Monitoring network of energy consumption in servers of a data center

José Ignacio Vega Luna[1], Gerardo Salgado Guzmán[1], Francisco Javier Sánchez Rangel[1], José Francisco Cosme Aceves[1], Mario Alberto Lagos Acosta[1],Víctor Noé Tapia Vargas[1]

1 Área de Sistemas Digitales-Departamento de Electrónica, Universidad Autónoma Metropolitana-Azcapotzalco, Ciudad de México, México

IJASR 2022
VOLUME 5
ISSUE 2 MARCH – APRIL **ISSN: 2581-7876**

Abstract: With the continued growth and complexity of computing, applications, and IoT, cloud computing has grown rapidly in recent years. This has created a significant increase in e-commerce applications, big data, and the emergence of a variety of computing platforms on the Internet, which require more processing power and resources. This has also generated the appearance of a significant number of data centers that house the IT infrastructure that supports the development of these activities. It is important that data centers provide efficient and reliable services to customers and users because productivity and satisfaction depend on this. One of the important aspects to consider when talking about services is the efficient management and monitoring of electrical energy consumption. This paper presents the development of a network for monitoring energy consumption in servers in a data center. The network is composed of four nodes based on a NODE MCU ESP32 module and a current transformer type sensor. Its purpose is to determine the price that the client must pay for the energy consumption made. The nodes periodically transmit the power consumed by the servers to the Think Speak Internet platform. Information received by Think Speak can be viewed from the Thing view app and the Think Speak web page. The tests carried out indicated that the monitoring network has an accuracy of 1.186% on average.

Keywords: Energy consumption, current transformer sensor, IoT, monitoring, Think Speak.

1. Introduction

With the continued growth and complexity of computing, applications, and IoT, cloud computing has grown rapidly in recent years. This has created a significant increase in e-commerce applications, big data, and the emergence of a variety of computing platforms on the Internet, which require more computing power and resources [1]. This has also generated the appearance of a significant number of data centers where the IT infrastructure that supports the development of these activities is located. Data centers are forced to provide efficient and reliable services to customers and users since productivity and satisfaction depend on this [2]. One of the important aspects to consider when talking about services is efficient energy management. The components of a data center that consume the most energy arethe computing, telecommunications equipment and the cooling systems [3]. The erroneous management of energy consumption in cloud data centers [4], in addition to affecting the environment due to carbon emissions, negatively impacts service-level agreements (SLA) and operating costs [5]. To achieve the goal of maximizing profits and offering reliable and cost-effective services, it is necessary to maximize energy consumed in data centers, considering that the price of electricity is constantly increasing and the depletion of non-renewable resources [6].

Data centers have sensor networks to monitor the status of the operation of servers, storage arrays, networking devices, and cooling systems. This allows the optimal operation of green data centers, satisfying user demand and at the same time offering adequate cost services based on the amount of resources used [7].

There are different types of technologies that allow determining the consumption of electrical energy, among which are the transduction method, capacitive, resistive, diode/FET-based, and MEMS sensors [8].The first type is one of the most widely used and includes contactless galvanically isolated magnetic field-based, or Hall-based, current sensors [9]. This technology is used in various applications such as power distribution, power electronics, and drive technologies [10] as it has advantages such as small size devices, low cost, low power consumption, and high dynamic range [11].

The objective of this work was to develop a system to remotely measure in real time the energy consumed by four servers in a data center. It was requested that the system be integrated by devices that are compact in size, reliable, easy to operate, low in cost and, of course, low in energy consumption. Another requirement was that the installation of the system does not require the modification of the existing network cabling in the data center or the installation of additional cabling, not intrusive. The system must have a user interface where the measurements can be viewed from the Internet and the amount of energy consumed by each server can be downloaded to a CSV file. The information in this file will be an element to be considered by the data center to determine the cost of the services offered to the client or user. Different Wi-Fi access points are installed in the data center for connection to the Internet. The furthest access point to one of the servers is 10 meters away. It was suggested that, depending on the results obtained and the operational characteristics of the developed system, the data center will decide whether to carry out a second version of the system with more functionalities.

In order to meet the requested characteristics, it was determined that the system is composed of a network of four nodes, one for monitoring each server. Each node integrates the current sensor, the controller and the WiFi wireless interface. It was also determined to use a current transformer as a sensor because it is compact, reliable and easily available. Similarly, the NODE MCU ESP32 embedded module was selected as the controller since it integrates the WiFi interface and an analog/digital converter to read the analog signal delivered by the current sensor.

