DAYANANDA SAGAR COLLEGE OF ENGINEERING

An Autonomous Institute Affiliated to VTU, Belagavi Approved by AICTE ; ISO 9001:2015 Certified
Accredited by National Assessment Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Lab Manual

Introduction to Cloud Analyst

"CloudAnalyst" is a simulator used for simulating large scaled applications along with a novel approach for such research oriented studies. there are several toolkits that can be used to model a simulated environment to study the behaviour of a large scaled application on the Internet. But having an easy to use tool with a level of visualisation capability is even better than just a toolkit. Such a tool separates the simulation experiment set up exercise from a programming exercise and enables a modeler to concentrate on the simulation parameters rather than the technicalities of programming. A graphical output of the simulation results enables the results to be analyzed more easily and more efficiently and it may also help in quickly highlighting any problems with the performance and accuracy of the simulation logic.

Features of the Simulator

2.2.1 Ease of use

Ease of setting up and executing a simulation experiment is the main point of having a simulation tool. The simulator needs to provide an easy to use graphical user interface which is intuitive yet comprehensive.

2.2.2 Ability to define a simulation with a high degree of configurability and flexibility. Perhaps the most important feature is the level of configurability the tool can provide. A simulation, especially of the nature of modelling something as complex as an Internet Application depends on many parameters and most of the time the values for those parameters need to be assumed. Therefore it is important to be able to enter and change those parameters quickly and easily and repeat simulations.

2.2.3 Graphical output

A picture is said to be worth a thousand words. Graphical output in the form of tables and charts is highly desirable to summarise the potentially large amount of statistics that is collected during the simulation. Such effective presentation helps in identifying the important patterns of the output parameters and helps in comparisons between related parameters.

2.2.4 Repeatability

Repeatability of experiments is a very important requirement of a simulator. The same experiment with the same parameters should produce similar results each time the simulation is executed. Otherwise the simulation becomes just a random sequence of events rather than a controlled experiment. It is also helpful to be able to save an experiment (the set of input parameters) as a file and also be able to save the results of an experiment as a file.

2.2.5 Ease of extension

As already mentioned simulating something like the Internet is a complex task and it is 9 unlikely a 100% realistic simulation framework and a set of input parameters can be achieved in a few attempts. Therefore the simulator is expected to evolve continuously rather than a program that is written once and for all and then used continuously. Therefore the simulator architecture should support extensions with minimal effort with suitable frameworks.

2.3 Simulation Output / What is being Measured

Following are the statistical measures produced as output of the simulation in the initial version of the simulator.

- Response time of the simulated application
  - o Overall average, minimum and maximum response time of all user requests simulated
  - o The response time broken down by user groups, located within geographical regions
  - o The response time further broken down by the time showing the pattern of change over the duration of a day
- The usage patterns of the application
  - o How many users use the application at what time from different regions of the world, and the overall effect of that usage on the data centers hosting the application
- The time taken by data centers to service a user request
  - o The overall request processing time for the entire simulation
  - o The average, minimum and maximum request processing time by each data center
  - o The response time variation pattern during the day as the load changes

2.4 Technologies Used

- Java – The simulator is developed 100% on Java platform, using Java SE 1.6.
- Java Swing – The GUI component is built using Swing components.
- CloudSim – CloudSim features for modelling data centers is used in CloudAnalyst.
- SimJava – Sim Java is the underlying simulation framework of CloudSim and some features of SimJava are used directly in CloudAnalyst.

CloudAnalyst Design

The CloudAnalyst is built on top of CloudSim tool kit, by extending CloudSim functionality with the introduction of concepts that model Internet and Internet Application behaviours.

