A Novel IoT-Based Energy Management System for Large Scale Data Centers

Conference Paper · July 2015

DOI: 10.1145/2768510.2768520

Sharad Shriram
Delft University of Technology

A Novel IoT- Based Energy Management System for Large Scale Data Centers

Sharad S
Dept. of Computer Science & Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
0422 - 268 5000
sharad.sriram@gmail.com

Bagavathi Sivakumar P.
Dept. of Computer Science & Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
0422 - 268 5000
pbsk@cb.amrita.edu

V. Anantha Narayanan
Dept. of Computer Science & Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
0422 - 268 5000
v_ananthanarayanan@cb.amrita.edu

ABSTRACT
The high energy consumption in data centers is becoming a major concern because it leads to increased operating costs and also, pollution, as fuel is burnt to produce the required energy. While many techniques and methods have been proposed by various organizations and researchers to minimize the energy consumption, there has been considerably less work done in making a smart-energy management system that is capable of collecting the data available and make decisions based on the energy consumption patterns. In this work, a smart system is proposed that uses Internet of Things to gather data and a machine learning algorithm for decision making.

Categories and Subject Descriptors
B.1.3 **[Hardware]:** *Control Structure and Microprogramming - Control Structure Reliability, Testing, and Fault-Tolerance*

Keywords
Large Data Centers, High Energy Consumption, IoT, wireless communication, Cloud; Beagle Bone Black, CC2500.

1. INTRODUCTION
In the information age, we have more data that is generated than ever in history. These data are stored and served by data centers. These data centers are the back-bone of the growing economy of the world. The number of data centers across the world is growing very fast and these data centers consume electricity as much as the amount needed for a small town. This high consumption has lead data centers be called as "polluters" - because of the large amounts of fuel is burnt to produce electricity which results in higher CO2 Emissions. Thus, an efficient system [2,3] is needed to manage and monitor the electricity consumed. The proposed system uses efficient algorithms and protocols that not only saves power, optimizes the trust and reliability but also enables real-time data sharing and also make a co-existing ecosystem of different devices within a single network - Internet of Things [1,5], ready.

e-Energy'15, July 14-17, 2015, Bangalore, India.

DOI: http://dx.doi.org/10.1145/ 2768510.2768520

The paper is organized as follows : Secion II, is a study on the energy consumption at data centers. In section III, the scope and related works are discussed. In Section IV, we discuss the application of IoT and Machine Learning. In Section V, we dicuss our proposed solution. The hardware and software implementation with pseudo codes and algorithms are discussed in section VI. Results are discussed in Section VII. Conclusion and the scope for further work are presented in Section VIII.

2. A STUDY ON ENERGY CONSUMPTIONS IN DATA CENTERS
Data centers face under-utilization of servers as a major hurdle to better energy efficiency. There is a loss of about 13% in the conversion from AC to DC. Moreover, the servers need to be kept in an air-conditioned environment, which are known to consume more power. Thus, we need a system to smartly monitor and manage the power consumed and minimize the loss and wastage [8]. The growing footprint of data centers with respect to electricity consumption is shown in figure1 [7].

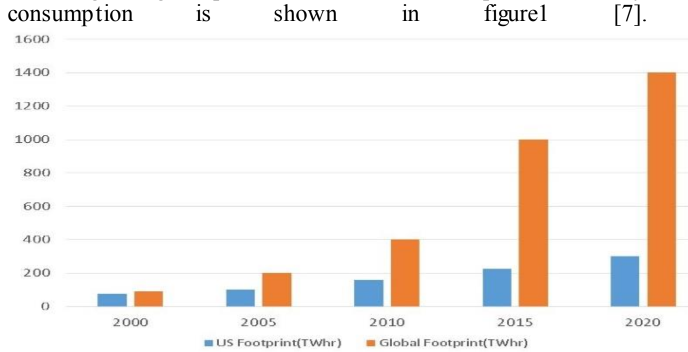

Figure 1. Projected data center electricity usage.

Natural Resource Defense Council (NRDC), is an international enivronment advocacy group which is based in the USA[11]. They have conducted a study on the energy consumed by data centers in the US and the details are shown in Table 1. Their study shows the estimated increase in terms of End-use Energy, Electricity Bill, the number of power plants needed and the carbon dioxide emissions.

