Semantic Enabled Cloud for Meteorological Data Processing Applications

Submitted By

Jimesh Rakesh Prajapati

13MCEN28

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Semantic Enabled Cloud for Meteorological Data Processing Applications

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Jimesh Rakesh Prajapati

(13MCEN28)

Guided By

Dr. Madhuri Bhavsar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled **"Semantic Enabled Cloud for Meteorological Data Processing Applications"** submitted by **Jimesh Rakesh Prajapati (Roll No: 13MCEN28)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering(Networking Technologies) of Institute of Technology, 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 project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Gaurang Raval
Associate Professor,
Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Associate Professor and Section Head,
Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

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

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Jimesh Rakesh Prajapati**, Roll. No. **13MCEN28**, give undertaking that the Major Project entitled "**Semantic Enabled Cloud for Meteorological Data Processing Applications**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering(Networking Technologies)** 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 my guide **Dr. Madhuri Bhavsar**, Sr.Associate Professor and Section Head, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her 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. K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would like to thank **Mr.Nitant Dube** for giving us their precious time and valuable guidance for this Project. Without them it would not have been possible and thank you to my Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Jimesh Rakesh Prajapati
13MCEN28

Abstract

Meteorological data is being captured regularly by satellites. It can be utilized in multiple ways to obtain useful predictions and meaningful information. This domain contains many applications which involves huge amount of image processing which are extremely compute intensive. To process and execute this type of high computational tasks,enormous amount of resources is needed. For such scenario, Cloud Computing is a better and smart approach for executing this tasks. As current Cloud Service Providers (CSP's) deliver resources in optimal way by using various scheduling techniques but also with manual intervention for selecting required resources, which degrades the resource usage performance. An approach is proposed to deploy this compute intensive tasks over Cloud, but with intelligent decision making capability as managing the resources productively over the Cloud is challenging. This type of cloud which is proposed with advanced and smart decision making ability is called '***Semantic Cloud***' which manages the semantic allocation of resources for adaptive formation of virtual machines(VM's) to deploy applications, as the automation has been needed for making cloud resources utilization potent.

Abbreviations

SLN	Similarity Link Network.
ALN	Association Link Network.
OWL-S	Ontology Web Language for Semantic Web.
RDF	Resource Description Format
CSP	Cloud Service Provider
QOS	Quality of Service
SLA	Service Level Agreement

–

Contents

List of Figures

Listings

Chapter 1

Introduction

Weather Forecasting or Climate estimating is one application in which it has discovered robust ground. This field is indeed boosting the importance of the cloud like never before, as time-discriminating information can be prepared and circulated routinely. In the meantime, precise climate estimation is turning into the standard all through the world as an after effect of cloud-driven activity. This type of tasks needs large quantity of resources like memory, storage, and processing power specially. Handling of the large resources is also a quite difficult task. Cloud Computing is the solution for the effective management and utilization of resources,which schedules the required resources. But in the world of automation where the decision making is performed by machines only ,why not to think of this automation over a Cloud Computing? where the resource allocation is still manual ,traditionally using menu interface,which results in inefficient consumption of resources allocated for various compute intensive tasks. For optimum resource management and for high availability and high performance, Semantic Cloud comes in to the light with this smart decision making capability.This Semantic Cloud service will provide Platform as a Service (PAAS) to the users to deploy the various image processing applications. A number of approaches are discussed below for building the semantic cloud but ontology is preferred for this project.

1.1 Background

Cloud Computing,the fantasy of processing as an utility, is a movement of calculation. We wont process on our PCs, yet move the project and information to the Cloud comprising of computation and storage utilities gave by outsiders ,which is not a new thought. At the point when time-offering frameworks were acquainted with the computer industry, individuals began to utilize figuring technology like Cloud.

The components involved in this project are as follows :

1.1.1 Image Processing

In imaging science, image processing is any type of signal processing for which the input is an image or picture , for example, a photo or feature outline; the output of image processing may be either an image or a set of qualities or parameters identified with the image. Most picture transforming techniques include treating the picture as a two-dimensional signal and applying standard signal-processing procedures to it.[1].

1.1.2 Weather Forecasting

Weather forecasting is the application of science and innovation to anticipate the condition of the atmosphere. Individuals have endeavored to predict the climate casually for centuries, and formally since the nineteenth century. Climate forecasts are made by gathering quantitative information about the current condition of the atmosphere and utilizing exploratory understanding of climatic methods to extend how the atmosphere will develop on that place[2].

1.1.3 Semantic Cloud

Cloud Computing is considered as a standout amongst the most guaranteeing processing ideal models. Most effectively existing frameworks, which give distributed computing, just give careful consideration to aggregating resources with the same processing capacity and give client nearby streamlined and discontinuous results furthermore a key obstruction that keeps associations from effectively managing services on the cloud is the absence of an incorporated methodology for adaptive service creation and deployment that gives an all encompassing perspective of the automate benefits on a Cloud. Semantic cloud means the cloud with ability to take decisions on its own giving high performance and high availability. To fulfill clients complex requirements its important to arrange resources with diverse capacities together and produce the customized and adaptive resource stream to clients. In order to do that, Semantic cloud is one of the best options for above scenario.[3].

1.2 Motivation for this Project

Dedicating top-line servers for executing experimental applications that run discontinuously, for example, serious climate recognition or generalized weather forecasting, wastes resources.The infrastructure-as-a-service (IaaS) model utilized by today's cloud platforms is appropriate for the bursty computational requests of these applications. Clouds are rising as the essential host platform for variety of applications, for example, Dropbox, icloud, and Google Music. These applications let clients store information in the cloud and access it from any place on the world. Business Clouds are additionally appropriate for leasing top of the high-end servers to execute applications that require computation resources sporadically.

Now a days cloud computing with advanced automation is more in demand ,where large amount of computational tasks are deployed over a cloud and optimal resource allocation is needed to be done without manual intention. One of such large computational tasks are satellite image or meteorological image processing tasks which requires more number of resources. Cloud is the best option for this tasks but managing the resources optimally is quite grueling job. For the optimum management of resources semantic cloud is proposed for deploying the meteorological image processing applications.

1.3 Objective of Study

The main objective behind this project is handling of large computational demands coming from scientific applications like weather predication. The Ontology based semantic cloud IS build for providing a user, the independence to deploy the image processing applications over the semantic cloud without manual intention. The proposed ontology scheme will decide the resources to be allocated using the training data-sets and generate virtual machines adaptively for deploying the applications. Users can access the results of this applications deployed over a cloud which are conserved on data-stores in a required format.

1.4 Scope of Work

The desired data set of images and the application created by users, Semantic enabled cloud service chooses the resources automatically required to form a virtual Machine for deployment of application in a optimum environment,and efficiently using the resources with high computation power. Cloud Resources, for example, virtual machines and progressed computing resources are utilized to deliver day by day applications like climate forecasts and help making models for forecasting weather in the area,etc. The result is estimates of storms, air quality, and issues identified with calamities. While such figures can affect a great many individuals, the same engineering gives comparative advantages to associations.

1.5 Outline of Thesis

This thesis will address the different tasks like Literature Survey, Proposed Architecture, Algorithms, Configuring the Cloud, Design Flow, and Finally Conclusion and Future Scope.

1. Chapter-2 reviews the Literature Survey done for creating Semantic Cloud using different approaches.

2. Chapter-3 shows proposed approach and Architecture for Semantic Cloud. Ontology and its different Phases and QOS parameters are also discussed.

3. Chapter-4 give details about proposed algorithms for implementing different ontology phases and VM creation and allocation over cloud.

4. Chapter-5 describes the procedure, showing the steps to configure open Nebula cloud.

5. Chapter-6 consists of detail design flow for the Web Utility created for accessing semantic cloud services. and snapshots.

6. Chapter-7 gives the conclusion and future scope of the proposed architecture.

Chapter 2

Literature Survey

2.1 General

In literature survey papers defining different approaches for building Semantic Cloud are referred for getting ideas to implement the appropriate methodology over Semantic Cloud.

There are many approaches for making cloud semantic:

2.1.1 Based on SLN and ALN

One of the methodologies is offered, in which a semantic layer for cloud computing is fabricate, which has a worldwide perspective of resources by the semantic representation and the semantic relationship. It concentrates on the similarity and association relation of resources, whereupon Similarity Link Network (SLN) and Association Link Network (ALN) are constructed to sort out resources and structure semantic cloud, i.e., SLN-Clouds and ALN-Clouds. Like the current cloud computing, SLN Clouds can aggregate resources of the same kind to create bound together service. Affiliation Link Network (ALN) is the extension over various types of resources, which has the capacity connection cross-domain resources [4].

2.1.2 Using semantic enabled search engine as a Middleware

Another approach for semantic cloud is Semantic enabled search engine for cloud applications. This Semantic Engine can act as a Middleware, which helps the cloud applications developers to find the appropriate API Functionalities and other Resources automatically. As Cloud based programming applications are as of now created without appreciating a typical programming model, open standard interfaces, sufficient administration level specification facilities and convenience characteristics. In this situation, vendors give diverse Cloud services at distinctive levels normally giving exclusive Application Program Interfaces (Api's) to application designers. In this way making troublesome or difficult to demand cloud administrations gave by diverse vendors, since this would require utilizing distinctive interfaces, languages and API's - the so called "Cloud vendor lock-in" issue. This issue is especially genuine in the current situation since it is presently hard to discover suppliers which completely address all clients' necessities, therefore making interoperability among services provided by different vendors an exceptionally attractive feature.For such sort of situation the mosaic Project, an EU subsidized Research Project in the Internet of Service and Virtualization region, addresses these issues by characterizing a integrated framework for getting to distinctive cloud supplier's resources and services. [5].

2.1.3 OWL-S based semantic cloud

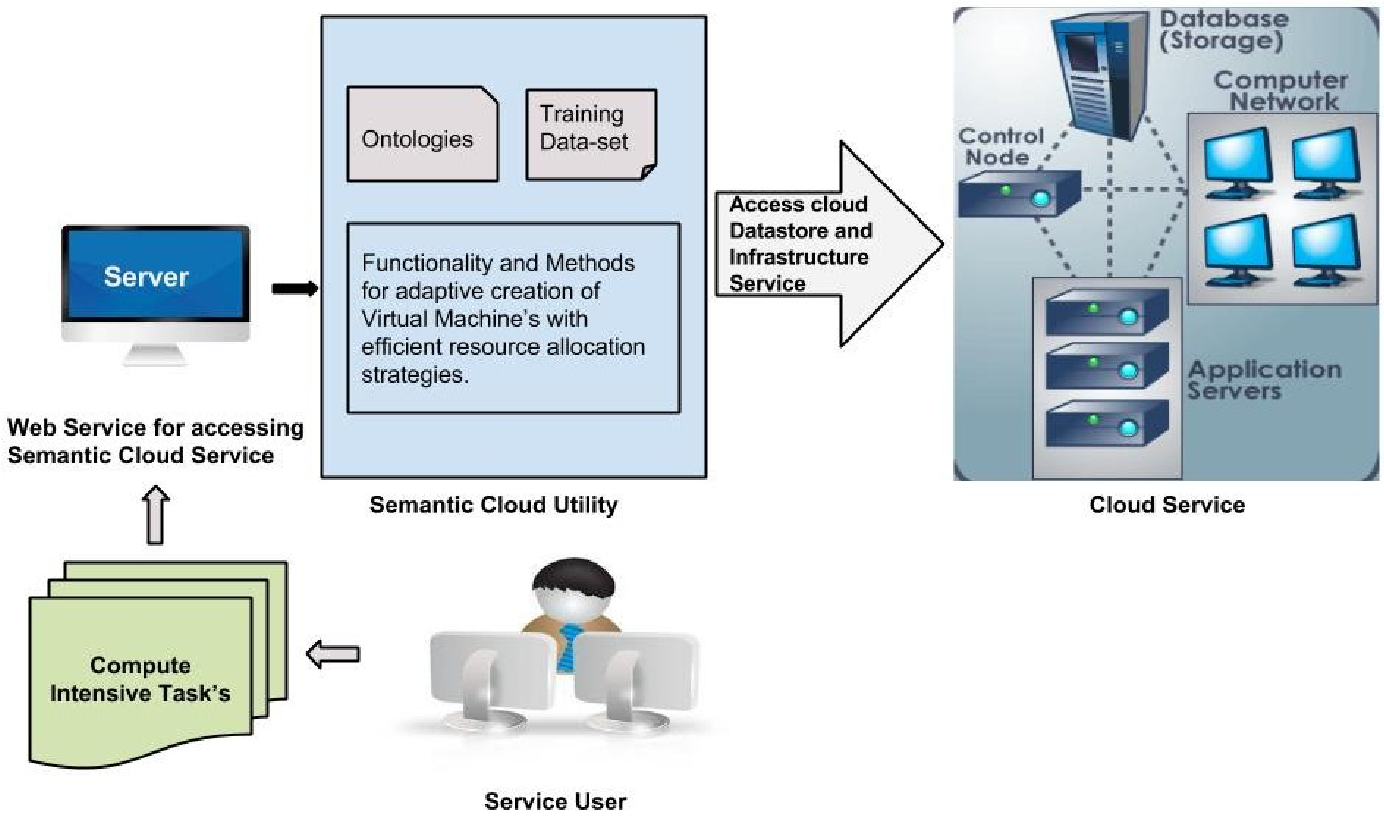
Utilizing Cloud Broker is additionally an alternate methodology for semantic cloud procurement, which deals with the utilization, execution and delivery of cloud services and arranges connections between cloud providers and cloud consumers. In real life scenarios, automated cloud services handling is regularly difficult because of the fact that the service descriptions may include complex constraints and require adaptable semantic matching. Besides, cloud providers frequently utilize non-standard formats prompting semantic interoperability issues. In [6], a cloud service brokering under a service oriented framework, and a novel OWL-S (Ontology) based semantic cloud service discovery and selection system is proposed. This proposed system supports dynamic semantic matching of cloud services described with complex constraints.