Steps tp install and Run Cloud Analyst

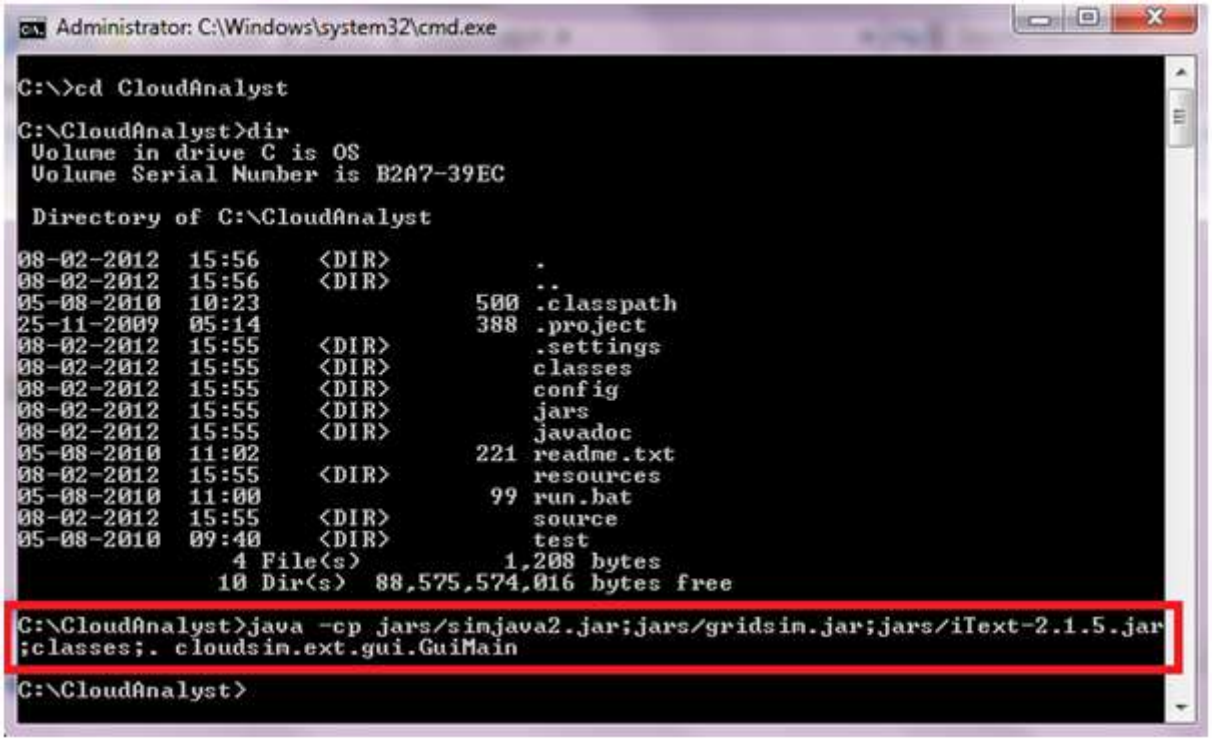
1. Download installer package
from http://www.cloudbus.org/cloudsim/CloudAnalyst.zi

2. Extract Files from the compressed package file which will give following folder structure

3. To start the simulator, you can got Command line then use the following command on the command prompt as shown in the screenshot below

C:\CloudAnalyst>java – cp jars/simjava2.jar;jars/gridsim.jar;jars/iText-2.1.5.jar;classes;. Cloudsim.ext.gui.GuiMain

