College o f Engineering

–

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

CS6712 – GRID AND CLOUD COMPUTING LABORATORY

VII SEMESTER - R 2013

LABORATORY MANUAL

Name : ______________________________

Reg. No. : ______________________________

Section : ______________________________

DHANALAKSHMI

VISION

is committed to provide highly disciplined, conscientious and enterprising professionals conforming to global standards through value based quality education and training.

MISSION

- To provide competent technical manpower capable of meeting requirements of the industry
- To contribute to the promotion of Academic Excellence in pursuit of Technical Education at different levels
- To train the students to sell his brawn and brain to the highest bidder but to never put a price tag on heart and soul

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

To strive for acquiring, applying and imparting knowledge in Computer Science and Engineering through quality education and to provide enthusiastic professionals with commitment

MISSION

- To produce highly competent and globally employable engineers in the field of Computer Science and Engineering
- To inculcate human values among the student community and make them realize their commitment to the society
- To exhibit excellence in pursuit of research and innovative products with a zeal to serve the society

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Fundamentals

To provide students with a solid foundation in Mathematics, Science and fundamentals of engineering, enabling them to apply, to find solutions for engineering problems and use this knowledge to acquire higher education

2. Core Competence

To train the students in Computer science and engineering technologies so that they apply their knowledge and training to compare, and to analyze various engineering industrial problems to find solutions

3. Breadth

To provide relevant training and experience to bridge the gap between theory and practice this enables them to find solutions for the real time problems in industry, and to design products

4. Professionalism

To inculcate professional and effective communication skills, leadership qualities and team spirit in the students to make them multi-faceted personalities and develop their ability to relate engineering issues to broader social context

5. Lifelong Learning/Ethics

To demonstrate and practice ethical and professional responsibilities in the industry and society in the large, through commitment and lifelong learning needed for successful professional career

PROGRAMME OUTCOMES (POs)

a) To demonstrate and apply knowledge of Mathematics, Science and engineering fundamentals in Electronics and Communication Engineering field

b) To design a component, a system or a process to meet the specific needs within the realistic constraints such as economics, environment, ethics, health, safety and manufacturability

c) To demonstrate the competency to use software tools for computation, simulation and testing of electronics and communication engineering circuits

d) To identify, formulate and solve electronic and communication engineering problems

e) To demonstrate an ability to visualize and work on laboratory and multidisciplinary tasks

f) To function as a member or a leader in multidisciplinary activities

g) To communicate in verbal and written form with fellow engineers and society at large

h) To understand the impact of Electronics and Communication Engineering in the society and demonstrate awareness of contemporary issues and commitment to give solutions exhibiting social responsibility

i) To demonstrate professional & ethical responsibilities

j) To exhibit confidence in self-education and ability for lifelong learning

k) To participate and succeed in competitive exams

CS6712 – GRID AND CLOUD COMPUTING LABORATORY

COURSE OBJECTIVES

SYLLABUS

Learn the working of ARM processor

Understand the Building Blocks of Embedded Systems

Learn the concept of memory map and memory interface

Know the characteristics of Real Time Systems

Write programs to interface memory, I/Os with processor

Study the interrupt performance

LIST OF EXPERIMENTS:

GRID COMPUTING LAB

Use Globus Toolkit or equivalent and do the following:

1. Develop a new Web Service for Calculator.
2. Develop new OGSA-compliant Web Service.
3. Using Apache Axis develop a Grid Service.
4. Develop applications using Java or C/C++ Grid APIs
5. Develop secured applications using basic security mechanisms available in Globus Toolkit.
6. Develop a Grid portal, where user can submit a job and get the result. Implement it with and without GRAM concept.

CLOUD COMPUTING LAB

Use Eucalyptus or Open Nebula or equivalent to set up the cloud and demonstrate.

1. Find procedure to run the virtual machine of different configuration. Check how many virtual machines

can be utilized at particular time.

2. Find procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine.
3. Install a C compiler in the virtual machine and execute a sample program.
4. Show the virtual machine migration based on the certain condition from one node to the other.
5. Find procedure to install storage controller and interact with it.
6. Find procedure to set up the one node Hadoop cluster.
7. Mount the one node Hadoop cluster using FUSE.
8. Write a program to use the API's of Hadoop to interact with it.
9. Write a wordcount program to demonstrate the use of Map and Reduce tasks

COURSE OUTCOMES

- Use the grid and cloud tool kits.
- Design and implement applications on the Grid.
- Design and Implement applications on the Cloud.
- Design network objects by applying the networking concepts.
- Create virtualization concept.
- Design web based applications.
- Apply cloud tools for resource allocation.
- Apply grid tools for network translation.
- Design private cloud using open stack.

CS6712 – GRID AND CLOUD COMPUTING LABORATORY

CONTENTS

INTRODUCTION TO GRID COMPUTING

GRID TECHNOLOGY

- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resource
- Grid architecture
  - Defined using services and protocols
  - Using the "sand hourglass" model similar to the TCP/IP protocol stack

"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities."

Grid computing makes it possible to dynamically share and coordinate dispersed, heterogeneous computing resources. Flexibility and ubiquity are essential characteristics of Web services technologies such as WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery, and Integration).

The Open Grid Services Architecture (OGSA) combines technologies to unlock and exploit grid-attached resources. OGSA defines mechanisms to create, manage, and exchange information between Grid Services, a special type of Web service. The architecture uses WSDL extensively to describe the structure and behavior of a service. Service descriptions are located and discovered using Web Services Inspection Language (WSIL). By combining elements from grid computing and Web services technologies, OGSA establishes an extensible and interoperable design and development framework for Grid Services that includes details for service definition, discovery, and life-cycle management.

GLOBUS TOOLKIT

The Globus Toolkit provides software tools to make it easier to build computational grids and grid-based applications. The Globus Toolkit is both an open architecture and open source toolkit.

- The Globus Toolkit is a product of the Globus Alliance (http://www.globus.org)
- It is middleware for developing grids
- The current release is 6.0.
- Four key protocols and APIs
  - Grid Security Infrastructure (GSI)
  - Grid Resource Allocation & Mgmt (GRAM)
  - Grid Resource Information Protocol (GRIP) and Index Information Protocol (GIIP)
  - Grid File Transfer Protocol (GridFTP)
- Implementations on many platforms
  - Resources, security systems, data models,
- Various collective layer protocols & tools
  - Info services, replica management, etc.
- A basis for many Grid-enabled tools & apps

  FTP, SSH, Condor, SRB, MPI, EDG, GridPort,

GLOBUS TOOLKIT™ COMPONENTS

- Security
  - ❑ GSI – Grid Security Infrastructure
- Resource Management
  - ❑ GRAM – Grid Resource Allocation Manager
    - ○ globusrun
    - ○ RSL
    - ○ gatekeeper
    - ○ job manager
    - ○ DUROC – Dynamically-Updated Request Online Coallocator
- Information Services
  - ❑ MDS – Monitoring and Discovery Service

    - GRIS – Grid Resource Information Service
    - GIIS – Grid Index Information Service
    - MDS Client
- Data Management
    - GridFTP, GASS

The Globus Toolkit latest version 6.0 includes:

GSI: security

GridFTP: file transfer

GRAM: job execution/resource management

MyProxy: credential repository/certificate authority

GSI-OpenSSH: GSI secure single sign-on remote shell

GLOBUS TOOLKIT INSTALLATION PROCEDURE

Step1: For installing Globus Toolkit nothing more easy than download the latest package:

http://toolkit.globus.org/ftppub/gt6/installers/repo/globus-toolkit-repo_latest_all.deb

sudodpkg -iglobus-toolkit-repo_latest_all.deb

Step2: Update the repositories: sudo apt-get update

Step3: To install Debian or Ubuntu package, download the globus-toolkit-repo package

from the link above and install it with the command:

root@sysa63:/home/#dpkg -iglobus-toolkit-repo_latest_all.deb

Do the following for Debian-based systems:

root@sysa63:/home/#apt-get install globus-data-management-client

Expt.No. 01: DEVELOP A NEW WEB SERVICE FOR CALCULATOR

Aim:

Develop a Web Service for new Calculator

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

1. Our first web service is an extremely simple Math Web Service, which we'll refer to asMathService. It will
2. allow users to perform the following operations:
   a. Addition
   b. Subtraction
3. Furthermore, MathService will have the following resource properties (RPs for short):
   a. Value (integer)
   b. Last operation performed (string)
4. We will also add a "Get Value" operation to access the Value RP. Once a new resource is created, the "value" RP is initialized to zero, and the "last operation" RP is initialized to "NONE". The parameter is added/subtracted to the "value" RP, and the "last operation" RP is changed to "ADDITION" or "SUBTRACTION" accordingly. Also, the addition and subtraction operations don't return anything.
5. Writing and deploying a WSRF Web Service is easier than you might think. You just have to follow five simple steps.
6. Define the service's interface. This is done with WSDL
7. Implement the service. This is done with Java.
8. Define the deployment parameters. This is done with WSDD and JNDI
9. Compile everything and generate a GAR file. This is done with Ant
10. Deploy service. This is also done with a GT4 tool

Sample Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitionsname="MathService"targetNamespace="http://www.globus.org/namespaces/examples/core/MathService
_instance"
xmlns ="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ResourceProperties-1.2-draft-01.xsd" xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
```

