CLOUD-BASED ELECTRIC DATA MANAGEMENT FOR SMART GRIDS

Arslan Ahmed[1], Waleed Ejaz[2], Mohamed Ibnkahla[3]
Deptartment of Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
[1]arslan.m.ahmed@gmail.com, [2] waleed.ejaz@yahoo.com, [3]ibnkahla@sce.carleton.ca

***Abstract*—With the advent of smart appliances and home energy management systems, the amount of data generated about consumer energy consumption has rapidly increased [1]. The existing grid system cannot efficiently meet the growing demand for power. This calls for a need to communicate all this data in real time between the grid, electricity generators and electricity consumers. All this should be done through a reliable, efficient, secured and cost-effective solution with the goal of managing the load at the grid. This huge amount of data cannot be stored and processed on local device because of their limited computation resources. Moreover, the amount of this data is increasing exponentially. Here comes the concept of Internet of Things (IoT) that helps us harnessing the data for our best interests using cloud-based services and providing intelligent control over our appliances. Therefore, we need to couple cloud computing solutions to our grid to manage data from millions of smart meters in reliable and scalable way. This paper provides a survey on different cloud computing technologies and applications for smart grids, primarily in the area of communications and data management. We also discuss different challenges existing in today' grid system that can be overcome using cloud technologies, and top four companies investing heavily into this paradigm.**

Keywords—Smart Grid, Cloud Computing, Internet of Things, Demand Response

I. INTRODUCTION

A smart grid is an electrical network where consumers and generators can communicate real time data with each other, and simultaneously allow the two-way power flow between them in order to deliver electricity more efficiently and reliably. Cloud computing is an emerging technology advocated for enabling reliable and on-demand access to different computing sources that can be quickly provisioned and released in a cost-effective way to the service providers. Using cloud services, utility company as well as consumers can gain insight to the electricity usage data anytime, from anywhere in the world. It will help both the utility and the consumers who can analyze it using algorithms of their choice. The exponential growth of data from smart meters and sensors calls for the smart grid to take advantage of the feature of cloud storage and analytics [2].

In this paper, we provide a survey of work done in order to integrate cloud computing technologies into smart grids, focusing primarily on communication and data management aspect. Some discussion is also made on top companies providing cloud computing services. Some future research directions are also explained in terms of data management in smart grid.

In short, our main objective in this paper is to present the following:

1. An overview of Smart Grids and how it is changing the traditional electric grid system.
2. A clear concept of applications of cloud computing in smart grid.
3. Brief discussion on four big giants in cloud computing i.e. Amazon, Microsoft, IBM and Google.
4. Course Project Report SYSC5801. Results are discussed in detail along with future recommendations
5. Discussion on using Intel Edison as a TA and replacing Byzantium with HSMM-Pi

The rest of the paper is organized as follows: We give a comprehensive overview of smart grid in Section 2. In Section 3, we describe in detail different types cloud computing models and explain their utility in different scenarios. In Section 4, we provide a brief survey on the companies which provide cloud computing services, focusing mainly on communication and information management systems in Smart Grids. Section 5 has our Course Project Report for SYSC 5801. Future work regarding this project is proposed in Section 6. In section 7, we discuss our work on Intel Edison and HSMM-Pi. Finally, a conclusion is presented in Section 8. Codes for course project are attached as appendix.

II. OVERVIEW OF SMART GRID

Before we start with the Smart Grid paradigm, we need to discuss a bit about the conventional power grid system. In conventional power system, generators generate the electricity and consumers consume it without providing any feedback to the grid. There is no communication of data between consumers and producers [3].

In Smart Grids, information about consumers' electricity consumption behavior is collected automatically with the use of the ICT. This helps increase the efficiency, reliability and performance of the electric grid [4]. The reliability of electric power system is a vital parameter for power delivery and

economic development for every community. Today, operation of traditional power system is limited in interoperability across applications.

Traditional electric grid is not capable of providing high power quality, reliability, availability, and security. But smart grids use large numbers of networked sensors, power electronic devices, distributed electricity generators, and communications appliances for bi-directional communication between consumers and generator [5]. In this way, electric grid becomes smarter with integration of large quantity of real-time information and data processing [6].

Smart energy management systems are key enablers of the envisioned efficiencies both on the demand and supply sides of the smart energy grids [7]. On the demand side, they aim at supporting end-users in optimizing their individual energy consumption, e.g., through the deployment of smart meters providing real-time usage and cost of the energy and the use of demand-response appliances that can be switched on/off at a given time depending on the user preferences, energy cost and carbon footprint [8]. On the supply side, the smart grid management systems aim at optimizing the network load, quality and reliability of the energy provision, e.g. thorough active monitoring and prediction of the energy usage patterns, and pro-active control of the reliable energy delivery over the network. It is also envisaged that they will be able to influence the demand through the dynamic adjustments of the energy price in order to influence the end-user behavior and energy usage patterns [1].

Fig. 1 illustrates a smart grid architecture with all the generators and consumers integrated to a single grid. Cloud over the grid shows the concept of cloud computing in Smart Grids.

