Chapter 1

INTRODUCTION

1.1 Overview of Cloud Computing

Cloud Computing is a variety, which is designed for the convenient of the users, on-demanding network access towards the shared computing resources such as Storage, Servers and Network. The agenda of cloud computing is that, it is done with minimal effort to provide services to the public. The cloud infrastructure is based on the basis of Pay-as-Per Use; this means users should pay for the services which have been used. There are many free tire services provided for the users.

The first movers in the cloud are Amazon, Google, and Microsoft.

1) Amazon- Amazon is the head company to provide cloud facilities to the workers. Some of the cloud services provided by Amazon are:

   - EC2- Elastic Compute Cloud, it provides virtual machines with extra CPU cycles to the establishments.
   - S3- Simple Storage Service, this storage provides to stock substances up to 5GB size in Amazon‟s virtual storage service.
   - SQS- Simple Queue Service, it uses message-passing API which allows the machines to communicate each other.

2) Google- As in Amazon there are root privileges, where we can write a file in our own directory, where as in Google App Appliance we can't carve a file into our own index. Hence Google has detached the feature file write as it is a security measure. The Google database is used to store the data.

3) Microsoft- The Microsoft created Microsoft Azure which is a cloud computing platform and infrastructure. This is created for the purpose of

managing, deploying and building applications and services all around the world through the network of Microsoft-managed datacentres. The Microsoft Azure platform includes:

- Windows Azure
- Microsoft SQL Services
- Live Services

AWS

Amazon Web Service is a collection of cloud computing which is offered by Amazon.com.

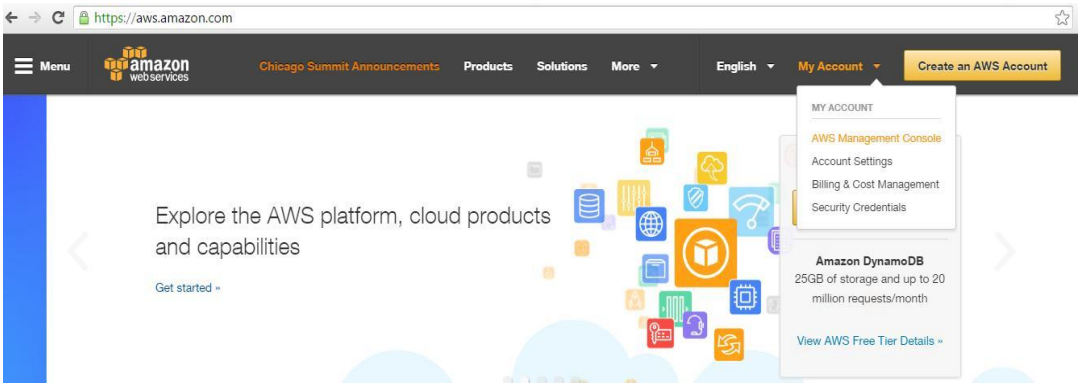

fig 1.1 login to the AWS Management Console

Fig 1.2 login page of AWS account.

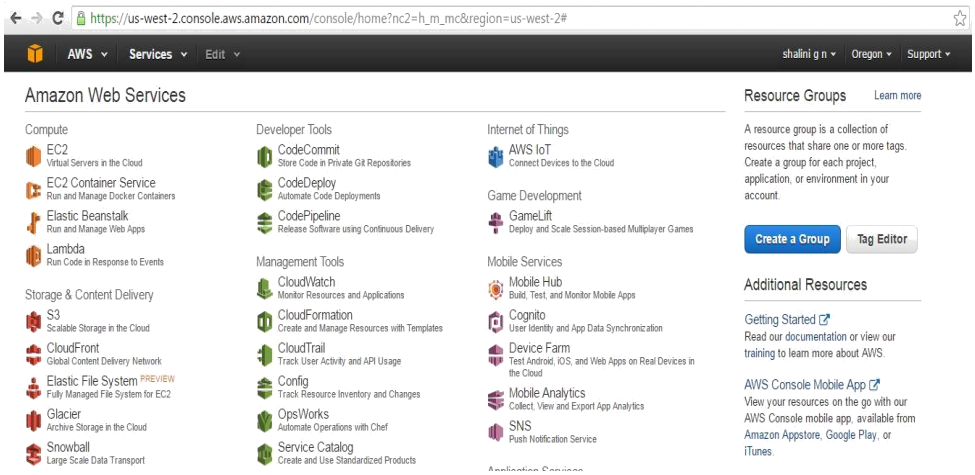

Fig 1.3 some of the services of Amazon

1.1.1 Basic Concepts

There are few cloud computing components namely- clients, datacentres and distributed servers. Each of the components has its specific role.

- Clients: The end operators communicate through the devices to accomplish their material going on the cloud. Clients are classified into three groups namely.
    1. Thick: It is a client which is a normal computer which uses a web browser such as Firefox or Internet Explorer to get connected to the cloud.
    2. Thin: It is a type of client, that is computers which do not ensure the need for inner hard drives, but it lets the server do all the work and then display the data.
    3. Mobile: This one is the type of customers which includes PDA‟s that is smartphones such as iPhone, Mobile smartphone etc..
- Datacentre: It is a collection of servers where the applications are housed and it is accessed via internet. As IT industries are growing day by day, it is a trend of having virtualizing servers. Where the software‟s are installed in multiple instances of the virtual servers that are being used. Hence by this, number of virtual servers can be run on single physical server.

- Scattered Servers: Here the servers need not exist in one same location they are distributed around the globe at different locations. But through cloud, these servers can be accessed easily as they act that they are placed in same location.

1.2 Introduction to Project

As in geo-distributed datacentre system, exist a basic demand to handover majority size of files from one datacentre to added datacentres and the efficiency of the time taken to transfer bulk data to the datacentres. Hence, the main challenge was how to handle and organize the majority of documents to allocate at different urgency level and to use the existing bandwidth of inter-datacentres.

To handle a raised challenge, the controller was developed which is known as SDN i.e. Software Defined Networking which decouples the device plane from the data paths. The other main challenge is how to synchronize the multiple datacentres and transfer the bulk data to the datacentres. These datacentres are selected dynamically to which the data should be sent which is taken care by the main controller.

Synchronization is one of the process or technique used between several systems. Synchronization in computing means, it is a process of constituting two or several data storage devices having same data provided in one data storage at a given time. Datacentre is centralized repository for storage, for managing data which is well ordered by a particular body.

We need multiple datacentres because datacentre performance is very poor since to improve the performance of datacentres the following reasons are:

- Geolocality permits tasks to be progressed nearer to the customer.
- The speediness of the light is considered to limit exactly how reckless data can be transported and becomes more important while working through the biosphere.
- Nearer is well and you can only be closer if your information is near to the user, which wants operating in several datacentres.
- Using multiple routers helps to slow the traffic.

1.3 Objective

The main objective of the project is to achieve proper synchronization between the inter-datacentres. As the main controller schedules the jobs that has to be submitted to the data servers according to the priority of the jobs given. The controller which is called as Software Defined Networking (SDN), it provides flexibility which enables dynamic allocation of jobs to the servers that means it is not known to the clients to which data server the job is admitted, this is done dynamically. Hence it utilizes the available bandwidth.

In the existing system, the users were aimed to bring the services close to the public, with regard to network failures, power consumption etc. As nowadays there are many cloud providers such as Amazon, Google and Microsoft which are already invested in building huge measure of datacentres round the sphere to afford their services. But the demand is for the transfer of bulk data around the world from one datacentre to another.

In the proposed system, the demand for the transfer of large volume of data from one datacentre to another requires the synchronization of inter-datacentres. As it is synchronized, the clients need not require waiting for the particular server to respond to the client, the available server will immediately respond to the client as the data is synched with other datacentres by using available bandwidth. Here the jobs which are let them pending as the deadline for the particular job is timed out, that job is not rejected but it will be resubmitted by utilizing the available bandwidth.

1.4 Problem Statement of the Proposed System

When their exist a demand for inter-datacenter file transmissions in huge dimensions. Then the challenge that is faced is by what method to plan the substance of file transfers at different urgency level with available inter-datacenter bandwidth.

1.5 Outline of the Project

The report of the mission contains following divisions:

Chapter 1: Overview

This chapter involve the introduction to the technology and the introduction to the project.

Chapter 2: Literature Survey

This chapter gives the analysis of the papers referred and the details of those papers.

Chapter 3: System Analysis

This section gives the idea of the necessities for the recommended system.

Chapter 4: System Design

This chapter gives the detailed information of the design of the proposed system, which consists of flow chart, use case, and sequence diagrams.

Chapter 5: Implementation

This chapter will contain the coding of the project and the details of the modules which are being used.

Chapter 6: Testing

This chapter will consist of different testing‟s done to the proposed system such as system testing, unit testing and integration testing.

Chapter 7: Snapshots

In this chapter it gives the overview of the working modules of the project with snapshots.

Conclusion

It gives the description of the project and the outline of the framework.

Future Enhancement

It gives ideas for the future implementation for the proposed system.

Chapter 2

LITERATURE SERVEY

2.1 Outline of Datacentres in Cloud

Datacentre is a consolidated repository which is meant for the storage purpose, management purpose and the distribution of data.

Years ago, these datacentres were only used in the private place which was hosted in the organizations where the individual servers were used to run a single application. Later the organizations started to manage internal datacentres to manage the increased level of business. Since they required third party as datacentres and the cloud providers which are the better ones to maintain and update the servers.

The datacentres deployed the servers and the applications which is used for the business purpose on the cloud.

2.2 Remunerations of Cloud Computing

Some of the benefits of cloud computing are:

- Flexible Cost: Some of the clouds are available for free tire which is specially meant for students and there are few clouds which are available for fewer Costs than the traditional methods.
- Always availability: The clouds are always available for the users to utilize them.
- Unlimited Storage: The cloud provides unlimited storage for the public to store their data on to the cloud.
- Backup and Recovery: once the data is stored on to the cloud, that data is kept backup and this is recovered easily than in the physical device.
- Easy Accessible: once we get registered to the cloud, we can easily access the data from anywhere around the world if the internet is available.

