MINOR-2 PROJECT

MID SEMESTER REPORT

Modelling Static and Dynamic Load Balancing Mechanisms in Edge Computing

Submitted by:

Name	Roll no.	Branch
Yashvardhan Singh Nathawat	R110218192	CSE CCVT
Shubham Bhatnagar	R110218151	CSE CCVT
Harsh Upparwal	R110218198	CSE CCVT

Under the guidance of

Mr. Amrendra Tripathi

Assistant Professor

Department of Virtualization

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Dehradun-248007

APPROVED BY

Mr. AMRENDRA TRIPATHI
PROJECT GUIDE

Dr. DEEPSHIKA BHARGAVA
DEPARTMENT HEAD

INDEX

S.no.	SYNOPSIS CONTENTS	Page no.
1.	TITLE	3
2.	ABSTRACT	3
3.	INTRODUCTION	4-6
4.	REPORT OVERVIEW	7
5.	PROBLEM STATEMENT	8
6.	LITERATURE OVERVIEW	9
7.	METHODOLOGY	10
8.	OBJECTIVE	11
9.	TIMELINE	11
10.	REFERENCES	12

Modelling Static and Dynamic Load Balancing Mechanisms in Edge Computing

ABSTRACT

In this project, as a part of design and implementation we have come with an architecture implementing 6 virtual machines (User, Edge server, Load balancer, 3 Main Servers) and sending tasks between them i.e., from user to edge server and to the main servers if required. Introduction of load balancing comes when the tasks are too much to perform for the edge server, then the edge server is going to transfer the tasks to the main server for computation through the load balancer which will schedule the tasks in such a way that the load on all the three main web servers remain limited. Edge computing on the other hand will be responsible for enhancing the system performance and minimizing the over burden caused at the cloud (in our case 3 main servers) by bringing down its service to the user edge. It improves the resources utilization efficiency by using the resources already available at the edge of the network. As a result, it decreases the cloud workload, reduces the latency, and enables a new breed of latency-sensitive applications such as the connected vehicles.

Keywords- *Virtual Machines, Edge Computing, Load Balancing, Latency, Load Balancer.*

INTRODUCTION

EDGE COMPUTING

It seems that the term “cloud computing” only recently "entered into our life” and another computing model appeared on the horizon - this time it is edge computing (peripheral, or boundary computing). As a real example, we can say software as a service (SaaS) instances, such as Google Apps, AWS, Instagram, and Netflix, have been widely used in our daily life.

Edge computing is a distributed computing paradigm that is used to improve response times and save bandwidth and brings computation and data storage closer to the Edge location. It’s based on CDN and performs data processing at the edge of the network, closer to users and sources of data. Serving as a bridge between physical and digital worlds, edge computing enables smart assets, smart gateways, smart systems, and smart services.

Now with IoT, a large quantity of data is generated by things that we are using in everyday life. A lot of applications will be also deployed at the edge to consume these data in the nearest future. Edge computing is going to be a significant driver for network evolution over the next five years. Spending on edge computing is forecast to be $9 billion per year by 2024, with 44% of that from North America, 28% from Europe, and 21% from the APAC region. The rollout of 5G will drive a lot of this expansion as edge computing technologies are essential in delivering the local high bandwidth, low latency network provision that 5G requires. Data generated by people and machines can reach 500 zettabytes by 2023, as estimated by Cisco Global Cloud Index And by 2025, 55% of IoT-created data will be stored, processed, analyzed at the edge of the network. Some IoT devices or applications might require very short response time and some might produce a large quantity of data which could be a heavy load for the networks. In this case, cloud computing is not efficient enough to support all these things.

When tasks forwarded to the users (Servers) are higher than the remaining resource of currently deployed hosts (Servers), the deployment of task events may fail. In the Cloud paradigm, data is processing in the large data centers, which are located far from end-users. The main goal of Edge computing is to reduce the latency between location of data processing and end-user. It allows us to improve overall performance as well.

Edge Computing performs data storage, processing, caching and computing under offloaded conditions, and in addition, it distributes requests and the cloud delivers the service. In IoT, edge computing is one of the essential frameworks and hence, the optimization of the edge furnishes the replica deployment to the users that improves the data availability and decreases the response time.

Edge computing allows processed data near to the source and at the same time reducing Internet bandwidth usage. Edge computing platform plays a very important role in the next scenarios:

✓ real-time services and applications

✓ short-term data

✓ edge decision-making.