Finally the approach given in [7], which implements the automation of resource flow for IT life cycle services using ontology on semantic web. This methodology utilizes the past work on ontologyś similar to OWL-S for service descriptions that concentrates on automating processes expected to acquire benefits on the cloud.

Out of the above discussed approaches for semantic cloud, our proposed approach is quite different as it is focusing more on the intelligent decision making capability of cloud for optimized resource allocation by creating virtual machines. Ontology is one of a good approach for automation of services over cloud[7].Also, using RDF for generating Ontology over semantic web service[8],[9] will help acquiring the automate resource provisioning strategies.

Chapter 3

Proposed Architecture



Ontologies
Training Data-set
Functionality and Methods for adaptive creation of Virtual Machine's with efficient resource allocation strategies.
Semantic Cloud Utility
Server
Web Service for accessing Semantic Cloud Service
Access cloud Datastore and Infrastructure Service
Database (Storage)
Computer Network
Control Node
Application Servers
Cloud Service
Compute Intensive Task's
Service User

Figure 3.1: Proposed Architecture

3.1 Proposed Approach

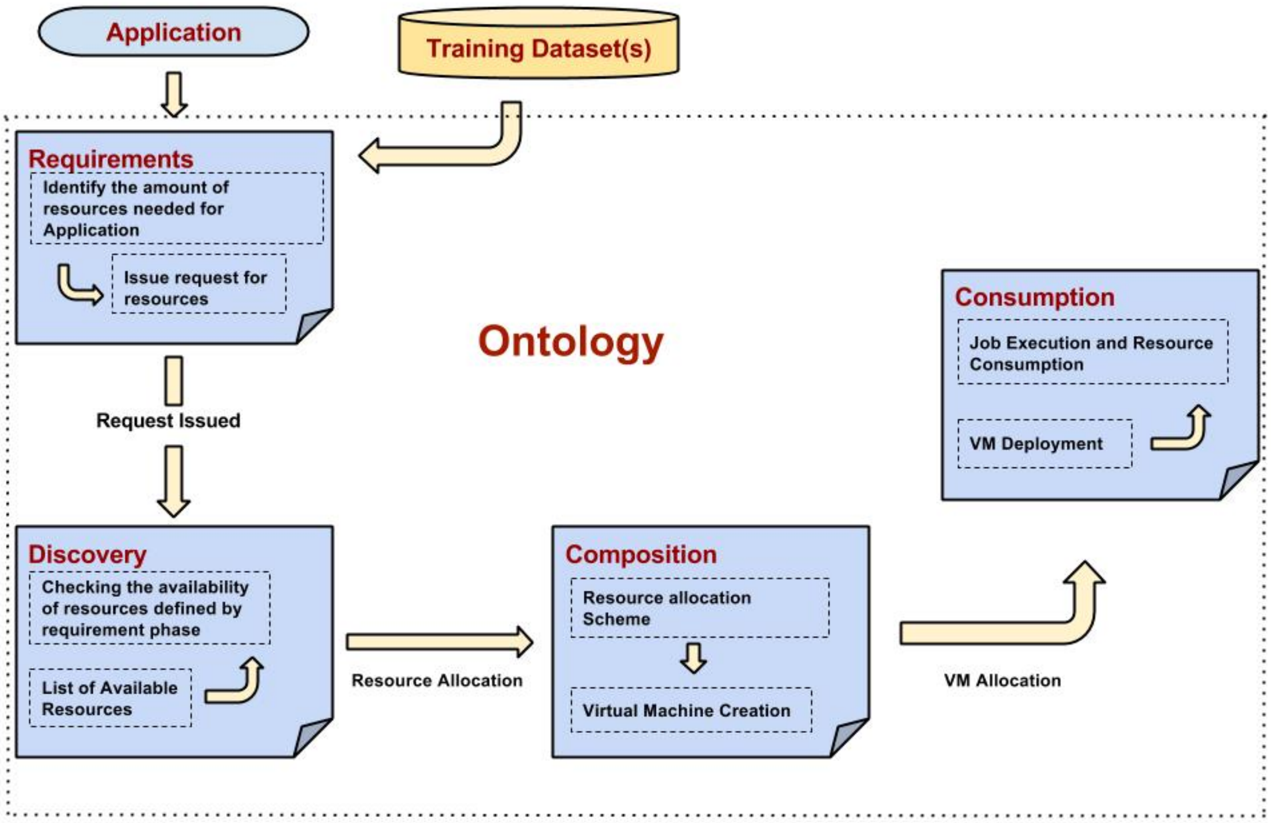
In order to deploy the Meteorological data applications for Image Processing over a Semantic Cloud service, Framework for dynamic application creation is also proposed where the users will formulate different type of applications, further , which will be deployed on Virtual Machines over a cloud.

1. The Users will create a variety of applications for exploration of satellite images using the Framework proposed for creating various Image Processing applications.
2. After the creation, one can deploy their application over a Semantic Cloud service, which will adaptively estimate the need of application like **Amount** of satellite data to be processed, **Number** of applications whether single or multiple to be uploaded. **Type** of Applications, e.g Cloud Masking, Land Temperature, etc. From this parameters the resource allocation strategy will be decided using the predefined set of ontology.
3. This Ontology will determine and allocate resources needed by the applications to formulate the Virtual Machines (VMs).

3.1.1 Proposed Ontology Flow

Ontology

In Computer Science, an Ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse or in short, Ontology is a set of rules or semantics that represent some kind of knowledge for specific purpose.



Application
Training Dataset(s)
Requirements
Identify the amount of resources needed for Application
Issue request for resources
Ontology
Request Issued
Consumption
Job Execution and Resource Consumption
VM Deployment
Discovery
Checking the availability of resources defined by requirement phase
List of Available Resources
Resource Allocation
Composition
Resource allocation Scheme
Virtual Machine Creation
VM Allocation

Figure 3.2: Ontology Flow

There will be different phases in ontology:

(a) **Service requirements**

In the service requirement phase the amount of resources needed by the applications created by the users will be known using the training datasets(s). The functional and technical specification's that applications needs to be fulfilled.

(b) **Service Discovery**

In service discovery phase the ontology will dredge up the resources defined by the service requirement phase.

(c) **Service composition**

In this phase different resources provided by semantic cloud are combined and delivered to the users application.

(d) **Service consumption**

After the resources allocation for applications, the user then begins consuming the resources and then after the execution, if not needed then resources is freed for other applications.

4. The Semantic Cloud service will act as Platform As a Service (PAAS) for the deployers of Image Processing applications as Semantic cloud allocates Virtual Machine(VMs) only for enormous computational tasks.

5. The Virtual Machine (VM) will be created automatically by the Service, for deploying the applications, on the basis of above defined ontology

6. VM can be scaled, migrated if necessary and destroyed after the execution of tasks.

3.2 Detailed Architecture

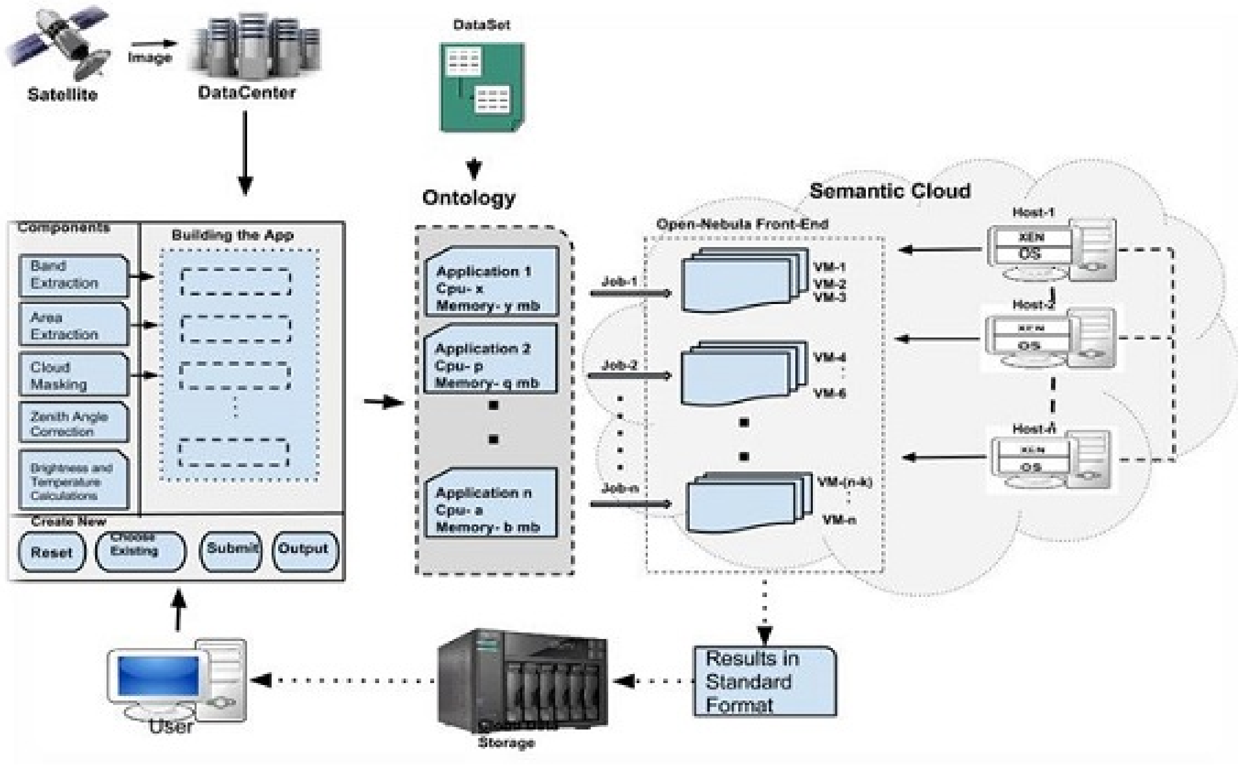


Figure 3.3: Detailed Architecture

3.3 Expected Outcomes

The results after the execution of jobs or applications on VM's over cloud can be stored on the cloud data-store according to the users specified format. VM's will be disposed after consumption and so the resources, giving efficient usage and management.

3.4 QoS Parameters

Quality of service (QoS) is the overall performance of any service and its network, particularly the performance seen by the users of the network.

To quantitatively measure quality of service in Semantic Cloud, several related aspects of the network of semantic cloud service are considered below.

Quality of Service provided by Semantic Cloud for dynamic image processing applications are follows:

1. Execution Time taken by Application for given computational tasks.

2. Response Time taken by Semantic Cloud.

3. Amount of Resources consumed by applications.

All QoS parameters will be compared for Application deployed on Single Host, on Cloud with manual Virtual Machine allocation and on Semantic Cloud.

On the basis of this QoS parameters, the predefined ontology will update its requirements for corresponding application type, achieving the efficient resource usage performance.

Chapter 4

Algorithms

4.1 Predefined Ontologyś and Algorithms for Proposed Ontology Flow

This chapter talks about the Parameters provided by the Incoming Jobs, Predefined Ontologyś and Algorithms used for adaptive allocation and Formation of Virtual Machines.
Parameters Provided by Applications:
The following parameters will be provided by the Application, that is generated by the Framework for creating Image Processing Applications.

Meta-Data from Incoming Jobs:

1. **Job-Type:** Cloud Masking, Land Temperature, etc.
2. **Input Data-Size:** In terms of MB/GB.
3. **No of Applications:** Single or Multiple.
4. **Processing Type of Job:** Sequential or Parallel

4.1.1 Pre-Defined set of Ontologyś during different phases

The Predefined Ontology will be used for resource estimation and decision making for resource allocation, defining how much amount of resources should be allocated to jobs. The parameters that are used in different ontologyś are:

1. **Resource Requirement:**

   (a) **Operating System Image Type:** Defines Which Type of OS with required tools and APIś can or will be required for Application.

   (b) **CPU:**Amount of CPU Cycles needed for Processing.

   (c) **Memory:** Memory required for running the JOBś

   (d) **Consumption_Type:** Defines type of consumption rate.

   (e) **Network:** Type of Network needed.

   (f) **Host:** Type of Host with required resource capacity and OS Image.

2. **Resource Description:**

   (a) **Operating System Images:** List of OS Images acquired by Cloud Service.

   (b) **Networks:** List of Networks defined within the Cloud, configured with different IPś.

   (c) **Hosts:** Total no. of hosts in Cloud Infrastructure.

3. **Resource Allocation:**

   This data-set consists different requirements for the different type of applications.

4. **Application Types:**

   This ontology defines different types of applications need.

   (a) **Data-Size:** Size of Satellite Image data to be computed.

   (b) **No. of Applications:** Defines no of applications user want to run.

   (c) **Type of Processing:** Sequential or Parallel.

In order to form an Ontology, JENA (`https://jena.apache.org/`) Framework is used. An RDF(Resource Description Format)[10] is used to define ontologyś for different phases.This predefined ontologyś will help in making, efficient resource allocation scheme based on the parameters like Application size,Input no. of satellite imageś etc.

Using RDF for Ontology has an advantage, Cloud Service can access or load this predefined set of ontology models using the URL defined for specific ontology. Using URL, Cloud Service will retrieve all the necessary parameters and constraints needed for resource estimation and allocation.