2.3 Analysis of Survey Papers

2.3.1 Review of Paper [1]

As the web based administration suppliers are a persistent operation and dependably accessibility to people in general shrink it may be arranged or unplanned.so here they advocate a helpful, connection mindful way to deal with datacentre movement crosswise over WANs to overcome with blackouts. It uses few of the methods such as:

- Live datacentre Migration across WANs:

  It concentrates on the nearby repetition systems which are not adequate. Uncommonly while giving administration accessibility when datacentre gets to be distracted. So the fundamental methodology is to relocate administrations between datacentres over the WAN. Some of the services which are required are:

  - Live virtual server migration
  - Networking requirements
  - Storage replication requirements

- There are some of the WAN migration scenarios they are:
  - Maintenance Outages
  - Unplanned Outages

2.3.2 Review of Paper [2]

As now when we see the growth of IOT that is Internet Of Things cloud has become biggest industries in the world. There are many organizations which are providing cloud such as Amazon, Google, Microsoft etc. when we are moving towards the new technologies we have to look towards the efficient approach for the cost optimization for the movement of big data. There are couple of methods indicated in this paper for the development of enormous information in cloud specifically:

- Migration Based Approaches:

The migration approach is considered as migration of big data from one site to another. It includes the complete transfer of data from one site to another site with minimum cost.

Some of the approaches for the minimum cost to migrate data around the world are mentioned as follows:

- Cloudward Bound: this is making arrangements for gainful movement of the applications to the cloud.
- PANDORA: this is planned for the budget constraint for the transfer of bulk data around the globe.
- Online cost minimizing approach: this approach tell to decide which the best site for data aggregation.
- SSH model: this is for the efficient transportation movement of big data.

➢ Data Placement-based Approach:

It tries to minimize the data movement instead of data movement cost.

Some of the methods suggested are:

- Data position system in light of hereditary calculation for logical work processes.
- Scaling online networking application into Geo-conveyed mists.
- Volley this is for mechanized information situation for Geo-conveyed cloud administrations.
- Cost minimization for huge information preparing in geo-appropriated server farms.

➢ Performance Analysis and Evaluation.

2.3.3 Review of Paper [3]

Cloud datacentre frameworks traverse different geographic areas which are regular nowadays, which is intending to get benefits near the clients, which ought to develop lower power cost, empower administrations for the disappointments.

The interest in geo-disseminated datacentre framework to exchange mass volumes of information from one datacentre to different datacentres. Presently the test in this paper is the means by which to plan progressively to exchange the information for between datacentre at various criticalness levels, and distinctive information exchange completing due dates at the accessible transmission capacity.

This paper proposes a novel enhancement model for element, very effective planning of mass information moves increase due in geo-dispersed datacentre framework. It will first detail the mass information move issue into a novel , that is ideal lump directing issue which boosts the total utility addition because of convenient exchange fulfillments before the predetermined due dates.

Besides they utilized three element calculations to take care of the ideal lump steering issue to be specific:

- Bandwidth-Reserving algorithm
- Dynamically Adjusting algorithm
- Demand Friendly algorithm

2.3.4 Review of Paper [4]

In distributed computing these days the pattern is gathering of geo-disseminated cloud administrations by expanding various datacentres at various geological areas, where it can furnish cloud stage with bigger limits.

In supporting online networking applications, challenges emerges, for example, by what method ought to the social networking substance be put away and moved crosswise over various cloud locales and review solicitations be appropriated, such that the operational expenses are minimized.

Thus the genuine test in this paper is to plan online calculation that can make utilization of cloud assets which can hold up element substance.

This paper proposes couple of calculations for dynamic and ideal scaling of online networking applications in geo-disseminated cloud:

- Enables pre-emptive content movement by forecasting future demand.
- To serve the predicted demand, such that it decides on one-shot optimal content migration.
- Presentation of the algorithm is gauged via large-scale trials under dynamic realistic settings on Amazon EC2.

Chapter 3

SYSTEM ANALYSIS

3.1 Existing System

As the datacentres are connected inside the network, the existing system used FIFO flow scheduling and TCP congestion control used for data flow transfer. These not aware of flow deadlines, that are when the data flow are submitted they will be assigned a specific time to complete within that time, if it is not completed, the data flow will be rejected once timed out. In the existing system, while the transfer of bulk data, which flows among the datacentres around the globe it uses end-to-end congestion control.

Some of the services which include in cloud to transfer bulk data around the globe, they are:

- Task affirmation control: when the information exchange errand is admitted to the controller, it plans the particular due date to finish the undertaking, if the exchange assignment is not finished inside the predefined time as indicated by the system accessibility then the undertaking will be rejected.
- Data directing: In information steering the transmission ways of the acknowledged errand from source to destination must be chosen through numerous middle of the road datacentres.

3.2 Limitations of Existing System

- Here, in network the data flows around the globe, it uses end-to-end congestion control where the data flow from source to destination.
- It is not aware of deadline flow, where it leads to the rejection of the data that is being flowed in between without completing.
- There is no synchronization of datacentres from one datacentre to other datacentres.

- Due to no synchronization between datacentres, time efficiency is delayed that is, the client should wait for the particular server to respond, if that server is busy it delays in responding to the client.

- The system runs in a time-slotted fashion, where there is a fixed time slot given to each task that is to be transmitted and should complete within the given time slot.

3.3 Proposed System

The main goal is to achieve the setting up majority of file transmissions at different urgency levels in demand to consume offered bandwidth. Here the controller is designed in such a way which controls the multiple datacentres, this controller is known as Software Defined Networking (SDN) which disassociate the control plane from the data paths, which enables overall optimization of information direction-finding in network. When the client does few operations such as uploading, delete or modify to the servers, the server will admit the task to controller. The controller will schedule dynamically to which datacentre the task should be admitted. The controller uses FIFO flow scheduling algorithm and the Priority algorithm for the scheduling of task.

Once the controller decides to which datacentre the task has to be sent, which is done dynamically. The controller will take the task which is received first from the server will be sent first to the datacentre, if the server is busy or the bandwidth is not compatible with the present task which is sent from the controller, the server will send an acknowledgement to the controller as the task which is sent cannot be completed due to the bandwidth, then the controller will see the priority of the jobs and then it will send back to the data servers.

After the task is sent to the server, the server should now get synchronized with other datacentres, this is because as the clients will be requesting to the servers to retrieve the data from the datacentres, if the server is busy and unable to respond the client in time, the task will be delayed, hence when the multiple datacentres are deployed around the world, we should have proper synchronization between the datacentres. This is done because there are many users who are using Facebook,

twitter and many other social medias simultaneously, if there are multiple servers kept and in each server the copies must be kept, so that whichever server is free will respond to the user spontaneously without any delay.

3.4 Advantages of Proposed System

- The proposed system is dynamic, where the controller admits the task to the dataservers randomly.
- The main goal in the proposed system is to achieve maximum data transfers in minimum time without any deadlock occurrence.
- To achieve proper synchronization between multiple datacentres which are deployed in geo-distributed datacentres, this is done because the users can have better performance and the time efficiency. The tasks are not delayed.
- In the proposed system, the scheduling of bulk data transfers between the datacentres is done efficiently.
- As the task is not completed in the given time slot, the task is not rejected rather than it will be resubmitted whenever the bandwidth is available.
- As the data is kept multiple copies in different datacentres, the clients can retrieve the data from any of the server without any delay, if in case one of the server is meet with failure, the copy of data will be available in other servers.

3.4 System Requirements

General Description:

This segment gives the depiction of the general variables about the item and its prerequisites. This area additionally gives nitty gritty data about the devices and technologies used in the proposed system. It makes user to understand in easier way.

Product Perspective:

The proposed system can be executed in a local host and it can also be executed via network to show how synchronization of datacentres happens. If we want to show in the local host the IP address of the system should be specified as local host. If we want to show through the network connectivity we should have LAN connection where many

systems are connected. In the network connectivity when many systems are connected we should know the IP address of each system to get connected. We should specify the IP address of the system where the particular module so that it get connected to that system, rather than specifying local host. In the proposed system we can see the synchronization of the data centres via network connectivity.

In the proposed system, it is also executed in real time by using cloud service provider. The amazon EC2 provided free tire for the cloud instances. We should first launch the number of instances needed. Later in Remote Desktop Connection we get connected by giving the public IP of the instance created. Once the connection is done the cloud desktop is shown. Later we can install the required software‟s to run the project in the cloud.

3.6 Specific Requirements

3.6.1 Software Requirements

Operating System : Windows7/8/XP(64-bit or 32-bit)

Front End : Java

Software : Eclipse

3.6.2 Hardware Requirements

Processor : Pentium 3

Speed : 3.5 GHz

RAM : 1 GB (minimum)

Hard Disk : 40 GB

3.7 Resource Requirements

3.7.1 About Eclipse

Eclipse is as platform that is blueprint for construction of Integrated Development Environments (IDE‟s). Eclipse platform was used to build full-featured Java Development environment.

Eclipse is just a foundation for building development environments such as

building arbitrary tools and applications. It is open source software which consist more than 150 projects. The eclipse IDE can be extended with additional software components, hence eclipse call these components as plug-ins. This plug-ins can be grouped into features.

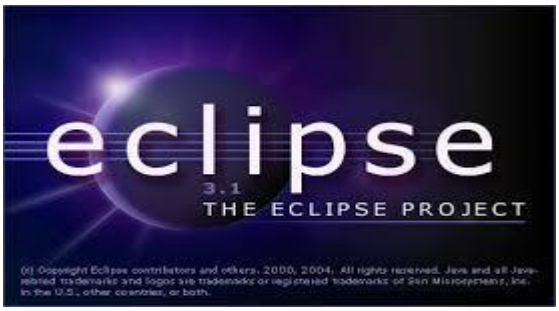

Fig 3.1 shows the eclipse software

The eclipse IDE is also available on different IDE languages such as c, c++, php, python and several languages. codebase was initially invented by IBM VisualAge. The Software Development Kit (SDK), that includes java development tools which is meant for java developers.

3.7.2 About Cloud Instances

The first movers of cloud were Amazon, the Amazon provided free tires to the users. An EC2 instance is a computer-generated server for running applications on the Amazon Web Services infrastructure. An EC2 provides number of virtual machines to run the applications on cloud. Once launching instances, we have to start running the instance. This done in the instance state, were there are many options such as to terminate the instance, to start an instance, and to stop an instance. After the instance is started running, there will be provided the public IP address which as to be copied and should put on Remote Desktop Connection and then get connected to the cloud instance.

