Design and development of prediction model for cloud federation

Submitted By

Maulik Doshi

14MCEC06

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Design and development of prediction model for cloud federation

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Maulik Doshi

(14MCEC06)

Guided By

Dr. Madhuri Bhavsar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Certificate

This is to certify that the major project entitled **"Design and development of prediction model for cloud federation"** submitted by **Maulik Doshi (14MCEC06)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Madhuri Bhavsar
Guide & Professor,
Section Head (IT),
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr P.N Tekwani
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Maulik Doshi**, Roll. No. **14MCEC06**, give undertaking that the Major Project entitled "**Design and development of prediction model for cloud federation**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. Madhuri Bhavsar

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **DR. Madhuri Bhavsar**, Professor & Section Head -IT, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr P N Tekwani**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Maulik Doshi

14MCEC06

Abstract

The demand of computing resources is expected to increase dramatically which creates the high demands for cloud resources. All available resources of single cloud service provider may not be enough to cope up with such demands. Solution to this problem is the federation of clouds. The goal of cloud federation is to make the assets cost-effective and optimize the resources among the heterogeneous environments where different clouds can cooperate together with the goal of obtaining unbounded and virtually infinite computation resources. But to determine whether federation is required or not depends on certain parameters. This research work is intended to design and develop the optimized model for cloud federation which will predict the high demand of resources and effectively use the cloud resources ultimately resulting to achieve the profit gain.

Abbreviations

CSP	Cloud Service Provider.
IaaS	Infrastructure as a service.
PaaS	Platform as a service.
SaaS	Software as a service

Contents

List of Figures

Chapter 1

Introduction

Cloud Computing is a new paradigm which delivers computing as a utility through the Internet [1]. Cloud service providers provide various services like infrastructure as service (IaaS), Platform as a service(PaaS) and software as Service(SaaS) by pay as you go model over Internet. This system may provide significant benefits to various organizations as they need not worry about setting up their hardware and software infrastructures and thus decreasing the cost.

As more and more organizations turn towards the usage of cloud services the need of cloud resources is increasing day by day. This cloud resources include hardware resources viz.:- CPU, network, Storage and software resources viz.:- databases, application servers and Web servers and applications. One of the key features of cloud computing is having virtually infinite computation resources available whenever required. CSPs may or may not have the amount of resources available that the consumer is requesting. Single a CSP may not cope up with the huge resource demands made by the consumers. It is essential for CSPs to handle the peak-load from consumers along with maintaining the property of elasticity. To achieve this goal of virtually infinite resources availability, a CSP opts for federation with other CSPs.

Cloud federation or Inter-cloud is an interconnected global "Cloud of Clouds". Where a cloud service provider, during the time of its need, takes on lease the resources from the other cloud service providers. Cloud federation enables the cloud service providers to add their resources in the Global Cloud Market. Consumers also may get the choice to

select the CSP from the federation as per their requirement.

The objective for this research work is to develop the prediction model to determine the requirement of the federation of cloud.

Chapter 2

Literature Survey

2.1 General :-

Cloud computing alludes to the web based innovation to give different sort of administrations to the customers. It is a sort of administration in which client can get assets on the fly and pay appropriately according to they use which is termed as "pay-as-you-go". All the expense related with respect to setting up foundation like equipment, power supply, area, cooling framework and so forth are to be finished by the cloud administration supplier. Client is charged just for the assets he/she has used. Since the client is charged just for PC resources utilized, cloud computing expenses are a small amount of conventional innovation uses.

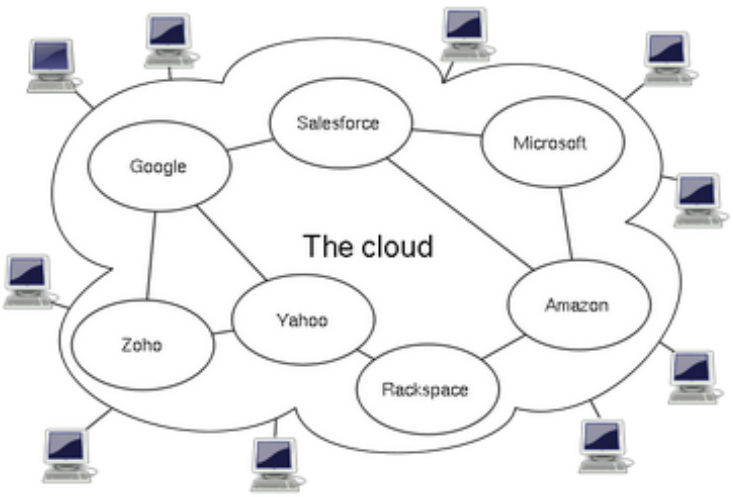

Figure 2.1: The cloud

2.2 Cloud Basics:-

Characteristics of Cloud

Cloud computing have following basic characteristic:

Shared Infrastructure — this permits sharing of physical resources like storage and networking by applying virtualization. It looks for the effective utilization of the available infrastructure by sharing it among the clients.

Dynamic Provisioning — this permits the procurement of resources based on the necessaties of the clients dynamically. This should be possible automatically by using software automation with elastic behavior i.e. enabling the expansion and contraction of services.

Network Access — Cloud needs to be accessed across broad range of devices via. internet.

Managed Metering — This is to be done for the billing of the consumers. Consuers are billed according to their usage of services.

Service Models

Cloud computing provide following three basic services: [2]

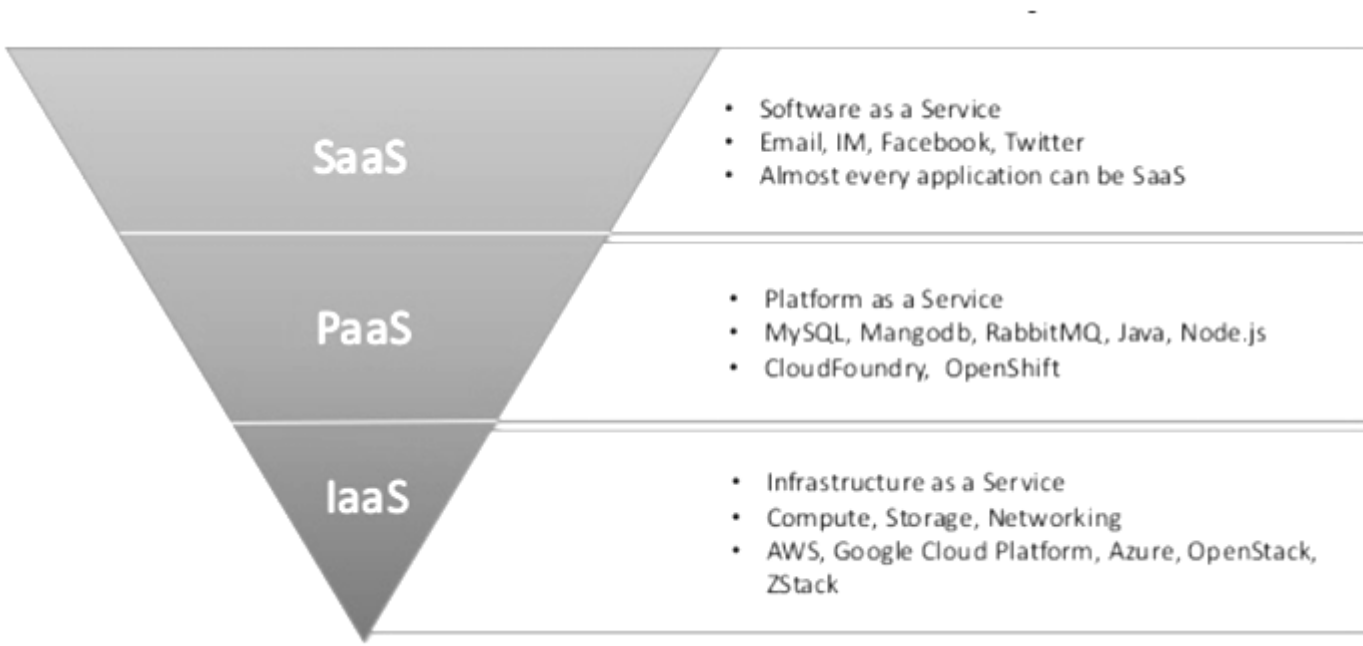

Figure 2.2: Cloud service model stack

Infrastructure as a service (IaaS)

Infrastructure as a service is a cloud service that delivers storage and compute capabilities over the network. Pool of Servers, storage systems, switches, routers, and other systems are made serve the workload requirements ranging from application components to HPC application. Commercial examples of IaaS include Google compute engine, Amazon EC2 etc.