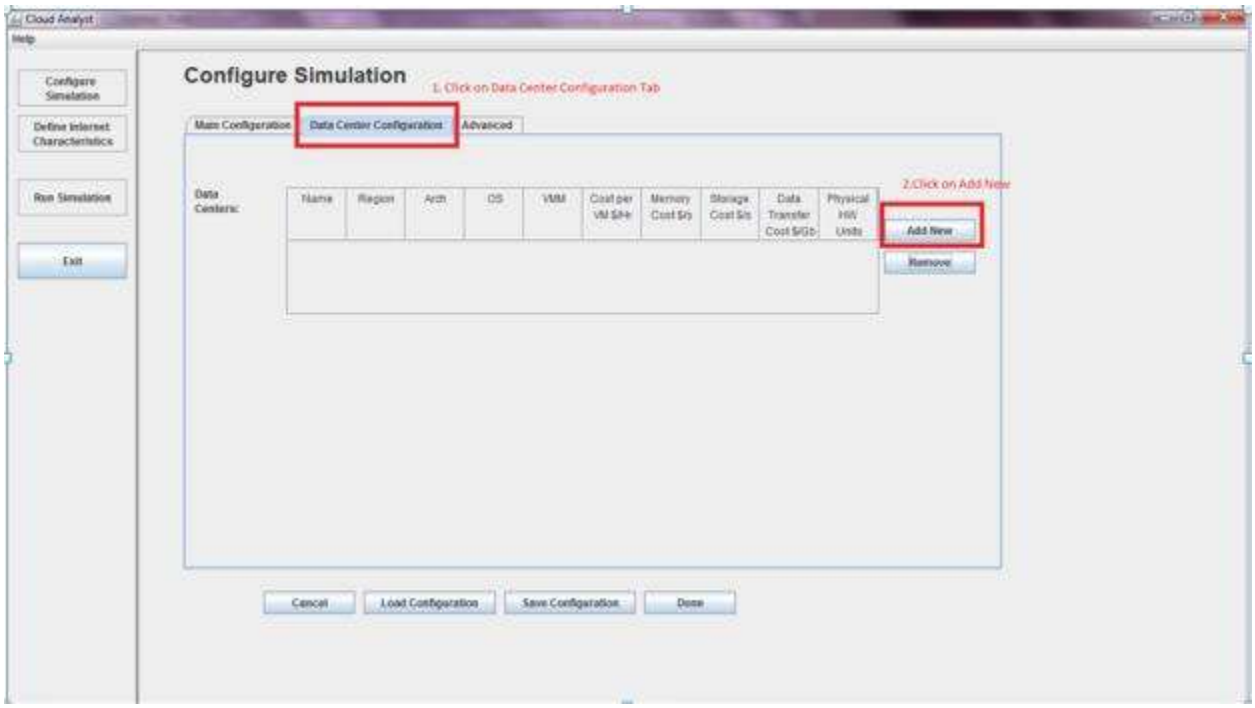
n

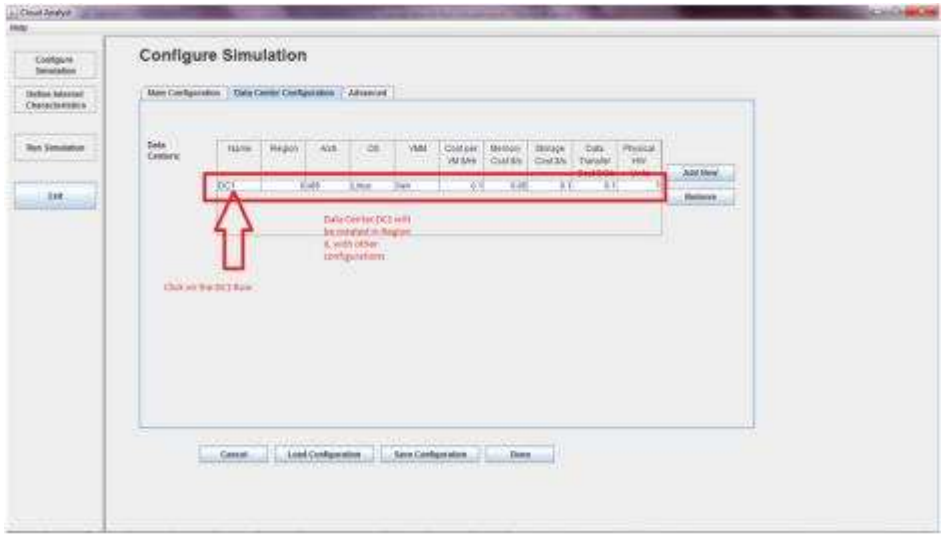


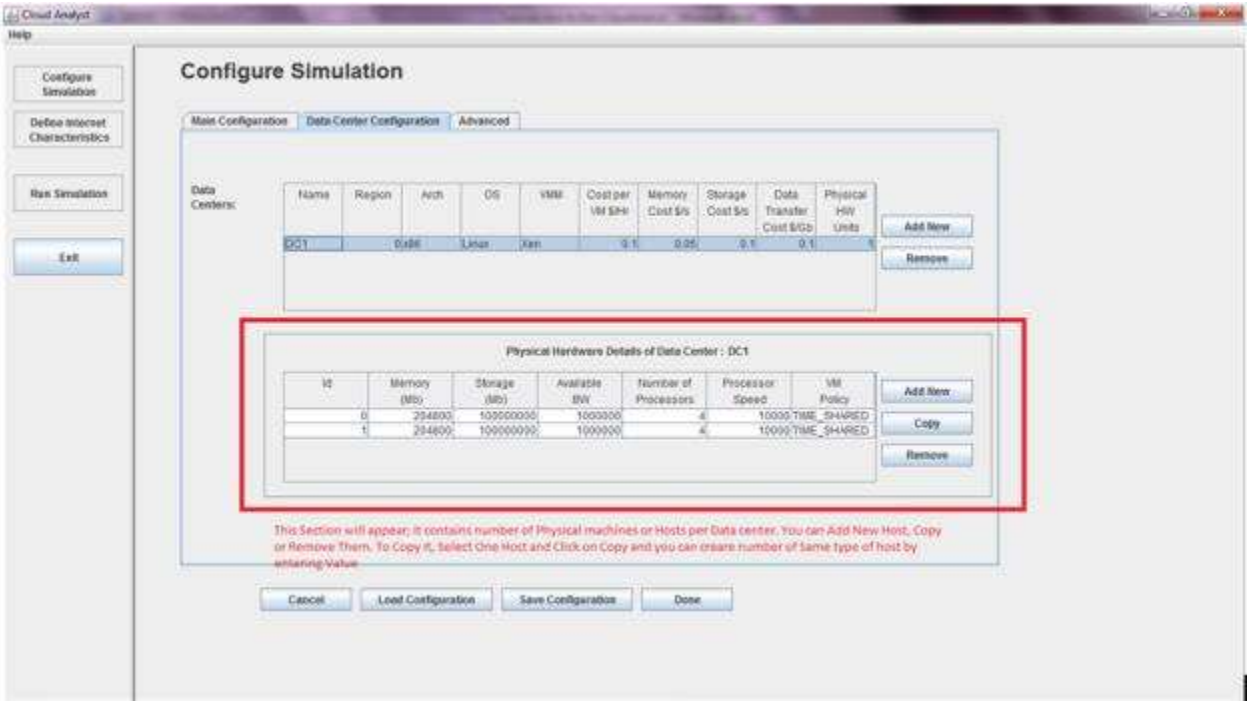
Alternatively you can also