SPECIAL SECTION ON INTELLIGENT DATA SENSING, COLLECTION AND DISSEMINATION IN MOBILE COMPUTING

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 5, 2019, accepted August 15, 2019, date of publication September 5, 2019, date of current version September 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939614

An Edge Computing Platform for Intelligent Operational Monitoring in Internet Data Centers

CONGFENG JIANG[1,2], YELIANG QIU[1,2], HONGHAO GAO[3], TIANTIAN FAN[1,2], KANGKANG LI[1,2], AND JIAN WAN[4]

[1]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China
[3]Computing Center, Shanghai University, Shanghai 200444, China
[4]School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Corresponding author: Honghao Gao (gaohonghao@shu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61972118 and Grant 61972358, and in part by the Key Research and Development Program of Zhejiang Province under Grant 2018C01098, Grant 2019C01059, and Grant 2019C03134.

ABSTRACT The increasing demand for cloud-based services, such as big data analytics and online e-commerce, leads to rapid growth of large-scale internet data centers. In order to provide highly reliable, cost effective, and high quality cloud services, data centers are equipped with sensors to monitor the operational states of infrastructure hardware, such as servers, storage arrays, networking devices, and computer room air conditioning systems. However, such coarse grained monitoring cannot provide fine grained real time information for resource multiplexing and job scheduling. Moreover, the monitoring of node level power consumption plays an important role in the optimization of workload placement and energy efficiency in data centers. In this paper, we propose an edge computing platform for intelligent operational monitoring in data centers. The platform integrates wireless sensors and on-board built-in sensors to collect data during the operation and maintenance of data centers. Using logical functions, we divide the data center clusters into grids, and then deploy wireless sensors and edge servers in each grid. As such, data processing on edge servers can reduce the latency in data transmission to central clouds and thereby enhance the real time resource mapping decisions in data centers. In addition, the proposed platform also provides predictions of resource utilization, workload characteristics, and hardware health trends in data centers.

INDEX TERMS Edge computing, data centers, monitoring, energy efficiency, intelligent operation.

I. INTRODUCTION

The cloud computing paradigm provides pervasive services to end users, such as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). The cloud services are usually hosted in large scale powerful internet data centers. The increasing demands for cloud-based services, such as big data analytics and online e-commerce, lead to a rapid growth of large-scale Internet Data Centers (IDCs). Global internet traffic in IDCs is expected to increase four times from 2014 to 2019, with a compound annual growth rate (CAGR) of 25% [2]. According to the report from the Lawrence Berkeley National Laboratory, energy consumption in US data centers will increase to 73 billion kilowatt-hours by 2020 [3]. Moreover, in each year, data centers cost \$13 billion in electrical bills and emit nearly 100 million tons of carbon pollution [4]. Among the huge energy consumption of global data centers, one major source of energy consumption in data centers is the energy consumption of cooling systems (∼37% [5]). However, due to the lack of access to data center wide monitoring and analytics tools, there exist many idle server capacities in data centers. In other words, the data center operators don't have enough visibility into what resources are actually being consumed by applications [6].

Energy consumption in data centers can lead to thermal management, cost, and system reliability issues. Previous works on energy consumption are mostly based on an approximated system model, which usually incorporates first-order effects of thermal, electrical, and mechanical principles [7]–[11]. However, in a typical modern data center, hardware devices are highly heterogeneous since they

The associate editor coordinating the review of this manuscript and approving it for publication was Anfeng Liu.

are replaced gradually generation by generation. Since the efficient operation and maintenance of data centers have various requirements, such as monitoring the environment in the data center, monitoring server resources, and scheduling the task efficient, the increase in the complexity of operation and maintenance of data centers results in the problem of intelligent monitoring of data centers, in both industry and academia. Researchers and practitioner have designed several protocols to address network congestion and safety issues [12]–[17]. As the operational costs of data centers and the demands for high-quality services increase, cloud service providers seek to operate and maintain IDCs with the joint consideration of service quality, environment of clusters, power consumption, and performance of the clusters.

The operators of IDCs need the monitoring of the real-time running status of all clusters in IDCs and the environment of cooling systems. They deploy wireless sensors as well as monitoring software based on built in sensors on server motherboards to monitor the data center's operating conditions including temperature, humidity, airflow, etc. System monitoring can provide information for data center health, energy efficiency, and jobs scheduling performance. Specifically, system monitoring can help make better decision for both data center operation itself and service provisioning in the long run by providing data center information of different granularity ranging from temperatures of CPU and memory modules to computer room humidity. For example, the cluster management system (CMS) in the data center may leverage the temperature distribution and energy consumption of servers to make temperature and energy aware workload placement, job scheduling, and workload migration [18]. Moreover, infrastructure monitoring also provides insights into data center capacity use, bottlenecks, and access attempts. Therefore, by using sensors, the servers and the data center are able to perceive their context, and via built-in networking capabilities they would be able to communicate with each other, access Internet services and interact with system administrators.

By mining the information of operating characteristics of servers, tasks on IDCs can be more efficiently scheduled and thereby power consumption of IDCs can be reduced if we consider both energy consumption of servers and workloads characteristics. Currently, existing monitoring methods for data centers usually consider a single control knob, such as IDC environment itself (temperature, humidity, and fire/smoke) and server resources (CPU, Mem, I/O, networking and so on). Data centers are equipped with sensors to monitor the operating states of hardware, such as servers, storage arrays, networking devices, and CRAC (Computer Room Air Conditioning) systems. However, such coarse-grained monitoring cannot provide fine-grained information for resource multiplexing and job scheduling. Therefore, data centers need an integrated monitoring system that can monitor all relevant information in the data center, such as power consumption in rack servers, energy consumption in cabinets, environment of the data center, and server resources. Given all the relevant information in the IDC, we can then schedule the tasks efficiently by mining and analyzing the collected information.

Since more and more enterprises and individuals are migrating from traditional IT systems to cloud based services, the constructions and operations of internet data centers are continually increasing in the last decades. In order to provide high performance and highly reliable computing capabilities, the data centers are running on dedicated infrastructure hardware such as high performance processors, memory modules, and large scale massive storage arrays. Moreover, cloud data centers provide massive data processing and analysis services using centralized database or distributed MapReduce based streaming data analytics. Although cloud computing can provide organizations dynamic, cloud-based operating models for cost optimization and increased competitiveness, it also has some disadvantages in many scenarios like industrial IoT(Internet of Things), connected autonomous vehicles (CAVs), smart homes, and smart cities. For example, cloud computing based processing requires huge volume of data transportation from end devices and sensors, which consumes large network bandwidth. Moreover, cloud data center based analysis is not possible for huge data generated from thousands of millions of end devices due to the incapability of computing and storage. Therefore, cloud computing based processing can't provide prompt responsiveness and short latency for big data analytics from massive IoT devices. Moreover, in some scenarios where data privacy and security is the first concern, cloud computing data centers are not trustful to conduct the data analytics. In contrast, data privacy preserving requires that the data is processed near its source, other than in the remote cloud data centers.

Edge computing is emerged as a promising paradigm that provides capabilities of processing or storing critical data locally and pushing all received data to a central data center or cloud storage repository [19]. For example, in IoT use cases, the edge devices collect data from sensors and process it there, or send it back to a data center or the cloud for processing if the local processing power is not enough. Therefore, this paper proposes an edge computing platform for intelligent operational monitoring in IDCs. This platform can integrate various types of monitoring data, process the data from sensors in real-time, store the data permanently, and map the data for system administrators to analyze. Based on machine learning algorithms, it can predict the operating characteristics of data centers, including trends in the health of hardware, operational quality of workloads, and forecasting of future workloads. Moreover, this platform provides decision-making for intelligent operation and maintenance in IDCs.