Platform as a service (PaaS)

Platform as a service is a cloud service that provides platform by encapsulating software layer. This software layer is used to build higher level software services.

Software as a service(SaaS)

Software as a service is a cloud service which offers a complete application on demand. A single instance of the software runs on the cloud and services multiple end users or client organizations. The most widely known example of SaaS is salesforce.com, though many other examples have come to market, including the Google Apps offering of basic business services including email and word processing. Although salesforce.com preceded the definition of cloud computing by a few years, it now operates by leveraging its companion force.com, which can be defined as a platform as a service.

2.2.1 Deployment Models

Public cloud

Public cloud also known as external clouds are run by third parties known as cloud service provider, and applications from different customers are likely to be mixed together on the cloud's servers, storage systems, and networks. A public cloud provides services to multiple customers.

Private cloud

Private clouds are built for the exclusive use of one client, providing the utmost control over data, security, and quality of service . The company owns the infrastructure and has control over how applications are deployed on it. Private clouds can be built and managed by a company's own IT organization or by a cloud provider

Hybrid cloud

Hybrid clouds is the combination of both public and private cloud models giving the benefits of both. This is most often seen with the use of storage clouds to support Web 2.0 applications.

2.3 Cloud Architecture

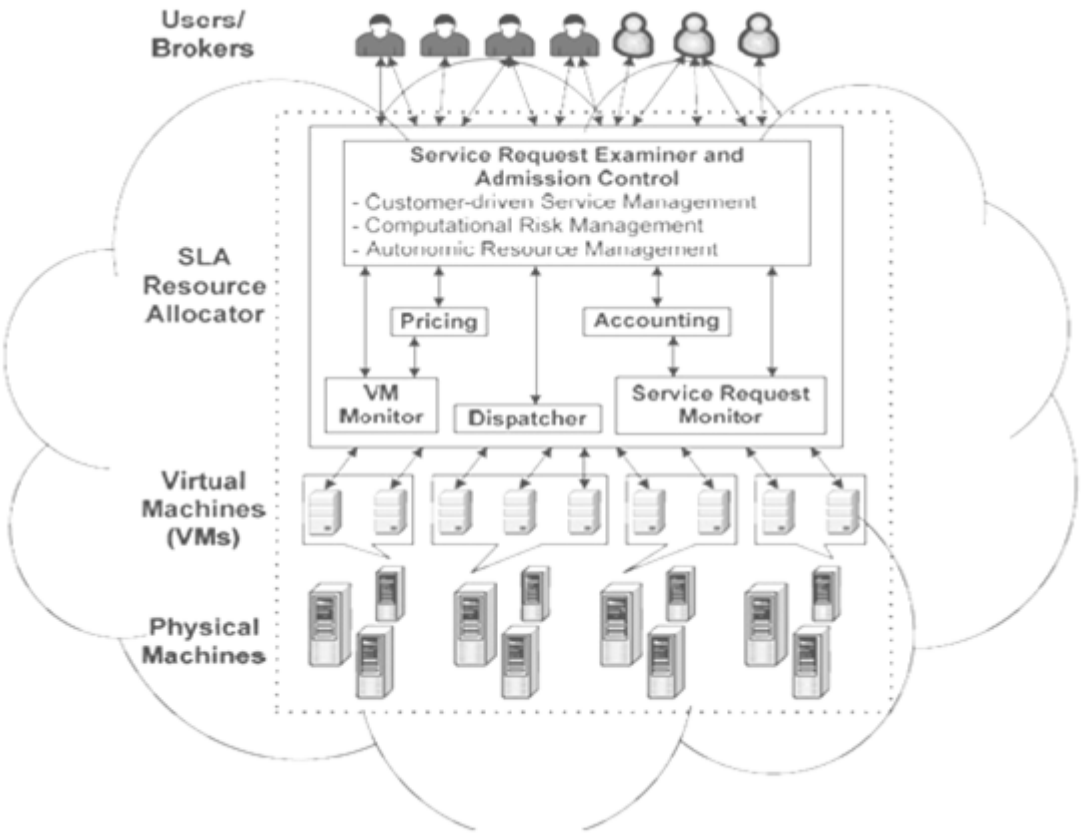

Figure 2.3: Market oriented cloud architecture

As shown in fig 2.3 The lowest layer of the architecture is of physical machines which provide the necessary infrastructure for the cloud. Above that the layer called virtual machine which virtualizes the below physical hardware. SLA resource allocator manages the resource provisioning and allocation to the users. Service Request Examiner and Admission control interprets the feasibility of QOS requirements of users request and assigns the VMs to the user if requirements are fulfilled. VM monitor mechanism is responsible for the information about the availability of the VMs. Dispatcher dispatches the resources to the accepted service request. Pricing manages the billing of the all users. Accounting maintains the log of the actual resource usage by the request.

2.4 Resources in cloud

Cloud resources are the any resources either physical or virtual that are requested by the users/brokers. Cloud resources include various resources and not limited to:

1.Specific Hardware

2.Operating system

3.Interconnection network

4.Computation power

5.Storage

Cloud service provider needs to maintain the information about this resources.This includes: Monitoring information of the physical hardware Information about virtual machine Cost of VM provisioning and VM allocation Data locality.

2.5 Cloud Platforms

OpenNebula

[3] OpenNebula is an open source framework to manage the virtual infrastructure of the cloud. Using OpenNebula one can build private,public, hybrid as well as federated clouds. OpenNebula offers adaptable structural engineering, interfaces and segments that could be coordinated into any datacenter. OpenNebula oversees virtualization, storage and other cloud resources thus joining the datacenters into a cloud. This tool suppots various hypervisors such as Xen, KVM and VMware . OpenNebula also allows access to Amazon EC2s.

Eucalyptus

[4] Eucalyptus is an opensource architecture that uses storage and computational infrastructure to provide a Cloud computing platform. Like OpenNebula, Eucalyptus takes into consideration alterations to the current APIs for distinctive prerequisites. Eucalyptus executes the Amazon Web Service (AWS) API which encourages interoperability with existing tools and services perfect with the AWS API. Eucalyptus gives a secluded, extensible structure with an Amazon EC2 compatible interface which can be used for cloud federation at the IaaS layer.

OpenStack

[5] OpenStack is a worldwide cooperation of developers and cloud experts to deliver the open standard cloud os for both public as well as private clouds.. CSPs , private organizations and government associations associations can exploit the unreservedly accessible, Apache authorized programming to construct hugely scalable cloud.environments. OpenStack at present comprises of three center programming activities: OpenStack Compute (code-name Nova), OpenStack Object Storage (code-name Swift), and OpenStack Image Service (code-name Glance). These tasks, alongside a lively community of innovation suppliers and future OpenStack activities in progress, convey a pluggable structure and working framework for public and private clouds.

Chapter 3

Cloud Federation

Inter-Cloud computing has been formally defined as

A cloud model that, with the end goal of ensuring service quality, for example, the execution and accessibility of every service, permits on-demand reassignment of resources and exchange of workload through an interworking of cloud frameworks of diverse cloud suppliers based on coordination of every buyers prerequisites for administration quality with every suppliers SLA and utilization of standard interfaces.

3.1 Classification of federated clouds.

As mentioned in [6] Cloud federation can be classified as follows:

3.1.1 Centralised Federation

In this type of federation , there is a central entity, cloud collaborator which acts as a middleware for negotiation. Clouds willing to take part in federation register themselves cloud collaborator. Cloud collaborator facilitates resource allocation when resources are required from the federation.

3.1.2 Peer-to-peer

In peer-to-peer type of federation, there is no middle party. Clouds collaborate directly by communicating with each other for negotiations. They share their resources based on their SLAs.