Reviewing the state of the art regarding the monitoring and efficient use of energy consumed in data centers, in recent years various investigations and surveys have been carried out on data management strategies [12]whose purpose has been to propose techniques and methods to monitor and reduce energy consumption. Some techniques consist of the use of virtual machines (VM) and adaptive algorithms to balance the load and applications that run on them [13-14], others are based on waste heat recovery [15], some more use fine-grained energy consumption monitoring model that focus on single system component such as CPU, memory, disks and network interfaces using the least median squares regression model[16-17], also some are based on temperature monitoring in server racks and avoid hotspots [18] and in cost optimization by capacity planning using dynamic algorithms [19]. Additionally, methods have been proposed to model the power consumption of the servers in datacenters based on ANN (Artificial Neural Networks) [20].

Regarding the techniques used to determine the energy consumption, the latest research carried out uses new measurement methods such as coreless current sensors based on circular arrays of magnetic field in both the closed loop and open loop operation [21], optical sensors based on magneto strictive material that employs a single fiber bragggrating (FBG) [22] and Rogowski coil current sensors [23].

2. Materials and methods

The monitoring network developed is made up of four nodes. Each node is installed at the rear of the corresponding server in which the electrical energy consumed, or power used, is monitored, as shown in Figure 1.
The architecture of the nodes is made up of three elements: the NodeMCU ESP32 module, the current sensor and the signal conditioning stage. Figure 2 shows the architecture of the network nodes.

Node MCU ESP32 module

This embedded system is based on the ESP32 microcontroller SoC. The main resources it has are the following: 240Mhz TensilicaXtensa Dual-Core 32-bit LX6 microprocessor, WiFi interface 802.11 b/g/n/e/i, Bluetooth interface v4.2 BR/EDR and Bluetooth Low Energy (BLE), 448 KB ROM, 520 KB SRAM, 4 MB Flash, 24 GPIO (General Purpose Input Output) terminals, two 12-bit/18-channel analog-to-digital converters and two UART ports.

The nodes of the network access the Internet through an access point installed in the data center.