The reminder of this paper is organized as follows. In Section 2, we introduce the background techniques and related work for data center monitoring. Then we introduce the architecture design of our intelligent data center monitoring system in section 3. In Section 4 we elaborate the main functions of the monitoring platform. We evaluate and explain the prototypical implementation in section 5 and provide experimental measurements in section 6. Finally, we summarize our work in section 7.

II. RELATED WORK

Large scale data centers generate a significant amount of heat that may compromise their integrity and reliability for supplying uninterrupted service. Therefore, sensor networks are widely deployed in large scale data centers for health monitoring and infrastructure maintenance. A sensor network may consist of a large number of small sensor nodes that are composed of sensing, data processing and communicating components. With the assistance of wireless sensor, it is possible to estimate the power draw in a rack and then to estimate if it is possible to insert new equipment, plan space for additional servers and arrays or rebalance power loads. For example, in large scale data centers, mobile sensor networks can be deployed for data sensing, aggregation, and transmission during emergencies when there exists local sensor network congestion or system failure. Moreover, wireless sensors can alleviate the stress of maintaining the environmental component of data center integrity. However, in mobile sensor network, a continuous and stable connection between a pair of mobile sensors is usually unavailable due to high mobility. Therefore, edge computing assisted solution provides low latency for real time communications between mobile sensors nodes, which can maximize the probability of sending data to the destination with tolerance of delays.

With the increase of the scale of the data center, the amount of data generated by IDCs and the demands for data processing are increasing. The information of tasks and resources in IDCs can help us design workload placement and job scheduling strategies to reduce energy consumption in the data centers [20]. For example, the work in [6], [21]–[23] performed task scheduling by analyzing the information of server resources and the characteristics of applications. Existing cluster resource monitoring systems aim to detect and monitor thousands of nodes [24]–[27]. For instance, *ganglia* [28] is an open-source cluster monitoring software that can collect data of all clients on the same network domain. The design of this architecture indicates that a server can manage tens of thousands of servers in hierarchical network topology. Similarly, *cacti* also has a set of graphic analysis tools to monitor network traffic based on software stack of *php*, *mysql*, *snmp*, and *rrdtool*. *Cacti* collects data by *snmpget* and draws graphs by *rrdtool*. Combined with LDAP (lightweight directory access protocol), *Cacti* can verify users and add templates by itself. When users request the data, *rrdtool* produces icons and then provides them to users. With a built-in information transmitting system, *cacti* can immediately inform the operation and maintenance administrators in case of an emergency. Therefore, mining of monitoring information based on machine learning algorithms can also help reduce energy consumption in data centers.

The above-mentioned monitoring systems only take IT resources into account without considering data center environment (i.e., temperature, humidity) and the energy consumption of servers. Such limitation makes it impossible to provide complete operating and healthy information for the cloud services provider when they want to make a decision by joint considering the energy consumption, performance, and environmental constraints of the clusters and QoS guarantees. Besides, the service location and selection also play crucial roles in monitoring system [29], [30]. And the QoS of data sensing and service aggregation are also important factors in resource monitoring workflow, which can influence the overall efficiency of a monitoring system. For example, techniques such as locality-sensitive hashing (LSH) VM migration, queuing delay utilization scheme optimization, and statistical approaches are proposed [31]–[34]. In sensor networks, data aggregation scheme is one of the most important protocols that affect the data collection performance and system reliability. Moreover, physical routing can also help the monitoring coverage and data collection efficiency. Directional sensors are also used in current data centers to detect environmental conditions such as server noise.

As an emerging platform, edge computing has gained widespread attention in recent years. Edge computing is emerged as a promising paradigm that provides capabilities of processing or storing critical data locally and pushing all received data to a central data center or cloud storage repository. Since data is increasingly being generated at the edges of networks, processing data at the edge of the network is also more important for some latency critical applications [35]–[37], has been introduced into the community. Edge computing is the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services. For example, Yi *et al.* [38] built a proof-of-concept platform to run a face recognition application, reducing the response time from 900ms to 169ms by moving computing from the cloud to the edge.

The aim of edge computing is to unify the scattered resources that close to each other in spatial or network distance to provide applications with the services of computing, storage and network. When it comes to the monitoring of IDCs, we make the whole monitoring platform easier to extend by introducing edge computing. By processing all kinds of information on edge nodes, the real-time performance of the platform can be achieved. At the same time, the operation and maintenance personnel can immediately observe the information and make the task scheduling decision intelligently through analysis. To this end, we propose an edge computing platform for intelligent operational monitoring in data centers. The platform integrates wireless sensors and on-board built-in sensors to collect data during the operation and maintenance of data centers. Using logical functions, we divide the data center clusters into grids, and then deploy wireless sensors and edge servers in each grid. Therefore, data processing on edge servers can reduce the latency in data transmission to central clouds and thereby enhance the real time resource mapping decisions in data centers. In the following sections we will elaborate the architecture design and its prototypical implementation of our proposed monitoring platform.

III. ARCHITECTURE DESIGN

Operating an enterprise level data center requires accurate real-time measurements of temperature, humidity, and airflow, as well as detailed inventories of IT and non-IT equipment characteristics, and performance. Currently, most data center infrastructure management (DCIM) tools deliver this information using sensors deployed throughout the data center and each facility. However, for an individual data center, the DCIM software must be carefully configured and customized to be most effective in any particular application. In this paper, we propose an intelligent operation and maintenance monitoring framework based on an edge computing platform, which is used to collect information of IDCs and make intelligent operation and maintenance decisions based on related information. This paper proposes an edge computing platform for data center operating monitoring, including infrastructure and component level resource monitoring, such as server rack power and thermal monitoring, CPU resource utilization, memory utilization, etc. For large cloud service providers whose data centers span across multiple geographically distributed sites, global monitoring and emergence event handling involve tedious transmission and actuation delay. Therefore this paper proposes an edge computing platform for intelligent operational monitoring to leverage the edge computing benefits. The edge computing based monitoring framework can provide low latency for data transmission and decision making. They deploy wireless sensor networks (WSNs) and on-board sensors in server's mainboard to collect operational data. WSN is defined as a multi-hop, self-organizing network system formed by wireless communication between a large number of low-cost spatially dispersed micro sensors. WSNs aim to cooperatively sense, collect and process the information of the perceived objects in the network coverage area and then send them to the observers.

The platform architecture is listed in Figure 1.

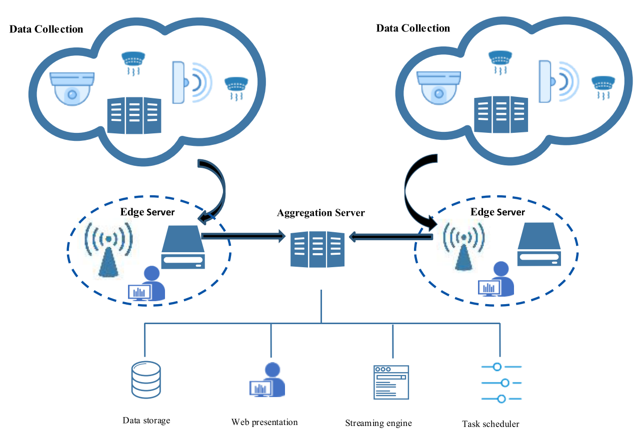


FIGURE 1. Architecture of data center intelligent operation and maintenance platform.

We will elaborate the design considerations and rationales of each part of the platform in the following sections.

A. DATA COLLECTION

Our proposed platform collects the following three kinds of information:

(1) Environment information of the data center, such as temperature, humidity, light, and fire/smoke.
(2) Resource information of clusters in the data center, such as CPU utilization and memory utilization.
(3) Hardware information of servers in the data center, such as energy consumption in cabinets, power consumption of servers, and power consumption of CPU and RAM.

Upon the receiving of the data, we use the *k-means* clustering algorithm to cluster all sensed data into several groups [39]. The monitoring data is divided into normal and abnormal groups, which help the operators analyze the operation health of the data center.

