shown in Figure 6.

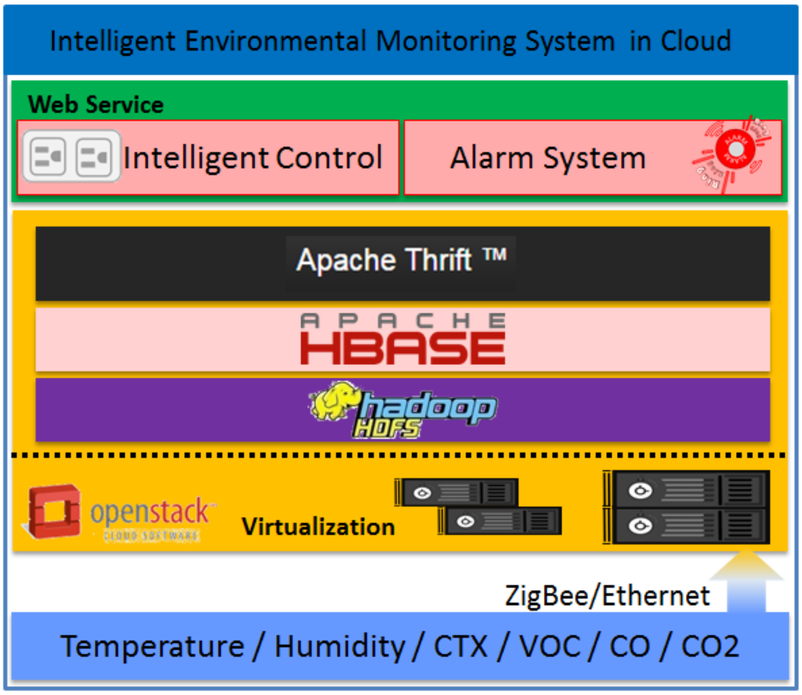


Figure 4: System Structure

3.3. Data Collection Service

Capturing various environmental information smoothly is the most important task in developing an environmental monitoring system. To obtain the information, we use ZigBee network protocols to transmit the environmental data and sensors data. The sensory data are packed by the ZigBee Router and transferred to the ZigBee Coordinator via the ZigBee wireless network protocol (IEEE 802.15.4). The ZigBee Coordinator is used for data collection, and it converts information from various sensors in a readable

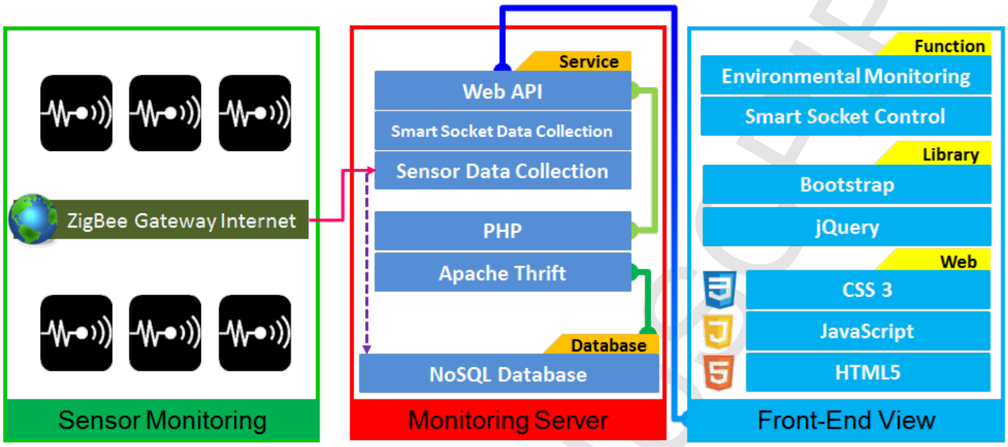


Figure 5: System Architecture of our work

data format. Figure 7 shows the architecture used for data transform and storage in the database. We store data into the database via Ethernet. After receiving messages in different communication protocols, e.g., TCP and UDP, the server processes the received data and then writes the processed data in database. Moreover, to ensure that the processed data are shown in real time mode, sensory data are also written to Notepad and directly displayed on the system web page by using PHP. This approach eliminates the time spent in writing data to the database before reading. In preparing the web content, JSON data are transmitted through the PHP to HTML, then, by using file chart.js, it is represented in a variety of charts, which can be used in subsequent data processing and analysis.

The flowchart of our work is described as following. There are two steps:

1. Get data source from sensor
2. Write into HBase database (NOSQL)

Algorithm 3.1: GET DATA INSTRUCTION()

```
while in record time
    do { get environment information via sensors
         write information into HBase
```

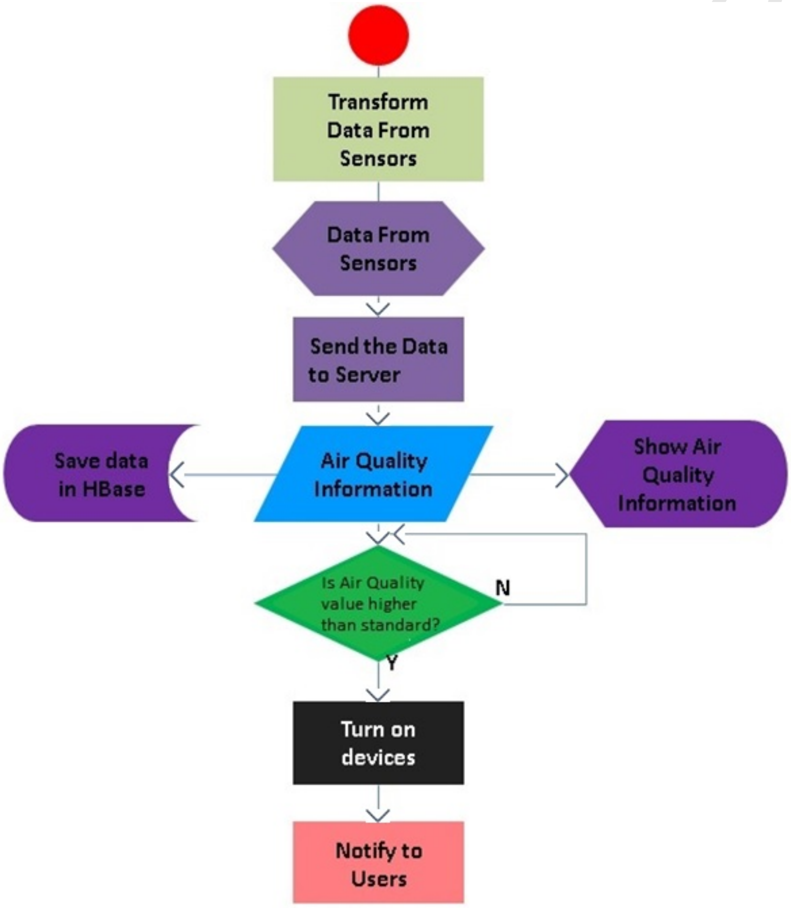


Figure 6: System Flow Chart of our work

3.4. Design Cloud Architecture

Figure 8 shows the implementation of iDEMS in cloud architecture consisting of multiple hosts and each host has its own vCPU, memory and storage. These hosts comprise a computing cluster in cloud based on the HDFS. NoSQL and Apache HBase are used for data storage in the cluster to store indoor air quality data collected by sensors. Data in HBase can be accessed by the HDFS; the web page can connects with HBase via Apache Thrift to access huge data. By using Cloudera(CDH) Cluster for the deployment and monitoring of each Data Node, the state of the cluster can be easily and quickly perceived, allowing the system to respond faster to problems.

