Thesis No: CSER-M-18-04

A STUDY ON BIG DATA MANAGEMENT STRATEGY USING FOG COMPUTING

By

Tajul Islam

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
November, 2018

A Study on Big Data Management Strategy Using Fog Computing

By

Tajul Islam

Roll No: 1607502

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Computer Science and Engineering

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

November, 2018

Declaration

This is to certify that the thesis work entitled "**A Study on Big Data Management Strategy Using Fog Computing**" has been carried out by Tajul Islam in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

28/11/2018

Signature of Supervisor

Tajul 28/11/18

Signature of Candidate

Approval

This is to certify that the thesis work submitted by Tajul Islam entitled "A Study on Big Data Management Strategy Using Fog Computing" has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in November, 2018

Board of Examiners

1. 28/11/2018 ________________ **Chairman** (Supervisor)
Dr. M. M. A. Hashem
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

2. Aminul 28/11/18 ________________ **Member**
Dr. Muhammad Aminul Haque Akhand
Professor & Head
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

3. 28.11.18 ________________ **Member**
Dr. K. M. Azharul Hasan
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

4. 28.11.18 ________________ **Member**
Dr. Sk. Md. Masudul Ahsan
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

5. 28.11.18 ________________ **Member** (External)
Dr. Md. Anisur Rahman
Professor
Computer Science and Engineering Discipline
Khulna University
Khulna, Bangladesh

Acknowledgement

At first, I would like to thank Almighty for showering all his blessings on me whenever I needed. It is my immense pleasure to express my indebtedness and deep sense of gratitude to my supervisor Dr. M.M.A. Hashem, Professor, Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET) for his continuous encouragement, constant guidance and keen supervision throughout of this study. I am especially grateful to him for giving me his valuable time whenever I need and always providing continuous support in my effort.

I am especially grateful to all the faculty members of the Department of CSE, KUET to have their privilege of intensive, in-depth interaction and suggestions for the successful completion of my master degree.

At last I am grateful to my parents, family member and friends for their patience, support and encouragement during this period.

November 2018 Author

Abstract

In modern ages, data is increasing day by day using internet of things for various purposes. Data intensive analysis is the major challenge because of the ubiquitous deployment of various kinds of sensors. Traditional cloud computing infrastructure is not enough for processing these large amount, variety or velocity of data that means big data. Traditional cloud computing structure is geographically centralized. Balancing load for big data is a crucial issue. A preprocessing stage is necessary to handle these big data for real time service oriented application. In this thesis, a big data management strategy is proposed using a preprocessing stage that is fog computing. Providing real time services from cloud is too many time consuming for big data. Here, Fog Computing plays a vital role. The maximum functions of processing data are implemented outside of cloud in the case of Fog Computing and one thing considered in Fog Computing is that here memory of fog devices is very little. So, a well-organized communication system is needed for data processing. Here, in this work an affordable, robust and secure power supply or third party memory management has been suggested which is Smart Grid and Smart Local Grid. The smart grid and smart local grid or third party memory management support the customer's real time services using fog infrastructure. In this work a hierarchical architecture has been proposed for creating well-organized communication system for data processing in the case of smart grid and smart local grid or third party memory management using fog infrastructure or nodes. For task scheduling of nodes in case of fog infrastructure queue based scheduling technique is used. In this work, an effective result for big data management and providing real time services has been found. Here, different parameters such as network latency, throughput have been used for measuring performance in real time services. The overall network latency is minimized and throughput is increased in case of fog computing.

Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

1.1 Background

In recent years, the number of Internet of Things or sensors has increased to a great extent for increasing smart things. So, large number of data is generated from IoT devices. These generated data are called big data because of large volume, variety, velocity, value and veracity. In case of big data volume refers the large amount of data, velocity refers to the speed at which vast amounts of data are being generated, collected and analyzed, value refers to correlation of the data, veracity refers to originate and availability of data and variety means data formation such as data is structure or unstructured.

The technology of IoT is reliant on cloud computing. Data from the billions of Internet-connected devices are voluminous and demand to be processed within the cloud data centers. Most of these IoT infrastructures, such as smart vehicular traffic management systems, smart driving and car parking systems, and smart grids are observed to demand real-time, low-latency services from the service providers. Since conventional cloud data centers computing involves processing, computation, and storage of the data only within , the massive data traffic generated from the IoT devices is anticipated to experience a huge network bottleneck and, in turn, high service latency and poor Quality of Service[1][2]. Moreover, in order to process and serve this high number of requests the data centers are required to be up and running around the clock which results in the consumption of enormous amount of energy and massive emission of CO_2[1].

Currently, the "pay-as-you-go" Cloud Computing paradigm is widely used in enterprises to address the emerging challenges of big data analysis because of its scalable and distributed data management scheme [2]. However, data centers in the Cloud faces great challenges on the burden of exploding amount of big data and the additional requirements of location awareness and low latency at the edge of network necessary for smart things [3].Nowadays big data is considered a crucial research interest with the progress of internet of things and also for internet of everything technologies. Recent technologies named cloud computing is not adequate for big data mainframe transmitting. Therefore, new strategies are necessary for reducing the intricacy and also for easiness in handling of big data. For reducing the load of big data in case of cloud computing, impure data pre-handling can be considered as the most intelligent ways. To minimize the load of managing data from cloud, fog computing paradigm plays an important role in pre-management. A hierarchical, layer wise and distributed fog computing architecture gives storage near to the outline sensors for latency perceiving applications.

1.2 Motivation

In modern ages, big data mainly generated from IoT devices or client tier. Central processing and storing of data are highly costly and real time services providing is not effective for IoT devices/Client tier [8]. So, processing and storing a big data is a vital issue in case of traditional cloud data. If we preprocessed these large volume of data that is helpful for load-balancing in cloud computing and providing real time services for latency sensitive application easily. In case of big data real time is related to velocity feature. In Figure 1.1 shows five properties or dimensions of big data. The properties are volume, velocity, value, veracity and variety.

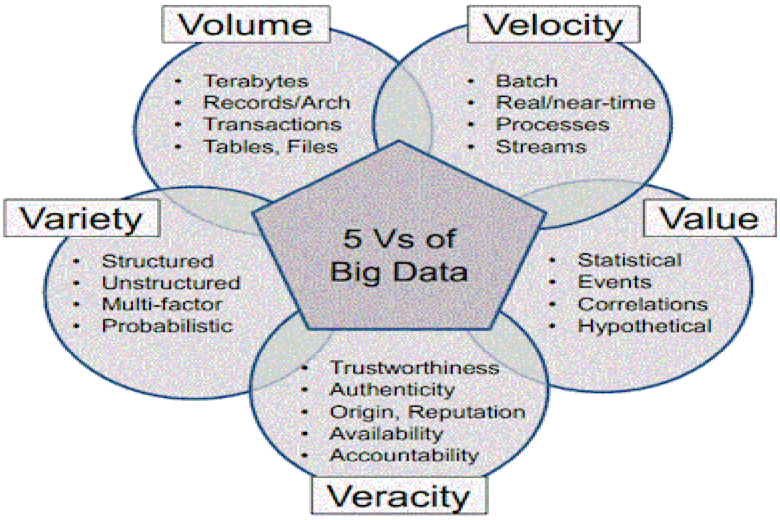

Figure 1.1: Big Data Properties [6]

Fog computing is a distributed computing paradigm that acts as an intermediate layer in between Cloud datacenters and IoT devices/sensors. It offers compute, networking and storage facilities so that Cloud-based services can be extended closer to the IoT devices or sensors [1]. IoT devices or sensors are highly distributed at the edge of the network along with real-time and latency-sensitive service requirements. Since Cloud data-centers are geographically centralized, they often fail to deal with storage and processing demands of billions of geo-distributed IoT devices/sensors. As a result, congested network, high latency in service delivery, poor Quality of Service is experienced [2].

