VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka, INDIA

PROJECT REPORT
on

"Enhancing QoS in Fog Architecture using peer-to-peer load distribution"

Submitted in partial fulfillment of the requirements for the VIII Semester

Bachelor of Engineering
IN
COMPUTER SCIENCE AND ENGINEERING

For the Academic year
2017-2018
BY

ADITYA AGARWAL	1PE14CS014
RAJEEV KONERU	1PE14CS110
SHASHANK SANKET	1PE14CS181
AMRUTHA K.	1PE14CS402

Under the Guidance of
Dr. Sudarshan TSB
Associate Director & Professor, Dept. of CSE
PESIT-BSC, Bengaluru-560100

Department of Computer Science and Engineering
PESIT BANGALORE SOUTH CAMPUS
Hosur Road, Bengaluru -560100

PESIT BANGALORE SOUTH CAMPUS

Hosur Road, Bangalore -560100

Department of Computer Science and Engineering

CERTIFICATE

*Certified that the project work entitled **"Enhancing QoS in fog-architecture using peer-to-peer communication"** is a bonafide work carried out by **Aditya Agarwal, Rajeev Koneru, Shashank Sanket** and **Amrutha K.** bearing USN **1PE14CS014**, **1PE14CS110, 1PE14CS181** and **1PE14CS402** respectively, students of **PESIT Bangalore South Campus** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2017-2018. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated and the project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.*

Signatures:

Project Guide **Dr. Sudarshan TSB** Associate Director & Professor, Dept. of CSE PESIT-BSC, Bengaluru	Head Dept of CSE **Dr. Sandesh B J** Professor, Dept. of CSE, PESIT-BSC, Bengaluru	Director/Principal **Dr. J. Suryaprasad** PESIT-BSC, Bengaluru

External Viva

Name of the Examiners	**Signature with date**
1. ______________________	______________________
2. ______________________	______________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned my effort with success.

We are indebted to our Guide, **Dr. Sudarshan TSB**, Associate Director & Professor, Department of Computer Science and Engineering, PESIT - Bangalore South Campus, who has not only coordinated our work but also given suggestions from time to time.

We are also extremely grateful to our Project Co-ordinators, **Dr. Snehanshu Saha,** Professor**, Prof. Pooja Agarwal,** Associate Professor**, Prof. Jyoti Desai,** Assistant Professors, Department of Computer Science and Engineering, PESIT Bangalore South Campus, for their constant support and advice throughout the course of preparation of this document.

We are greatly thankful to **Dr. Sandesh B J**, Professor and HOD, Department of Computer Science and Engineering, PESIT Bangalore South Campus, for his able guidance, regular source of encouragement and assistance throughout this project.

We would like to express our immense gratitude to **Dr. J. Suryaprasad,** Director and Principal, PESIT Bangalore South Campus, for providing us with excellent infrastructure to complete our project work.

We gratefully acknowledge the help lent out to us by all faculty members of the Department of Computer Science and Engineering, PESIT Bangalore South Campus, at all difficult times. We would also take this opportunity to thank our college management for the facilities provided during the course of the project. Furthermore, we acknowledge the support and feedback of my parents and friends.

Aditya Agarwal
Rajeev Koneru
Shashank Sanket
Amrutha K.

ABSTRACT

Fog Architecture aims at using edge devices to enhance user experience by distributing application load dynamically across the edge devices. An appropriate solution should be able to reduce latency, increase efficient resource usage and reduce power consumption.

The project begins with gaining a thorough understanding of current workload distribution in fog network. Simulations of same help understand the various bottlenecks and current resource efficiency.

Based on observation, enhancement in workload optimization algorithms is aimed by this work. The project also aims at providing pre-emption of services based on varying criteria such as privacy, latency, etc.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction

The revolutionary Cloud Computing paradigm resulted in the birth of millions of end-user devices connected to the Internet, supported by a myriad of interconnecting devices. Following, was the dawn of Internet of Things (IoT), a support for offloading computations to end-user devices.

Although IoT provided computational offloading, it proved to be resource intensive in every other way. More devices lead to more request, which demand more network bandwidth. More data sources, required higher data storage capacities. Power consumption and network delay were uniformly affected, parameters which are used to wholesomely judge a network characteristics.

All this while, the swarm of interconnecting devices (edge devices), supporting the cloud servers and end-user-devices, were used minimally. In 2014, Cisco proposed a new paradigm, FOG Computing, with the intent of using these edge devices for aiding computation, storage and network services. This study is aimed at providing an increment to that vision.

1.1 Purpose

FOG Computing vision could be inferred as an optimization problem where specific parameters which define network characteristics, such as Latency, Criticality, Power Consumption) are optimized. This is done by providing a better workload balancing technique.

1.2 Scope

Communication is restricted by the applications demanding it and the network protocols which make the same possible. Due to the heterogeneity of these applications and their demands, predicting network usage characteristics even approximately is a far-fetched concept, if applied to the whole Internet. The same though, is not true to a deterministic homogeneous set of applications and their network demand. The proposed solution maintains this homogeneity and deterministic view.

