International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

An Economical Incorporation of IoT and Edge/ Cloud Computing for Dynamic Distribution of IoT Analytics and Organized Utilization of Network Resources

M. Azhagiri, Sanjesh Chevanan, John Vivin, Samuel, Paul Davis

***Abstract:** As the Internet of Things (IoT) keeps on picking up application in telecommunication networks, an enormous number of devices are relied upon to be associated and utilised sooner rather than later. Wide scale Internet of Things (IoT) systems with several deployed IoT devices, such as sensors, actuators and so on, that generate a high volume of data is expected. This means that the volume of data is foretold to increase substantially. Customarily, cloud services have been executed in huge datacenters in the central network. Be that as it may, this is definitely not a long-haul adaptable choice as an exceptionally enormous number of gadgets are required to be associated and utilised sooner rather than later. An adaptable and productive arrangement is to disperse the IoT analytics between the core cloud and the network edge. This paper uses edge IoT analytics to viably and economically convey the information from the IoT between the core cloud and the network edge. First analytics can be completed on the edge cloud and just the essential information or results are sent for further investigation. We have identified an approach to make this operation increasingly affordable.*

***Keywords:** IoT, Edge Computing, Datacenter, Cloud Network, Dynamic distribution*

I. INTRODUCTION

The Internet-of-Things (IoT) is, where everything will be associated. Since the Internet-of-Things has been presented, it is considered as one of the rising innovations giving incredible chances to numerous vertical enterprises. The Internet of Things gives the capacity to people and computers to take in and interface from a great number of things that incorporate sensors, actuators, services, and other Internet-associated objects.

Revised Manuscript Received on November 30, 2019.
* Correspondence Author
M. Azhagiri, Department of Computer Science and Engineering, SRM Institute of Science and Technology
Sanjesh Chevanan, Department of Computer Science and Engineering, SRM Institute of Science and Technology
John Vivin, Department of Computer Science and Engineering, SRM Institute of Science and Technology
Samuel, Department of Computer Science and Engineering, SRM Institute of Science and Technology
Paul Davis, Department of Computer Science and Engineering, SRM Institute of Science and Technology

A gigantic proportion of detecting gadgets produce and get data after some time for a wide extent of fields and applications. Gathering such enormous volumes of information form IoT devices and analysing it utilising a moderately modest number of core datacenters isn't versatile as the number of IoT devices are anticipated to increment significantly.

Studies have uncovered that cloud/edge computing based services will assume a significant job in broadening the cloud via doing intermediate services at the network edge and help encourage versatility. Edge figuring empowers information gathered by Internet of Things gadgets to be put away in and prepared by neighbourhood cloud/edge hubs just as permits IoT clients to get to IoT applications by means of these hubs simultaneously. This will encourage a wide assortment of utilisation cases from various vertical ventures, for example, automobile and mobility, healthcare, energy, media, future factories, or energy.

II. LITERATURE SURVEY

[1]Babatunji Omoniwa in 2018 introduced an extensive overview on cutting edge IoT writing over the period 2008-2018 and propose the FECIoT network which covers the passing innovations, services, and open research issues[1].

[2]Shree Krishna Sharma and Xianbin Wang in 2017 introduced a paper to recognize and portray the potential key empowering influences for the proposed edge-cloud community framework, the related key difficulties and some fascinating future research bearings[2].

[3]Bin Cheng in 2017 distributed a paper to showcase how smart city use cases can be acknowledged with FogFlow, we portray three use cases and actualise a model application for irregularity location of energy utilisation in smart urban areas. They likewise investigate FogFlow's presentation dependent on microbenchmarking results for message propagation latency, throughput, and versatility[3].

[4]Hongming Cai in 2016 exhibited a paper to report an investigation of basic IoT application distributions and research themes dependent on related scholarly and industry publications[4].

[5]Byungseok Kang in 2017 proposed a paper to coordinate protocol for networks, help control storage and edge analytics on the information, and make information stream safely between edge devices and the cloud[5].

[6]Anne H. Ngu in 2016 published a paper to direct an exhaustive investigation of the difficulties and the empowering advances in building up an IoT middleware that grasps the heterogeneity of IoT gadgets and furthermore underpins the basic elements of composition, versatility, and security aspects of an IoT framework [6].