Table 1. Study done by NRDC on the energy consumed by data centers in the US [6]

Year	End-Use Energy (B kWh)	Electric Bill (US ,$B)	Power Plants(in 500MW)	CO2 Emission (in million MT)
2013	91	9.0	34	97
2020	139	13.7	51	147
2013-2020 increase	47	4.7	17	50

3. SCOPE and RELATED WORK

There are many methods that are employed by organizations to minimize the energy consumptions at data centers and many researchers have also contributed with efficient ways to minimize energy consumptions [12-14]. The most common methods include using energy-efficient power supplies and fans, optimizing the data center layout and use optimized compilers, etc. [10]. The following are the practices that are suggested by Google [9] to reduce the energy consumption at datacenters.

1. Manage PUE (Power Usage Effectiveness):

The PUE, is an industrial measure used to reduce the energy consumed by non- computing functions like cooling systems, etc.

$$\text{PUE} = \frac{\text{IT Equiment Energy} + \text{Facility Overhead Energy}}{\text{IT Equipment Energy}}$$

Where:

- IT Equipment Energy – energy consumed by the servers, storage & networking devices and devices where the actual work is done
- Facility Overhead Energy – energy consumed by the lighting, cooling units and power distribution systems

2. Manage Airflow:

Good air flow management is crucial for the data center to operate efficiently. A well designed containment minimizes the mixing of hot and cold air. Furthermore, one can use Computational Fluid Dynamics to characterize and optimize air flow.

3. Adjust Thermostat:

Many manufacturers now allow servers to be run at 26 C, or more. The use of economizer allows you to run elevated cold aisle temperatures and offers higher savings.

4. Use free cooling:

"Free Cooling", refers to the process of removing the heat without a chiller. The proposed methods here are using low temperature ambient air, evaporating water, etc. It is seen water and air-side economizers can also be used.

5. Optimize power distribution:

Power distribution losses can be minimized by eliminating as many power conversion steps as possible by using efficient transformers & power distribution units. The UPS results in the largest loss of power, so an efficient model is to be used.

As discussed above, there is a need for monitoring the energy consumption over a long period of time. In Google's case the energy consumed is measured quarterly and trailing – 12 month performance of the data center. In our implementation, we can measure every fortnight, as the measurement taken few hours once may not be useful for decision making. Even though, we compute the energy consumptions every fortnight, we still continue to monitor our devices and their respective sensor values to identify and assure that the devices are functioning properly. This will use only minimal energy and since we send only a few bytes of data, this method has a very low resource overhead.

4. USE of IoT and Machine Learning

This work aims to use the benefits of two research areas namely, Machine Learning and Internet of Things. We use the concepts of IoT to collect the data and cloud for storage. The data thus collected is used as inputs to our machine learning algorithm. The machine learning algorithm can crunch the data and provide the necessary optimizations in the settings to manage and minimize energy consumption in data centers. The novelty lies with the concept of using our algorithm on the data which is acquired from the cloud and take necessary actions in real-time frame. The inputs to the algorithm are the real-time data from the sensors measuring temperature, humidity, power consumption, etc. This enables the proposed system, to provide a very efficient energy management.

5. PROPOSED SOLUTION

Our proposed system provides an audio- visual indication to the administrator when there is a sudden rise or unnecessary usage of energy in the data center. Generally, the primary appliances that consume a lot of energy are air-conditioner, cooling system, etc. This system senses the energy consumption at each point (where the appliance is present), and based on the usage and the input energy, it makes a decision whether or not that appliance requires more energy. This decision is based on the sensor values from that point and also from the nearby points.