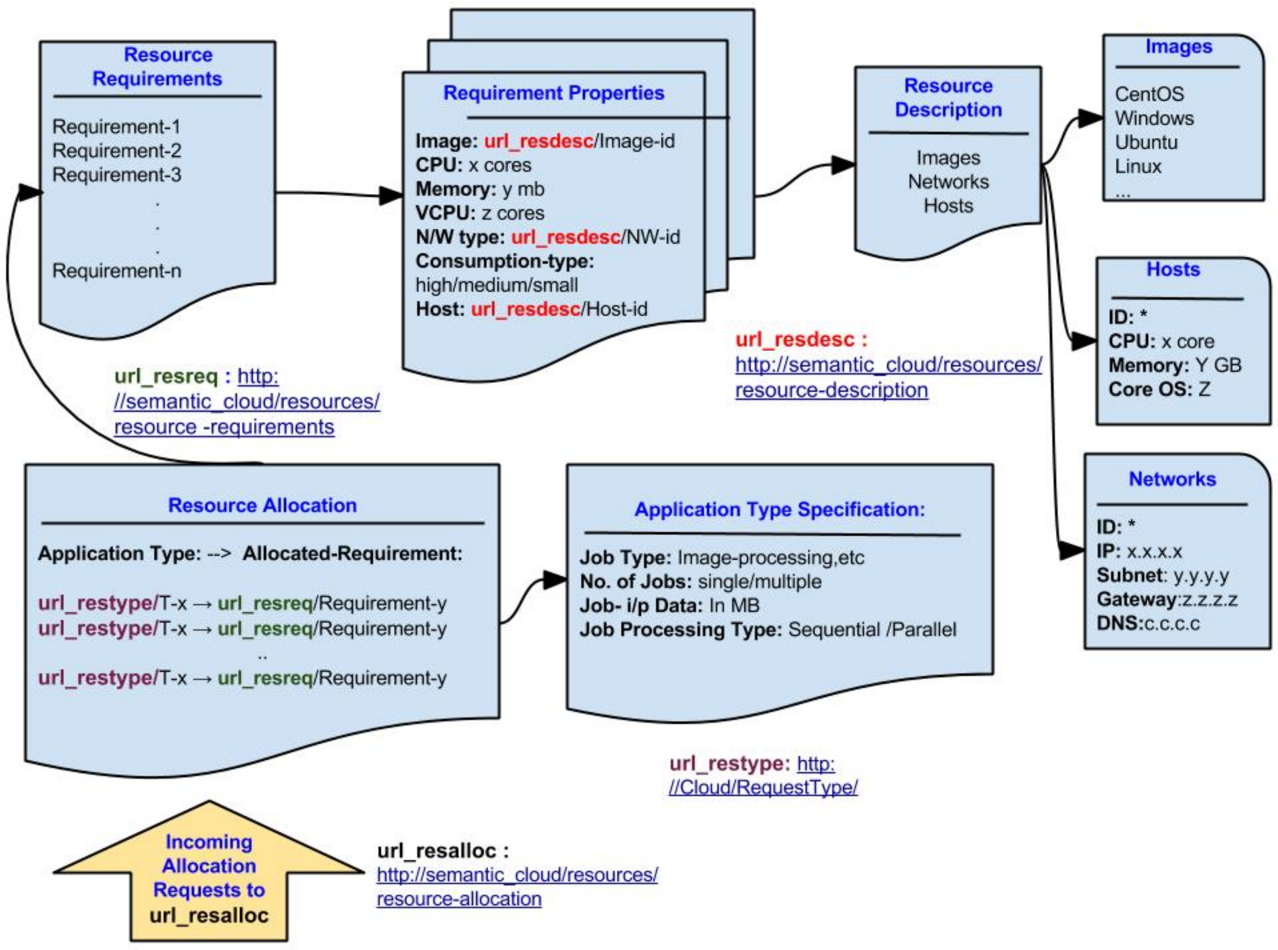

Resource Requirements
Requirement-1
Requirement-2
Requirement-3
Requirement-n
Requirement Properties
Image: url_resdesc/Image-id
CPU: x cores
Memory: y mb
VCPU: z cores
N/W type: url_resdesc/NW-id
Consumption-type: high/medium/small
Host: url_resdesc/Host-id
Resource Description
Images
Networks
Hosts
Images
CentOS
Windows
Ubuntu
Linux
Hosts
ID: *
CPU: x core
Memory: Y GB
Core OS: Z
Networks
ID: *
IP: x.x.x.x
Subnet: y.y.y.y
Gateway:z.z.z.z
DNS:c.c.c.c
url_resdesc : http://semantic_cloud/resources/resource-description
url_resreq : http://semantic_cloud/resources/resource -requirements
Resource Allocation
Application Type: --> Allocated-Requirement:
url_restype/T-x → url_resreq/Requirement-y
url_restype/T-x → url_resreq/Requirement-y
url_restype/T-x → url_resreq/Requirement-y
Application Type Specification:
Job Type: Image-processing,etc
No. of Jobs: single/multiple
Job- i/p Data: In MB
Job Processing Type: Sequential /Parallel
url_restype: http://Cloud/RequestType/
Incoming Allocation Requests to url_resalloc
url_resalloc : http://semantic_cloud/resources/resource-allocation

Figure 4.1: Detailed Ontology Flow

1. **Ontology for Requirementś Phase**

   This phase keeps different type of requirements that will be needed by different applications with specified parameters like uploaded data-size, expected response time, etc.

Listing 4.1: Resource Requirement

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
<rdf:Description rdf:about="http://Cloud/
   Resource_Requirement">
    <vcard:CLASS>Cloud_Requirements</vcard:CLASS>
    <vcard:GROUP rdf:resource="http://Cloud/
```

```
        Resource_Requirement/R1"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R2"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R3"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R4"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R5"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R6"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R7"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Requirement/R8"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/
      Resource_Requirement/R1">
    <vcard:KEY>Image:1</vcard:KEY>
    <vcard:KEY>CPU:90</vcard:KEY>
    <vcard:KEY>VCPU:180</vcard:KEY>
    <vcard:KEY>Memory:900</vcard:KEY>
    <vcard:KEY>Consumption_type:Small</vcard:KEY>
    <vcard:KEY>Network:VN2</vcard:KEY>
    <vcard:KEY>Host:1</vcard:KEY>
  </rdf:Description>
<rdf:Description rdf:about="http://Cloud/
    Resource_Requirement/R5">
    <vcard:KEY>Image:1</vcard:KEY>
    <vcard:KEY>CPU:180</vcard:KEY>
    <vcard:KEY>VCPU:360</vcard:KEY>
    <vcard:KEY>Memory:1800</vcard:KEY>
```

```
      <vcard:KEY>Consumption_type:Loaded_Small</vcard:KEY>
      <vcard:KEY>Network:VN2</vcard:KEY>
      <vcard:KEY>Host:1</vcard:KEY>
   </rdf:Description>
   <rdf:Description rdf:about="http://Cloud/
        Resource_Requirement/R2">
      <vcard:KEY>Image:1</vcard:KEY>
      <vcard:KEY>CPU:200</vcard:KEY>
      <vcard:KEY>VCPU:380</vcard:KEY>
      <vcard:KEY>Memory:1900</vcard:KEY>
      <vcard:KEY>Consumption_type:Medium</vcard:KEY>
      <vcard:KEY>Network:VN1</vcard:KEY>
      <vcard:KEY>Host:2</vcard:KEY>
   </rdf:Description>
<rdf:Description rdf:about="http://Cloud/
     Resource_Requirement/R6">
      <vcard:KEY>Image:1</vcard:KEY>
      <vcard:KEY>CPU:380</vcard:KEY>
      <vcard:KEY>VCPU:760</vcard:KEY>
      <vcard:KEY>Memory:3800</vcard:KEY>
      <vcard:KEY>Consumption_type:Loaded_Medium</vcard:KEY>
      <vcard:KEY>Network:VN2</vcard:KEY>
      <vcard:KEY>Host:2</vcard:KEY>
   </rdf:Description>
   <rdf:Description rdf:about="http://Cloud/
        Resource_Requirement/R3">
      <vcard:KEY>Image:1</vcard:KEY>
      <vcard:KEY>CPU:340</vcard:KEY>
      <vcard:KEY>VCPU:680</vcard:KEY>
      <vcard:KEY>Memory:2900</vcard:KEY>
      <vcard:KEY>Consumption_type:Medium_High</vcard:KEY>
      <vcard:KEY>Network:VN1</vcard:KEY>
```

```
    <vcard:KEY>Host:3</vcard:KEY>
  </rdf:Description>
<rdf:Description rdf:about="http://Cloud/
   Resource_Requirement/R7">
    <vcard:KEY>Image:1</vcard:KEY>
    <vcard:KEY>CPU:680</vcard:KEY>
    <vcard:KEY>VCPU:1360</vcard:KEY>
    <vcard:KEY>Memory:5800</vcard:KEY>
    <vcard:KEY>Consumption_type:Loaded_Medium_High</vcard:
       KEY>
    <vcard:KEY>Network:VN2</vcard:KEY>
    <vcard:KEY>Host:3</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/
     Resource_Requirement/R4">
    <vcard:KEY>Image:1</vcard:KEY>
    <vcard:KEY>CPU:400</vcard:KEY>
    <vcard:KEY>VCPU:800</vcard:KEY>
    <vcard:KEY>Memory:4000</vcard:KEY>
    <vcard:KEY>Consumption_type:High</vcard:KEY>
    <vcard:KEY>Network:VN1</vcard:KEY>
    <vcard:KEY>Host:4</vcard:KEY>
  </rdf:Description>
<rdf:Description rdf:about="http://Cloud/
   Resource_Requirement/R8">
    <vcard:KEY>Image:1</vcard:KEY>
    <vcard:KEY>CPU:800</vcard:KEY>
    <vcard:KEY>VCPU:1600</vcard:KEY>
    <vcard:KEY>Memory:8000</vcard:KEY>
    <vcard:KEY>Consumption_type:Loaded_High</vcard:KEY>
    <vcard:KEY>Network:VN2</vcard:KEY>
    <vcard:KEY>Host:4</vcard:KEY>
  </rdf:Description>
 </rdf:RDF>
```

Another Ontology defining different types of Application requirements is built, which will be needed for selecting type of requirement for specified applications.

Listing 4.2: Application type specification

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
  <rdf:Description rdf:about="http://Cloud/Request_Type">
    <vcard:CLASS>Cloud_Request_Types</vcard:CLASS>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T1
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T2
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T3
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T4
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T5
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T6
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T7
       "/>
    <vcard:GROUP rdf:resource="http://Cloud/Request_Type/T8
       "/>
    </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T1">
    <vcard:KEY>Data_Size:12-530</vcard:KEY>
    <vcard:KEY>no_of_app:single</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T2">
    <vcard:KEY>Data_Size:12-530</vcard:KEY>
    <vcard:KEY>no_of_app:multiple</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T3">
    <vcard:KEY>Data_Size:531-2650</vcard:KEY>
    <vcard:KEY>no_of_app:single</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T4">
   <vcard:KEY>Data_Size:531-2650</vcard:KEY>
    <vcard:KEY>no_of_app:multiple</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T5">
   <vcard:KEY>Data_Size:2651-5300</vcard:KEY>
    <vcard:KEY>no_of_app:single</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T6">
   <vcard:KEY>Data_Size:2651-5300</vcard:KEY>
    <vcard:KEY>no_of_app:multiple</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T7">
   <vcard:KEY>Data_Size:5301-10600</vcard:KEY>
    <vcard:KEY>no_of_app:single</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/Request_Type/T8">
   <vcard:KEY>Data_Size:5301-10600</vcard:KEY>
    <vcard:KEY>no_of_app:multiple</vcard:KEY>
  </rdf:Description>
</rdf:RDF>
```

2. **Ontology for Discovery Phase**

   In this phase, availability of resources will be queried on the Pre-defined ontology containing the available resources.Efficient selection of the available resource will be made for service composition phase.

Listing 4.3: Resource Availability

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
<rdf:Description rdf:about="http://Cloud/
    Resource_Description">
     <vcard:CLASS>Cloud_Resources</vcard:CLASS>
     <vcard:CATEGORIES rdf:resource="http://Cloud/
         Resource_Description/Image"/>
<vcard:CATEGORIES rdf:resource="http://Cloud/
    Resource_Description/Hosts"/>
     <vcard:CATEGORIES rdf:resource="http://Cloud/
         Resource_Description/Networks"/>
   </rdf:Description>
<rdf:Description rdf:about="http://Cloud/
    Resource_Description/Hosts">
     <vcard:GROUP rdf:resource="http://Cloud/
         Resource_Description/Hosts/Host1"/>
     <vcard:GROUP rdf:resource="http://Cloud/
         Resource_Description/Hosts/Host2"/>
<vcard:GROUP rdf:resource="http://Cloud/Resource_Description
    /Hosts/Host3"/>
<vcard:GROUP rdf:resource="http://Cloud/Resource_Description
    /Hosts/Host4"/>
   </rdf:Description>
<rdf:Description rdf:about="http://Cloud/
    Resource_Description/Networks">
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Description/Networks/VN1"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Description/Networks/VN2"/>
  </rdf:Description>
 <rdf:Description rdf:about="http://Cloud/
    Resource_Description/Image">
    <vcard:KEY>4:Linux</vcard:KEY>
    <vcard:KEY>3:CentOS</vcard:KEY>
    <vcard:KEY>2:Ubuntu</vcard:KEY>
    <vcard:KEY>1:windows</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/
     Resource_Description/Networks/VN2">
    <vcard:KEY>DNS:10.1.19.28</vcard:KEY>
    <vcard:KEY>Gateway:10.1.19.27</vcard:KEY>
    <vcard:KEY>Subnet:255.255.0.0</vcard:KEY>
    <vcard:KEY>N/W:10.1.3.41</vcard:KEY>
  </rdf:Description>
  <rdf:Description rdf:about="http://Cloud/
     Resource_Description/Networks/VN1">
    <vcard:KEY>DNS:10.1.19.28</vcard:KEY>
    <vcard:KEY>Gateway:10.1.19.27</vcard:KEY>
    <vcard:KEY>Subnet:255.255.0.0</vcard:KEY>
    <vcard:KEY>N/W:10.1.3.42</vcard:KEY>
  </rdf:Description>
<rdf:Description
rdf:about="http://Cloud/Resource_Description/Hosts/Host4">
    <vcard:KEY>Memory:8000</vcard:KEY>
    <vcard:KEY>VCPU:1600</vcard:KEY>
    <vcard:KEY>CPU:800</vcard:KEY>
    <vcard:KEY>ID:4</vcard:KEY>
  </rdf:Description>
  <rdf:Description
 rdf:about="http://Cloud/Resource_Description/Hosts/Host3">
     <vcard:KEY>Memory:4000</vcard:KEY>
     <vcard:KEY>VCPU:960</vcard:KEY>
     <vcard:KEY>CPU:480</vcard:KEY>
     <vcard:KEY>ID:3</vcard:KEY>
  </rdf:Description>
<rdf:Description
 rdf:about="http://Cloud/Resource_Description/Hosts/Host2">
     <vcard:KEY>Memory:3800</vcard:KEY>
     <vcard:KEY>VCPU:760</vcard:KEY>
     <vcard:KEY>CPU:380</vcard:KEY>
     <vcard:KEY>ID:2</vcard:KEY>
  </rdf:Description>
  <rdf:Description  rdf:about="http://Cloud/
       Resource_Description/Hosts/Host1">
     <vcard:KEY>Memory:1860</vcard:KEY>
     <vcard:KEY>VCPU:360</vcard:KEY>
     <vcard:KEY>CPU:180</vcard:KEY>
     <vcard:KEY>ID:1</vcard:KEY>
  </rdf:Description>
</rdf:RDF>
```

3. **Ontology for Composition phase**

   From the above Ontologies, This ontology will select the requirements from ontology defined for Requirements on the bases of the type of Application and different specified requirements.

Listing 4.4: Resource Allocation