click on **run.bat** file and click **Done**!!

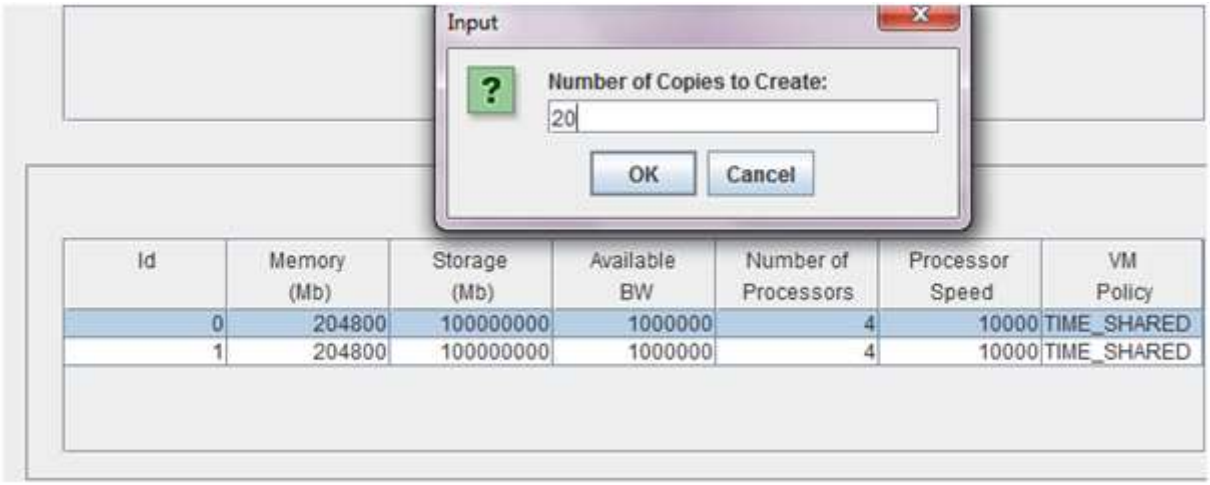
4. The welcome screen of the simulator looks like the following image.