1.3 Literature Survey

1. Title of the paper: “Fog and Iot: An Overview of Research Opportunities”
Author: Mung Chiang, Tao Zhang
The article summarizes the opportunities and challenges of Fog, focusing primarily on the networking context of IoT. It talks about the need of new architecture to meet the stringent requirements and how Fog architecture can be evolved to meet them.
Result:

Fog enables new and disruptive business models, highly suitable in varying scenarios with huge untapped potential.

2. Title of the paper: "Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption"

Author: Ruilong Deng, Rongxing Lu, Chengzhe Lai, Tom H. Luan, Hao Liang

The paper investigates tradeoff between power consumption and transmission delay in fog-cloud computing system It further formulates it as a workload allocation problem which with optimal workload focusing on minimal power consumption with constrained service delay.

Result:

The problem is broken down into three subproblems which are solved separately. Sacrificing computation resources to save communication bandwidth and reduce transmission delay improved fog computing performance significantly.

3. Title of the paper: "Using DEVS for Modeling and Simulating a Fog Computing Environment"

Author: Mohammad Etemad, Mohammad Aazam, Marc St-Hilaire

The paper suggest an approach of analyzing both cloud-only and cloud-fog scenarios in the contect of processing delays and power consumption according to increasing number of users, on the basis of varying server load. The simulation is done through Discrete Event System Specification (DEVS)

Result:

Two different scenarios, one with cloud only and one with cloud-fog, and both were simulated using DEVS approach to showing cloud-fog outperforming cloud-only approach.

1.4 Existing Systems

Various approaches to workload allocation exist. These started from basic dynamic load distribution, similar to cluster computing to more advances optimizing techniques. The one used as benchmark was proposed by Deng which attempted to reduce power consumption while ensuring an upper limit for network latency throughout transmission. It assumed homogeneous application, capable of finishing a task independently.

1.5 Proposed System

The proposed system optimizes the workload optimization problem further buy proposing a hybrid network structure instead of a purely hierarchical one. It also introduces prioritizing of tasks and preemption if required.

1.6 Problem Statement

To enhance the current workload allocation optimization approach in order to enhance Fog Computing quality-of-service.

1.7 Summary

This chapter gives a brief introduction to the need of Fog Computing with its purpose and scope. The literature survey is done to know about the previous attempts at workload allocation in a Fog architecture. There are various existing methods, one of which are studied under existing systems in this chapter. Finally, the chapter contain the methods used by the proposed system and problem statement.

CHAPTER 2

SYSTEM REQUIREMENTS SPECIFICATION

Chapter 2

System Requirements Specifications

2.1 Software Requirements Specifications

Requirement specification is the activity of translating the information gathered during analysis into requirement document. Required specification gives a description of the intended environment for the development of the system under consideration. It is the activity of translating the information collected during literature survey and analysis into a requirement document.

System Requirement Specification (SRS) is a central report, which frames the establishment of the product advancement process. It records the necessities of a framework as well as has a depiction of its significant highlight.

2.1.1 Operating Environment

This section provide details about the hardware and software prerequisites for the project.

Hardware Requirements

- **Processor**: 2.2GHz Intel Pentium IV or 2.4Ghz AMD Athlon or faster processor
- **RAM**: 4GB or higher
- **Storage**: 5GB of available hard disk space
- **Graphic Card**: 1GB or higher (with computation capability of 3.5 or higher)

Software Requirements

- **Operating system**: Windows 7 or above, Mac OS (after El Capitan), Linux (any with GNOME)
- **Programming languages**: Java, Python
- **Documentation**: Overleaf

2.1.2 Functional Requirements

Functional requirements are a formal way of expressing the expected services of a project. We have identified the functional requirements for our project as follows:

- Setting Network Topology – Inputting specification about the servers, end-user devices and the interconnecting devices.
- Task Understanding – Realizing the needs of particular tasks in terms of power consumption, criticality and latency constraints.
- Simulation Capabilities - The proposed workload allocation should be supported through simulation

2.1.3 Non-Functional Requirements

Non functional requirements are the various capabilities offered by the system. These have nothing to do with the expected results,but focus on how well the results are achieved.

The following list of non-functional requirements is expected from the system:

- Power Consumption: The proposed allocation method should not consume more power than cloud architectures
- Latency: Network delay should be minimal, guaranteeing best QoS.
- Performance: The results should be consistent on every occurrence.

- Robust: The system should be able to handle single/multiple point of failures at-least as well as a purely cloud architecture would

2.1.4 User Characteristics

- Any user with knowledge and interest in computer networks should be able to use this paradigm.

2.2 Applications

- Private Dedicated Network: The allocation system supports homogeneous applications with deterministic characteristics. Many systems in industry sectors deploy such networks and can use the proposed system.
- Medical monitoring system: Devices and sensors deployed in hospital often form a private network with pre-determinable usage requirements

2.3 Summary

In this chapter, we understood that there are specific requirements that must be adhered to during the development of the product. The 3 functional requirements and 4 nonfunctional requirements were discussed. The application areas and the advantages of the system have also been enlisted.