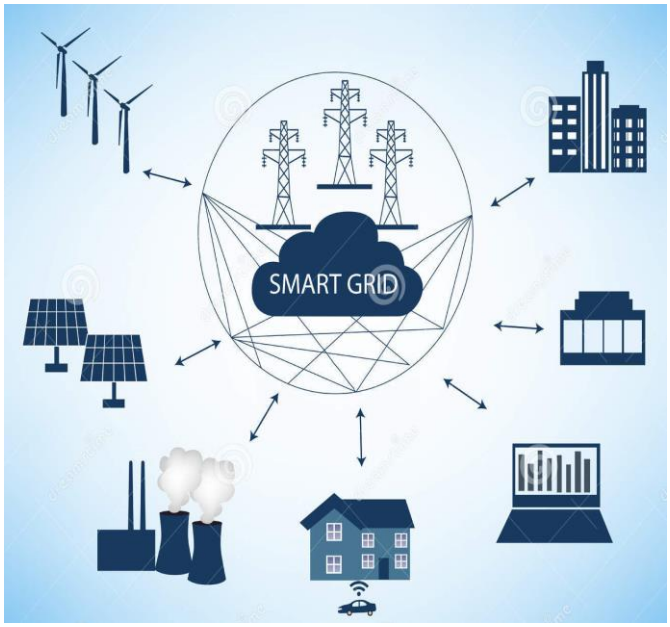


Fig. 1. Smart Grid with cloud computing illustration

III. Overview of Cloud Computing

Cloud computing is an emerging computation model that provides on-demand facilities, and shared resources over the Internet. In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of local hard drive [9].

Cloud computing provides three different types of services- Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

1) Infrastructure as a Service (IaaS), contains the basic building blocks for cloud and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today. [10].

With IaaS, it's like leasing a car. Keeping the car repaired is someone else's problem, you just need to supply it with fuel (setting it up, maintaining software, etc.) and you get to go pretty much wherever you want to. This system offers very high flexibility [11]. IaaS is more popular among users into research and high computing areas due to high flexibility and relativity difficult use as compared to others.

Examples for IaaS include: AWS Elastic Compute Service (EC2), Google Compute Engine (GCE), Azure Virtual Machines

2) Platform as a service (PaaS): Platforms as a service removes the need for organizations to manage the underlying infrastructure like hardware, OS, security features etc. and allow you to focus on the deployment and management of your applications. This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application [10]. PAAS is more popular among developers as they can put all their concentration on developing their apps and leave the rest of management and execution to the service provider.

PaaS is a bit like getting a taxi. You get in and choose your destination and route. Keeping the car running and figuring out the details is up to the driver.

Examples for PaaS include: IBM Bluemix, AWS Elastic Beanstalk, Google App Engine, Azure Cloud Services.

3) Software as a Service (SaaS): Software as a Service provides you with a completed product that is run and managed by the service provider. Usually, people referring to SaaS are referring to end-user applications. With a SaaS you don't have to think about how the service is maintained or how the underlying infrastructure is managed, you only need to think about how you will use that particular piece of software or service. A common example of a SaaS application is web-based email where you can send and receive email without having to manage feature additions to the email product or maintaining the servers and operating systems that the email

program is running on. [24]. SAAS is more popular among with consumers, who bother about using the application.

SaaS resembles public transport system. It's cheap, you just get to use it to reach your destination, driver and the transport authority is to worry about all the route planning, fare, stops etc. This comes at the price of not always getting as close as you want (less customizability).

Examples for SaaS include: Google Apps (Gmail, calendar, spreadsheets), MS Office 365, Dropbox

Fig. 2 compares IaaS, PaaS and SaaS with respect to how much efforts user has to contribute with respect to the vendor.

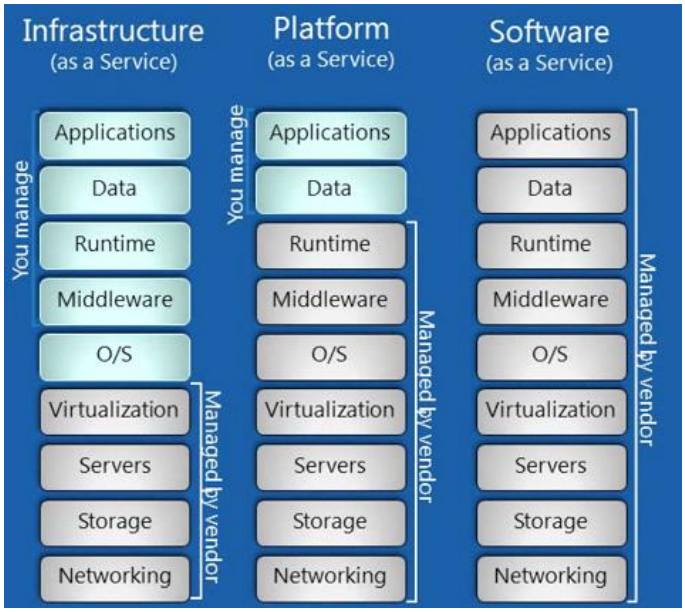


Fig. 2. Comparison of IaaS, PaaS and SaaS [21]

On the other hand, cloud can also be categorized, depending on the deployment models, as—private, public, and hybrid [11].

1) Public cloud. Public clouds are owned and operated by a third-party cloud service provider, which deliver their computing resources like servers and storage over the Internet. Microsoft Azure is an example of a public cloud. With a public cloud, all hardware, software, and other supporting infrastructure is owned and managed by the cloud provider. You access these services and manage your account using a web browser. The purpose of this type of cloud application is to serve its own business applications [12].

2) Private cloud: A private cloud refers to cloud computing resources used exclusively by a single business or organization. A private cloud can be physically located on the company's on-site datacenter. Some companies also pay third-party service providers to host their private cloud. A private cloud is one in which the services and infrastructure are maintained on a private network. Hybrid cloud. Hybrid cloud is the extension of cloud computing with private, public, and community cloud computing techniques [26]. The private, public, and community clouds are integrated together to perform several tasks which can handle the requirements of private, public, and community organizations.

3) Hybrid Cloud: Hybrid clouds combine public and private clouds, bound together by technology that allows data and applications to be shared between them [16]. By allowing data and applications to move between private and public clouds, hybrid cloud gives businesses greater flexibility and more deployment options.