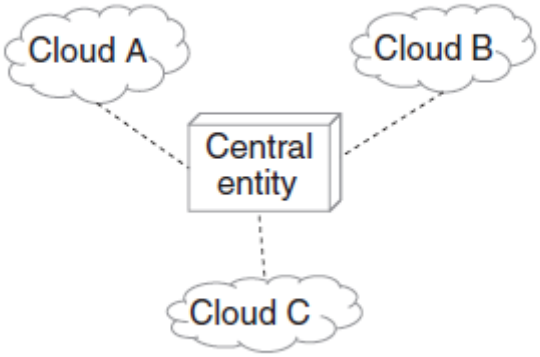

Figure 3.1: Centralised federation

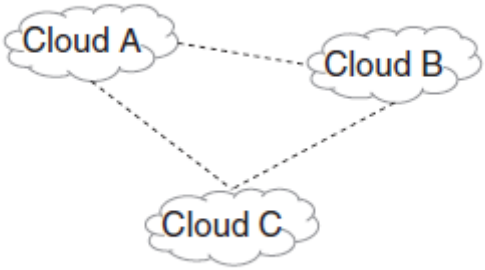

Figure 3.2: Peer-to-Peer federation

3.2 Why Federation?

Though cloud computing holds lots of promise for enterprise computing, there are numbers of limitations in current offerings such as:

1.Inherently limited scalability of Single CSP

2.No Interoperability between two CSPs

3.No In-Built service mangament support for businesses.

3.3 Benefits of cloud federation

As mentioned in [6] following benefits can be achieved by cloud federation:

Cost effectiveness

Cloud federation provides virtually infinite amount of resources. This may help in reducing the time to completion which in turn improves the cost-effectiveness.

Acceleration

Huge level of parallelism can be achieved by running application on cloud. By cloud federation constraint we get more computation power which may be exploited to get additional level of parallelism.

Resilence

When there is some failure or unexpected situation at any of the CSP ,he can request the other CSPs for the resources in the cloud federation.

Energy efficiency

Energy efficient cloud computing can be done by using federated cloud computing. If two or more CSPs are running less number of VMs. VMs can be migrated from one CSP to other and one of the CSP can be kept in sleep mode to save energy.

3.4 Cloud federation literature

3.4.1 Layer to Layer Federation in cloud.

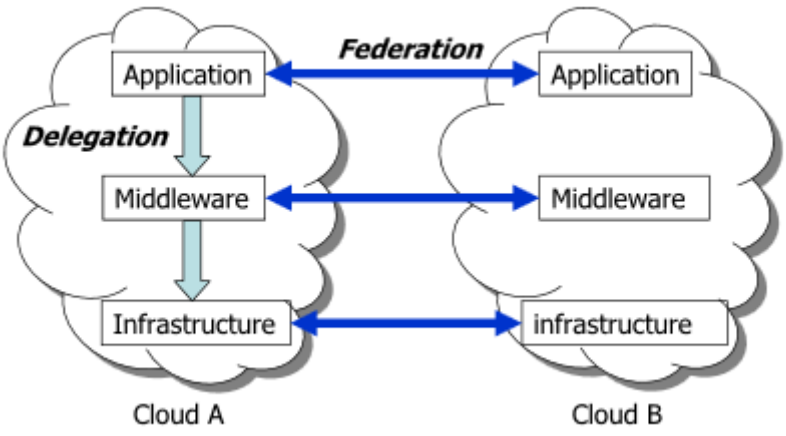

Figure 3.3: Layer to Layer Federation in cloud

In [7] Cloud Federation in a layered service model, they have show the federation at each layer level between the clouds participated in federation. This layer to layer federation takes place as per the SLAs agreed by the all the parties involved in the federation.

In this approach clients requests to the cloud stack layer which evaluates if adequate local resources are accessible to service the requests within a predetermined time. On the off chance that the that layer can't meet its objectives locally it can alternatively satisfy requests through an independent federation layer provider of the same service as indicated by the horizontal (federation) line interfacing Cloud A to B.

3.5 Horizontal cloud Federation Model

In [6] Cloud Federation, they have specified two dimensions of cloud federation in their model. One is horizontal model and the other is vertical model In their research work they have focused on horizontal model. Two types of scenarios : Redundancy and migration.

3.5.1 Redundancy

Redundancy is utilized at whatever point there is a subset of (appropriately sorted out) cloud administration offerings that give preferable utility to a customer over any single administration.

3.5.2 Relocation

Relocation can be activated when another cloud offering offers preferred utility to a customer over any already utilized administration advertising

3.6 Brokering Mechanism for cloud federation

Brokering mechanism in cloud federation environment is implemented in following way.[8]

3.6.1 SLA based

Service Provider specify the brokering requirements in an SLA in the form of constraints and objectives. The cloud provider or the Inter-Cloud service acting on behalf of the client decides on brokering approach honouring the specified SLA.

3.6.2 Event-Trigger based

Here service provider decides the specific events which triggers the action to create federation. This event or trigger can be any activity or set of activites with predetermined condition. For example, CPU utilization of VM increase above threshold may trigger event to start Computer federatioin.

Chapter 4

Proposed Model

4.1 Proposed Cloud Model

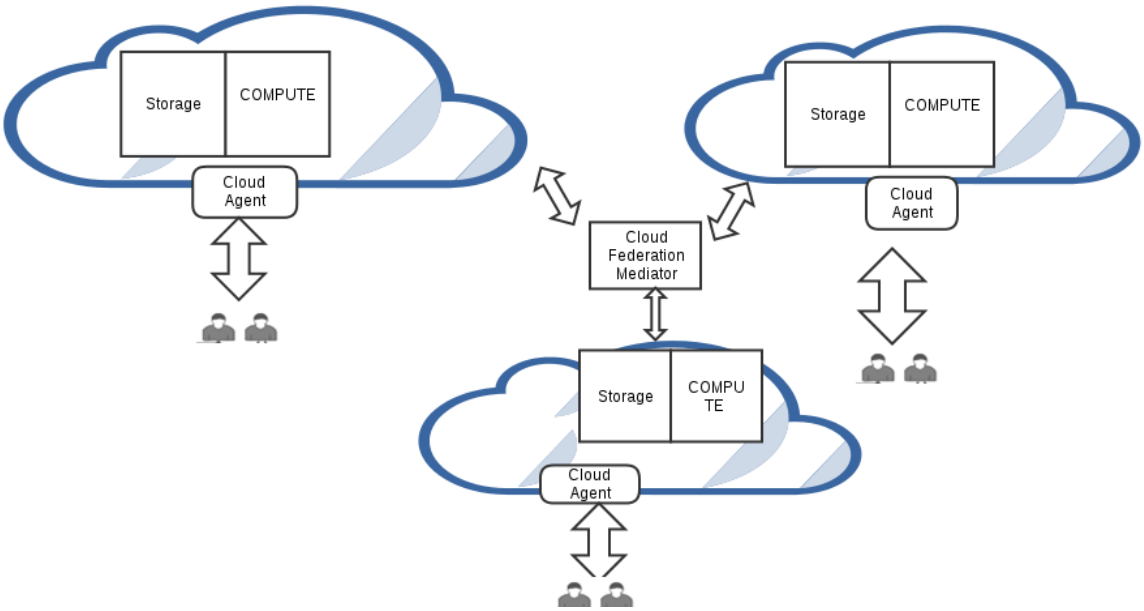

Figure 4.1: Proposed cloud model

In our proposed model of cloud there are 3 entities. Here we are proposing federation in centralised manner. All clouds are working independently when there is no huge demand. But when they run out of resources they opt for federation.

In figure 4.1 there are independent cloud service providers. Users request their requirements for compute or storage resources.Each CSP has a mechanism called cloud agent. Cloud agent is responsible for the allocation of resources to the clients. For this Cloud agent maintains the information log about the status of each of the resources and their

availability in the cloud. When user request for any resources, it is the responsibility of cloud manager to check and grant the resources to the user if available.
If cloud service provider is out of resources, it requests the cloud federation mediator for federation. Cloud federation mediator maintains the information about all the clouds in the federation. When any CSP ask for federation cloud service mediator finds the cloud service provider with minimum load and provide their resources to the heavily loaded CSP.For this Cloud service mediator maintains the information about the load of each cloud in the federation.