CHAPTER 3

HIGH LEVEL DESIGN

Chapter 3

High Level Design

The system architecture is discussed in this chapter. We will talk about the non-functional requirements such as measuring power consumption and network delay. This stage is independent of implementation details and is more affected by the chosen architecture.

3.1 Design Approach

The design requires reliable and realistic simulation of a network, along with network characteristics such as bandwidth constraints and propagation delay, with measurement of Qos parameters.

3.1.1 Assumption and Dependencies

- User specifies the the types of devices (end-user, server, gateways/hubs) along with their links and the link specification.
- Each task has an approximate pre-determined resource requirements.

- Task can be accomplished by each inter-connecting device independently, as long as resources as available.
- The network links can be monitored for transmission success/failure and duration.

3.1.2 Goals and Constraints

The new workload allocation system must provide so ensuring that:

- The network performance is at-least as good as cloud-only architecture in any scenario.
- The network should be able to handle single point failures.
- The delay experienced should be minimal.
- Resources should be efficiently used, load distributed as uniformly as possible.

3.2 System Architecture

System Architecture shows the general structure of a network, the types of devices present and their connection. The figure below represents depicts multiple end-user-devices connected to routers, which communicate amongst themselves and the edge gateways. The edge-gateways are connected with cloud gateway which provide access to the cloud server clusters.

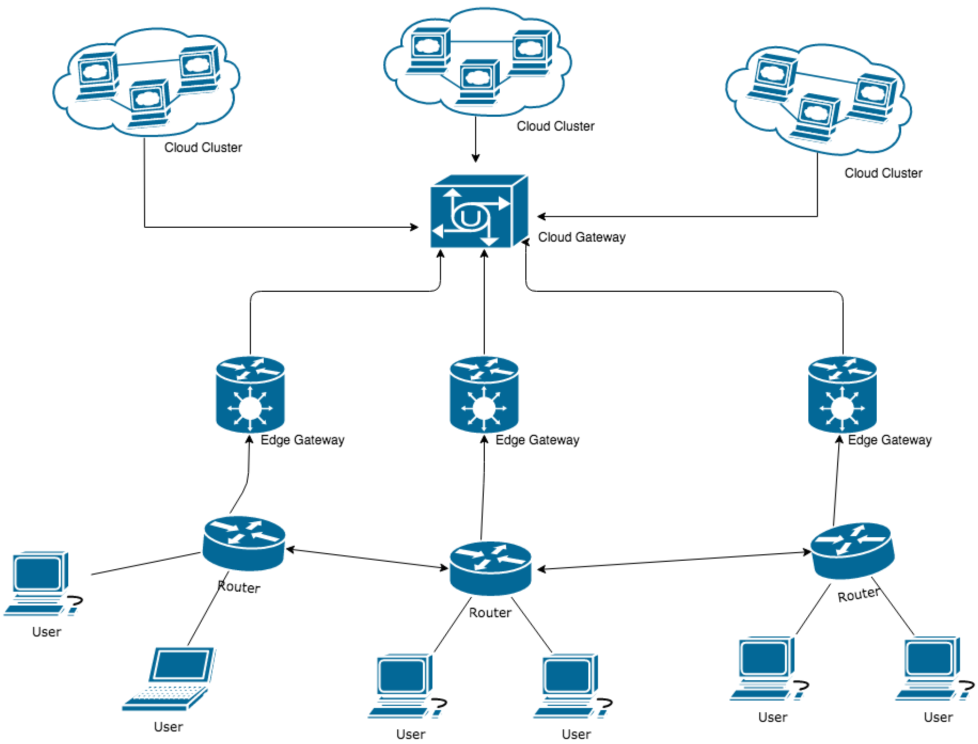

Figure 3.1: Network Architecture

3.3 Control Flow Diagram

This section provides a description of the flow of control when a new task is generated. The System starts with deciding priority of the task based on the constraints specified by the user, then determines the node which can handle it and passing it to the next node, till one can. User.