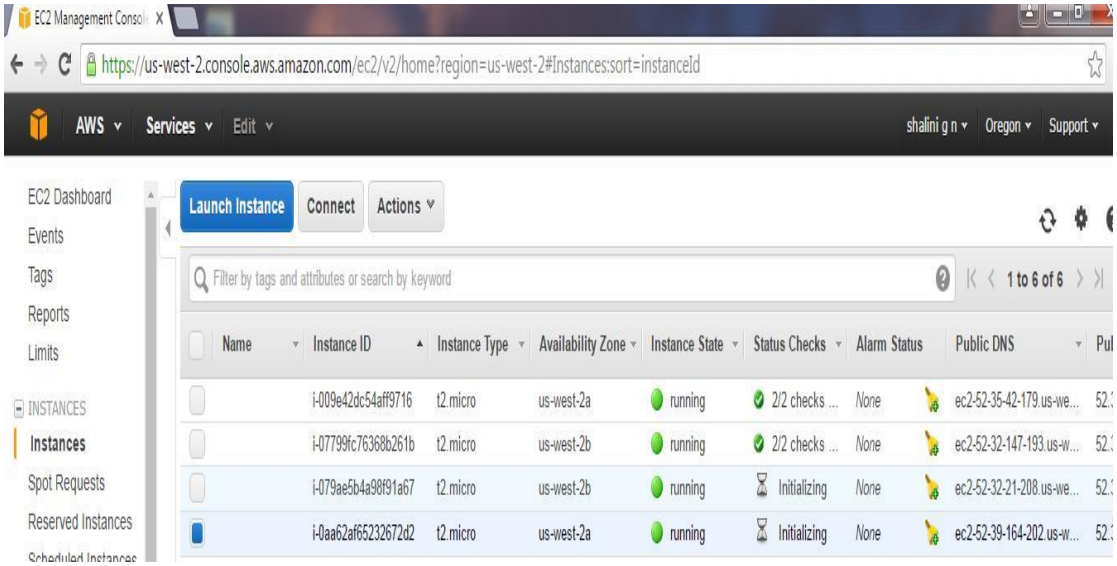

Fig 3.2 shows how to launch the instances

After launching an instance it should be connected to the RDC which is known as Remote Desktop Connection. Here we have to give the IP address of the launched instance and to access the files of our pc into the instance we have to enable the drives option in RDC. By doing this we can access the files of our pc into the cloud. Later we should install the required software's into the cloud to run our project.

Fig 3.3 shows how to connect to instance through RDC

3.7.3 Core Tools and Technologies Used

In PC programing, Eclipse is a synchronized advancement environment (IDE). It encloses a base workstation and an extensible module outline for redoing the earth composed for the most part in java, Eclipse can be used to create applications. By method for different modules. Obscuration can likewise be utilized to create bundles for the product Mathematica. Advancement situations incorporate the overshadowing Java Development Tools (JDT) for java and scala, shroud CDT for c/c++ and shroud PDT for php.

Swing it is an essential Java GUI gadget toolbox. It is an API for giving a GUI that is Graphical User Border for java programs. Swing was produced to give an arrangement of GUI segments than the AWT that is Abstract Window Toolkit. The swing gives a look and feel those backings a pluggable look and feel that permits applications to observe and feel random to the fundamental stage. The swings are more effective and adaptable parts than the AWT.

The AWT gives numerous classes to developers to utilize. It is association between the application subterranean insect the GUI. The AWT conceals you from the basic subtle elements of the GUI application will keep running on and consequently is at abnormal state of reflection. It takes the most minimized shared variable way to deal with hold movability. There is no gliding toolbars.

The java bundle can be utilized as a part of any java program by importing java.awt.* by means of the import watchword.

Chapter 4

SYSTEM DESIGN

Co-ordination plan is a process, which describes the overall architecture of the system, logical plan of the system and the physical intention of the system.

The architecture design describes the structure, view and the behaviour of the entire system. The architecture design makes the user to understand the process of the entire system how actually it works and the modules of the system.

The logical design describes the data flow, flow of the system and the entity relationship of the modules. The data flow diagram is a graphical representation of the data flow in the system. The flow chart of the system shows how the control of the system flows. The entity relationship gives the idea of how the modules are related to each other.

The physical design includes the inputs and outputs required for the processes of the system. The physical design gives overall ideal of how the system is working such as having information about the input requirements, output requirements, processing requirements etc.

4.1 System Architecture

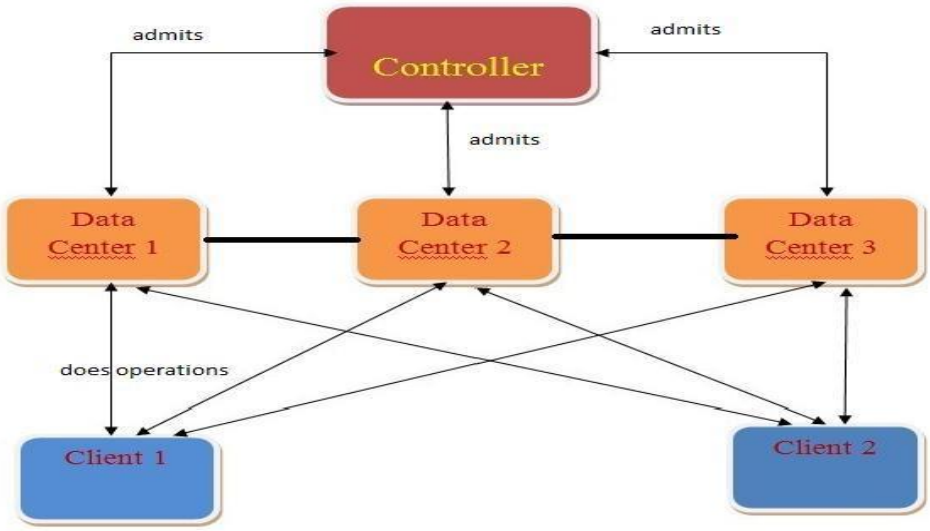

Fig 4.1 The system architecture.

The system framework of the suggested system consists of three modules namely:

- Controller
- Datacentres
- Clients

In the proposed system, there is a controller which is known as SDN that is Software Defined Networking which schedules the task and then admits to the servers. There can be „n‟ number of clients and „m‟ number of datacentres. As in the proposed system we considered three datacentres and two clients.

When the clients perform some operations such as set, get and delete, the clients need to upload those tasks to the server. Then the server will admit that task to the controller, the controller maintains the queue where the tasks are kept. Once the controller starts to schedule the tasks to the datacentres, the controller will use the FIFO algorithm, Priority algorithm. The controller then admits the tasks to the datacentres.

The controller admits the task to the datacentre dynamically, that is it does not know to which particular datacentre it is sending the task. Once the client receives the acknowledgement from the controller as to which datacentre the task has being admitted with the IP address of that server.

The task which is received from the controller to one of the server, now the server should get synched with other servers that is multiple datacentres. This is the main agenda of the proposed system. This is done because there should be multiple copies of the task in multiple datacentres if the client request for a task if that particular server is busy to respond the client, then the time is not efficiently used, so if there are copies of same data in multiple datacentres then any of the server which is free can respond to the client.

4.2 Flow Chart

Flow chart is meant to describe the flow control of the proposed system. This graphical representation makes the readers easy to understand the flow of working system.

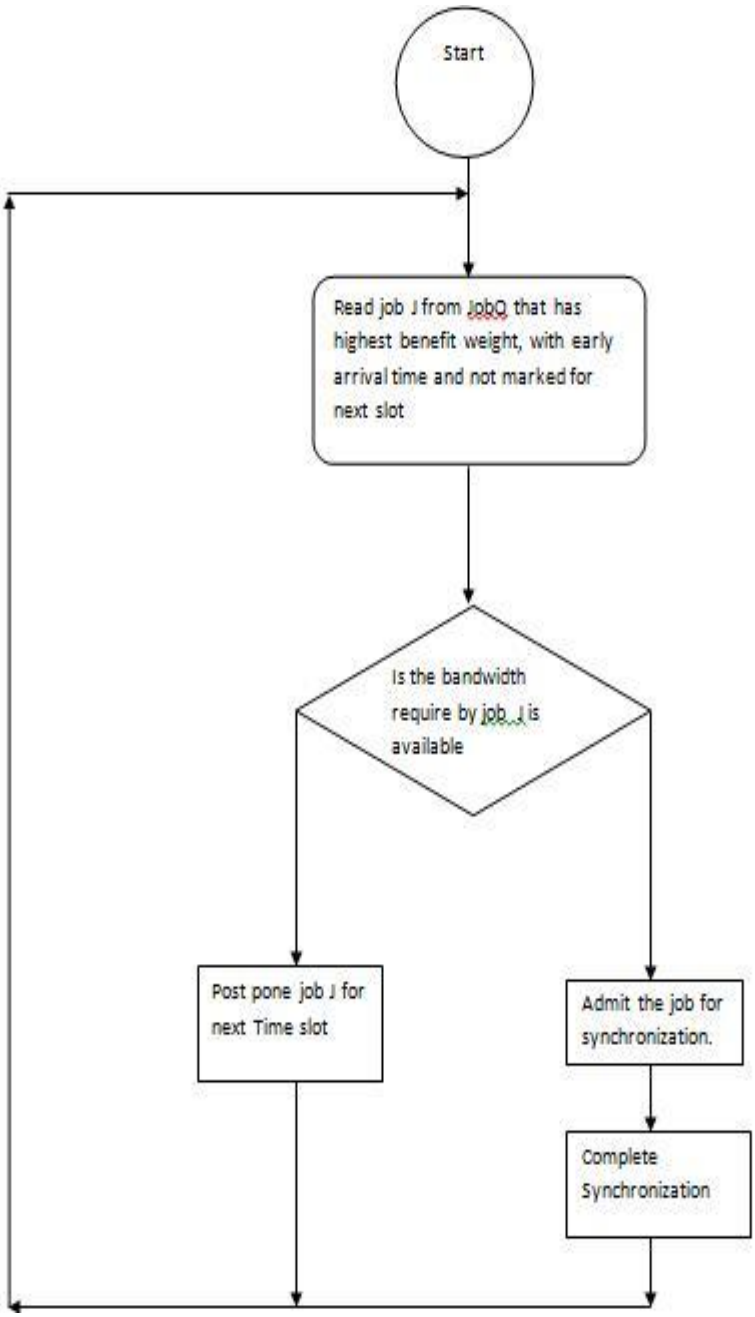

Fig 4.2 shows the flow control of the system.

The fig 4.2 represents the flow of the proposed system, once the client starts performing the task or operations, the server will first read the job or task which is arrived. The server then admits the job to the controller, now the controller will first read the job which has the highest weight with early arriving time which is not marked for next slot. Then it will check for the available bandwidth of the job. If it is yes, then it will admit the job for synchronization, and it completes the synchronization. If it is no, then it will postpone the job for next time slot and again the procedure will start from the first.

4.3 Sequence Diagram

Sequence diagram is a graphical representation which describes how the system works in the sequence with the detailed description.

In fig 4.3 shows the graphical representation of the proposed system, in the proposed system we considered two clients and two datacentre servers and a controller. The clients can simultaneously upload the data to the servers. The data is uploaded dynamically to the servers. Hence the clients don‟t have the knowledge to which particular server the data is uploaded. Then the server will admit those data or jobs to the controller, the controller will admit those jobs to the queue. Later the controller will read the job which has the highest benefit weight and the job which is arrived first and which is not marked for the next time slot.