Fog computing enhances cloud computing and services provided by the network. Many devices can be connected with the Internet of things (IoT) and can run straight away at the network border by enabling the applications. Generally, there is a precise inter connection between cloud and Fog Computing at the big data management and analytics sector. Fog Computing has the ability to alleviate service latency, take measures to provide location knowing and enhance quality of service. For analyzing real time big data, Fog Computing can be considered well organized. It clinches deeply distributed information collection points. Fog devices are little memory so system is needed another third party management services. We had chosen a strong, affordable and invulnerable system that is smart Grids. Smart Grids can be formatted with Controls, Computers and automation. Here we have used BFS order for Smart Grid in case of numbering. Through breath first search, the nodes can be numbered and can be easily found. So it will be easy for managing data and it can also ensure security. Smart Grid can be considered as a summation of infrastructure of communication and electrical network. It is constructed by power network having load balancing capability. This network can run on many devices through flexible, dependable, scalable, in secure and in efficient ways. It traces accessibility and low cost scheduling based on demand of energy.

For sketching the future smart grid, fog computing can be considered as playing amazing roles of inspiration. Smart Local Grid consists of many micro-grid network and communication infrastructure. Integrating information and solution to the local level is the main focus of smart local grid. Fog and Cloud always maintain a favorable relationship. Cloud-fog interface is the main limitation considered so far. Fog is unable for ensuring the large storage and complex massive computation. In addition big data consists of large size, quantity, verity, severity of data. So, organizing and processing these large volumes of data is an important issue. In this work proposed architecture so that fog memory related issue and big data management related issue can be easily understood.

In Figure 1.2, it is essentially three tier architecture. The tier1 combined with majorly smart, wireless sensor nodes that sense heterogeneous physical parameters and transmit the same to the immediate upper tier. The tier2 or the middle layer is also known as the fog computing layer. The uppermost tier is commonly known as the cloud computing tier. Several high-end servers and data centers comprise this tier.

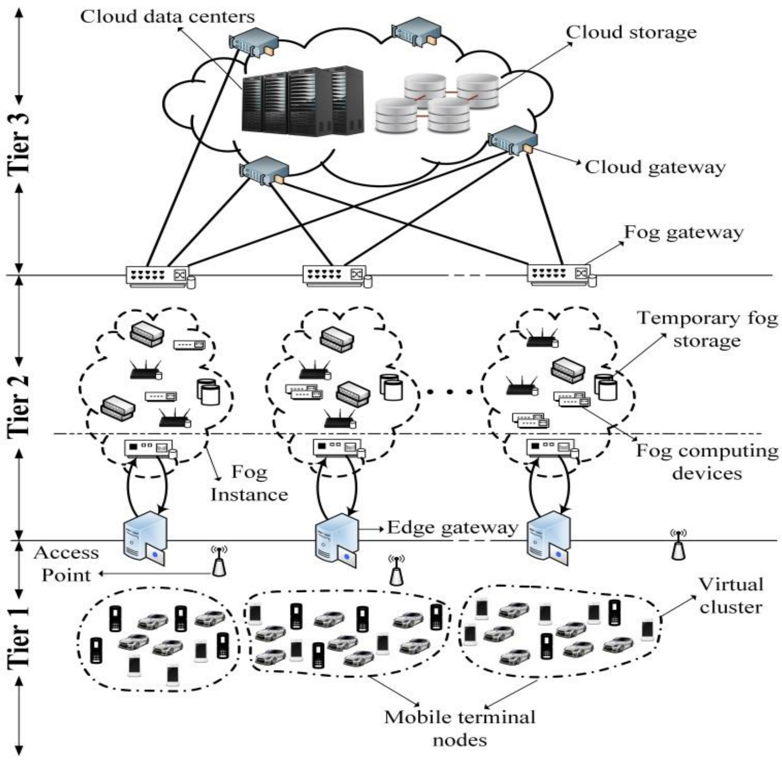

Figure 1.2: Fog Layer Presentation [1]

1.3 Problem Statement

Existing Fog infrastructure and data management in cloud has the following limitations.

- Traditional cloud infrastructure cannot provide massive processing for handling big data and it cannot provide real time services properly. In latency sensitive application, real time services are a necessary part. Big data is currently becoming a critical research focus along with the growth of Internet of Things. Relying on current technologies such as cloud computing is not efficient for addressing the requirements of big data management. Hence, new technologies are needed to reduce the complexity, edge management and boost the processing of big data.
- Fog devices use small memory for data processing. So, processing data into fog infrastructure is a vital issue. Additional computational platform is needed for processing data.

1.4 Objectives of this thesis

The classical cloud computing standard is not adequate for big data processing that is generated by internet of things. Moreover, providing real time services from cloud is too many time consuming. The objectives of this thesis given below:

- To ensure adequate computing for big Data.
- To provide real time services.
- To organize data processing model and finding fresh data or variety, value of data.
- To ensure less time for finding fault node easily from large amount of fog nodes.

Moreover, Finding fresh data from the big data and managing big data into fog infrastructure is a vital issue.

1.5 Contributions

In this work, we wanted to find fresh data from big data for minimizing data volume, varity and value. To provide real time services from locally selected storage using fog infrastructure. Summarizing our contribution:

- In this work mainly proposed a new architecture to provide real time services using fog computing. This architecture consists of internet of things, fog infrastructure, smart grid or third party memory management. This architecture was helpful for providing real time services improvement from big data easily.
- Representing different sub layers in fog computing for finding fresh data from big data. Here, we divided three sub layers: sublayer1 collecting IoT data, sublayer2 filtering unnecessary data and sublayer3 collecting fresh data.
- Designing hierarchical architecture for computing data processing into fog computing. Task scheduling into fog computing we followed master-slave model or complete binary tree structure.

Moreover, in our architecture using third party memory management we have removed memory shortage problem in fog device's data saving for real time service providing.

1.6 Organization of the Thesis

The thesis is organized as below:

- **Chapter I** briefly explains the introduction of the thesis and motivation of works. Overview of the field and the objectives of thesis are also discussed elaborately here.

- **Chapter II** discusses the related works of the thesis.
- **Chapter III** explains the proposed architecture details, real time services, managing data etc.
- **Chapter IV** explains the implementation methods..
- **Chapter V** concludes this thesis together with the outline of probable future directions of research opened by this work. This chapter also explains the experimental results and the analyzing results are also discussed this chapter

CHAPTER II

Literature Review

Fog computing architecture works as a preprocessing system to handle the big data. In traditional data processing system, data are directly sent into cloud computing layer. But now, data can be sent into fog infrastructure then go to the cloud computing layer. Fog computing architecture works as a semi-permanent processing layer. In this chapter, related works about fog computing and big data has been discussed.

2.1 Related Works

Subhadeep Sarkar et al. [1] described the difference between Fog Computing and cloud computing. They also analyzed the facilities of using Fog Computing over cloud computing. They compared various performance matrices of Fog Computing and cloud computing such as CO_2 emission, real time services providing. They suggested that Fog Computing is better than cloud computing. They didn't show usability of fog computing.

B.Tang et al. [2] discussed service providing for latency perceiving applications from fog layers. They partitioned the fog layer into few sub layers for calculation purpose. They also provides hierarchical architecture for data and control flow in case of fog computing. They didn't tell about latency sensitive application for providing real time services. Handing big data and load balancing in cloud is also a vital issue.

Rabindra K.Barik et al. [7] presented a framework which name is FogGIS. GIS means Geographic Information System. They used FogGIS framework to analyze big data in context of geospatial data. They used Fog Computing standard for reducing latency and increasing throughput at the end user. They didn't focus proper way to provide real time services and any processsing step to handle big data.

Shanehe Yi et al. [19] represented some applications based on Fog Computing. They discussed about the Smart Grid applications and components of the Fog Computing. The components of Fog Computing are location services, offloading management, Authorization and Authentication, System monitor, VM scheduling and resource management. The proper use of smart grid in fog computing is a vital issue.

Qianyu Liu et al.[37] discussed about task scheduling in internet of things for smart cities.. Here, mainly proposed task scheduling algorithm ADGTS based on adaptive double fitness genetic algorithm. They didn't focus about the task scheduling in fog computing.

Farhoud Hosseinpour et al.[38] discussed about the smart data. These smart data were found from analyzing the big data. Here also discussed about the benefits of using Smart data in perspective of fog computing. They didn't give the proper direction where the storage of these large number of data for finding smart data.