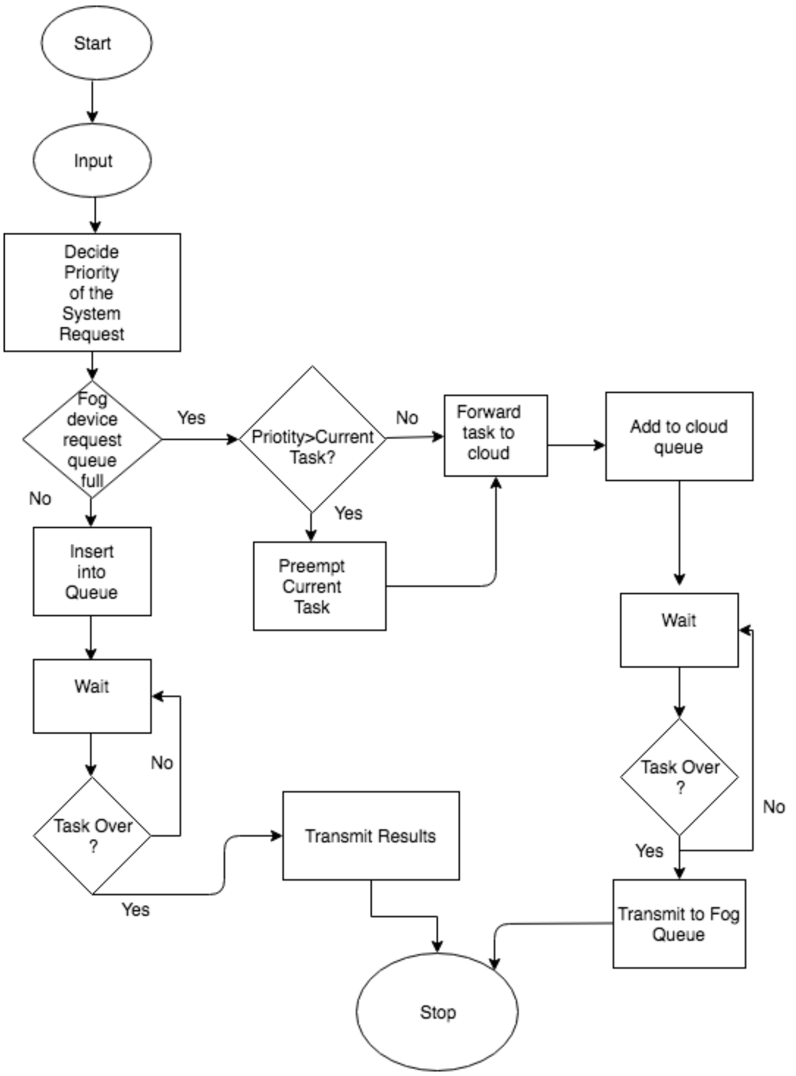

Figure 3.2: Control Flow Diagram

3.4 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how different actors operate with one another and in what order. It is a construct of a action Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

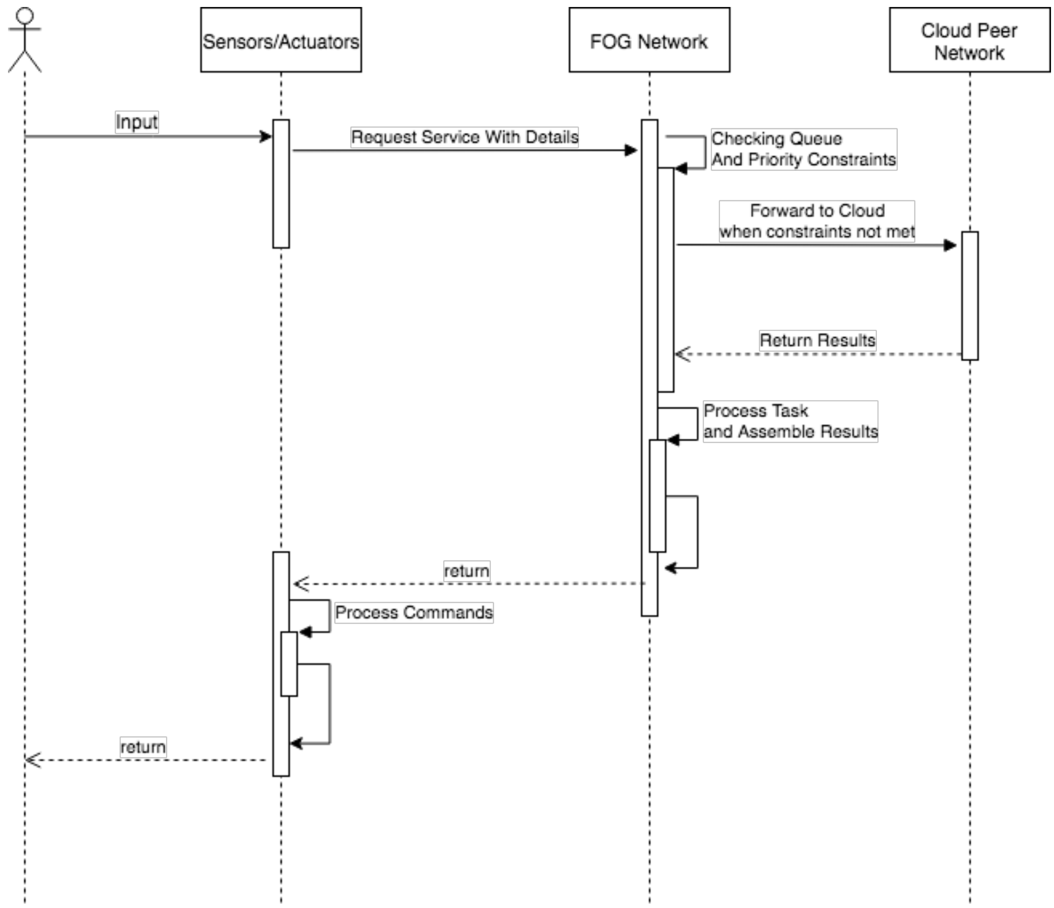

Figure 3.3: Sequence Diagram

3.5 Summary

In this chapter we discussed the different design patterns which can be used in any product development cycle. For our project mainly sequence diagram isused. We also discussed the flowchart which shows the control flow between various components. This chapter even described the high level design.