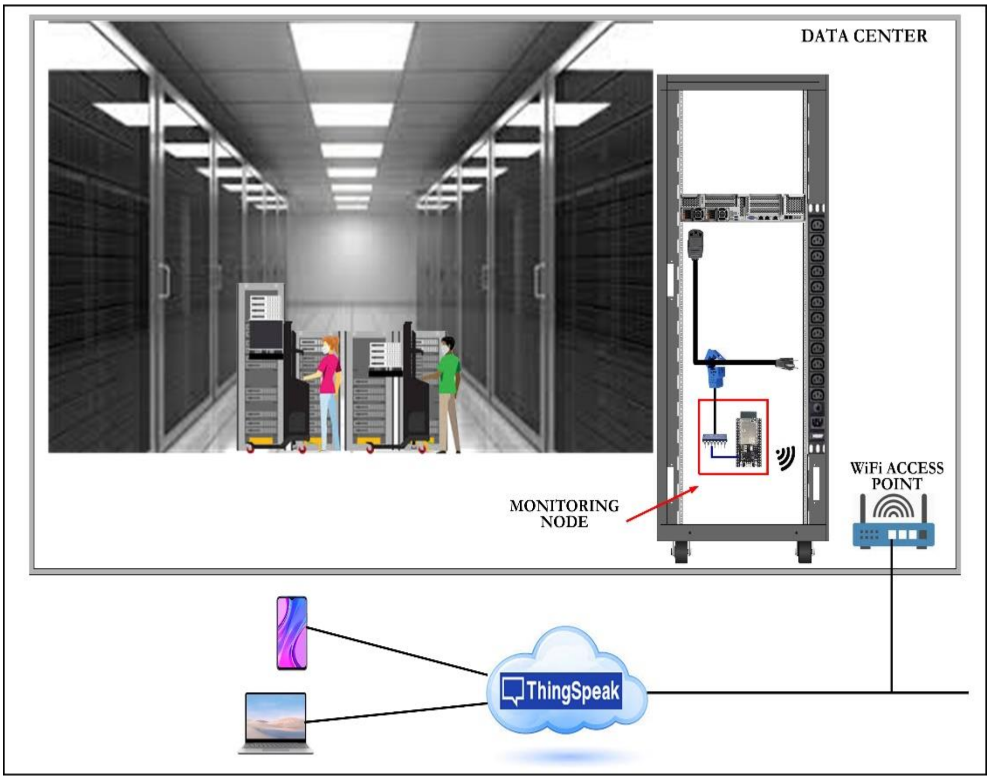

Figure 1. Monitoring network node

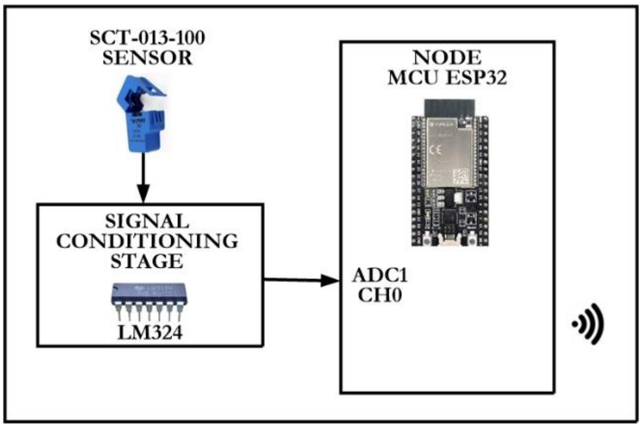

Figure 2. Network node architecture

Current sensor

There are different technologies that allow determining the consumption of electrical energy. One of them is through sensors that measure the flow of current in an electrical circuit. The SCT-013-XXX series sensors are devices whose operation is based on a transformer. The primary winding of the transformer is the cable through which the current of the circuit to be monitored circulates. The number of turns of the secondary winding depends on the model of the sensor. The magnitude of the current flowing through the circuit cable is a function of the number of turns of the secondary and determines the value of the current or voltage delivered by the sensor. Sensors that deliver voltage to the output have an internal load resistor.For the above reasons, SCT-013-XXX sensors are also known as current transformers(CT). One of their advantages is that they are non-invasive, since the core of the transformer is made up of two sections that can be separated to enclose the electrical power cable of the monitoring circuit, similar to a clothes peg, as shown in Figure 3.

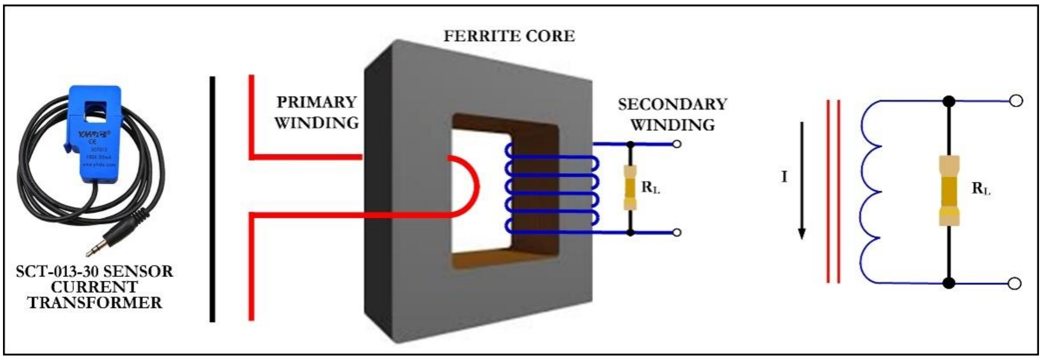

Figure 3. Current sensor/current transformer

In this work the current sensor SCT-013-100 was used. This sensor allows measuring a current of up to 100 A, delivers an output of 50 mA for 100 A (100A:50mA), has an accuracy of +/- 1% and does not have an internal load resistance. The sensor's alternating output signal is between -50 mA to +50 mA, so that if you connect a 20 ohm load resistor to this output, a voltage value of -/+ 1 V (50 mA x 20 Ω)will be obtained.This voltage signal must be read by the ESP32 through the input terminal of channel 0 of analog/digital converter 1 (ADC1 CH0).However, already the voltage range allowed by the ADC is 0 V to 3.6 V, the sensor output voltage must be supplied to a conditioning stage to obtain positive voltages.

Signal conditioning stage

The signal conditioning stage was implemented using an LM324 operational amplifier, as indicated in Figure 4, powered by +5 V and configured as a voltage follower, or buffer, which uses the positive cycles of voltage. The ADC1 Channel 0 provides 12-bit accuracy, and the attenuation was not changed in the configuration, it is 0 dB, to allow the input voltage to be between 0 V and 1 V.