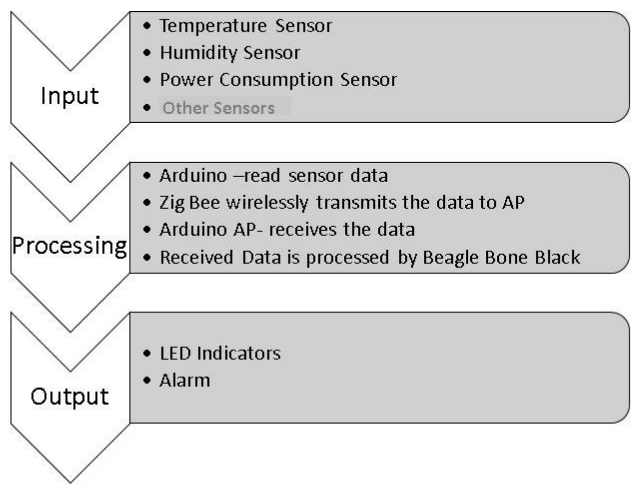

Figure 2. Top-Level Design

The proposed solution is to have an automated Energy Monitoring system. This top level design is shown in Fig.2. This system will have a set of micro-controllers that uses sensors to measure the temperature, humidity, power consumption, etc. These data are sent to a common access-point, another micro-controller, wirelessly [4] using the ZigBee protocol. The data from the access-point is sent to the decision hub, a single-board computer (SBC).

The SBC is configured as a cloud server as well as a decision hub. The Beagle Bone Black (BBB) acts as a local server and uploads all sensor data to a cloud and also a database. The cloud database is created using Owncloud, a Linux package. This database is populated with the sensor data using a python script. The functions of the SBC as a decision hub are as follows:

- Control the parameters
- Monitor the parameters at regular intervals
- Alert the administrator at critical scenarios
- Log the data into a cloud-based database
- Use the inputs for Machine Learning

In a large-scale data center where there is a vast area to be covered with a lot of servers and appliances to be sensed we propose the following hierarchy as shown in figure 3. The description is that, the number of end–devices is proportional to the number of columns of servers in the data centers. The number of access-points is less than the number of end-devices, this also depends on the area of the data center. The bigger the area, we may use more acces points to ensure realiable data transfers.There will be a single SBC, BeagleBone Black in this case to gather the incoming data and upload it into the cloud.

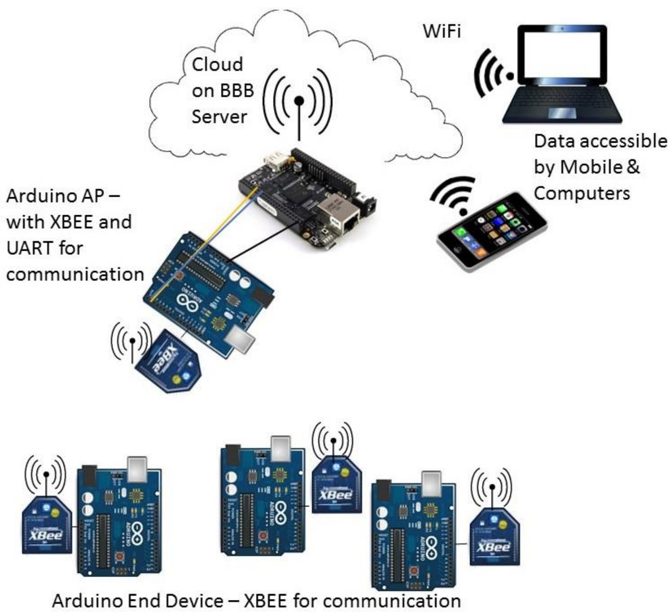

Figure 3.The hierarchy of devices in large-scale deployment

6. IMPLEMENTATION

6.1 Hardware Implementation

The proposed solution has two devices namely the base system and the end device. The proposed model is a warning system that uses sensory inputs from various points across the data center and sounds an alarm. The sensors are controlled by a micro-controller (Arduino) which is in turn controlled by a Beagle Bone Black. The Beagle Bone controls all the micro-controllers and is in constant communication with the administrator of the data center and, this forms the base system as shown in figure 4. This is to give some control to the administrator and also report any damages or failures. The communication between the Beagle Bone and Arduino micro controllers is wireless and is encrypted for security purposes. The AES or XXTEA may be the cryptographic support in the setup.