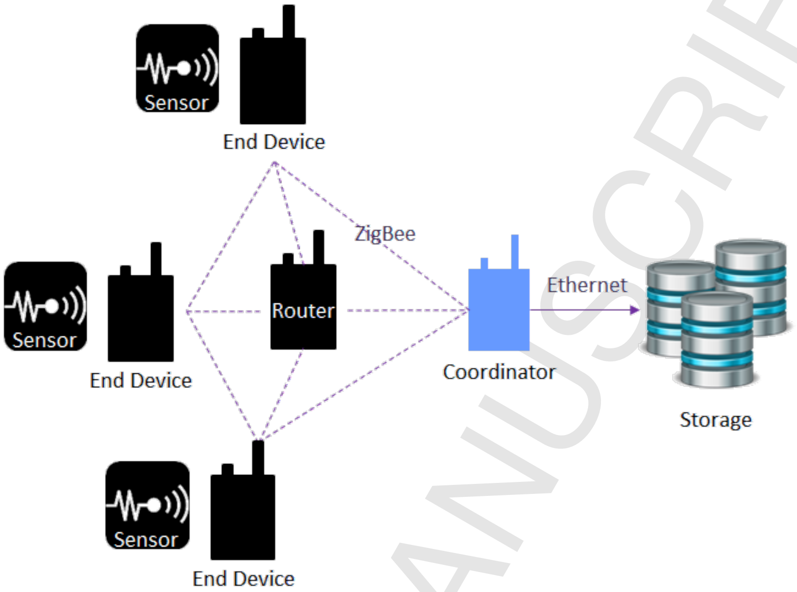


Figure 7: Data transform and storage into databases

3.4.1. Data Conversion Storage Service

The accumulative amount of environmental data keeps increasing every second, so it is essential to build the cloud platform that is suitable for conversing and storing of the huge collected data. The system needs conversing and then importing existing data stored in MySQL database; besides, it needs accurately collecting and storing the subsequent real time environmental data into its database. In this work we mainly import data in single-threaded mode, in which data are imported one by one by scanning rows of columns. To import data into the distributed database, the Zookeeper communicates with HBase and writes data into it. We will make adjustment to enhance the performance of importing data.

- **Data Collection Storage - “Put” and “PutList”**

  The write operation in HBase can be divided into two types. One type is the single put, mainly used for executing a single write operation to put a specified row key record into HBase, as shown in Figure 9. First, a data item is read from MySQL, then according to the format, it is set to be matched with each corresponding row and column. To import data, after the Zookeeper communicating with HBase, the data

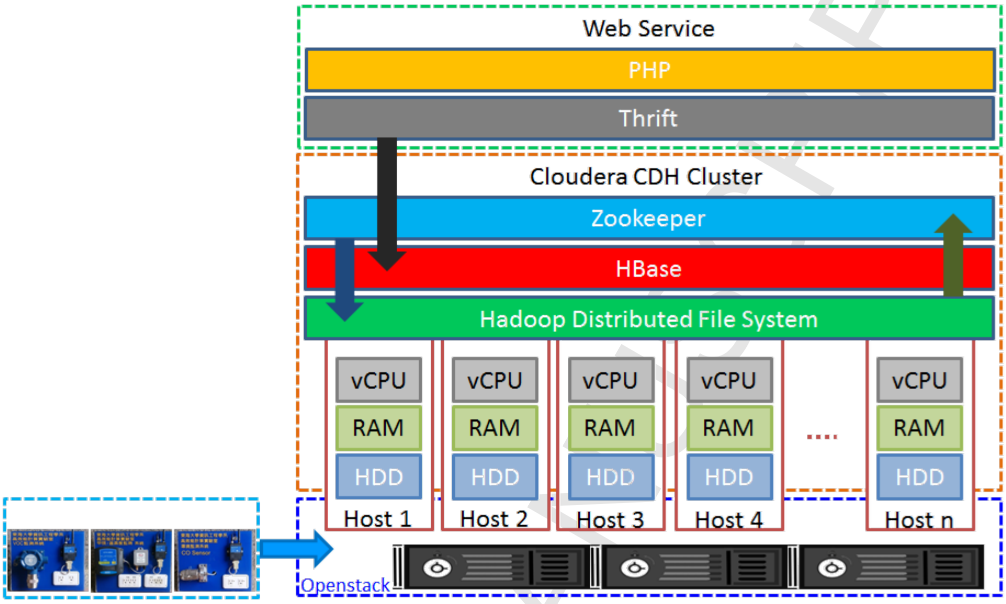


Figure 8: Cloud Architecture in our system

are written one-by-one in HBase. This method is applicable for small amount data or the case in which only a single data item is written at each time. When the amount of data becomes huge, too much time is called for data import. Another type is PutList for a multiple write operation, as shown in Figure 10. At first, a put list is established and pre-set with a writing limit; data are written to the list until it is full, then all the data in the list are written once into HBase. It saves the waste of time and computing resources of write operations performed each time after reading in a data item.

- **Data Collection Storage - Adjust**

  Since writing data into HBase consumes much time and resources, we adjust three parameters of HTable to improve the data import performance. The three parameters are setAutoFlush(), setWriteToWAL(), and setWriteBufferSize() which are described as follows.

  - setAutoFlush():

    When setAutoFlush of HTable is set as false, the automatic flush of HTable writing to Client is turned off; thus, data can be written

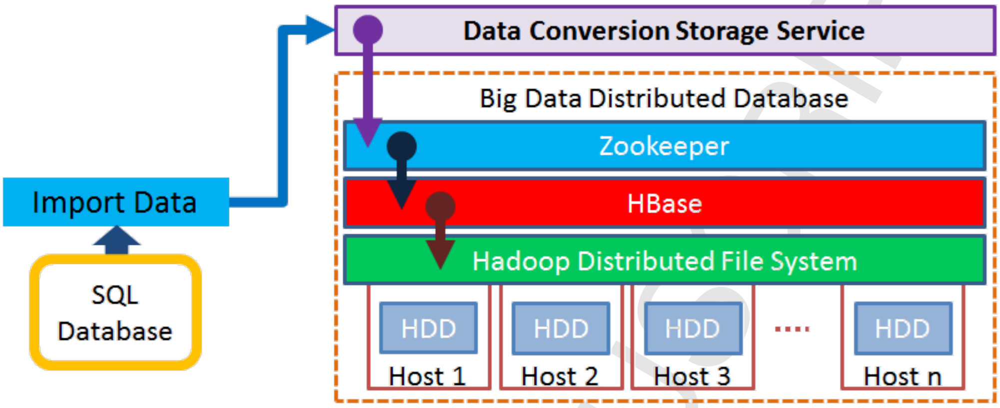


Figure 9: Data collection service with Put

into HBase in batch instead of writing one put per time by sending a request to RegionServer for update. Only when client's temporary storage is full, a request is sent to HBase for data import. The default of setAutoFlush is true.

– setWriteToWAL():

In HBase, when a client submits data to the RegionServer in the cluster, it first writes the "Write Ahead Log" (WAL), which is shared in all Regions in a RegionServer; and only when WAL write is successfully, it goes on writing MemStore, and then the client is notified with the successful submission of data. If WAL writing fail, the client is notified with the failed submission of data. The advantage of this approach is that the data can be recovered if the RegionServer clashes. Therefore, for writing less important data, one can adjust setWriteToWAL as false to give up writing WAL log and improve data import performance. The default of setWriteToWAL is true.

– setWriteBufferSize():

By adjusting WriteBufferSize in HTable, we can set the writing buffer size at the HTable's Client. If the new buffer size is less than the size of current data do writing into it, the data in the buffer will be flushed to the server. The unit of WriteBufferSize is in bytes; the writing buffer size can be set according to the realistic size of written data.

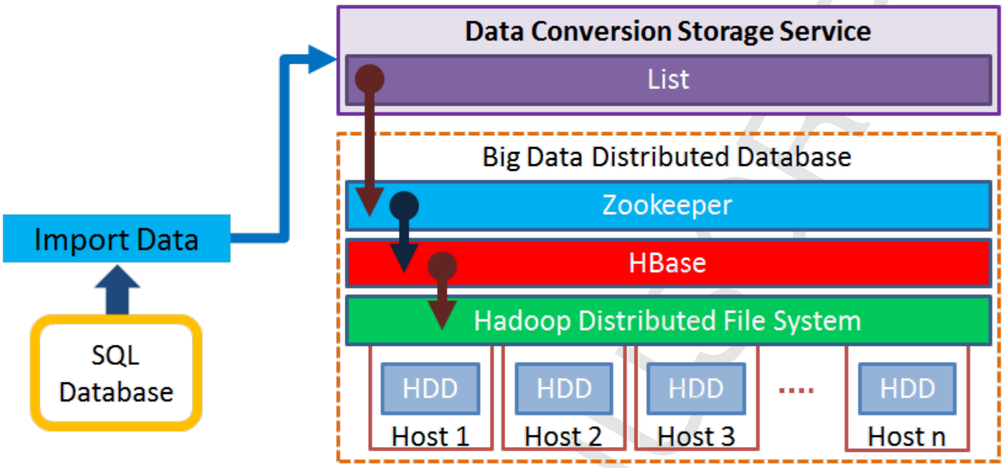


Figure 10: Data collection service with PutList

3.4.2. Web Application Service

Figure 11 shows data processing of various system services. The Web Application Service is responsible for communication with the front end, i.e., it receives service requests on stored data coming from the front end, and coordinates other system services such as Data Record Service and Data Analysis Service. Finally, the Web Application Service sends the query results back to the Front-end View to display.