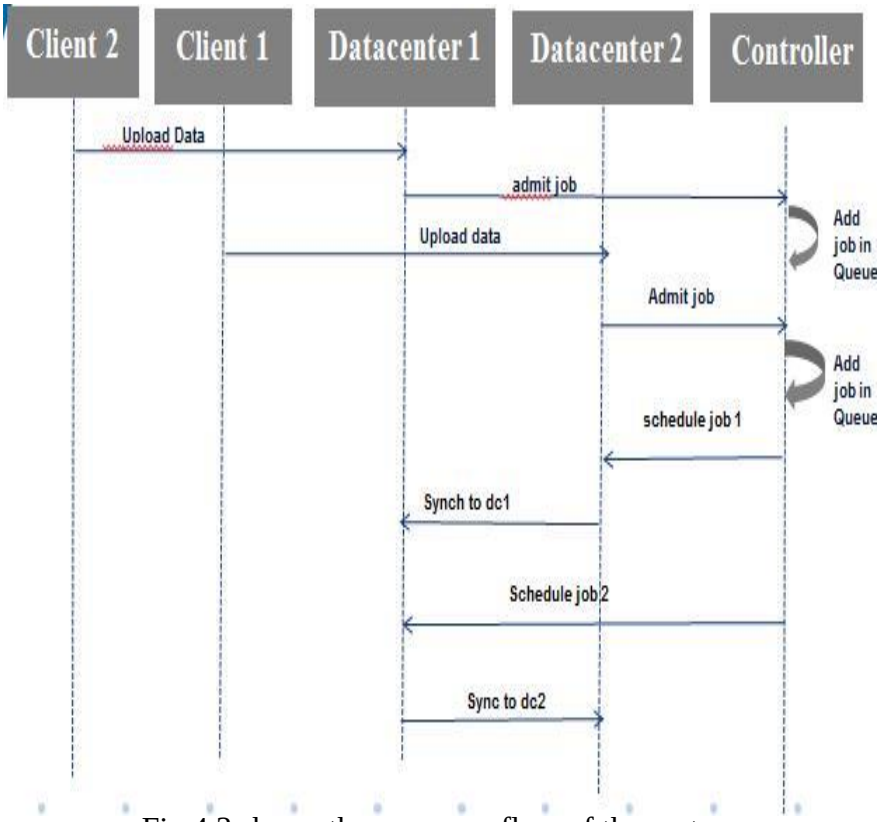

Fig 4.3 shows the sequence flow of the system.

Once the controller read the job which is in the queue, the controller will schedule the first job to the server which is done dynamically, that is there is no particular specification to which server the controller scheduled the job. The client receives the acknowledgement from the controller as to which server the job is admitted. Once the job is admitted to one of the server, now the server should synchronize with other servers.

4.4 Use Case Diagrams

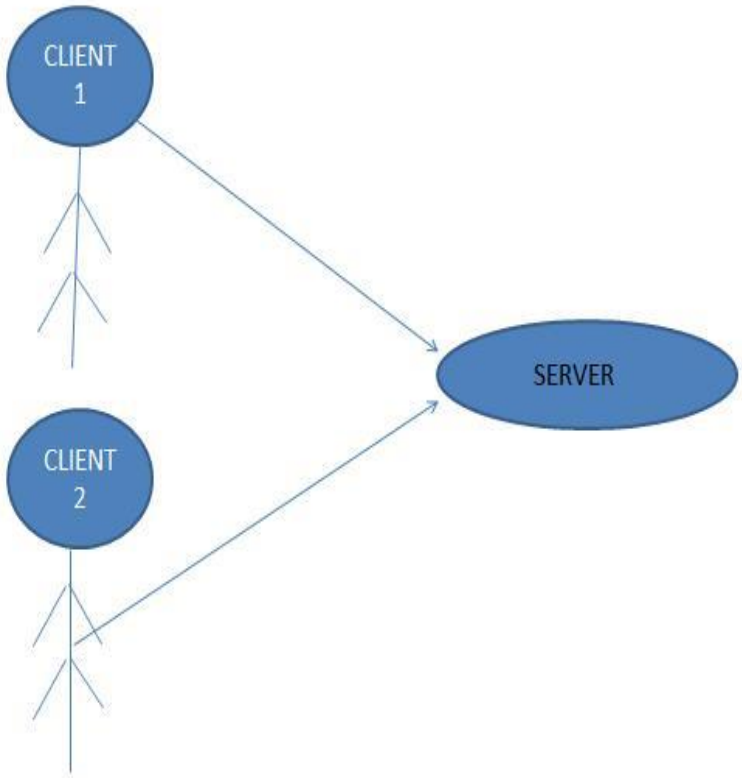

Fig 4.4 use case diagram with two clients and one server.

Here in fig 4.4 it represents the use case diagram with two clients and one server. When the two clients start working simultaneously that is doing operations at the same time, there is only one server which is accepting the requests from the clients. As in globe there are n numbers of users who are accessing the same server at the same time. When the server is busy which is responding to one of the users at the same time the other client is requesting for the service to the server, but the server will be unable to respond to the request because the server is busy. Then the time is not efficiently utilized and doesn‟t fulfill the urgency level.

Fig 4.5 Use Case Diagram of Client, Server and Controller

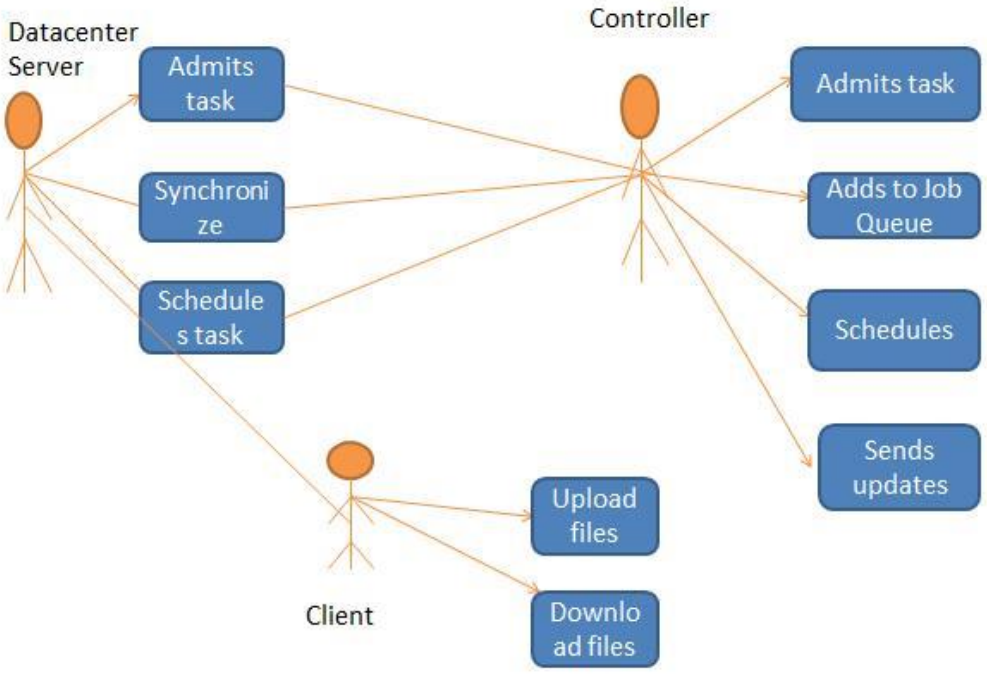

Fig 4.5 Use Case diagram of server, controller and client

Chapter 5

IMPLEMENTATION

5.1 Introduction

Implementation is the main part of the project where the unproven design of the proposed system is turned out into a working system. The implementation stage is considered one of the critical stage as the proposed system modules should work successfully.

The implementation part will include the coding of the modules which involves planning, future enhancement of the existing system, designing of the proposed system modules. To achieve a successful proposed system the user should have confidence and the system should work effectively.

5.2 Proposed System Modules

In the proposed system it mainly consists of three modules namely:

- Controller
- Datacentre Servers
- Clients

5.2.1 Controller module

The proposed system mainly consists of controller which is known as Software Defined Networking (SDN). This module will decouple the control planes from data paths. The controller is designed in such a way that, it controls the servers and act as a main controller.

As when the tasks or jobs are admitted to the controller from the servers, the controller maintains a queue where all the jobs are kept in a queue. The controller checks the earliest job arrived and the highest benefit weight of the job and that job which is not allotted for next time slot will be scheduled to the servers.

The controller does few of the operations such as:

- Job Scheduling: This is the operation performed by the controller that is the controller is responsible for scheduling the jobs to the datacentre servers. Once the controller admits the job back to the server which is done dynamically with the scheduling algorithms. Here the first come first server algorithm is used as with the priority of the jobs given.
- Job Queue: The controller maintains a queue, where it will keep all the arriving jobs from different servers. This is queue is maintained because the controller should schedule the jobs according to the priority and the available bandwidth. It checks these conditions and then schedules the task to the servers. Hence it maintains the job queue.
- SynchRequestListener: This is a operation performed by the controller. It will request the datacentre Servers to get synchronized with other servers.

5.2.2 Datacentre Servers Module

This is a module where there are three datacentre servers in the proposed system. When the client uploads the files or requesting the files from the server, if there is only one server and much number of clients who are performing at the same time, then the server can respond to only one client. Since the proposed system enhances the concept of synchronization with other servers so that any of the server which is free can respond to the client by utilizing efficient time.

The datacentre server performs few set of operations such as:

- SynchwithMe: This is the operation performed by one of the server which has received the job or task from the controller. Once the task is received from the controller, the controller will request the server to synch with that particular server.

- SynchWithOthers: This is the operation performed by the server, one of the server receives the scheduled task from the controller, that particular server will get synched that means all the data which is present in that server will be copied to other servers. This is known as SynchWithOthers. If this is done the clients can retrieve the data from any of the servers.
- FileServer: This operation is performed by the server as the server will receive the files from the client, where the files are having the key value pairs where the file server will receive the key as well as the value which is sent by the client.

5.2.3 Client Module

There can be any number of clients, but in the proposed system we specified only two clients. These clients are connected randomly to the servers without any prior knowledge. When these clients upload files to the servers, it is randomly admitted to the servers.

The client module performs few set of operations such as:

- Get operation: The get operations performs as, the client should had given the key and value. These updates are sent to the servers that the get operation is done, means if we give the key, it will return the value which is given for the particular key.
- Set Operation: The set operation performs as, the client will give the key and value and then it will set those key value pairs. This is updated in the server table as the put command is ok.
- Delete Operation: This operation is performed as, when the client sets the key value pairs, this updated is sent to all the servers. If in case we want to delete particular key value pair this operation is performed. If any key or value is deleted by the client, this is updated with all the servers.