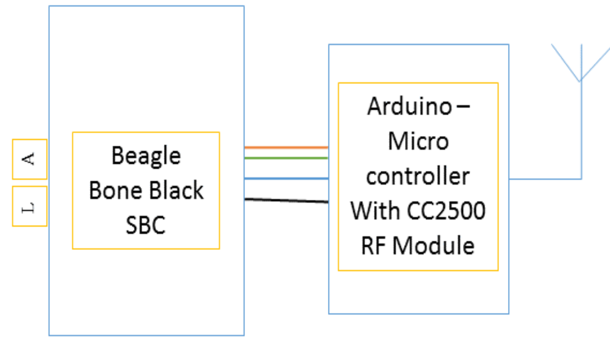

Figure 4. Design of the Base System

The **Base-System:** this is the decision hub along with the Access Point. This device has LED to provide visual indications and an alarm to indicate and alert the operator when the system reaches a critical level. This is also the system which acts as a local server and hosts all the sensor data in a cloud. The communication is wireless between the end-devices and the access-point. The access point is an Arduino board interfaces with a ZigBee wireless module. The design also features a complementing set of sustainable power sources like Battery and Solar Cells that can be used in case of a power failure. The administrator is warned of abnormal energy consumptions by audio (an alarm (A)) and visually by a set of LEDs (L) as shown in figure.4. The communication between the Arduino – Access Point and the Beagle Bone Black is done using the hardware UART. The Access Point receives data from the various end-devices, decrypts it and then sends to Beagle Bone Black.

The other device is the end-device. This is also an Arduino board which is interfaced with a Zigbee module for wireless communication to the access-point, in addition to that this also has sensors(S1,S2,S3) attached to it as shown in figure 5. We use Texas Instrumnets' CC2500 Wireless Modules, for communication between the End Device and Access Point. The CC2500, operates in SPI mode which saves a lot of power and extends the battery life of the device.

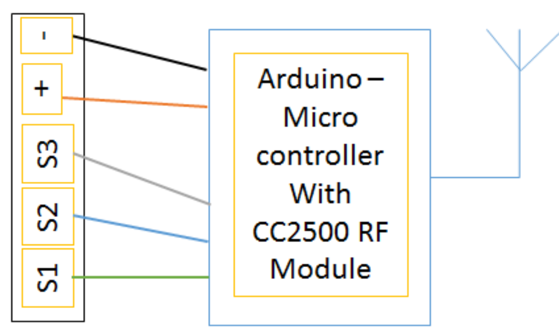

Figure 5. The design of the end device

The sensor values are read and is clubbed to form a packet with the device ID . This packet is sent to the access point. These sensors are controlled by the micro-controller.

The functionality of these two devices can be summarized as follows: the Arduino board transmits the sensor inputs and also about the state/working condition of the sensors. Beagle Bone Black acts like a hub interacting with the administrator and the micro-controller. Thus, BBB receives data/instructions from the administrator and sends usage reports, sensor readings and also triggers alerts in case of any mishaps or failure in the vicinity of the device setup by uploading the data to a secure cloud.

6.2 Data Organization and Data Flow

The data which is collected by the end devices is uploaded to the cloud server that is created using beagle bone black. This data is uploaded every 30 minutes. The data is processed using a python script which appends the received data to a local data base and the data is also appended to a comma separated value (.csv) file which is maintained at the cloud server. Thus, we get the proposed outputs - the graph - generated using the google graphs API and a .csv file which can be accessed from the cloud server by the administrator or the authorized personnel in the network. The .csv file forms the basis for the learning and prediction. The data extracted from the .csv file contains the details of the sensor reading from all the sensors along with a timestamp, each in separate columns. The columns contain the values measured by the temperature, humidity, power consumption sensors.

We are presently conducting system tests in the data center at Amrita University, Coimbatore to analyze the implementation costs and performance of our proposed system. The threshold value can be computed from this pilot work and then this can be scaled up to large data centers.

6.3 Software Implementation

This section gives the details of the software implementation. The algorithm or pseudo steps are shown below.

6.3.1. End Device – Arduino

a) Read sensor values from sensors.

b) Store the read ADC values into a variable.

c) Add precision with desired representation for values.

d) Push these into an array of characters

e) Array [0] = '*' and array [n-1] = '#' where n is the array size.

f) Array [1:2] => the device ID assigned to the end device.

g) Encrypt the data using the XXTEA algorithm.

h) Send the data Via SPI using ZigBee.

6.3.2. Access Point – Arduino

a) Receive the data via SPI from different End Devices.

b) Store the data in a buffer

c) Serially send the buffer contents to Beagle Bone Black

d) Sleep for 500 ms.