5. To configure the parameters for experimentation click on "**Show Region Boundaries**". Here you can datacenters and other physical hardware details of the datacenter as shown in the following screenshots.

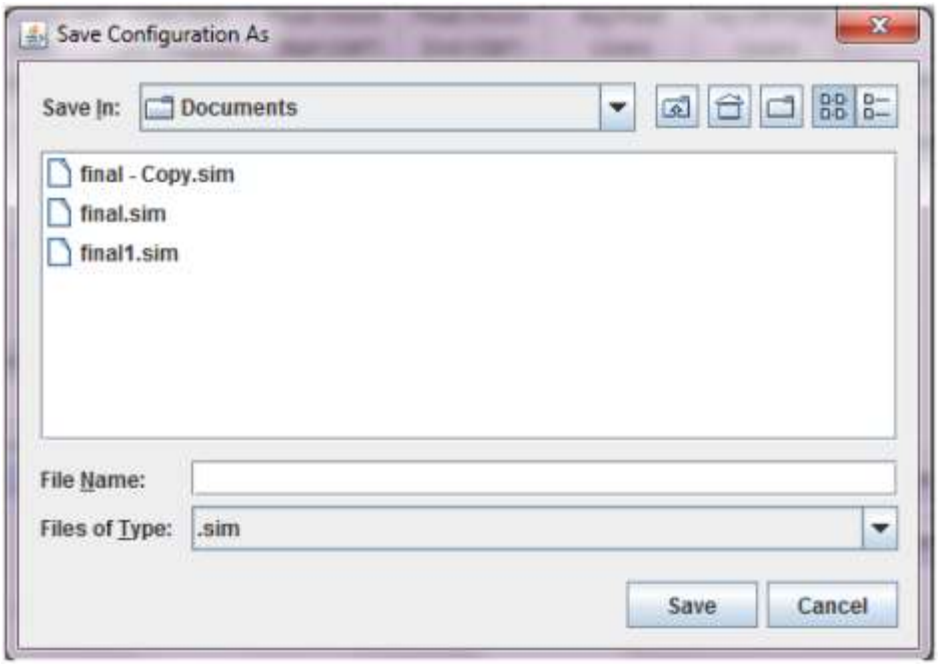

Cloud Analyst
Help
Configure Simulation
Define Internet Characteristics
Run Simulation
Exit
Configure Simulation
1. Click on Data Center Configuration Tab
Main Configuration
Data Center Configuration
Advanced
Data Centers:
Name
Region
Arch
OS
VMM
Cost per VM $/Hr
Memory Cost $/s
Storage Cost $/s
Data Transfer Cost $/Gb
Physical HW Units
2.Click on Add New
Add New
Remove
Cancel
Load Configuration
Save Configuration
Done


Configure Simulation
Main Configuration
Data Center Configuration
Advanced
Data Centers:
Add New
Remove
Cancel
Load Configuration
Save Configuration
Done





6. You can Save this Configuration as well in case you want to use it later. It is stored as .sim file. XML data is generated and saved as Sim file.