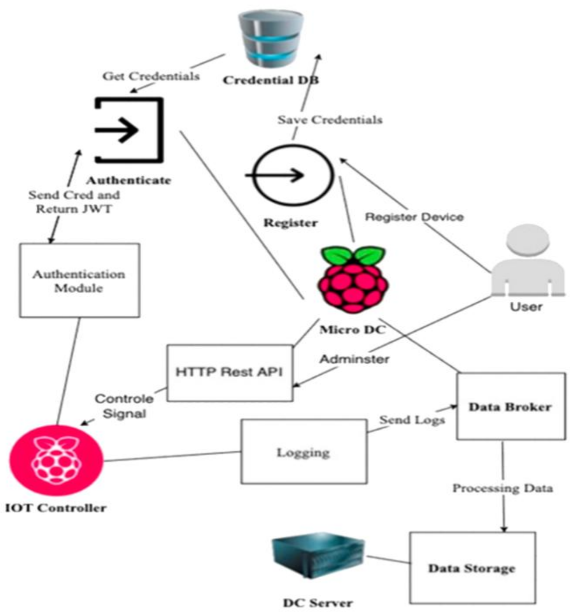
III. INCORPORATION OF IoT AND EDGE/CLOUD COMPUTING USING ARM BASED SMALL DATACENTER

The implementation of small datacenter is carried out using ARM based Raspberry pi. The Raspberry pi zero is used as the IoT device/IoT Controller, Raspberry pi 4 used as the Micro Datacenter and Intel laptop used as the main DC server.

A. Authentication of IoT device and distribution of IoT analytics

1. Begin
2. If request_user == admin
3. then
4. data = request_accept
5. If data == True
6. then
7. control = true
8. else
9. control = false
10. while connection
11. then
12. connection.senddata(DC)
13. End

1. **Deployment of Micro DC at the network edge**



2. **Incorporation and dynamic distribution of IoT analytics**

IV. OPERATION OF DYNAMIC DISTRIBUTION OF IoT ANALYTICS

For the IoT device to be authenticated and controlled we are using the JWT web token mechanism to authenticate the IoT node. First, we need to authenticate the user as the admin user in the micro DC here also we are using the JWT token mechanism to authenticate the user. First, the user credentials are sent to the user login API running in the micro DC and the credentials are checked with the user database (the prerequisite is that the admin user credentials are already added to the database). If the user credentials match, the API sends back an admin access JWT token. This token is used to control or modify the process running in the micro DC without which the processes can’t be controlled or modified. After obtaining admin access we need to register the node device without which the micro DC can’t find the individual node in the network so the registration credentials are the “node name” and “IP address”. These aforementioned are the credentials which need to be sent to the micro DC to register API from the admin user but the credentials which are added to the database are “id”, “public id”, “node name”, “encoded JWT”, “IP” and “updated time”. The public id is a randomly generated ID by using uuid4 in the UUID module. The encoded JWT is the encoded version of the data that is sent to the register API. The encoding uses SHA256 algorithm to generate a hashed data using a pass-key for the encoded data to be decoded. We need to get the pass-key and this is always kept as an environment variable which will make the data more secure and easy to access. Next, the update time is the time when the node credentials are added to the database so the added time of the node can be identified. The search API which is used to search the database of the nodes with the public key as the index, if we are authenticated as the admin we will get all the data returned. Using this we can get the encoded JWT token of the node and use it to control the node. For controlling the node there will be an API running in the node which uses a GPIO module to control the device which is attached to the node.

A. Flask:

Flask is a small scale system which is utilized to manufacture dynamic web-servers and API. We can utilize various kinds of strategies with the flask API like GET, POST, DELETE and so forth to adjust the data. It has no database abstraction layer, form validation, or whatever other segments where previous third-party libraries provide common function.

B. Flask_sqlalchemy:

Flask-SQLAlchemy is an augmentation for Flask that adds support for SQLAlchemy to your application. It expects to rearrange utilizing SQLAlchemy with Flask by giving valuable defaults and additional aides that make it simpler to achieve common tasks.

Retrieval Number: A4921119119/2019©BEIESP
DOI: 10.35940/ijitee.A4921.119119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

C. JWT:

JWT is utilised for basic web authentication and persistency connection. JWT uses HS256 algorithm to hash the data that is given by the user and stores it in both the server and the client so the connection can be persistent.