3.4.3. Controlling Intelligent Sockets Service

The algorithm of transmission, processing, and analysis for real time sensory data is shown in Figure 12. Besides, display of the real time information in charts and intelligent sockets are used in iDEMS. The ZigBee router sends sensory data in hexadecimal control codes to the Coordinator, and then iDEMS determines whether the sensory data locate in a normal range. If the sensory data are not normal, iDEMS will give warning messages on the monitoring web page and/or trigger the intelligent socket to remotely control the system. Since iDEMS can be set in either "automatic control" or "manual control" mode, our system can be automatically controlled by the intelligent socket when there is no staff on site for real time monitoring of the environment.

Following algorithm is used for filter abnormal THI and show out system warning.

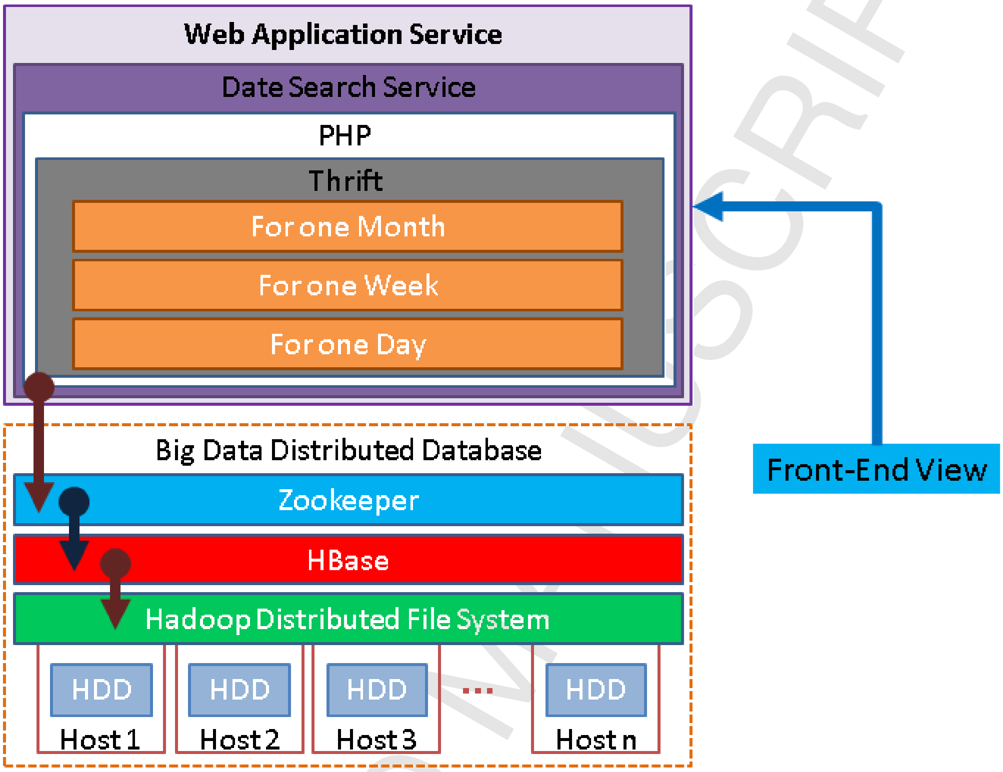


Figure 11: Data processing of system service

Algorithm 3.2: CONTROLLING INTELLIGENT SOCKET(c)

```
while in record time
get environment information via sensors
do
    if ( THI value is not normal )
    then
        if ( the socket is setting in automatic)
            then
                system warning
                turn on electrical products
        else if ( the socket is setting in manually)
            then
                system warning
    else continue;
```

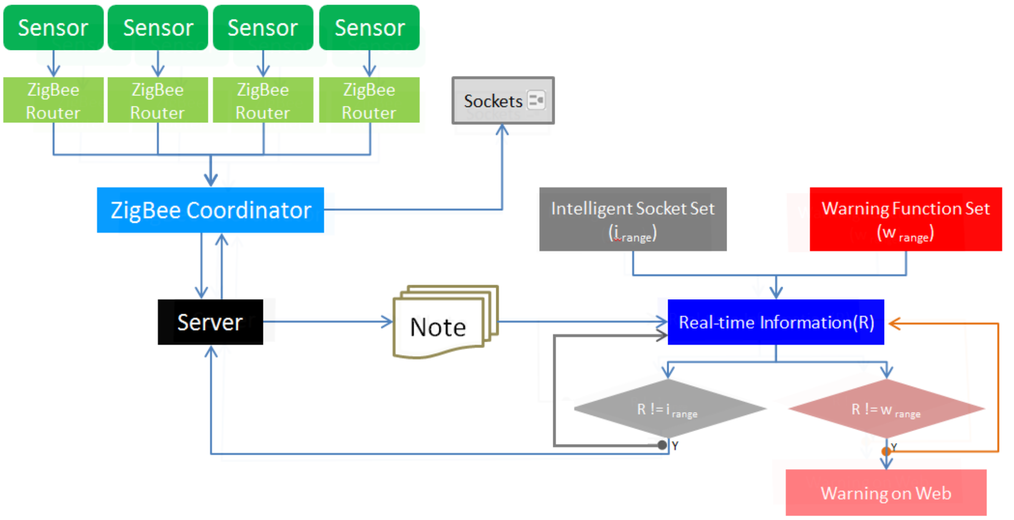
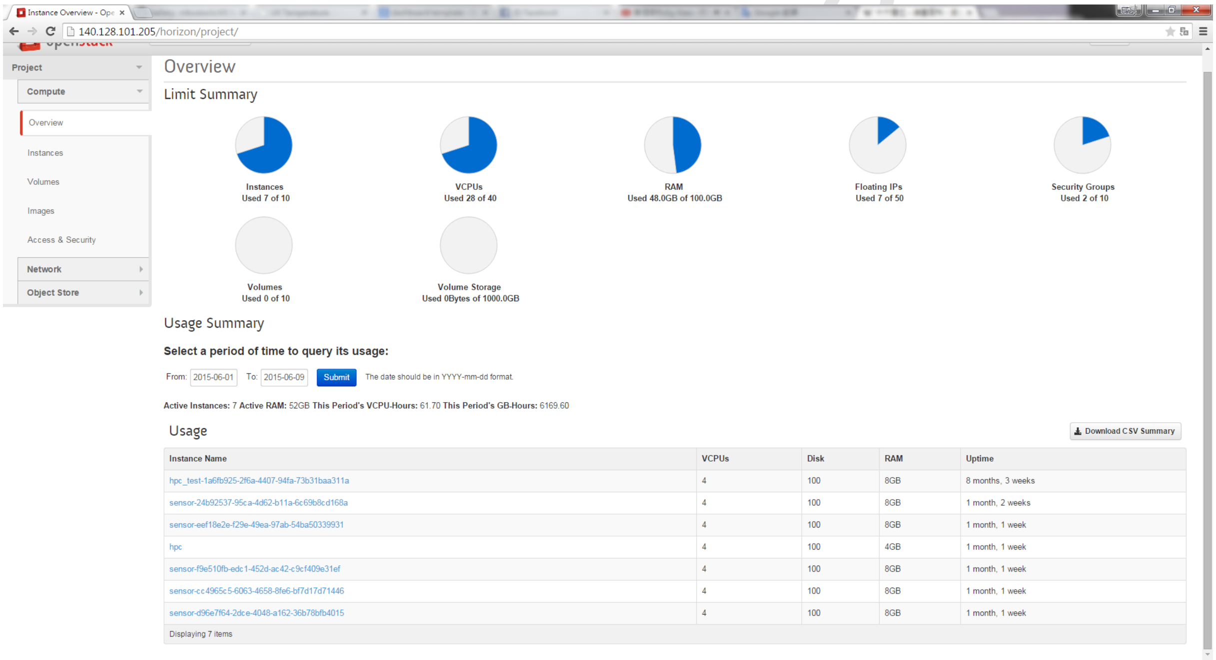


Figure 12: Flow chart of environmental data transmission

3.5. System Implementation

Since different data are collected by various environmental sensors, the original real time data are in different formats and coding. In this work, we used OpenStack to build four virtual machines (VMs) with same specifications to effectively use resources in the physical machines, since any physical machine with idled VMs could be turned off to save energy. Figure 13 depicts usage of VMs via a web interface on OpenStack.