```
ResourceProperties-1.2-draft-01.wsdl"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
</definitions>
```

Result**:**

Thus the program for developing web service for new calculator was successfully executed.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Grid by using Web Service for new Calculator

Applications:

To make a calculator in Windows application using web service

Viva-voce

1) What is Web Service?

2) What are the advantages of web services?

3) What are the different types of web services?

4) What is SOAP?

5) What are the advantages of SOAP web services?

6) What are the disadvantages of SOAP web services?

7) What is WSDL?

8) What is UDDI?

9) What are RESTful web services?

10) What are the advantages of RESTful web services?

11) What is the difference between SOAP and REST web services?

12) What is SOA service architecture?

13) What tools are used to deploy web services?

14) What is Web Service architecture?

15) What are the advantages of web services in globus toolkit?

16) What are the different types of web services?

17) What is SOAP web services?

18) What is UDP and TCP?

19) What is stateless SOA?

20) What is data balancing in WSDL?

Expt.No. 02: DEVELOP NEW OGSA-COMPLIANT WEB SERVICE

Aim:

To develop a new OGSA-Compliant Web service in Grid Service using .NET language.

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

1. Developed by the Global Grid Forum
2. Aims to define a common, standard, open architecture for Grid Applications.
3. Defines a set of rules that make up a grid service.
4. Sharing and Coordinated use of diverse resources in Dynamic VO's

Result:

Thus the program for developing OGSA- Complaint web service was successfully executed.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Grid to develop a new OGSA-Compliant Web service in Grid Service using .NET language

Applications:

- Sharing of information among diverse components of large heterogenous grid systems
- WAN

Viva-voce

1. How would you decide what style of Web Service to use? SOAP WS or REST?
2. Does the service expose data or business logic?
3. Do consumers and the service providers require a formal contract?
4. Do we need to support multiple data formats?
5. Do we need to make AJAX calls?
6. Is the call synchronous or asynchronous?
7. What level of security is required?
8. What level of transaction support is required?
9. Do we have limited band width?
10. What tools do you use to test your Web Services?
11. What is the difference between SOA and a Web service?
12. What is a microservice architecture (aka MSA)?
13. What is data integrity?
14. What is cloud infrastructure?

Expt.No. 03: USING APACHE AXIS DEVELOP A GRID SERVICE

Aim:

To develop a Grid service using Apache Axis

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

1. Creating the New Level in the Package
2. Edit the Configuration Files
3. Modify the Service Code
4. Modify the Client
5. Compile and Deploy
6. Starting the Container
7. Compile the Client
8. Run the Client

Sample Output:

```
Addition was successful
Subtraction was successful
Multiplication was successful
Division was successful
Current value: 20.
```

Result:

Thus the program for Grid Service using Apache Axis was successfully executed.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Grid to develop a Grid service using Apache Axis

Applications:

- To create web service in Java
- To work with Globus Tool kit standards web service

Viva-voce

1. What are the different application integration styles?
2. What is Grid Computing?
3. What is QOS?
4. What are the derivatives of grid computing?
5. What are the features of data grids?
6. What is load balancing?
7. What is grid infrastructure?
8. Define – Distributed Computing.
9. Define – OSGI.
10. Define – OSGA.
11. What is the use of API's in cloud services?
12. What is the difference between cloud and grid?
13. What is data communication?
14. What are the elements of data communication?

Expt.No. 04: DEVELOP APPLICATIONS USING JAVA OR C/C++ GRID APIS

Aim:

To develop an application in Java using Grid APIs

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

1. Import all the necessary java packages and name the file as GridLayoutDemo.java
2. Set up components to preferred size
3. Add buttons to experiment with Grid Layout
4. Add controls to set up horizontal and vertical gaps
5. Process the Apply gaps button press
6. Create the GUI
7. Create and set up the window,Set up the content pane and Display the Window
8. Schedule a job for the event dispatch thread
9. Show the application's GUI

Sample Output:

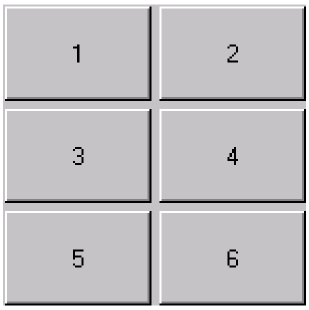


Figure 1: Horizontal, Left-to-Right

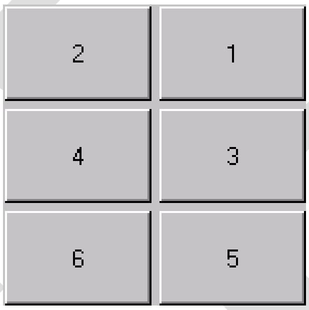


Figure 2: Horizontal, Right-to-Left

Result:

Thus the program to develop an application in java using Grid APIs was successfully executed.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Grid to develop an application in Java using Grid APIs

Applications:

Used for defining a application programme interface for common distributes computing functionality

Viva-voce

1) Describe the two most important advantages of using Distributed/Grid Computing.
2) What is the main function of DNS?
3) Briefly outline two applications of Public Key systems.
4) Provide one major difference between synchronous and asynchronous communication.
5) Provide one major difference between ASCII and Unicode encodings.
6) Provide two major differences between sequential computing and parallel computing.
7) Write the any three Grid Applications.
8) Give the examples of Hardware service provider.
9) Give the example of software application ASP.
10) What are grid portals? Give example.
11) What are the high level services including in existing globus tool kit?
12) Name the components available in Nimrod architecture?
13) What are the major objectives of Euro grid project?
14) What is the application specific work packages identified for the Euro grid?
15) Define dynamic accounting system.

EX.NO. 05: DEVELOP SECURED APPLICATIONS USING BASIC SECURITY MECHANISMS AVAILABLE IN GLOBUS TOOLKIT.

Aim:

To develop a secured applications using a basic security mechanisms available in Globus toolkit

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

The Globus Toolkit incorporates multiple security components that establish the identity of users or services (authentication), protect communications, and determine who is allowed to perform what actions (authorization), as well as manage user credentials.

- GSI C: The Globus Toolkit GSI C component provides APIs and tools for authentication, authorization and certificate management.
- MyProxy: MyProxy is open source software for managing X.509 Public Key Infrastructure (PKI) security credentials (certificates and private keys).
- GSI-OpenSSH: GSI-OpenSSH is a modified version of OpenSSH that adds support for X.509 proxy certificate authentication and delegation, providing a single sign-on remote login and file transfer service.

The Globus Toolkit GSI C component provides APIs and tools for authentication, authorization and certificate management. The authentication API is built using *Public Key*Infrastructure (PKI) technologies, e.g. X.509 Certificates and TLS. In addition to authentication it features a delegation mechanism based upon X.509 *Proxy Certificates*. Authorization support takes the form of a couple of APIs. The first provides a generic authorization API that allows callouts to perform access control based on the client's credentials (i.e. the X.509 certificate chain). The second provides a simple access control list that maps authorized remote entities to local (system) user names. The second mechanism also provides callouts that allow third parties to override the default behavior and is currently used in the Gatekeeper and GridFTP servers. In addition to the above there are various lower level APIs and tools for managing, discovering and querying certificates.

Components for Grid Security

1. Basic Security Mechanisms
2. Components for Credential Generation
3. Components for Credential Management

Beyond verifying the identities of users and services, basic Grid security mechanisms leave access control decisions to services. The Grid community has developed authorization and access control tools for storing and providing access to system-wide authorization information and for creating a central data store for supporting decentralized control mechanisms.

Basic Security Mechanisms

- Pre-Web Services Authentication and Authorization - A non-Web services implementation of the Grid Security Infrastructure (GSI), containing the core libraries and tools needed to secure applications using GSI mechanisms
- Web Services Authentication and Authorization - A Web services implementation of the Grid Security Infrastructure (GSI), containing the core libraries and tools needed to secure applications using GSI mechanisms

Result:

Thus the program to develop a security application available in Globus toolkit was successfully executed.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Grid to develop a secured applications using a basic security mechanisms available in Globus toolkit

Applications:

To provide easy acces to best breed open source network security

Viva-voce

1. What are the collective services available in grid computing?
2. What are the basic principles of autonomous computing?
3. What are the four essential characteristics of on demand business?
4. What are the essential capabilities provided by on demand business?
5. What are the two most important technologies for building semantic webs?
6. Define Peer to Peer computing?
7. What is the combination of Globus GT3 toolkit?
8. What is a GT3 core?
9. What are the components available in service model?
10. Define WS-Trust
11. Define WS –Federation
12. Name some representational use cases from OGSA architecture working group?
13. Who are the actors in CDC?
14. Mention the scenarios in CDC?
15. What are the functional requirements of CDC on OGSA?

Expt.No. 06: DEVELOP A GRID PORTAL, WHERE USER CAN SUBMIT A JOB AND GET THE RESULT, IMPLEMENT IT WITH AND WITHOUT GRAM CONCEPT.

Aim:

To develop a Grid portal and implement it with and without GRAM concept

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

Multiple times that the likely user interface to grid applications will be through portals, specifically Web portals. A grid portal may be constructed as a Web page interface to provide easy access to grid applications. The Web user interface provides user authentication, job submission, job monitoring, and results of the job.

Globus Resource Allocation Manager (GRAM)

When a job is submitted by a client, the request is sent to the remote host and handled by a gatekeeper daemon. The gatekeeper creates a job manager to start and monitor the job. When the job is finished, the job manager sends the status information back to the client and terminates.

The GRAM subsystem consists of the following elements:

- The **globusrun** command and associated APIs Resource Specification Language (RSL)
- The gatekeeper daemon The job manager Dynamically-Updated Request Online Coallocator (DUROC)

Each of these elements are described briefly below.

The globusrun command

The **globusrun** command (or its equivalent API) submits a job to a resource within the grid. This command is typically passed an RSL string (see below) that specifies parameters and other properties required to successfully launch and run the job

Resource Specification Language (RSL)

RSL is a language used by clients to specify the job to be run. All job submission requests are described in an RSL string that includes information such as the executable file; its parameters; information about redirection of stdin, stdout, and stderr; and so on. Basically it provides a standard way of specifying all of the information required to execute a job, independent of the target environment. It is then the responsibility of the job manager on the target system to parse the information and launch the job in the appropriate way.

The syntax of RSL is very straightforward. Each statement is enclosed within parenthesis. Comments are designated with parenthesis and asterisks, for example, (* this is a comment *). Supported attributes include the following:

rsl_substitution: Defines variables

executable: The script or command to be run

arguments: Information or flags to be passed to the executable

stdin: Specifies the remote URL and local file used for the executable stdout: Specifies the remote file to place standard output from the job stderr: Specifies the remote file to place standard error from the job queue: Specifies the queue to submit the job (requires a scheduler) count: Specifies the number of executions

directory: Specifies the directory to run the job

project: Specifies a project account for the job (requires a scheduler) dryRun: Verifies the RSL string but does not run the job

maxMemory: Specifies the maximum amount of memory in MBs required for the job

minMemory: Specifies the minimum amount of memory in MBs required for the job

hostCount: Specifies the number of nodes in a cluster required for the job environment: Specifies environment variables that are required for the job

jobType: Specifies the type of job single process, multi-process, mpi, or condor

maxTime: Specifies the maximum execution wall or cpu time for one execution

maxWallTime: Specifies the maximum walltime for one execution maxCpuTime: Specifies the maximum cpu time for one execution

gramMyjob: Specifies the whether the gram myjob interface starts one process/thread (independent) or more (collective)

Result:

Thus the program to develop Grid Portal was successfully executed.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Grid to develop a Grid portal and implement it with and without GRAM concept

Application:

- It is useful and necessary for interfaces for the performance of operations on the Grid. The Grid Portal Development Kit (GPDK) facilitates the development of Grid portals and provides several key reusable components for accessing various Grid services.
- A Grid Portal provides a customizable interface allowing scientists to perform a variety of Grid operations including remote program submission, file staging, and querying of information services from a single, secure gateway.
- The Grid Portal Development Kit leverages off existing Globus/Grid middleware infrastructure as well as commodity web technology including Java Server Pages and servlets .

Viva-voce

1) What are the expression evaluators supported in GT3?
2) What are the major goals of OGSA?
3) What are the more specific goals of OGSA?
4) What are the main purposes of use case defined by OGSA?
5) Name some representational use cases from OGSA architecture working group?
6) What are the layers available in OGSA architectural organizations?
7) What are the OGSA basic services?
8) What are the two aspects involved in GRAM?
9) What are the two kinds of lifecycle model associated with state data recovery?
10) What is a GT3 core?
11) What are the major components of default server side framework?
12) What is Grid container?
13) What are the two levels of security available in GT3?
14) What are the expression evaluators supported in GT3?
15) What are the two different message-level authentication mechanisms provided by GT3 framework?
16) What are the most common GT3 security handlers?

INTRODUCTION TO CLOUD COMPUTING

What is cloud computing?

Cloud computing means that instead of all the computer hardware and software you're using sitting on your desktop, or somewhere inside your company's network, it's provided for you as a service by another company and accessed over the Internet, usually in a completely seamless way. Exactly where the hardware and software is located and how it all works doesn't matter to you, the user—it's just somewhere up in the nebulous "cloud" that the Internet represents.

Cloud computing is a buzzword that means different things to different people. For some, it's just another way of describing IT (information technology) "outsourcing"; others use it to mean any computing service provided over the Internet or a similar network; and some define it as any bought-in computer service you use that sits outside your firewall.

Types of cloud computing

IT people talk about three different kinds of cloud computing, where different services are being provided for you. Note that there's a certain amount of vagueness about how these things are defined and some overlap between them.

- **Infrastructure as a Service (IaaS)** means you're buying access to raw computing hardware over the Net, such as servers or storage. Since you buy what you need and pay-as-you-go, this is often referred to as utility computing. Ordinary web hosting is a simple example of IaaS: you pay a monthly subscription or a per-megabyte/gigabyte fee to have a hosting company serve up files for your website from their servers.
- **Software as a Service (SaaS)** means you use a complete application running on someone else's system. Web-based email and Google Documents are perhaps the best-known examples. Zoho is another well-known SaaS provider offering a variety of office applications online.
- **Platform as a Service (PaaS)** means you develop applications using Web-based tools so they run on systems software and hardware provided by another company. So, for example, you might develop your own ecommerce website but have the whole thing, including the shopping cart, checkout, and payment mechanism running on a merchant's server. App Cloud (from salesforce.com) and the Google App Engine are examples of PaaS.

Advantages and disadvantages of cloud computing

Advantages

The pros of cloud computing are obvious and compelling. If your business is selling books or repairing shoes, why get involved in the nitty gritty of buying and maintaining a complex computer system? If you run an insurance office, do you really want your sales agents wasting time running anti-virus software, upgrading word-processors, or worrying about hard-drive crashes? Do you really want them cluttering your expensive computers with their personal emails, illegally shared MP3 files, and naughty YouTube videos—when you could leave that responsibility to someone else? Cloud computing allows you to buy in only the services you want, when you want them, cutting the upfront capital costs of computers and peripherals. You avoid equipment going out of date and other familiar IT problems like ensuring system security and reliability. You can add extra services (or take them away) at a moment's notice as your business needs change. It's really quick and easy to add new applications or services to your business without waiting weeks or months for the new computer (and its software) to arrive.

Drawbacks

Instant convenience comes at a price. Instead of purchasing computers and software, cloud computing means you buy services, so one-off, upfront capital costs become ongoing operating costs instead. That might work out much more expensive in the long-term.

If you're using software as a service (for example, writing a report using an online word processor or sending emails through webmail), you need a reliable, high-speed, broadband Internet connection functioning the whole time you're working. That's something we take for granted in countries such as the United States, but it's much more of an issue in developing countries or rural areas where broadband is unavailable.

An Introduction to Cloud Computing with OpenNebula

An OpenNebula Private Cloud provides infrastructure users with an elastic platform for fast delivery and scalability of services to meet dynamic demands of service end-users. Services are hosted in VMs, and then submitted, monitored and controlled in the Cloud by using Sunstone or any of the OpenNebula interfaces:

- Command Line Interface (CLI)
- XML-RPC API
- OpenNebulaRuby and Java Cloud APIs

The aim of a Private Cloud is not to expose to the world a cloud interface to sell capacity over the Internet, but to provide local cloud users and administrators with a flexible and agile private infrastructure to run virtualized service workloads within the administrative domain. OpenNebula virtual infrastructure interfaces expose user and administrator functionality for virtualization, networking, image and physical resource configuration, management, monitoring and accounting.

Expt.No.1: FIND PROCEDURE TO RUN THE VIRTUAL MACHINE OF DIFFERENT CONFIGURATION. CHECK HOW MANY VIRTUAL MACHINES CAN BE UTILIZED AT PARTICULAR TIME.

Aim:

To Find procedure to run the virtual machine of different configuration. Check how many virtual machines can be utilized at particular time

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

Creating Virtual Machines

In OpenNebula the Virtual Machines are defined with Template files. The Template Repository system allows OpenNebula administrators and users to register Virtual Machine definitions in the system, to be instantiated later as Virtual Machine instances. These Templates can be instantiated several times, and also shared with other users.

Virtual Machine Model

A Virtual Machine within the OpenNebula system consists of:

- A capacity in terms memory and CPU
- A set of NICs attached to one or more virtual networks
- A set of disk images
- A state file (optional) or recovery file, that contains the memory image of a running VM plus some hypervisor specific information.

The above items, plus some additional VM attributes like the OS kernel and context information to be used inside the VM, are specified in a template file.

Defining a VM in 3 Steps

Virtual Machines are defined in an OpenNebula Template. Templates are stored in a repository to easily browse and instantiate VMs from them. To create a new Template you have to define 3 things

- **Capacity & Name**, how big will the VM be?

Attribute	Description	Mandatory	Default
NAME	Name that the VM will get for description purposes.	Yes	one-<vmid>
MEMORY	Amount of RAM required for the VM, in Megabytes.	Yes	
CPU	CPU ratio (e..g half a physical CPU is 0.5).	Yes	
VCPU	Number of virtual cpus.	No	1

Disks. Each disk is defined with a DISK attribute. A VM can use three types of disk:

- **Use a persistent Image** changes to the disk image will persist after the VM is shutdown.
- **Use a non-persistent Image** images are cloned, changes to the image will be lost.
- **Volatile** disks are created on the fly on the target host. After the VM is shutdown the disk is disposed.

Persistent and Clone Disks

Attribute	Description	Mandatory	Default
IMAGE_ID and IMAGE	The ID or Name of the image in the datastore	Yes	
IMAGE_UID	Select the IMAGE of a given user by her ID	No	self
IMAGE_UNAME	Select the IMAGE of a given user by her NAME	No	self

Volatile

Attribute	Description	Mandatory	Default
TYPE	Type of the disk: swap, fs. swap type will set the label to swap so it is easier to mount and the context packages will automatically mount it.	Yes	
SIZE	size in MB	Yes	
FORMAT	filesystem for **fs** images: ext2, ext3, etc. raw will not format the image. For VMs to run on vmfs or vmwareshared configurations, the valid values are:	Yes	

Attribute	Description	Mandatory	Default
	vmdk_thin, vmdk_zeroedthick, vmdk_eagerzeroedthick		

Network Interfaces. Each network interface of a VM is defined with the NIC attribute.

Attribute	Description	Mandatory	Default
NETWORK_ID and NETWORK	The ID or Name of the image in the datastore	Yes	
NETWORK_UID	Select the IMAGE of a given user by her ID	No	Self
NETWORK_UNAME	Select the IMAGE of a given user by her NAME	No	Self

The following example shows a VM Template file with a couple of disks and a network interface, also a VNC section was added.