```
<rdf:RDF
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
<rdf:Description rdf:about="http://Cloud/Resource_Allocation
   ">
    <vcard:CLASS>Resource_Allocation</vcard:CLASS>
    <vcard:CATEGORIES rdf:resource="http://Cloud/
        Resource_Allocation/Temperature"/>
    <vcard:CATEGORIES rdf:resource="http://Cloud/
        Resource_Allocation/Cloud_Masking"/>
<vcard:CATEGORIES rdf:resource="http://Cloud/
    Resource_Allocation/Multiple"/>
  </rdf:Description>

<rdf:Description rdf:about="http://Cloud/Resource_Allocation
   /Cloud_Masking">
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Cloud_Masking/A1"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Cloud_Masking/A2"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Cloud_Masking/A3"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Cloud_Masking/A4"/>
  </rdf:Description>
<rdf:Description rdf:about="http://Cloud/Resource_Allocation
   /Temperature">
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Temperature/A1"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Temperature/A2"/>
    <vcard:GROUP rdf:resource="http://Cloud/
        Resource_Allocation/Temperature/A3"/>
    <vcard:GROUP rdf:resource="http://Cloud/
```

```
         Resource_Allocation/Temperature/A4"/>
   </rdf:Description>
<rdf:Description rdf:about="http://Cloud/Resource_Allocation
    /Multiple">
     <vcard:GROUP rdf:resource="http://Cloud/
         Resource_Allocation/Multiple/A1"/>
     <vcard:GROUP rdf:resource="http://Cloud/
         Resource_Allocation/Multiple/A2"/>
     <vcard:GROUP rdf:resource="http://Cloud/
         Resource_Allocation/Multiple/A3"/>
     <vcard:GROUP rdf:resource="http://Cloud/
         Resource_Allocation/Multiple/A4"/>
   </rdf:Description>

   <rdf:Description rdf:about="http://Cloud/
       Resource_Allocation/Cloud_Masking/A1">
     <vcard:KEY>Requirement:R1</vcard:KEY>
     <vcard:KEY>Type:T1</vcard:KEY>
   </rdf:Description>
   <rdf:Description rdf:about="http://Cloud/
       Resource_Allocation/Cloud_Masking/A2">
     <vcard:KEY>Requirement:R2</vcard:KEY>
     <vcard:KEY>Type:T3</vcard:KEY>
   </rdf:Description>
   <rdf:Description rdf:about="http://Cloud/
       Resource_Allocation/Cloud_Masking/A3">
     <vcard:KEY>Requirement:R3</vcard:KEY>
     <vcard:KEY>Type:T5</vcard:KEY>
   </rdf:Description>
   <rdf:Description rdf:about="http://Cloud/
       Resource_Allocation/Cloud_Masking/A4">
     <vcard:KEY>Requirement:R4</vcard:KEY>
```

```
      <vcard:KEY>Type:T7</vcard:KEY>
    </rdf:Description>
    <rdf:Description rdf:about="http://Cloud/
        Resource_Allocation/Temperature/A4">
      <vcard:KEY>Requirement:R4</vcard:KEY>
      <vcard:KEY>Type:T7</vcard:KEY>
         </rdf:Description>
    <rdf:Description rdf:about="http://Cloud/
        Resource_Allocation/Temperature/A3">
      <vcard:KEY>Requirement:R3</vcard:KEY>
      <vcard:KEY>Type:T5</vcard:KEY>
    </rdf:Description>
    <rdf:Description rdf:about="http://Cloud/
        Resource_Allocation/Temperature/A2">
      <vcard:KEY>Requirement:R2</vcard:KEY>
      <vcard:KEY>Type:T3</vcard:KEY>
    </rdf:Description>
<rdf:Description rdf:about="http://Cloud/Resource_Allocation
   /Temperature/A1">
      <vcard:KEY>Requirement:R1</vcard:KEY>
      <vcard:KEY>Type:T1</vcard:KEY>
    </rdf:Description>
    <rdf:Description rdf:about="http://Cloud/
        Resource_Allocation/Multiple/A4">
      <vcard:KEY>Requirement:R8</vcard:KEY>
      <vcard:KEY>Type:T8</vcard:KEY>
         </rdf:Description>
    <rdf:Description rdf:about="http://Cloud/
        Resource_Allocation/Multiple/A3">
      <vcard:KEY>Requirement:R7</vcard:KEY>
      <vcard:KEY>Type:T6</vcard:KEY>
    </rdf:Description>
```

```
  <rdf:Description rdf:about="http://Cloud/
      Resource_Allocation/Multiple/A2">
    <vcard:KEY>Requirement:R6</vcard:KEY>
    <vcard:KEY>Type:T4</vcard:KEY>
  </rdf:Description>
<rdf:Description rdf:about="http://Cloud/Resource_Allocation
    /Multiple/A1">
    <vcard:KEY>Requirement:R5</vcard:KEY>
    <vcard:KEY>Type:T2</vcard:KEY>
  </rdf:Description>
</rdf:RDF>
```

After the Composition Phase, the application with required satellite images will be loaded in VM , which will be available to users for consumption of resources.

4.1.2 Algorithms

Algorithm 1 : Checking for Jobs from above Meta-Data

```
1: procedure
2:     Initialize Job_Agent
3:     Check Meta-Data & get Job-Id's
4:     if $no\_of\_Jobs != null$ then
5:         goto:
6:         Requirement_Phase(Job-Id)
7:     else
8:         Print: "No jobs to run!!"
```

Algorithm 2 : Requirement_Phase(Job-Id)

```
1: procedure
2:     Initialize Array Job_Parameters[][]
3:     From Job-Id retrieve Job-Type,Job-Size from Meta-Data
4:     Retrieve Parameters using Training Data-Set
5:     Job_Parameters[Job-Id] = training_data(Job-Id,Job-Type,Job-Size)
6:     goto:
7:     Discovery_Phase(Job-Id,Job_Parameters[Job-Id])
8:     set Status = "Initializing"
```

Algorithm 3 : training_data(Job-Id, Job-Type, Job-Size)

```
1: procedure
2:     Initialize Parameters[]
3:     Retrieving CPU,Memory, and VCPU's
       from Job-Id ,Job-Type,Job-Size in Training Data-set
4:     Initialize ArrayTD[]
5:     where, TD will store App-Type,App-Size,CPU,Memory,and VCPU respectively.
6:     if App-Type == TD[0] && App-Size == TD[1] then
7:         initializeint X=0;andint n=length(TD);
8:         for int i = 2 to n do
9:             Parameter[X] = TD[i]
10:            X++
11:    else
12:        Parameters = Default Parameter allocation
```

Algorithm 4 : Discovery_Phase(Job-Id,Job_Parameters(Job-Id)))

```
1: procedure
2:     Initialize param = Job_Parameters(Job-Id)
3:     if param[0] <Available_CPU && param[1] <Available_Memory then
4:         Try:
5:         VM_Allocate(param,Job-Id)
6:     else
7:         Print: "Resources fully utilized..please wait"
```

Algorithm 5 : VM_Allocate(param[], Job-Id))

```
1: procedure
2:     Initialize Template[]
3:     Template(param,Job-Id)
4:     {
5:     Initialize parameters OS,CPU,MEMORY,VCPU,DISK,NETWORK
6:     OS = Default_OS
7:     CPU = param[0]
8:     MEMORY = param[1]
9:     VCPU = param[2]
10:    DISK = Default_Disk
11:    OS = Default_OS
12:    NETWORK = Default_Network
13:    }
14:    Template = Template(param,Job-Id)
15:    Try:
16:    Instantiate(Template)
17:    set Status = "Running"
18:    Catch:
19:    Print:"VM_allocation Error"
```

Algorithm 6 : VM_De_Allocate(param[], Job-Id)