In the past decade, the concept of Smart City has drawn great interest in both science and engineering fields as a means to overcome the challenges associated with rapidly growing urbanization. A smart city is an urbanized area where multiple sectors cooperate to achieve sustainable outcomes through the analysis of contextual, real-time information or velocity improvement in case of big data.. Smart cities reduce traffic congestion and energy waste, while allocating stressed resources more efficiently and improving quality of life. For instance, in 2013, Seattle partnered with Microsoft to launch its High-Performance Building program, allowing for real-time tracking of energy efficiency, reducing energy costs and carbon emissions [1]. Smart city technologies are projected to become massive economic engines in the coming decades, and are expected to be worth a cumulative 1:565 tril-lion dollars by 2020, and 3:3 trillion dollars by 2025. Today, companies are actively vying for a central role in the smart city ecosystem, creating an expanding number of technologies. and employment opportunities. Already, IBM, Intel, GE, and many other companies have initiated projects to integrate their products and services into a smart city framework [2]. Hundreds of millions of jobs will be created to facilitate this smart city conversion; in June 2014, Intel and the city of San Jose, CA began collaborating on a project implementing Intel's Smart City Demonstration Platform, installing a network of air quality and climate sensors which alone fostered 25,000 new high tech jobs in San Jose [3].

It is expected that more than 9 billion people will live on the planet in 2050, and about two-thirds of them in cities [2][5]. This substantial population growth in urban areas will result in increasing demands on resources, services, and infrastructures in cities. Hence, to ensure the efficiency, sustainability and safety of urban communities, the cities should be becoming smarter with the intelligent technologies to integrate massive infrastructure components and services in the areas of energy, building, transportation, healthcare, education, real estate, and utilities [6][7][8]. From the above analysis, we noticed that data is generating day by day. So managing these big data is a vital issue.

While rapid urbanization provides numerous opportunities, building smart cities presents many challenges. Firstly, it is essential to build accurate, real-time, and large-scale geospatially distributed sensing networks in future smart cities to monitor the structural health of critical infrastructure components, such as bridges, gas, oil, water pipelines, roads, and subways. Secondly, the widely distributed sensor networks generate a massive volume of data, which leads to a “Big Data” analysis challenge [7][8]. Thirdly, the machine-to-machine communication among massive numbers of sensors will dominate future communication network traffic, namely Internet of Things (IoT) [11][12], instead of traditional Internet of Contents in human-to-human and human-to-machine communication. Last but not least, the integration of infrastructure components and services in smart cities requires an efficient monitoring and a quick feedback control (intelligent decision making) system to ensure the safety of urban communities. For example, an automatic valve closure is necessary when a rupture along a segment of pipeline is detected, and an optimal distribution of municipal emergency services must be made when a natural disaster is taking place. In summary, the ubiquitous deployment of sensors in smart cities requires a high-performance computing paradigm to support big data analysis with smart technologies and communications in IoT, providing location-awareness and latency-sensitive computing near the data sources.

In cloud we have some issues like dependency on the internet causing latency, limited bandwidth causing delays, security issues due to failing data protection mechanisms and requirement of high-speed Internet connectivity and that is where fog comes in. Fog doesn't work on a cloud it works on a network edge so it's faster.[11]. In cloud architecture devices were directly connected to the data center or cloud. Now we have fog in the middle to bring the Internet of Things to life by delivering distributed computing capabilities and enabling creation of an intermediate layer between the things and the cloud. Fog basically supports Internet of Things applications and in today's world of cutting-edge technology we have integrated Internet of Things into our lives. Every aspect of our lives is now monitored by this technology.

Cloud storage is the backbone of Internet of Things and now we are storing gigantic amount of expanded data, the future of big data is proceeding towards edge computing. Behind this new technology is to build a better operational connectivity between a server core and remote application reader sitting at the farthest point which we call the edge.

Fog supports Internet of Things applications that demand real-time response, especially as fog is closer to the end-user and it supports mobility at the same time [30]. Fog has less demand for internet bandwidth because data is aggregated at certain points instead of sending over cloud channels so it's faster. It has location awareness it supports stability and there's very low delay.

The necessity for extend cloud computing with the help of fog computing appeared around 2012, for coping with increasing number of IoT devices and big data volumes in order to support real-time and low-latency applications [11].

From these above mentioned work, it is observed that handling this large amount of data is a big issue. Moreover, traditional cloud infrastructure is not enough for providing real time services to client tier from the large amount of data.

CHAPTER III

Methodology

This chapter combines the concepts of Internet of Things, Big Data, Cloud Computing, Fog Computing. In below describes proposed architecture details, data management, processing and controlling data, to provide real time service and node organizing for managing big data using fog computing.

3.1 Architecture Details

The architecture shown in Figure 3.1, at first data was generated from IoT devices or client tier layer. The IoT devices or client tier layer is called layer1. Then data will be passing through fog devices for providing location awareness, reducing service latency and will improve quality of service. For processing or managing, data will come into SLG or SG. A smart grid is an electrical grid which includes variety of operational and energy measures including smart meters, smart appliances, renewable energy resources, and energy efficient resources. Electronic power conditioning and control of the production and distribution of electricity are important aspects of the smart grid. The smart local grid and smart grid layer is called layer3 and layer4.After this stage, these data will go to cloud computing. Here, Smart Local Grid works as a local storage and smart grid works as a central semi-permanent computing stage. In the beginning we said that data will be originated from client tier or IoT devices layer. These generated data are called IoT data. For processing or managing, these IoT data are sent into the fog computing layer. Fog computing was done into smart local grid and smart grid. After processing these data, we collect the fresh data and delete the unnecessary data. The fresh data are kept in locally storage area. Collecting these locally storage data, we send these whole fresh data into central storage area. Finally, we permanently store these fresh data into the cloud. Moreover, IoT devices are mainly responsible for generating the big data because smart things or smart infrastructure are growing rapidly. So, organization of these IoT sensors is a vital issue. Moreover, fog computing worked as a semi-permanent or preprocessing stages and cloud computing works as a permanent processing area. To provide real time services smart grid or smart local grid plays a vital role.

Layer1	Layer2	Layer3	Layer4	Layer5

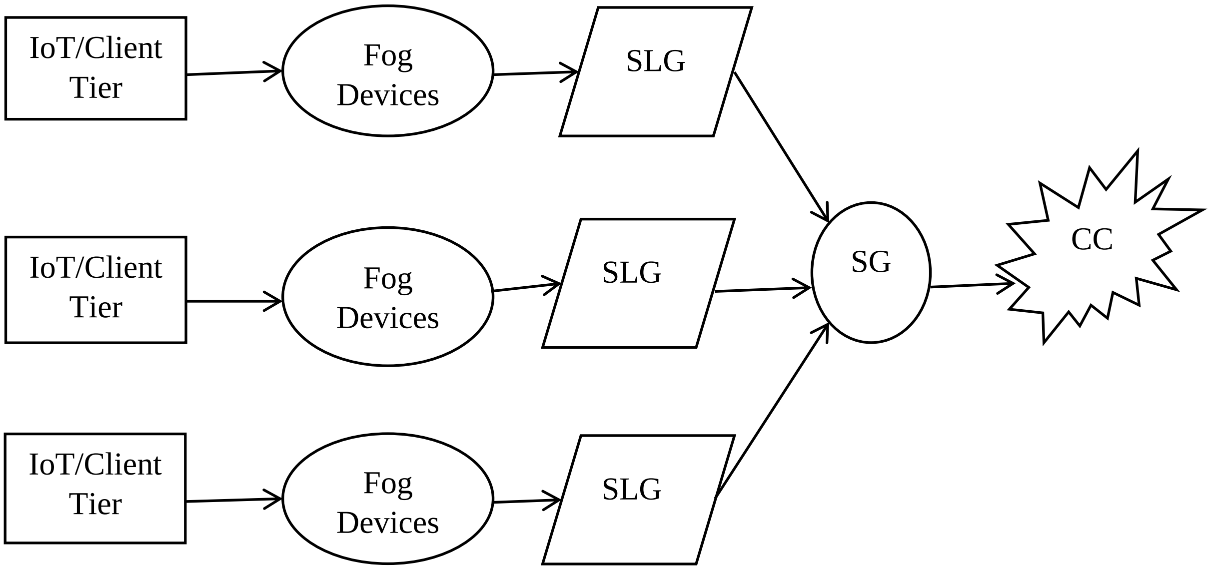

Figure.3.1: Big Data Management Architecture

3.2 Data Management

For managing or processing data into smart grid or smart local grid, the following parameters have been used in my architecture:

Data security and privacy**:** Here we have used hierarchical structure for data processing and Breadth First Search (BFS) order can be used for numbering nodes. So we can easily find problem node without visiting all of the nodes.

