Kurdistan Journal of Applied Research (KJAR)
Print-ISSN: 2411-7684 | Electronic-ISSN: 2411-7706
Volume 3 | Issue 2 | December 2018 | DOI: 10.24017/science.2018.3.4

Received: 14 October 2018 | Accepted: 20 November 2018

A Novel Cloud-Based Temperature Monitoring Service to Datacenter Environment

Najmadin Wahid Boskany
Department of Computer
College of Science
University of Sulaimani
Sulaimani, Iraq
najm.wahid@univsul.edu.iq

Mustafa Ibrahim Khaleel
Department of Computer
College of Science
University of Sulaimani
Sulaimani, Iraq
mustafa.khaleel@univsul.edu.iq

Miran Taha Abdulla
Department of Computer
College of Science
University of Sulaimani
Sulaimani, Iraq
miran.abdullah@univsul.edu.iq

Abstract: ***As temperature-efficient of cloud datacenters becomes one of many challenges faces the engineers in today's business environment, virtualized monitoring the resources over these datacenters is the cloud provider's interest in a way the providers can control their temperature remotely anywhere around the world. Techniques proposed by existing works are focused on executing many cloud modules on less cloud-server which cannot be practical because massive volume of cloud modules will be scheduled by cloud tenants. This motivated us to present a novel solution and propose a real-time temperature aware for cloud network datacenters. The proposed paradigm is constructed based on defining dual thresholds per datacenter to control the level of temperature. Moreover, we have set the upper threshold to 25 degree and the lower one to 15 degree. Surpassing these values alerts network administrator through cloud-based software service to operate the cooling unit(s). The microcontrollers that have been incorporated are both Arduino and GSM Shield. The simulation results showed that our heuristic had the ability to control the temperature per datacenter approximately 63%.***

Keywords: Cloud monitoring, Temperature awareness, Arduino microcontroller, Cloud underlying network.

1. INTRODUCTION

Growing the large-scale of computation modules executed on cloud servers has increased. Seeking for an effective cloud scheduler to achieve optimal network performance has been attracting provider desire to establish many real network deployment experiments in cloud datacenters [1]. Hundreds of thousands of inter-networks connected servers are operated inside these datacenters in order to retrieve requiring instants made by cloud tenants. However, troubleshooting and maintaining the temperature of these datacenters has become crucial in cloud network system [2] which motivated provider to call for some novel real-time monitoring techniques to adjust the level of temperature to avoid system disturbed. Factors such as number of racks, number of servers, and electricity power cloud raise the temperature of a datacenter. Several heuristic algorithms have been proposed to cumbersome the problem of lowering down the temperature of datacenters. Most of these paradigms had the ability to adjust the measure of temperature locally. With advanced cloud network techniques, today, monitoring the temperature of these servers can be done remotely using the concept of cloud web-based and wireless-sensor temperature monitoring systems [3]. In this paper, we present a novel solution and propose a real-time temperature aware for cloud network datacenters. The proposed paradigm mainly optimizes three objectives. In the first step, a dual threshold has been predefined to maintain the level of temperature and keep the values between these two thresholds. In the second step, a message will be directed to the network administrator in case of violating either the upper threshold or the lower threshold. Moreover, the statistics data will be reported and directed to the cloud-based storage. Finally, based on the necessities of datacenter, a cooling unit or units will be initiated to lowering down the temperature of the computational servers. The simulation is based on SIM900 Quad-band GSM/GPRS and Arduino microcontrollers. The temperature of the server room will be sensed by Arduino sensor and when it reaches the upper threshold, which we set it to 25 degree, two actions will be placed; first an alter message will be redirected to the administrator through SIM900 Quad-band GSM/GPRS microcontroller and then an operate signal will be sent to the cooling units. However, the same situation occurs for the lower threshold. When it surpasses 15 degree, the same two messages will be redirected while in this time is to tear down the cooling units. The simulation results showed that our heuristic had the ability to control the temperature per datacenter approximately 63%.