We created multiple VMs to construct a cloud cluster in OpenStack. The Cloudera Manager was used to build related packages, such as ZooKeeper, Hadoop HDFS, Hadoop MapReduce, and HBase. To deploy the platform environment, we adopted Cloudera Manager to monitor status of system services, such as Hosts, HBase, HDFS, MapReduce, and ZooKeeper. The iDEMS also has real time monitoring of system service resources, such as CPU usage, Disk I / O, network I / O, and HDFS I / O in the cluster and displays states of real time system services on the Cloudera Manager interface, as shown in Figure 14. The Cloudera Manager can monitor the status of each node to validate normal connections of each host. This management system checks connections from time to time. If a connection is abnormal or in bad quality, or any cluster service or system problem occurs, the Cloudera manager will set up warning messages.



Select a period of time to query its usage:

From: 2015-06-01 To: 2015-06-09 The date should be in YYYY-mm-dd format.

Active Instances: 7 **Active RAM:** 52GB **This Period's VCPU-Hours:** 61.70 **This Period's GB-Hours:** 6169.60

Instance Name	VCPUs	Disk	RAM	Uptime
hpc_test-1a6fb925-2f6a-4407-94fa-73b31baa311a	4	100	8GB	8 months, 3 weeks
sensor-24b92537-95ca-4d62-b11a-6c69b8cd168a	4	100	8GB	1 month, 2 weeks
sensor-eef18e2e-f29e-49ea-97ab-54ba50339931	4	100	8GB	1 month, 1 week
hpc	4	100	4GB	1 month, 1 week
sensor-f9e510fb-edc1-452d-ac42-c9cf409e31ef	4	100	8GB	1 month, 1 week
sensor-cc4965c5-6063-4658-8fe6-bf7d17d71446	4	100	8GB	1 month, 1 week
sensor-d96e7f64-2dce-4048-a162-36b78bfb4015	4	100	8GB	1 month, 1 week

Displaying 7 items

Figure 13: OpenStack status

We used four hosts, one for NameNode and three for DataNodes. From status information the status of every machine in the cluster can be known. Each physical machine is configured with a static IP address. The detailed states of the whole system can be shown in a web page, as shown in Figure 15.

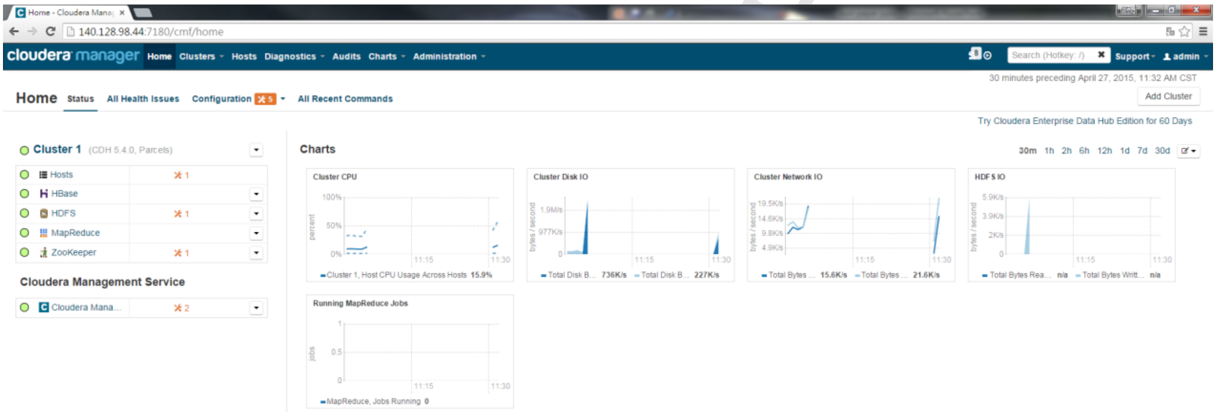
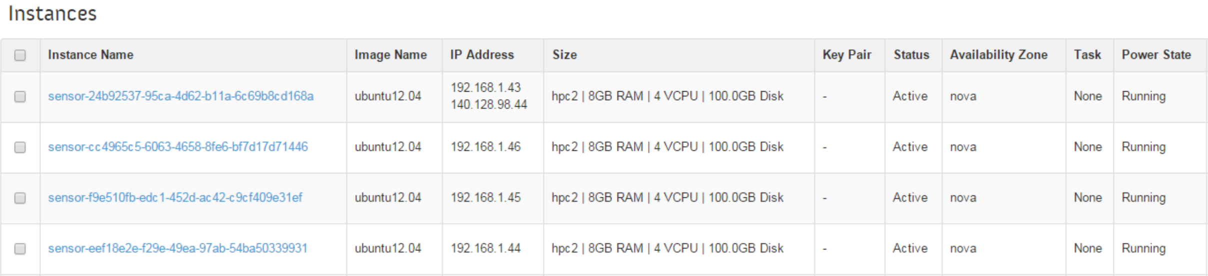


Figure 14: Cloudera manager status



Instances

☐	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State
☐	sensor-24b92537-95ca-4d62-b11a-6c69b8cd168a	ubuntu12.04	192.168.1.43 140.128.98.44	hpc2 | 8GB RAM | 4 VCPU | 100.0GB Disk	-	Active	nova	None	Running
☐	sensor-cc4965c5-6063-4658-8fe6-bf7d17d71446	ubuntu12.04	192.168.1.46	hpc2 | 8GB RAM | 4 VCPU | 100.0GB Disk	-	Active	nova	None	Running
☐	sensor-f9e510fb-edc1-452d-ac42-c9cf409e31ef	ubuntu12.04	192.168.1.45	hpc2 | 8GB RAM | 4 VCPU | 100.0GB Disk	-	Active	nova	None	Running
☐	sensor-eef18e2e-f29e-49ea-97ab-54ba50339931	ubuntu12.04	192.168.1.44	hpc2 | 8GB RAM | 4 VCPU | 100.0GB Disk	-	Active	nova	None	Running

Figure 15: System Test Content

4. Experimental Environment and Results

This experiment section is divided into two parts. The first is to explain the relevant hardware and software equipment used in building experimental environment. The second is our experimental results on the effectiveness of the adjustment.

4.1. Experimental Environment

To build platforms that provide services in the proposed iDEMS, we first introduce our hardware and software specification as follows. Table 1 shows the hardware specification of the computer and sensors we use in experiments. We implement the system in campus as a case study, therefore it can be implemented in other area in the future. In detail, we use two different kinds of environmental sensors in different type and specification of each sensor is listed in Table 1.

Table 1: Hardware for Sensing service

Hardware for Sensing service		
Sensor no.	Sensor Type	Specification
Sensor 1	Temperature and humidity	Series WHT
	Formaldehyde	CTX 300
	VOC	OLCT 100 XP
	Carbon monoxide	OLCT 20 D
Sensor 2	CO2, Temperature, Humidity	ZGw08VRC

In addition to the above ambient sensors in the cloud environment, we use one Physical machine and four virtual machines. The physical machine is with Intel(R) Core(TM)2 CPU6420@ 2.13GHz. The four virtual machines are built on OpenStack, whereas each virtual machines contain of 4 vCPU, so in total we have 16 vCPU. We use Ubuntu 12.04 64-bit version for operating system of all machines. It shows as Table 2. For the cloud cluster service, we use the Cloudera CDH to build cloud cluster. We used four VMs configured with the same specifications of the machine as an environment for processing Big Data and computing. Each VM is configured 4 cores, 8GB RAM and 100GB of storage space. The system software adopts Zookeeper 3.4.5, Hadoop2.6.0, HDFS2.6.0, and HBase1.0.0, as shown in Table 3.

Table 2: Hardware and Software for all service

Hardware and Software for all service		
Web Server *1	CPU	Intel(R) Core(TM)2 CPU6420@ 2.13GHz
	RAM	2GiB DIMM SDRAM Synchronous*2
	HDD	250GB Hitachi HDT72502*2
	OS	Linux Ubuntu 12.04.5 LTS
Cloud Server *1	vCPU	4 cores
	RAM	8GB
	HDD	100GB
	OS	Linux Ubuntu 12.04 LTS

Table 3: Hardware and Software for cloud service

Hardware for cloud service			
Node name	Cores	RAM	HDD
NameNode	4	8	100
DataNode 1	4	8	100
DataNode 2	4	8	100
DataNode 3	4	8	100
Software for cloud service			
	Version		
OS	Ubuntu 12.04 LTS		
Compiler	JavaSE Development Kit 7u15		
Cloudera	CDH 5.4.0		
ZooKeeper	3.4.5		
Hadoop	2.6.0		
HDFS	2.6.0		
HBase	1.0.0		

4.2. Experimental Results

To test the general case of data reading and writing, we use the Linux system. As shown in Figure 16, we measure the different ratio of data to compare the performance of Linux in the term of file size and processing time. We observe that the time it takes to read and write is higher, and therefore we have to analyze what information can be obtained. In order to efficiently analyze Big Data, we need to store data into HBase cloud platform. However, it spends a lot of time in converting a large amount of data to cloud platform. For input data, literally HDFS is one of the key factors that influence the effectiveness of import time. So we have a time effectiveness comparison for MySQL and HBase, as shown in Figure 17. We observe that HBase has better performance than MySQL. The results obtained after calculating the average is shown in Figure 18. As can be seen, the execution time of Put is much higher than PutList when executing a single writing operation to put a specified row key record into HBase. However, when the amount of data becomes huge PutList is a better method for multiple writing operation based on the experiment. In addition, AutoFlush adjustment can reduce more time it takes to write in PutList. It allows us to have a faster conversion, so we can avoid the long time conversion process and missing the important message. We integrate the above experimental data to the chart.