```
NAME   = test-vm
MEMORY = 128
CPU    = 1

DISK = [ IMAGE  = "Arch Linux" ]
DISK = [ TYPE     = swap,
       SIZE     = 1024 ]

NIC = [ NETWORK = "Public", NETWORK_UNAME="oneadmin" ]

GRAPHICS = [
  TYPE    = "vnc",
LISTEN  = "0.0.0.0"]
```

Simple templates can be also created using the command line instead of creating a template file. The parameters to do this for onetemplate are:

Parameter	Description
--namename	Name for the VM
--cpucpu	CPU percentage reserved for the VM (1=100% one CPU)

Parameter	Description
--vcpuvcpu	Number of virtualized CPUs
--archarch	Architecture of the VM, e.g.: i386 or x86_64
--memorymemory	Memory ammount given to the VM
--diskdisk0,disk1	Disks to attach. To use a disk owned by other user use user[disk]
--nicvnet0,vnet1	Networks to attach. To use a network owned by other user use user[network]
--rawstring	Raw string to add to the template. Not to be confused with the RAW attribute. If you want to provide more than one element, just include an enter inside quotes, instead of using more than one -raw option
--vnc	Add VNC server to the VM
--ssh[file]	Add an ssh public key to the context. If the file is omited then the user variable SSH_PUBLIC_KEY will be used.
--net_context	Add network contextualization parameters
--contextline1,line2 *	Lines to add to the context section
--bootdevice	Select boot device (hd, fd, cdrom or network)

A similar template as the previous example can be created with the following command:

```
$ onetemplate create --name test-vm --memory 128 --cpu 1 --disk "Arch Linux" --nic Public
```

Managing Virtual Machines

Assuming we have a VM Template registered called *vm-example* with ID 6, then we can instantiate the VM issuing a:

```
$ onetemplate list

  ID USER     GROUP    NAME                     REGTIME
   6 oneadminoneadminvm_example           09/28 06:44:07

$ onetemplate instantiate vm-example --name my_vm
```

```
VM ID: 0
```

If the template has *USER INPUTS*defined the CLI will prompt the user for these values:

```
$ onetemplate instantiate vm-example --name my_vm

There are some parameters that require user input.

  * (BLOG_TITLE) Blog Title: <my_title>
  * (DB_PASSWORD) Database Password:
VM ID: 0
```

Afterwards, the VM can be listed with the onevmlist command. You can also use the onevmtop command to list VMs continuously.

```
$ onevm list

    ID USER     GROUP    NAME        STAT CPU     MEM        HOSTNAME        TIME
     0 oneadminoneadminmy_vm        pend   0      0K                00 00:00:03
```

After a Scheduling cycle, the VM will be automatically deployed. But the deployment can also be forced by oneadmin using onevmdeploy:

```
$ onehost list

  ID NAME              RVM   TCPU   FCPU   ACPU   TMEM   FMEM   AMEM   STAT
   2 testbed            0    800    800    800    16G    16G16G     on
$ onevm deploy 0 2

$ onevm list

    ID USER     GROUP    NAME        STAT CPU     MEM        HOSTNAME        TIME
     0 oneadminoneadminmy_vmrunn   0      0K        testbed 00 00:02:40

$ onevm show 0

VIRTUAL MACHINE 0 INFORMATION

            ID :0

             NAME                         : my_vm

             USER              : oneadmin

             GROUP              : oneadmin
```

```
        STATE             : ACTIVE
        LCM_STATE         : RUNNING
        START TIME        : 04/14 09:00:24
        END TIME          : -
        DEPLOY ID:         : one-0
        PERMISSIONS
                OWNER         : um-
                GROUP         : ---
                OTHER         : ---
VIRTUAL MACHINE MONITORING
        NET_TX                  : 13.05
        NET_RX                  : 0
        USED MEMORY             : 512
        USED CPU                              : 0
VIRTUAL MACHINE TEMPLATE
VIRTUAL MACHINE HISTORY

 SEQ       HOSTNAME REASON        START       TIME      PTIME

  0       testbednone  09/28 06:48:18 00 00:07:23 00 00:00:00
        testbednone  09/28 06:48:18 00 00:07:23 00 00:00:00
```

Result:

Thus the program to run virtual machines on different configuration was successfully executed & its utilization time is checked in various machines.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to find procedure to run the virtual machine of different Configuration. Check how many virtual machines can be utilized at particular time

Applications:

- An advantage of virtualizing the workload's I/O path enables hardware independence by abstracting vendor specific drivers to more generalized versions that run on the hypervisor.
- It allows the live migration, which is one of virtualization's greatest availability strengths
- The sharing of aggregate resources, such as network paths

Viva-voce

1. What is cloud computing?
2. What are the benefits of cloud computing?
3. What are the different data types used in cloud computing?
4. What are the different layers in cloud computing?
5. What do you mean by software as a service?
6. What is on-demand functionality? How is it provided in cloud computing?
7. What are the platforms used for large scale cloud computing?
8. What are the different models for deployment in cloud computing?
9. What is the difference between cloud computing and mobile computing?
10. What are the open source cloud computing platform databases?
11. What is the difference between cloud and traditional datacenters?
12. Why API's is used in cloud services?
13. What are the different datacenters in cloud computing?
14. Define API in data virtualization?
15. List of API in cloud?

Expt.No.02: FIND PROCEDURE TO ATTACH VIRTUAL BLOCK TO THE VIRTUAL MACHINE AND CHECK WHETHER IT HOLDS THE DATA EVEN AFTER THE RELEASE OF THE VIRTUAL MACHINE.

Aim:

To find a procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

VirtualBox uses a special kernel module called vboxdrv to perform physical memory allocation and to gain control of the processor for guest system execution. Without this kernel module, you can still use the VirtualBox manager to configure virtual machines, but they will not start. In addition, there are the network kernel modules vboxnetflt and vboxnetadp which are required for the more advanced networking features of VirtualBox.

The VirtualBox kernel module is automatically installed on your system when you install VirtualBox. To maintain it with future kernel updates, for those Linux distributions which provide it -- most current ones -- we recommend installing Dynamic Kernel Module Support (DKMS). This framework helps with building and upgrading kernel modules.

If DKMS is not already installed, execute one of the following:

On an Ubuntu system:

```
sudo apt-get install dkms
```

If DKMS is available and installed, the VirtualBox kernel module should always work automatically, and it will be automatically rebuilt if your host kernel is updated.

Otherwise, there are only two situations in which you will need to worry about the kernel module:

1. The original installation fails. This probably means that your Linux system is not prepared for building external kernel modules.

Most Linux distributions can be set up simply by installing the right packages - normally, these will be the GNU compiler (GCC), GNU Make (make) and packages containing header files for your kernel - and making sure that all system updates are installed and that the system is running the most up-to-date kernel included in the distribution. *The version numbers of the header file packages must be the same as that of the kernel you are using.*

- With Debian and Ubuntu releases, you must install the right version of the linux-headers and if it exists the linux-kbuild package. Current Ubuntu releases should have the right packages installed by default.
- In even older Debian and Ubuntu releases, you must install the right version of the kernel-headers package.
- On Fedora and Redhat systems, the package is kernel-devel.
- On SUSE and openSUSE Linux, you must install the right versions of the kernel-source and kernel-syms packages.
- If you have built your own kernel, you will need to make sure that you also installed all the required header and other files for building external modules to the right locations. The details of how to do this will depend on how you built your kernel, and if you are unsure you should consult the documentation which you followed to do so.

2. The kernel of your Linux host was updated and DKMS is not installed. In that case, the kernel module will need to be reinstalled by executing (as root):