```
1: procedure
2:     get param = Job_Parameters(Job-Id)
3:     Try:
4:     Release(param,Job-Id)
5:     set Status = "End"
6:     Print:"VM for Job-Id released!!."
```

Chapter 5

Configuring the Cloud

5.1 Hardware and Software

1. **Hardware**

   (a) **System:** CentOS (6.4 Final) (Both Masternode and Workernode's)

   (b) **Kernal:** 3.13.0-32 Generic

   (c) **Architecture:** x86_64 (64bit)

   (d) **CPU op-mode(s):** 32-bit, 64-bit

   (e) **CPU(s):** 2

   (f) **Core(s) per socket:** 2

   (g) **Memory:** 2GB

   (h) **Vendor ID:** Genuine-Intel

   (i) **Virtualization:** VT-x

2. **Softwares / packages**

   (a) **Cloud Utility:** OpenNebula-Sunstone Server , opennebula-common (CentOS 6.4)

   (b) **Hypervisor:** KVM

   (c) **Libvirt-bin:** Programs for the libvirt library

   (d) **RUBY:** Interpreter of object-oriented scripting language Ruby (default version)

(e) **Openssh-Server:** secure shell (SSH) server, for secure access from remote machines

(f) **Network File Sharing:** nfs-kernel-server

(g) **Other language and API'S** Java OpenNebula Cloud API

5.2 Opennebula and KVM Configuration

1. **FrontEnd**

   (a) **Install the repo:** Add the OpenNebula repository from
   `http://downloads.opennebula.org/repo/Ubuntu/repo.key`

   (b) **Install opennebula:** fire *opennebula-sunstone and nfs-kernel-server start* in cmd window to start server.

   (c) **Configure NFS:**Use *service nfs-kernel-server start* in cmd window for NFS.

   (d) **Configure SSH Public Key:** OpenNebula will need to SSH passwordlessly from any node (including the frontend) to any other node. which can be done copying rsa public keys to authorized keys folder by firing command *cp* $/.ssh/id_rsa.pub$ $/.ssh/authorized_keys$ also add below snippet to
   $/var/lib/one/.ssh/configfile$
   $Host*$
   $StrictHostKeyCheckingno$
   $UserKnownHostsFile/dev/null$
   and give read only permissions to other nodes using
   $chmod600$ $/.ssh/config$

2. **Installation in Workernodes**

   (a) **Install the repo:** Add the OpenNebula repository from
   `http://downloads.opennebula.org/repo/Ubuntu/repo.key`

   (b) **Install opennebula:** Install Opennebula-Node NFS-Common bridge-utils

   (c) **Configure the Network:** Add bridge to the workernodes so that all other nodes can access the network under the masternode.

(d) **Configure NFS:** Mount the datastores export. Add the following to your $/etc/fstab$: $masternodeIP$:$/var/lib/one//var/lib/one/nfssoft, intr, rsize = 8192, wsize = 8192, noautoanddomount/var/lib/one/$

(e) **Configure Qemu :**The oneadmin user must be able to manage libvirt as root,so add the following to $/etc/libvirt/qemu.conf$

$user = "oneadmin"$

$group = "oneadmin"$

$dynamic_ownership = 0$

$anddoservicelibvirt - binrestart$

3. **Working on cloud**

(a) **starting opennebula:** Go to browser and go on *http://frontend:9869*