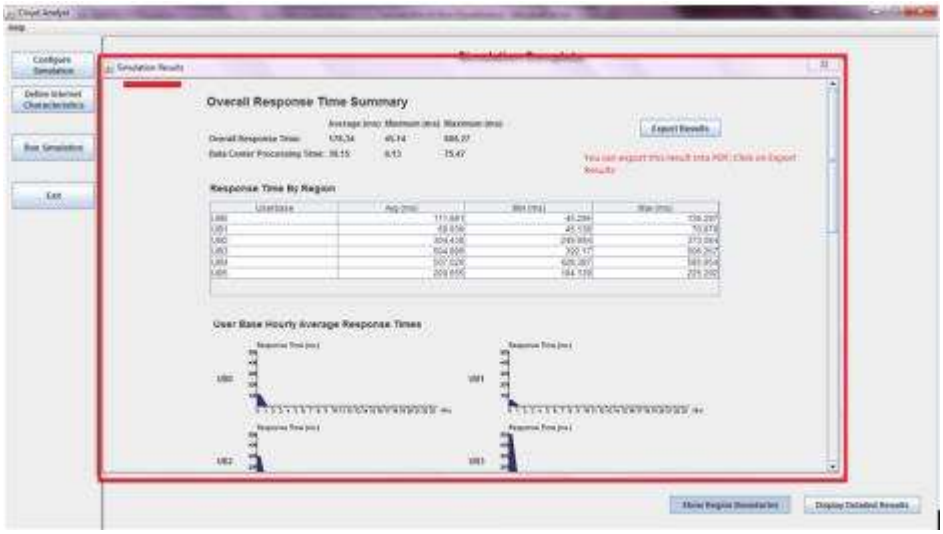


Saved configurations can be loaded anytime easily into Cloud Analyst.

7. Once your are done with Configuration; click on Done.

8. To run the simulation click on Run Simulation

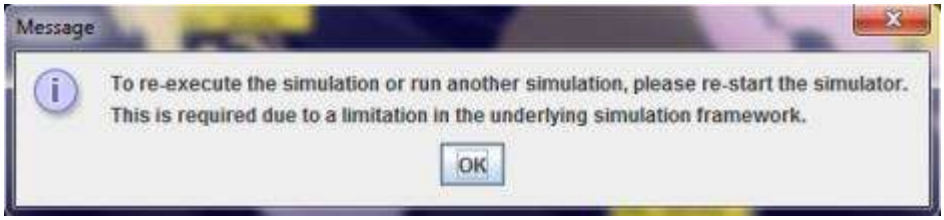
9. Simulation Results Window will Open



Then click on “Close it.”

10. Main Window will give all statistics

11. If you will try to run Simulation again then it will give Error



12. Click on Exit & restart the simulator.

Experiment 1:

In given cloud simulator to host a simple web application with the below given configurations

i. six user bases and one datacenter with 50 virtual machines each with 1024 Mb of memory and processor speed as 100 MIPS.
ii. four user bases and one datacenter with 25 virtual machines each with 1024 Mb of memory and processor speed as 100 MIPS.

Analyze the average, minimum and maximum response time and datacenter processing time and write the findings.

Procedure for (i):

Step1: Start the Cloud Analyst simulator by clicking on **run.bat** file.

Step 2:Once the simulator starts navigate to **Configure Simulation** tab.

Step 3:In Main Configuration tab, add six user bases by clicking add new button in the **User Bases** table.

Step4: Set the memory to 1024 Mb in the **Application Deployment Configuration** table.

Step 5: Set the number of VM's to 50 in the **Application Deployment Configuration.**

Step 5: Navigate to **Data Center Configuration** tab and click on the Datacenter name ie DC1.

Step 6: The **physical hardware details of Data Center** table is displayed where you have to set the processor speed to 100 MIPS.

Step 7: Click on **Save Configuration** to save the simulation by naming it.

Step 8: Click **Done.**

Step 9: Click on **Run Simulation** to run the experiment.

Step 10: Wait for the results to be loaded.

Step 11: Note down the average, maximum and minimum response time for the user bases & datacenter processing times.

Step 12: Draw the corresponding graphs.

Procedure for (ii):

Step1: Start the Cloud Analyst simulator by clicking on **run.bat** file.