5.3 Algorithm

Algorithm is a step-by-step procedure where the operations are carried out. It is a graphical representation of line by line through which it is written in a programing language. The algorithms are used to solve the problem by one or the other way. They can also solve simple mathematical calculation, it also makes simpler for the users to perform complex functions through an algorithm.

Step 1: Start

Step 2: Ready to read the job j, which is having highest weight and not marked for next time slot

Step 3: It will check for the availability of bandwidth required by the job If the bandwidth is available

Then admit the job j for the synchronization

Else

Post pone the job for the next time slot

Then start the job to read again

Step 4: Stop

Fig 5.1 Algorithm for the proposed System

5.4 Mathematical Model of the Proposed System

$$\sum_{i=1}^{N} \mathbf{Wi*Fi} \longrightarrow \quad (5.1)$$

Where,

N are the number of jobs submitted with in time duration T,

M are the jobs Competed,

N-M jobs are pending in the Job Q with in time duration T,

Wi is the benefit (weight) associated with job i.

Fi is the flag indicates if job is completed or not.

The main goal of the proposed system is to achieve maximum transfer of data with minimum time required. The data should complete within the time duration T and there should not occur deadlock.

Chapter 6

TESTING

The testing is done to find out the faults or failures occurred in the system. It enhances a way to check the components weather they are functioning properly. Testing can be done to each of the components used in the proposed system. Some of the testing cases are:

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing

6.1 Unit Testing

Unit testing is designed to test the internal logic is functioning properly or not. The unit testing comes under white box testing which tests the internal structure and the languages of the software.

Unit test case 1 for Controller

#Testcase	UTC-1
Forename of the Test	Unit Testing of SDN Controller
Item being Tested	Jobs in Queue
Sample Input	Proper Scheduling of Jobs
Sample Output	Admits jobs to the Datacentre Servers
Genuine Output	Same as expected Output
Comments	Successful

Table 6.1 Unit Test Case for Controller

Unit Test case 2 for Datacentre Servers

#Testcase	UTC-2
Forename of the Test	Unit Testing of Datacentre Servers
Item being tested	Status of the task or job
Sample Input	Scheduled job or task
Sample Output	Synchronizing to other datacentre servers
Genuine Output	Same as expected output
Comments	Successful

Table 6.2 Unit Test case for Datacentre Server

Unit Test case 3 for
Client

# Testcase	UTC-3
Forename of the Test	Unit Testing of Client
Item being tested	Uploading Files
Sample Input	Performing set of operations
Sample Output	Downloading the Files
Genuine Output	Same as expected output
Comments	Successful

Table 6.3 Unit Test for
Client

6.2 Integration Testing

Integration testing is designed for the purpose of testing the integrated software components which will combine the functions of two modules that is integrate as one program and then run them as a single program.

Integration Test Case 1 for Client and DataServer

# Test case	ITC-1
Forename of the Test	Integration Testing of Client and Datacentre server
Item being tested	Uploading the files or performing set of Operations
Sample Input	Set of key value pairs are performed
Sample Output	Operations performed successfully
Genuine Output	Same as expected output
Comments	Successful

Table 6.4 Integration Testing for Client and Datacentre Server

Integration Test Case 2 for Datacentre and Controller

# Test Case	ITC-2
Forename of the Test	Integration Testing of Datacentre server and Controller
Item being Tested	Admitting the task and scheduling the tasks
Sample Input	Jobs in Queue
Sample Output	Scheduling successful
Genuine Output	Same as expected output
Comments	Successful

Table 6.5 Integration test case for Datacentre server and controller

Integration Test Case 3 for Datacentre server with multiple Datacentre Servers

# Test Case	ITC-3
Forename of the Test	Integration Testing of one server with multiple Servers
Item being tested	Synchronizing the server with multiple servers
Sample Input	Uploading files
Sample Output	Synchronization is successful
Genuine Output	Same as expected output
Comments	Successful

Table 6.6 Integration Test case for Datacentre server with multiple Servers

6.3 System Testing

System Testing is designed for testing the entire framework of the proposed system. The system testing is done to the software‟s which has been used. This testing is done to check the compatibility of the software‟s are supported for the particular Operating System.

In the proposed system as we have used the cloud instances the system testing is done for the cloud instance weather the cloud is compatible to the OS.

System Test Case 1 for the Cloud instance

#Test Case	STC-1
Forename of the Test	System Test Case for Cloud Instance
Item being tested	Cloud instance which is been launched
Sample Input	Giving IP address of the current instance to the RDC
Sample Output	Cloud instance launched successfully
Genuine Output	Same as expected Output
Comments	Successful

Table 6.7 System Testing for Cloud Instance

System Testing Case 2 for the framework environment

# Test Case	STC-2
Forename of the Test	System Testing case for different Operating Systems
Item being tested	Versions of Operating System
Sample Input	Running in Windows 7, XP,2000,Linux
Sample Output	The most compatible version
Genuine Output	Same as expected output
Comments	Successful

Table 6.8 System Test Case for working framework environment.

6.4 Black Box Testing

Black-box testing is one of the techniques of Software testing which is used for checking the functionality of applications without appearing into its internal working streams.

Black-box testing can be done to the following applications such as:

- Decision table testing
- Error guessing testing
- Cause-effect testing

The black-box testing finds faults or failures in the external code they are:

- Missing functionality
- Interface errors
- Termination errors
- Performance errors

6.5 White Box Testing

White box testing is one of the techniques of Software testing which is used to test the internal working of the applications.

The white box testing can be done to the following applications:

- Data-flow testing
- Control-flow testing
- Path testing

Chapter 7

RESULTS and DISCUSSION

A screenshot taken how the modules are working.

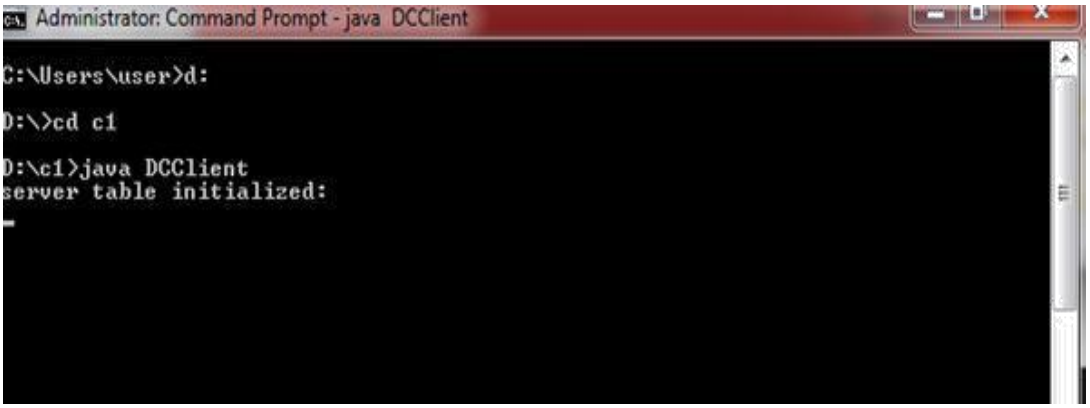

Fig 7.1 shows how the controller is started.

To run the controller, in the command prompt we have to specify the path where the controller is located. Here the file is located in the D: drive. Later we have to run the file as java filename that is java DCControler.

Fig 7.2 shows the SDN controller

Fig 7.3 shows how to initialize the client table

Here in fig 7.3 we specified the client path in the command prompt, then run the client which is given as java DCClient. Now the client table is initialized.

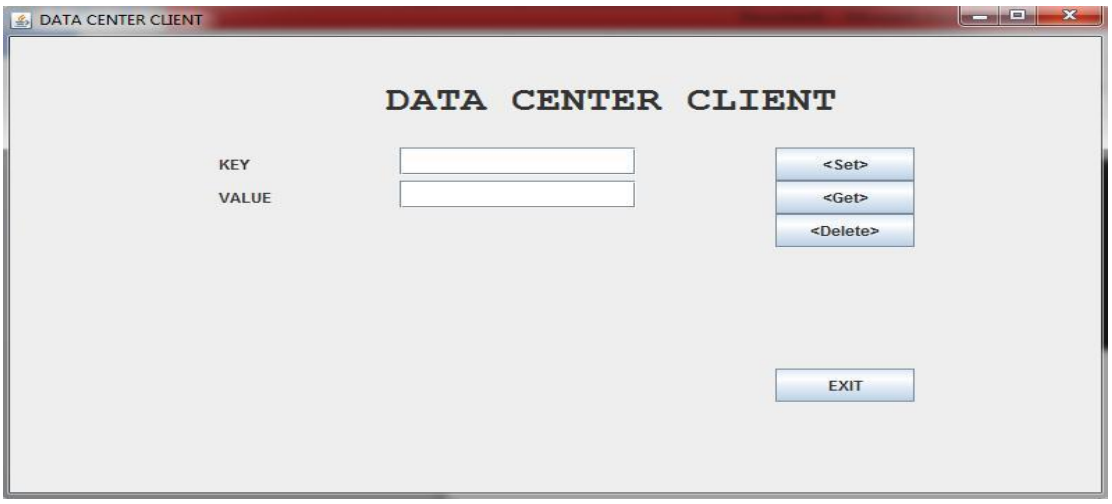

Fig 7.4 shows the datacentre client

Here in fig 7.4 shows the datacentre of the client, where it consists of key value pairs. Here there are few operations to be performed such as set, get and delete operations.

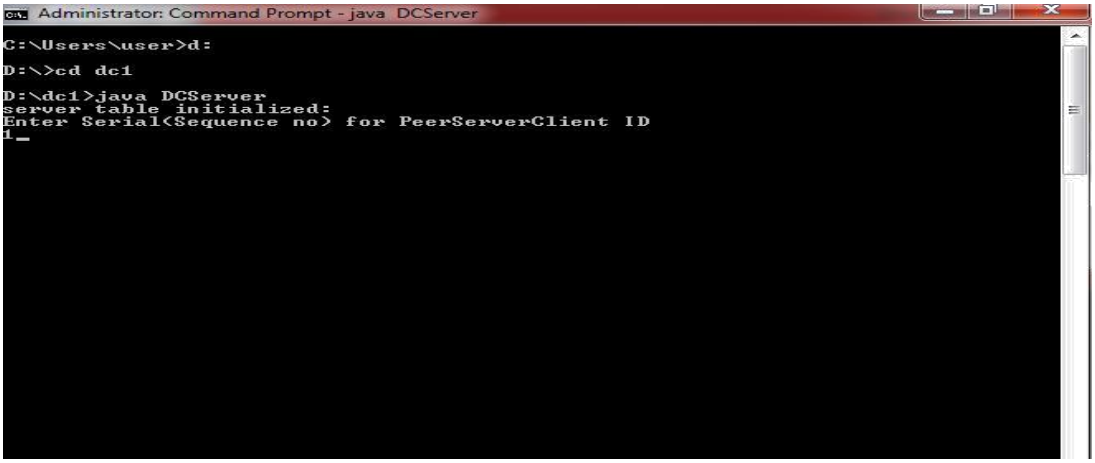

Fig 7.5 shows the initialization of server table.