```
rcvboxdrv setup
```

Performing the installation

VirtualBox is available in a number of package formats native to various common Linux distributions. In addition, there is an alternative generic installer (.run) which should work on most Linux distributions.

Installing VirtualBox from a Debian/Ubuntu package

First, download the appropriate package for your distribution. The following examples assume that you are installing to a 32-bit Ubuntu Wily system. Use dpkg to install the Debian package:

sudodpkg -i virtualbox-5.0_5.0.20_Ubuntu_wily_i386.deb

The installer will also try to build kernel modules suitable for the current running kernel. If the build process is not successful you will be shown a warning and the package will be left unconfigured. Please have a look at /var/log/vbox-install.log to find out why the compilation failed. You may have to install the appropriate Linux kernel After correcting any problems, do

sudorcvboxdrv setup

This will start a second attempt to build the module.

If a suitable kernel module was found in the package or the module was successfully built, the installation script will attempt to load that module.

Once VirtualBox has been successfully installed and configured, you can start it by selecting "VirtualBox" in your start menu or from the command line.

Using the alternative installer (VirtualBox.run)

The alternative installer performs the following steps:

- It unpacks the application files to the target directory,

  /opt/VirtualBox/

  which cannot be changed.

- It builds the VirtualBox kernel modules (vboxdrv, vboxnetflt and vboxnetadp) and installs them.

- It creates /sbin/rcvboxdrv, an init script to start the VirtualBox kernel module.

- It creates a new system group called vboxusers.

- It creates symbolic links in /usr/bin to the a shell script (/opt/VirtualBox/VBox) which does some sanity checks and dispatches to the actual executables, VirtualBox, VBoxSDL, VBoxVRDP, VBoxHeadless and VBoxManage

- It creates /etc/udev/rules.d/60-vboxdrv.rules, a description file for udev, if that is present, which makes the USB devices accessible to all users in the vboxusers group.

- It writes the installation directory to /etc/vbox/vbox.cfg.

The installer must be executed as root with either install or uninstall as the first parameter.

sudo ./VirtualBox.run install

Or if you do not have the "sudo" command available, run the following as root instead:

./VirtualBox.run install

After that you need to put every user which should be able to access USB devices from VirtualBox guests in the group vboxusers, either through the GUI user management tools or by running the following command as root:

```
sudousermod -a -G vboxusers username
```

Performing a manual installation

If, for any reason, you cannot use the shell script installer described previously, you can also perform a manual installation. Invoke the installer like this:

```
./VirtualBox.run --keep --noexec
```

This will unpack all the files needed for installation in the directory install under the current directory. The VirtualBox application files are contained in VirtualBox.tar.bz2 which you can unpack to any directory on your system. For example:

```
sudomkdir /opt/VirtualBox
sudo tar jxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
```

or as root:

```
mkdir /opt/VirtualBox

tarjxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
```

The sources for VirtualBox's kernel module are provided in the src directory. To build the module, change to the directory and issuemake

If everything builds correctly, issue the following command to install the module to the appropriate module directory:

```
sudo make install
```

In case you do not have sudo, switch the user account to root and performmake install

The VirtualBox kernel module needs a device node to operate. The above make command will tell you how to create the device node, depending on your Linux system. The procedure is slightly different for a classical Linux setup with a /dev directory, a system with the now deprecated devfs and a modern Linux system with udev.

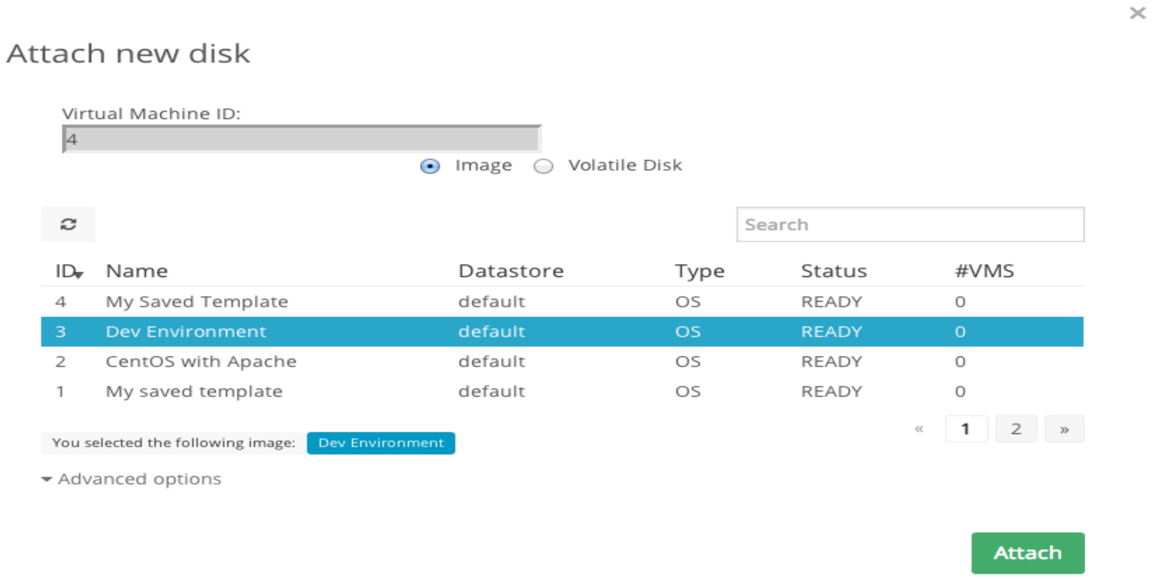
Note that the /dev/vboxdrv kernel module device node must be owned by root:root and must be read/writable only for the user.Next, you will have to install the system initialization script for the kernel module:

```
cp /opt/VirtualBox/vboxdrv.sh /sbin/rcvboxdrv
mkdir /etc/vbox
echo INSTALL_DIR=/opt/VirtualBox> /etc/vbox/vbox.cfg
```

and, for convenience, create the following symbolic links:

```
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VirtualBox
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxManage
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxHeadless
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxSDL
```

Sample Output:



Attach new disk

Virtual Machine ID: 4

● Image ○ Volatile Disk

ID	Name	Datastore	Type	Status	#VMS
4	My Saved Template	default	OS	READY	0
3	Dev Environment	default	OS	READY	0
2	CentOS with Apache	default	OS	READY	0
1	My saved template	default	OS	READY	0

You selected the following image: Dev Environment

Result:

Thus the program to attach virtual block to virtual machine was successfully executed & checked whether it holds the data after the release of the virtual machine.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to find a procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine

Application:

- To support the Cloud Computing with the Virtual Block Store System. The fast development of cloud computing systems stimulates the needs for a standalone block storage system to provide persistent block storage services to virtual machines maintained by clouds.

Viva-voce

1. What is Type-1 and Type-2 hypervisor?
2. What is the use of virsh command?
3. Explain how you can transfer volume from one owner to another in Open Stack?
4. Discuss about KVM Features?
5. What is Virtual block?
6. What is Virtualization?
7. What are virtual clusters?
8. How virtualization happens in data center?
9. List some of the open source grid middleware packages.
10. What is a programming model?
11. What is Hadoop?
12. What are map and reduce functions?
13. How to run a job in hadoop?
14. What is HDFS?
15. What are the open source grid middleware packages?
16. What is peer to peer computing?
17. What are node clusters?
18. How virtual migration happens in data center?

Expt.No.03: INSTALL A C COMPILER IN THE VIRTUAL MACHINE AND EXECUTE A SAMPLE PROGRAM.

Aim:

To Install a C compiler in the virtual machine and execute a sample program

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

1. Create a VM and Install any OS on it.
2. Install a C compiler on OS
3. Open a Editor and type the sample program and Save.
4. Compile and run the Program

Example:

C programming on Linux based Environement

1. Open Terminal (Applications-Accessories-Terminal)
2. Open gedit by typing “gedit&” on terminal(You can also use any other Text Editor application)
3. Type the following on gedit
   (or any other text editor)

```
#include<stdio.h>
main()
{
printf("Hello World\n");
}
```

Save this file as
“helloworld. C”

5. Type “ls” on Terminal to see all files under current folder
6. Confirm that “helloworld.c” is in the current directory.
If not, type cd DIRECTORY_PATH to go to the
directory that has “helloworld.c”
7. Type “gcchelloworld.c” to compile, and type “ls” to
confirm that a new executable file “a.out” is created
8. Type “./a.out” on Terminal to run the program

Result:

Thus the program to install a C complier is done and the sample program was executed successfully.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to Install a C compiler in the virtual machine and execute a sample program

Application:

The application of installing a C compiler in the virtual machine benefits to have a portable executable, for each platform.

Viva-voce

1. What is the difference between Xen & KVM ?

2. What are different hypervisors available in Linux ?

3. Which command is used to list all virtual machine running on the KVM hypervisor ?

4. What are the different states of a VM in Xen hypervisor ?

5. How to forcefully shutdown the KVM based virtual machine from the command line ?

6. What is virtual Machine?

7. What is Compiler?

8. What is Directory?

9. List the C Compilers available?

10. What is virtualization?

11. List out VM operations

12. What are different hypervisors available in windows?

13. How to implement applications on the cloud to install C compiler?

14. What are grid and cloud tools available?

15. What is hypervisor?

Expt.No.04: SHOW THE VIRTUAL MACHINE MIGRATION BASED ON THE CERTAIN CONDITION FROM ONE NODE TO THE OTHER.

Aim:

To Show the virtual machine migration based on the certain condition from one node to the other

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

1.Update source and target virtual machines to latest package versions

To help ensure that platform images between cloud providers are running the same version of key operating system packages, update these packages to the latest versions on **both source and target** virtual machines.

sudo apt-get update

sudo apt-get upgrade

2. **Install rsync and screen packages on source and target virtual machines**

The migration of application packages and files in this process will use rsync over ssh between source and target virtual machines. The actual transfer of files between virtual machines can take some time, so I also recommend using screen so that you can easily re-attach to an in-progress migration session if you are inadvertently disconnected.

Ensure that rsync and screen packages are installed on both the source and target virtual machines with these commands:

sudo apt-get install rsync

sudo apt-get install screen

3. Add a consistent user account to both source and target virtual machines

To facilitate the migration process, ensure that you have a consistent user account configured on both source and target virtual machines with sudo enabled. The newly provisioned target virtual machines from **Task 3** already include a user named **azureuser** with sudo enabled. To configure this same user on **each source virtual machine**, use the following commands:

```
sudogroupadd -g 500 azureuser

sudouseradd -u 500 -g 500 -m -s /bin/bash azureuser

sudopasswdazureuser
```

4. Start a screen session for the migration

On the source virtual machine, enter a new screen session for the migration by using the following command:

```
sudo screen -S AzureMigration
```

If you are disconnected from the source virtual machine during the migration process, you can reconnect to the detached screen session by using the following command after signing in again to the source virtual machine:

```
sudo screen -r
```

5. Build an exclusion list of directories and files

During the migration, we want to be careful to skip any files that include configuration information relating to the identity of the source virtual machines, such as IP addresses, hostnames, ssh keys, etc. For the Ubuntu-based virtual machines that we migrated, we used the following commands on **each source virtual machine** to build our list of directories and files to exclude from the migration process:

```
EXCLUDEFILE=/tmp/exclude.file

EXCLUDELIST='/boot /etc/fstab /etc/hostname /etc/issue /etc/hosts
/etc/sudoers /etc/networks /etc/network/* /etc/resolv.conf
/etc/ssh/* /etc/sysctl.conf /etc/mtab /etc/udev/rules.d/*
/lock /net /tmp'

EXCLUDEPATH=$(echo $EXCLUDELIST | sed 's/\ /\\n/g')

echo -e $EXCLUDEPATH > $EXCLUDEFILE

find / -name '*cloud-init*' >> $EXCLUDEFILE

find / -name '*cloud-config*' >> $EXCLUDEFILE

find / -name '*cloud-final*' >> $EXCLUDEFILE
```

The actual list of directories and files that you exclude may vary from this list, based on the Linux distro version, packages and applications that you are migrating.

Credits: Kudos to Kevin Carter who wrote a great article a couple years ago that provided a useful starting point for building a list of directories and files to consider excluding as part of a Linux-to-Linux migration process!

6. Stop applications during migration

To minimize application data changes from occurring during the migration process, stop the related applications and daemons on the **source virtual machines**. The application that we migrated was a web application built using Apache2, so we simply stopped the related Apache2 daemon.

```
sudo service stop apache2
```

7. Migrate the application files and data

From **each source virtual machine**, migrate application files and data using two rsync passes over ssh. The first pass performs the bulk of the data transfer, whereas the second pass uses checksums to confirm that all files were transferred successfully.

```
    TARGETVM="insert_target_vm_public_ip_address"

    rsync -e "ssh" -rlpEAXogDtSzh -P -x –exclude-from="$EXCLUDEFILE" –rsync-path="sudorsync" –verbose –progress / azureuser@$TARGETVM:/

    rsync -e "ssh" -crlpEAXogDtSzh -P -x –exclude-from="$EXCLUDEFILE" –rsync-path="sudorsync" –verbose –progress / azureuser@$TARGETVM:/
```

8. **Restart each target virtual machine**

After both rsync passes have completed, restart **each target virtual machine** to complete the migration process.

```
    sshazureuser@$TARGETVM

    shutdown -r now
```

Result:

Thus the program to implement migration of virtual machine was executed successfully.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to Show the virtual machine migration based on the certain condition from one node to the other

Application:

- Its easier to migrate a Virtual Machine from one server to another, than migrating the operating system and application(s) individually.
- Using Virtual Machine Migration, its possible to migrate operating systems and applications from older servers to newer servers easily and without disrupting the services.

Viva-voce

1. What are the basic requirements of VM live migration in KVM?

2. Which command is used in KVM for VMs live migration?

3. How to get hardware information of KVM guest machine?

4. What is VM migration?

5. Which type of virtualization is also characteristic of cloud computing ?

6. Which virtualization standard does the WebSphere Application Server Hypervisor Edition use?

7.Which three tools are included in the IBM Rational Jazz Collaborative Application Lifecycle Management (C/ALM) solution?

8. What are two ways a public cloud helps customers reduce their IT costs?

9.What functionality is provided by the IBM Security Network Intrusion Prevention System Virtual Appliance for a cloud environment?

10. What are two important benefits of using cloud computing?

11. What is the value of IBM Security Information and Event Manager in a cloud provider environment?

12. What is the main purpose of an IBM CloudBurst solution?

EX.NO.5: PROCEDURE TO INSTALL STORAGE CONTROLLER AND INTERACT WITH IT.

Aim:

Find procedure to install storage controller and interact with it

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

To install Storage Control, follow these steps.

Ensure that Systems Director 6.3 is installed and running.

Ensure that you are logged in to Systems Director with a user ID that has administrative privileges.

Windows only: Restart the DB2 Management server.

a) Go to **Start** >**Administrative Tools** >**Services**.

b) Select **DB2 Management Service** from the services window and restart it.

If you want to download and install Storage Control, go to step 5. If you want to install Storage Control from read-only media, such as a CD or mounted .iso image, go to step 8.

From the Systems Director summary page, click the link **Try Storage Control** in the upper right corner.

A download page opens. Download the appropriate file for your operating system.

7.Extract the files to the directory where you want to install Storage Control, then go to step 9.

8. Copy the Storage Control installer directory from the CD or the mounted .iso image into a temporary directory close to the system root. For example, /SCInstall for AIX or **Linux and C:\SCInstall for Windows.**

9**.** Access a command window and navigate to the directory where you extracted the files or where you copied the installer directory. Run the appropriate script. If you do not want the license agreement to display, use the **-i**option when you run the script. For example, StorageControlInstall.sh -i.

Important: If you are not using IBM DB2 managed by Systems Director, then the DB2 user ID used must have DB2 Administrator privileges.

On Microsoft Windows systems, run the script StorageControlInstall.bat.

On Linux and AIX systems, run the script StorageControlInstall.sh.

10. Restart Systems Director as directed.

Result:

Thus the program to install storage controller was executed successfully.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to find procedure to install storage controller and interact with it

Application:

A controller offers a level of abstraction between an operating system and the physical drives. A RAID controller presents groups to applications and operating systems as logical units for which data protection schemes can be defined.

Viva-Voce

1. What is a snapshot?
2. What is Thick Provision Lazy Zeroed?
3. What is Thick Provision Eager Zeroed?
4. What is Thin Provision?
5. What is VDI?
6. What is storage vMotion?
7. How many virtual CPUs can I use on a Fault Tolerant virtual machine?
8. What is the use of vmware tools?
9. What happens if vCenter Server is offline when a failover event occurs?
10. What are the monitoring methods used for vSphere HA?
11. What are the roles of a master host in vSphere HA?
12. What is the hardware version used in VMware ESXi 5.5?
13. What is server virtualization?
14. What is Network Virtualization?
15. What is template?

EX.NO.6: PROCEDURE TO SET UP ONE HADOOP CLUSTER

Aim:

To Find a procedure to set up the one node Hadoop cluster

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

Follow the steps given below to have Hadoop Multi-Node cluster setup.

Installing Java

Java is the main prerequisite for Hadoop. First of all, you should verify the existence of java in your system using “java -version”. The syntax of java version command is given below.

```
$ java-version
```

If everything works fine it will give you the following output.

```
java version "1.7.0_71"
Java(TM) SE RuntimeEnvironment(build 1.7.0_71-b13)
JavaHotSpot(TM)Client VM (build 25.0-b02, mixed mode)
```

If java is not installed in your system, then follow the given steps for installing java.

1. Download java (JDK - X64.tar.gz) by visiting the following link http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html
Then jdk-7u71-linux-x64.tar.gz will be downloaded into your system.

2. Generally you will find the downloaded java file in Downloads folder. Verify it and extract the jdk-7u71-linux-x64.gz file using the following commands**.**

```
$ cdDownloads/
$ ls
jdk-7u71-Linux-x64.gz
$ tar zxf jdk-7u71-Linux-x64.gz
$ ls
jdk1.7.0_71 jdk-7u71-Linux-x64.gz
```

3. To make java available to all the users, you have to move it to the location "/usr/local/". Open the root, and type the following commands.

```
$ su
password:
# mv jdk1.7.0_71 /usr/local/
# exit
```

4. For setting up PATH and JAVA_HOME variables, add the following commands to ~/.bashrc file.

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
export PATH=PATH:$JAVA_HOME/bin
```

Now verify the **java -version** command from the terminal as explained above. Follow the above process and install java in all your cluster nodes.

Creating User Account

Create a system user account on both master and slave systems to use the Hadoop installation.

```
# useraddhadoop
# passwdhadoop
```

Mapping the nodes

You have to edit **hosts** file in **/etc/** folder on all nodes, specify the IP address of each system followed by their host names.

```
# vi /etc/hosts
```

enter the following lines in the /etc/hosts file.

```
192.168.1.109hadoop-master
192.168.1.145 hadoop-slave-1
192.168.56.1 hadoop-slave-2
```

Configuring Key Based Login

Setup ssh in every node such that they can communicate with one another without any prompt for password.

```
# suhadoop
$ ssh-keygen-t rsa
$ ssh-copy-id-i~/.ssh/id_rsa.pub tutorialspoint@hadoop-master
$ ssh-copy-id-i~/.ssh/id_rsa.pub hadoop_tp1@hadoop-slave-1
$ ssh-copy-id-i~/.ssh/id_rsa.pub hadoop_tp2@hadoop-slave-2
$ chmod0600~/.ssh/authorized_keys
$ exit
```

Installing Hadoop

In the Master server, download and install Hadoop using the following commands.

```
# mkdir /opt/hadoop
# cd /opt/hadoop/
# wget http://apache.mesi.com.ar/hadoop/common/hadoop-1.2.1/hadoop-1.2.0.tar.gz
# tar -xzf hadoop-1.2.0.tar.gz
# mv hadoop-1.2.0 hadoop
# chown -R hadoop /opt/hadoop
# cd /opt/hadoop/hadoop/
```

Configuring Hadoop

You have to configure Hadoop server by making the following changes as given below.

core-site.xml

Open the **core-site.xml** file and edit it as shown below.