Cost optimization: If the architecture can be finally fixed one time, then only maintenance costs are needed.

Data storage and privacy: Distributed system can be used for data storage and privacy because it can be helpful for real time services.

Dynamic pricing: Users can maintain the price varying the number of nodes into fog infrastructure.

3.3 Processing and Controlling Data or Reduce Volume in Big Data

In Figure3.2 shows that fog computing layer divided into three sub layers. Sublayer1 has collected IoT data. Then sublayer2 is filtered and unnecessary data have been deleted. Sublayer3 has collected these fresh data. The fresh data means metadata, valuable data in respect of conditioning, meaningful data and timestamps of these should be kept also. In our architecture, Fog devices are sublayer1 and Smart Grid or Smart Local Grid works as a sublayer2 and sublayer3.

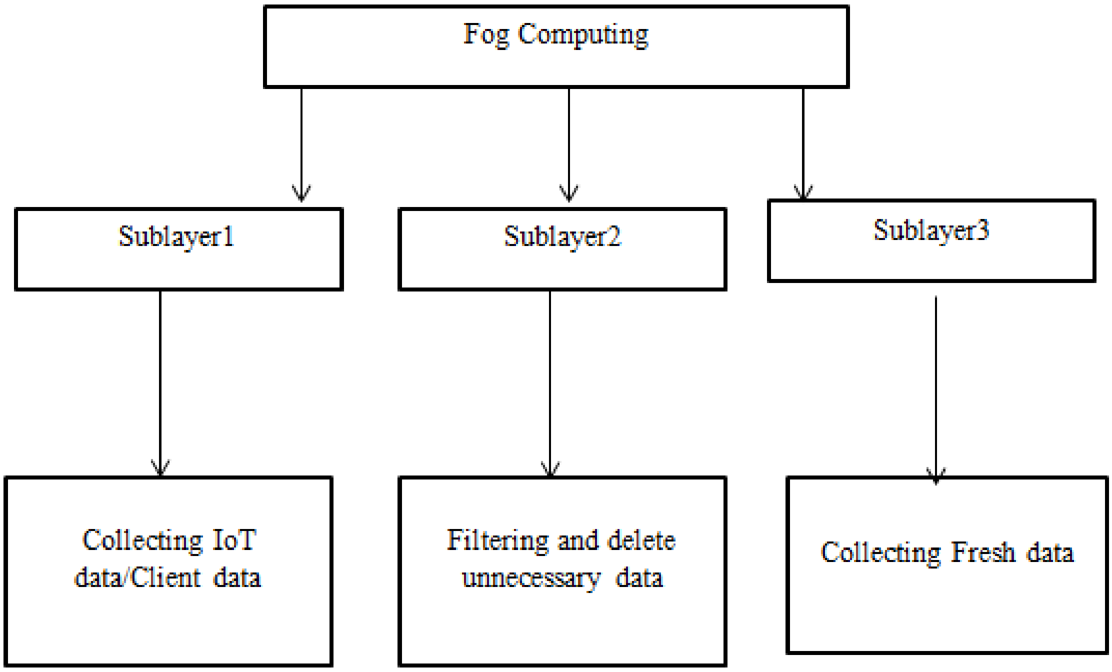

Figure.3.2: Fog Computing Sublayer to Find Fresh Data from Big Data

3.4 Real Time Service Delivery

Using fog computing we can easily provide real time services in case of latency sensitive applications. Fog devices are providing location awareness easily. One of the major problem is fog devices memory is very little. So, we introduced third party memory management in our architecture that is called local storage or smart local grid and central storage or smart grid. The benefits of local storage is when we need real time services, this local storage can easily send the necessary data. In fog computing, we have used Smart Grid or Smart Local Grid for semi-permanent storage which provides real time services, In Figure 3.3. Historical data should be sent into cloud because time is a very vital issue to provide real time services.

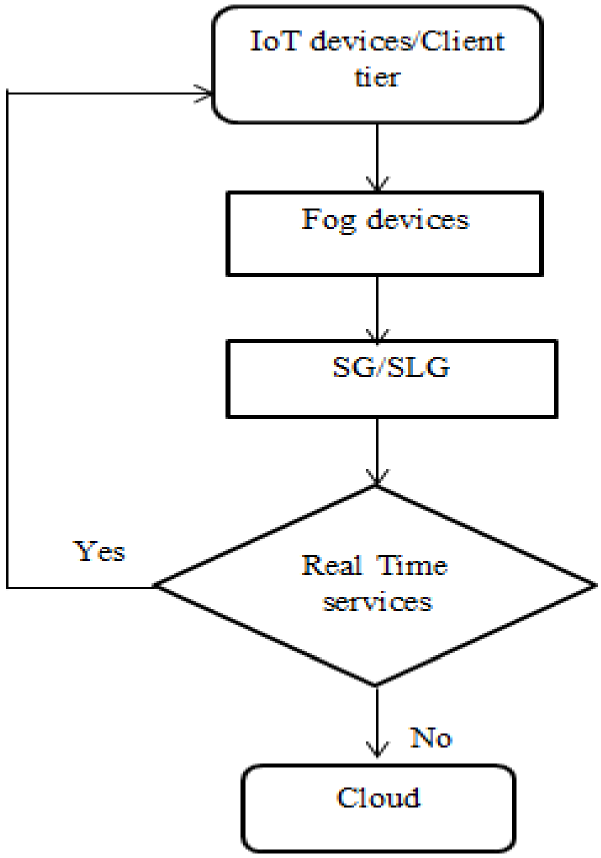

Figure.3.3: Flow Chart for Real Time Service from Fog Computing Layer

3.5 Node Organization

In case of big data veracity is one of the major parts. Node organization helps to find the originate of data easily. Here we proposed two types of node organization:

i) Master- Slave model structure.

ii) Complete binary tree structure.

3.5.1 Master- Slave Model Structure

In Figure 3.4. we have proposed a hierarchical structure for computing big data into fog layer. Here, we have processed big data as mater-slave node architecture basis. For example, we have shown 14 nodes for data processing. In this strategy, data processing easily maintained and faulty node easily found.

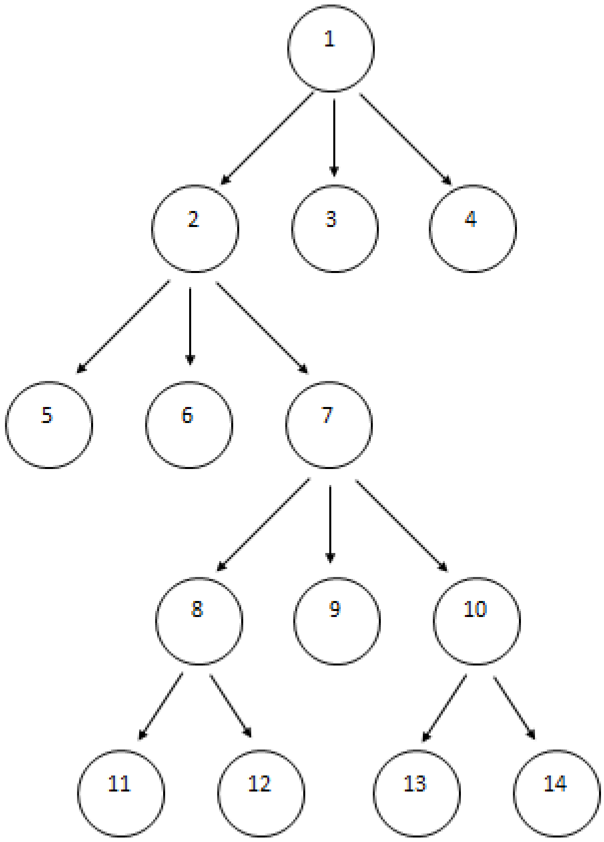

Figure.3.4: Node Organization in Fog Layer Using Master-Slave Model

3.5.2 Binary Tree Structure

A Binary Tree is considered as a complete Binary Tree if all levels are completely filled except possibly the last level and the last level has all keys as left as possible. In Figure 3.5 we used binary tree for node organization in fog infrastructure.