Here in fig 7.5, we have to create datacentres, the datacentres can be created n number. So first we have to specify the path where the file is being located, here the file is located in the d: drive, now we have to run the file using java filename such as java DCServer. And it will ask the sequence number for peerserverclient ID which is indicated as the id for servers.

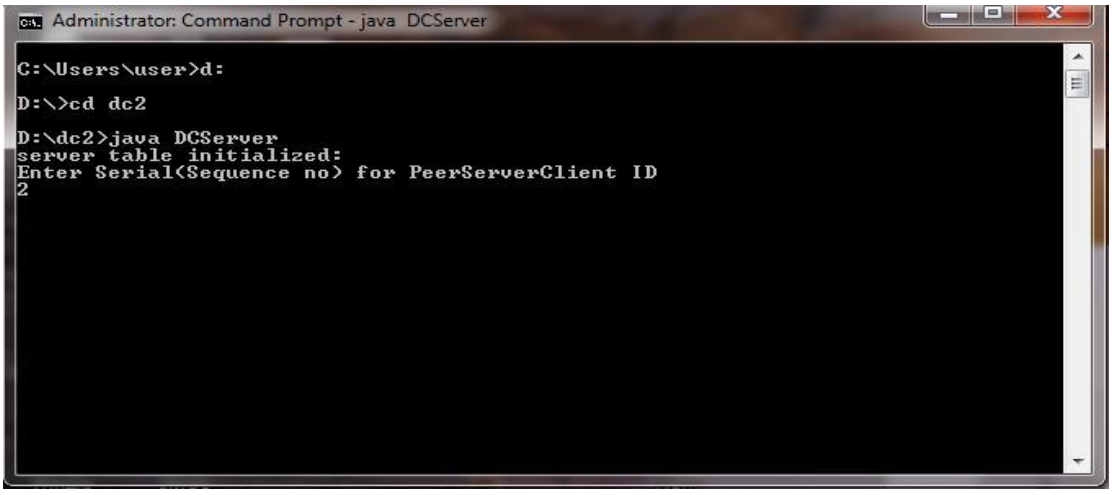

Fig 7.6 shows the second datacenter initialization

The initialization of second datacenter server with the sequence number 2.

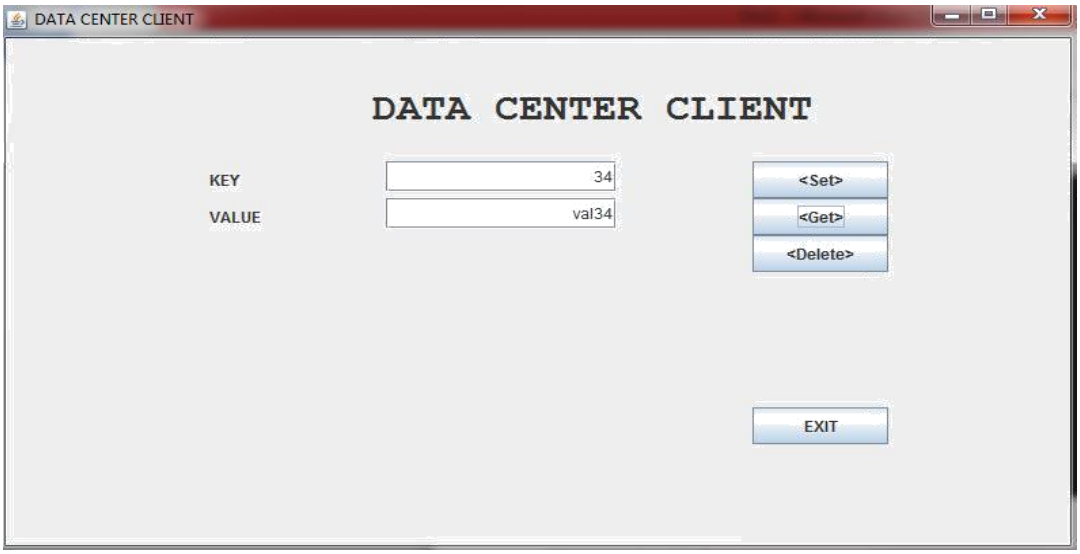

Fig 7.7 this table shows the entry of key value pairs.

The table consists key and value pairs, once we enter the key and value we can set these values and which will be submitted to the controller with the IP address of the client. The client will show to which server the task has been admitted because this is done dynamically.

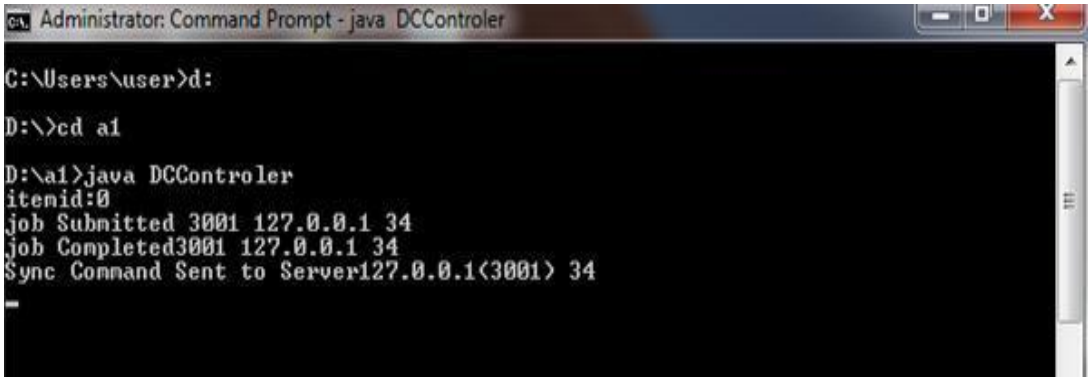

Fig 7.8 shows the task admitted to the controller.

As shown in the above figure, the key and value which is entered in the client table is admitted to the controller with the IP address of the client and it also shows that the synch command sent to which server.

Fig 7.9 shows that the client table is initialized with set key and connected to the server 2

As we entered the key and value in the client table and given the operation as Set, the set operation is initialized and the PUT operation is ok. And it is dynamically sent to the server 2 which is specified as the set key is connected to server id 2.

Fig 7.10 shows that the set operation is initialized to the server 2.

As shown above in the figure the datacentre server 2 as received the put command from the client. And the synch command is sent from the controller. And this server will get synched with the other datacentre server, which is given as the update request is sent to the server with specifying the host number.

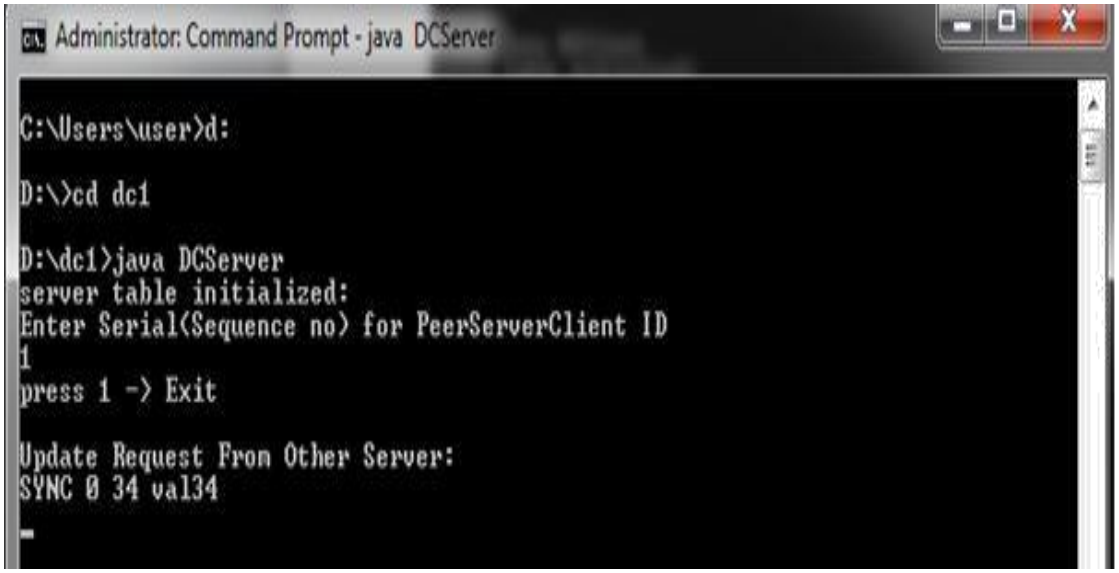

Fig 7.11 shows the update sent from the dc2 to the dc1.

As the update from the datacentre 2 is sent to the datacentre 1 which is known as synched with each other datacentre.