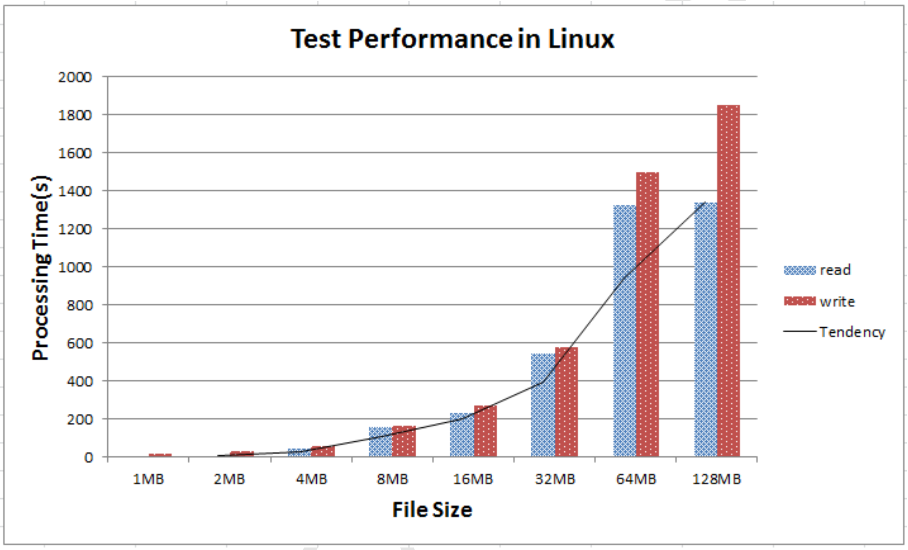


Figure 16: Test Performance in Linux

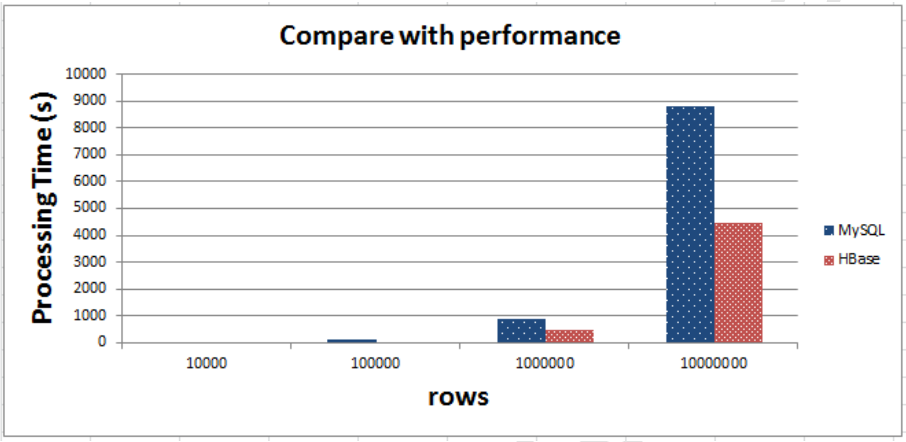


Figure 17: Compare With Performance

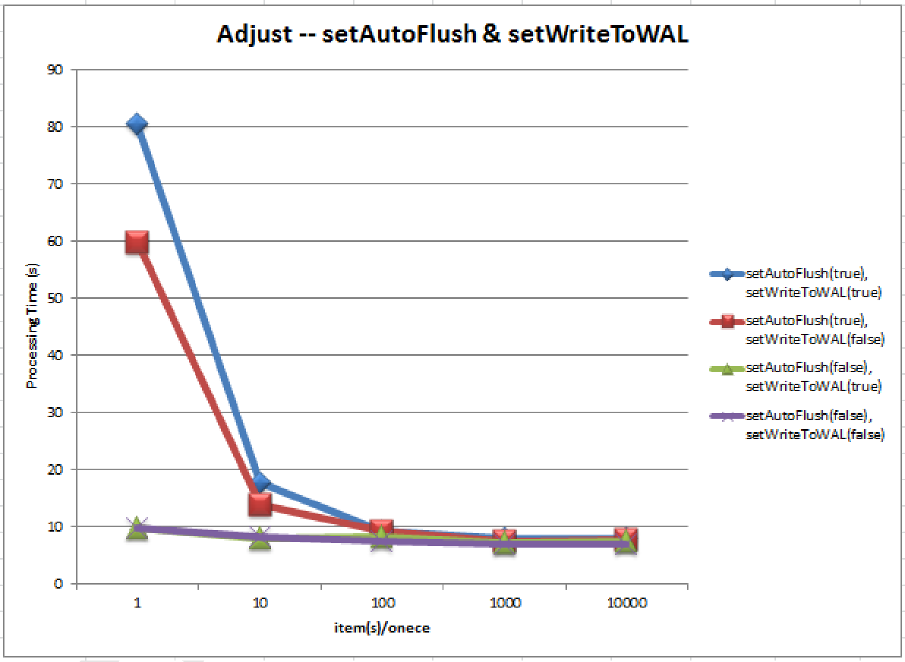


Figure 18: The execution time of Adjust in AutoFlush and WriteToWAL

It shows the Big Data cost processing time with reading and displaying in different amount of data. We know from Figure 19 obtained from the process of import data in MySQL and HBase, the more information would spend lots of time. Comparing data for one week and one month has a significant gap. We can imagine, how much time needed to reading the data for one month. Different from importing data process, to display the data in one month we can calculate it in one month average and show in one data row after it being calculated. Finally, we know that if we want to reduce the processing time, we must reduce the rows displayed. In Figure 27, Big Data accessing and processing platform is divided into several functions: sensor position, list of specific area in each position, real-time monitoring, search the environmental data, filter and analysis. The search function has three search modes, namely ”For one Day”, ”For one Week” and ”For one Month” searching modes. The rules are used for filtering the kind of sensor and the time setting option. All sensor positions in our campus are shown in a map view in Figure 20

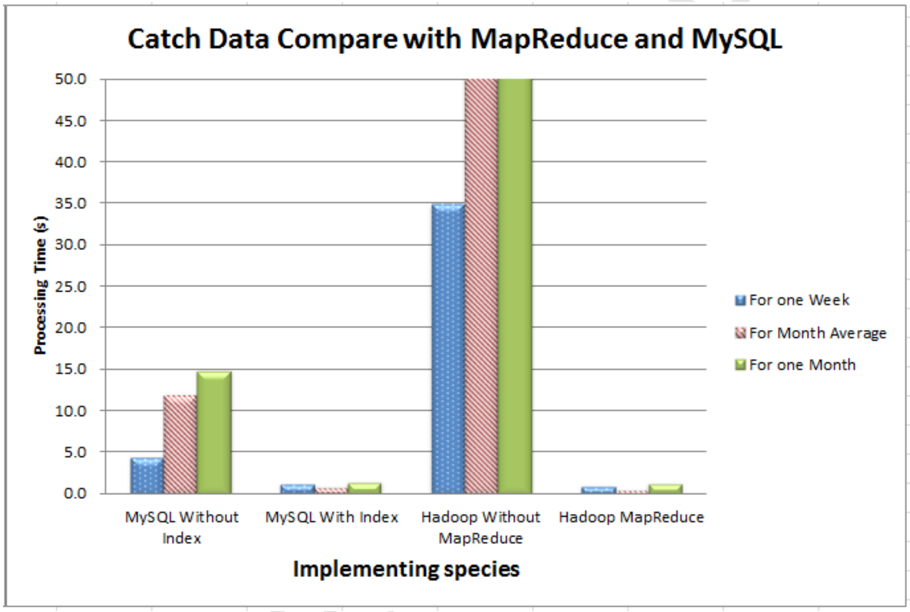


Figure 19: Compare with use Index in MySQL and use MapReduce in HBase

Inside the map we can click at a point of sensor link, then it will display the name of department with the blue url as shown in Figure 21. If we click this blue url, it will show the detail information of air quality in specific point of sensor as shown in Figure 22. We also can zoom in the table of air quality information as shown in Figure 23. Some other functions in this system are in daily, weekly and monthly informations of several air pollutant

地點	溫度	濕度	CO	甲醛	VOC
理學院樓	25.6	75	1	3	0
大智慧科技大樓	25.6	75	1	3	0
人文大樓	25.5	78	2	5	0.1
工學院	27.6	74	2	5	0.1
創意學院	25.7	76	2	5	0.1
校友會館	25.7	76	2	5	0.1
銘賢堂	21.6	74	3	5	0.1
學生福音中心	21.6	72	3	4	0.1
行政大樓	25.6	75	1	3	0
地點	溫度	濕度	CO2	甲醛	VOC
圖書館	24.35	76	650	null	null

Figure 20: Sensor positions in campus

parameters as mentioned previously. The reports can be seen in Figure 24 as the daily information, Figure 25 as the weekly information and Figure 26 as the monthly information.