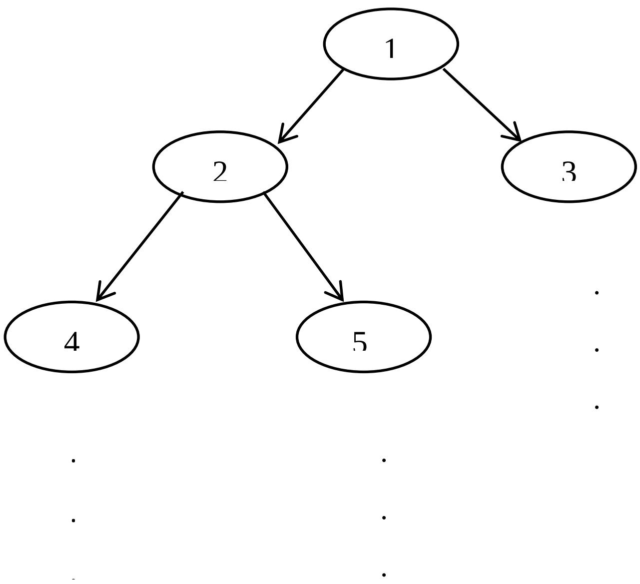

Figure.3.5: Node Organization in Fog Layer Using Binary Tree Structure

CHAPTER IV

Implementation

4.1 Experimental Tools

In this chapter describes different method and there implementation. In case of data centeric analysis,we have used Cloudsim 3.0 simulator and eclipse ide for simulation. Moreover, we used Core object-oriented programming finding output into cloudsim. Cloudsim simulation toolkit is developed using java language.

4.1.1 Cloudsim Simulator

CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centres, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning.

4.1.2 Eclipse Software

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment [40]. Eclipse is written mostly in Java and its primary use is for developing Java applications,

4.1.3 Java Programming Language

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented,and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation [41]. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

4.1.4 Arduino Software

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring [42].

For task scheduling into fog nodes we proposed an algorithm. Moreover, we wanted to find maximum data flow into our network ford-fulkerson algorithm is implemented. In case of big data for scheduling task helps to variety feature. Here, also given a real time example for finding fresh data from temperature based data and arduino prototype board and LM35 temperature sensor used.

4.2 Task Scheduling

Here, in Figure 4.1 shows the task graph consists of 9 tasks. In the table 4.1, according to task graph, we given arrival time each task and task was kept into the queue. Moreover, maintaining queue and arrival time for task. According to arrival time here each task gives into the higher priority list to complete the task. If two tasks has same arrival time, then maximum queue task number given higher priority to complete the task.

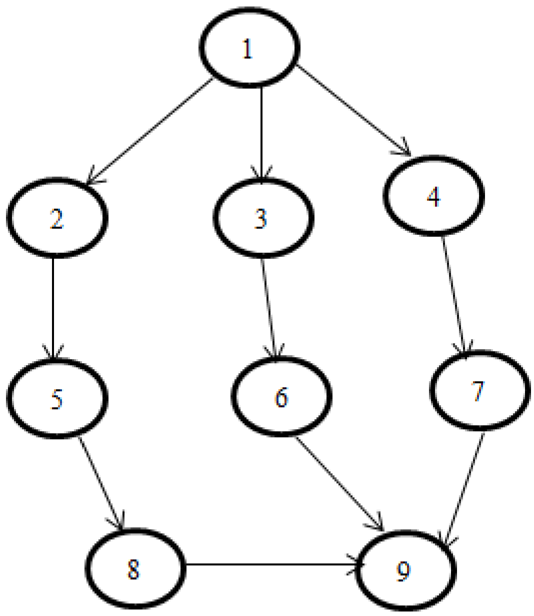

Figure.4.1: Task Scheduling into Fog Infrastructure for Big Data Processing.

4.3 Proposed Algorithm

Input: *Queue based task= qt, Arrival time= ft*
Output**:** *Task priority= tp*

1. *Sort the arrival time in ascending order.*
2. *If (Queue list is not empty)*
   2.1 *Give tp based on ft.*
   2.2 *If (finish time is equal)*
      2.2.1 *Give higher tp based on qt.*
3. *Else stop.*

TABLE 4.1: Task Priority into Fog Infrastructure for Big Data Processing.

Sl	**Queue (qt)**	**Arrival time Time(ft)**	**Priority (pt)**
1	1	5	P1
2	2	7	P4
3	3	6	P2
4	4	10	P8
5	5	12	P9
6	6	6	P3
7	7	8	P5
8	8	9	P6
9	9	9	P7

4.4 Maximum Data Flow

Here we used ford-fulkerson algorithm to find maximum data flow into a network. The ford–fulkerson algorithm is a greedy algorithm that computes the maximum flow in a flow network. It is called a "method" instead of an "algorithm" as the approach to finding augmenting paths in a residual graph is not fully specified or it is specified in several implementations with different running times.

In figure 4.2 shows node organization, initial layer is normally IoT/Sensor layer. Then, we introduced fog edge node or data center layer. Final layer is consideration for computing data that is generated from IoT/Sensor layer. This final layer consists in master-slave model. For measuring load capacity and total output capacity in master-slave model, used ford-fulkerson algorithm.

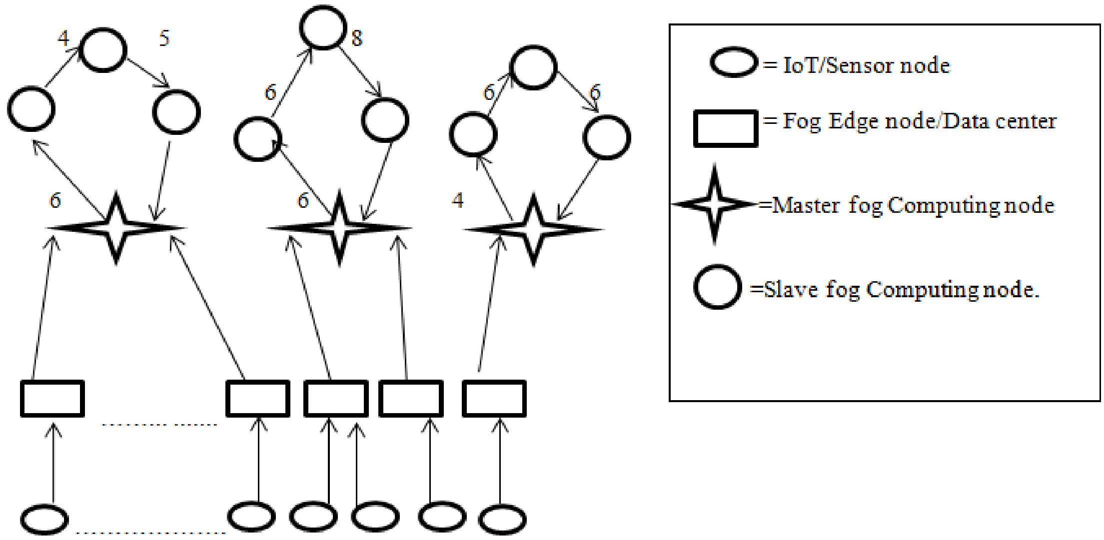

Figure 4.2: Node Organization into Different Layer for Fog Computing

In figure.4.2 seen three master-slave model. We had given different load capacity in master-slave model. Calculate different load capacity based on figure .4.2 master slave model in figure.4.3,4.4, 4.5,here used ford-fulkerson algorithm.