CHAPTER 4

DETAILED DESIGN

Chapter 4

Detailed Design

This phases fills in details to the high level design discussed before. Actions of components are discussed in-depth. Decision making procedure is elaborated upon, enabling easy and quick implementation of the actual system.

4.1 Network Hierarchy Model

Workload allocation models based on a purely hierarchical models suggest optimal allocation decisions based on minimal power consumption with a constrained service delay.

4.1.1 Network Parameters

Following terms are used during workload allocation:

- $\mathbf{X}_i$: Workload given a node, where total **N** nodes are present.
- $\mathbf{Y}_j$: Workload given to cloud server, where total **M** servers are present.
- λ_{ij} : Bandwidth between node *i* and cloud server *j*.

- $\mathbf{P}_i^f og$ **&** $\mathbf{P}_j^{cloud}$: Power consumption of Fog and Cloud devices.
- $\mathbf{D}_i^{fog}$, $\mathbf{D}_j^{cloud}$, $\mathbf{D}_{ij}^{comm}$: Delay at fog node, delay at cloud and delay during communication between devices.

4.1.2 Workload Balance

Let ***L*** denote the total load/tasks requested. The traffic arrival rate for a device i is l_i. This implies:

$$L = \sum_{i=0}^{N} l_i \qquad \text{...(1)}$$

Let ***X*** and ***Y*** denote the workload allocated for fog computing and cloud computing respectively. Workload balance constraints are:

$$l_i - x_i = \sum_{j=0}^{M} \lambda_{ij} \ \forall i \in [1, N] \qquad \text{...(2)}$$

$$\sum_{i=0}^{N} \lambda_{ij} = y_j \ \forall j \in [1, M] \qquad \text{...(3)}$$

4.1.3 Problem Formulation

Power Consumption in the hierarchical model is:

$$P^{sys} = \sum_{i=0}^{N} P_i^{fog} + \sum_{j=0}^{M} P_j^{cloud} \qquad ...(4)$$

The delay of the system is defined as:

$$D^{sys} = \sum_{i=0}^{N} D_i^{fog} + \sum_{j=0}^{M} D_j^{cloud} + \sum_{i=0}^{N} \sum_{j=0}^{M} D_{ij}^{comm} \qquad ...(5)$$

The problem of minimizing power consumption while guaranteeing the required delay Constrain D:

$$min_{x_i, y_j, \lambda_{ij}, n_j, \sigma_j} \ P^{sys} \ \ S.T. \ \ D^{sys} \leq \bar{D} \qquad ...(6)$$

4.2 Hybrid Model

This workload allocation model exploits communication at the same layer for load distribution, instead of strict hierarchy. This requires the introduction of security, as it might be required that the information should needs restricted access. It also entertains the idea of preemption, allowing a higher priority task to preempt a lower one. The following changes are proposed:

- Fog nodes can forward unmet requests to other Fog Nodes, instead of only Cloud Servers.
- Cloud Nodes can circulate services amongst themselves if required
- An application module can specify priorities
- A node can choose to halt a running service, to execute an incoming one of higher priority

4.2.1 Modified Network Parameter List

The peer-to-peer communication and preemption causes following additions to power supply and delay measurement:

- Preemption factor(κ) : Average ratio of services preempted due to higher priority services (in terms of Load)
- Peer-to-Peer Communication Delay (λ'_{ij}): Latency in forwarding a request from one

Fog Device to another

- Peer Acceptance Ratio (ρ): Ratio of forwarded jobs accepted by a peer device (in terms of Load)

4.2.2 Modified Delay/Speed up

Maximum Added Delay: The delay occurs due to either preemption of a service or due to chaining. Thus it is calculated as:

$$D' = D^{sys} + \kappa * L + \rho * L * [(N-1) * \lambda_{ij}^{'} - 2 * \lambda_{ij}]$$

In a general network, delay between 2 devices at a same level is significantly less than that between an node and it's corresponding gateway/cloud server.

Chapter 5

Implementation

The proposed workload allocation is simulated and visualized for.

5.1 Simulator Selection

iFogSim is a toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. It provides the functionality to:

- Specify Network Topology: Devices constituting a network along with their communication link characteristics can be specified with ease.
- Availability of CloudSim packages: iFogSim is built on top of CloudSim libraries, along simulation of cloud only and cloud-fog architectures.
- Editable Java Packages: The simulator libraries are available in Java, allowing modifications.