Our proposed platform provides open interfaces for the data requesting for any types of monitoring data, such as CPU data analysis in Intel RAPL, which can achieve the millisecond-level monitoring performance.

B. STREAMING ENGINE

The streaming processing engine in our platform processes the monitoring data from the data collection module in real-time. This engine can integrate machine learning and artificial intelligence algorithms such as ARIMA [40] and LSTM [41] to analyze real-time and historical data and thereby predicting the possible operations of the data center in the future. With the hardware information such as temperature, humidity, and server energy consumption in the data center, then the data center operator can predict the trends in hardware status of the data center. Moreover, by collecting the resource information of clusters in the data center, the data center operator can also predict the operational quality of workloads, the probability of mission failure, and workload characteristics. The stream processing engine can also model the data in the IDC in order to provide security guarantee for intelligent operation and maintenance, which help the operation and maintenance administrators to assess the healthy condition of the data center in advance, and thereby to make timely decisions for workload placement, job scheduling and migration.

C. EDGE SERVER

Due to the large number of data center servers, we divide the data center into multiple grids according to the logical function or geographical location, and each grid is responsible for collecting and preprocessing the monitoring data of this grid. As shown in Figure 1, since the edge server is closer to the data source, it can process the data in real time and present it to the user faster. And the edge server will send the processed data to the data aggregation server.

D. DATA AGGREGATION SERVER

The data aggregation server collects data from the edge server and forms a hierarchy with the edge server. This server usually collects information about multiple grids. The server stores the detected IDC information persistently and uses the open source chart drawing software such as *echarts* and *jfreechart* on the server to display the data in a line chart, bar chart, and other legends to facilitate the management [42], [43]. For example, *echarts* is a web-based, cross-platform JavaScript icon library which can provide intuitive, vivid, interactive, and highly customizable data visualization charts. It supports 12 types of charts such as line chart, bar chart, scatter chart, which enhance the user experience greatly. While the *jfreechart* is a Java chart library which supports a wide range of chart types and is easy to extend. Using the open source charting software, it is simple to display professional quality charts in our applications. Moreover, since the data aggregation server has a global view of the data center, it can make global job scheduling, including analyzing the operation of the entire data center and scheduling tasks according to the performance of each grid.

E. TASK SCHEDULING

We propose a task scheduling strategy to optimize the energy efficiency in the data center according to monitoring information. Based on historical information of servers in the data center, we can analyze the energy proportionality (EP) [44] and energy efficiency of servers.

The EP metric gives a quantification of server's energy consumption on different workload levels. Specifically, an ideally energy proportional server consumes 0 power when it's idle, and its power consumption should be proportional to its workload level (or resource utilization level) when it performs the computing. For example, an ideally energy proportional server should consume 2 times of energy at 20% utilization level than that at 10% level. However, for real world commercial servers, they exhibit rough or even no proportionality when workload intensities vary. Figure 2 illustrates one example of the EP curve between a real server's energy proportionality (EP = 1.02) and the energy proportionality curve of an ideally energy proportional server (EP = 1.0).

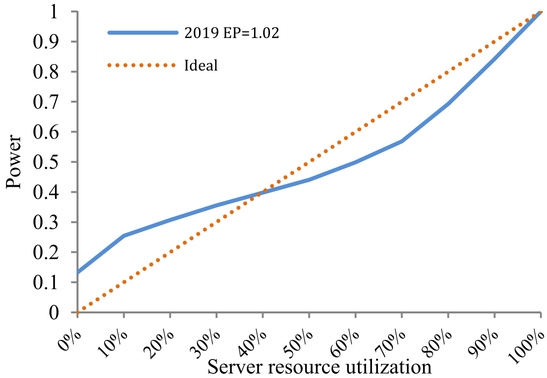


FIGURE 2. The energy proportionality curve.

The dotted line in Figure 2 is the energy proportional curve of the ideal server, and the solid curve is a real server's energy proportionality curve. It can be seen that the real server has a better working range than server with EP = 1.0. Between 60% and 90% utilization, putting the server in this load situation will be the best case for energy efficiency. If we can put each type of server in the entire data center into the appropriate working range, it will greatly help reduce the overall energy consumption of the data center. The extended experiments based on our previous work [18] shows that when the total power budget of the data center is fixed, running the server at the peak energy efficiency utilization point can not only increase the number of servers to be turned on, but also increase the total throughput of the data center. Specifically, the number of tasks completed in a unit of time is maximized. For example, running the server at the peak energy efficiency point can achieve an average of 47.3% more nodes than running the server at the 100% utilization point, and the total number of tasks in the entire data center increases by 3.69%. Therefore, the task scheduler can make energy aware decisions based on the real time monitoring of a variety of resources, such as CPU resources, memory resources, hard disk resources, network delays, etc., and the resources collected in the edge server can help task scheduler to make a more reasonable decision-making under different constraints.

Note that the environment monitoring network in the monitoring framework (i.e., the management network) is isolated from the data network in the IDC, which reduces the network management complexity and makes the monitoring network easier to expand. Moreover, this isolation improves the reliability of monitoring network and even if the data network in IDC is down, the monitoring network can still work normally.

IV. THE COMPONENTS OF THE PLATFORM

A. THE WORKFLOW OF PLATFORM

Current solutions for data center monitoring are based on coarse grained techniques, including data collection and visualization. These approaches can't provide predictive information for maintenance and operation. In this paper, our proposed platform for intelligent operational monitoring in IDCs provides better responsiveness for data acquisition, which is crucial for emergence handling and fine-grained resource scheduling. The proposed platform integrates wireless sensors and on-board built-in sensors to collect data during the operation and maintenance of IDCs. Figure 3 shows the workflow of the platform.

In the proposed platform, we firstly divide the IDC cluster into grids based on logical functions as shown in Figure 4. Each grid is configured with an edge server and an intelligent accessing point (AP). Secondly, through the establishment of the sensor network, the AP collects the environmental information of each grid and the data classification algorithm is provided on the AP, and the data is clustered and sent to the edge server. We incorporate an alarm program on the smart

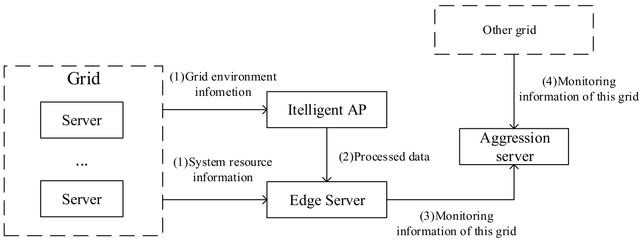


FIGURE 3. The workflow of the edge computing based monitoring platform for IDCs.

AP to ensure that even if the network in the data center fails, it does not affect the operation of the sensor network. The edge server and the aggregation server have the same functions, such as the streaming engine and the task scheduler, except that they have different data views. Thirdly, the edge server collects system resource information of the servers in the grid and each edge server has the environment information and system resource information of the grid. Since the edge server is closer to the data source, the information of this grid can be displayed and processed in real time. With this information, the edge server can make local decisions. Finally, the edge server sends the processed information to the aggregation server, which has global information that can be used to make decisions based on multiple sources of information.

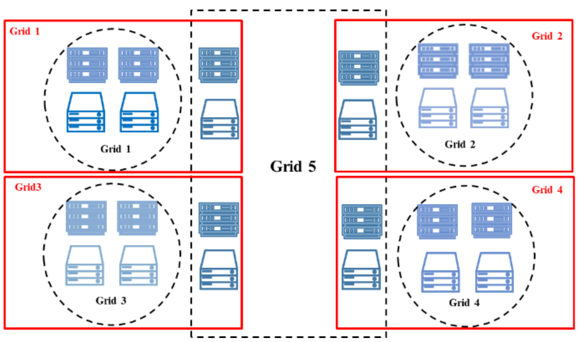


FIGURE 4. Dividing cluster into grids by logical function.