2. RELATED WORK

The problem of adjusting the temperature of datacenters has been addressed by many research works. Some of these studies proposed ideas based on scheduling cloud loads onto less computing hosts as much as possible to decrease the level of generated heats by these centers. However, these solution cloud give a tremendous benefit when cloud tenants schedule moderate modules. But the issue arises when these assigned jobs get heavier. One of the most advanced techniques to overcome such concern is to define multi-threshold per cloud inter-network servers and keep the rates of the temperature between these two thresholds for system reliability. According to [4], more than 60% of the energy consumed in the datacenter is wasted by the cooling system in inefficient ways. So, a wireless sensor prototype was presented to enhance the energy consumption per datacenter and improve the system performance. The proposed system was constructed based on Arduino open source microcontroller integrated with XBee RF module which was programmed to operate with the ZigBee mesh network systems [5]. The problem of mismatching between the requirement of IT equipment and the supplement by the facility fans is discussed to overcome the issue of cooling inefficiency [6]. Two virtual sensors namely, volumetric airflow and outlet temperature sensor, are proposed to decease the heat generated by servers. Both sensors are exposed to the Intel Data center Manager (DCM) using IPMI commands. A web-based temperature monitoring system was presented to allow the administrator to follow the temperature rates for the server rooms via any node with data connection [7]. When the temperature reaches undesired value, the system automatically will direct a text message to the administrator. A microcontroller was programmed using MP LAB IDE to justify the temperature of the room [8]. The proposed system allows the administrator to input different temperature ranges through four-key keypad. Interestingly, when the temperature surpasses the heat threshold, the Opto-coupler will switch the room heater to AC and vice versa. A new paradigm has been proposed in [9]. The technique predicts the temperature level of cloud-based virtual machines through cross-cutting datacenter infrastructure parameters such as server capacity and initiating cooling system.

3. PROPOSED CLOUD FRAMEWORK

Cloud tenants submit their modules within the QoS requirement to the cloud scheduler to be dispatched over cloud resources. This massive of scheduled modules need instant process which in turn heats up the level of temperature for executing VM-based units. Our heuristic architecture includes four main entities. The cloud application, cloud meta-mapper, cloud scheduler, and cloud underlying hardware.

3.1 Cloud Software Multitenancy: Within this software architecture, a single instance of cloud network software initiates over a server and multiple cloud-based service users. However, users submit their module requests (i.e., single task, DAG- structured workflow applications, QoS requirements, etc.) from anywhere around the world to the Cloud-meta mapper.

3.2 Cloud Network Meta-Scheduler: Meta-Scheduler (Mapper) considers as an interface between Cloud-based tenants and Cloud-based infrastructure to enable utilization efficient services. This includes satisfying requirements such as tenants specified deadline, transfer user's demands to the most temperature-aware datacenter and keep monitoring both energy consumption for the cloud hardware and cooling system.

3.3 Cloud Local Scheduler: receives modules from the main meta-mapper and dispatch them over cloud resources when they are inter-network connected. However, these schedulers are operating based on both Arduino and GSM shield microcontrollers.

3.4 Cloud Network Underlying Hardware: handers of thousands of servers are operated in this based infrastructure. The temperature consumption from these servers will be sensed by LM35 Arduino temperature sensor. Dual temperature thresholds have been predefined per room and monitored by cloud local scheduler to keep the temperature status in steady level.