These eliminate costs and provide the ability to process data without ever putting it into a public cloud. This is very important for sensitive data. In the Edge Computing Concept, there are a large number of geographically distributed Processing locations. The main goal is to decrease latency, processing time and improve QoS.

LOAD BALANCING

Nowadays, maintaining the load is the most challenging issue for a researcher in a cloud environment. In the IT field, Cloud computing area is a tremendous technology and its usage of computing resources that provide various services over the internet and its charges is based on usage of resources on the cloud. It provides many services to the users which are productive, reliable and low cost. So, users of cloud computing are increasing because every organization, government, and education department are moving toward the use of cloud services. Therefore, when many user requests for cloud resources arrive, we can use load balancing technique to fulfil the need of users. Load balancing process is useful to adjust the load on a node (virtual machine) using distributing the load on another underloaded node.

The main objective of load balancing is to make the virtual machine balanced which should be not underloaded or overloaded. We have modelled a Load balancing problem on a simple topology for edge computing and propose an algorithm, to solve it. Load balancing in Edge Cloud networks refers to the efficient distribution of the incoming workload across a group of processing compute nodes. When the load is low then one of the simple load balancing methods will suffice. In times of high load, the more complex methods are used to ensure an even distribution of requests.

LOAD BALANCING ALGORITHMS

Basically, there are 2 types of load balancing algorithm depending on their implementation method according to:

1. Static Load Balancing Algorithm: It does not depend on the current state of the system. Earlier before the incoming request it decides at which server the request will be executed.

- **Round-robin load balancing** is one of the simplest and most used load balancing algorithms. Client requests are distributed to application servers in rotation. For example, if you have three application servers: the first client request to the first application server in the list, the second client request to the second application server, the third client request to the third application server, the fourth to the first application server and so on. This load balancing algorithm does not take into consideration the characteristics of the application servers i.e., it assumes that all application servers are the same with the same availability, computing and load handling characteristics.

2. Dynamic Load Balancing Algorithm: The load balancer analyses the current load statistics at each available server and executes requests at appropriate server.

In this paper, we have implemented two load balancing algorithms for aforementioned scenarios. For static load balancing, we have used Round Robin and for complex/increased workloads we have implemented a least connection algorithm.

- **Least Connection load balancing** is a dynamic load balancing algorithm where client requests are distributed to the application server with the least number of active connections at the time the client request is received. In cases where application servers have similar specifications, an application server may be overloaded due to longer lived connections; this algorithm takes the active connection load into consideration.

Socket Programming:

When we want to connect two different machines, we do it with the help of socket programming, in this a socket connection is setup and by socket connection we mean that the machines have information about the IP address and TCP port of the other machine.

Socket socket = new Socket("127.0.0.1", 9999)

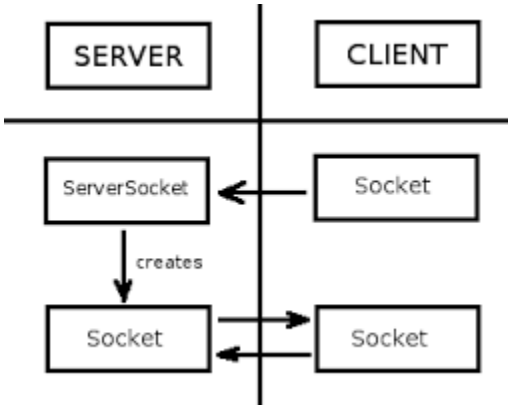
("127.0.0.1", 9999) -> Socket Address

There are 65536 ports

0-1023 well known ports

1024-49151 Registered Ports

49152-65535 Dynamic/Ephemeral Ports



Client Side:

For communicating with the server, a socket is created and request is sent to server and for data transmission streams are used to both input and output the data.

Server Side:

On the server side two sockets are needed, one for listening to client requests and second for communication. getOutputStream() and getInputStream() are used for getting and sending inputs and outputs.

REPORT OVERVIEW

Rest of the report consists of the Problems Statements on which the project will be working on. The multiple objectives of the project are also mentioned further. The detailed Literature overview is also mentioned. And the references of all the research papers and articles are mentioned.