In the figure 4.3, 4.4, 4.5 shown some networks based on master-slave model. It is a portion of figure 4.2. In networks{\displaystyle G=(V,E)}GGG with flow capacity c, a source node su, a sink node si, u is a starting point of edge and v is a ending point of edge.

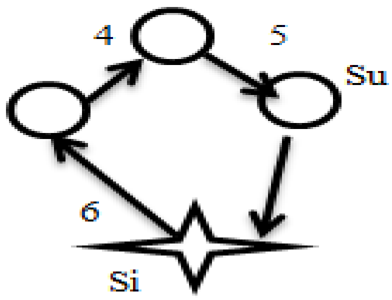

Figure.4.3: Master-Slave network model with different network capacity

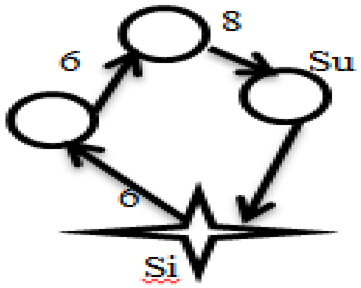

Figure 4.4: Master-Slave network model with different network capacity

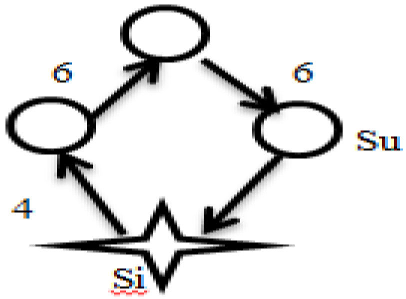

Figure 4.5: Master-Slave network model with different network capacity

If we apply ford-fulkerson algorithm the whole network capacity and output sink node value are given in the Table 4.2. The network capacity and output sink node values are helped to find out these nodes which are not assigned the task. Moreover, the capacity of the nodes those are able or not able to do the assigned work.

TABLE 4.2: Finding Maximum Data Flow into Different Network Using Ford-Fulkerson Algorithm

Source node (Su)	**Slave node (Si)**	**Maximum data flow in network**
1	3	4
2	2	6
3	3	4

4.5 Real Time Example Fog Computing

We used temperature sensor LM35 that takes data from the environment. Here, we basically try to see that instead of sending data to cloud if we processing data in our fog nodes in cloud load is balance. For data passing into fog nodes which is our local PC using arduino prototype board. Using arduino ide and doing program we collect data into our data sheet. Using different feature based on different criteria we delete the unnecessary data. When we need data for real time services we send into end user. Otherwise, we send into the permanent cloud storage. In this way, we have two benefits: one is unnecessary data are deleted into primary storage, another is real time services easily provided.

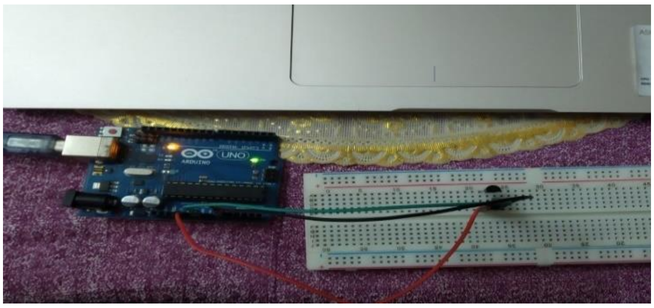

Figure.4.6: Real Time Fog Computing Example.

4.6 Data Centric Analysis

In case of providing services from same data center, it is time consuming and also services is really poor. Moreover, maintenance of data center is more costly. From the Table 4.3, we see that same datacenter works with two different tasks. When one task is completed then other task is assigned. So, in this case real time service providing is really very tough because one task dependent on another task and waited for completion the work. In this situation, fog computing paradigm plays an important role and storage problem can be easily solved. Because we knew that fog is generally worked as distributed processing.

TABLE 4.3: To Provide Service from Same Datacenter for Different Task

Cloudlet Id	Data Centre	Vm Id	Time	Start Time	Finish Time
0	2	0	80	0.1	80.1
1	2	1	160	0.1	160.1

4.7 Transmission Technology

Here we need to use MaaS (Monitoring as a Service) always because in our architecture, one stage is depended on another stage. Moreover, 5G internet facility is needed for data transfer through internet. Different types of FDS (Fog Data Service) are also be needed for communicate easily in different levels of our proposed architecture. we can use LEACH (Low Energy adaptive Clustering Hierarchy) routing protocol for our Smart Grid node organization. TORA (Temporally Ordered Routing Algorithm) routing protocol can also be used for route creation and maintenance. Moreover, for our architectural communication, Bluetooth 4 can be used in IoT devices and Fog nodes. For communicating between Cloud and Fog Nodes, socket input output programming can be used. Many wireless communication protocols are created based on IEEE 802.15.4 Such as ZigBee for the purpose of communication between fog and IoT devices. Moreover. we can use another technology such asit is necessary to impose (caas) control as a service, (maas) monitoring as a service because one layer is controlled by another layer. Moreover data transmission from one layer to another layer 5G internet services is needed. For providing security into IoT (internet of things), data blockchain technology is helpful and is an easier way. We can use LORA as well as LORA WAN protocol for providing long distance data transmission. Data collection from IoT devices or sensors fiber optic sensors also can play a vital role. Communication in fog and cloud layer socket input output programming is beneficial.

CHAPTER V

Results and Discussions

In this section, experimental result shows with network latency,throughput and data centric analysis.For finding, network latency we use different parameter such as distance,transmission medium speed, packet size, data transmission and finding throughput we have used data size,time elapsed parameter.

5.1 Performance Evolution

Performance Evolution consists of different stages. They are described below.

- Output of network latency
- Output of throughput
- Output of data centric analysis
- Real time fog computing example

5.2 Network latency

Latency is a measure of delay. In a network, latency measures the time it takes for some data to get to its destination across the network. It is usually measured as a round trip delay - the time taken for information to get to its destination and back again. The round trip delay is an important measure because a computer that uses a TCP/IP network sends a limited amount of data to its destination and then waits for an acknowledgement to come back before sending any more. Thus, the round trip delay has a key impact on the performance of the network. Latency is usually measured in milliseconds (ms)

5.2.1 Latency Importance

People often assume that high performance comes from high bandwidth, but that's not really the case.

- The bandwidth of a network or network circuit, commonly measured in Megabits per second (Mbps), refers to its capacity to carry traffic.
- A higher bandwidth means that more traffic can be carried (eg more simultaneous conversations). It does not imply how fast that communication will take place (although if you attempt to put more traffic over a network than the available

bandwidth, you'll get packets of data being discarded and re-transmitted later, which will degrade your performance)

So, latency refers to the length of time it takes for data that you feed into one end of your network to emerge at the other end (albeit we usually measure the round trip time).

- It's fairly intuitive that a bigger delay means a slower connection.
- Network latency measures the delay. It measures the time for sending data to the destination of a network. So, its measurement is vital for latency sensitive applications.

If, $Distance = di, Speed = sp, packet\ size = pas,\ Transmission\ rate = trr$ $,Propagation\ Delay = PrD, Serialization\ Delay = SeD, Network\ Latency = NeL.$

$$PrD = di/sp,$$

$$SeD = pas/trr,$$

$$NeL = PrD + SeD$$

TABLE 5.1: Finding Network latency Based on Different Distance and Same Speed, Packet Size, Data Transmission

Source to Destination (km) input1	Transmission Medium Speed (m/s) Input2	Packet size In (bytes) input3	Data Transmission (kbps)	Network Latency (output1) (ms)
5000	197863.022	1500	512	48.7075
4000	197863.022	1500	512	43.6535
3000	197863.022	1500	512	38.5995
2000	197863.022	1500	512	33.5455
1000	197863.022	1500	512	28.4915
500	197863.022	1500	512	25.9645

From the Table 5.1, we observe that the latency of a network is minimized when distance is minimized. So, it is helpful for providing real time services. Moreover, we know that fog is nearest to the infrastructure layer than cloud layer.