4.2 Use-case Diagram

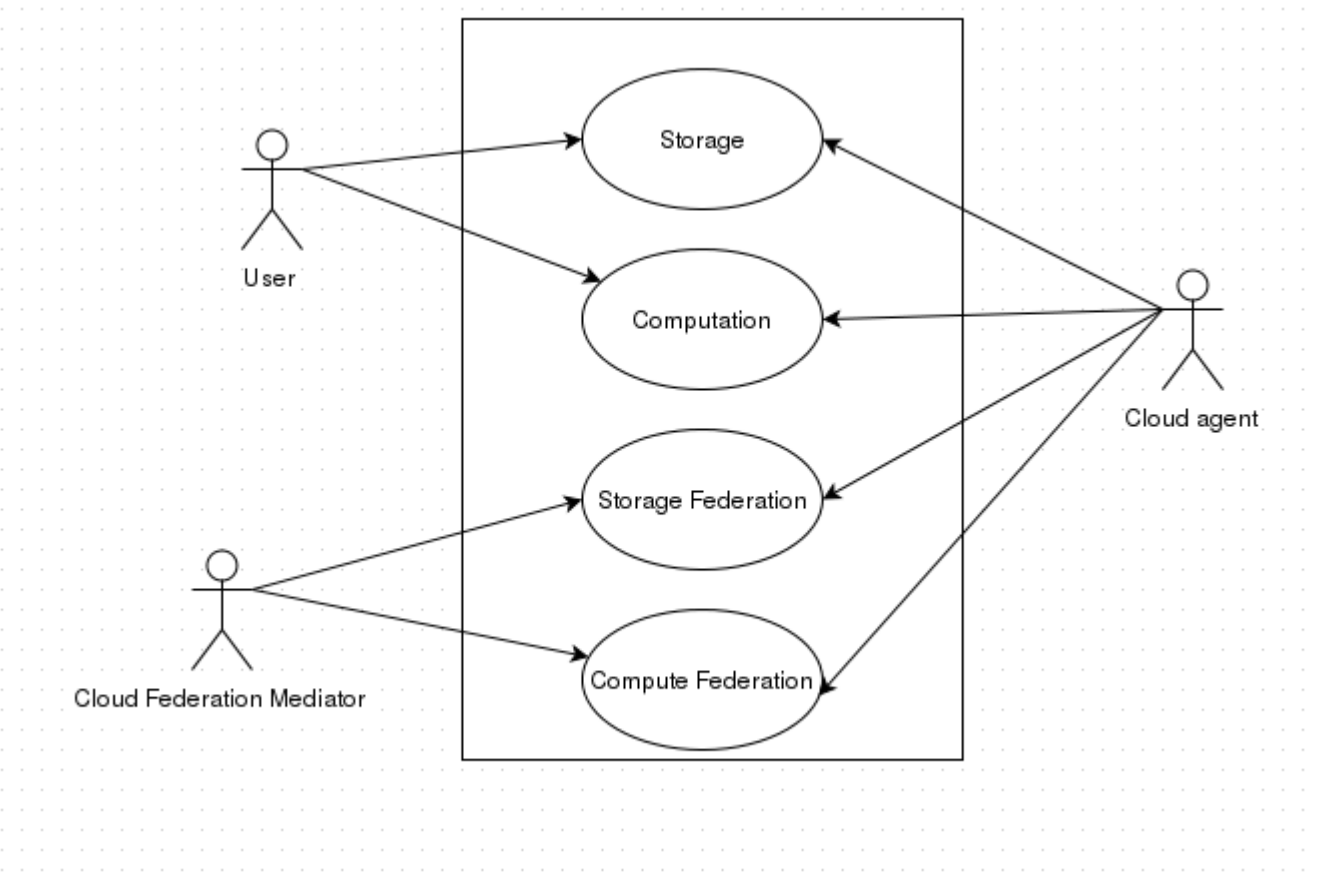

Figure 4.2: Use-case Diagram

4.2.1 Actors

1.Users

2.Cloud Agent

3. Cloud Federation Mediator

4.2.2 Activity

1. User request for resource allocation
2. User request for resource deallocation
3. cloud agent manages storage and computation resources
4. Cloud agent ask for storage or compute federation
5. Cloud Federation Mediator manages the compute and storage federation.

4.3 Proposed Algorithm

The algorithm is intended to predict the requirement for the federation of cloud. It is divided in 3 parts