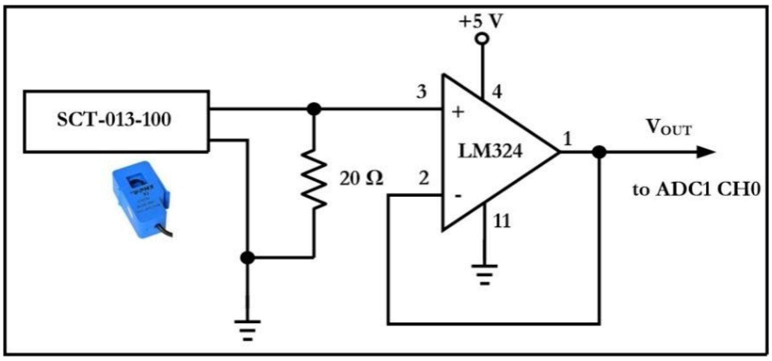

Figure 4. Signal conditioning stage

ESP32 programming was done using MicroPython. Initially, the program configures and activates the WiFi interface and the ADC and then determines the value of the electrical energy consumed.

When working with alternating current, the active, or real, power is calculated by taking the average of the product of the instantaneous values of voltage and current, because the instantaneous power changes with time and is difficult to measure. It is easier to determine the actual power by averaging the instantaneous power over a period of time using Formula 1.

$$P = \frac{1}{T}\int_0^T p(t)dt \quad (1)$$

Instantaneous power can be determined using RMS (root mean square) current, as it indicates the effectiveness of a current source in providing power to a load.RMS current, or Irms, is the current that supplies the same power, on average, to a load as its direct or direct current equivalent. Since the term effective value indicates the square root of the average of the square of a periodic electrical signal, or mean square value, in such a way that Formula 2 is used to calculate the value of the RMS current.

$$i = \sqrt{\frac{1}{T}\int_0^T i^2 dt} \quad (2)$$

In such a way that to obtain IrmsFormula 3 is used over a period of time, where N is the number of samples in the period.

$$i = \sqrt{\frac{1}{N}\sum_{n=0}^{N} i_n^2} \quad (3)$$

Thus, once Irms is determined, it is multiplied by the voltage value, 110 VAC/60 Hz in North America, México and some countries in America, to obtain the power magnitude.

To determine the value of the power at each moment, and based on what was indicated above, the program initializes the sum of the accumulated current to zero and enters a cycle in which it performs the following tasks: obtains the digital word provided by the ADC1 CH0 which indicates the value of the voltage present at the terminal of this channel, determines the value of the sensor current using the digital word, performing the necessary scaling, obtains the square root of the current and adds it to the sum. The loop tasks are executed 150 times, that is, 150 iterations. This is because the AC signal frequency is 60 Hz, 60 cycles per second. The average value of the current could be obtained in 30 cycles, every 0.5 seconds (1/60 Hz) * 30 = 500 milliseconds, however, the sensor current reading was performed every 100 milliseconds to have greater precision in the result, for which the program cycle consists of 150 iterations. Next, the square root of the average of the sum is obtained to determine the RMS current whose result is multiplied by 120 VAC to obtain the power every second.Finally, the program sends the power value every 30 seconds to the Internet platform ThinkSpeak.

ThingSpeak is an open source IoT Platform that allows receiving, visualizing and analyzing information in the cloud. IoT applications can be developed in which devices and sensors transmit data to the ThinkSpeak server which can be displayed on the ThinkSpeak portal or in a user interface. To use ThinkSpeak, a channel must be created with up to eight fields associated with it. The fields store the information transmitted by the sensors. ThinkSpeak assigns a channel ID, a WRITE_API_KEY and a READ_API_KEY that allow users to send or access the information received by the channel. You can transmit or write data to a channel through different mechanisms supported by ThinkSpeak: using HTTP calls from the REST API, MQTT Publish method or through MATLAB.

REST is a representational state transfer architecture that works under a request/response scheme using HTTP. MQTT (Message Queue Telemetry Transport) and uses the publish/subscribe model based on TCP/IP sockets or WebSockets. In this way, it is possible to send data to the channel via REST GET or POST request or using MQTT Publish and access the channel data using REST GET request or MQTT Subscribe. MQTT is the publish/subscribe standard messaging protocol used in the IoT.