Figure 22 shows the latest environmental information. It collects environmental information graphically presented on the web-based monitoring platform, contains the quality index of the current environment.

We can use this environmental information as an indicator to control and adjust the indoor temperature and humidity environment. THI has a large impact on human health, when THI is between 20 and 26 the human body feel comfortable and also it is a great circumstances to rest at night. On the other hand, if THI is high it is not conductive to fatigue recovery especially for a long time in a high THI indoor environment. Likewise, if THI is low then human body feels increasingly cold. This out of limits situation will affect significantly to the elderly and children health. In this paper we calculate the THI values and show the level of THI in color as shown in Figure 28. This following chart is ”Daily Statistics” as shown on Figure 29 This chart describes the graph of data changes in daily. We also add the calculation of THI value and external environment from sensors, such as sunny hours. This effective environmental information is determined by the environmental quality standards.

This monitoring system provides filter data function, whereas the user can select a specific data. For example, user can choose various kinds of

Sensor.png Sensor.bb

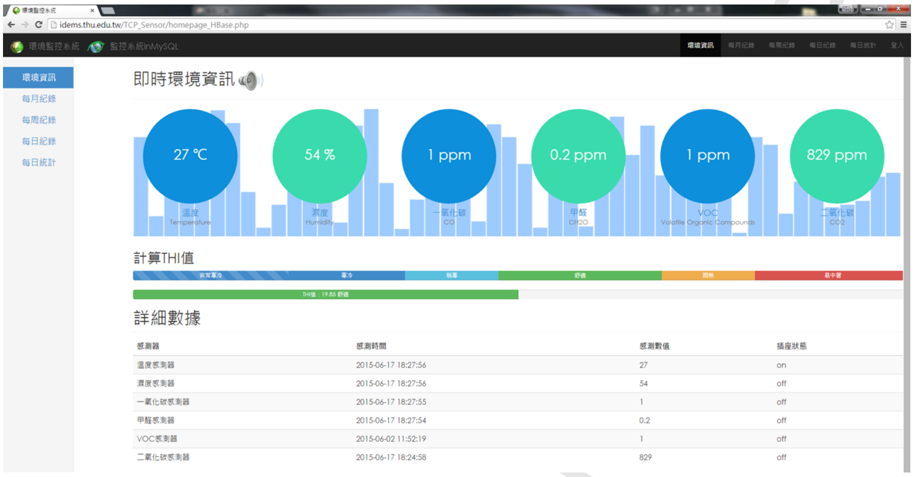
Figure 21: Specific Sensor

environmental sensors in a specific period of time. The result of the filter would be the calculation of the environmental value per month such as average, maximum and minimum value. The sample result is shown in the following Figure 30.

Indoor quality and safety are an important factor in the environmental monitoring system. In this work, in order to improve iDEMS, we add an intelligent socket regulation and alarm system that can enhance the management of indoor environment controlling system. Without this intelligent socket, the controlling system is operated by human. Whereas human operator error is can be a cause of information failures. To avoid this matter, we added the warning system feature that could automatically give warning when the air quality exceed the certain legal limit based on the air quality index rule from the authority through artificial approach.

There are many users that utilize this system, to securing this system, we add a login page to distinguish between users and manager who has permission to set alerts of the switch and socket. Sign in into management page could set socket switch and warning system. In this page, there are two parts of instructions and warning systems smart socket:

- Intelligent control socket



感測器	感測時間	感測數值	插座狀態
溫度感測器	2015-06-17 18:27:56	27	on
濕度感測器	2015-06-17 18:27:56	54	off
一氧化碳感測器	2015-06-17 18:27:55	1	off
甲醛感測器	2015-06-17 18:27:54	0.2	off
VOC感測器	2015-06-02 11:52:19	1	off
二氧化碳感測器	2015-06-17 18:24:58	829	off

Figure 22: Real-time data

In the iDEMS, we use the intelligent socket to replace the traditional socket to provide further range to connect this system control. This intelligent socket not only can save a lot of unnecessary energy consumption, but also can give a warning automatically. There are two kinds of "socket set" controlling, i.e., "manual control" and "automatic control". When administrator selects "Manual control", they can manually monitoring the environmental system and do the right steps based on their decisions if air quality value is out of the limits. Conversely, if the setting is "automatic control", administrator must set the threshold value at first. When the value exceeds the threshold the socket would automatically give a warning. The following Figure 31 is an experiment test when the socket is set as "automatic control". A warning lamp is connected on the system, when the air quality is poor then a warning message will appear in the monitoring screen and the lamp will turn on. This intelligent socket also can connect to a variety device to remind the operators when the quality of air is in worse status.

- Alarm System
  Warning system can be set through management and setting system interface allowing the monitoring operator notices environmental air quality in graphical mode. When receives abnormal value as seen in monitor, there would be a warning sound and flashing icons at the top

地點	溫度	濕度	CO	甲醛	VOC
理學院樓	26.1	71	1	1	0
大智慧科技大樓	26.1	71	1	1	0
人文大樓	26	74	2	3	0.1
工學院	28.1	70	2	3	0.1
創意學院	26.2	72	2	3	0.1
校友會館	26.2	72	2	3	0.1
銘賢堂	22.1	70	3	3	0.1
學生福音中心	22.1	68	3	2	0.1
行政大樓	26.1	71	1	1	0
地點	**溫度**	**濕度**	**CO2**	**甲醛**	**VOC**
圖書館	23.35	68	677	null	null

Figure 23: Air Quality Value

of the homepage.

In Figure 32, during the student final exam week, we run the test environment at 8 AM to 23 PM to capture the indoor air quality in the room. We learn from Figure 32 after the opening time, CO2 value was gradually increased. We did the setting in two-stage warning, the first stage of the setting value was 1200 ppm, the indoor air-conditioner became stronger but it still no warning. The air-conditioner only let carbon dioxide concentration was spread to every corner of the environment, and it is cannot effectively reduce the air value of indoor carbon dioxide. After a period of time, carbon dioxide concentration was getting rise again. Until the value reach 1600 ppm, in this second stage, a warning sound was triggered to remind environmental managers to improve the environment air quality. In this experimental phase, to balancing the air quality the action to take was opening the windows to let the amount of fresh air brought indoors, and after that the results

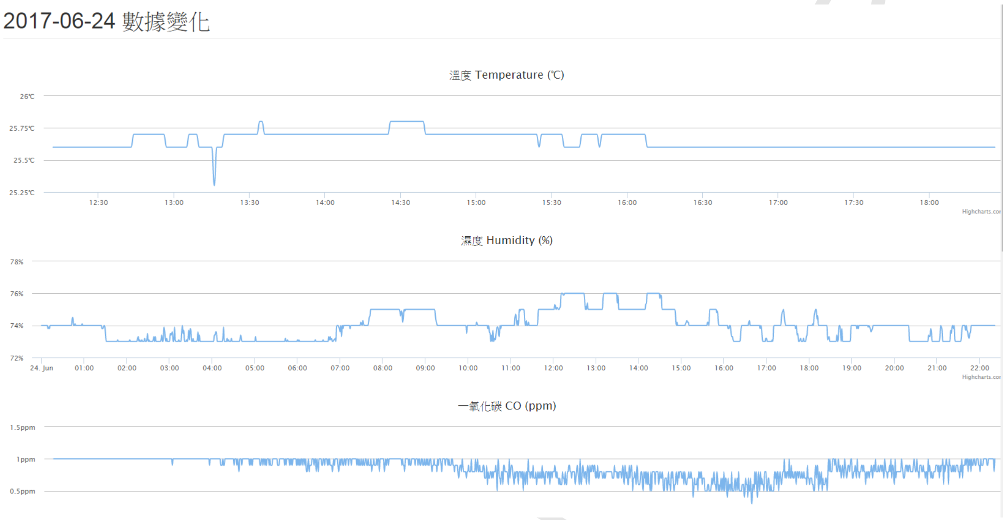


Figure 24: Daily information of Science and Technology Building

can be obtained from the Figure 32, as can be seen the carbon dioxide concentration in the environment reduced effectively. From this experiment, we can learn that iDEMS in indoor monitoring system is beneficial to monitor air circulation. The authority can take an action, such as opening the windows to balance the fresh air outdoor and dirty air indoor when the warning system is triggered. But of course the outdoor air quality must be in a good conditions. It means that, there is a domino effect in this case. If we want to provide a healthy air quality indoor, such as at home, in the workplace, in the classroom or other buildings, we must have a healthy outdoor air quality also.