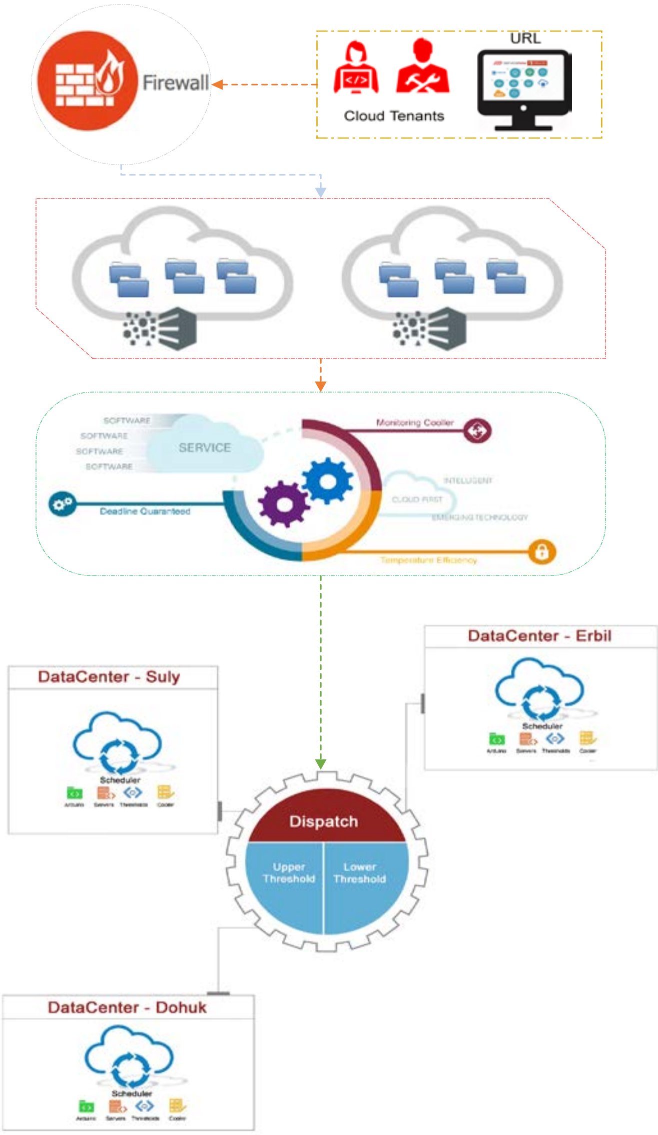

Figure 1: Proposed cloud system framework [14]

4. METHODOLOGY

The massive scheduled workloads by cloud consumers increase the rate of energy consumption by cloud data centers which in turn affects not only the system utilization interest but also the environments. Based on aforementioned facts, cloud providers are highly required to design an automated monitoring system to adjust the layer of temperature generated by the servers within a datacenter. Our intention is to define dual-thresholds, the upper and lower thresholds, to maintain the temperature consumption at steady level. To attain such objectives, our paradigm periodically senses the temperature consumed by cloud resources through LM35 temperature sensor circuit. The values are directed to the Arduino microcontroller to either initiate the cooling units or tearing down them. When the temperature surpasses the upper-thresholds, 25^0, a signal will be directed to turn on the cooling systems. However, this system will be turned off when the temperature falls down under lower threshold, 15^0. This back and forth will save a tremendous amount of energy. Network administrator will be alerted in both cases, when the temperature surpasses the upper threshold or falls down lower threshold, via SIM900 GSM module. The statistic values of these variations are directed to a database server for further processing. As mentioned before, two main microcontrollers have been conducted in this work namely ATmega328 microcontroller which consumes low power [10] and a SIM900 Quad-band GSM/GPRS microcontroller has been applied to notify cloud admin via portable node by sending short message using GSM network.

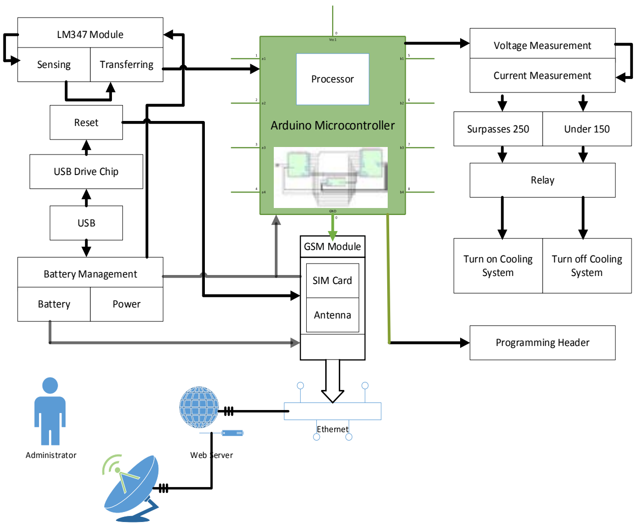

Figure 2: Microcontroller components

Moreover, we also adapted a temperature sensor namely LM35 to monitor the temperature of established cloud rooms. This type of sensor prior to the others by having more storage capacity and ability to instant process [10]. The LM35 does not require any external calibration or trimming to provide typical accuracies of +_-1/40C at room temperature and +- 8/40C over a full -55 to +1500C temperature range. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. The LM35 thus has an advantage over linear temperature sensors calibrated in 0 Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 is rated to operate over a -550 to +1500 temperature range [7].

5. SYSTEM FLOW CHART

The proposed cloud system model first set its parameters (i.e.; reading sensor input). Then temperature values continually sent to file in attached cloud web server, if received temperature value exceeded a certain threshold which is equal to 25C (i.e.; Yes in flow chart), it means the sensor received high temperature value, then microcontroller sends signal to activate alarm (buzzer and LED), turn on cooling fan, and send notification through SMS to admin's mobile phone. Otherwise, if the temperature value is under 15C value then microcontroller sends signal to turn of cooling fan in order to reduce power consumption. The flow chart of the system is provided in figures 3. Therefore, the message flow includes an input node that provides the messages that are processed in the system, as shown in figure 4, it explains that the connected temperature sensor to the Arduino which sends data to the equipment, this is in order to obtain the information of datacenter's temperature then the information is sent to cloud computing to monitor, when the system detects abnormal information, the system sends a message to Arduino to active the cooling system.