In Fig. 3, we see the relation between three cloud computing models i.e. public, private and hybrid cloud.

Fig. 3. hree cloud computing models i.e. public, private and hybrid cloud [23]

Cloud computing is such a useful technology for smart grid information management due to the following reasons [13]:

1) The requirements of information processing in smart grid fit well with the computing and storage mechanisms available for cloud applications
2) In a smart grid, information sharing is one of the most important issues. Information from different components, and the supply and demand state conditions can be shared with the help of cloud computing [14].
3) Shared information is accessible to the micro-grids, end-users, and utilities, though they function in the islanded mode.
4) Management of massive data is complex, costly, and may be beyond the capacity of existing data management systems in the smart grid [15].

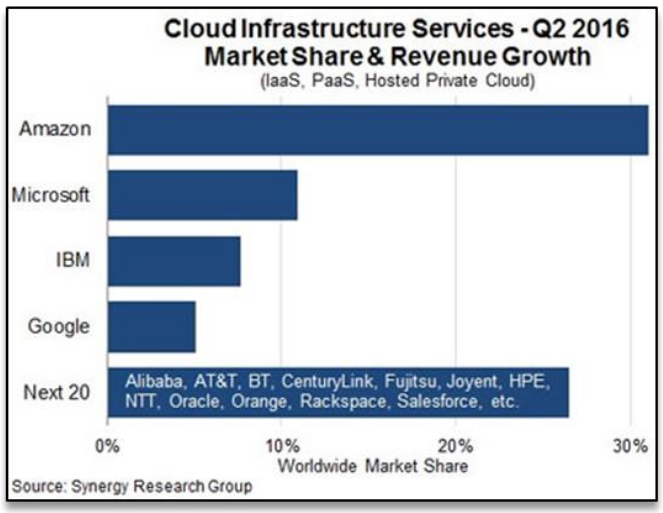
IV. Big Giants in Cloud Computing

Top four companies in Cloud Computing are:

1) Amazon Web Services
2) Microsoft Azure
3) IBM Cloud Solutions
4) Google Cloud

All these service providers differ in their computational resource allowance, storage and networking facilities. The basic difference lies in their pricing model and the ease of use, otherwise all these companies provide more or less same services. Synergy provides quarterly market tracking and segmentation data on IT and Cloud related markets, including

vendor revenues by segment and by region. As per its survey of Q2 2016, cloud computing market share is lead by Amazon Web Services. It has market share of more than 30%. It is followed by Microsoft Azure which has market share of around 10%. Next comes IBM which is relatively new in cloud computing market. They have recently introduced their Bluemix cloud services that we are using in our project. When it comes to IaaS and PaaS, IBM beats google with market share of around 7%, while Google having market share of less than 5% as per Synergy Group. However, google is leading in SaaS because of its quality services like Google Docs, Gmail. For our project, we choose IBM Bluemix because of its ease of use, free trial for first month and that too without entering credit card number unlike Amazon. Moreover, they provide integrated solutions for data management, storage without SQL, analytics and other services that are very easy to use as beginners. Here in figure below, we can see the statistics for Q2 2016 by Synergy group.



Node red at IBM Bluemix

V. Course Project Report

A. Abstract

Here in this project, we propose an automated and intelligent bidirectional information sharing system between home and cloud that also provides graphical view of all the energy consumption data on the web dashboard in user friendly way for utility company.

B. Brief Background of Project

This course project is an extension of Smart Grids final year project in which an IP based local Wireless Mesh Network (WMN) has been created between the Customer Agents (CAs) and a Transformer Agent (TA). Here CA represents homes in a neighborhood and TA represents nearby transformer for all those homes in that neighborhood.

This project is being done with industrial collaboration of Hydro Ottawa (electric utility company for the city of Ottawa). Different communication standards have been developed to help standardize the smart grids industry and accelerate its development. IEEE 2030.5 (Smart Energy Profile) is one of those standards that has been used in this project as per the requirement of Hydro Ottawa.

TA is responsible for getting real time communication with all CAs. It receives electric data from CA as xml files and sends them recommendations regarding usage after some processing over the data. Hardware used as TA and CA is Raspberry Pi and OSLR routing protocol has been used to set-up WMN using Raspberry Pi's internal Wi-Fi.

C. Overiview of the Project

As discussed previously in the survey, the data coming from all CAs will be too large to store and process at the TA. So either we replace the Raspberry Pi as TA with some very expensive hardware capable to perform high computation, or we can send all this data to the cloud for storage and processing. Later option was chosen for this project.

The CA regularly sends information about its household usage to the TA, from where it is sent to the cloud for processing and storage. LTE is used here for this purpose because of the following reasons:

a) TA is located at the pole along with transformer so there will be no WiFi internet connectivity owing to the remote location.

b) Internal WiFi of the Raspberry Pi is already being used to make a mesh network with the CAs.

The utility can view all this data at the cloud level. Here, the energy consumption of each neighborhood will be analyzed and processed as per specific algorithms designed by utility to meet the overall grid's dynamic capacity limit. Based on the result of this processing, demand response information specific to each CA will be sent to the corresponding TA, which can send control commands to that CA through the local WMN to manage the load on the grid.

D. Block Diagram of the Project

Complete block diagram for our project is shown in Fig. 4.