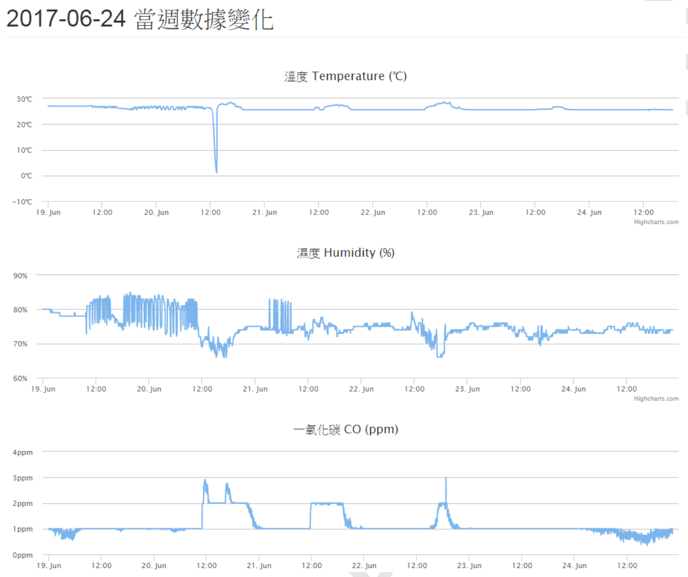


Figure 25: Weekly information of Science and Technology Building

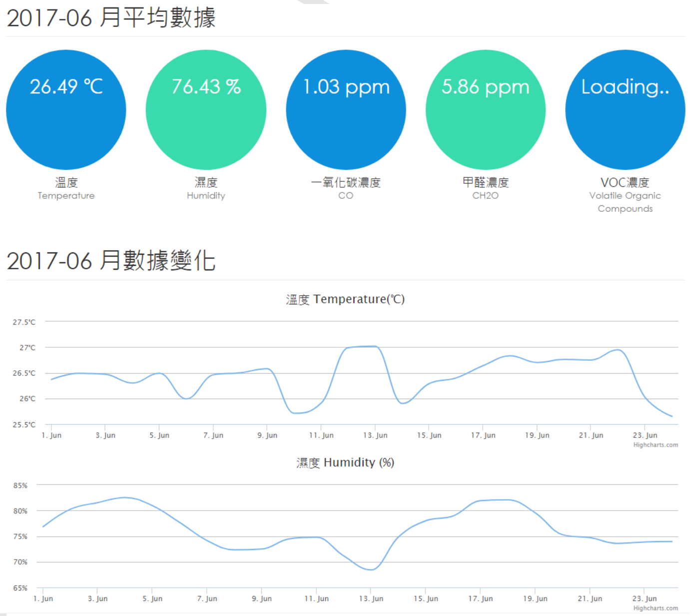


Figure 26: Monthly information of Science and Technology Building

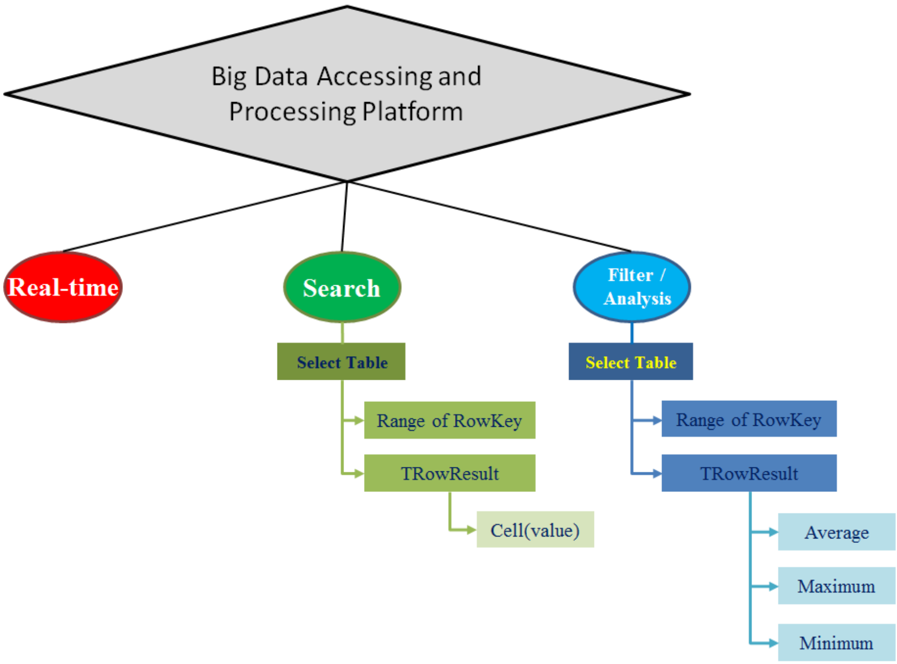


Figure 27: Big Data Accessing and Processing Platform

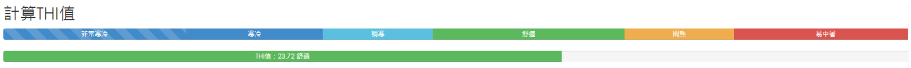


Figure 28: The evaluate result of environment quality

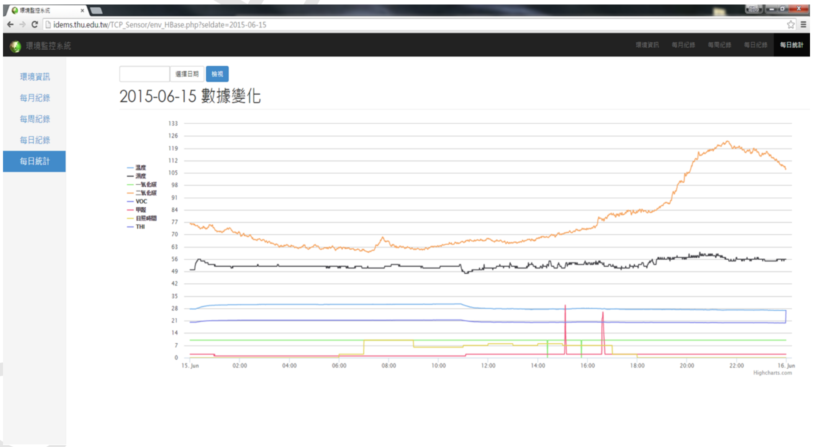


Figure 29: Record data per day for all environment data

idems.thu.edu.tw/TCP_Sensor/Filter_HBase.php

2015-06 數據分析

Environment_Temperature

29.03 ℃
本月平均
Average

32.16 ℃
本月最高
Max.

23.71 ℃
本月最低
Min.

Figure 30: Record data analysis for all environment data

Figure 31: Practical test with intelligent socket

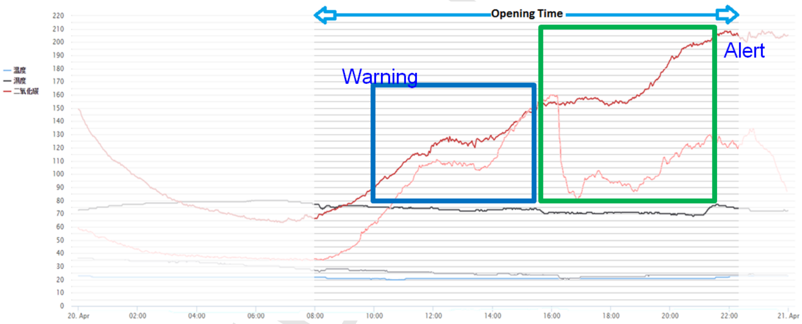


Figure 32: Practical test with warning

5. Conclusions and Future Work

In the term of data collection, we implement IoT for collecting the data of air quality in real-time. We use Zigbee wireless sensor network technology which is has many beneficial features like low-speed, low power, low cost, supporting a large number of network nodes and variety of network topologies, and its application is simple, fast, reliable and secure. For storing data, we use HBase to process environmental data captured by sensors. With the distributed cloud architecture, HBase is capable to store a large amount of data, i.e., Big Data. In addition, Hadoop MapReduce with distributed computing architecture is used for processing and displaying of Big Data. One important feature of the cloud platform is that it stores the sensory data into the system fast without interrupting.