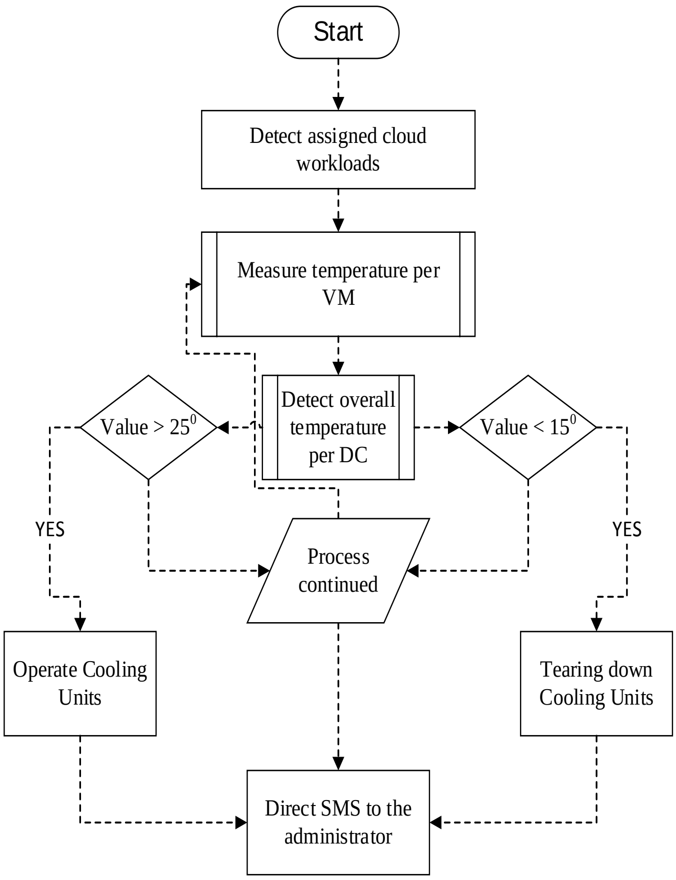

Figure 3: Cloud system flowchart

6. ENERGY CONSUMPTION MODEL

According to [11-14], Essentially, a watt-minute/hour meter is designed to measure energy of power consumed within a period of time. It can be shown as Energy = Power * Time. In simple terms, the average of power can be calculated by the following equation:

$$w = \frac{1}{T}\int_0^T (v * i)\, dt \qquad (1)$$

Where: v is instantaneous voltage, i is instantaneous current, t = time and T= period. The method used to measure power, which is taken from the basic definition of power equation. The integration is performed by numerical method, where integration can be depicted in equation 2.

$$w = \frac{1}{N}\sum_{1}^{N} vi * i_i \qquad (2)$$

Means that taken from the instantaneous values and then divide it by a number of times to get the average power which equals the real power. The system can measure instantaneous voltage and current, and real power.

7. ENERGY CONSUMPTION ALGORITHM

In this section, we explain the parameters are used to provide the decision on proposing our algorithm for reducing power consumption. Generally, the operating areas of the electric motor can be classified into three levels of power consumption; when the cooling system starts to rotate, which consumes higher energy (Startup time: time of higher consumption). The operation time, which is the best performance of the system (Operation time: time of controlling rotation of fan speed) and turnoff time, which is the fan stopped to rotate or very slow rotating (Off time: time of low consumption or nothing). Therefore, the parameters of our proposed algorithm include; maximum and minimum temperature values, startup time of rotating the fan while cooling the datacenter, controlled rotation of the fan and sleep time of the fan. In order to consider power saving relative to upper and lower level of temperature, the values of 25 and 15 are selected as maximum and minimum temperature value for the datacenter environment. When temperature is greater than maximum threshold, the fan starts up to rotate in the system however the fan is stopped when the temperature under the minimum threshold. In order to reduce power consumption, the operation time of rotating speed of cooling system is divided into three series of rotation such as high rotation, medium rotation and stop rotation. Startup time of the fan is based on rotating from low speed to high speed. High rotation and medium rotation related to the availability of a certain temperature degree. Pseudo code 1 explains the process of energy consumption related to fan speed. Moreover, revolution value of fan speed is determined in different scales, which is provided better efficiency for power consumption in the system.