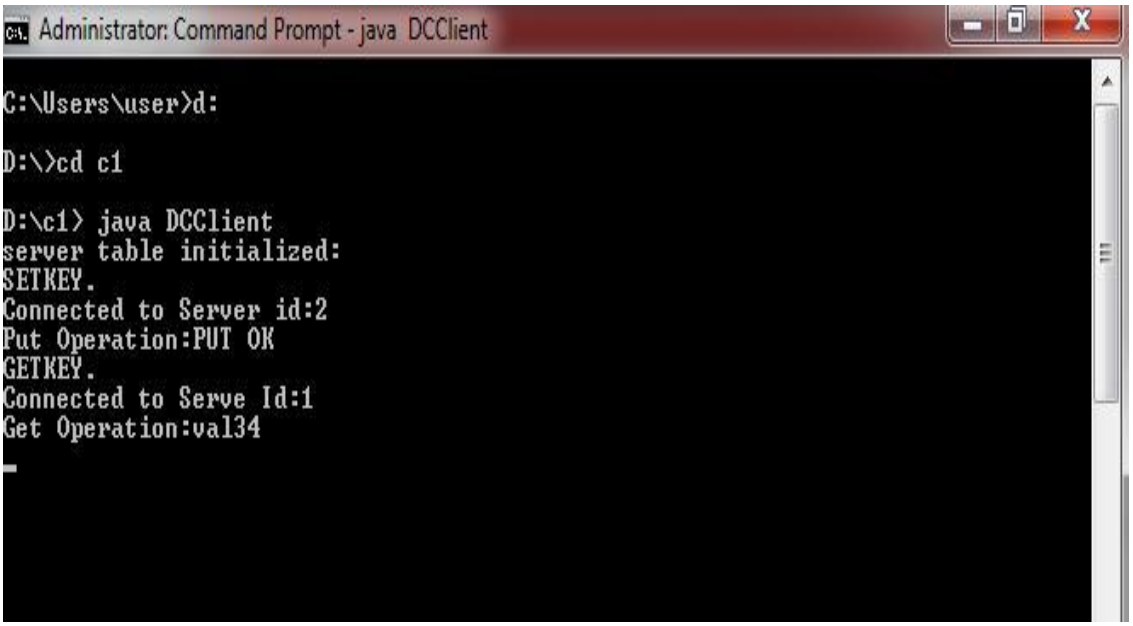

Fig 7.12 shows the get operation

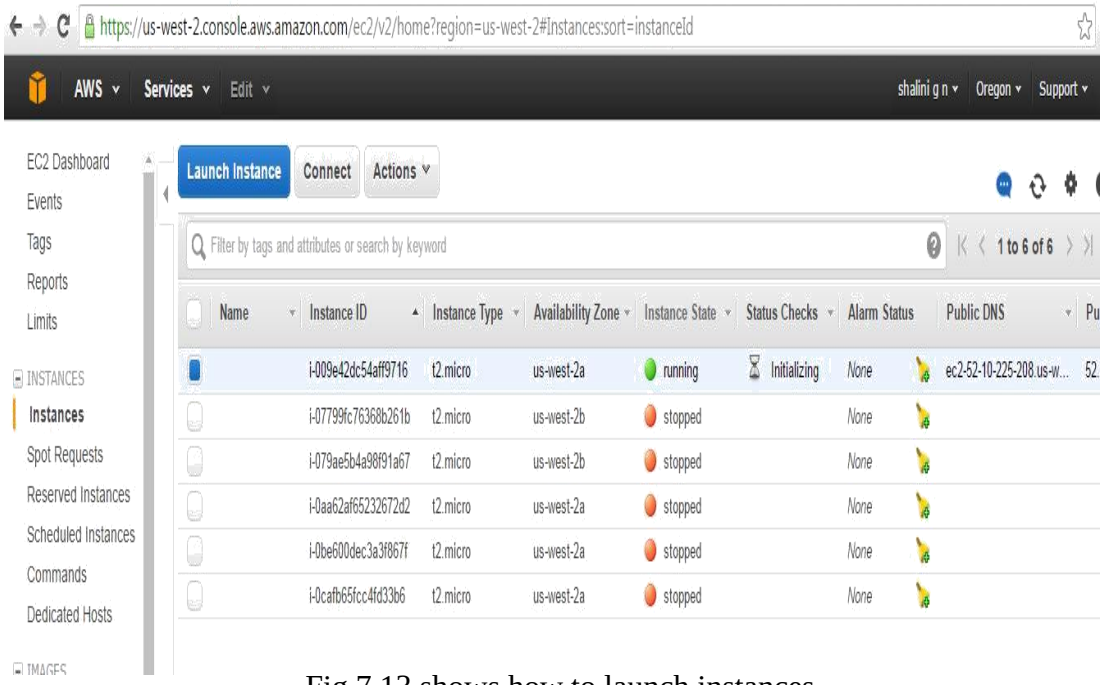

Fig 7.13 shows how to launch instances

The above figure 7.13 shows how to launch instances in AWS, here we can launch any number of instances.

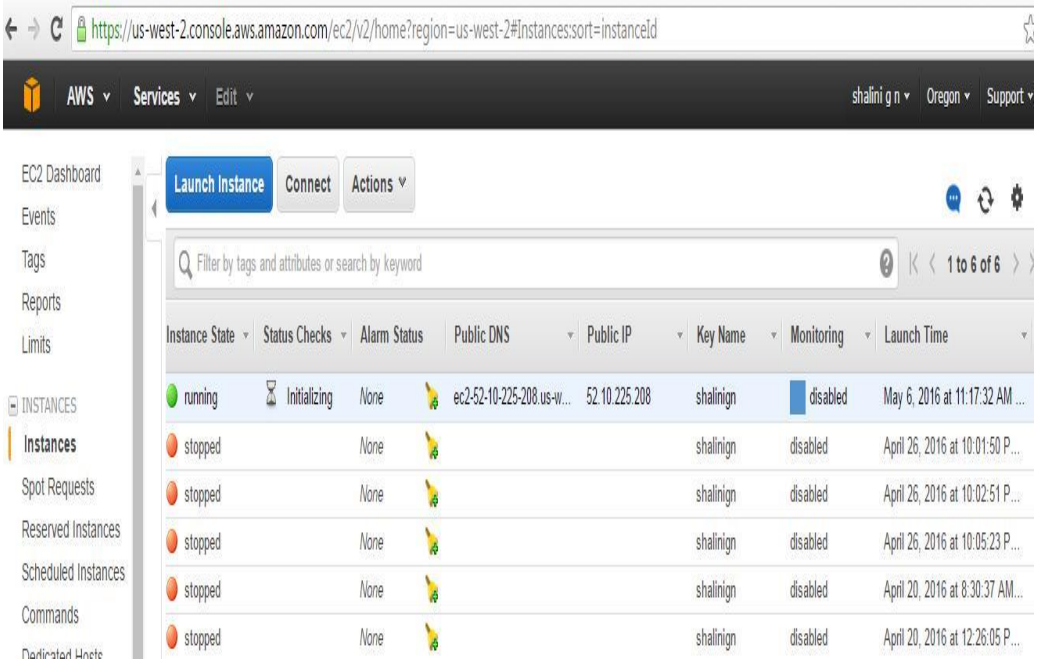

Fig 7.14 shows the Public IP as an current instance

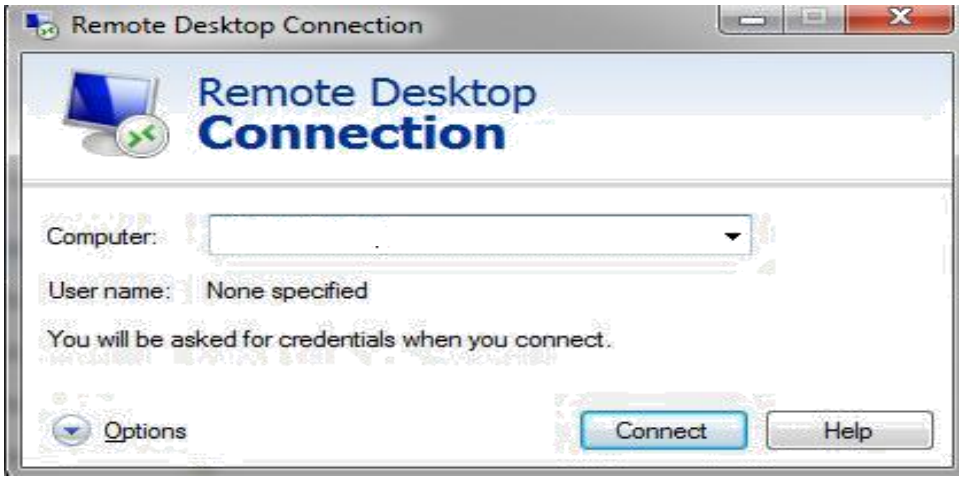

Fig 7.15 shows the RDC

Here we have to specify the current launched instance Public IP address.

Fig 7.16 Here in RDC we should specify user name.

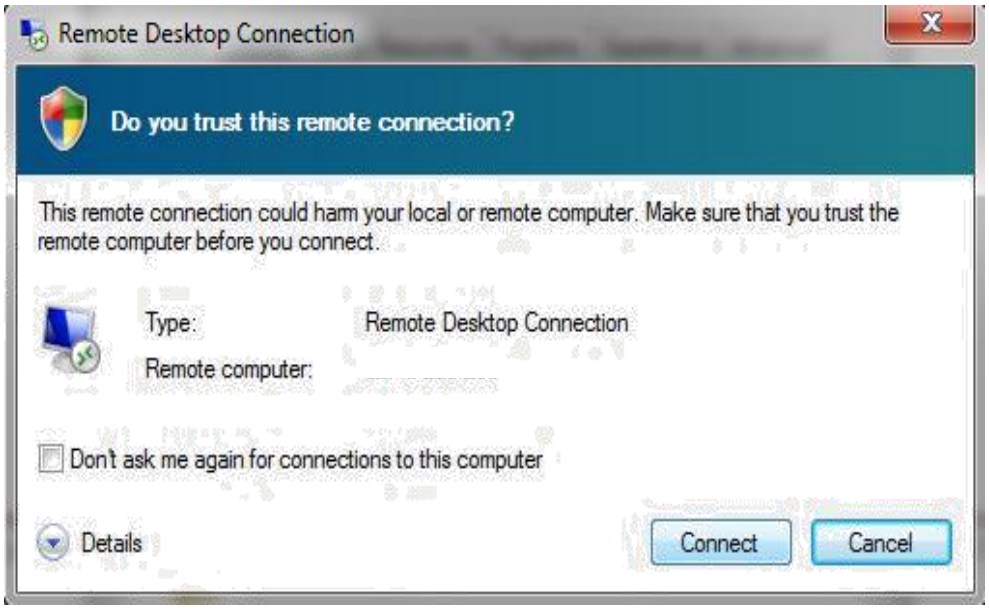

Fig 7.17 To connect to the RDC

Fig 7.18 after connecting it asks for user name and password.