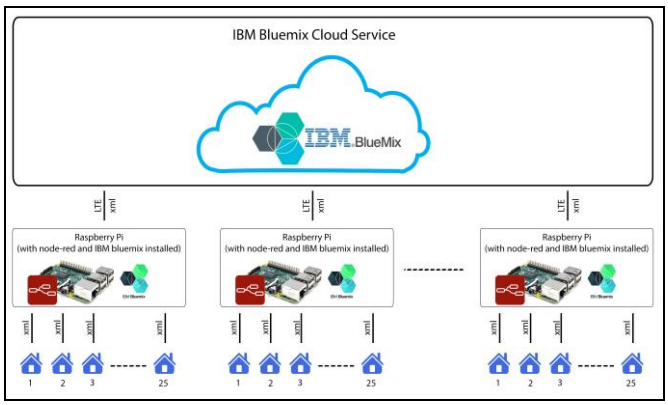


Fig. 4. Block diagram of the project.

Here the lines without arrow indicate bi-directional communication. Since real time feed is not available from

CAs, so we have randomly generated 25 xml files at the TA, assuming that this data is coming from CAs. Each file here represents a home in a neighborhood that means we are considering a sample of 25 homes and 1 neighborhood in this project as we are using one Raspberry Pi. Each Raspberry Pi represents a neighborhood TA.

Following are the parameters of xml files that we have used:

Current load of the CA (or the house) in Ampere

Negotiate parameter (boolean value), indicating whether the house has agreed to negotiate with the utility or not.

House ID. This is the most important paramenter as it helps to identify each CA.

If the negotiation parameter is set to Yes, it means the house is willing to receive recommendations from the utility. This is a win-win situation for both as it helps the utility to optimize the load at the grid and also helps the household to decrease its un-necessary electricity consumption which in turn decreases the bill.

E. Hardware/Software tools used in the Project

Raspberry Pi 3 with 32GB SD Card was the core hardware used in this project. It has the following specifications*:*

a) SoC: Broadcom BCM2837
b) CPU: 1.2 GHZ quad-core ARM Cortex A53
c) GPU: Broadcom VideoCore IV @ 400 MHz.
d) Memory: 1 GB LPDDR2-900 SDRAM.
e) USB ports: 4.
f) *Network: 10/100 MBPS Ethernet, 802.11n Wireless LAN, Bluetooth 4.0*

LTE Sierra Wireless Aircard A330U with Rogers SIM and monthly data plan

IBM Bluemix cloud services*:* I am storing all our data and analyzing it on cloud services provided by IBM. So I installed IBM Bluemix on Raspberry Pi.

Node-red: It is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together different flows. Raspberry Pi and IBM Bluemix, both provid support for Node-red so it was easy to use in our project.

VNC(Virtual Network Computing) Viewer: To make a remote connection between our laptop and Raspberry to display Raspberry Pi's screen on our laptop.

F. Inside Raspberry Pi Block

Node-red and VNC are already installed in Raspberry Pi but we have to install IBM Bluemix to send the data from Pi to the cloud. Each Pi is a TA which represents a neighborhood, so it is given a unique Device ID which in our case is "TA-1". This is not only to identify the TA on the cloud, but also it is an added layer of security or privacy that prevents the release of its data to other TAs of different neighborhoods.

We use Node-red of R Pi to write our program as it is very powerful and easy to use. As discussed previously, it opens up in browser window where we can create and edit flows. We have divided it into two flows: Raspberry Pi to Cloud, and Cloud to Raspberry Pi.

a) Raspberry Pi to Cloud

In this flow, we take data from "input files" folder of local storage of Raspberry Pi and send it to IBM Bluemix. There are 25 different xml files stored on Raspberry Pi. Data is taken after every 10 seconds. This interval can be changed using timestamp node written in the start of the flow. After taking the data, we parse it as xml and then convert into JSON format. This is because JSON format is the only one accepted by IBM Bluemix. Then each file is given a unique key before sending to the cloud. This is to maintain the privacy of their data and avoid the mixing of data due any bug in between the Raspberry Pi and cloud. Graphical view of node-red flow of Cloud to Raspberry Pi block is shown in Fig. 5, and its code is attached in Appendix 1.

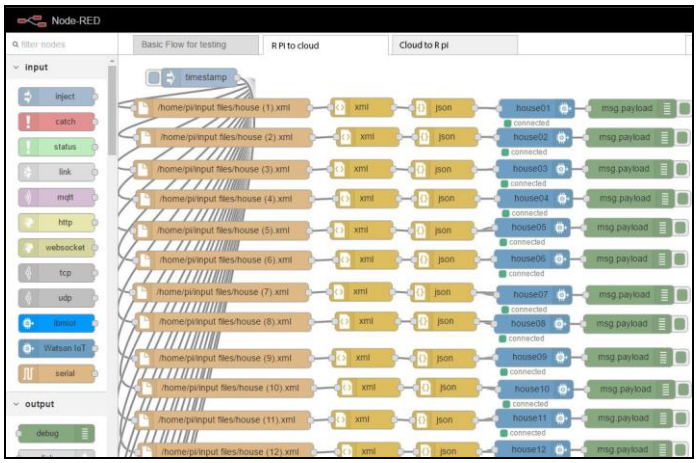


Fig. 5. Raspberry Pi to Cloud Node-red

After sending the file to cloud, we can see the contents of sent file along with the timestamp in the debug panel of node-red on the right as shown in the Fig. 6.

Fig. 6. Data that is being sent from Raspberry Pi to Cloud

b) Cloud to Raspberry Pi

This flow takes the modified data from IBM Bluemix cloud. While receiving the data, first we match the key of the file with the file sent and if it matches, we send it to output files folder of local storage. This received file has some recommendations made by the utility at the cloud. For example, if the current load parameter is greater than the recommended value set by the utility, the received file may recommend the house to lower its electric load by turning off some appliances.