The usual client-server architecture without an Edge server leads to a lot of delay in real world implementation due to multiple clients trying to access the same servers for their computational needs. In this paper by the involvement of edge computing the users can send a particular workload to a close by edge server first which can quickly compute the task and send the result back, as it is close to our user.

Without the involvement of a load balancer the servers can get overloaded by the tasks coming to them regularly which can lead to failure of a certain server. So, making a load balancer will resolve that problem and keep the tasks divided among those servers equally.

PROBLEM STATEMENTS

Problems

1. Overloading of servers-leading to server breakdowns

   Imagine a huge architecture involving clients from all over the world sending tasks to servers situated in a computational availability zone or region. Now in such a scenario there will be requests coming every second from all over the world resulting in huge traffic coming over to the servers. Such huge traffic may result in server breakdowns as some servers might overuse their capacity and get crashed.

2. Lot of Latency introduced in each task-Leading to lot of delay in response

   Imagine some clients are having some tasks where a loss of milliseconds in response is unacceptable, such as in manufacturing or financial services. So, if a client or user will send the task to the main server located far away geographically, it will lead to a lot of delay in response which can lead to a lot of loss for the client.

Solutions

1. Introducing Load Balancing Algorithms through a Load balancer in between Client and server

   To Overcome this situation of overloading of servers we use some techniques/algorithms to balance the load/tasks on the servers. These are load balancing algorithms. By introducing a load balancer which will implement load balancing algorithms we can distribute loads such that no server is compromised. Static load balancing algorithm will not consider the amount of work already been going on the server and will distribute them in a particular manner. Dynamic load balancing on the other hand will consider the amount of load already on a server and distribute to the one having least load.

2. Introducing Edge Servers with limited but high computational power near to users

   To overcome the latency problem edge servers are introduced. Every major cloud distribution is already using Edge server technology to decrease the delay in fast computation needs. Introducing an Edge server near to the user will decrease the delay for some particular tasks which do not require very high computation power.

LITERATURE REVIEW

[1] In this paper, the aim is to provide a systematic **comparative study of existing load balancing algorithms in cloud computing.** It has focussed on analyzing various static and dynamic load balancing algorithms with their merits and demerits. It has also addressed the importance of load balancing in the cloud, highlighting the **need for dynamic load distribution** among the nodes to achieve maximum resource utilization and high user satisfaction ratio.

[2] In this research paper, the various load balancing methods in edge computing are elucidated. **A detailed survey of current state-of-the-art load balancing techniques at edge** including optimization-based, traffic load-based, heterogeneous, multi-access, joint-load, heuristic, security, allocation, distributed load and the dynamic load-based techniques are explained. It reviewed that dynamic load-based techniques are mostly used in current research whereas CloudSim is a very widely used toolset in the existing research to perform output simulation.

[3] This paper focuses on the existing static load balancing algorithms available for distributed systems. It presents the **comparative study of three static load balancing algorithms: Round Robin, Randomized and Threshold using CloudSim Framework.** It also shows that the static load balancing algorithms are more stable.

[4] Karan D. Patel and Tosal M.Bhalodia proposed a **load balancing algorithm by combining two algorithms for balancing the workload over the cloud system.** They used **modified honey bee behavior inspired algorithm for priority based tasks** and enhanced weighted **round-robin algorithm used for non-priority based tasks.** This proposed work is developed for achieving better resource utilization, minimum completion time and improving the performance of systems in a cloud computing environment.

METHODOLOGY

1. **Research and analysis**
   We will gain information about Edge Computing, Socket Programming, Client Server Architecture and Load Balancing algorithms
2. **Design**

   Here we will create pseudo code and understand the data flow of the code we will implement.
3. **Implementation**
   In this phase we will implement communication between nodes and send computation workloads to edge and server nodes
4. **Testing**
   Test the algorithms against different sizes of workloads and different parameters and finally compare their complexities.