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://hadoop-master:9000/</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
</configuration>
```

hdfs-site.xml

Open the **hdfs-site.xml** file and edit it as shown below.

```
<configuration>
<property>
<name>dfs.data.dir</name>
<value>/opt/hadoop/hadoop/dfs/name/data</value>
<final>true</final>
</property>

<property>
<name>dfs.name.dir</name>
<value>/opt/hadoop/hadoop/dfs/name</value>
<final>true</final>
</property>

<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

mapred-site.xml

Open the **mapred-site.xml** file and edit it as shown below.

```
<configuration>
```

```
<property>
<name>mapred.job.tracker</name>
<value>hadoop-master:9001</value>
</property>
</configuration>
```

hadoop-env.sh

Open the **hadoop-env.sh** file and edit JAVA_HOME, HADOOP_CONF_DIR, and HADOOP_OPTS as shown below.

Note: Set the JAVA_HOME as per your system configuration.

```
export JAVA_HOME=/opt/jdk1.7.0_17 export HADOOP_OPTS=-Djava.net.preferIPv4Stack=trueexport
HADOOP_CONF_DIR=/opt/hadoop/hadoop/conf
```

Installing Hadoop on Slave Servers

Install Hadoop on all the slave servers by following the given commands.

```
# suhadoop
$ cd/opt/hadoop
$ scp-r hadoop hadoop-slave-1:/opt/hadoop
$ scp-r hadoop hadoop-slave-2:/opt/hadoop
```

Configuring Hadoop on Master Server

Open the master server and configure it by following the given commands.

```
# suhadoop
$ cd/opt/hadoop/hadoop
```

Configuring Master Node

```
$ vietc/hadoop/masters
hadoop-master
```

Configuring Slave Node

```
$ vietc/hadoop/slaves
hadoop-slave-1
hadoop-slave-2
```

Format Name Node on Hadoop Master

```
# suhadoop
$ cd/opt/hadoop/hadoop
$ bin/hadoopnamenode–format
11/10/1410:58:07 INFO namenode.NameNode:
STARTUP_MSG:/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = hadoop-master/192.168.1.109
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 1.2.0
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.2 -r 1479473;
compiled by 'hortonfo' on Mon May 6 06:59:37 UTC 2013
```

```
STARTUP_MSG: java = 1.7.0_71 ************************************************************/11/10/1410:58:08 INFO
util.GSet:Computing capacity for map BlocksMapeditlog=/opt/hadoop/hadoop/dfs/name/current/edits
…………………………………………………….
…………………………………………………….
…………………………………………………….11/10/1410:58:08 INFO common.Storage:Storage directory
/opt/hadoop/hadoop/dfs/name has been successfully formatted.11/10/1410:58:08 INFO namenode.NameNode:
SHUTDOWN_MSG:/************************************************************ SHUTDOWN_MSG: Shutting down
NameNode at hadoop-master/192.168.1.15 ************************************************************/
```

Starting Hadoop Services

The following command is to start all the Hadoop services on the Hadoop-Master.

```
$ cd $HADOOP_HOME/sbin
$ start-all.sh
```

Adding a New DataNode in the Hadoop Cluster

Given below are the steps to be followed for adding new nodes to a Hadoop cluster.

Networking

Add new nodes to an existing Hadoop cluster with some appropriate network configuration. Assume the following network configuration.

For New node Configuration:

```
IP address :192.168.1.103
netmask:255.255.255.0
hostname : slave3.in
```

Adding User and SSH Access

Add a User

On a new node, add "hadoop" user and set password of Hadoop user to "hadoop123" or anything you want by using the following commands.

```
useraddhadoop
passwdhadoop
```

Setup Password less connectivity from master to new slave.

Execute the following on the master

```
mkdir-p $HOME/.ssh
chmod700 $HOME/.ssh
ssh-keygen-t rsa-P ''-f $HOME/.ssh/id_rsa
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
chmod644 $HOME/.ssh/authorized_keys
```

Copy the public key to new slave node inhadoop user $HOME directory

```
scp $HOME/.ssh/id_rsa.pub hadoop@192.168.1.103:/home/hadoop/
```

Execute the following on the slaves

Login to hadoop. If not, login to hadoop user.

```
suhadoopssh-X hadoop@192.168.1.103
```

Copy the content of public key into file **"$HOME/.ssh/authorized_keys"** and then change the permission for the same by executing the following commands.

```
cd $HOME
mkdir-p $HOME/.ssh
chmod700 $HOME/.ssh
cat id_rsa.pub >>$HOME/.ssh/authorized_keys
chmod644 $HOME/.ssh/authorized_keys
```

Check ssh login from the master machine. Now check if you can ssh to the new node without a password from the master.

```
ssh hadoop@192.168.1.103or hadoop@slave3
```

Set Hostname of New Node

You can set hostname in file **/etc/sysconfig/network**

Onnew slave3 machine

```
NETWORKING=yes
HOSTNAME=slave3.in
```

To make the changes effective, either restart the machine or run hostname command to a new machine with the respective hostname (restart is a good option).

On slave3 node machine:

```
hostname slave3.in
```

Update **/etc/hosts** on all machines of the cluster with the following lines:

```
192.168.1.102 slave3.in slave3
```

Now try to ping the machine with hostnames to check whether it is resolving to IP or not.

On new node machine:

```
ping master.in
```

Start the DataNode on New Node

Start the datanode daemon manually using **$HADOOP_HOME/bin/hadoop-daemon.sh script**. It will automatically contact the master (NameNode) and join the cluster. We should also add the new node to the conf/slaves file in the master server. The script-based commands will recognize the new node.

Login to new node

```
suhadooporssh-X hadoop@192.168.1.103
```

Start HDFS on a newly added slave node by using the following command

```
./bin/hadoop-daemon.sh start datanode
```

Check the output of jps command on a new node. It looks as follows.

```
$ jps
7141DataNode
10312Jps
```

Removing a DataNode from the Hadoop Cluster

We can remove a node from a cluster on the fly, while it is running, without any data loss. HDFS provides a decommissioning feature, which ensures that removing a node is performed safely. To use it, follow the steps as given below:

1. Login to master

Login to master machine user where Hadoop is installed.

```
$ suhadoop
```

2 Change cluster configuration

An exclude file must be configured before starting the cluster. Add a key named dfs.hosts.exclude to our **$HADOOP_HOME/etc/hadoop/hdfs-site.xml** file. The value associated with this key provides the full path to a file on the NameNode's local file system which contains a list of machines which are not permitted to connect to HDFS.

For example, add these lines to **etc/hadoop/hdfs-site.xml** file.

```
<property>
<name>dfs.hosts.exclude</name>
<value>/home/hadoop/hadoop-1.2.1/hdfs_exclude.txt</value>

<description>DFS exclude</description>
</property>
```

3. Determine hosts to decommission

Each machine to be decommissioned should be added to the file identified by the hdfs_exclude.txt, one domain name per line. This will prevent them from connecting to the NameNode. Content of the **"/home/hadoop/hadoop-1.2.1/hdfs_exclude.txt"** file is shown below, if you want to remove DataNode2.slave2.in

4. Force configuration reload

Run the command **"$HADOOP_HOME/bin/hadoopdfsadmin -refreshNodes"** without the quotes.

```
$ $HADOOP_HOME/bin/hadoopdfsadmin-refreshNodes
```

This will force the NameNode to re-read its configuration, including the newly updated 'excludes' file. It will decommission the nodes over a period of time, allowing time for each node's blocks to be replicated onto machines which are scheduled to remain active.On **slave2.in**, check the jps command output. After some time, you will see the DataNode process is shutdown automatically.

5. Shutdown nodes

After the decommission process has been completed, the decommissioned hardware can be safely shut down for maintenance. Run the report command to dfsadmin to check the status of decommission. The following command will describe the status of the decommission node and the connected nodes to the cluster.

```
$ $HADOOP_HOME/bin/hadoopdfsadmin-report
```

6. Edit excludes file again

Once the machines have been decommissioned, they can be removed from the ‘excludes’ file. Running **"$HADOOP_HOME/bin/hadoopdfsadmin -refreshNodes"** again will read the excludes file back into the NameNode; allowing the DataNodes to rejoin the cluster after the maintenance has been completed, or additional capacity is needed in the cluster again, etc.

Result:

Thus the program to install storage controller was executed successfully.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to find a procedure to set up the one node Hadoop cluster

Application:

Many organisations are required to deal with large data sets. To handle the large data sets these organisations use hadoop cluster. But they need to set up hadoop cluster with different number of nodes several times.

Viva-Voce

1) What is Big Data?
2) What are the four characteristics of Big Data?
3) What are real-time industry applications of Hadoop?
4) What all modes Hadoop can be run in?
5) What are the most common Input Formats in Hadoop?
6) Define DataNode
7) What are the core methods of a Reducer?
8) What is Job Tracker role in Hadoop?
9) What is the use of RecordReader in Hadoop?
10) What companies use Hadoop, any idea?
11) Why do we need Hadoop?
12) What is the basic difference between traditional RDBMS and Hadoop?