6.3.3. Data Processing at Beagle Bone Black

a) Open a Serial port and always listen.

b) Read the incoming serial data

c) Decrypt the data and append it to the database.

d) Also write the data into a csv file

e) Sleep for 10ms

6.4. HARDWARE HEALTH MONITORING

The scenarios which indicate that the system is not functioning and when the administrator is needed to take the decision are as follows:

1. When a sensor fails - this is detected when the packet has a null value for a consistent time period, then the End Device alerts the SBC which in turn alerts the administrator using an alarm
2. When an End Device fails – this is detected at the access point, when data is not received over a long time (time in terms of hours), then the access point alerts the SBC which in turn alerts the administrator.

6.5. APPLICATION OF MACHINE LEARNING ALGORITHM

For further analysis we plan to use the csv file generated as discussed in section 6.2. This forms a training set for our machine learning algorithm. But, in real-time implementation of this work, the administrator has an option to set the frequency at which the energy consumption is to be measured.

Once the data set is ready, our proposal is to implement and try with machine learning algorithms. The algorithm aims to learn and find a pattern for the energy consumption in the data center. We get a distribution curve, from the gathered data; this curve will help us to find out the servers that are operated at max-load or are shut-down or throttled. We are still working on how to use the data available from the distribution function mentioned above and to further optimize the power consumption. Furthermore, we can use the learned function to find the pattern of energy consumption, analyze the energy consumption with the past and present consumptions and make suitable decisions or prediction.

7. RESULTS

This section discusses the results obtained from the implementation methods discussed in section 6. The first task, was to read the sensor data and make a simple comma-separated packet. In 6, the sensor read data is shown. This packet is sent for encryption and then will be sent wirelessly as discussed in section 6.3.1.

The encryption module, used in our proposed module is the XXTEA encryption algorithm. The reason for choosing XXTEA is that it uses less resources for the encryption process which makes it suitable for embedded devices. For ensuring that there is no ambiguity with the data that is received we send them as packets. Each packet is made with certain conventions like:

- Each packet begins with a ‘*’.
- This is followed by a 2 digit number – which is the device ID which is user assigned
- Following the device id, we have the sensor values each separated by some special characters. These characters are useful to extract the data in the later stages

The need for security is to prevent data loss, bit errors in data during the communication and also to prevent intruder attacks like phishing, etc. The encryption takes place at the Arduino -End Device soon after the packet is formed and just before sending it. The decryption takes place at the Arduino – access point, right after a packet is received. The packet when received by Beagle Bone Black, is split into variables as device Id, timestamp, and sensor data which are then uploaded to a cloud created using Owncloud, as discussed in Section 6.2. The data are also uploaded into a local database which is used with a web page to give a graphical plot of the sensor values for each end-device, and is used for maintenance and monitoring.

```
*01 , 0304 , 0227 , 0196 , 0168#
*01 , 0122 , 0107 , 0115 , 0120#
*01 , 0115 , 0102 , 0113 , 0115#
*01 , 0113 , 0100 , 0110 , 0113#
*01 , 0130 , 0120 , 0132 , 0128#
*01 , 0113 , 0099 , 0110 , 0113#
*01 , 0116 , 0102 , 0113 , 0115#
*01 , 0111 , 0098 , 0109 , 0111#
*01 , 0112 , 0098 , 0109 , 0111#
*01 , 0111 , 0098 , 0109 , 0111#
*01 , 0109 , 0096 , 0108 , 0110#
*01 , 0110 , 0097 , 0108 , 0110#
*01 , 0111 , 0097 , 0108 , 0111#
*01 , 0111 , 0097 , 0109 , 0111#
*01 , 0110 , 0097 , 0108 , 0110#
*01 , 0107 , 0093 , 0104 , 0106#
*01 , 0110 , 0097 , 0108 , 0110#
*01 , 0108 , 0094 , 0105 , 0108#
*01 , 0109 , 0096 , 0107 , 0109#
*01 , 0109 , 0096 , 0107 , 0109#
```

Figure 6. Read data made as packets

Table 2 shows the data at the end-device after encryption. Table 3 shows the data at the access point after decryption.