In the term of data processing, we build an iDEMS using OpenStack, which is it can provide the standard hardware of cloud computing service we developed. With OpenStack, we can manage the operations of the compute module, networking module and storage module for our system. We also embed our system with the intelligent socket which is proven to give an alert when the level of air quality is out of the threshold based on THI value. In addition, in this work we compare the time spent for storing data into HBase using different methods. Moreover, we also tune up the system by identifying the most suitable data import method according to the experimental results to accelerate the data input speed for the cloud platform.

In the term of information monitoring, we use Thrift to connect back-end and front-end, process HBase data, and use Java code as a client to create the user interface in our system. With Thrift, user can use RowKey to quickly get corresponding information of the row, and in a search process, use RowKey Range to specify the start and end of RowKey to quickly obtain data within the range. Thus, by above two RowKey functions, this work can efficiently access data. We also provide a filter function for data which is user can filter the information according to time intervals, and find the average and trends of data based time interval.

Finally, in the future we hope to add more environmental monitoring spots to comprehensively monitor the whole iDEMS. We also hope to add more analysis capabilities in the cloud system to promote personnel health and extend the effectiveness of our study to the application fields and technical views fully.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology, Taiwan ROC, under grants number 104-2221-E-029-010-MY3, 105-2622-E-029-002-CC3, and 106-3114-E-029-003.

References

[1] K. K. Nguyen, M. Cheriet, M. Lemay, V. Reijs, A. Mackarel, A. Pastrama, Environmental-aware virtual data center network, Computer Networks 56 (2012) 2538–2550.

[2] M. Rönkkö, J. Heikkinen, V. Kotovirta, V. Chandrasekar, Automated preprocessing of environmental data, Future Generation Computer Systems 45 (2015) 13–24.

[3] S. Trilles, A. Calia, Óscar Belmonte, J. Torres-Sospedra, J. Huerta, Deployment of an open sensorized platform in a smart city context, Future Generation Computer Systems 76 (2017) 221–233.

[4] D. Meana-Llorián, C. G. García, B. C. P. G-Bustelo, J. M. C. Lovelle, N. Garcia-Fernandez, Iofclime: The fuzzy logic and the internet of things to control indoor temperature regarding the outdoor ambient conditions, Future Generation Computer Systems 76 (2017) 275–284.

[5] C.-T. Yang, C.-J. Liao, J.-C. Liu, W. Den, Y.-C. Chou, J.-J. Tsai, Construction and application of an intelligent air quality monitoring system for healthcare environment, Journal of Medical Systems 38:15 (2014).

[6] T. Kun-Fu, Bridge apache hbase and relational database operation mechanism design and implementation based on php and mysql application, 2012. `https://etds.stu.edu.tw/etdservice/download_file?etdun=U0084-2908201210312100&fileName=stu-101-s99639102-1.pdf`.

[7] S. S. Manvi, G. K. Shyam, Resource management for infrastructure as a service (iaas) in cloud computing: A survey, Journal of Network and Computer Applications 41 (2014) 424–440.

[8] N. Sultan, Discovering the potential of cloud computing in accelerating the search for curing serious illnesses, International Journal of Information Management 34 (2014) 221–225.

[9] J. Liu, X. Huang, J. K. Liu, Secure sharing of personal health records in cloud computing: Ciphertext policy attribute based signcryption, Future Generation Computer Systems 52 (2015) 67–76.

[10] Openstack.org, 2015. `https://www.openstack.org/`.

[11] A. Corradi, M. Fanelli, L. Foschini, Vm consolidation: A real case based on openstack cloud, Future Generation Computer Systems 32 (2014) 118–127.

[12] J. P. Molina, J. M. L. Vega, J. M. L. Soler, A. Corradi, L. Foschini, Dargos: A highly adaptable and scalable monitoring architecture for multi tenant clouds, Future Generation Computer Systems 29 (2013) 2041–2056.

[13] E. Feller, L. Ramakrishnan, C. Morin, Performance and energy efficiency of bigdata applications in cloud environments: A hadoop case study., Journal of Parallel and Distributed Computing 79–80 (2015) 80–89.

[14] Apache hbase, 2015. `http://hbase.apache.org/`.

[15] Apache hadoop, 2015. `http://hadoop.apache.org/`.

[16] Q. Ashraf, M. HadiHabaeb, Autonomic schemes for threat mitigation in internet of things, Journal of Network and Computer Applications 49 (2015) 112–127.

[17] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.

[18] WissamRazouk, G. V. Crosby, AbderrahimSekkaki, New security approach for zigbee weaknesses, Procedia Computer Science 37 (2014) 376–381.

[19] Z. ya Liu, Hardware design of smart home system based on zigbee wireless sensor network, AASRI Procedia 8 (2014) 75–81.

[20] Honeywell, 2015. `http://honeywell.com/Pages/Home.aspx`.

[21] N. Maheshwari, R. Nanduri, V. Varma, Dynamic energy efficient data placement and cluster reconfiguration algorithm for mapreduce framework, Future Generation Computer Systems 28 (2012) 119–127.

[22] C.-T. Yang, J.-C. Liu, S.-T. Chen, H.-W. Lu, Implementation of a big data accessing and processing platform for medical records in cloud, Journal of Medical Systems 41:149 (2017).

[23] C.-T. Yang, W.-C. Shih, L.-T. Chen, C.-T. Kuo, F.-C. Jiang, F.-Y. Leu, Accessing medical image file with co-allocation hdfs in cloud, Future Generation Computer Systems 43–44 (2015) 61–73.

[24] H. Lee, B. Shao, U. Kang, Fast graph mining with hbase, Information Sciences 315 (2015) 56–66.

Chao-Tung Yang is Distinguished Professor of Computer Science at Tunghai University in Taiwan. He received his Ph.D. in computer science from National Chiao Tung University in July 1996. In August 2001, he joined the Faculty of the Department of Computer Science at Tunghai University. He is serving in a number of journal editorial boards, including International Journal of Communication Systems, KSII Transactions on Inter- net and Information Systems, Journal of Applied Mathematics, Journal of Cloud Computing. He has published more than 280 papers in journals, book chapters and conference proceedings. His present research interests are in cloud computing and service, big data, parallel computing, and multicore programming. He is a member of the IEEE Computer Society and ACM.

Shuo-Tsung Chen received the B.S. degree in Mathematics from National Cheng Kung University, Tainan in 1996 and M.S. degree in Applied Mathematics from Tunghai University, Taichung in 2003, Taiwan. In 2010, he received the Ph.D. degree in Electrical Engineering from National Chinan University, Nantou, Taiwan. Now he is an Adjunct Assistant Professor in Department of Electronic Engineering, National Formosa University and Department of Applied Mathematics, Tunghai University.

Walter Den received the B.S. degree, M.S. degree, and PhD in Civil and Environmental Engineering, University of Southern California, USA. Now he is a Professor in Department of Environmental Science, Tunghai University, Taichung, Taiwan.

Yun-Ting Wang received the B.S. degree in Computer Science from Tunghai University, Tainan in 2013 and M.S. degree in Computer Science from Tunghai University, Taichung in 2015, Taiwan. Now she is a Data Engineer, Wanin International, Taichung, Taiwan.

Endah Kristiani received the B.S. degree in Informatics Management and Computer Engineering from Institut Sains dan Teknologi Akprind, Yogyakarta in 1998, Indonesia and M.S. degree in Electrical Engineering (Information Technology) from Universitas Gadjah Mada, Yogyakarta in 2007, Indonesia. Now she is a PhD student in Department of Industrial Engineering and Enterprise Information, Tunghai University, Taichung, Taiwan.

Highlight:

- This work proposes an Intelligent Indoor Environment Monitoring System (iDEMS).
- The proposed system combines environmental sensors with ZigBee wireless sensor network technology.
- The proposed system stores and processes environmental data in HBase.
- The proposed system presents the resulting information by a web-based Monitoring Platform.