CHAPTER 5

IMPLEMENTATION

5.2 Platform Selection

5.2.1 Any OS with Netbeans/Eclipse support

The simulator, written in Java, has the advantage of being platform independent. Still, IDEs such as Netbeans or Eclipse allow usage of the simulator with ease and hence are preferred over raw terminal usage.

5.2.2 Java

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.

A notable feature of Java is its platform independence and a hybrid interpreter-compiler approach.

5.3 Libraries Required

- **commons-maths-3.5**: The mathematical library supporting calculation various statistical measures.
- **CloudSim**: Libraries providing simulation of a cloud network.
- **AWT**: Used to build GUI, mainly primitives, due to it's machine-dependent nature.

- **Swing**: Specify a container for window driven applications, all of which is platform-independent.

5.4 Summary

This chapter dealt with the various techniques and software platforms used in the development of the project, starting with the language and platform selection to finally explain the entire process of implementation steps.

CHAPTER 6

USER INTERFACE

Chapter 6

User Interface

6.1 Graphical User Interface

A GUI for specifying the network topology is accompanied along with iFogSim library which is extended in this study to input custom network layout and link characteristics.

For implementation of UI Java's awt & swing libraries are used.

AWT helps in specifying primitive objects. Swing is built on top of AWT and provides a light-weight container for creating window-based applications.

Following are the advantages of using Swing:

- Java swing is platform-independent.
- Swing components are lightweight.
- Swing supports pluggable look and feel.
- Swing provides more powerful components such as tables, lists, scrollpanes, color-chooser, tabbedpane, etc.

- Swing follows MVC.

Below attached snapshots provide a feel of how the user interface looks and gives an insight on it's usage. First one shows the the menu for specifying network Topology. The second demonstrates menu to monitor the network resources during simulation.

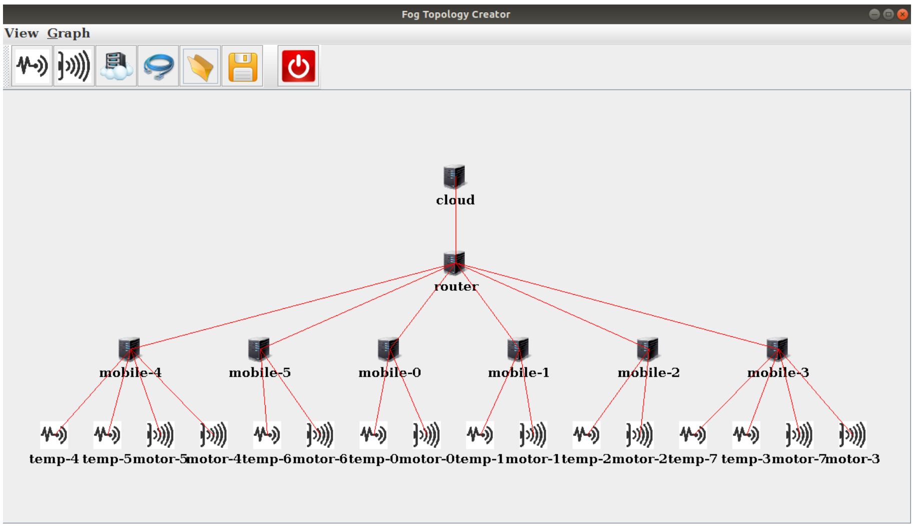

Figure 6.1: Network Topology Builder

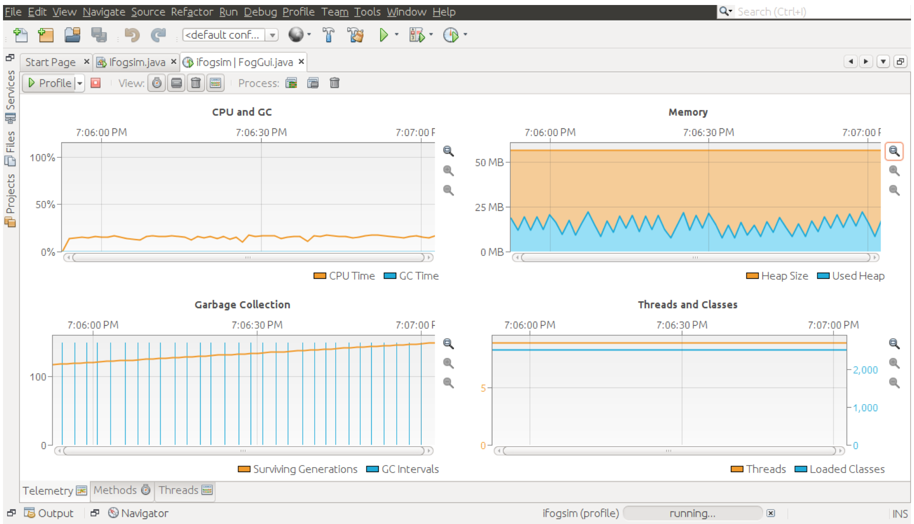

Figure 6.2: Resource Monitor

CHAPTER 7

ANALYSIS & CONCLUSION

Chapter 7

Analysis & Conclusion

7.1 Analysis

The implementation stage provided a simulator capable of testing load distribution for three types of network:

- Cloud only architecture (current standard).
- A simple Fog-Cloud architecture (Following the pure hierarchical model).
- A balanced Fog-Cloud architecture (Workload proposed by in the study).