8. PERFORMANCE EVALUATION

In this section, we explain the approach is taken to obtain data and evaluate the results. The complete hardware prototype of cloud cooling datacenter system is based on monitoring the temperature of these cloud datacenters to save extra energy consumption. In order to provide the prototype, we employed Arduino Uno microcontroller, processing unit, fan as system cooling, and relay to generate high volte. Within a datacenter, the cooling system attached to Arduino to turn on/off automatically when the temperature of the system reaches a threshold value. The system can detect the maximum and minimum temperature of the datacenters current values. Generally, underlying cloud hardware is made up of these components; the main cloud servers, which provide cloud web service, cloud tenant's application and GSM service. The cloud web server is the unit responsible for holding all temperature data. It is developed using mainly Apache server and PHP programming language. However, cloud tenant's application provides accessing the system from a remote location. It provides a Graphical User Interface (GUI) for client's operation to facilitate excellent user-friendly experience to monitor the environment's temperature. GMS service can deliver alert messages to the clients in emergency case. Accordingly, the experiments phases are explained as follows.

Pseudo code1, description of the proposed algorithm

```
Begin
float Temp; \\ Tempreture
int Fan_rotation;
void setup()
Fan_rotation = 0;
Void_loop ()
If (Temp > =25) then
Fan_rotation = 59; \\ low revolution
sleep (10000)        \\ waiting time
Else if (Temp > 25)
Fan_rotation = Fan_rotation + 1; \\ increase revolution
Else if  (Temp > 20 && Temp < 25 )
Fan_rotation = 59;  \\decrease revolution
Fan_rotation = Fan_rotation -1; \\descrease revolution
Else if  (Temp < 25 )
Fan_rotation = 0;  \\ off time
End
End
```

8.1. Temperature monitoring

In this experiment, in order to monitor the accurate temperature value of cloud datacenter, three types of temperature sensors have been installed to monitor temperature of the cloud datacenter. This includes LM35, MAX665 and TMP36. Environmental temperature is monitored to show stability and precise of values. In this case, these types of sensors are connected to three Arduinos separately; Arduinos have same characteristics and versions. The first phase of the experiment is shown in figure 5. In this layer, we monitor the temperature when the degrees have the highest values. The red line depicts LM35 sensor, blue line corresponds to MAX6675 and green line presents temperature of TMP36 sensor. According to the experiment in figure 6, diversity of the both sensors MAX6675 and TMP36 is higher than LM35 where LM35provides stable recording temperature value when the environment temperature of the datacenter is changed according to variability of devices temperature. The next phase of the experiment is related to monitoring the temperature of the cloud datacenters using aforementioned sensors. As explained in figure7,at the beginning, the sensors are recording the lower ranges of the temperatures, then the temperature of the environment raise up to high range. According to this experiment, the blue line depicts the temperature value of LM35, red line presents MAX6675 and green line presents TMP36 respectively, the blue line shows better stability of