B. THE FUNCTIONALITY OF EDGE SERVER

As shown in Figure 1, in the server environment, we can arrange various sensors in the room and use wireless sensor network to connect them. We set up an intelligent AP and edge server for environmental data collection and processing. Due to the large scale of the IDC, different clusters may have different service functions. As shown in figure 4, we divide the clusters into different grids according to their logical functions. This grid can be expanded or changed on its own according to the service needs. In order to ensure real-time monitoring and mapping of data, we set up an edge server under each grid. Its functions are listed as follows:

1) Collecting IDC information of the grid and presenting it to the user interface. Because it is closer to the data source, it has better real-time performance.
2) Data processing. The sensor data can be pre-processed, processed into intermediate results, and reduced in data volume. For example, the monitoring data in a period time can be averaged and then transmitted to the edge for storage.

In the platform, the monitoring module polls equipment to collect server data, operation trends, and reports threshold violations for temperature and power. With built in alerting, the data center operator can react before failures impact users and services. It can be used as a distributed dispatching agent to make local decisions and perform reasonable task scheduling according to the task type of its owning grid. Since the edge server has a global view and has all the data of each grid, it can analyze and process information of the entire data center through the streaming engine and can use machine learning and artificial intelligence algorithms to count or predict the future load situation of the entire data center, the network traffic, power usage, task completion status, task failure status, etc.

C. THE FUNCTIONALITY OF THE WHOLE PLATFORM

Our framework also provides the following support interface for intelligent operation and maintenance in the data center:

1) Hardware lifetime prediction. With global data in the entire data center, such as years of temperature and humidity data in the server room, we can predict the hardware lifetime and the server's damage rate based on the information of servers.
2) Airflow CFD modeling and optimization. Using the data collected by temperature sensors and airflow sensors in each cabinet, the airflow path can be optimized to reduce the cost of the data center.
3) Task scheduling optimization. According to the response time in each grid and task completion status, we can further optimize the task allocation in the IDC. Since the configurations and performance in each grid are different, we can allocate the high-priority tasks to the high-performance grids.
4) Energy efficiency simulation. With global data in the entire data center, we can obtain a diagram to compare energy efficiency for each type of servers. IDC administrators can use this diagram to achieve energy efficiency in the entire data center. It also provides suggestions to eliminate those servers with low energy efficiency in a timely manner to optimize the energy efficiency of servers, and thereby saves energy consumption in the data center.

The monitoring system framework has the characteristics of high scalability and reliability. For example, the edge server in the framework can rely on its advantage of being closer to the data source, which makes it possible to process and analyze the data of the grid near the data source and reduce the data volume by centralized processing. Moreover, the edge server can also perform data analysis and prediction or perform distributed task scheduling by using machine learning algorithms based on the aggregated monitoring cluster information.

V. IMPLEMENTATION OF CASE STUDY

In this section, we implement a prototype of monitoring system for a small-size data center based on our framework proposed in Section III.

A. DATA COLLECTION

To effectively monitor the environmental conditions of data center zone such as temperature, humidity, smoke, light, and intrusion detection, we adopted wireless sensor network. The IDC environment monitoring system based on the wireless sensor network has the following advantages:

1) Autonomy. The high autonomy of WSN leads to no side effect on the normal operation of the network when nodes are joining and leaving the network.
2) Non-intrusive monitoring. Since our monitoring system is non-intrusive, we can deploy complex network devices in an existing complex data center network.
3) Energy-efficient. Due to the benefit of low-energy-consumption, sensor nodes can harness the energy harvest devices that use the heat generated from the servers in the data center.

Figure 5 illustrates the high-level overview of the wireless sensor network architecture.

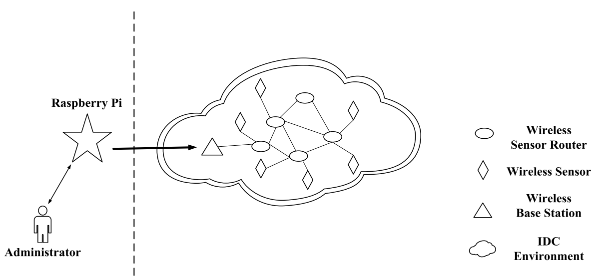


FIGURE 5. Wireless sensor network topology.

In such architecture in Figure 5, micro nodes are deployed to form a network in a self-organizing way. These micro nodes are integrated with sensors, data processing units, and communication modules. With embedded sensors, these micro nodes can directly measure the surrounding environment. Both wired and wireless devices can be used to operate as the communication module. However, short-distance low power wireless technologies are usually used as the medium in industry.

We adopt *Arduino* [45] as the sensor nodes collection platform, *ZigBee* [46] as the wireless data transmission protocol and *Raspberry Pi* as the monitoring platform (edge server). As shown in Figure 6, we use *ZigBee* for the data collected from the sensor nodes to be sent to the routing nodes in the network, and then sent to the base station node on the connection management platform. The base station node stores the data into the database and then use web server to update the database to administrators' monitoring interface. By building web server and database server in *Raspberry Pi*, visual monitoring service can be provided for server room administrators. Moreover, we use *DHT-11* temperature and humidity sensor, *MQ2* smoke sensor, infrared detector and light-sensitive resistors to collect the data of temperature, humidity, intrusion and light. We use *Mega* 8 *Arduino* board and the network module for *XBee* module of the wireless sensor network is from the *Digi* company.

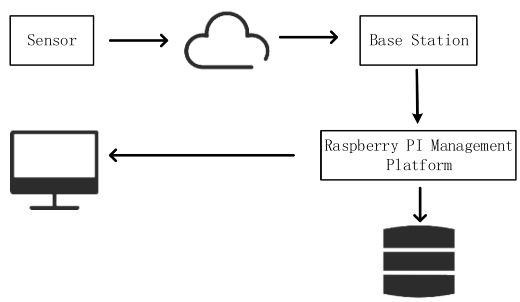


FIGURE 6. Monitoring data channel and flowchart.

In order to guarantee the scalability of the node, a tree topology supported by the *ZigBee* protocol is used as the topology of the wireless sensor network, which can provide self-organized, multi-hop, and reliable networking communication with low power consumption which is applicable for edge devices constrained by the battery lifetime. In our system implementation, as shown in figure 7, the initialization of the *ZigBee* is controlled by the coordinator.

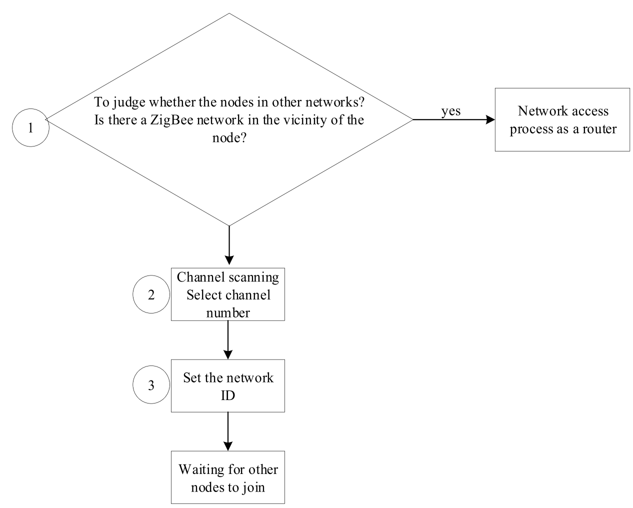


FIGURE 7. Process of ZigBee node network access.

Before establishing the network, the coordinator first judges whether the node is connected to other network, and if there is no connection with others. Otherwise, it will search the appropriate channel and apply a unique network identifier to the channel as a new network waiting for other nodes to connect. After the establishment of network connection, the node will scan the coordinator and request the permission for access to the network. When it receives the request, the coordinator will send an acknowledgement response and inform the upper layer. Then the *MAC* layer decides whether to allow the node to join the network based on its own resources (storage and power). When the upper layer agrees that the node can enter the network, the coordinator initiates a connection request to the client node, then the node joins the network successfully finally. The workflow of the network access is shown in Figure 8.