Table 2. Data in the End Device

Plain Text – from End Device	Encrypted Text – Sent from End Device
*01@310$236%204^179#	Ïk(ƒ{"r+Ý Ïk(ƒ{"r+
*01@136$121%127^132#	Ž‡R¯WûñˆXšU Ž‡R¯WûñˆXš
*01@127$114%122^125#	¿íÉ1ð[˜‹r‹å ¿íÉ1ð[˜‹r‹
*01@126$113%121^124#	²viR¢ˆ£l* N ²viR¢ˆ£l*
*01@126$113%121^124#	8ó…bjkêL€K� 8ó…bjkêL€K

Table 3. Data in the Access Point

Received Text at Access Point	Decrypted Text – at Access Point
Ïk(ƒ{"r+Ý Ïk(ƒ{"r+	*01@310$236%204^179#
Ž‡R¯WûñˆXšU Ž‡R¯WûñˆXš	*01@136$121%127^132#
¿íÉ1ð[˜‹r‹å ¿íÉ1ð[˜‹r‹	*01@127$114%122^125#
²viR¢ˆ£l* N ²viR¢ˆ£l*	*01@126$113%121^124#
8ó…bjkêL€K� 8ó…bjkêL€K	*01@126$113%121^124#

The data stored as a .csv file is shown in figure 7. This file is downloaded from the cloud server, which we created using BBB. This file can be downloaded by the administrator and personnel with the correct user name & password.

	A	B	C	D
1	time_stamp	Temperature	Humidity	Current
2	15-01-2015 13:01	18.8717896	41.79128373	241.9353437
3	16-01-2015 13:01	19.69651585	41.2914899	238.7431818
4	17-01-2015 13:01	19.0062998	41.1550225	238.9108178
5	18-01-2015 13:01	18.41751864	41.15060604	239.3359684
6	19-01-2015 13:01	18.67465577	40.61333002	238.9756181
7	20-01-2015 13:01	19.02040625	40.50230712	239.6869579
8	21-01-2015 13:01	18.98470534	40.93267361	238.4867019
9	22-01-2015 13:01	18.99363854	40.99504336	238.1661016
10	23-01-2015 13:01	18.95953329	40.77795314	238.5597949
11	24-01-2015 13:01	18.64115868	41.23134922	240.3600644
12	25-01-2015 13:01	19.32267255	40.42835616	241.604791
13	26-01-2015 13:01	19.3480558	40.56397824	239.5538768
14	27-01-2015 13:01	18.8863302	40.7438205	240.8735028
15	28-01-2015 13:01	19.77388424	41.28753091	239.584819
16	29-01-2015 13:01	19.90835456	40.77785043	240.7926488
17	30-01-2015 13:01	18.27005139	41.79245728	240.7296895
18	31-01-2015 13:01	18.94560186	40.88807274	241.3729795
19	01-02-2015 13:01	19.95808706	41.96150349	239.173598
20	02-02-2015 13:01	19.51452019	40.45662065	242.2580277
21	03-02-2015 13:01	19.45350944	41.60946641	242.827221
22	04-02-2015 13:01	19.74062943	41.04452311	241.7174744
23	05-02-2015 13:01	19.11399226	40.27400162	241.354964

amrita_dc

Figure 7. The .csv file, downloaded from the cloud

Currently this work is in progress in the data center at Amrita University, Coimbatore. The end device is kept on the server rack as shown in figure 8. The base system (BBB + Arduino AP), is kept close to the controlling system, shown in figure 9. Figure 10 shows the data center.