D. SQLite:

SQLite is a relational database management system (RDBMS) contained in a C library. As opposed to numerous other DBMS, it isn't a client–server database engine. It is rather inserted into the final program. SQLite is ACID-compliant and actualises the vast majority of the SQL standard, by and large following PostgreSQL syntax. Be that as it may, SQLite utilises a progressively and weakly typed SQL syntax that doesn't ensure the domain integrity.

E. UUID:

Universal Unique Identifier is a python library that helps generate random objects of 128 bits as ids. It gives the uniqueness as it produces ids based on time, Computer equipment (MAC and so on.).

F. Datetime:

Datetime is a python module which is used for calculations with date and time. There are many functions in this module for getting the current time and for generating time stamp.

G. Kafka:

Apache Kafka is an open-source stream-processing software platform created by LinkedIn. The task expects to give a brought together, high-throughput, low-latency platform for dealing with constant information feed. Kafka can associate with outside frameworks by means of Kafka Connect and gives Kafka Stream, a Java stream processing library.

H. Docker:

Docker is a set of PaaS products that utilise OS-level virtualisation to convey programming in bundles called containers. Containers are disconnected from each other and pack their very own software, libraries and configuration files. They can connect with one another through well-characterised channels. All containers are controlled by a sole operating-system kernel and are accordingly increasingly lighter than virtual machines.

V. RESULTS

The existing system proposes a very developable and efficient solution is to spread out the IoT analytics between the core cloud and the edge of the network. However, the implementation of micro datacenter and small datacenter is very expensive and can be excessive for smaller IoT devices. In the existing model, the micro DC is an Intel NUC Kit with and Intel i7 6770HQ processor and 32 GB RAM with a TDP of more than 60W. This setup would cost upwards of $600 and may be over-kill for certain IoT devices. Our proposed model consists of an ARM based raspberry pi which costs less than $70 and a power draw of less than 12W at maximum load. The proposed model is capable of handling a considerable amount of load from small scale IoT devices. The results indicate that the proposed system is a much more economical version of the existing system. We introduce ARM-based micro and small data centres that are a fraction of the cost and can handle a significant amount of data. ARM's future is certain for prominence due to its software integration and adaptability. Because of this software integration, the power consumption is significantly lesser and is highly customisable. ARM based devices can also operate on a considerably smaller form factor.

VI. CONCLUSION

Internet of Things (IoT) is an undeniably prominent innovation that empowers gadgets, vehicles, household apparatuses, and so forth., to impart and even entomb work with each other. It has been generally utilised in industries and social use including smart home, healthcare services, and mechanical automation. Since the number of IoT devices are forecasted to grow significantly in the near future, scalability must be feasible and efficient. This paper proposes a very scalable and cost-efficient option to address this issue.

REFERENCES

1. Babatunji Omoniwa ; Riaz Hussain ; Muhammad Awais Javed ; Safdar Hussain Bouk ; Shahzad A. Malik, Fog/Edge Computing-Based IoT (FECIoT): Architecture, Applications, and Research Issues, 2018[1].
2. Shree Krishna Sharma ; Xianbin Wang, Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks, 2017[2].
3. Bin Cheng ; Gürkan Solmaz ; Flavio Cirillo ; Ernö Kovacs ; Kazuyuki Terasawa ; Atsushi Kitazawa, FogFlow: Easy Programming of IoT Services Over Cloud and Edges for Smart Cities, 2017[3].
4. Hongming Cai ; Boyi Xu ; Lihong Jiang ; Athanasios V. Vasilakos, IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges, 2016[4].
5. Byungseok Kang ; Daecheon Kim ; Hyunseung Choo, Internet of Everything: A Large-Scale Autonomic IoT Gateway, 2017[5].
6. Anne H. Ngu ; Mario Gutierrez ; Vangelis Metsis ; Surya Nepal ; Quan Z. Sheng, IoT Middleware: A Survey on Issues and Enabling Technologies, 2016[6].
7. https://ieeexplore.ieee.org (Internet source)[7]
8. www.ijiert.org (Internet source)[8]

AUTHORS PROFILE

Sanjesh chevanan is studying computer science and engineering at SRM Institute of science and technology. He is interested in data analytics and cloud computing. He has no previous publications or research work.