Received file is in the JSON format but the Hydro Ottawa's requirement is to receive xml file at the CA. So in this flow, after receiving the file, it is converted from JSON to xml and then stored in local storage. Graphical view of node-red flow of Cloud to Raspberry Pi block is shown in Fig. 7, and its code is attached in Appendix 1.

Fig. 7. Cloud to Raspberry Pi Node-red.

We can see the received file contents in the debug panel along with the received timestamp on the right side as shown in the Fig. 8.

Fig. 8. Data that is being received by Raspberry Pi from Cloud

G. Inside IBM Bluemix (cloud) Block

This is the main part of our project where all the data is stored and analyzed in real time. It is very powerful cloud service provided by IBM. There are 25 input nodes in this flow, corresponding to 25 houses we are assuming in this project. Each node has a key and receives the data only from the file when its key matches to one of the keys of files sent from Raspberry Pi. This is to make sure that each house's data file does not mix with the other at the cloud end. When sending the data back from cloud to Raspberry Pi, same key is used so that Raspberry Pi can identify which file is coming and can store it accordingly. This is a very good authentication feature provided by IBM in order to secure the data. Graphical view of node-red flow of IBM Bluemix block is shown in Fig. 9, and its code is attached in Appendix 1.

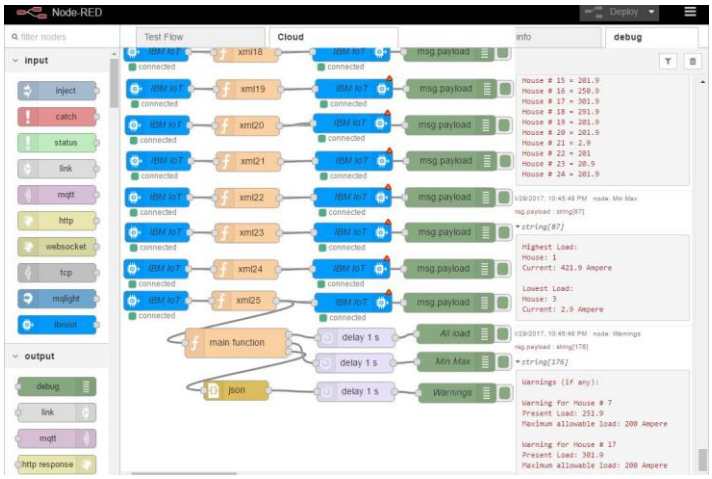


Fig. 9. Node red at IBM Bluemix

Here we perform four different functions with the data:

a) Check for highest and lowest value of load in the neighborhood (shown in Fig. 10).

Fig. 10. Highest and Lowest load being displayed on cloud

b) Display all the load values in the debug panel of Node-red on the right (shown in Fig. 11).

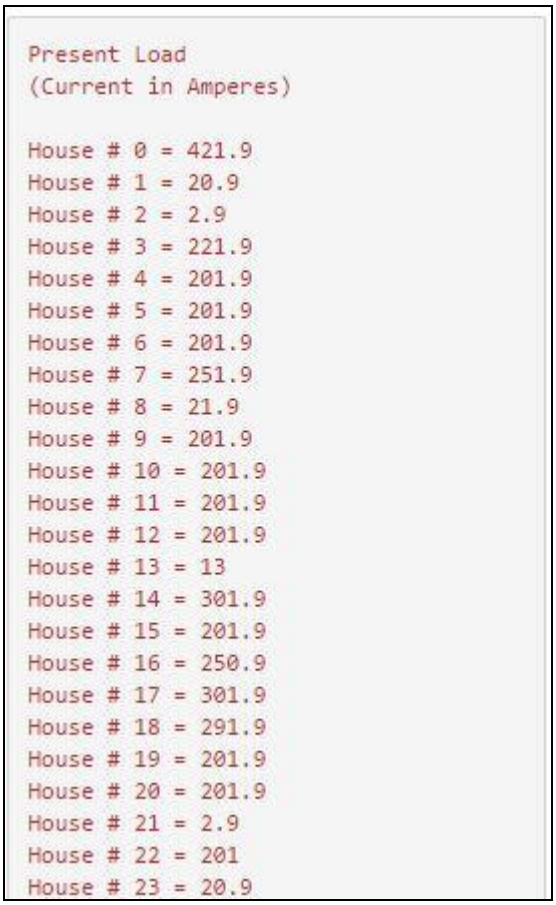


Fig. 11. List of different load values for different houses on cloud

c) Show warnings for the houses who are exceeding the recommended load limit and have not opted for negotition with Hydro Ottawa. Hydro Ottawa can see these warnings and can take necessary actions against those houses. For example, house 4 and 5 are exceeding the limit and not willing to negotiate as shown in the Fig. 12.

Fig. 12. Warnings displayed on cloud.

d) For the houses who are exceeding the recommended load limit but have opted for negotition with the utility, give recommendations to lower their electricity. This is done by modifying their xml file and inserting the recommended load value for the parameter "current_load". For example, compare Fig 8 and Fig9. You will see that the xml file sent from Raspberry Pi to cloud for house#1 has the current_load value of 421.9Amp which exceeds the recommended value of load by Hydro Ottawa i.e. 200Amp. But the received file from cloud to Raspberry Pi for the same house has current_load value of 200Amp which is the recommendation by Hydro Ottawa.

e) Display line chart of current_load values in the IBM Bluemix dashboard. This will make easy for the utility company to compare the trend of the load. This graph shows real time values of load of different houses. We can also go back in time to see previous trends or check any previous value of load for any particular house. Time window can be adjusted as per utility's needs. See figure 5 for reference. Similar to the current_load graph, we can plot real time graphs of other parameters like voltage etc when the project scales up. Overview of IBM Bluemix dashboard is shown in Fig. 13. Line chart graph is shownn in Fig. 14.