5.3 Throughput

Throughput is the maximum rate of production or the maximum rate at which something can be processed.

When used in the context of communication networks, such as Ethernet or packet radio, throughput or network throughput is the rate of successful message delivery over a communication channel. The data these messages belong to may be delivered over a physical or logical link, or it can pass through a certain network node. Throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second (p/s or pps) or data packets per time slot. Throughput means the amount of time needed for processing amount of data. . Moreover, throughput also mean amount of data enter and goes through the system. If data enter a large scale into a system it create traffic into the system, it may harmful for providing real time services.

If, $data\ size = ds, time\ elapsed = te,$

$$throughput = th.$$

Then, $th = ds/te$

From Figure 5.1, we saw that data size is increased then throughput time is increased. So, it is advantage to give real time service because data processing in large within per second. Fog computing is beneficial for providing real time services and balancing the load of computing.

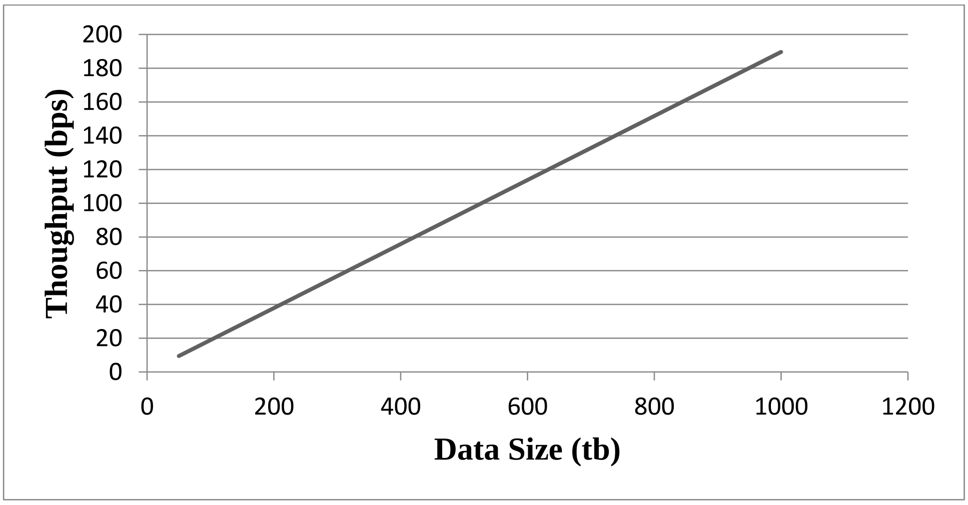

Figure 5.1: Finding Throughput from Varying Data Size.

5.4 Node Organization

For the organization of SG/SLG, we want to follow complete binary tree structure. In the Table 5.2, the node numbers are assigned according to level wise. Data processing node can be easily identified by the following formula:

5.4.1 According to binary tree

$$If, level\ number = l,$$

$$then, maximum\ node\ number = 2^{l}$$

Table 5.2: Level Calculation for Binary Tree

Level No	**Node**	**Node Number**
00	1	1
01	2, 3	2
02	4, 5, 6, 7	4
. . .	. . .	. . .

If any problem occurs in one node, it can be easily found through the analysis of problem table.. We also know that data only come from parent node as it is hierarchical data management structure. So we can say that checking node numbers is decreased. Here backtracking technique is used for checking nodes. From Table 5.3, we also see more problem node, checking node and total number of node for finding problem node

TABLE 5.3: Finding Problem Node for Binary Tree Structure

Problem Node	Checking Node	Total Node
1	0	0
2	1	1
3	1	1
. . .	. . .	. . .
10	5,2,1	3
11	5,2,1	3
12	6,3,1	3
13	6,3,1	3
. . .	. . .	. . .
64	32,16,8,4,2,1	6
65	32,16,8,4,2,1	6

5.4.2 According to master-slave model

In Figure 3.4, seen node organization for fog computing. This was the depth based hierarchical node organization. From the Table 5.4, we have observed the number of nodes in each level.

TABLE 5.4: Node Calculation for Master-Slave Model

Depth	Node	Node number
1	1	1
2	2, 3,4	3
3	5,6,7	3
4	8,9,10	3
5	11,12,13,14	4

If any problem happens in one node, it can be easily found through the study of Table 5.5. The node organization of fog computing is binary tree and master-slave modeled also. So, any problem in one node, we can solve the computational issue by only finding this master node. For finding problem node, minimum 1 node is required to search or master nodes are needed to be searched in the case of maximum.

TABLE 5.5: Finding Problem Node for Master-Slave Model

Problem Node	Checking Node(max)	Checking Node(min)	Total Node (max or min)
1	0	0	0
5	2,1	2	2 or 1
11	8,7,2,1	8	4 or 1

Now we want to tell research impact, we know that in the present world for building smart cities or using smart technologies rapidly growing the data. Processing, managing and storing these big data is a vital issue because providing real time services from traditional cloud platform is given high latency. Moreover, giving storage into cloud datacenters is not enough for this large amount of data. So, proposing these data for minimizing the amount and providing real time services we extended fog computing layers and giving the techniques of providing real time services. Moreover, for proposing this data task scheduling into nodes and load balancing is vital issue. So, we using our propose algorithm queue based to solve the task scheduling problem and for load balancing we have used Ford-fulkerson algorithm. We think that in real life, it is helpful for handling these big data and saving time, storage or into traditional data centers.

The tradition cloud computing was not enough for providing real time services for latency sensitive application. Moreover, big data management and analysis these big data providing real time services were also a vital issue. In this work, was easily providing the real time services using fog computing infrastructure. Representation different sublayers for finding fresh data from big data, this technique were beneficial for delete unnecessary data. In this work, for performance analysis measured parameter was network latency, throughput. Moreover, for maximum data flow here used ford-fulkerson algorithm. Using maximum data flow technique data traffic was easily solved. Also proposed hierarchical architecture for data processing in fog computing. Using hierarchical architecture processing data easily and faulty node easily founded.

5.5 Discussion

In this thesis, we mainly focused on evaluating the fitness of big data processing outside of cloud. Here, we have proposed an architecture using Fog Computing and third party memory management. We have used Smart Grid. Moreover, Fog Computing can be helpful for real time service providing because cloud computing is based on data center. Nowadays the capacity of data center is not enough for handling this big data and it cannot be helpful for real time analysis. So, Fog Computing is appreciable and it works as a semi-permanent storage. Here we have used smart grid for data processing computation because fog cannot provide massive storage of data.

5.6 Recommendation for Future works

In conventional fog architecture [1], they only said about providing location awareness, reducing latency. On the other hand, in our proposed strategy we have said about real time service providing technique as well as we proposed algorithm for task scheduling for handling big data. Because, in large amount of data for proposing is a vital issue. For control flow of data and computational complexity of data, we proposed hierarchical architecture. Using Hierarchical architecture, we can easily find the problem node and computational node for data. In future, we will want to work with security of node data easily provided.

REFERENCES

[1] Subhadeep Sarkar, Subarna Chatterjee and Sudip Misra, “Assessment of the Suitability of Fog Computing in the Context of Internet of Things”, IEEE Transaction on Cloud Computing, 2017.

[2] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei and H. He, “Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities,” IEEE Transactions on Industrial Informatics, vol. 3, no. 5, 2017.

[3] D. Parygin, N. Nikitsky, V. Kamaev, A. Matokhina, A. Finogeev and A. Finogeev, “ Multi Agent Approach to Distributed Processing Big Sensor Data Based on Fog Computing Model for The Monitoring of the Urban Infrastructure Systems ,” presented at 5th International Conference on Systems Modeling & Advancement in Research Trends, Moradabad, November, 25-27, 2017.

[4] M. A. Rahman, M. S. Hossain, E. Hassanain and G. Muhammad, “Semantic Multimedia Fog Computing and IoT Environment : Sustainability Perspective,” IEEE Communications Magazine, 2018.

[5] R. K. Barik, H. Dubey and K. Mankodiya, “Soa - Fog Secure Service-Oriented Edge Computing Architecture for Smart Health Big Data Analysis,” presented at Global SIP 2017, Montreal, CN, November 14-16, 2017.