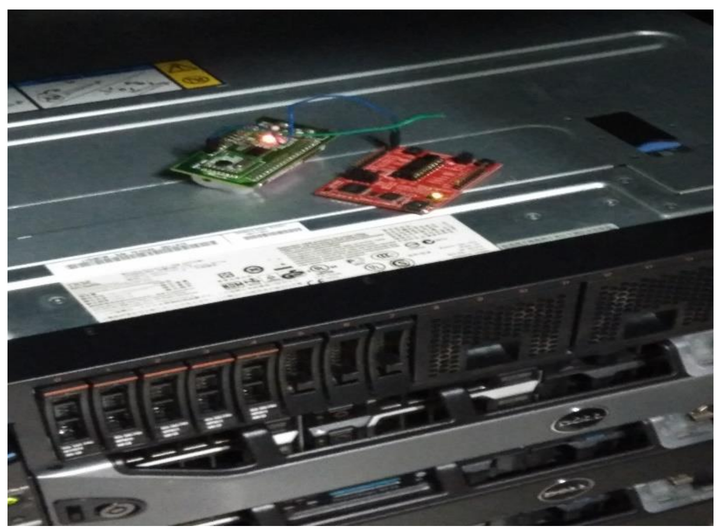

Figure 8. End Device on a Server Rack

Figure 9. Access Point with SBC along with Client System

Figure 10 the data center at Amrita University, Coimbatore

8. CONCLUSION AND FUTURE WORK

In this paper, we introduce the prototype of a smart energy management system for large scale data centers that uses IoT and machine learning concepts. Our implementation in the data center at a university environment shows that the overall system works perfectly & consumes minimal energy. The results reported in section (implementation), forms the basic test bench with the data being accumulated in a cloud. In future, this will be used as the dataset for the proposed machine learning algorithm to predict future energy consumptions and also the estimate the amount of energy required to cater the demand. The proposed design can be scaled up to optimize energy consumptions in various other domains like: hospital, home and office –based energy management systems by modifying the sensors used, and a minor modification to the proposed algorithm.

ACKNOWLEDGMENTS

We would like to extend our heartfelt gratitude to the Mobile and Wireless Networks Lab, Dept. of CSE, Amrita School of Engineering, Coimbatore and the ICTS Department, Amrita Vishwa Vidyapeetham, Coimbatore for the support extended in carrying out this work.

9. REFERENCES

[1] Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems 29, no. 7 (2013): 1645-1660.

[2] Chase, Jeffrey S., Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, and Ronald P. Doyle. "Managing energy and server resources in hosting centers." In ACM SIGOPS Operating Systems Review, vol. 35, no. 5, pp. 103-116. ACM, 2001.

[3] Younis, Mohamed, Moustafa Youssef, and Khaled Arisha. "Energy-aware routing in cluster-based sensor networks." In Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on, pp. 129-136. IEEE, 2002.

[4] Baronti, Paolo, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu. "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards." Computer communications 30, no. 7 (2007): 1655-1695.

[5] Kortuem, Gerd, Fahim Kawsar, Daniel Fitton, and Vasughi Sundramoorthy. "Smart objects as building blocks for the internet of things." Internet Computing, IEEE 14, no. 1 (2010): 44-51.

[6] Patrick Thibodeau, Data Centers are the new polluters,http://www.computerworld.com/article/2598562/data-center/data-centers-are-the-new-polluters.html, accessed: 5.4.2015

[7] Jason Verge, NRDC: Multi-Tenant Data Centers need to play a bigger energy efficiency role, http://www.datacenterknowledge.com/archives/2014/08/26/data-center-energy-efficiency-role/, accessed: 5.4.2015

[8] America's Data Centers Consuming and Wasting Growing amounts of energy,http://www.nrdc.org/energy/data-center-efficiency-assessment.asp, accessed: 6.4.2015

[9] Google's Green Data Centers: Network POP Case Study, Google Inc., Available: http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en/us/corporate/datacenter/dc-best-practices-google.pdf ,2011, accessed: 24.4.2015.

[10] Reducing Data Center Energy Consumption, Intel, Available: http://www.cs.berkeley.edu/~istoica/classes/cs294/09/CERN_Whitepaper_r04.pdf, accessed: 30.4.2015.-

[11] Natural Resources Defense Council, http://www.nrdc.org/about/, [accessed] 07.04.2015.

[12] Rahman, A.; Xue Liu; Fanxin Kong, "A Survey on Geographic Load Balancing Based Data Center Power Management in the Smart Grid Environment," Communications Surveys & Tutorials, IEEE , vol.16, no.1, pp.214,233, First Quarter 2014

[13] Junwei Cao; Keqin Li; Stojmenovic, I., "Optimal Power Allocation and Load Distribution for Multiple Heterogeneous Multicore Server Processors across Clouds and Data Centers," Computers, IEEE Transactions on, vol.63, no.1, pp.45,58, Jan. 2014

[14] Liang Yu; Tao Jiang; Yang Cao, "Energy Cost Minimization for Distributed Internet Data Centers in Smart Microgrids Considering Power Outages," Parallel and Distributed Systems, IEEE Transactions on , vol.26, no.1, pp.120,130, Jan. 2015