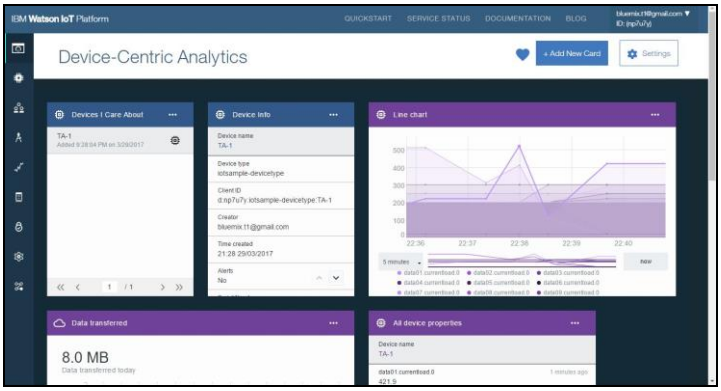


Fig. 13. Dashboard of IBM Bluemix

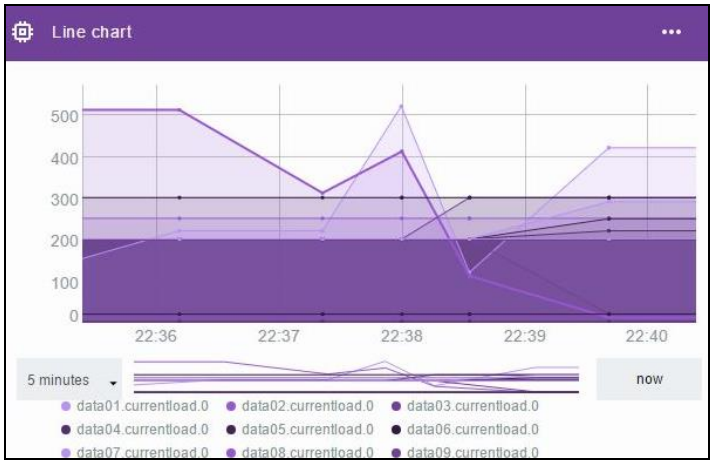


Fig. 14. Line chart comparison of curren_load of different houses on IBM Bluemix

f) IBM Bluemix dashboard also displays a list view of all the data collected from diferent houses as shown in Fig. 15. This makes easy for Hydro Ottawa to search for a particular paramenter of a particular house when this project scales up.

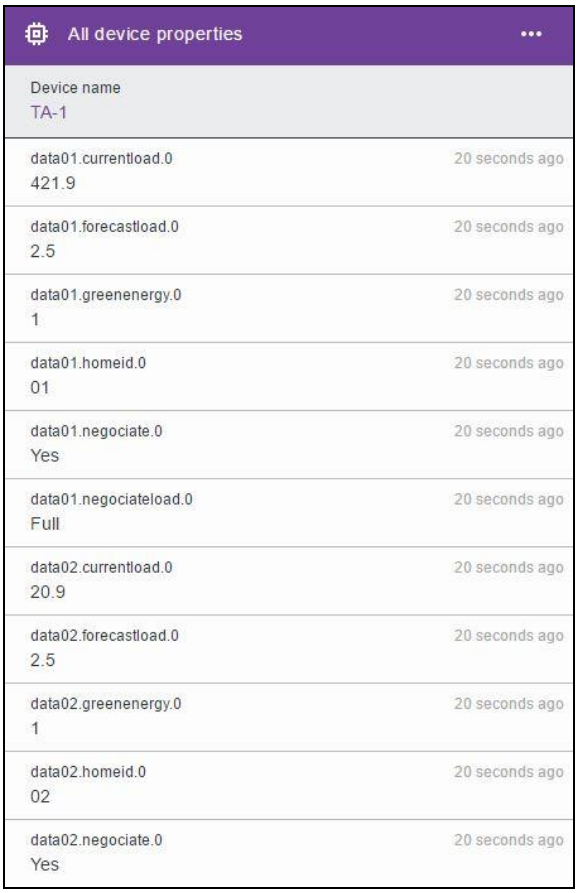


Fig. 15. List view on IBM Bluemix displaying all the parameters for all the houses

H. Chellenges during the Project

We faced three major challenges during this project.

1) Connectivity of LTE module with Raspberry Pi:

Major challenge was to connect Sierra LTE module with Raspberry Pi because most of the resources available on internet suggest Huawei devices. We were given the Sierra Wireless AirCard A330U device which didn't even have complete datasheet, and I did not find anyone in literature review who used it successfully with Raspberry Pi. Basically, it was designed to work with Windows so its drivers were not available for Linux and we had to manually configure the device to make it compatible with Raspberry Pi.

Most popular methods of running 3G/4G device on Raspberry Pi are:

a) Sakis 3g: This is only suitable for 3g devices and has become a little old method now a days but is widely used because of reliability [16].

b) Network Manager for Raspberry Pi: It didn't work for us because as soon as we inserterd the module, network manager went into not responding mode [17].

c) WvDial method: We were successful using this method to connect our LTE module to Raspberry Pi. It needs a few configration files to be added and if properly configured, it works well. These files are shown in Appendix 1.

2) Frequent dysconnectivity of LTE module:

Second challenge was the dysconnectivity of LTE module after few seconds or few minutes. After digging into literature review and consulting from many peers, we came to conclusion that the issue is with power. Raspberry Pi is not able to provide enough power to LTE module due to which the protection circuitry of Raspberry Pi shuts of the power to LTE, which then re-starts itself. Its proposed solution is to use a powered USB Hub for the LTE module. We were using 2.5A power adapter to power Raspberry Pi and LTE module but it didn't work so a wise guess is to use 3A USB powered hub.