1. Algorithm for cloud agent.
2. Prediciton algorithm.
3. Algorithm for cloud federation mediator.

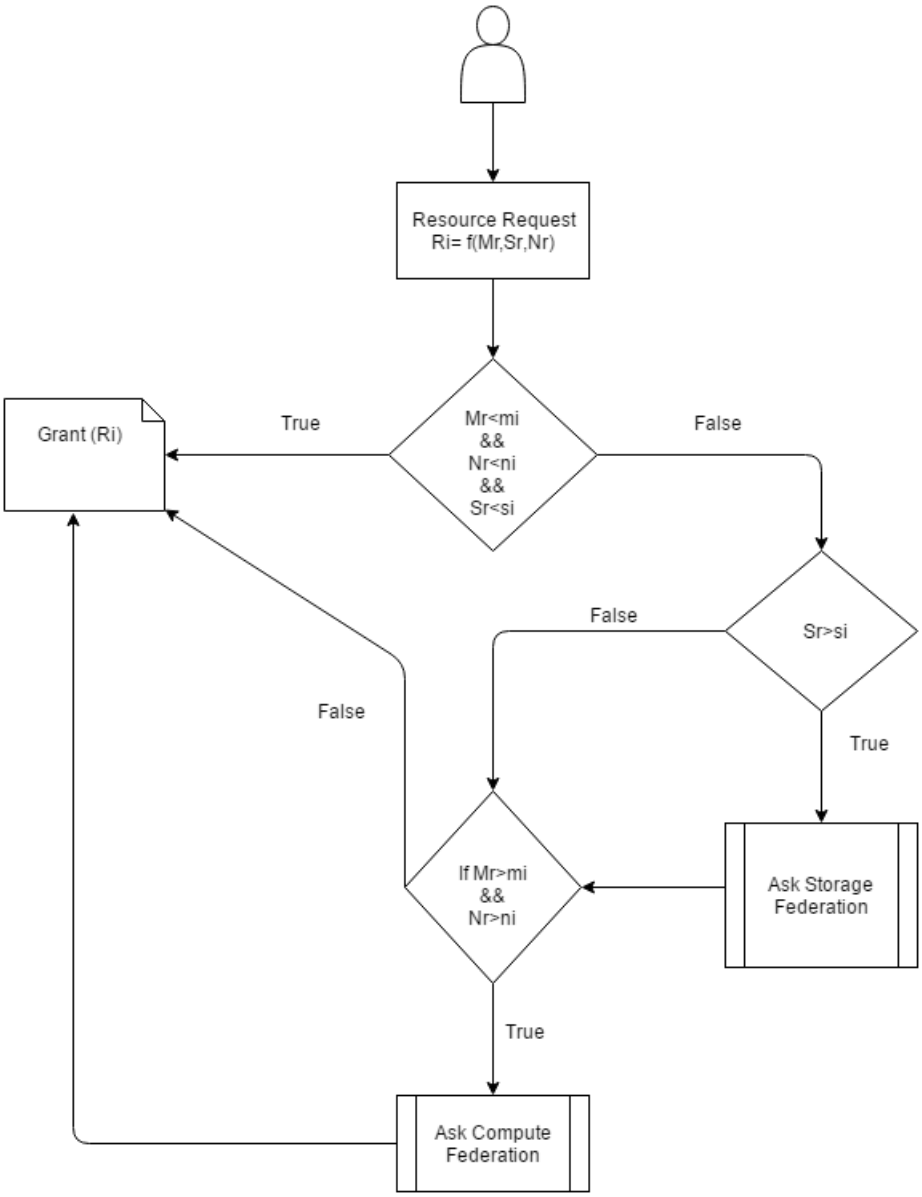

Figure 4.3: Flowchart of Algorithm 1

Algorithm 1 Cloud Agent

I ={C1,C2,... Cn} is set of cloud providers in federation.
n=sizeof(I) is total number of cloud in federation.
M(i)={M1,M2,... Mn} is Max Memory of each cloud Ci
P(i)={P1,P2,... Pn} is Max CPU in infrastructure of Cloud Ci from I
S(i)={S1,S2,... SN} is Max storage in infrastructure of Cloud Ci from I
mi={m1,m2,... mn} is amount of available memory of each Ci from I.
pi={p1,p2,... pn} is amount of available CPU of each Ci from I.
si= {s1,s2,... sn} is amount of available Storage of each Ci from I
Bi= {B1,... Bn} is cloud Broker for cloud Ci from I
Mr= {Mr1... Mrn} is amount of memory requested by user to cloud Ci
Nr= {Nr1,Nr2... Nrn} is amount of Cpu requested by user to Cloud Ci
Sri={Sr1,....Srn} is amount of Storage requested by users to Cloud Ci.
L(t)i= {L(t)1, L(t)2,... L(t)n} is Computation load on cloud Ci at time t.
ECL(t)i= { ECL(t) 1,ECL(t)2,... ECL(t)n} is estimated compute load at time t.
Sl(t)i= { Sl(t)1, Sl(t)2,... Sl(t)n} is Storage load on cloud Ci at time t.
ESl(t)i= { ESl(t)1, ESl(t)2,... ESl(t)n} is Estimate Storage utilization at time t.
UTHsi= { UTHs1,UTHs2,... UTHsn} is Upper Storage Threshold for cloud Ci
UTHci= { UTHc1,UTHc2,... UTHcn} is Upper Computation Threshold cloud Ci
LTHsi = { LTHs1,LTHs2,LTHsn} is Lower Storage Threshold for cloud Ci
LTHci= { LTHc1, LTHc2, LTHcn} is Lower Computation Threshold for cloud Ci.
Get Allocation Request from Users Ri = f(Mr,Nr,Sr
if (Mr<mi && Nr<ni && Sr<si) **then**
    mi=mi-Mr
    ni=ni-Nr
    si=si-sr
    Grant(Ri).
else
    if (Sr > si) **then**
        Ask_Storage_Federation(Ci)
    end if
    if (Mr>mi || Nr> ni) **then**
        Ask_Compute_ Federation(Ci)
    end if
end if
Get Deallocation request from user Rj f(Mr,Nr,Sr)
  mi=mi+Mr
  ni=ni+Nr
  si=si+sr
  Free (Rj)

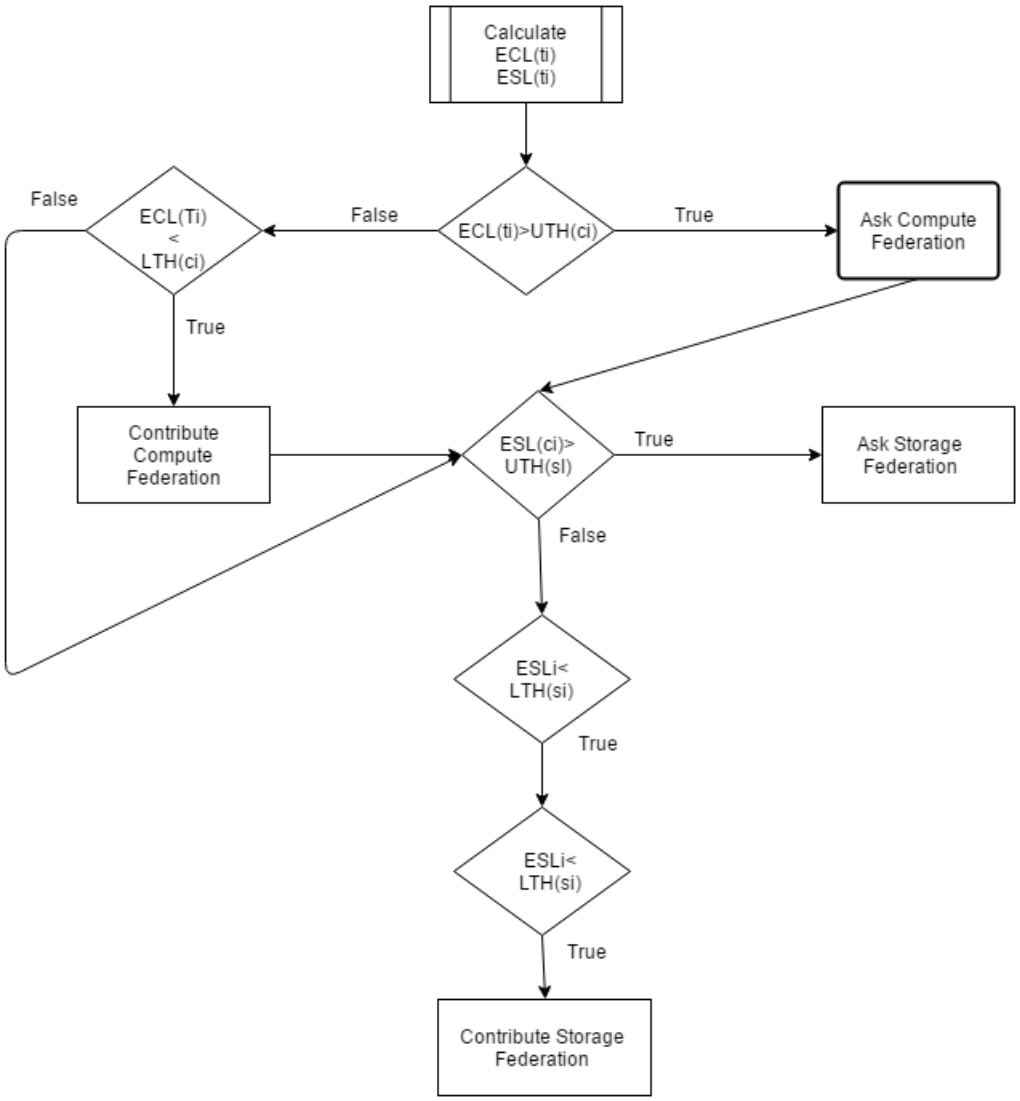

Figure 4.4: Flowchart of Algorithm 2

In figure 4.3 flowchart of algorithm for cloud agent is shown. In this algorithm user request for the resources like storage, memory and computation from cloud agent. Cloud agent checks if the requested resources are available. If yes, cloud agent grants the resources to the user else cloud agent request for the federated resources from the cloud federation mediator.

In figure 4.4 flowchart of algorithm for prediction is shown. Here cloud agent calculates the current load and estimated load for all the resources. If the estimated load is above resource upper threshold, the process of federation of that resource is initiated. If the estimated load is below lower threshold, the process of donating that resource to cloud is initiated.

In algorithm 3 various functions of cloud federation mediator are mentioned.It includes

maintaining and managing the resources in federation. Cloud federation mediator maintains the information of all the resources, either donated or consumed, by clouds taking part in federation.

Algorithm 2 Prediction Algorithm

```
L(t)i = Max ( (Mi-mi)/Mi , (Ni-ni)/Ni)
SL(t)i= Si-si/Si
ECL(t)i= αECL(t-1) + (1-α)L(t)
ESL(t)i= β ESL(t-1) + (1-β) SL(t)
if ( ECL(t)i >UTHci) then
    Ask_Compute_Federation(Ci,c)
else
    if (ECL(t)i < LTHci) then
        Contribute_Compute_Federation(Ci,c)
    end if
end if
if (ESLi > UTHsi) then
    Ask_Storage_Federation.(Ci,s)
else
    if (ESLi< LTHsi) then
        Contribute_Storage_Federation(Ci,s)
    end if
end if
```

Algorithm 3 Cloud Federation Mediator

I ={C1,C2,...Cn} is Set of cloud providers in federation.
n=sizeof(I) is Total number of cloud in federation.
SClist ={Sc1, Sc2,...Scn} is list of cloud providers contributing storage in federation.
CClist = {CC1,CC2,....CCn} is list of cloud providers contributing computation in federation.
Rslist = {Rs1, Rs2,...Rsn} is list of cloud providers requesting storage from federation.
Rclist = {Rc1,Rc2,...Rcn} is list of cloud providers requesting computation from federation.
Rrsi= {Rrs1, Rrs2,...Rrsn} is amount of storage resource requested by cloud Rsi from Rslist
Rrci= {Rrc1,Rrc2,...Rrcn} is amount of compute resource requested by cloud Rci from RClist
Crs= {Crs1,Crs2,....Crsn} is amount srorage resource contributed by cloud Scifrom Sclist
Crc= {Crc1,Crc2,...Crcn} is amount of compute resource contributed by cloud from CClist.
Frs=Total storage resource of federation.
Frc= Total compute resource for federatioin

```
while (true) do
    For each cloud Ci from I
    if (Contribute_Storage_Federation(Ci,s)) then
        Add Ci in SClisti.
        Crsi=s
        Frs=Frs+Crsi
    end if
    if (Contribute_Compute_Federatioin(Ci,c)) then
        Add Ci in CClisti
        Crc=c
        Frc=Crc+c
    end if
    if (ASK_For_Storage Federation(ci,s)) then
        Add Ci in Rslisti
        Rrsi=s
        if (rrsi<frs) then
            Allocate resource from clouds from SClist.
            Frs=frs-rrsi
            Return(s)
        end if
    end if
    if (Ask_For_ComputeFederation(ci,c)) then
        Add Ci in Rclist.
        if (rrci<frc) then
            Allocate resource c from clouds from CClist.
            Frc=frc-rrci
            Return(c)
        end if
    end if
end while
```