SYSTEM REQUIREMENTS

Hardware Interface:

1. Minimum RAM requirement for proper functioning is 8 GB.
2. Required input as well as output devices.

Software Interface:

1. JDK 15
2. Eclipse IDE
3. Virtual Box
4. Sever ISO image of Linux

OBJECTIVE

Objective 1

Studying and grasping the background knowledge related to the Edge Computing, Socket Programming, Client Server Architecture and Load Balancing algorithms:

1. Round Robin Load Balancing
2. Least Connection Load Balancing
3. Resource Based Load Balancing
4. Previous works related to Load Balancing in Client Server Architecture

Objective 2

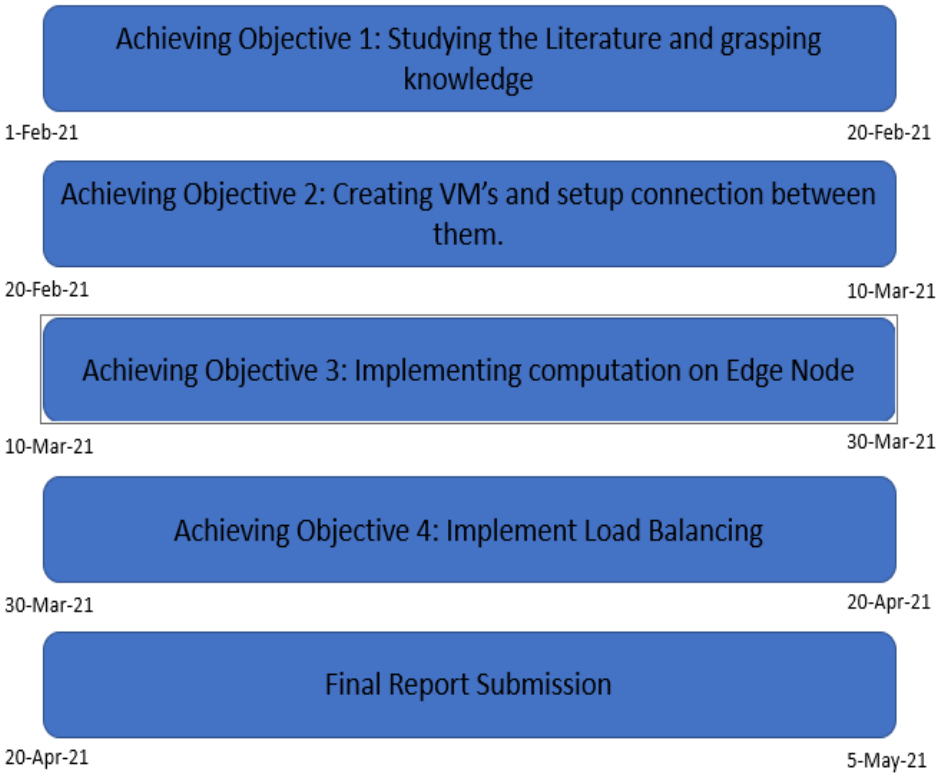
Creating all the required Virtual Machines and Setup connection between them.

Objective 3

Implementing computation on the edge node.

Objective 4

Implementing communication between Edge and Server node and performing tasks on Servers using Load Balancing algorithms



REFERENCES

- [1] T. Deepa, Dr. Dhanaraj Cheelu, "A Comparative Study of Static and Dynamic Load Balancing Algorithms in Cloud Computing" in International Conference on Energy, Communication, Data Analytics and Soft Computing, pp, August 2017
  LINK:https://www.researchgate.net/publication/325981627_A_comparative_study_of_static_and_dynamic_load_balancing_algorithms_in_cloud_computing

- [2] Harikrishna Pydi, Ganesh Neelakanta Iyer, "Analytical Review and Study on Load Balancing in Edge Computing Platform" in Fourth International Conference on Computing Methodologies and Communication, pp, 11-13 March 2020
  LINK : https://ieeexplore.ieee.org/abstract/document/9076553

- [3] Supriya P Belkar, Vidya Handur, "Comparative Study of Static Load Balancing Algorithms in Distributed System Using CloudSim" in International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST), Volume 02, Issue 10 (October 2013)
  LINK:https://www.ijert.org/comparative-study-of-load-balancing-algorithms-in-cloud-computing-environment

- [4] Karan D. Patel, Tosal M.Bhalodia, "An Efficient Dynamic Load Balancing Algorithm for Virtual Machine in Cloud Computing" in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), pp, 15-17 May 2019
  LINK: https://ieeexplore.ieee.org/document/9065292

- [5]Pramod Kumar, Dr. Mahesh Bundele, Mr. Devendra Somwansi, "An Adaptive Approach for Load Balancing in Cloud Computing Using MTB Load Balancing" in 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering, 22-25 November 2018
  LINK: https://ieeexplore.ieee.org/document/8710433