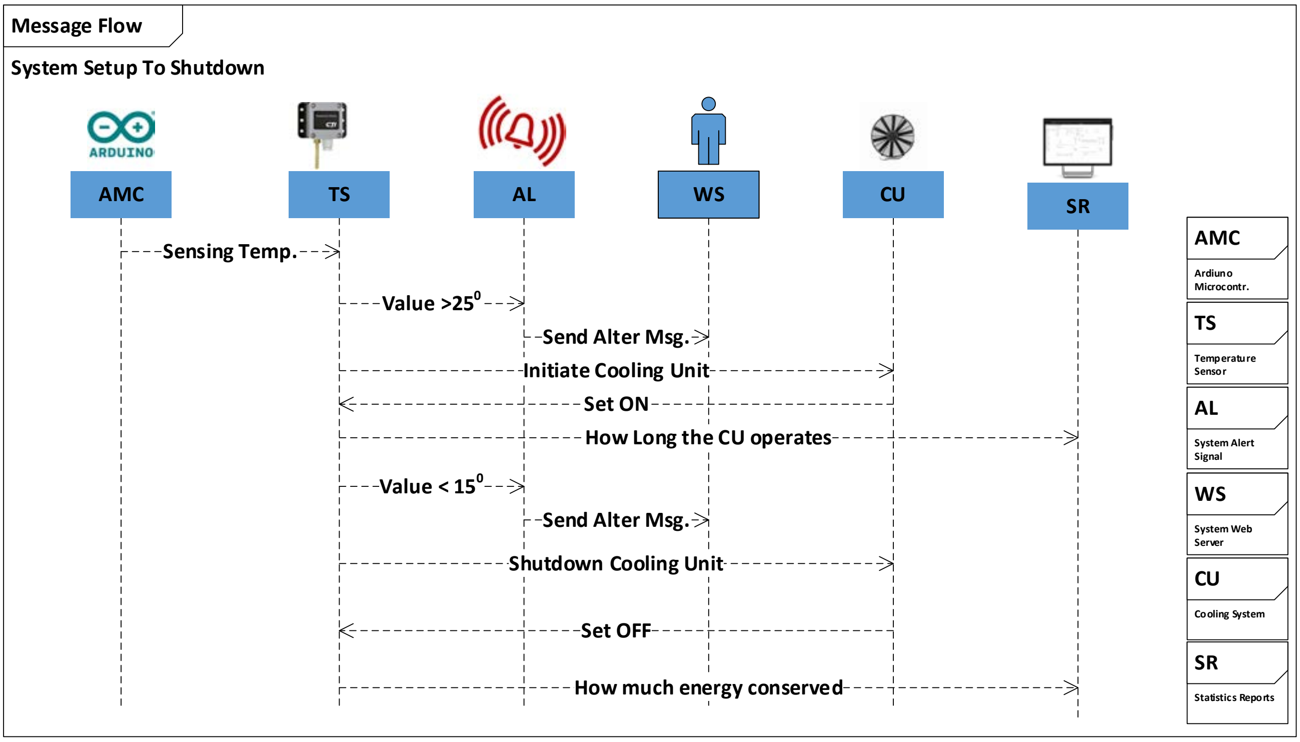

Figure 4: Flow message procedure of the system.

recording the environment temperature then other sensors.

8.2. Energy Consumption

In order to present the performance of our proposed cloud heuristic in terms of power consumption [15], we have evaluated some measures to show the comparison between our proposed algorithm and conventional approach for cooling system in cloud datacenter. The scenario states as follow. 4 pins fan are boxed with 12-volte input where PWM (Pulse Width Modulation) control is used to control fan speed. The fan speed is related to the temperature of cloud execution units. Therefore, the fan speed records values in percentage; when the fan reaches its high percentage, this means fast rotation and low percentage, which presents slow rotation of the fan system cool. According to the algorithm, the cooling system will be on when the temperature reaches to 25 C°, then the relay will be activated, and the fan starts to rotate. Therefore, speed of the fan will be changed according to temperature of the datacenter. The fan speed rotates to high percentage it can be estimated 59-99% when temperature between 20 -25 C° and for the temperature under 18 C° the rotate can be estimated to 59-0%. 0 means the fan stops rotating. Difference speeds are purposed to provide energy consumption. For this experiment, 30 seconds are taken as shown in Figure8. When the system booted up, the current temperature is pointing in high, and the fan speed is rotating 90% duty cycles. When the temperature reaches to lower degree, fan's rotation is slowly decreased to close to 0. As demonstrated in the figure, red line shows temperature of the environment, at beginning of the scenario the temperature is 23.3 C° and then it raised to 24 C°, when the temperature reaches to 24.9 the fan starts to rotate to cool the equipment. Accordingly, green line shows proposed approach, rotation of the fan is stopped at 12thto 16thand this duration is less in traditional approach which is 13th to 15th Therefore, in traditional approach high percent of rotation is used to cool the equipment as shown at times; 10th and 16th. In the last experiment, energy consumption is observed according to sort of dramatic mount of fan's speed for the cooling system as shown in figure5. Green line presents proposed algorithm and red line presents traditional approach, the observation is done for one hour, it is aimed to show accurate observation and efficient consumption of power at long period. As depicted in the figure the green line shows lower range of energy consumption at 14th min. to 24th min. and 32th min. to 36th min. the ranges are different in traditional approach, which are shorter to save energy. As a result, the proposed algorithm is presented better efficiency of power consumption in watts than traditional approach.