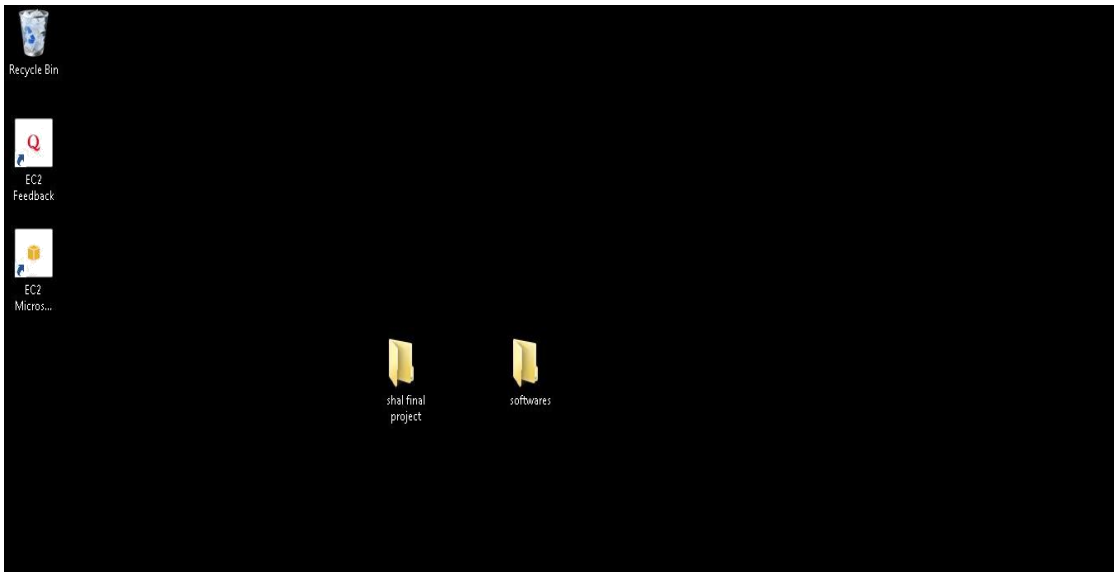

Fig 7.19 the framework of the cloud instance

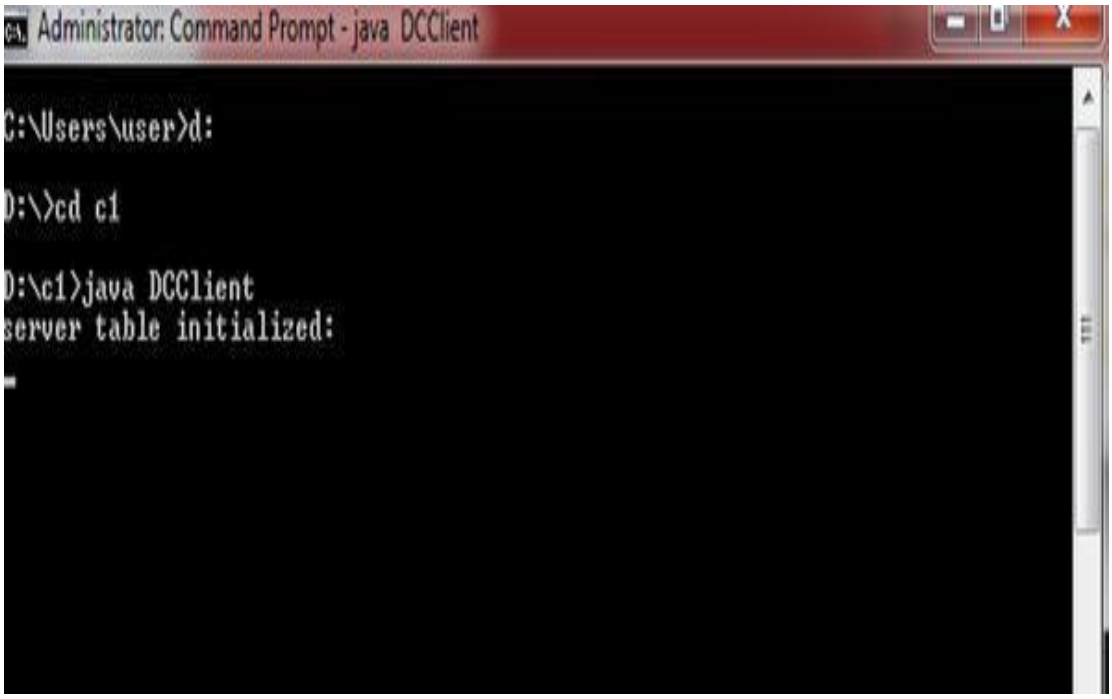

Fig 7.20 client table initialized.

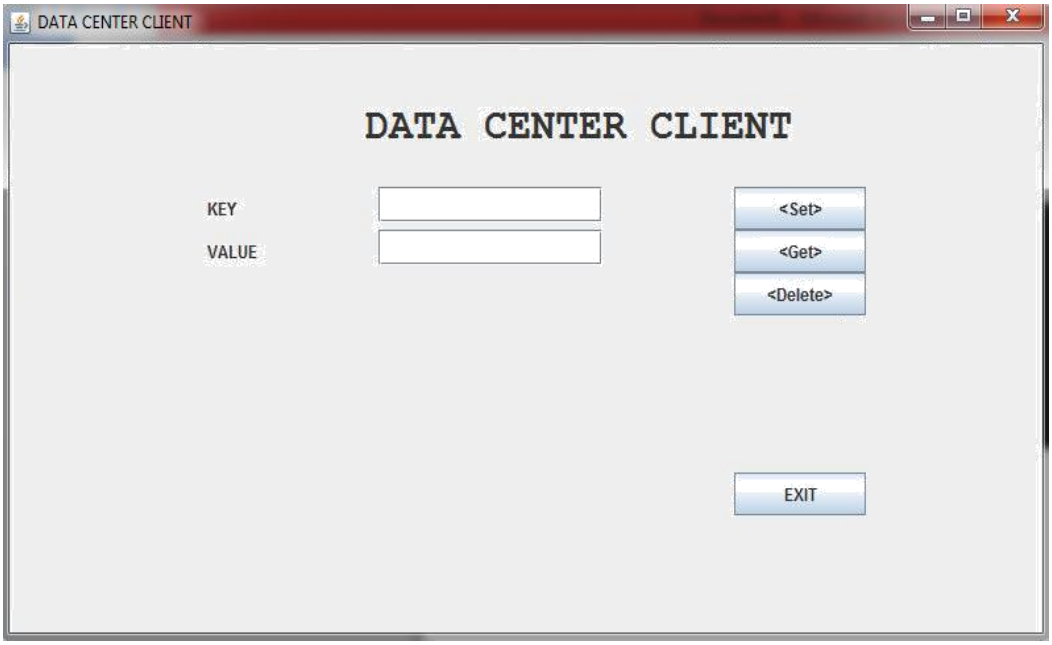

Fig 7.21 client table with key value pairs.

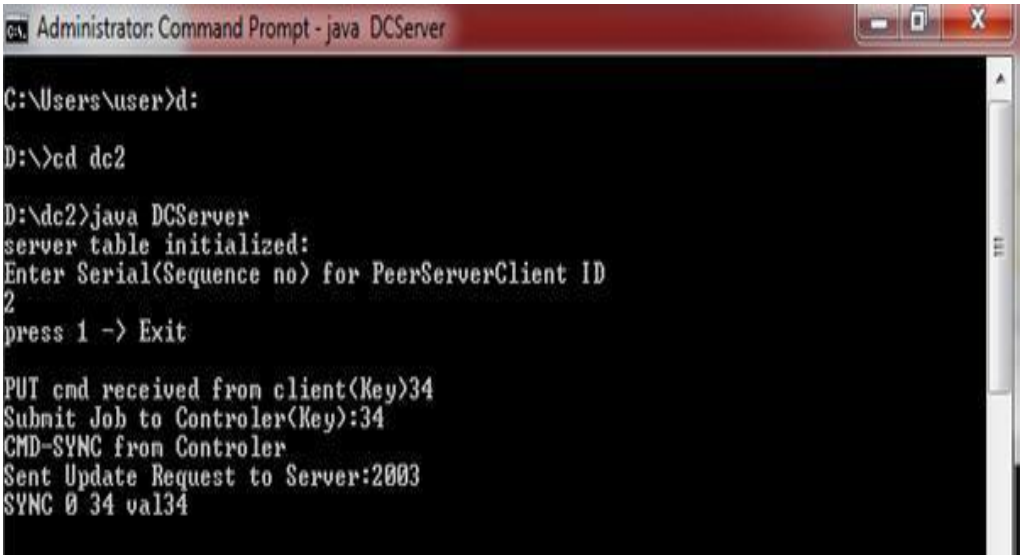

Fig 7.22 shows the values entered in the client table in servers

CONCLUSION

As multiple datacentres are deployed around the globe, which makes the users to efficiently utilize the resources. As there are multiple datacentres, public need not wait for the particular datacentre for the storage. Once there are multiple datacentres, there is required for the synchronization of the datacentres because in todays world number of people will be accessing the social networks simultaneously and the bulk amount of data will be generating every minute. Since there is bulk data to be transmitted around the world we need scheduling. One of the main challenge in the proposed system is to handle the scheduling of data in the datacentres, so there was an introduce of the controller known as Software Defined Networking (SDN), which controls the system and schedules the data in a proper manner. Hence the proposed system has given solution for the challenge faced.

FUTURE ENHANCEMENT

In the proposed system, we have used only key value pairs in the client side using hashing algorithm. They can further use all the type of data to be uploaded from the client side to the server.

Some of the improvements that can be further for the proposed system are:

- It can be implemented in other Operating System
- It can written in other Programming language such as python, php, etc
- In further they can use some other dynamic algorithms for achieving optimal chunk routing.

REFERENCES

[1] G. O. Young, ―Synthetic structure of industrial plastics (Book style with paper title and editor), in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. M. Lau, "Scaling Social Media Applications into Geo-Distributed Clouds," in INFOCOM, 2012.

[3] K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, "Live Data Center Migration across WANs: A Robust Cooperative Context Aware Approach," in Proceedings of the 2007 SIGCOMM workshop on Internet
network management, ser. INM '07, New York, NY, USA, 2007, pp. 262–267.

[4] SDN,https://www.opennetworking.org/sdn-resources/sdn-definition.

[5] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, "Better Never than Late: Meeting Deadlines in Datacenter Networks," in Proceedings of the ACM SIGCOMM, New York, NY, USA, 2011, pp. 50–61.

[6] B. Vamanan, J. Hasan, and T. Vijaykumar, "Deadline-aware Datacenter TCP (d2tcp)," in Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication. ACM, 2012, pp. 115–126.

[7] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter Bulk Transfers with Netstitcher," in Proceedings of the ACM SIGCOMM 2011 conference, New York, NY, USA, 2011, pp. 74–85.

[8] B. B. Chen and P. V.-B. Primet, "Scheduling Deadline-constrained Bulk Data Transfers to Minimize Network Congestion," in Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid, 2007, pp. 410–417.

[9] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving High Utilization with Software-Driven WAN," in Proceedings of the ACM SIGCOMM, Hong Kong, China, 2013.

[10] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[11] DataCenterMap, http://www.datacentermap.com/datacenters.html.

APPENDIX

1. Shalini G N, Dr. Arun Biradar, "A Survey On: Synchronizing Multiple Datacentres across the Globe Hosted by Service Providers" in Advances in Computing, Communication and Networking (ACCNet-2016), Sponsored by Savitribai Phule Pune University and in association with IJCA journal [USA] during 9th & 10th February 2016 at G. H. Raisoni College of Engineering and Management, Wagholi, Pune, ISSN awaiting.

2. Shalini G N, Prasanna Kumar M, "Synchronizing Multiple Datacentres across the Globe Hosted by Service Providers", International Journal for Science and Advance Research in Technology, website: www.ijsart.com , ISSN [ONLINE]: 2395-1052, Paper ID: IJSARTV2I53331, Volume 2, Issue 5 in May 2016