Chapter 5

Testbed Using OpenSource Middleware

To execute the system, private cloud is formed using Open Source Middleware OpenNebula.Steps for the configuration of the cloud are as follows:

5.1 Installation in the frontend

1.Install the repo

2.Install the required packages

3.Configure and Start the Services

4.Configure NFS

5.Configure SSH Public Key

5.2 Installation in worker node

1.Install the repo

2.Install the required packages

3.Configure and Start the Services

4.Configure NFS

5.Configure Qemu Key

5.3 Snapshots

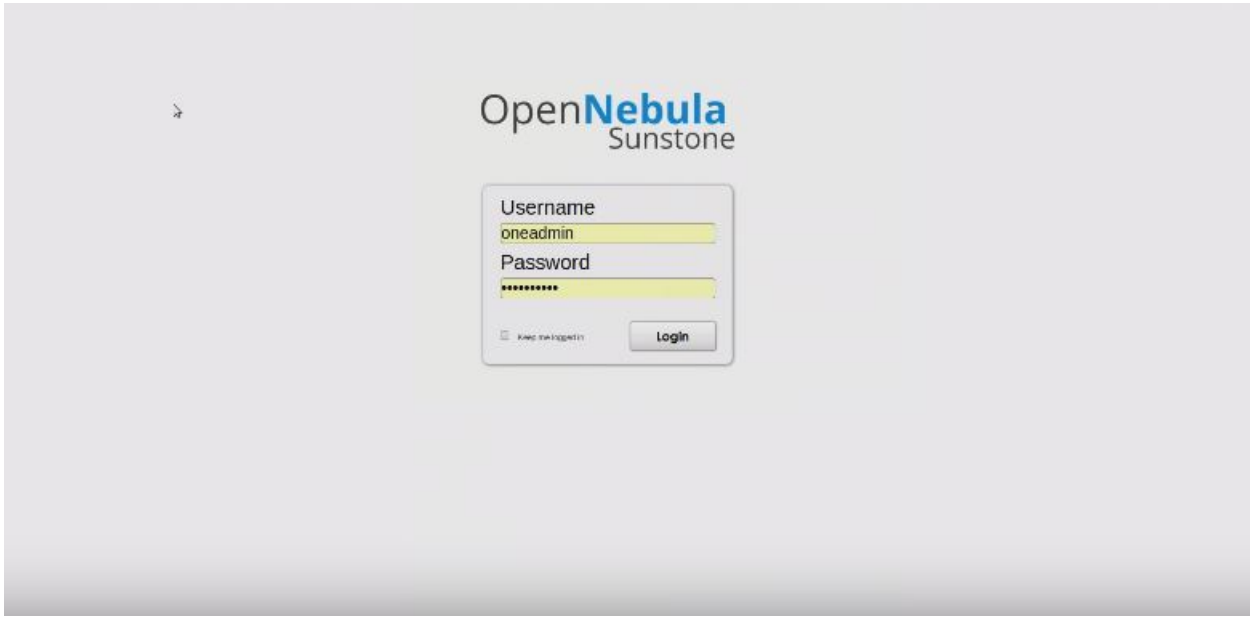

Figure 5.1: OpenNebula login

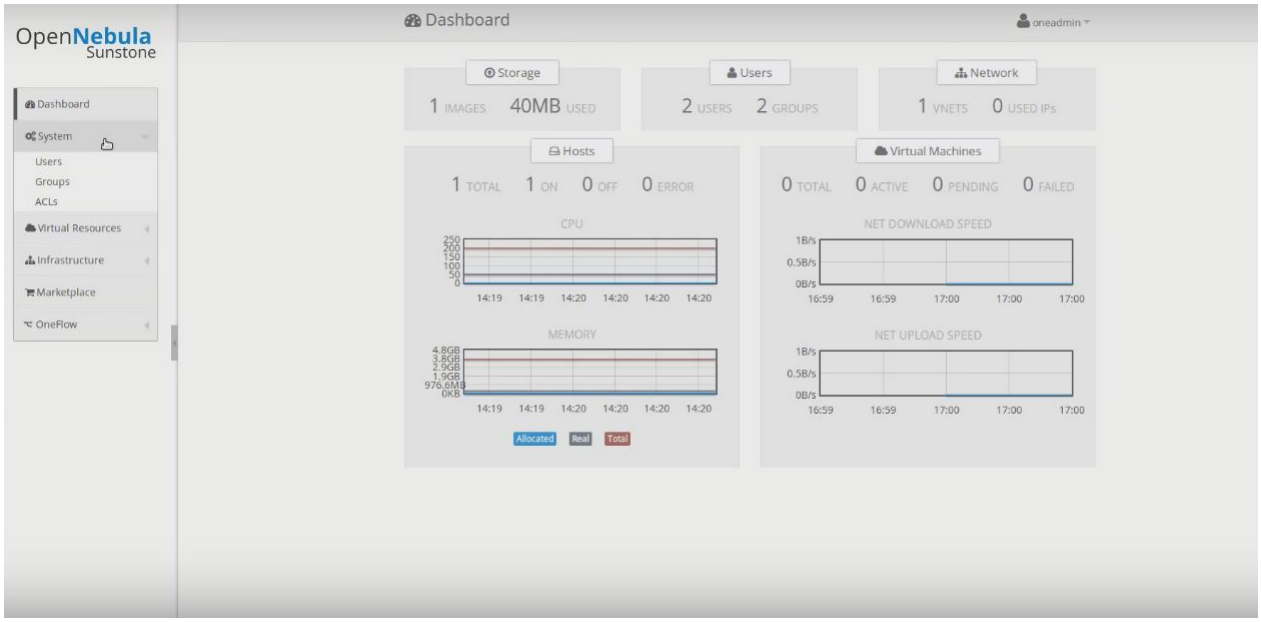

Figure 5.2: Sunstone Dashboard page

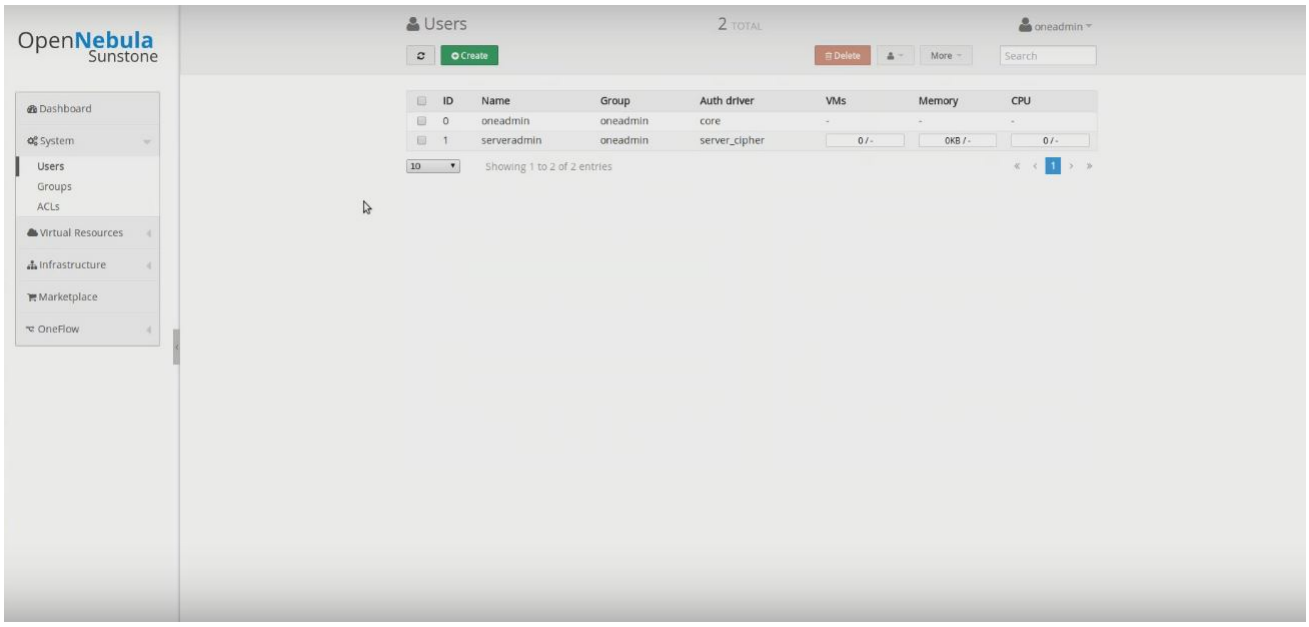

Figure 5.3: Sunstone User page

Chapter 6

Implementation

To check the results of our algorithm on cloud federation with large amount of resources we have implemented our algorithm in a simuleted environment. For this we have used CloudSim [9] simulation tool which is a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms

```
int hostId=0;
int ram = 40000; //host memory (MB)
long storage = 10000000; //host storage
int bw = 1000000;
```

Figure 6.1: Host Parameters

```
//VM Parameters
long size = 10000; //image size (MB)
int ram = 500; //vm memory (MB)
int mips = 100;
long bw = 1000;
int pesNumber = 1; //number of cpus
String vmm = "Xen"; //VMM name
```

Figure 6.2: VM Parameters

In figure 6.1 the host parameters are mentioned which in configured in each cloud and in figure 6.2 parameters of virtual machines to be created are mentioned.