(b) **Adding a Host:** Add the workernodes using *onehost create workernode-ip -i kvm -v kvm -n dummy*

(c) **Adding virtual resources:** To create networks, we need to create first a network template file mynetwork.one that contains attributes like:

$NAME = "private"$

$BRIDGE = br0$

$LEASES = [IP = 192.168.0.100]$

$LEASES = [IP = 192.168.0.101]$

$LEASES = [IP = 192.168.0.102]$

(d) **Running a Virtual MachineS:** Inititate the template created above for VM using *onetemplate instantiate* $"CentOS - 6.5" - -name"MyScratchVM"$.

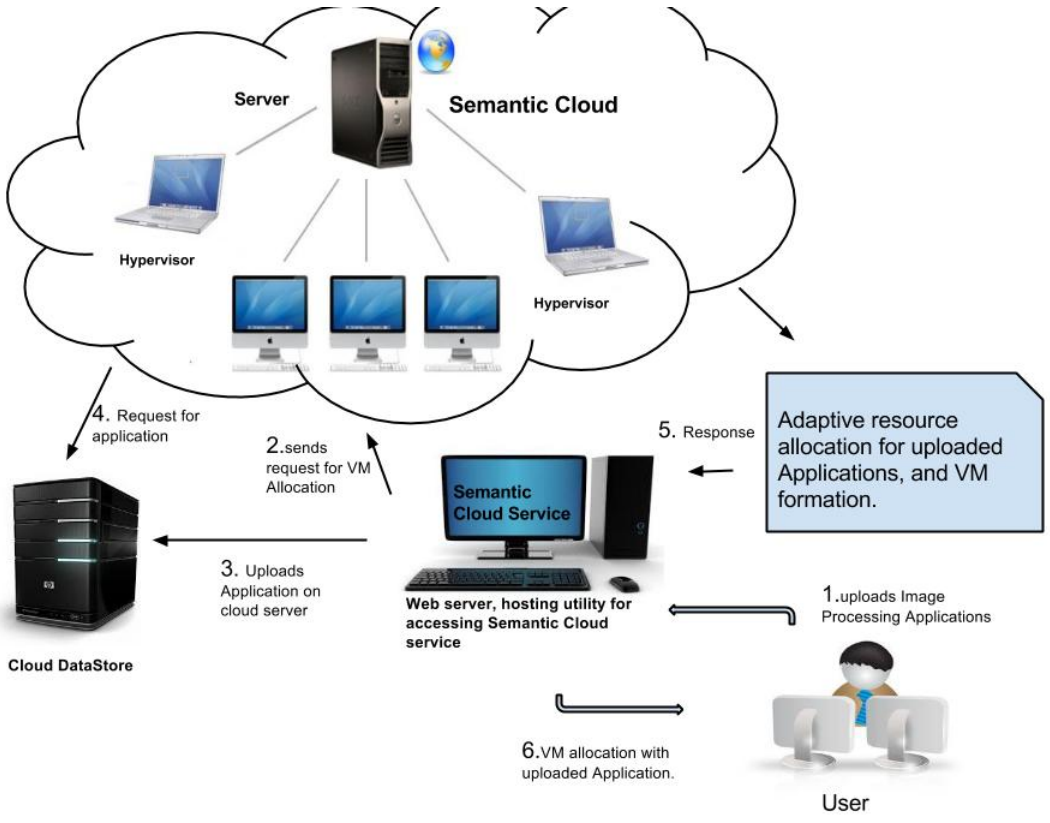
5.3 Libraries used for building Web Utility for accessing Semantic Cloud Services

1. Java API's for Open-Nebula
2. JENA library for generating Ontology
3. Apache Tomcat Server for Deploying Services over WEB.

Chapter 6

Design Flow of Web-Utility for Accessing Semantic Cloud Services

6.1 Web-Utility Design-Flow



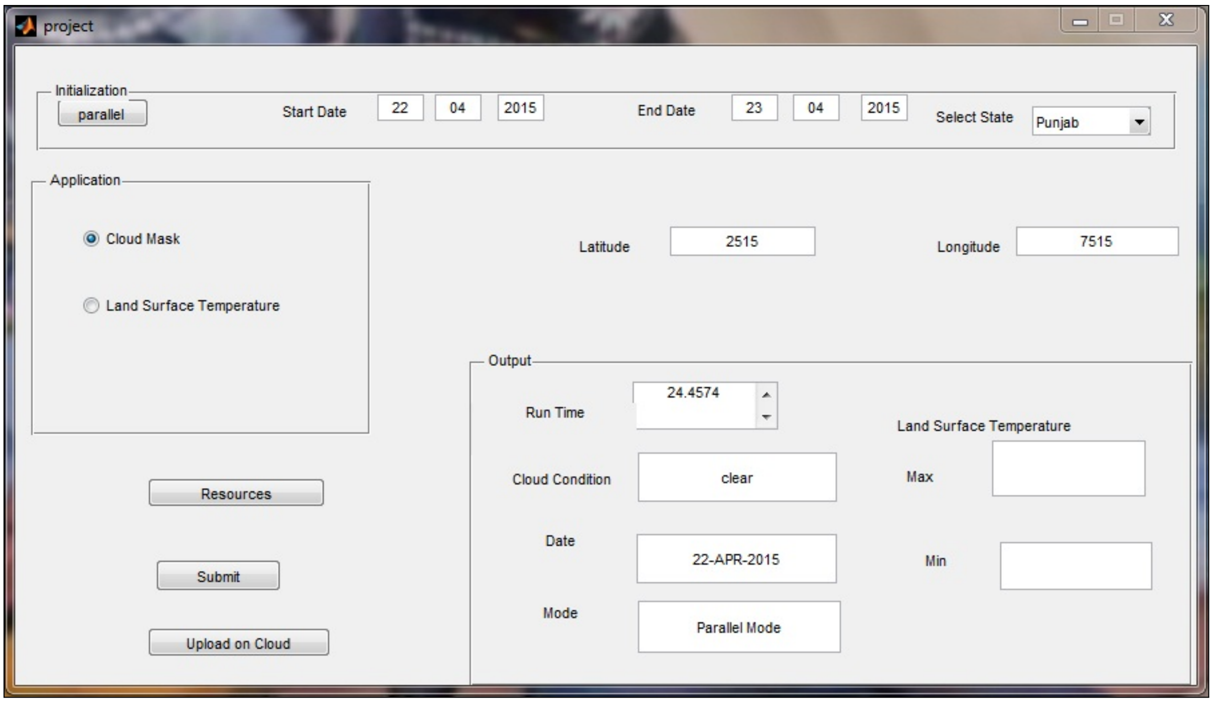
Server
Semantic Cloud
Hypervisor
Hypervisor
4. Request for application
2.sends request for VM Allocation
5. Response
Adaptive resource allocation for uploaded Applications, and VM formation.
Semantic Cloud Service
3. Uploads Application on cloud server
Web server, hosting utility for accessing Semantic Cloud service
1.uploads Image Processing Applications
Cloud DataStore
6.VM allocation with uploaded Application.
User

Figure 6.1: Design Flow of Semantic Cloud Web Service

6.1.1 Flow

1. User will upload an Application on Web-Utility for accessing Semantic Cloud,for research,deploying or testing purpose.
2. Web Service will sends a request to Semantic Cloud Service for appropriate VM Allocation for uploaded Application.
3. Semantic Cloud will orchestrate the required resources for creation of VM, using the training dataset.
4. Virtual Machine will be created on the bases of resource required and Allocated to User.

6.2 Screen Shots


project

Initialization: parallel

Start Date 22 04 2015

End Date 23 04 2015

Select State Punjab

Application: [✓ Cloud Mask] Land Surface Temperature

Latitude 2515

Longitude 7515

Output

Run Time 24.4574

Cloud Condition clear

Date 22-APR-2015

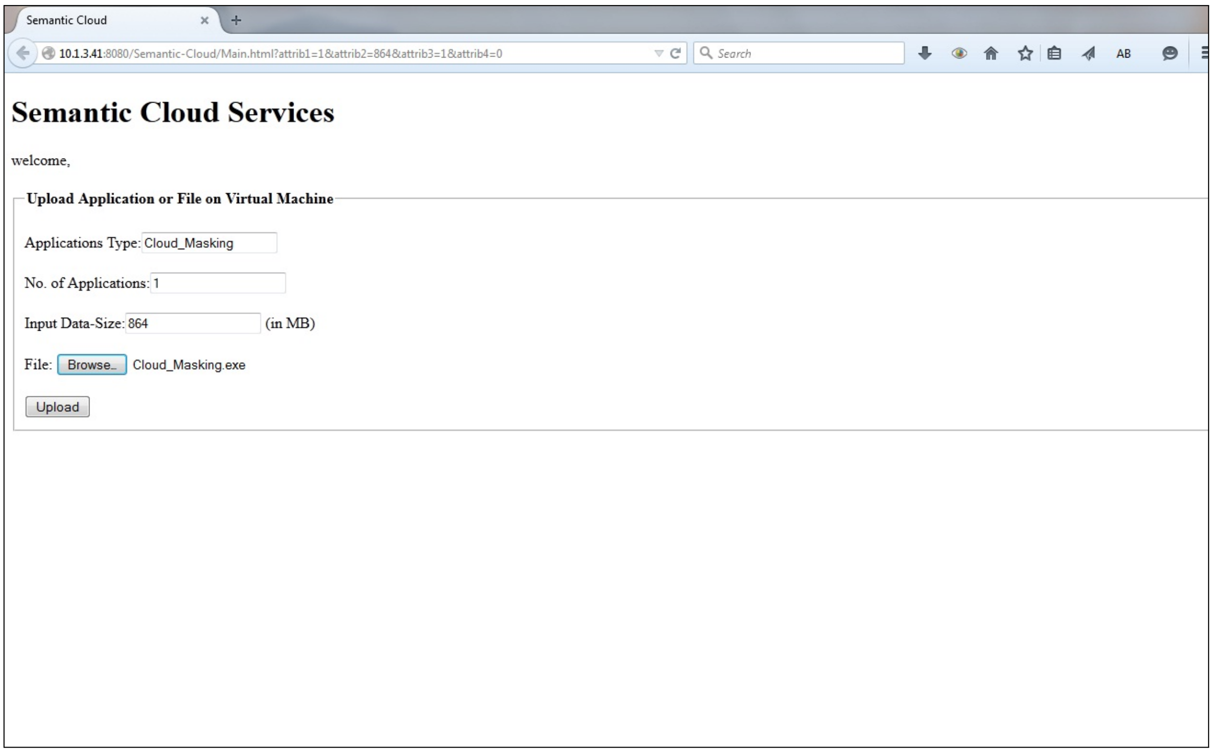
Mode Parallel Mode

Land Surface Temperature

Max

Min

Figure 6.2: Image Processing Application



Semantic Cloud

10.1.3.41:8080/Semantic-Cloud/Main.html?attrib1=1&attrib2=864&attrib3=1&attrib4=0

Semantic Cloud Services

welcome,

Upload Application or File on Virtual Machine

Applications Type: Cloud_Masking

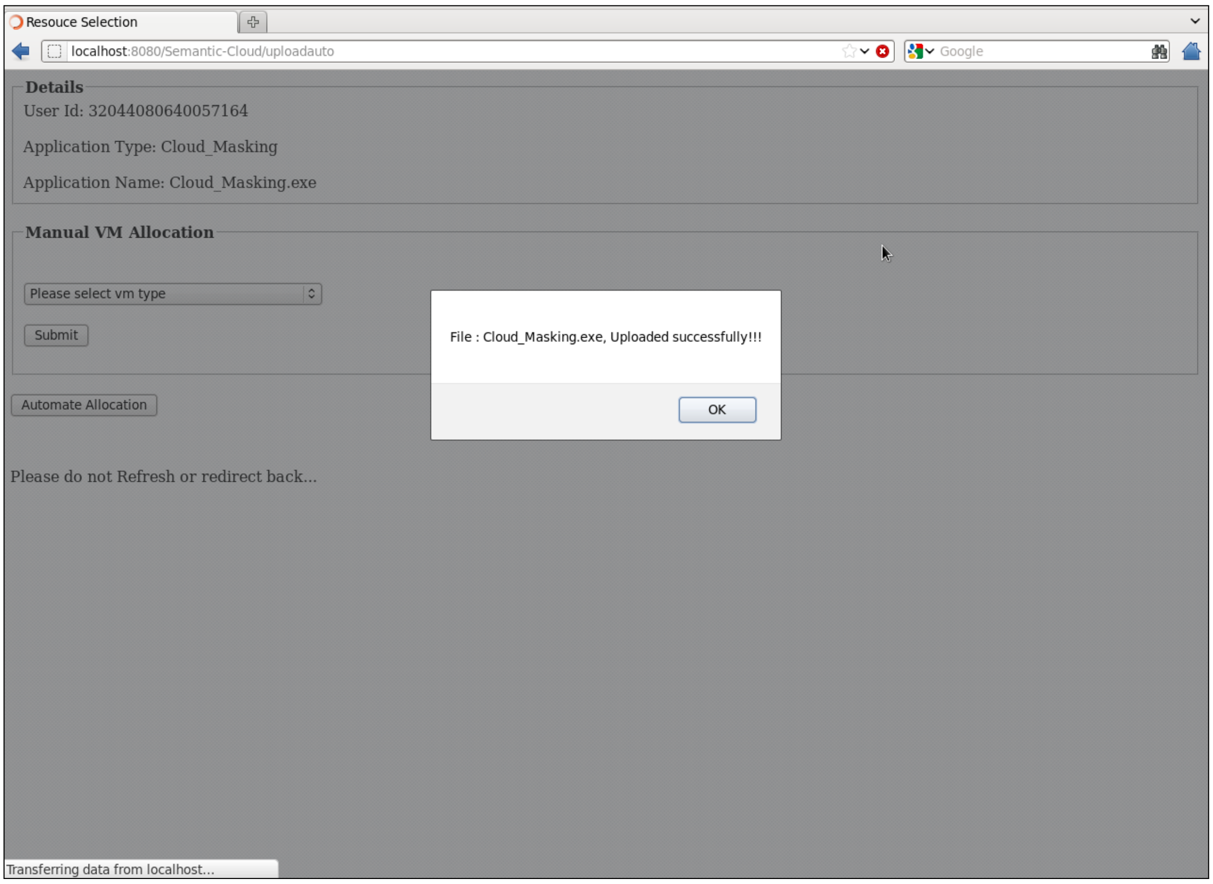
No. of Applications: 1

Input Data-Size: 864 (in MB)

File: Browse... Cloud_Masking.exe

Upload

Figure 6.3: Uploading application, to be deployed on Virtual Machine



Resouce Selection

localhost:8080/Semantic-Cloud/uploadauto

Details

User Id: 32044080640057164

Application Type: Cloud_Masking