[6] https://www.bbva.com/en/five-vs-big-data/ Access Date: 1-3-2018.

[7] Rabindra K.Barik,Harish chandra Dubey,ArunB.Samaddar,Rajan D.Gupta and PrakashK.Ray, “FogGIS: Fog Computing for Geospatial Big Data Analysis,” IEEE Conference on Electrical,Computer and Electronics Engineering,2016.

[8] Paola G. Vinueza Naranjo, Mohammad Shojafar, Leticia Vaca-Cardenas, Claudia Canali,“Big Data Over SmartGrid- A Fog Computing Perspective” in Vehicular Fog Energy Efficient QoS Mining Discimination Of Multimedia Big Data Streams, project by Sapienza University of Rome, 2016.

[9] Nicholas Constant et.al “Fog Assisted WIoT: A Smart Fog Gateway for End-to-End Analytics in Wearable Internet if Things,” in IEEE HPCA 2017.

[10] R. K. Barik, H. Dubey, C. Misra, D. Borthakur , N. Constant, S. A. Sasane , R. K. Lenka , B, S. P. Mishra, H. Das, and K. mankodiya, “Fog Assisted Cloud Computing in Era of Big Data Internet-Of-Things: Systems, Architectures and Applications,” in

cloud Computing for optimization : foundations, Applications, Challenges, p.23. Springer ,2018.

[11] Firas Al-Doghman,Zenon Chaczko,Alina Rakhi Ajayan,Ryszard Klempous, “A Review on Fog Computing technology,” International conference on systems. Man. and Cybernetics,IEEE,2016.

[12] Flavion Bonomi ,Rodolfo Milto, Jiang Zhu, and Sateesh Addepalli, “Fog Computing and its Role in the Internet Of Things,”in ACM Mobile Cloud Computing,2012.

[13] Farhoud Hosseinpour,Juha Plosila and Hannu Tenhunen, “ An Approach for Smart Management of Big Data in the Computing Context,” IEEE 8th International Conference on Cloud Computing Technology and Science,2016.

[14] Y.Demchenko,P.Grosso,C.De Laat and P.Menbrey, “Addressing Big Data Issues in Scientific Data Infrastructure,” in International Conference on Collaborarive Technologies and Systems, pp. 48-55,2013,.

[15] Mohammad Aazam and Eui-Nam Huh, “Fog Computing: The cloud-IoT/IOE Middleware Paradigm” IEEE potentianls, vol.35,issue.3, pp 40-44,2016.

[16] S. Muratchaev, A. Bakhtin, A. Volkov and A. Baskakov, “Modeling the Process of Network Scaling for Lorawan Basen on Ns3,” IEEE Magazine, 2018.

[17] S.Azodolmolky, P. Wieder, and R. Yahyapour. “Cloud Computing Networking: Challenges and Oppor-tunities for Innovations. IEEE Communications Magazine,” 2013.

[18] L. Angrisani, P. Arpaia, F. Bonavolonta, M. Conti and A. Liccardo, “ LoRa Protocol Performance Assessment in Critical Noise Conditions,” IEEE, 2017.

[19] Shanhe Yi, ZijiangHao, Zhengrui Qin and QunLi, “Fog Computing: Platfrom and Applications,” IEEE Conference on Electrical, Computer and Electronics Engineering,2016.

[20] V. C. Gung¨or,¨ D. Sahin, T. Kocak, S. Ergut,¨ C. Buccella, C. Cecati, and G. P. Hancke, “Smart grid technologies: communication technologies and standards,” IEEE transactions on Industrial informatics, vol. 7, no. 4, pp. 529–539, 2011.

[21] W. Su, H. Eichi, W. Zeng, and M.-Y. Chow, “A survey on the electrifica-tion of transportation in a smart grid environment,” IEEE Transactions on Industrial Informatics, vol. 8, no. 1, pp. 1–10, 2012.

[22] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proceedings of the first edition of the MCC workshop on Mobile

cloud computing, pp. 13–16, 2012.

[23] Romana Rahman Ema,Ashrafi Akram,Md.Alam Hossain and subrata Kumar Das, “Performance analysis of DSDV,AODV and AOMDV Routing protocals based on fixed and mobility network model in wireless sensor model”,Global Journal of Computer Science and Technology,vol-14,issue-6,2014.

[24] Amir vahid Dastjerdi, Harshit Gupta, Rodrigo N. calheiros, soumya K.Ghosh and Rajkumar Buyya, “Fog Computing: principles, Architectures and Applications”.

[25] Andreas kamilaris, Feng Gao,,Francesc X.Prenafeta-Boldu and Muhammad Intizar Ali, “Agri-IoT: A Semantic Framework for Internet of things-enabled smart farming applications”in IEEE 3rd World Forum on Internet of Things (WF-IoT),2017.

[26] K. Su, J. Li, and H. Fu, “Smart city and the applications,” in In-ternational Conference on Electronics, Communications and Control (ICECC), pp. 1028–1031, 2011.

[27] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, “Fog Computing: Mitigating insider data theft attacks in the cloud,” in IEEE Symposium on Security and Privacy Workshops, May 2012, pp. 125–128.

[28] Ruilong Deng, Rongxing Lu, Chengzhe Lai, Hao Liang, “Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption”, IEEE Internet of Things Journal, Vol: 3 , Issue: 6 , 2016.

[29] Prakash P, Darshaun K.G, Yaazhlene. P, Medidhi Venkata Ganesh, Vasudha B, “Fog Computing: Issues, Challenges and Future Directions” International Journal of Electrical and Computer Engineering (IJECE) Vol. 7, No. 6, December 2017, pp. 3669~3673.

[30] M. Aazam and E.-N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for iot,” in IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 687–694,, March 2015.

[31] Atlam, H.F.; Alenezi, A.; Walters, R.J.; Wills, G.B.; Daniel, J. “Developing an adaptive Risk-based access control model for the Internet of Things.” In 2017, IEEE International, UK, pp. 655 -661,2017.

[32] M. A. Rahman, M. S. Hossain, E. Hassanain and G. Muhammad, “Semantic Multimedia Fog Computing and Iot Environment : Sustainability Perspective,” IEEE Communications Magazine, 2018.

[33] [Online].Availablehttps://en.wikipedia.org/wiki/FordFulkerson_algorithm.[Accessed

Date: 25- May- 2018].

[34] Xuan-Qui Pham,Nguyen Doan Man, Nguyen Dao Tan Tri, Ngo-Quang Thai and Eui-Nam Huh, “A Cost and Performance-Effective Approach for Task Scheduling Based on Collaboration between Cloud and Fog Computing,” International Journal of Distributed Sensors Network,2017.

[35] Xuan-Qui Pham and Eui-Nam Huh, “Towards Task Scheduling in a Cloud-Fog Computing System,”Institute of Electronics, Information and Communication Engineers , 2016.

[36] Tejaswini Choudhari, Melody Moh and Teng-Sheng Moh, “Prioritized Task Scheduling in Fog Computing”, ACM SE’18, March 29-31, 2018.

[37] Q. Liu, Y. Wei, S. Leng and Y. Chen, “Task Scheduling in Fog Enabled Internet Of Things for Smart Cities,” presented at 17th IEEE International Conference on Communications Technology, Chengdu, CN, October 27-30, 2017.

[38] Farhoud Hosseinpour,Juha Plosila and Hannu Tenhunen, “ An Approach for Smart Management of Big Data in the Computing Context,” International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pp 468-471,2016.

[39] Q. Liu, Y. Wei, S. Leng and Y. Chen, “Task Scheduling in Fog Enabled Internet Of Things for Smart Cities,” presented at 17th IEEE International Conference on Communications Technology, Chengdu, CN, October 27-30, 2017.

[40] https://en.wikipedia.org/wiki/Eclipse_(software) Access Date: 1-3-2018.

[41] https://en.wikipedia.org/wiki/Java_(programming_language) Access Date: 1-3-2018.

[42] https://en.wikipedia.org/wiki/Arduino_IDE Access Date: 1-3-2018.