EX.NO.07: MOUNT THE ONE NODE HADOOP CLUSTER USING FUSE

Aim:

To Write a program to use the API's of Hadoop to interact with it

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

Interfaces

Following are the important interfaces:

- **Client**<-->**ResourceManager**

  By using YarnClient objects.

- **ApplicationMaster**<-->**ResourceManager**

  By using AMRMClientAsync objects, handling events asynchronously by AMRMClientAsync.CallbackHandler

- **ApplicationMaster**<-->**NodeManager**

  Launch containers. Communicate with NodeManagers by using NMClientAsync objects, handling container events by NMClientAsync.CallbackHandler

Writing a Simple Yarn Application

Writing a simple Client

- The first step that a client needs to do is to initialize and start a YarnClient.

- YarnClientyarnClient = YarnClient.createYarnClient();
- yarnClient.init(conf);

- yarnClient.start();
- Once a client is set up, the client needs to create an application, and get its application id.

- YarnClientApplication app = yarnClient.createApplication();
- GetNewApplicationResponseappResponse = app.getNewApplicationResponse();
- The response from the YarnClientApplication for a new application also contains information about the cluster such as the minimum/maximum resource capabilities of the cluster. This is required so that to ensure that you can correctly set the specifications of the container in which the ApplicationMaster would be launched. Please refer to GetNewApplicationResponse for more details.

- The main crux of a client is to setup the ApplicationSubmissionContext which defines all the information needed by the RM to launch the AM. A client needs to set the following into the context:

- Application info: id, name

- Queue, priority info: Queue to which the application will be submitted, the priority to be assigned for the application.

- User: The user submitting the application

- ContainerLaunchContext: The information defining the container in which the AM will be launched and run. The ContainerLaunchContext, as mentioned previously, defines all the required information needed to run the application such as the local **Resources (binaries, jars, files etc.), Environment settings (CLASSPATH etc.), the Command to be executed and security T**okens (*RECT*).

The ApplicationReport received from the RM consists of the following:

- *General application information*: Application id, queue to which the application was submitted, user who submitted the application and the start time for the application.

- *ApplicationMaster details*: the host on which the AM is running, the rpc port (if any) on which it is listening for requests from clients and a token that the client needs to communicate with the AM.

- *Application tracking information*: If the application supports some form of progress tracking, it can set a tracking url which is available via ApplicationReport'sgetTrackingUrl() method that a client can look at to monitor progress.

- *Application status*: The state of the application as seen by the ResourceManager is available via ApplicationReport#getYarnApplicationState. If the YarnApplicationState is set to FINISHED, the client should refer to ApplicationReport#getFinalApplicationStatus to check for the actual success/failure of the application task itself. In case of failures, ApplicationReport#getDiagnostics may be useful to shed some more light on the the failure.

- If the ApplicationMaster supports it, a client can directly query the AM itself for progress updates via the host:rpcport information obtained from the application report. It can also use the tracking url obtained from the report if available.
- In certain situations, if the application is taking too long or due to other factors, the client may wish to kill the application. YarnClient supports the killApplication call that allows a client to send a kill signal to the AM via the ResourceManager. An ApplicationMaster if so designed may also support an abort call via its rpc layer that a client may be able to leverage.

- yarnClient.killApplication(appId);

Writing an ApplicationMaster (AM)

- The AM is the actual owner of the job. It will be launched by the RM and via the client will be provided all the necessary information and resources about the job that it has been tasked with to oversee and complete.

- As the AM is launched within a container that may (likely will) be sharing a physical host with other containers, given the multi-tenancy nature, amongst other issues, it cannot make any assumptions of things like pre-configured ports that it can listen on.

- When the AM starts up, several parameters are made available to it via the environment. These include the ContainerId for the AM container, the application submission time and details about the

NM (NodeManager) host running the ApplicationMaster. Ref ApplicationConstants for parameter names.

- All interactions with the RM require an ApplicationAttemptId (there can be multiple attempts per application in case of failures). The ApplicationAttemptIdcan be obtained from the AM's container id. There are helper APIs to convert the value obtained from the environment into objects.

- In setupContainerAskForRM(), the follow two things need some set up:
- Resource capability: Currently, YARN supports memory based resource requirements so the request should define how much memory is needed. The value is defined in MB and has to less than the max capability of the cluster and an exact multiple of the min capability. Memory resources correspond to physical memory limits imposed on the task containers. It will also support computation based resource (vCore), as shown in the code.

- Priority: When asking for sets of containers, an AM may define different priorities to each set. For example, the Map-Reduce AM may assign a higher priority to containers needed for the Map tasks and a lower priority for the Reduce tasks' containers.

- After container allocation requests have been sent by the application manager, contailers will be launched asynchronously, by the event handler of the AMRMClientAsync client. The handler should implement AMRMClientAsync.CallbackHandler interface.
- When there are containers allocated, the handler sets up a thread that runs the code to launch containers. Here we use the name LaunchContainerRunnable to demonstrate. We will talk about the LaunchContainerRunnable class in the following part of this article.
- heNMClientAsync object, together with its event handler, handles container events. Including container start, stop, status update, and occurs an error.

- After the ApplicationMaster determines the work is done, it needs to unregister itself through the AM-RM client, and then stops the client.

Result:

Thus the procedure to mount the one node hadoop cluster using FUSE was executed successfully.

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to write a program to use the API's of Hadoop to interact with it

Application:

The hadoop-hdfs-fuse package enables you to use your HDFS cluster as if it were a traditional filesystem on Linux. It is assumed that you have a working HDFS cluster and know the hostname and port that your NameNode exposes.

Viva-Voce

1) 1. What is Apache Hadoop?
2) Why do we need Hadoop?
3) What are the core components of Hadoop?
4) What are the Features of Hadoop?
5) Compare Hadoop and RDBMS?
6) What are the limitations of Hadoop?
7) Explain Data Locality in Hadoop?
8) What is a “Distributed Cache” in Apache Hadoop?
9) How is security achieved in Hadoop?
10) What does jps command do in Hadoop?
11) Is it possible to provide multiple input to Hadoop? If yes then how?
12) Is it possible to have hadoop job output in multiple directories? If yes, how?

EX.NO.08: WRITE A PROGRAM TO USE THE API'S OF HADOOP TO INTERACT WITH IT

Aim:

To Write a wordcount program to demonstrate the use of Map and Reduce tasks

Software Requirements:

Globus Toolkit or equivalent Eucalyptus or Open Nebula or equivalent

Hardware Requirements:

Standalone desktops 30 Nos

Procedure:

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce*job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Typically the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

The MapReduce framework consists of a single master ResourceManager, one slave NodeManager per cluster-node, and MRAppMaster per application.

Minimally, applications specify the input/output locations and supply *map* and *reduce* functions via implementations of appropriate interfaces and/or abstract-classes. These, and other job parameters, comprise the *job configuration*.

The Hadoop *job client* then submits the job (jar/executable etc.) and configuration to the ResourceManager which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Inputs and Outputs

The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

The key and value classes have to be serializable by the framework and hence need to implement the Writable interface. Additionally, the key classes have to implement the Writable Comparable interface to facilitate sorting by the framework.

Input and Output types of a MapReduce job:

(input) <k1, v1> ->map-><k2, v2> ->combine-><k2, v2> ->reduce-><k3, v3> (output)

Assuming environment variables are set as follows:

```
export JAVA_HOME=/usr/java/default
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

Compile WordCount.java and create a jar:

```
$ bin/hadoopcom.sun.tools.javac.Main WordCount.java
$ jar cf wc.jar WordCount*.class
```

Assuming that:

- /user/joe/wordcount/input - input directory in HDFS
- /user/joe/wordcount/output - output directory in HDFS
-

Sample text-files as input:

```
$ bin/hadoop fs -ls /user/joe/wordcount/input/ /user/joe/wordcount/input/file01 /user/joe/wordcount/input/file02

$ bin/hadoop fs -cat /user/joe/wordcount/input/file01
Hello World Bye World

$ bin/hadoop fs -cat /user/joe/wordcount/input/file02
Hello Hadoop Goodbye Hadoop
```

Run the application:

```
$ bin/hadoop jar wc.jar WordCount /user/joe/wordcount/input /user/joe/wordcount/output
```

Sample Output:

```
$ bin/hadoop fs -cat /user/joe/wordcount/output/part-r-00000`

Bye 1

Goodbye 1

Hadoop 2

Hello 2

World 2`
```

Result:

Thus the program using hadoop APIs was executed successfully

Outcomes:

At the end of the course, the student should be able to

- Use the grid and cloud tool kits
- Design and implement applications on the Cloud to write a wordcount program to demonstrate the use of Map and Reduce tasks

Application:

In Hadoop, MapReduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel cross a cluster of servers. The results of tasks can be joined together to compute final results.

Viva-Voce

1.What is a Combiner?

2.Explain about YARN?

3.What is Hadoop MapReduce?

4. What is MapReduce framework ?

5.How many types of special znodes are present in Zookeeper ?

6.What are the primary phases of reducer?

7.What is HDFS?

8.Which utility is used for checking the health of a HDFS file system?

9.The difference between standalone and pseudo-distributed mode ?

10.What is the default input format?

11.What is HBASE?

12.What are the operations available in HDFS files?

13.How to restart the namenodes?

14.What if a Namenode has no data?

15.What is the basic difference between traditional RDBMS and Hadoop?

16.What is a Namenode?

17.What is a job tracker?

18.What is a heartbeat in HDFS?