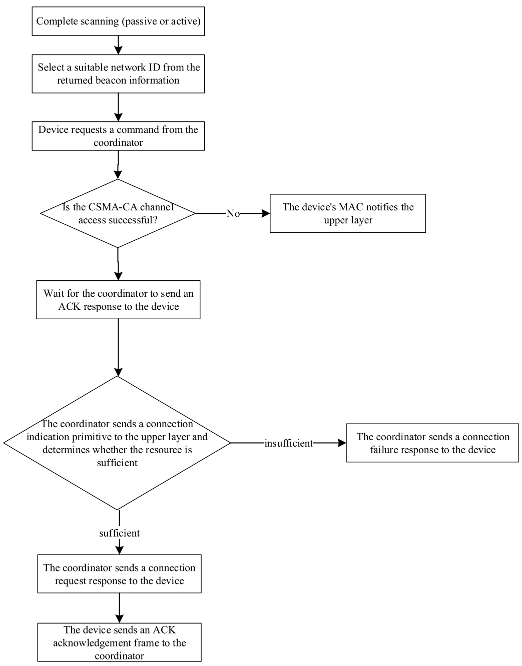


FIGURE 8. ZigBee network initialization process.

The branch nodes of the initial network are composed of routers, which connects the node in the same network. In this topology, the terminal device does not participate in message forwarding, while the routing nodes can establish a subnet independently of the coordinator and then forward the message. To achieve the scalability of nodes, the coordinator is used as a base station node, and the remaining nodes are networked in a tree topology as shown in Figure 9.

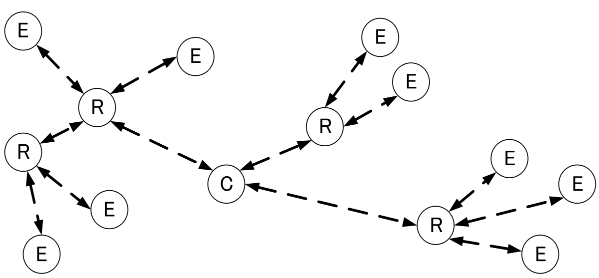


FIGURE 9. Tree topology.

Thanks to our previous design, the environment detection module makes it possible that the monitoring in the server room is global, autonomous and non-invasive. The use of *Raspberry Pi*, *Arduino* module and necessary sensors can process the data and send the data to the data aggregation server, which meets the basic functional requirements of the monitoring system. Note that if we use Ad hoc networking policy, more than 40 nodes are not recommended due to connection failures in our implementation. Therefore we recommend using source routing to learn routing in large-scale sensor networks. However, because of the limited scale of the experiment, we still use the ad hoc networking method in the prototype implementation.

Figure 10 illustrates the workflow of data transmitting and data receiving in our platform. The sensors are responsible for data collecting and transmitting while the *Raspberry Pi* is used for base station for data receiving and processing. Figure 10(a) shows how the sensor collects the data(take *DHT11* temperature sensor as an example), it queries the system time first to compute the time interval between the last sampling and this sampling, then it will check if the time interval can satisfy our requirement, then the data collection can be started once the requirement is satisfied. And figure 10(b) illustrates the workflow of data receiving process. After receiving the data packet, the *Raspberry Pi* will check the validity of the data, and then it will discard the invalid packets and parse the valid packets, and finally it will send the parsed data to the database for persistent storage.

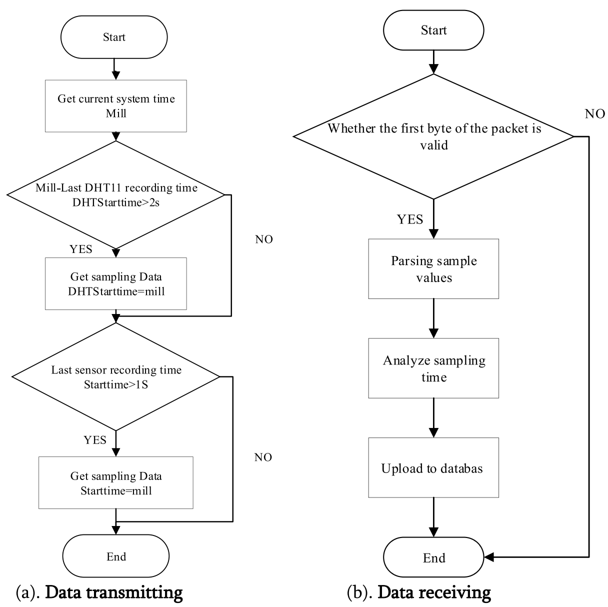


FIGURE 10. The data transmitting and receiving flowchart.

B. HARDWARE AND SYSTEM RESOURCE INFORMATION COLLECTION

In this section we implemented server hardware and resource monitoring of the prototype system. As shown in Figure 11, we use the *IPMI* (intelligent platform management interface) protocol to implement accurate and real-time collection of power consumption data for servers and provide more convenient monitoring tools for the server operation and maintenance managers. With IPMI capability, users can obtain physical status such as voltage, power and temperature of servers.

The edge server polls the host to be monitored according to the configuration at configured intervals (2s-3s), obtains parameters such as CPU operating temperature and power supply, and then writes them into the database. The monitoring host uses the Linux *sar* statistics command to obtain information such as the CPU and memory usage and disk utilization of the targeted monitoring host, and then writes it to the database.

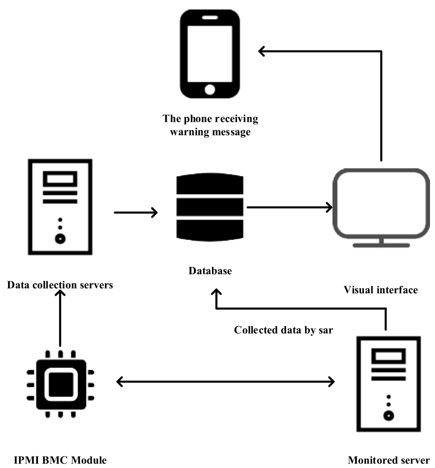


FIGURE 11. Server monitoring function structure.

The obtained information can also be displayed in the form of a line chart in real time. The system also allows the user to set the listening parameter threshold in advance and the client monitoring parameters. In case the parameter is abnormal for a period of time, an alarm is issued and then it will send a short message to system administrator. Figure 12 shows a flow chart of the resource monitoring system.

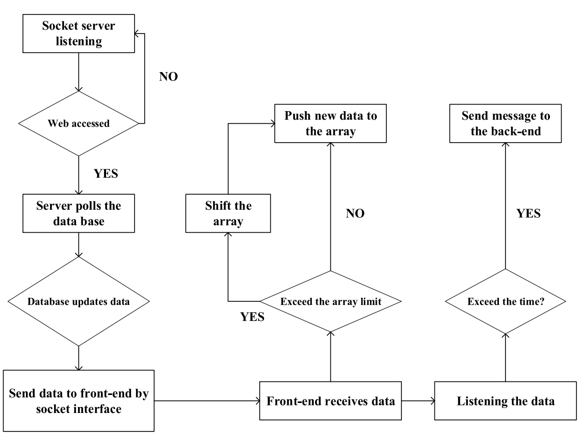


FIGURE 12. Flowchart of resource monitoring.

C. SERVER CABINET ENERGY CONSUMPTION COLLECTION

In our prototypical implementation, the power distribution module (PDM) is a *Sugon* box, which provides the query capability with configured baud rate, input maximum current, and version number. In the display data information mode, users can view the input and output data information of PDM, including the voltage of three-phase *L1*, *L2* and *L3*, total current, active power, apparent power and the branch current corresponding to the three-phase output. The collection process consists of two steps: the extraction and analysis of power information from the cabinet PDM and displaying the information on the web page for visualization. It provides data support for data center management and task scheduling by collecting circuit information of cabinets in data center. The functions of the two modules are listed as follows:

1) Information extraction. The information extraction module includes two functional modules, which are the extraction and analysis of electrical information. The data is read by the edge server through the *RS485* interface of the PDM. The process includes opening the serial port and the serial data can be read in a complete and effective way after proper configuration. Then the data from the serial port is analyzed and written into the database through a *Java* program installed on the edge server.