Secure MQTT communication is done via SSL (Secure Sockets Layer). It is possible to update data to a ThingSpeak channel either using a REST GET or POST request or using MQTT Publish method. Channel data can be accessed via a REST GET request or MQTT Subscribe. In this work, MQTT was used to access the ThinkSpeak platform since it allows to transfer data to a channel quickly using limited bandwidth, few hardware resources and a minimum amount of code with small size MQTT messages.The operation of the MQTT protocol is based on a broker and a client. The broker is the central element that controls the reception and sending of messages. The client is the remote device, in this case the monitoring node, that publishes data or subscribes to topics to access the information. The client sends messages and publishes them to a certain topic and can receive messages subscribes from a topic.

ThinkSpeak has an MQTT API that was used in this work by invoking the *umqtt.simple* library from the MicroPython program of the network nodes. The API functions allow the MQTT client to update and receive updates from the channel via the ThingSpeak MQTT broker. The monitoring node connects to the MQTT broker URL mqtt3.thingspeak.com and port 1883, to publish to a channel or subscribe to updates from that channel. To send a publication from the nodes program, the following API functions were used: *client.connect()*, *client.publish(topic,payload)* and *client.disconnect()*, whose sequence of operation is indicated in Figure 5.

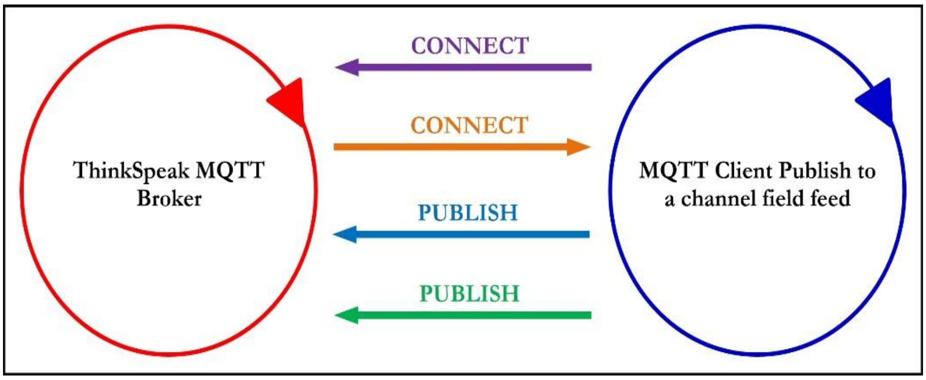

Figure 5. MQTT communication model used

3. Results and discussion

A set of tests was carried out on each of the four nodes.The servers where the nodes were installed are of different types, models and characteristics.The tests consisted of comparing the value of the power consumed by the server, determined by the node and transmittedto ThinkSpeak, against the magnitude of the powerindicated by a wattmeter used in the certification of data center services, employing this last measurement as the reference.500 measurements were made at each node at different times.Based on the difference between the node measurement and the reference measurement, the test results indicatedthat the four nodes have an accuracy of 1.253%, 1.234%, 1.117%, 1.176% and 1.154%, respectively. These values are slightly higher than theaccuracy indicated on the sensor data sheet which is +/-1%. Derived from the above, it is possibleindicate that the monitoring network has an accuracy of 1.186% on average.The accuracy can be adjusted in the node program, so that the measured energy consumption reported is equal to that obtained with the wattmeter.

To view the energy consumption measurement in real time at each node of the monitoring network, the Thingviewapp can be usedor access the ThinkSpeak web page from a browser. Figure 6 shows the main screen of Thingview where the four channels are indicated,one for each node, through which ThinkSpeak receives the information from the nodes.This figure also indicates the graphs of the amount of energy consumed by each monitored server. Figure 7 presents the four channels and the power consumption graphs of each node on the ThinkSpeak web page.