Application Name: Cloud_Masking.exe

Manual VM Allocation

Please select vm type

Submit

Automate Allocation

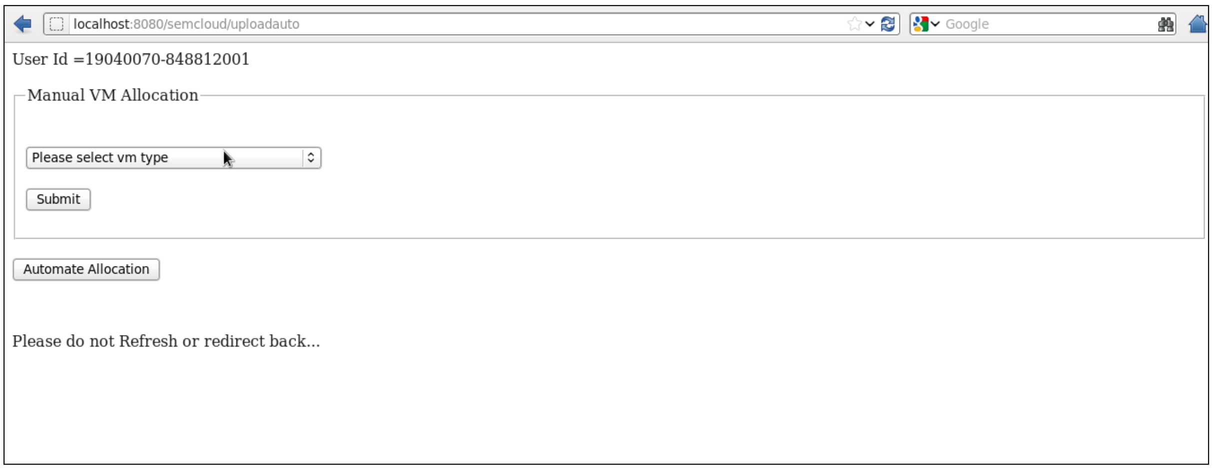
Please do not Refresh or redirect back...

File : Cloud_Masking.exe, Uploaded successfully!!!

OK

Transferring data from localhost...

Figure 6.4: Upload success

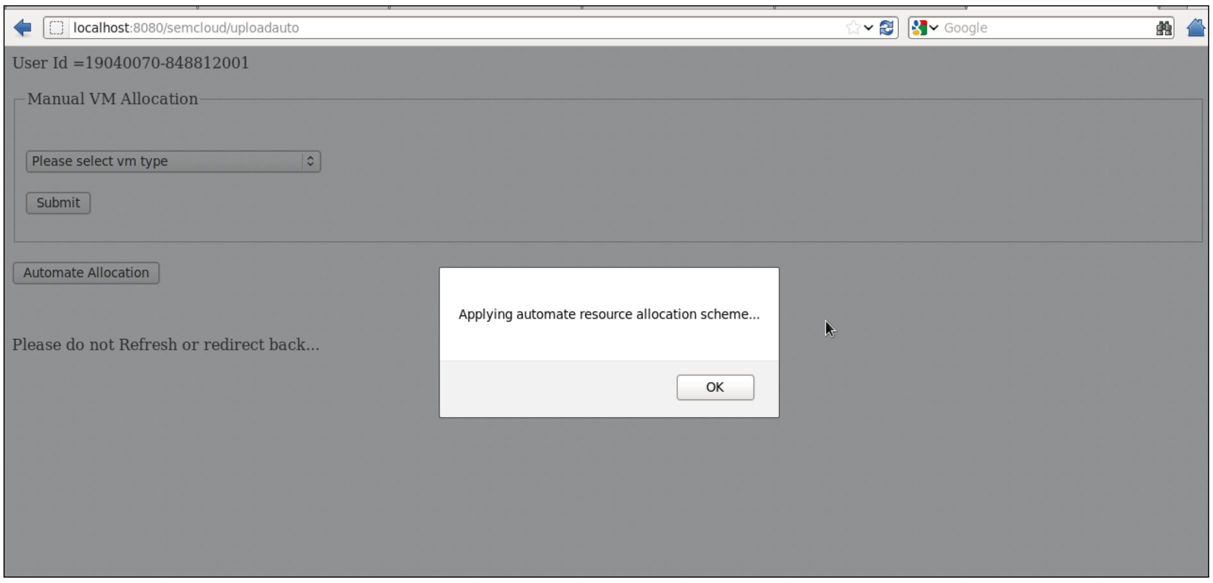

localhost:8080/semcloud/uploadauto

User Id =19040070-848812001

Manual VM Allocation

Please do not Refresh or redirect back...

Figure 6.5: Allocation schemes


localhost:8080/semcloud/uploadauto

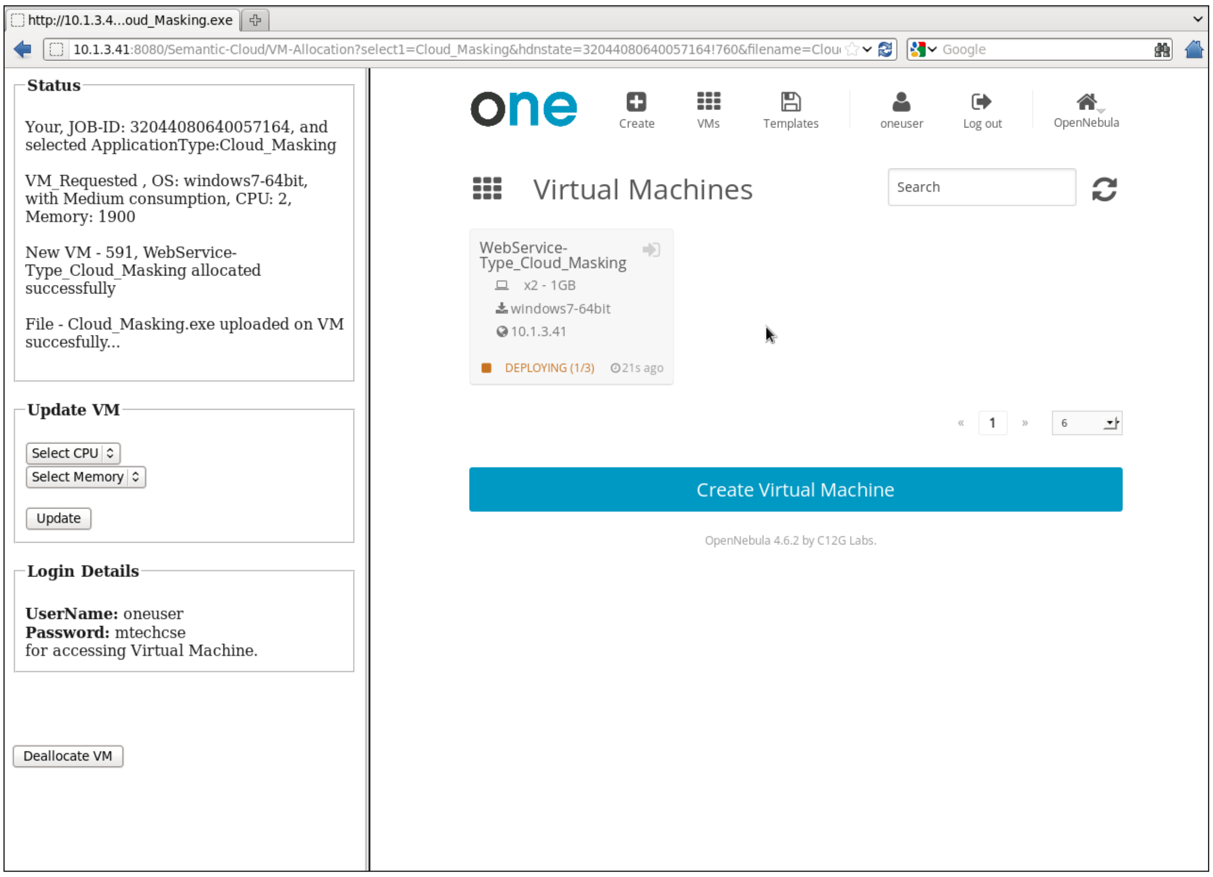
User Id =19040070-848812001

Manual VM Allocation

Please do not Refresh or redirect back...

Applying automate resource allocation scheme...

Figure 6.6: Automate allocation of Resources


http://10.1.3.4...oud_Masking.exe

10.1.3.41:8080/Semantic-Cloud/VM-Allocation?select1=Cloud_Masking&hdnstate=32044080640057164!760&filename=Clou

Status

Your, JOB-ID: 32044080640057164, and selected ApplicationType:Cloud_Masking

VM_Requested , OS: windows7-64bit, with Medium consumption, CPU: 2, Memory: 1900

New VM - 591, WebService-Type_Cloud_Masking allocated successfully

File - Cloud_Masking.exe uploaded on VM succesfully...

Update VM

Select CPU
Select Memory
Update

Login Details

UserName: oneuser
Password: mtechcse
for accessing Virtual Machine.

Deallocate VM

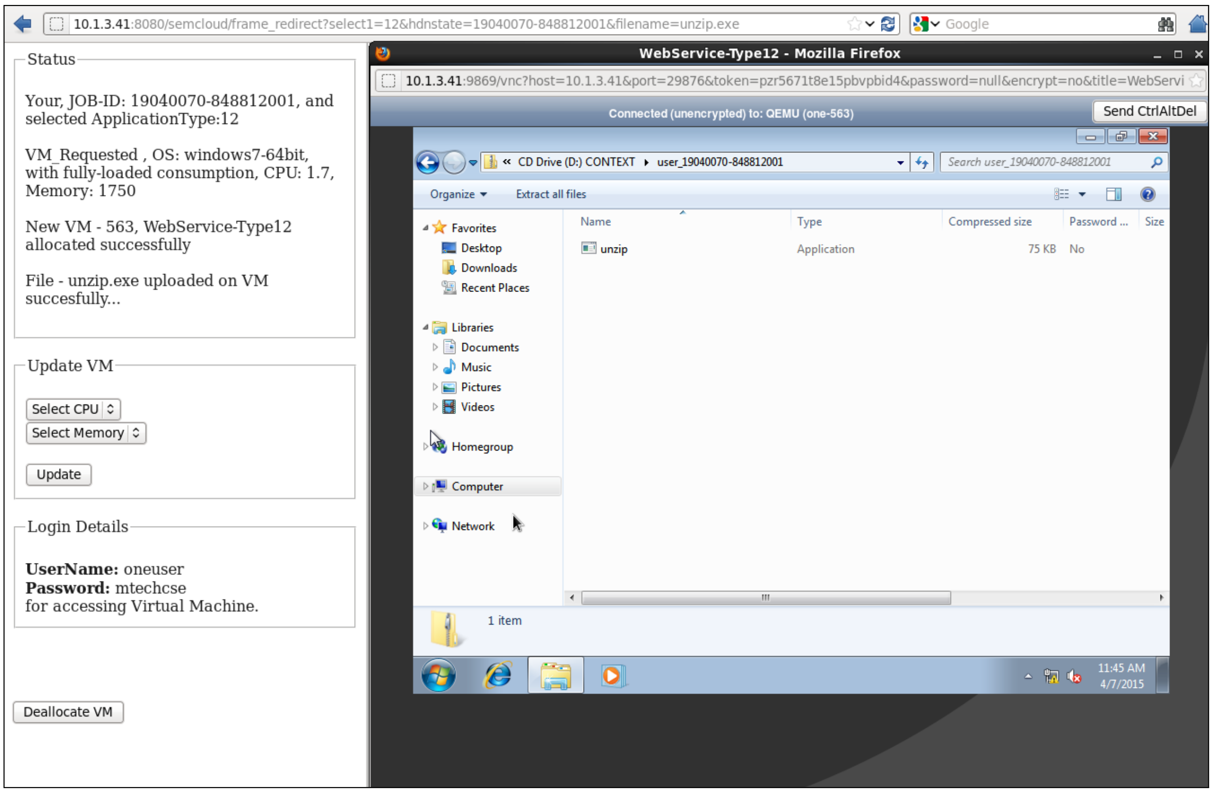
Virtual Machines

WebService-Type_Cloud_Masking
x2 - 1GB
windows7-64bit
10.1.3.41
DEPLOYING (1/3) 21s ago

Create Virtual Machine

OpenNebula 4.6.2 by C12G Labs.

Figure 6.7: VM_Creation


10.1.3.41:8080/semcloud/frame_redirect?select1=12&hdnstate=19040070-848812001&filename=unzip.exe

Status

Your, JOB-ID: 19040070-848812001, and selected ApplicationType:12

VM_Requested , OS: windows7-64bit, with fully-loaded consumption, CPU: 1.7, Memory: 1750

New VM - 563, WebService-Type12 allocated successfully

File - unzip.exe uploaded on VM succesfully...

Update VM

Select CPU
Select Memory
Update

Login Details

UserName: oneuser
Password: mtechcse
for accessing Virtual Machine.

Deallocate VM

WebService-Type12 - Mozilla Firefox

10.1.3.41:9869/vnc?host=10.1.3.41&port=29876&token=pzr5671t8e15pbvpbid4&password=null&encrypt=no&title=WebServi

Connected (unencrypted) to: QEMU (one-563)

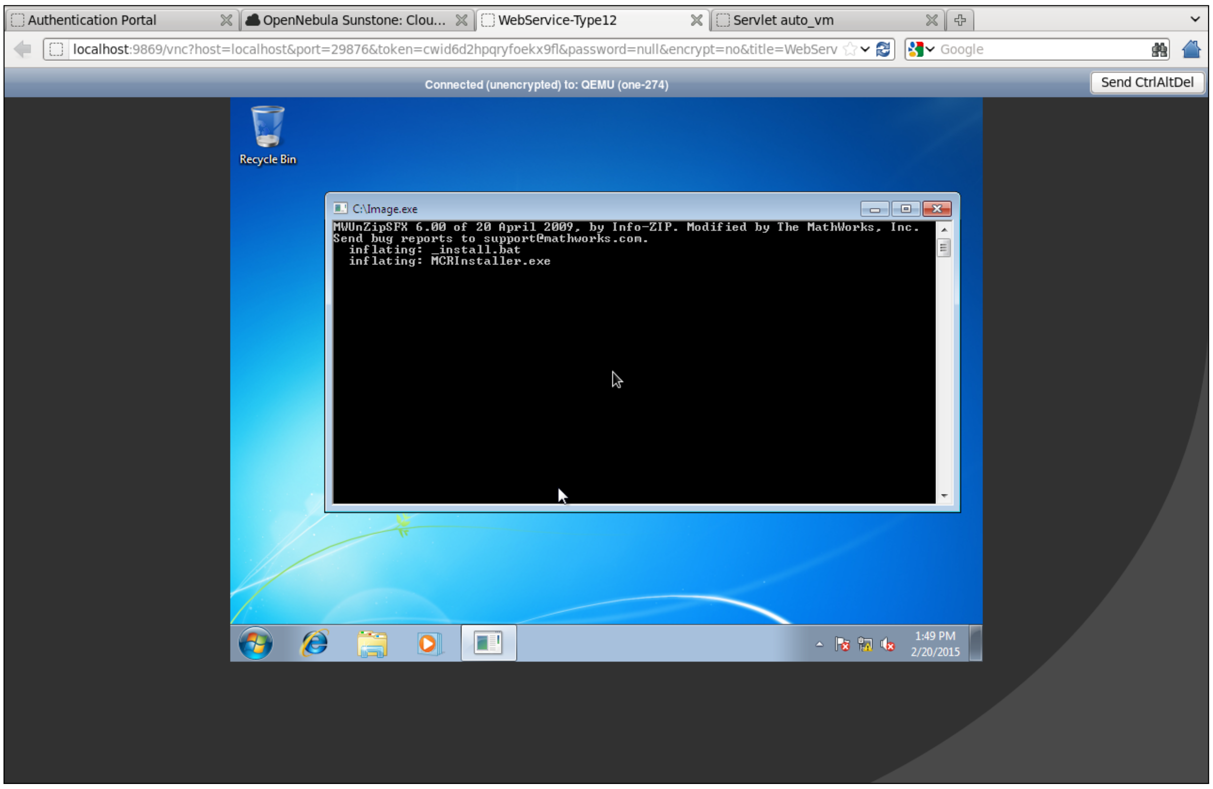
« CD Drive (D:) CONTEXT ▸ user_19040070-848812001

Name	Type	Compressed size	Password ...	Size
unzip	Application	75 KB	No	

1 item

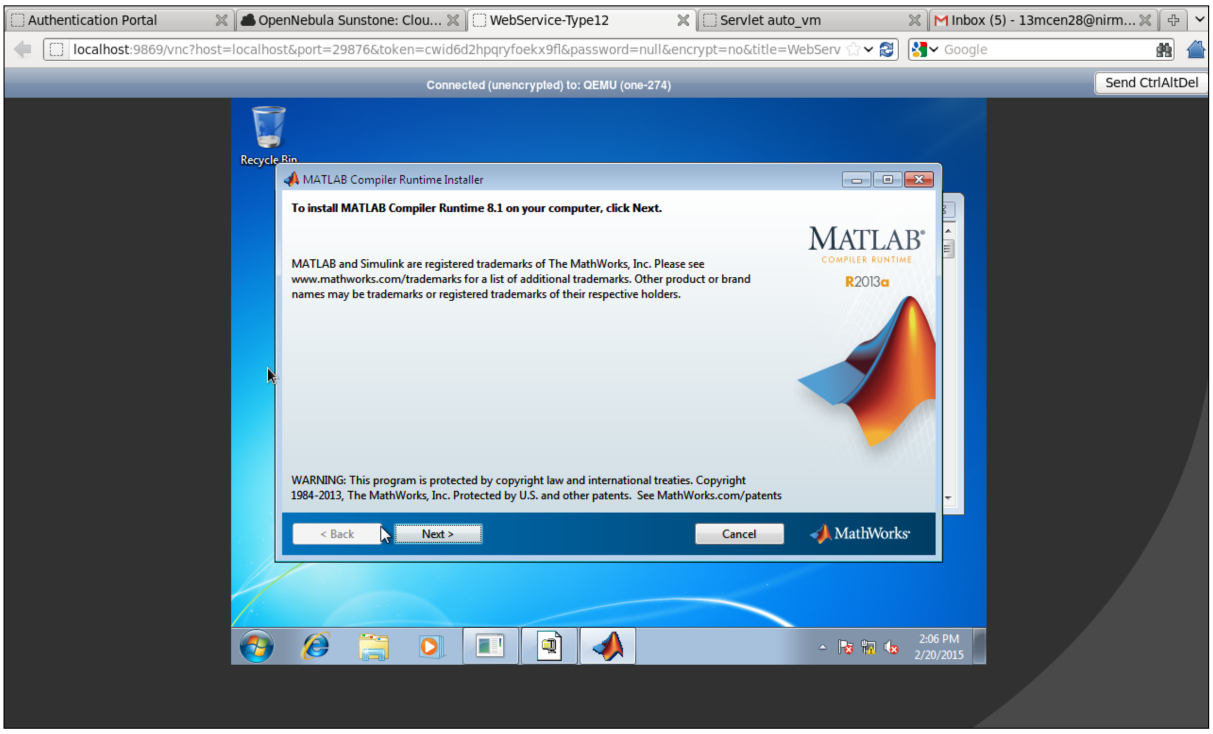
11:45 AM
4/7/2015

Figure 6.8: VM Consumption


Authentication Portal
OpenNebula Sunstone: Clou...
WebService-Type12
Servlet auto_vm
localhost:9869/vnc?host=localhost&port=29876&token=cwid6d2hpqryfoekx9fl&password=null&encrypt=no&title=WebServ
Google
Connected (unencrypted) to: QEMU (one-274)
Send CtrlAltDel
Recycle Bin
C:\Image.exe
1:49 PM
2/20/2015