2) Data visualization. This module realizes data visualization on web pages with charts generated by *echarts*.

D. REAL-TIME INFORMATION DISPLAY

In order to enable data visualization in real time and ensure interactivity between the monitoring system and users, we use *websocket*, which is one of the new features of *HTML5*.Websocket is based on the socket interface provided by internet browsers, which is similar to the interaction between the traditional client/server architecture. Specifically, the browser achieves the interaction between the socket and the server side. Instead of using *http* protocol, *websocket* uses the *ws* protocol for better performance, which is based on the traditional *TCP* protocol. Therefore, WS protocol achieves a two-way communication between the server and the browser. Compared to traditional polling, *websocket* is more prevalent due to its better performance. Figure 13 is the performance comparison of *websocket* and *polling*.

In the comparison in Figure 13, three use cases are presented(i.e., Use Case A, Use Case B and Use Case C), where it is set as 1000 clients polling per second, 10000 clients polling per second, and 100000 client polling per second, respectively.

We use *JSON* (JavaScript Object Notation) for data exchange between the front and back ends. It is a light-weight data exchange format due to its independence of programming languages and it is convenient for users to write and read data with lightweight overhead. In addition, *JSON* is also easier to be analyzed by machines. We use *Node.js* engine

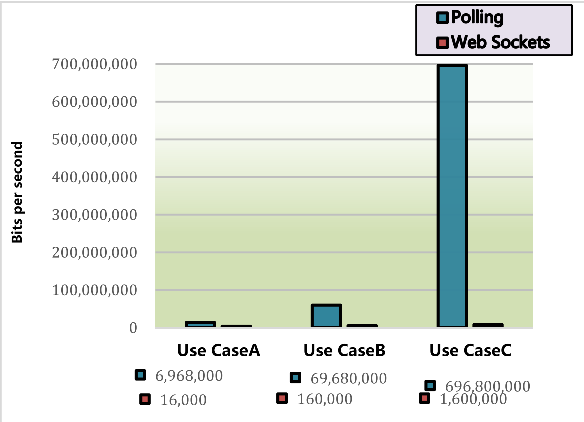


FIGURE 13. Performance comparison between websocket and polling mode (lower is better).

driven *Javascript* to build *websocket* server. The rationale behind is as following:

Firstly, the background client opens *websocket* service and waits for the access requests from user browsers. Once the browser's connection is monitored, the database begins to be polled at a certain time interval. Once the data collected from the data acquisition module is updated, the new updated data is sent to the front desk through the *websocket* interface. In addition, since the front-end uses the open source visual library *Echarts.js*, the front-end pages send sensed data into the original array, and dynamically update the online visualization graph.

We use *echarts* to draw discount diagrams. *Echarts* is a JavaScript-based open source visualization library compatible with most of the browsers such as Microsoft IE8/9/10/11, Google Chrome, Firefox, and Apple Safari. The bottom layer of *echarts* is light-weight vector graphics library *ZRender*, which offers users with intuitive, interactive, and highly-customized data visualization charts.

Whenever the front-end receives the data from the *websocket*, it is necessary to match and compare it with the preset threshold. Once the threshold is exceeded or below the threshold, the value is continuously monitored. If the time exceeds a certain length, the abnormal situation is reported to the background through the *websocket*, and the messaging interface is called by the background to send short message to the administrator's mobile phone. If the normal situation is not achieved for a period of time, it will continue to send short messages until the abnormal situation is resolved.

VI. EVALUATION

In this section, we evaluate our proposed prototype system to demonstrate that it meets the basic requirements of a monitoring system in large data centers.

A. ENVIRONMENTAL INFORMATION COLLECTION EVALUATION

Figure 14 represents the monitoring of humidity, temperature, light, and smoke in data center, respectively. In the database, we can directly access real-time data and use the data for deeper mathematical analysis. Figure 14 shows that we have collected data from 21:15 P.M. to 23:24 P.M. on May 26, 2018.

Firstly, we analyze the humidity data in the monitoring system. During the monitoring period, a total of 736 records were recorded. The mathematical statistics are shown in figure 14 (a). We can observe that, the humidity data is basically stable at 19%, which is a low humidity value and will greatly reduce the risk of equipment rot and short circuit due to moisture in the data center. And the mean square error is 0.9663. Therefore, the humidity in the room is in a quite stable condition, which helps servers to operate smoothly.

Similarly, the statistical analysis of temperature, smoke, and light are shown in figure 14 (b)(c)(d). We can observe that, the mean square error is at a very low level, indicating that the data is relatively stable. In addition, the average value of temperature is 19 degrees (Celsius), which is safe for server operation in the data center.

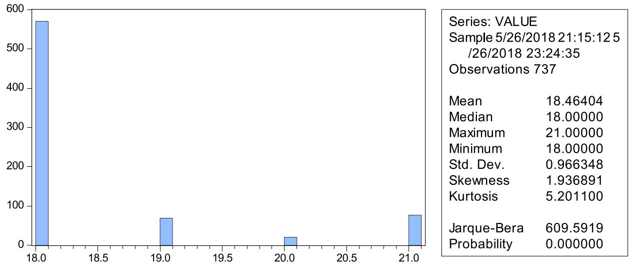
B. SYSTEM RESOURCES INFORMATION COLLECTION EVALUATION

We run *SPECpower* and *STREAM* benchmark on clusters to test our monitoring system. *SPECpower* is an industry-standard benchmarking tool for measuring server power and performance at different utilization levels and then calculate its overall energy efficiency scores. The *STREAM* benchmarking tool is a comprehensive benchmark program which is widely used in industry to measure the performance of memory bandwidth. It is an industry-recognized tool to benchmark the performance of memory bandwidth and a universal tool to measure the performance of server memory.

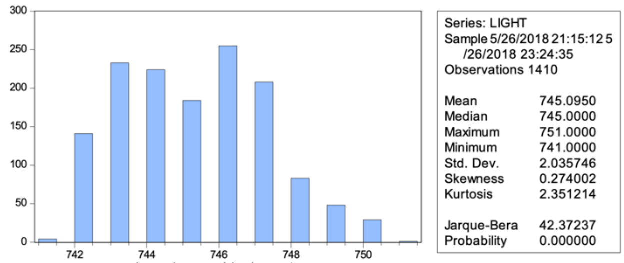
We first read the data from the multi-meter and *IPMI* interface under idle conditions to collect the power information for comparison. After the data acquisition system is started, the output power is collected by a multi-meter and *IPMI* interface. Note that in our testing the accuracy of the server's power information measured by the external multi-meter is better than IPMI. We collect data every one second for 1000 groups and then get the result as shown in Figure 15.

We measured 116.32 watts using the multi-meter when the server was idle. The *IPMI* interface command is used to calculate the average calculation data. Since the *IPMI* command collects data as an integer, the energy consumption is 116W. This shows that the error of the two sets of data measurement is relatively small, which is within 0.5watts. We can conclude that, when the server is idle, the information collected by the acquisition system is accurate and effective, which is very close to the multimeter's results. Hence, under the idle workload, our monitoring system fully meets the expected requirements. Then we run *SPECpower* to observe power and memory usage of the server. The power data was collected and analyzed as shown in Figure 16.