3) Asynchronous behavior of Node-red flows:

Another challenge was the asynchronous behavior of Node-red flows which didn't allow the simultaneous injection of data into nodes. Due to few milliseconds delay, it was not possible to combine all 25 xml files, and the display of data was also not synchronous. To cater this problem, We processed each xml file independently using separate nodes throughout the project, and introduced a one second delay while displaying the final output value.

VI. Future Work

A lot of improvement can be done in this project in terms of features and functionality. Few ideas that come into our mind are:

1) Node-red is very powerful and customizable tool. We can add a lot of nodes like graphs, charts into it. It also allows us to develop a GUI to facilitate the operator. Hence, in future, we may use IBM Bluemix only for storage and analytics but for display and comparison, we can use node-red.

2) Adding maps to the dashboard will pin-point the location of different homes. This will not only enable us to classify energy usage based on location, but also give all the details in multi-line map-snippet form instead of tabular form.

3) Many other electric parameters can be included like total energy in kWh, voltage, power factor, etc. This will help to study voltage fluctuations in a particular neighborhood and utility can act accordingly.

4) This project was done using only neighborhood. It can be tried using 10s of neighborhoods that means 10s of Raspberry Pi instead of one. We may increase the size of each neighborhood from 25 to 250 or even more.

5) Scaling the project means playing with more and more data. So, we can apply some big data analytics over it to study the electricity usage behavior of certain neighborhood and provide incentives accordingly, or we even predict the future usage based on the collection of data. All this is supported by IBM cloud services like we can integrate Apache Spark and Cloudant NoSQL DB in our app. By implementing a comprehensive data analytics program, utility companies can also reconcile the demands of greenhouse gas legislation and establishing a meaningful return on investment from smart grid deployments.

6) Currently, consumers do not have real-time access to their own data and are limited to the analysis provided by their utility. Future prospect of this project will be providing the consumers with real time access to their private usage data. Therefore, they can analyze it with personalized algorithms and optimize their usage accordingly.

VII. Intel Edison and HSMM-Pi

The Intel Edison is a computer-on-module offered by Intel as a development system for wearable devices and Internet of Things devices. The system contains a dual-core Intel Quark x86 CPU at 400 MHz communicating via Bluetooth and Wi-Fi [18].

As discussed in the beginning, an IP based Wireless Mesh Network was established to connect the CAs and the TA. Application (Byzantium) used to control this WMN from the TA has some limitations. It works only on Porteus version of Linux, which is only supported by Intel architecture. This means Raspberry Pi cannot be used as a TA because it is based on ARM processor [19]. Therefore, Intel Edison has been considered as an alternate option for TA but since it is relatively very new, so there are not much resources available on internet about its usage, and the intel Edison community is also not so large as compared to Raspberry Pi. We tried to set it up and was able to successfully configure it, and run few basic codes on it using Arduino IDE. We were also able to SSH into it by using the serial communication of Edison and Putty in Windows, and it worked fine on its built-in Operating System known as Yocto Linux.

After being successful with its native OS, we tried to install Porteus version of Linux on it but it turned out that there are few complexities in this process.

1) If Edison needs to boot from SD card, we need to format the SD card with ext4 partition (unlike the commonly used FAT32 or NTFS) system and the tool used for this runs on Linux machine. We still have to try this by borrowing linux machine from some friend.

2) Edison does not have a GUI, so We am not sure how We will be able to configure it after booting with external SD card. For now, we were using SSH but with the local OS. With Porteus, we are not sure how to start SSH because no one has ever used Porteus OS on Intel Edison as per the literature.

3) We may need GUI at-least one time initially to establish a mesh network, but still not sure about it.

4) As per our knowledge, a very few people tried to run Byzantium on Intel Edison but eventually they gave up. So, no one is able to do it successfully till now.

5) We tried to post our query on Intel Edison community, hoping that someone can address us officially but the Intel guys themselves are not aware of whether Porteus runs on it or not.

6) Same query We sent to Byzantium founders but We did not get any reply.

7) On GitHub, we asked from someone who tried using Porteus on Edison, he replied: Locked compare and exchange is used throughout multi-core 686 code for locks. Edison can hang on this instruction, but being single-threaded, does not need LOCK." We still need to figure out what does he exactly mean to say. [22]

In short, replacing TA with Intel Edison still needs to be studied in depth and we need to spend more time on further experiments on it.

HSMM-Pi project is a possible alternative to Byzantium project. It can use built-in Wifi of Raspberry Pi to establish wireless mesh network. There could be any number of mesh nodes in the Ad-Hoc WiFi Network. The route among the nodes is managed entirely with OLSR protocol just like with Byzantium, so we only need to change the application layer in our previous project. As per the documentation [25], HSMM-Pi has two modes: Internal mode and Mesh Gateway mode, which means Raspberry Pi can work as a TA instead of laptop or Intel Edison. Its block diagram is shown in Fig. 16.

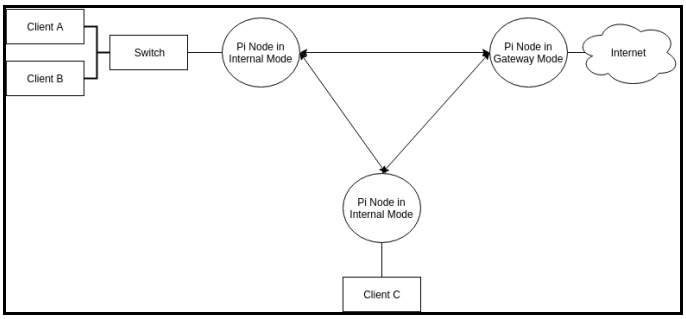


Fig. 16. Block diagram of mesh network setup using HSMM-Pi

We are not yet sure whether IPv6 will work on it or not, but they have an active community on Google Plus [20] where one article describes a possible solution to IPv6 problem in HSMM-Pi. We need to further dig into this option before shifting towards it from Byzantium.