Following additional modeling assumptions were made:

- Task bursts are uniformly distributed over all the fog nodes, with a mean 1.25/ms and S.D. of 0.05 per node.
- Service time of a task was also distributed normally with a mean of 1.0ms and S.D. of 0.05 ms.

The observed network delay, server utilization values for the 3 workload distribution approaches were recorded and are discussed below.

7.1.1 Change in Service Time

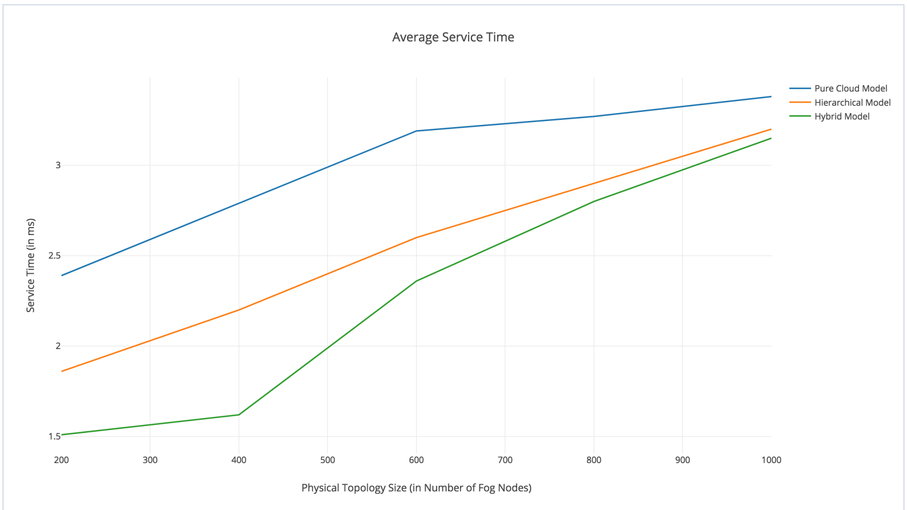

Figure 7.1: Observed Service Time

Table 7.1: Service Time Statistics (in brackets, speed up from pure-cloud model)

Number of Nodes	Pure Cloud Model (ms)	Hierarchical Model (ms)	Hybrid Model (ms)
200	2.37	1.88 (+20.67%)	1.51 (+36.28%)
400	2.77	2.23 (+19.49%)	1.63 (+41.15%)
600	3.17	2.6 (+17.98%)	2.4 (+24.29%)
800	3.25	2.87 (+11.69%)	2.78 (+14.46%)
1000	3.36	3.2 (+4.76%)	3.14 (+6.54%)

Table 7.1 presents the recorded observations and Figure 7.1 depicts a graph for the same. Following observations are made:

- Service time of Hybrid Model is 20-25% lesser than that of Hierarchical Model and 36-41% lesser than Pure-Cloud Model for smaller networks.

- The speed up achieved in Hybrid Model over hierarchical, decreases with increase in network size and is almost for large networks.
- Even for large network size, both Hybrid and Hierarchical Model are capable of providing a 5-6 % increase than Pure-Cloud Model.

7.1.2 Change in Server Utilization

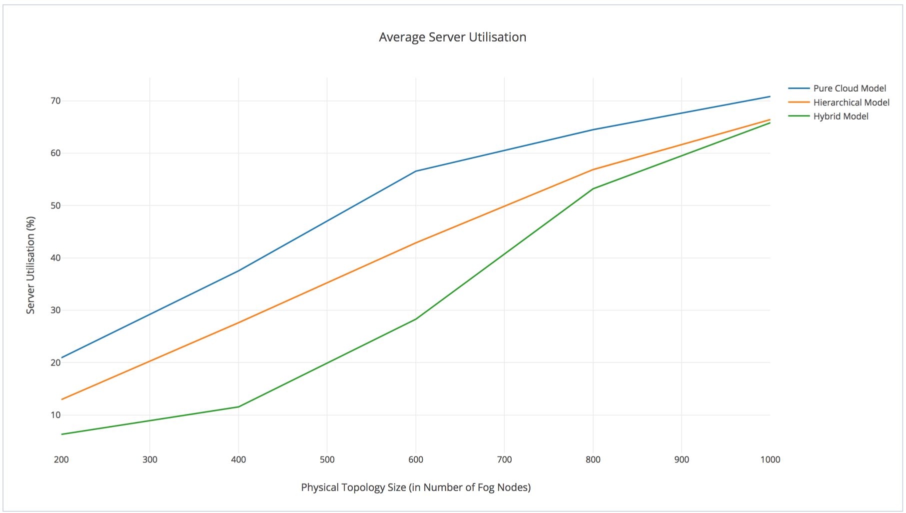

Figure 7.2: Observed Server Utilization

Table 7.2: Service Utilization Statistics (in brackets, speed up from pure-cloud model)