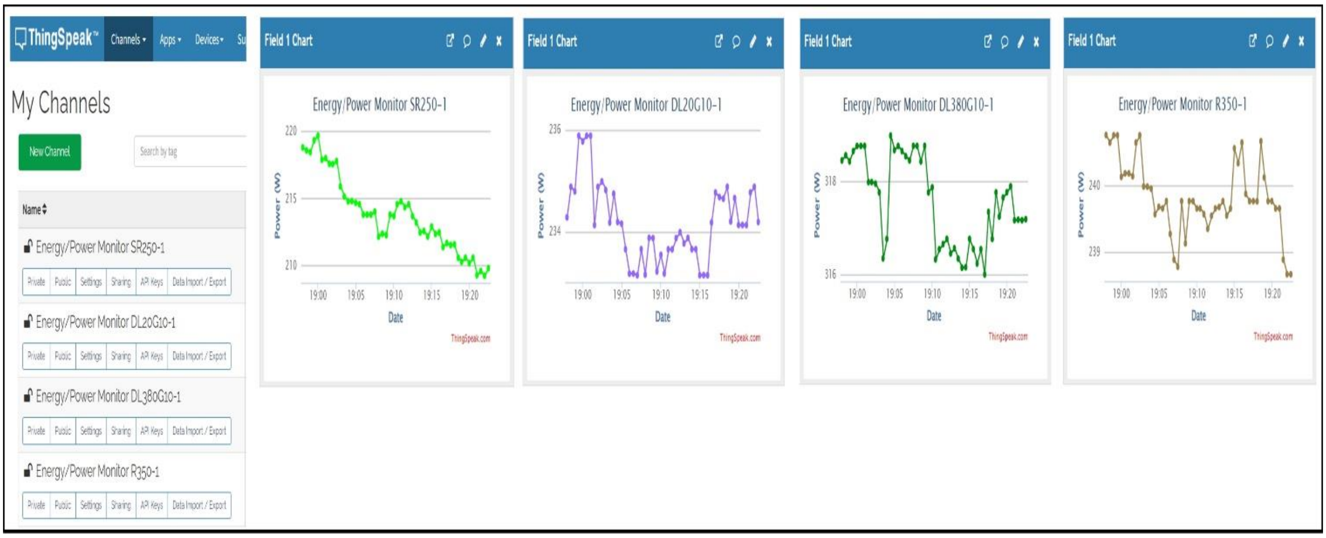

Figure 6. Results of the measurements obtained in the network nodes seen in ThikSpeak web page.

Figure 7. Results of the measurements obtained in the network nodes seen in Thingview

It is possible to download, or export, from the ThinkSpeak web page to a file in CSV format the information transmitted by each node of the network to the corresponding channel. This file is used by the data center application that determines the cost to be paid by the client, or user, of the monitored server. The amount of energy each server consumes is just one of the elements that indicate the cost above. The amount of other operating expenses must be included, such as cooling, security and lighting systems, among others.

3. Conclusion

A network of four nodes was obtained that monitors in real time the energy consumption of servers in a data center. The amount of energy is reported to the ThinkSpeak IoT platform from where it can be viewed remotely through the web page or in the Thingview app.Considering the results obtained, the data center decided to start with the development of the second version of the system to which more nodes and other types of sensors will be integrated. To extend the network, more nodes must be added with the same architecture as the existing ones. To integrate sensors to the nodes and monitor other types of variables, a field must be added, for each sensor, in the MQTT publications that the node sends to the ThinkSpeak broker.

If it is necessary to extend the wireless communication range of the network, in case the nodes are far from the Internet access point, the network design can be modified so that the nodes use a longer-range technology transceiver than WiFi such as Bluetooth Low Energy (BLE) or LongRange (LoRa). The cost of LoRa transceivers is higher than WiFi but distances of kilometers can be reached.

References

1. K. Kaur, S. Garg, G. Kaddoum, E. Bou-Harb and K. R. Choo. (2019). A Big Data-Enabled Consolidated Framework for Energy Efficient Software Defined Data Centers IoT Setups, IEEE Transactions on Industrial Informatics, 16(4), pp. 2687-2697.
2. K. M. U. Ahmed, M. H. J. Bollen and M. Alvarez (2021). A Review of Data Centers Energy Consumption and Reliability Modeling, IEEE Access, 9(1), pp. 152536-152563.
3. Y. Berezovskaya, C. -W. Yang, A. Mousavi, V. Vyatkand T. B. Minde (2020). Modular Model of a Data Centre as a Tool for Improving Its Energy Efficiency, IEEE Access, 8(1), pp. 46559-46573.
4. Z. Zhou et al. (2018). Fine-Grained Energy Consumption Model of Servers Based on Task Characteristics Cloud Data Center, IEEE Access, 6(1), pp. 27080-27090.
5. G. Zhang, S. Zhang, W. Zhang, Z. Shen and L. Wang (2020). Distributed Energy Management for Multiple Data Centers with Renewable Resources and Energy Storages, IEEE Transactions on Cloud Computing (Early Access).
6. H. Yuan, H. Liu, J. Bi and M. Zhou (2021). Revenue and Energy Cost-Optimized Biobjective Task Scheduling for Green Cloud Data Centers, IEEE Transactions on Automation Science and Engineering, 18(2), pp. 817-830.