8.3. Benchmark comparison

In order to show the benchmark comparison between our proposed algorithm and another recent work on reducing energy consumption for cooling system in the datacenters, we consider some parameters in order to show the comparison. The parameters are included; maximum and minimum temperature values are considered for experiments, rotation speed of the cooling system, and sleep time of the fan. Selecting all these parameters is related to our contribution as aforementioned in the proposed algorithm for energy consumption.
According to the paper [16], the considered maximum and minimum temperature respectively is 25 C° and 45 C°, and Fan speed revolution is categorized into four levels: 25%, 50%, 75% and 100%. When the temperature between 30 C° and 25 C° fan speed is rotated 25%, furthermore the temperature between 40 C° and 30 C° the fan speed is rotated 50%, moreover the temperature between 40 C° and 45C° the fan speed is rotated 75%, on the other hand, temperature greater than 45 C° the fan

speed is 100%. According to the results obtained in test experimental as shown in Fig. 7 and Fig. 8, our algorithm is provided better approach to reduce power consumption than [16]. Although, three parameters such as control of fan revolution and startup time and off time are not considered in [16] to reduce power consumption.

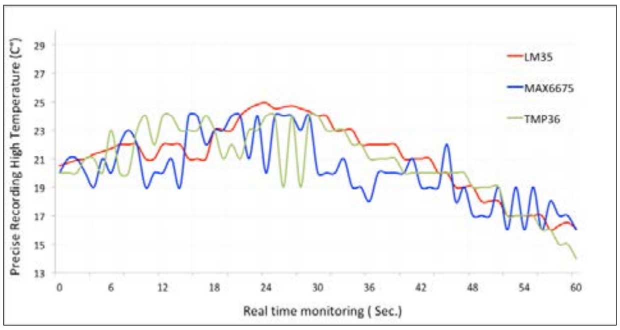

Figure 5: High temperature sensing

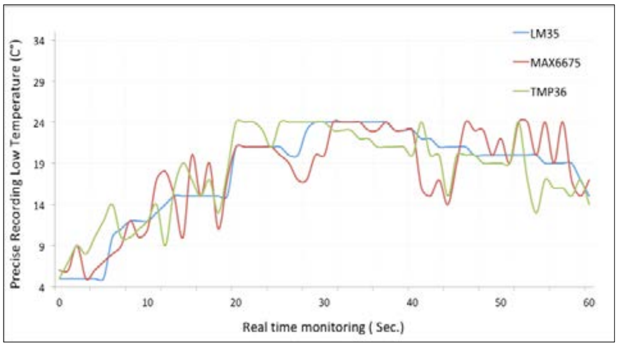

Figure 6: Low temperature sensing

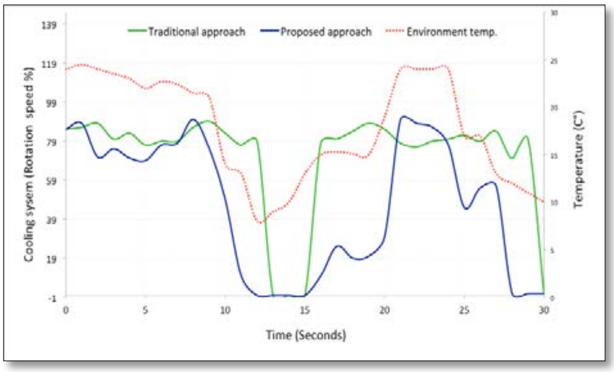

Figure 7: Cooling system rotation

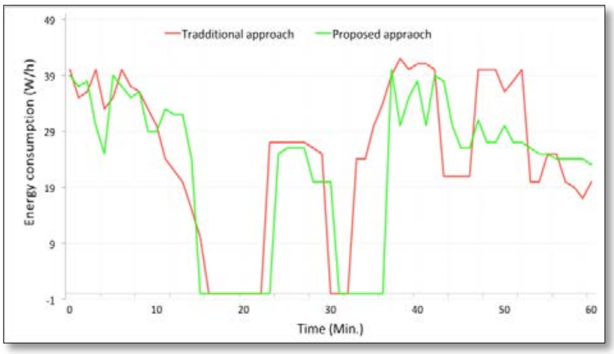

Figure 8: Underlying energy consumption

9. CONCLUSION

Temperature-aware workload management in cloud infrastructure has been shown promising in maximizing utilization rates of cloud resources. The massive assigned cloud tenant demands require providers to establish energy-efficient cloud datacenters to have remotely access to resources such as servers, storages, and software. Monitoring the temperature of these resources is challenged engineers and designers to come up with schedulers that can increase system throughput and decrease energy consumption. This paper is proposing a smart hybrid algorithm that takes into account both over and underutilized cloud resources computation. In this paradigm a dual threshold is predefined to control the level of power consumption and continue to stable the system performance. The upper threshold has been set to 25 degree while the lower one has been set to 15 degree. Therefore, the proposed energy consumption algorithm provided better-reduced power consumption for cooling system in datacenters. Surpassing any of these two values results in directing an automatic message to the network administrator to take action and control the environment whether by switching on or off the cooling systems. Two main microcontrollers have been conducted in this simulation namely Arduino and GSM Shield. In future work, we will work on a utilizing machine learning to design the algorithm to provide energy efficient for system cooling.