6.1 Output

```
[Broker: Allocation of VM #351 to host #0 failed by RAM
Broker: Availabler RAm is 0And Current Requested Ram is 500which is more than tha available ram
Memory Federation required
```

Figure 6.3: VM failure without prediction

When there are no resources available for vm, creation of VM gets failed which is shown in figure 6.3. Here prediction algorithm is not applied.

```
36.40000000000024: Broker: VM #357 has been created in cloud #4, Host #0
36.40000000000024: Broker: VM #358 has been created in cloud #4, Host #0
36.40000000000024: Broker: VM #359 has been created in cloud #4, Host #0
```

Figure 6.4: VM implemented after federation

In figure 6.4 After failure of VM creation, Broker finds resources from another cloud in federation and VM is created in that cloud.

```
[Broker]:Here MemoryUtilizaion is 80.3030303030303%
Memory Utilization above threshold. Memory Federation Predicted
Memory after prediction67000
Memory taken from cloud3
```

Figure 6.5: Prediciton Output

In figure 6.5 output of oure prediction algorithm is shown. Here when the utilization of memory reaches above threshold, broker preditcs the requirement of federation and start the federation process before the next vm request.

In Figure 6.6 the time taken to create VMs in clouds with different capacities are shown. Here we have tried to create VMs by both with and without prediction process.

6.2 Result

Time Without Prediction.

RAM/VM	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160
30000	0.2	0.2	1.299	2.3	4.4	6.39										
60000	0.2	0.2	0.2	0.2	1.299	2.3	3.3	4.3	6.39	8.39	10.39	12.39				
90000	0.2	0.2	0.2	0.2	0.2	0.2	1.29	2.3	3.3	4.3	5.29	6.29	8.39	10.39	12.39	14.39
120000	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	1.3	2.3	3.3	4.3	5.29	6.29	7.3	8.3
150000	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	1.29	2.3	3.3	4.3	5.3	6.3
180000	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	1.29	2.29	3.3	4.3

Time With Prediciton

RAM/VM	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160
30000	0.59	1.01	1.4	2.1	2.99	3.89										
60000	0.59	1.01	1.4	1.8	2.2	2.6	3.1	4	4.89	5.79	6.69	7.59				
90000	0.59	1.01	1.4	1.8	2.2	2.6	3	3.4	3.8	4.2	5.1	5.99	6.89	7.79	8.7	9.59
120000	0.59	1	1.4	1.8	2.2	2.6	3	3.4	3.8	4.2	4.6	5	5.4	6.1	7	7.89
150000	0.59	1	1.4	1.8	2.2	2.6	3	3.4	3.8	4.2	4.6	5	5.4	5.8	6.2	6.6
180000	0.59	1	1.4	1.8	2.2	2.6	3	3.4	3.8	4.2	4.6	5	5.4	5.8	6.2	6.6

Figure 6.6: Readings

In figures 6.7 to 6.12 time taken to create VMs, with and without prediction, are shown in different cloud environments.