Step 2:Once the simulator starts navigate to **Configure Simulation** tab.

Step 3:In Main Configuration tab, add four user bases by clicking add new button in the **User Bases** table.

Step4: Set the memory to 1024 Mb in the **Application Deployment Configuration** table.

Step 5: Set the number of VM's to 25 in the **Application Deployment Configuration.**

Step 5: Navigate to **Data Center Configuration** tab and click on the Datacenter name ie DC1.

Step 6: The **physical hardware details of Data Center** table is displayed where you have to set the processor speed to 100 MIPS.

Step 7: Click on **Save Configuration** to save the simulation by naming it.

Step 8: Click **Done.**

Step 9: Click on **Run Simulation** to run the experiment.

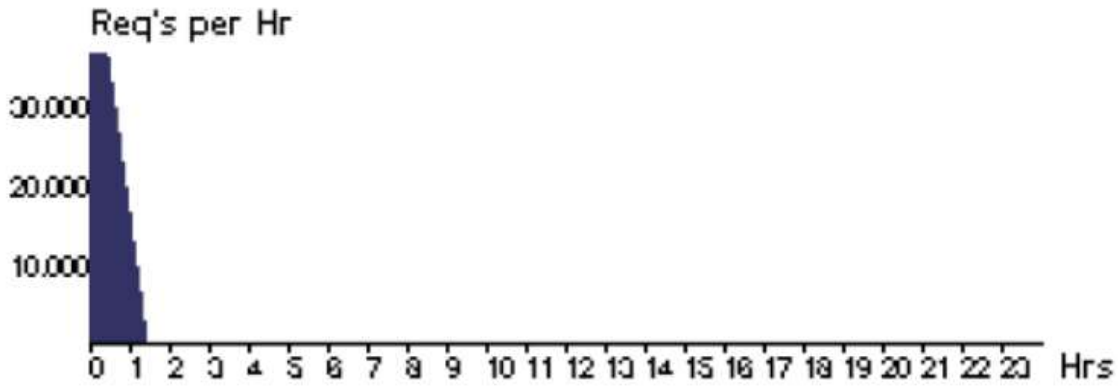
Step 10: Wait for the results to be loaded.

Step 11: Note down the average, maximum and minimum response time for the user bases & datacenter processing times.

Step 12: Observe & note the corresponding graphs.

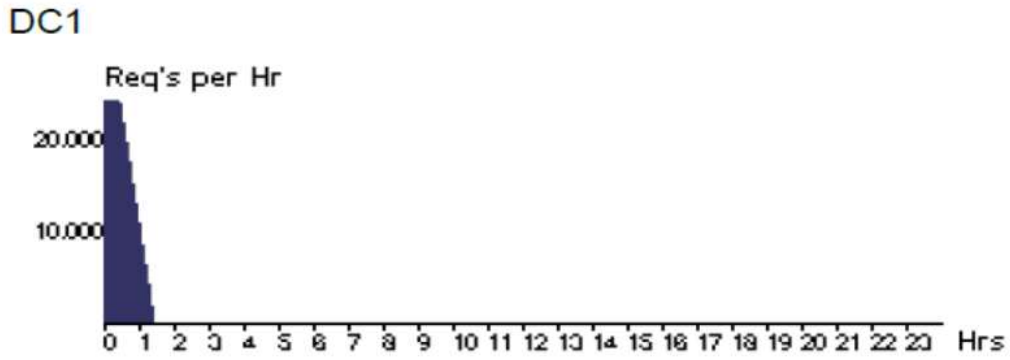
Output of (i):

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	319.17	234.64	450.60
Data Center processing time:	19.52	0.07	124.06



Output of (ii)

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	300.59	232.81	375.33
Data Center processing time:	0.55	0.04	0.86



Analysis of the experimentation:

The response times & datacenter processing time in (i) is more than (ii) as we have more number of user bases in (i).