VIII. Conclusion

In this paper, we presented an overview of smart grid and cloud computing technologies.

Next, we explained in detail about the course project for SYSC 5801. This project is designed to be highly scalable and secure using the authentication keys at multiple levels. Total time for between sending all the data from Raspberry Pi to IBM Bluemix cloud, processing it, and the receiving all the recommendations to Raspberry Pi takes less than one second in case of 25 files, and these files are being sent after every 10 seconds. This shows that the latency of the system is very low. We believe that even if the project gets scaled up, the latency would not be highly effected, considering the small size of xml file.

We also provided some future recommendations about our project regarding what type of improvements can be

made in it and how can these can be implemented, specially talking about Big Data Analytics in Smart Grids.

At the end, we talked about Intel Edison and HSMM-Pi with respect to TA. We concluded that Intel Edison is quite complicated and we need much more time to study and experiment on it before reaching our goal. HSMM-Pi may prove better than Byzantium as per the official documentation, but we still need to test it out in real world scenario.

References

[1] Markovic, Dragan S., et al. "Smart power grid and cloud computing." Renewable and Sustainable Energy Reviews 24 (2013): 566-577.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] J. Popeanga, "Cloud computing and smart grids", Database Syst. J., vol. 3, no. 3, pp. 57-66, 2012.K. Elissa, "Title of paper if known," unpublished.

[3] El-Hawary, M. E. (2014). The smart grid—state-of-the-art and future trends. Electric Power Components and Systems, 42(3-4), 239-250.Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[4] Yigit, M., Gungor, V. C., & Baktir, S. (2014). Cloud computing for smart grid applications. *Computer Networks*, *70*, 312-329.

[5] N. Lu, P. Du, P. Paulson, F. Greitzer, X. Guo, M. Hadley, "A multi-layer hierarchical information management system for the smart grid", Proc. IEEE Conf. Power Energy Soc. General Meet., pp. 1-8, 2011

[6] Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2013). A survey on smart grid communication infrastructures: Motivations, requirements and challenges. IEEE communications surveys & tutorials, 15(1), 5-20.

[7] Aghaei, J., & Alizadeh, M. I. (2013). Demand response in smart electricity grids equipped with renewable energy sources: A review. Renewable and Sustainable Energy Reviews, 18, 64-72.

[8] H. Kim, Y. J. Kim, K. Yang, M. Thottan, "Cloud-based demand response for smart grid-architecture and distributed algorithms", Proc. IEEE Int. Conf. Smart Grid Commun., pp. 398-403, 2011.

[9] E. Qaisar, "Introduction to cloud computing for developers: Key concepts the players and their offerings", Proc. IEEE TCF Inform. Technol. Prof. Conf., pp. 1-6, 2012.

[10] https://aws.amazon.com/types-of-cloud-computing/

[11] https://blog.webspecia.com/cloud/iaas-paas-saas-explained-examples-comparison#What_is_PaaS_Platform_as_a_service

[12] Yau, S., & An, H. (2011). Software engineering meets services and cloud computing. *Computer*, *44*(10), 47-53.

[13] Bera, S., Misra, S., & Rodrigues, J. J. (2015). Cloud computing applications for smart grid: A survey. IEEE Transactions on Parallel and Distributed Systems, 26(5), 1477-1494.

[14] Hindia, M. N., Reza, A. W., Dakkak, O., Awang Nor, S., & Noordin, K. A. (2014). Cloud computing applications and platforms: a survey.

[15] Chaichi, N., Lavoie, J., Zarrin, S., Khalifa, R., & Sie, F. (2015, August). A comprehensive assessment of cloud computing for smart grid applications: A multi-perspectives framework. In Management of Engineering and Technology (PICMET), 2015 Portland International Conference on (pp. 2541-2547). IEEE.

[16] Lupu, C., Văduva, A. J., & Rughiniș, R. (2016, September). Invictus: Open-source solution for an intelligent vehicle system. In RoEduNet Conference: Networking in Education and Research, 2016 15th (pp. 1-5). IEEE.

[17] Lu, D., Li, Z., Huang, D., Lu, X., Deng, Y., Chowdhary, A., & Li, B. (2016, July). VC-bots: a vehicular cloud computing testbed with mobile robots. In Proceedings of the First International Workshop on Internet of Vehicles and Vehicles of Internet (pp. 31-36). ACM.

[18] https://software.intel.com/en-us/iot/hardware/edison

[19] project-byzantium.org/faqs/

[20] https://plus.google.com/communities/109339235992496491942

[21] http://www.silverlighthack.com/post/2011/02/27/IaaS-PaaS-and-SaaS-Terms-Explained-and-Defined.aspx

[22] F. Luo, Z. Y. Dong, Y. Chen, Y. Xu, K. Meng, K. P. Wong, "Hybrid cloud computing platform: The next generation IT backbone for smart grid", Proc. IEEE Conf. Power Energy Soc. Gen. Meet., pp. 1-7, 2012.

[23] Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: A critical review. International Journal of Computer Network and Information Security, 6(3), 20.

[24] Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A., & Buyya, R. (2015). Big Data computing and clouds: Trends and future directions. Journal of Parallel and Distributed Computing, 79, 3-15.

[25] https://github.com/urlgrey/hsmm-pi

[26] https://azure.microsoft.com/en-ca/overview/what-is-cloud-computing/?cdn=disable