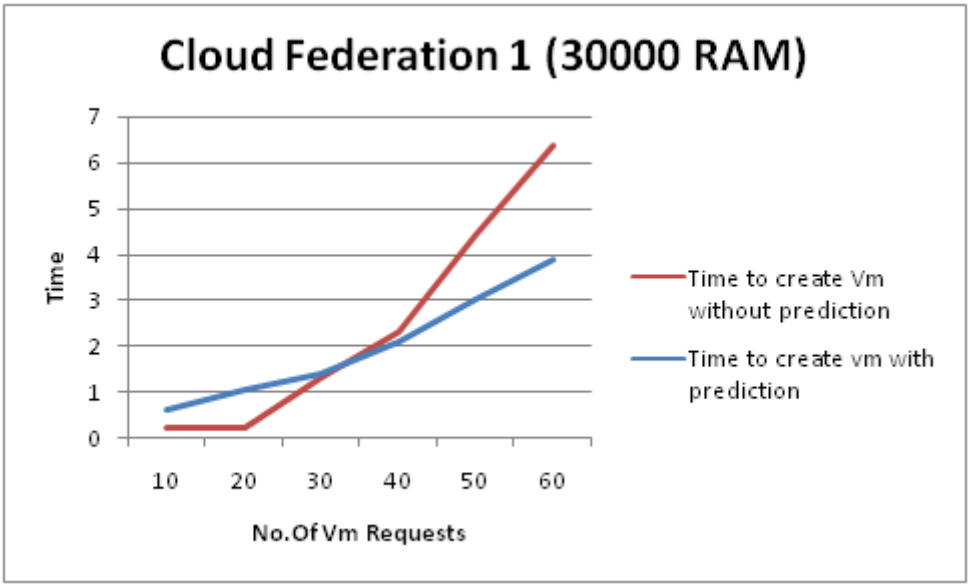

Figure 6.7: Cloud Federation with 30000 RAM

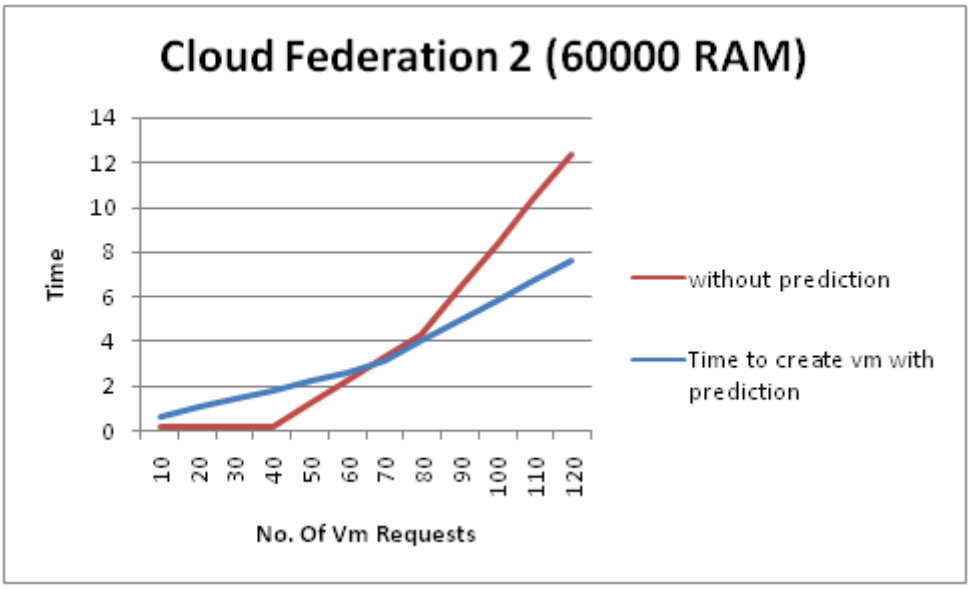

Figure 6.8: Cloud Federation with 60000 RAM

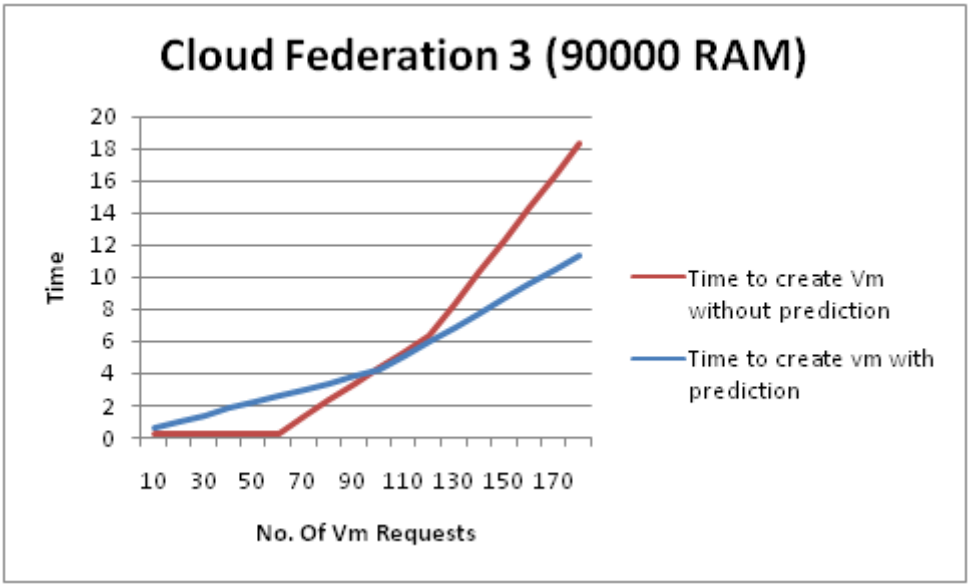

Figure 6.9: Cloud Federation with 90000 RAM

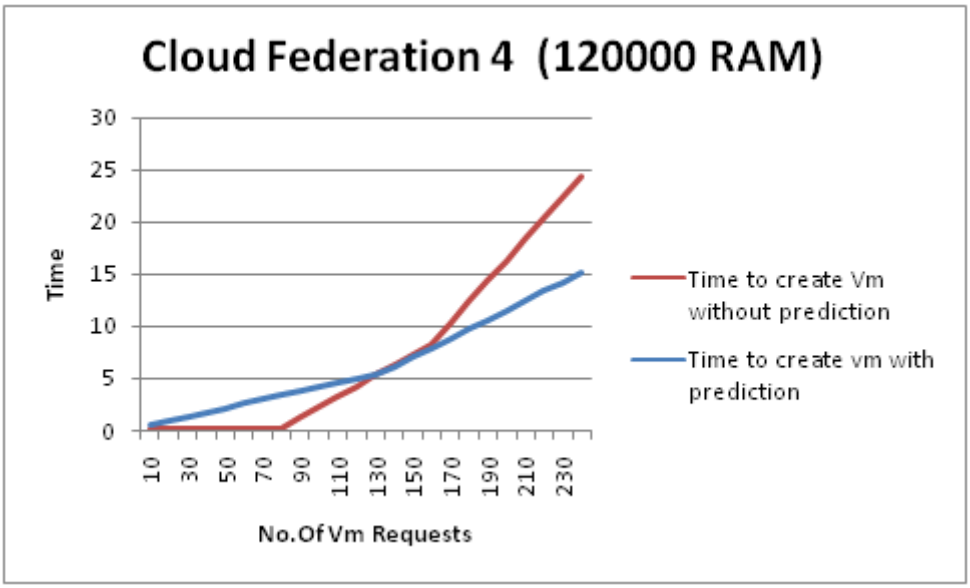

Figure 6.10: Cloud Federation with 120000 RAM

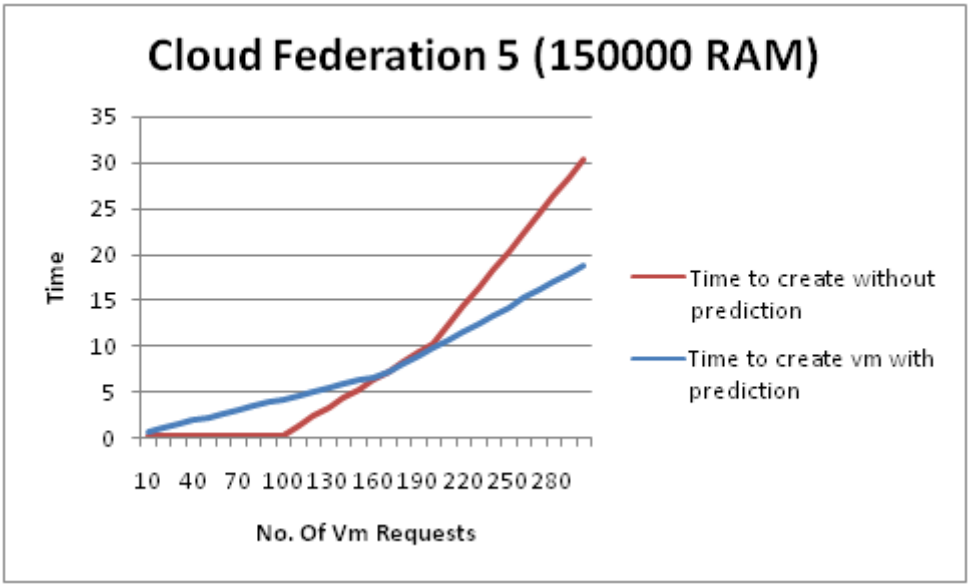

Figure 6.11: Cloud Federation with 150000 RAM

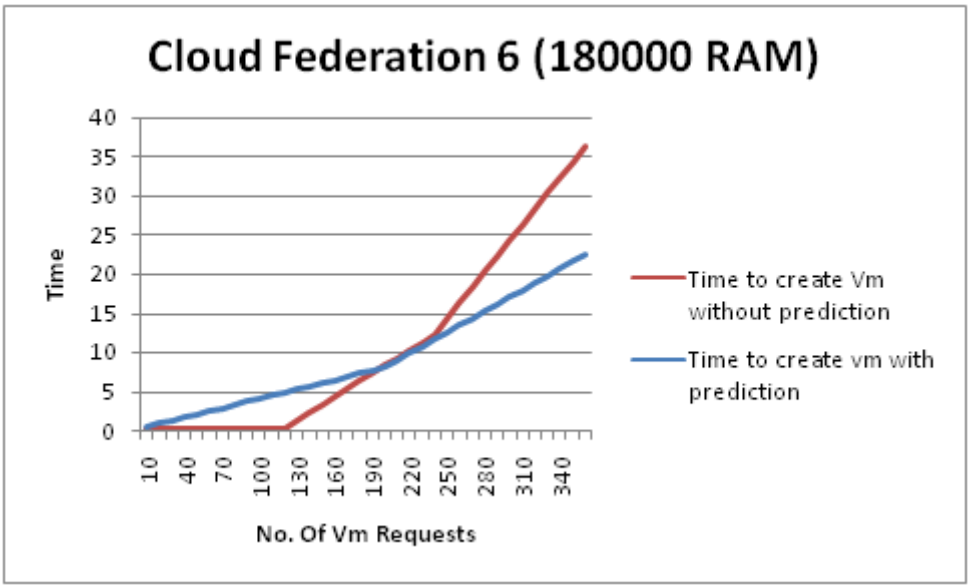

Figure 6.12: Cloud Federation with 180000 RAM

Chapter 7

Conclusion and future work

7.1 Conclusion

Cloud Federation is a concept which takes the cloud computing technology to next step. Service providers with comparatively small infrastructure. availability can still provide seamless cloud services by creating federation. In this research work, we developed a prediction algorithm to find the requirement of federation so that the process of creating federation can be started even before the request arrives. According to our result, our algorithm works efficiently in cloud environment where chances of VM failures are more.

7.2 Future work

Develop an efficient algorithm to predict federation with more parameters

Apply various machine learning concepts to predict federation

Develop prediction algorithm for heterogeneous cloud.

Bibliography

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, no. 13, p. 2009, 2009.

[2] R. Kanday, "A survey on cloud computing security," in *Computing Sciences (ICCS), 2012 International Conference on*, pp. 302–311, IEEE, 2012.

[3] D. Milojičić, I. M. Llorente, and R. S. Montero, "Opennebula: A cloud management tool," *IEEE Internet Computing*, no. 2, pp. 11–14, 2011.

[4] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on*, pp. 124–131, IEEE, 2009.

[5] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "Openstack: toward an open-source solution for cloud computing," *International Journal of Computer Applications*, vol. 55, no. 3, 2012.

[6] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," *CLOUD COMPUTING*, vol. 2011, pp. 32–38, 2011.

[7] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. M. Sadjadi, and M. Parashar, "Cloud federation in a layered service model," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330–1344, 2012.

[8] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014.

[9] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*, pp. 1–11, IEEE, 2009.