REFERENCES

[1] M. Khaleel and M. M. Zhu, “Energy-Aware Job Management Approaches for Workflow in Cloud.” IEEE International Conference on Cluster Computing,, Chicago, IL, USA,2015, pp. 506-507. Doi: 10.1109/CLUSTER.2015.85.

[2] M. Khaleel and M. M. Zhu., “Energy-efficient task scheduling and consolidation algorithm for workflow jobs in cloud.” International Journal of Computational Science and Engineering, Vol. 13, No. 3, pp. 268-2. Doi: 10.1504/IJCSE.2016.078933.

[3] W. Theophiluand S.Bhudi, “A Microcontroller based Room Temperature Monitoring System.” International Journal of Computer Applications,Vol. 53, no. 1, 2012.

[4] M. G. Rodriguez, L. E. Ortiz Uriarte, Y. J.Kazutomo Yoshii, R. Ross and P. H. Beckman , “Wireless sensor network for data-center environmental monitoring.” Fifth International Conference on Sensing Technology. IEEE, 2011. pp. 533-537.

[5] N. Ahuja, “Real time monitoring and availability of server airflow for efficient data center cooling.” En Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), 29th Annual IEEE. IEEE, Almerston North, New Zealand 2013, pp. 243-247.

[6] R. Solorio “A Web-Based Temperature Monitoring System for the College of Arts and Letters”, Electronic Theses, Projects, and Dissertations, Vol. 129, 2015, doi:https://scholarworks.lib.csusb.edu/etd/129.

[7] Wu, X. Li, P. Garraghan, X. Jiang, K. Ye and A. Y. Zomaya, "Virtual Machine Level Temperature Profiling and Prediction in Cloud Datacenters," 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, 2016, pp. 735-736.doi: 10.1109/ICDCS.2016.62

[8] A.L. Amoo, H. A. Guda, H. A. Sambo, and T. L. G. Soh, “Design and implementation of a room temperature control system: Microcontroller-based.” IEEE Student Conference on Research and Development, BatuFerringhi, Malaysia, 16-17 Dec. 2014.,pp. 1-6.doi: 10.1109/SCORED.2014.7072989.

[9] L.Curtis, “Environmentally Sustainable InfrastructureDesign, The Architecture Journal. Microsoft” , Vol. 18. 2008.

[10] F. K. Handhal and A.Turky Rashid“Design and building a single-phase smart energy meter using Arduino and RF communication

system.",The 3rd International Scientific Conference for Renewable Energy, Iraq, 2018.
[11] A.ChalakShakir and N. WahidBoskany, "Design and Implementation of Counter System using Arduino Wireless Motion Sensor."International Journal of Advanced Scientific and Technical Research, Vol. 3, no. 3, 2015.
[12] Available online on http://www.rspublication.com /ijst/index.html ISSN 2249-9954. Last accessed on June 10 2018.
[13] N. Wahid Boskany and, R. M. Abdullah, "Intelligent Anti-Theft Car Security System based on Arduino and GSM network". International Journal of Multidisciplinary and Current Research, Vol. 4, 2016, pp. 538-541.
[14] M. I. Khaleel, "Load Balancing and Thermal-Aware in Geo-Distributed Cloud Data Centers Based on Vlans." Science Journal of University of Zakho, Vol. 6, no 3, 2018, p. 113-117. doi:https://doi.org /10.25271/sjuoz.2018.6.3.515
[15] L. Garcia, J. M. Jiménez, M. Taha and J. Lloret, " Wireless Technologies for IoT in Smart Cities". Network Protocols and Algorithms, Vol. 10, No. 1, 2018 pp. 23-64,doi: 10.5296/npa.v10i1.12798
[16] J. M.Kumaran,I.Vikram,S.Kishore Kumar,R.Rajesh Kumar and S.Lokesh, "Design of An Automatic Fan Speed Controlling System Using Arduino UNO", International Journal of Intellectual Advancements and Research in Engineering Computations Vol. 6 No. 2, 2018.