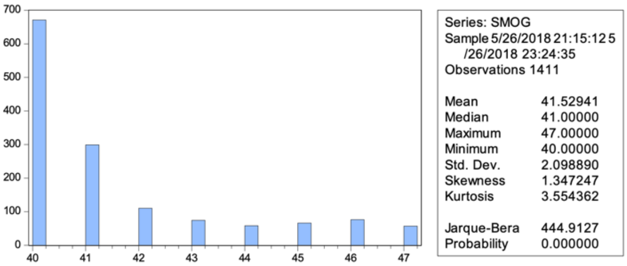
From Figure 16, we can observe that, the two curves of power consumption obtained by the two methods are basically the same, and the correlation coefficient is 0.97.



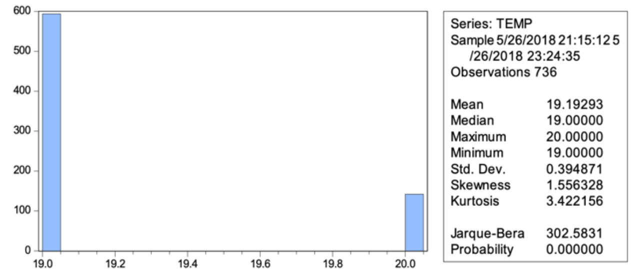
(a). Statistical analysis of temperature values



(b) Statistical analysis of light values



(c). Statistical analysis of smog values



(d). Statistical analysis of temperature values

FIGURE 14. Computer room environment data analysis results.

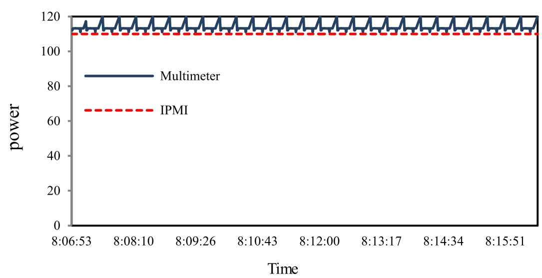


FIGURE 15. Monitoring system and multimeter acquisition power comparison chart (idle).

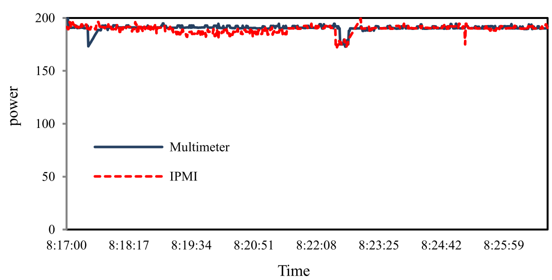


FIGURE 16. Monitoring system and multimeter acquisition power comparison chart (SPECpower benchmark).

Therefore, Figure 15 and Figure 16 verify that our approach can accurately obtain power consumption data under idle and intense workload conditions. The STRAM benchmark can set different memory usage values by adjusting the size of the array. We measure server power consumption by running STREAM with three different array sizes. As shown in Table 1, the *IPMI* has an error of less than 2 Watts compared to the average power consumption of the multi-meter.

TABLE 1. IPMI and meter average power comparison.

	IPMI Power Average	Meter Power average
Array Size=500,000,000	143.8W	144.0W
Array Size=1,000,000,000	154.6W	155.5W
Array Size=1,500,000,000	157.9W	158.6W

In terms of system resource information collection, we use the edge node for data collection and processing. Each grid will send its servers' information, such as CPU usage, memory usage and so on, to its own edge server. The frequency of collecting information can be adjusted according to the size of the cluster. If the cluster is very large, the cluster can be divided into more grids to reduce the pressure on the edge servers. We provide interface that users can use to access the values the want. And the processed information can be sent to the aggregation server by the edge server. In this way, it puts less pressure on the central servers and reduces the data transmission which can be extended to large clusters easily. Our system can accurately obtain the information of system resource server power consumption and data center environment information.

VII. CONCLUSION

Data centers are becoming increasingly popular for the provisioning of various services including computing, storage, and network. The cost and operating expenses of data centers have increased significantly with the increase in computing capacity of large data centers. Services availability and performance guarantee are the key goals for data center operation. Therefore, real time monitoring for data centers are vital to accomplish this goal. Data center level monitoring can help maintain hardware assets as well as SLAs in complex environments where there are different requirements for services availability, data retention, and network availability. Moreover, it can maximize server uptime, server energy efficiency, and increase the data center's productivity. To this end, this paper proposes an edge computing platform for intelligent operational monitoring in IDCs. The platform uses sensor networks, *IPMI*, and other tools to collect information of data center wide environment, system resource and power consumption data. By dividing the logic of the data center clusters into grids and placing edge servers in each grid, we can perform data processing, data analysis, task prediction, and task scheduling, which improves the real-time performance of the monitoring system, and reduces the latency in data transmission to central clouds. In the aggregation server, we have global information about the data center. Based on this global information, we can integrate algorithms such as deep learning algorithms to analyze the new condition of energy efficiency, probability of mission failure, and energy consumption of the data center. According to the specific energy efficiency of each grid, energy efficiency in the entire data center can then be optimized through data center wide simulation.

For future work, we will implement a stream processor that can schedule tasks in each grid, which improves service quality and reduces resource overhead. Moreover, since energy monitoring and simulation is one of the most concerns of current data centers, such as CFD modeling of airflow, from the scratch and after the servers' retirement generation by generation, we will also conduct experiments and simulation based on monitoring and predication in a real large data center to verify the efficacy of this platform.

ACKNOWLEDGMENT
This article was presented at the HiDEC2018 Workshop [1].

REFERENCES

[1] Y. Qiu, C. Jiang, T. Fan, and J. Wan, "An edge computing platform for intelligent Internet data center operational monitoring," in *Proc. Int. Workshop HPC Supported Data Anal. Edge Comput. (HiDEC)*, Beijing, China, Jun. 2018, pp. 16–28.

[2] *Cisco Global Cloud Index: Forecast and Methodology*. pp. 2014–2019. Accessed: Jun. 13, 2019. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud Index White Paper.html

[3] A. Shehabi, S. J. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin, and W. Lintner, "United states data center energy usage report," Lawrence Berkeley National Laboratory, Berkeley, CA, USA, Tech. Rep. LBNL-1005775, 2016.

[4] *America's Data Centers Consuming and Wasting Growing Amounts of Energy*. Accessed: Jun. 13, 2019. [Online]. Available: http://www.nrdc.org/energy/data-center-efficiency-assessment

[5] *Data Centre Energy Efficiency Benchmarking -E2 Singapore*. Accessed: Jun. 13, 2019. [Online]. Available: http://www.e2singapore.gov.sg/DATA/0/docs/Resources/NEADCEnergyBenchmarkingSummary-FinalReport.pdf

[6] C. Jiang, G. Han, J. Lin, G. Jia, W. Shi, and J. Wan, "Characteristics of co-allocated online services and batch jobs in Internet data centers: Aoud," *IEEE Access*, vol. 7, pp. 22495–22508, 2019.

[7] A. Farrell and H. Hoffmann, " MEANTIME: Achieving both minimal energy and timeliness with approximate computing," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2016, pp. 421–435.

[8] H. Hoffmann, "JouleGuard: Energy guarantees for approximate applications," in *Proc. 25th Symp. Operating Syst. Princ.*, Oct. 2015, pp. 198–214.

[9] C. Imes, D. H. Kim, M. Maggio, and H. Hoffmann, "POET: A portable approach to minimizing energy under soft real-time constraints," in *Proc. 21st IEEE Real-Time Embedded Technol. Appl. Symp.*, Apr. 2015, pp. 75–86.

[10] N. Mishra, H. Zhang, J. D. Lafferty, and H. Hoffmann, "A probabilistic graphical model-based approach for minimizing energy under performance constraints," *ACM SIGPLAN Notices*, vol. 50, no. 4, pp. 267–281, Mar. 2015.

[11] Q. Wu, Q. Deng, L. Ganesh, C.-H. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, and Y. J. Song, "Dynamo: Facebook's data center-wide power management system," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2016, pp. 469–480.