```
MVUnZipSFX 6.00 of 20 April 2009, by Info-ZIP. Modified by The MathWorks, Inc.
Send bug reports to support@mathworks.com.
  inflating: _install.bat
  inflating: MCRInstaller.exe
```

Figure 6.9: Application running on VM

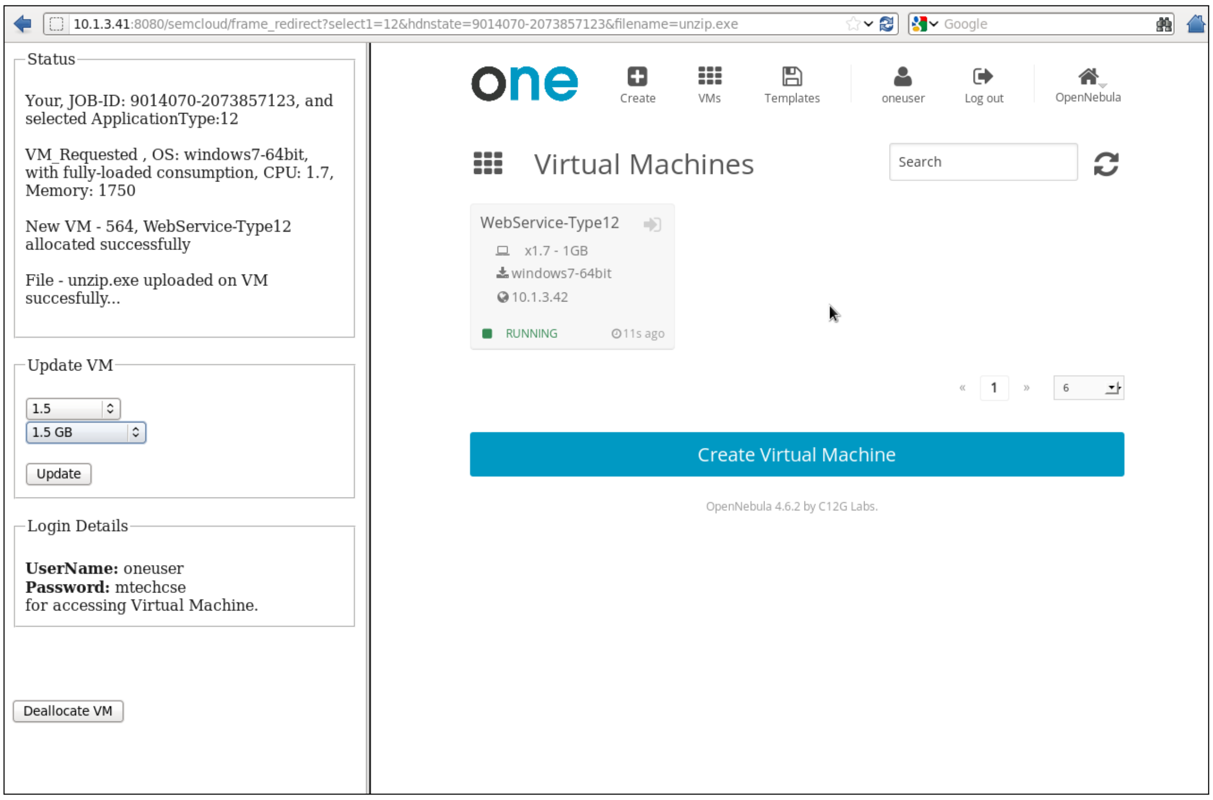

Authentication Portal
OpenNebula Sunstone: Clou...
WebService-Type12
Servlet auto_vm
Inbox (5) - 13mcen28@nirm...
localhost:9869/vnc?host=localhost&port=29876&token=cwid6d2hpqryfoekx9fl&password=null&encrypt=no&title=WebServ
Google
Connected (unencrypted) to: QEMU (one-274)
Send CtrlAltDel
Recycle Bin
MATLAB Compiler Runtime Installer
MATLAB
COMPILER RUNTIME
R2013a
< Back
Next >
Cancel
MathWorks
2:06 PM
2/20/2015

To install MATLAB Compiler Runtime 8.1 on your computer, click Next.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Please see www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

WARNING: This program is protected by copyright law and international treaties. Copyright 1984-2013, The MathWorks, Inc. Protected by U.S. and other patents. See MathWorks.com/patents

Figure 6.10: Accessing Application



10.1.3.41:8080/semcloud/frame_redirect?select1=12&hdnstate=9014070-2073857123&filename=unzip.exe

Status

Your, JOB-ID: 9014070-2073857123, and selected ApplicationType:12

VM_Requested , OS: windows7-64bit, with fully-loaded consumption, CPU: 1.7, Memory: 1750

New VM - 564, WebService-Type12 allocated successfully

File - unzip.exe uploaded on VM succesfully...

Update VM

1.5
1.5 GB
Update

Login Details

UserName: oneuser
Password: mtechcse
for accessing Virtual Machine.

Deallocate VM

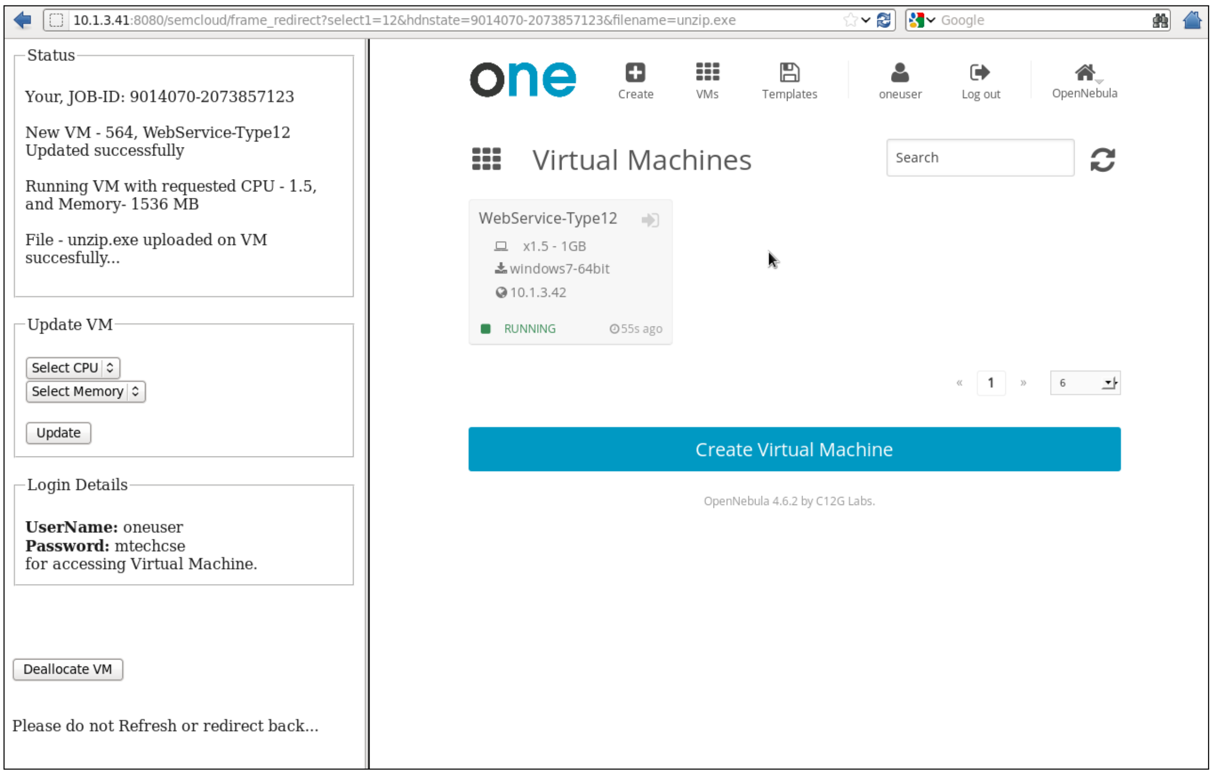
Virtual Machines

WebService-Type12
x1.7 - 1GB
windows7-64bit
10.1.3.42
RUNNING 11s ago

Create Virtual Machine

OpenNebula 4.6.2 by C12G Labs.

Figure 6.11: Updating Resource requirement



10.1.3.41:8080/semcloud/frame_redirect?select1=12&hdnstate=9014070-2073857123&filename=unzip.exe

Status

Your, JOB-ID: 9014070-2073857123

New VM - 564, WebService-Type12 Updated successfully

Running VM with requested CPU - 1.5, and Memory- 1536 MB

File - unzip.exe uploaded on VM succesfully...

Update VM

Select CPU
Select Memory
Update

Login Details

UserName: oneuser
Password: mtechcse
for accessing Virtual Machine.

Deallocate VM

Please do not Refresh or redirect back...

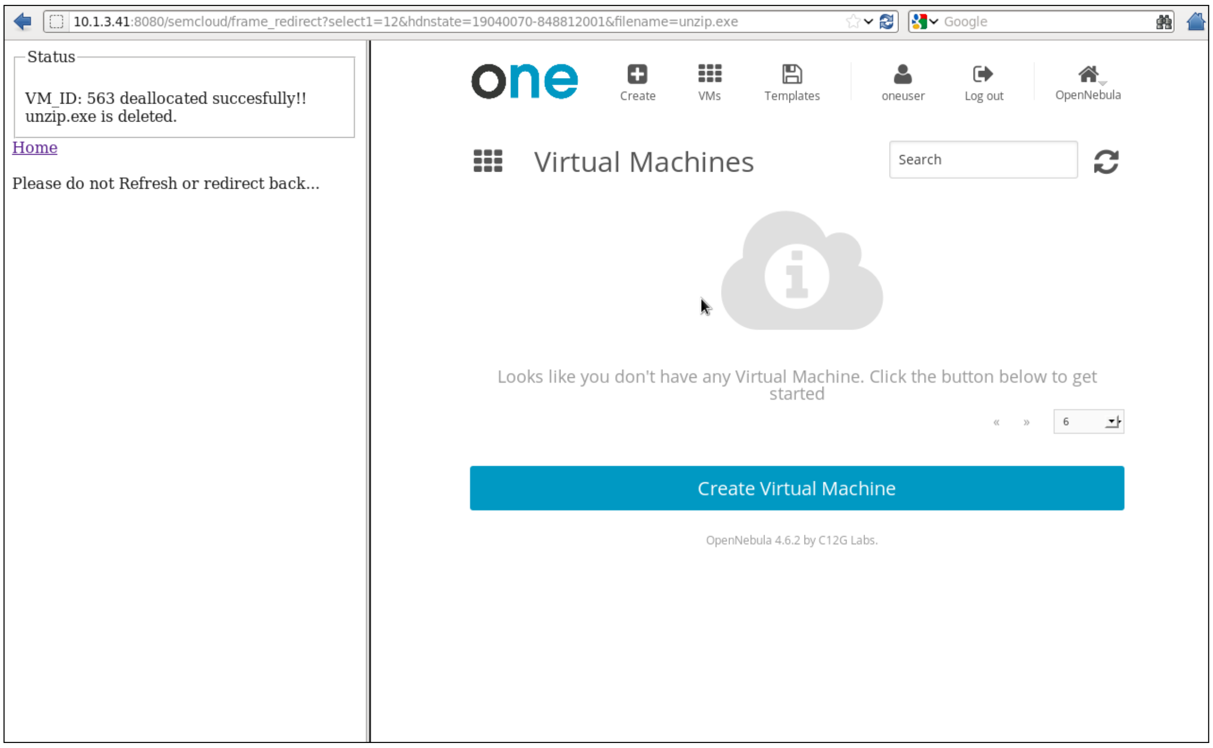
Virtual Machines

WebService-Type12
x1.5 - 1GB
windows7-64bit
10.1.3.42
RUNNING 55s ago

Create Virtual Machine

OpenNebula 4.6.2 by C12G Labs.

Figure 6.12: After Updation


10.1.3.41:8080/semcloud/frame_redirect?select1=12&hdnstate=19040070-848812001&filename=unzip.exe

Status

VM_ID: 563 deallocated succesfully!!
unzip.exe is deleted.

Home

Please do not Refresh or redirect back...

Virtual Machines

Looks like you don't have any Virtual Machine. Click the button below to get started

OpenNebula 4.6.2 by C12G Labs.

Figure 6.13: Deallocation of VM

Chapter 7

Conclusion and Future Scope

7.1 Graphs representing resource utilization

The section describes the difference between utilization of resources while running the Applications on VMś, Semantic Cloud allocates the resource based on training data-set of Predefined Ontologyś, Initially, which is created by running the different type of applications on VMś with different resource allocation schemes.The Training data-set then will train its data-set based on certain parameters,which helps in making resource utilization efficient.

The following graphs shows the graphs of VMś, Hostś that runs VMś, Network Utilization, during Initialization phase and during the running state of Applicationś.

7.1.1 Results

1. Resource utilization of Virtual Machineś(VM)

   (a) CPU utilization

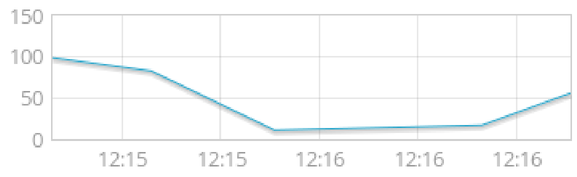


Figure 7.1: CPU utilization during manual allocation

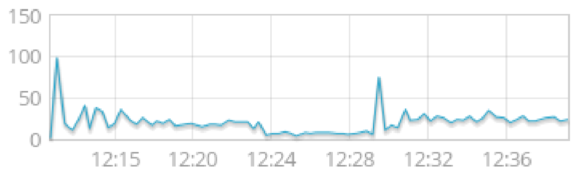


Figure 7.2: CPU utilization during automate allocation

(b) Memory utilization

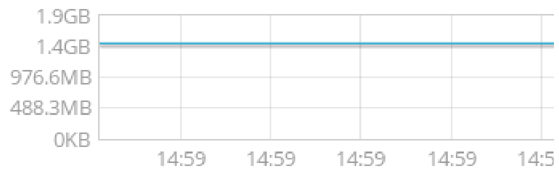


Figure 7.3: Memory utilization during manual allocation

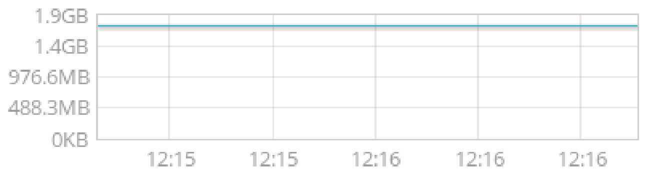


Figure 7.4: Memory utilization during automate allocation

Above graphs shows the utilization of CPU and Memory under automated and manual allocation of resources. It demonstrates that during the manual allotment CPU is under expended, while automated one smoothly utilizes CPU and Memory.

2. Resource utilization of Hostś running VMś.

   (a) CPU utilization

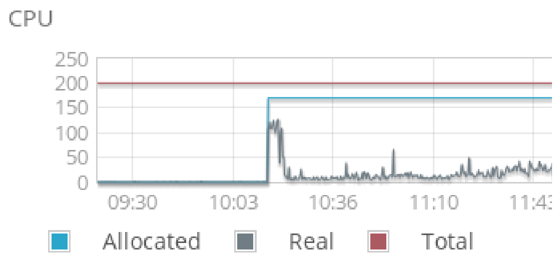


Figure 7.5: CPU utilization during manual allocation

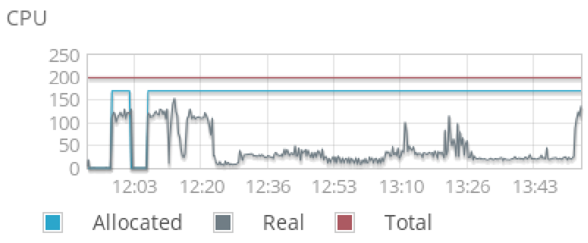


Figure 7.6: CPU utilization during automate allocation

   (b) Memory utilization

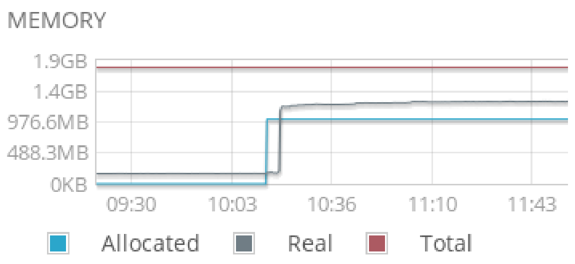


Figure 7.7: Memory utilization during manual allocation

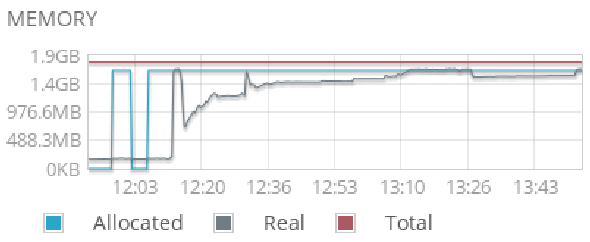


Figure 7.8: Memory utilization during automate allocation

This graphs demonstrates the CPU and Memory utilization of hosts, running VMs. Under manual allocation, it is seen that aduring initial stage it squanders the resources while remaining ideal.While automated allotment utilizes the resources only when needed, giving he optimum resource usage performance.

3. Network utilization over Semantic Cloud.

   (a) Network Upload/Download Speed

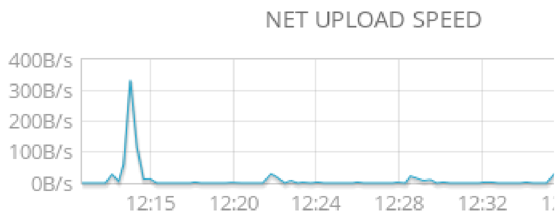


Figure 7.9: Upload Speed over N/W

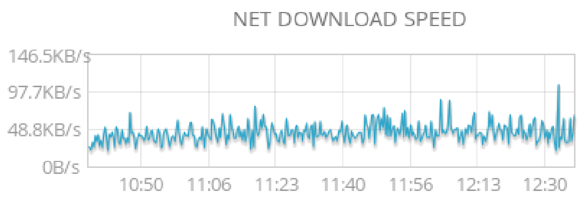


Figure 7.10: Download Speed over N/W

   (b) Transferred/Received rate of No. of Bytes over Network

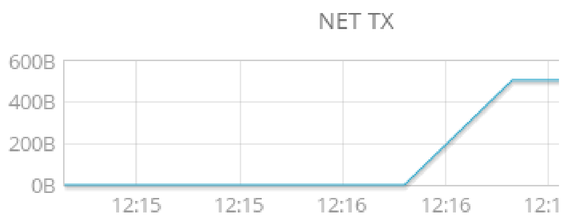


Figure 7.11: Rate of No. of Bytes Transferred

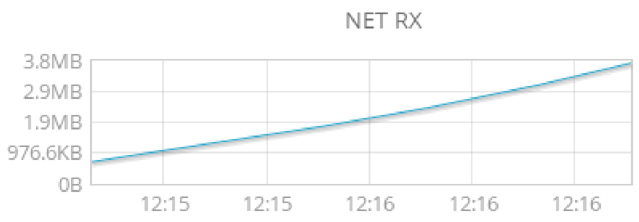


Figure 7.12: Rate of No. of Bytes Received

The Network utilization graphs for upload and Download speeds shows the utilization of bandwidth while uploading of Jobs and downloading of data.

Transferred rate and Received rate of Bytes describes the rate number of bytes transferred during the execution of Jobs over semantic web service.

7.2 Conclusion and Future Scope

A survey of papers on semantic cloud is done and so, decision for using Ontology over semantic cloud is taken. Finally, an Ontology based semantic cloud is proposed for meteorological data processing applications, where the managing of resources is automated using the ontology and training data-set. The algorithms for automated virtual machine and training data-set will be developed further.

Also from the above graphs, it is concluded that resource consumption rate in ontology based automate selection is efficient compared to manual one. As the proposed algorithms will update the ontologies, if found more better resource allocation strategy for job, making the automate allocation more and more efficient.As the work on resource estimation for dissimilar jobs or applications on Cloud.[11], which consists the classification of resource requirements based on application behaviours and [12],which defines a model to determine the resource costs based on SLA for deploying compute-intensive jobs in a cloud , The future work can be extended on this resource estimation techniques for making the semantic cloud more resource efficient.

References

[1] Wikipedia, "Image processing."

[2] Wikipedia, "Weather forecasting."

[3] O. T. Giuseppe Di Modica, Giuseppe Petralia, "A semantic framework to support cloud markets in interoperable scenarios," IEEE, 2012.

[4] J. Y. Fangfang Liu, Xiangfeng Luo and G. Liang, "Semantic cloud based on sln and aln," pp. 314 – 317, IEEE Fifth International Conference on Semantics,Knowledge and Grid, October 2009.

[5] G. Cretella and B. Martino, "Towards a semantic engine for cloud applications development," pp. 198–203, IEEE Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, 2012.

[6] R. K. Le Duy Ngan, "Owl-s based semantic cloud service broker," pp. 560–567, IEEE 19th International Conference on Web Services, 2013.

[7] Y. Y. K. Joshi and T. Finin, "Automating cloud services lifecycle through semantic technologies," pp. 109–122, IEEE TRANSACTIONS ON SERVICE COMPUTING,Volume-7, 2013.

[8] D. R. Jeremy J. Carroll, "Jena: Implementing the semantic web recommendations," W3C Technical Reports.

[9] B. F. Matthias Klusch, "Automated semantic web service discovery with owls-mx," AAMAS 2006 May 8-12, Hakodate, Hokkaido, Japan.

[10] Jena.Apache, "Rdf."

[11] D. K. S. G. T. V. George Kousiourisa, Andreas Menychtasa, “Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms,” vol. 32, pp. 27 – 40, Journal of Future Generation Computer Systems, Elseveir, March 2014.

[12] F. Z. Rizwan Mian, Patrick Martin, “Estimating resource costs of data-intensive workloads in public clouds,” JProceeding MGC ’12 Proceedings of the 10th International Workshop on Middleware for Grids, Clouds and e-Science, Article No. 3, ACM New York, NY, USA, 2012.