Number of Nodes	Pure Cloud Model (%)	Hierarchical Model (%)	Hybrid Model (%)
200	20.93	12.96 (+38.07%)	7.42 (+64.54%)
400	37.54	27.64 (+26.37%)	11.34 (+69.79%)
600	56.56	42.88 (+24.18%)	28.67 (+49.31%)
800	64.48	56.86 (+11.81%)	52.89 (+17.97%)
1000	70.81	66.4 (+6.22%)	65.8 (+7.07%)

Table 7.2 presents the recorded observations and Figure 7.2 depicts a graph for the same. Following observations are made:

- Server Utilization is lesser in smaller networks as the interconnecting edge devices are capable of handling the few task requests forwarded by low level fog devices.
- With increase in network size, the number of end-user devices increase exponentially in comparison to interconnecting devices, thus higher number of forwarded tasks to upper network level.
- For large networks, even P2P approach of Hybrid Model struggles to perform processing without cloud servers and is able to save only 7% more than Cloud Model. % increase than Pure-Cloud Model.

7.2 Conclusion

In smaller networks, number of tasks forwarded to upper network layers are small. Hence the interconnecting devices are able to handle a major part of the requests. This reduces the service time and network latency as not all the requests are forwarded to the cloud servers. Hybrid Model improves this by pushing tasks to devices in the same network layer, communication with which, is almost always faster.

As the network size increase, increase in number of interconnecting devices is significantly lesser compared to the end-user devices, requiring usage of higher network layer devices. Thus a higher percentage of requests are forwarded to cloud servers, increasing their usage, thus causing greater network delay and higher power consumption.

Hybrid Model significantly reduces the network latency by using the same layer devices to process tasks and each layer of the network. Also, since lower level devices require less power and are less expensive, there is significant drop in power consumption and cloud server costs can be reduced

7.3 Future Scope

The load distribution between clusters of fog nodes, using BSS algorithm, causes high-overhead which can be reduced, further improving performance. Also a more efficient queuing method could be generated, rather than all nodes in a cluster storing the task request.

CHAPTER 8

REFERENCES

Chapter 8

References

[1] Ruilong Deng, Rongxing Lu, Chengzhe Lai, Tom H. Luan and Hao Liang, ”Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption”, IEEE Internet of Things Journal, vol. 3, no.6, Dec. 2016, pp 1171-1181.

[2] Mung Chiang, Tao Zhang, ”Fog and IoT: And Overview of Research Opportunities”, IEEE Internet of Things Journal, vol. 3, issue 6, Dec 2016, pp 854-864.

[3] Subhadeep Sarkar, Subarna Chatterjee, Sudip Misra, ”Assessment of the Suitability of Fog Computing in the Context of Internet of Things”, IEEE Transactions on Cloud Computing, vol. 6, issue 1, Jan-Mar 2018, pp 46-59.

[4] Mohammad Etemad, Mohammad Aazam, Marc St-Hilaire, ”Using DEVS for Modeling and Simulating a Fog Computing Environment”, Workshop on Computing, Networking and Communications (CNC) 2017, 26-29 Jan, 2017, Santa Clara, USA.

[5] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, Rajkumar Buyya, ”iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments”, Journal of Software: Practice and Experience, Special Issue on Cloud and Fog Computing, Wiley, Vol 47, Issue 9, September 2017, pp 1275-1296.

[6] Bastien Confais, Adriend Lebre, Benoit Parrein, ”Performance Analysis of Object Store Systems in a Fog and Edge Computing Infrastructure”, Transactions on Large-Scale

Data and Knowledge Centered Systems, XXXIII, Lecture Notes in Computer Science, Vol 10430, Springer Berlin, Heindelberg, 2017.

TEAM INFORMATION

NAME: Aditya Agarwal

USN: 1PE14CS014

CONTACT NUMBER: 9980806627

EMAIL ID: adityag2511@gmail.com

ADDRESS: C3-011 Westend Heights, DLF Newtown, Akshayanagar, Near Ample Mart, Begur, Bangalore, Karnataka - 560076

NAME: Rajeev Koneru

USN: 1PE14CS110

CONTACT NUMBER: 8861059040

EMAIL ID:rajeev.koneru123@gmail.com

ADDRESS: 302, Ranganath Residency, Tikkle Road, Vijayawada, Andhra Pradesh - 520010

NAME: Shashank Sanket

USN: 1PE14CS181

CONTACT NUMBER: 9066249720

EMAILID:shashank.sanket1995@gmail.com

ADDRESS: D Block 301,
C/o sandeep kumar
5th cross, Attur Main Rd, Attur Layout, Yelahanka New Town, Bengaluru, Karnataka 560064

NAME: Amrutha K.

USN: 1PE14CS402

CONTACT NUMBER: 9108250883

EMAIL ID: amruthatanu57@gmail.com

ADDRESS: #1207, 3^{rd} cross, Ashok Nagar, Mandya-571401