[12] L. A. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synth. Lectures Comput. Archit.*, vol. 4, no. 1, pp. 1–108, Jan. 2009.

[13] D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica, "Clipper: A low-latency online prediction serving system," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2017, pp. 613–627.

[14] C. Delimitrou and C. Kozyrakis, "Quasar: Resource-efficient and QoS-aware cluster management," *ACM SIGARCH Comput. Archit. News*, vol. 42, no. 1. ACM, 2014, pp. 127–144.

[15] J. Li, W. Liu, T. Wang, H. Song, X. Li, F. Liu, and A. Liu, "Battery-friendly relay selection scheme for prolonging the lifetimes of sensor nodes in the Internet of Things," *IEEE Access*, vol. 7, pp. 33180–33201, 2019.

[16] W. Zhang, W. Liu, T. Wang, A. Liu, Z. Zeng, H. Song, and S. Zhang, "Adaption resizing communication buffer to maximize lifetime and reduce delay for WVSNs," *IEEE Access*, vol. 7, pp. 48266–48287, 2019.

[17] J. Tan, W. Liu, M. Xie, H. Song, A. Liu, M. Zhao, and G. Zhang, "A low redundancy data collection scheme to maximize lifetime using matrix completion technique," *EURASIP J. Wireless Commun. Netw.*, to be published. doi: 10.1186/s13638-018-1313-0.

[18] Y. Qiu, C. Jiang, Y. Wang, D. Ou, Y. Li, and J. Wan, "Energy aware virtual machine scheduling in data centers," *Energies*, vol. 12, no. 4, p. 646, Feb. 2019.

[19] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Towards computation offloading in edge computing: A survey," *IEEE Access*, to be published. doi: 10.1109/ACCESS.2019.2938660.

[20] R. Bose, S. Sahana, and D. Sarddar, "An adaptive cloud service observation using billboard manager cloud monitoring tool," *Int. J. Softw. Eng. Appl.*, vol. 9, no. 7, pp. 159–170, Jul. 2015.

[21] N. Kalyani, "An energy efficient dynamic schedule based server load balancing approach for cloud data center," *Int. J. Future Gener. Commun. Netw.*, vol. 8, no. 3, pp. 123–136, 2015.

[22] C. Jiang, T. Fan, Y. Qiu, H. Wu, J. Zhang, N. N. Xiong, and J. Wan, "Interdomain I/o optimization in virtualized sensor networks," *Sensors*, vol. 18, no. 12, p. 4395, Dec. 2018.

[23] C. Jiang, Y. Wang, D. Ou, Y. Li, J. Zhang, J. Wan, B. Luo, and W. Shi, "Energy efficiency comparison of hypervisors," *Sustain. Comput. Inform. Syst.*, vol. 22, pp. 311–321, Jun. 2019.

[24] T. Wang, B. Qin, Z. Su, Y. Xia, M. Hamdi, S. Foufou, and R. Hamila, "Towards bandwidth guaranteed energy efficient data center networking," *J. Cloud Comput.*, vol. 4, no. 1, p. 9, May 2015.

[25] X. Wang, X. Wang, G. Xing, J. Chen, C.-X. Lin, and Y. Chen, "Towards optimal sensor placement for hot server detection in data centers," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 899–908.

[26] H. Chan, J. Connell, C. Isci, J. O. Kephart, J. Lenchner, C. Mansley, and S. McIntosh, "A robot as mobile sensor and agent in data center energy management," in *Proc. 8th ACM Int. Conf. Autonomic Comput.*, Jun. 2011, pp. 165–166.

[27] G. Katsaros, J. Subirats, J. O. Fitó, J. Guitart, P. Gilet, and D. Espling, "A service framework for energy-aware monitoring and VM management in Clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 8, pp. 2077–2091, Oct. 2013.

[28] M. Massie, B. Li, B. Nicholes, V. Vuksan, R. Alexander, J. Buchbinder, F. Costa, A. Dean, D. Josephsen, P. Phaal, and D. Pocock, *Monitoring With Ganglia: Tracking Dynamic Host and Application Metrics at Scale*. Newton, MA, USA: O'Reilly Media, 2012.

[29] Y. Yin, L. Chen, and J. Wan, "Location-aware service recommendation with enhanced probabilistic matrix factorization," *IEEE Access*, vol. 6, pp. 62815–62825, 2018.

[30] H. Gao, W. Huang, X. Yang, Y. Duan, and Y. Yin, "Toward service selection for workflow reconfiguration: An interface-based computing solution," *Future Gener. Comput. Syst.*, vol. 87, pp. 298–311, Oct. 2018.

[31] M. Huang, W. Liu, T. Wang, H. Song, X. Li, and A. Liu, "A queuing delay utilization scheme for On-path service aggregation in services-oriented computing networks," *IEEE Access*, vol. 7, pp. 23816–23833, 2019.

[32] Y. Liu, A. Liu, X. Liu, and X. Huang, "A statistical approach to participant selection in location-based social networks for offline event marketing," *Inf. Sci.*, vol. 480, pp. 90–108, Apr. 2019.

[33] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Commun. Mobile Comput.*, Apr. 2018, Art. no. 3075849.

[34] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.

[35] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, Apr. 2011, pp. 301–314.

[36] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervas. Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[37] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[38] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[39] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[40] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.

[41] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 473–479.

[42] *ECHARTS*. Accessed: Jun. 13, 2019. [Online]. Available: https://echarts.baidu.com/

[43] *JFreeChart*. Accessed: Jun. 13, 2019. [Online]. Available: http://www.jfree.org/jfreechart/

[44] F. Ryckbosch, S. Polfliet, and L. Eeckhout, "Trends in server energy proportionality," *Computer*, vol. 44, no. 9, pp. 69–72, Sep. 2011.

[45] *ARDUINO*. Accessed: Jun. 13, 2019. [Online]. Available: https://store.arduino.cc/components/components-sensors

[46] J. Lee, Y. Su, and C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *Ind. Electron. Soc.*, vol. 5, pp. 46–51, Nov. 2007.

CONGFENG JIANG is currently an Associate Professor with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Hangzhou, China. He is also with the School of Computer Science and Technology, Hangzhou Dianzi University. He has published more than 70 articles in grid computing, cloud computing, virtualization, and big data systems. His research interests include system optimization, performance evaluation, and distributed system benchmarking. He is also a member of ACM and CCF.

YELIANG QIU is currently pursuing the master's degree with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Hangzhou, China. He is also with the School of Computer Science and Technology, Hangzhou Dianzi University. His research interests include cloud computing, distributing systems, and power management.

HONGHAO GAO received the Ph.D. degree in computer science and started his academic career at Shanghai University, China, in 2012, where he is currently with the Computing Center. His research interests include service computing, model checking-based software verification, and sensors data application. He is currently a Fellow of IET, BCS, and EAI. He is also a Senior Member of CCF and CAAI.

TIANTIAN FAN is currently pursuing the master's degree with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Hangzhou, China. She is also with the School of Computer Science and Technology, Hangzhou Dianzi University. Her research interests include cloud computing, edge computing, and distributing systems.

KANGKANG LI received the Ph.D. degree in computer science from the Department of Computer Science and Engineering, University of Notre Dame. He is currently an Assistant Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His research interests include cloud computing and high-performance computing.

JIAN WAN received the B.S. degree in mechanical engineering, the M.S. degree in mathematics, and the Ph.D. degree in computer science from Zhejiang University, in 1990, 1993, and 1996, respectively. From 2000 to 2015, he was the Dean of the School of Computer Science and Technology, Hangzhou Dianzi University. He is currently a Professor with the School of Information Engineering, Zhejiang University of Science and Technology, Hangzhou, China. He is also the Vice President of the Zhejiang University of Science and Technology and the Vice Director of the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou. He has published more than 150 articles in grid and services computing, cloud computing, virtualization, and distributed systems. His research interests include distributed systems, computer networks, and big data analytics. He is also a member of CCF.