7. C. Jiang, Y. Qiu, H. Gao, T. Fan, K. Li and J. Wan (2019). An Edge Computing Platform for Intelligent Operational Monitoring Internet Data Centers, IEEE Access, 7(1), pp. 133375-133387.
8. D. Ying and D. A. (2021). Hall, Current Sensing Front-Ends: A Review and Design Guidance, IEEE Sensors Journal, 21(20), pp. 22329-22346.
9. M. Crescentini, S. F. Syeda and G. P. Gibiino (2021). Hall-Effect Current Sensors: Principles of Operation and Implementation Techniques, IEEE Sensors Journal (Early Access).
10. A. Itzke, R. Weiss and R. Weigel (2019). Influence of the Conductor Position on a Circular Array of Hall Sensors for Current Measurement, IEEE Transactions on Industrial Electronics, 66(1), pp. 580-585.
11. X. Duan, B. Li and W. Zhao (2020). Energy Consumption Minimization for Near-Far Server Cooperation NOMA-Assisted Mobile Edge Computing System, IEEE Access, 8(1), pp. 133269-133282.
12. X. You, X. Lv, Z. Zhao, J. Han and X. Ren (2020). A Survey and Taxonomy on Energy-Aware Data Management Strategies Cloud Environment, IEEE Access, 8(1), pp. 94279-94293.
13. R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy and Y. -C. Tian (2018). Adaptive Energy-Aware Algorithms for Minimizing Energy Consumption and SLA Violation Cloud Computing, IEEE Access, 6(19), pp. 55923-55936.
14. Z. Zhou, M. Shojafar, M. Alazab, J. Abawajy and F. Li (2021). AFED-EF: An Energy-Efficient VM Allocation Algorithm for IoT Applications a Cloud Data Center, IEEE Transactions on Green Communications and Networking, 5(2), pp. 658-669.
15. Z. Ding, Y. Cao, L. Xie, Y. Lu and P. Wang (2019). Integrated Stochastic Energy Management for Data Center Microgrid Considering Waste Heat Recovery, IEEE Transactions on Industry Applications, 55(3), pp. 2198-2207.
16. W. Zhang, R. Yadav, Y. -C. Tian, S. K. K. Sah Tyagi, I. A. Eelgendy and O. Kaiwartya (2022). Two-Phase Industrial Manufacturing Service Management for Energy Efficiency of Data Centers, IEEE Transactions on Industrial Informatics (Early Access).
17. B. Gul et al. (2020). CPU and RAM Energy-Based SLA-Aware Workload Consolidation Techniques for Clouds, IEEE Access, 8(1), pp. 62990-63003.
18. C. Guo, K. Xu, G. Shen and M. Zukerman (2021). Temperature-Aware Virtual Data Center Embedding to Avoid Hot Spots Data Centers, IEEE Transactions on Green Communications and Networking, 5(1), pp. 497-511.
19. H. Yeganeh, A. Salahi and M. A. Pourmina (2019). A Novel Cost Optimization Method for Mobile Cloud Computing by Capacity Planning of Green Data Center With Dynamic Pricing, Canadian Journal of Electrical and Computer Engineering, 42, 1(1), pp. 41-51.
20. W. Lin, G. Wu, X. Wang and K. Li (2020). An Artificial Neural Network Approach to Power Consumption Model Construction for Servers Cloud Data Centers, IEEE Transactions on Sustainable Computing, 5(3), pp. 329-340.
21. R. Weiss, A. Itzke, J. Reitenspieß, I. Hoffmann and R. Weigel (2019). A Novel Closed Loop Current Sensor Based on a Circular Array of Magnetic Field Sensors, IEEE Sensors Journal, 19(7), pp. 2517-2524.
22. C. Floridia et al. (2022). An Improved Solution for Simultaneous Measurement of Current and Temperature on Terfenol-D FBG Optical Sensor, IEEE Sensors Journal, 22(1), pp. 357-364.
23. S. Kulkarni and D. Divan (2021). An Edge-Intelligent, Clip-on Rogowski Current Sensor With Wide Dynamic Range, IEEE Sensors Journal, 21(2), pp. 1059-1071.