Outcome:

We learnt that the user base, which is collection of users, plays a vital role in generating requests. As the number of user bases increase the time taken to process the user request also increases. The datacenter which is used to process the user requests generated by the user base, needs more processing time. Hence the response time of the datacenter also increases.

Experiment 2:

In given cloud simulator to host a simple web application on cloud with the configurations given below

i. two datacenters with 50 VM's each with 1024 Mb memory and processor speed as 100 MIPS. Set the number of user bases to 6.

Conduct experimentation by changing different service broker policies to analyze the average, minimum and maximum response time and total datacenter processing time. Also, analyze the total cost required to run the application.

Procedure for (i):

Step1: Start the Cloud Analyst simulator by clicking on **run.bat** file.

Step 2:Once the simulator starts navigate to **Configure Simulation** tab.

Step 3:In **Main Configuration** tab, add six user bases by clicking add new button in the **User Bases** table.

Step 4: Navigate to **Data Center Configuration** tab and click on the **Add New** to add second datacenter.

Step5: Navigate back to **Main Configuration** tab, in the **Application Deployment Configuration** table click on **Add New** to configure the datacenter.

Step 6: Configure the datacenters DC1 & DC2 with 50 VM's each and Memory of 1024 Mb each.

Step 7: Configure the **Service Broker Policy** to **Closest Datacenter** for first run. This is the default policy.

Step 8: Click on **Save Configuration** to save the simulation by naming it.

Step 9: Click **Done.**

Step 11: Click on **Run Simulation** to run the experiment.

Step 12: Wait for the results to be loaded.

Step 13: Note down the average, maximum and minimum response time for the user bases & datacenter processing times.

Step 14: Note down the cost computed to run the simulation.

Note:

- For second run all the above mentioned steps remain the same except Step 8, where you have to change the service broker policy to **Optimise Response time**
- For third run all the above mentioned steps remain the same except Step 8, where you have to change the service broker policy to **Reconfigure Dynamically with Load.**

Output

(i)Service Policy: Closest Datacenter

	Avg (ms)	**Min (ms)**	**Max (ms)**
Overall response time:	362.88	226.27	489.53
Data Center processing time:	63.33	0.13	125.56

Cost

Total Virtual Machine Cost: $10.04

Total Data Transfer Cost : $0.38

Grand Total : $10.42

(ii)Service Policy: Optimise Response Time

	Avg (ms)	**Min (ms)**	**Max (ms)**

Overall response time:	301.21	226.26	380.77
Data Center processing time:	1.46	0.13	1.82

Cost

Total Virtual Machine Cost: $10.04

Total Data Transfer Cost : $0.38

Grand Total : $10.42

(iii)Service Policy: Reconfigure Dynamically with Load

	Avg (ms)	**Min (ms)**	**Max (ms)**
Overall response time:	303.83	226.41	7503.02
Data Center processing time:	4.17	0.13	7178.01

Cost

Total Virtual Machine Cost: $15.40

Total Data Transfer Cost : $0.38

Grand Total : $15.79

Analysis:

The data center's response time & processing time for each of the service policy is different. We find that the response time and processing time is efficient when we use optimize response time service policy.

Outcome of experimentation:

In this experiment we noted the response time and processing time of two datacenters by varying the service policies. It was noted that when we use optimize response time service policy, the response time and processing time of the datacenter is better compared to the other two service policies.

Experiment 3:

In given cloud simulator to host a simple web application on cloud with the configurations given below

i. two datacenters with 50 VM's each with 1024 Mb memory and processor speed as 100 MIPS. Set the number of user bases to 6.

Conduct experimentation by changing different load balancing policies and different Service Policies ,conclude which load balancing policy is better by clearly stating the reasons.

Procedure for (i):

Step1: Start the Cloud Analyst simulator by clicking on **run.bat** file.

Step 2:Once the simulator starts navigate to **Configure Simulation** tab.

Step 3:In **Main Configuration** tab, add six user bases by clicking add new button in the **User Bases** table.

Step 4: Navigate to **Data Center Configuration** tab and click on the **Add New** to add second datacenter.

Step5: Navigate back to **Main Configuration** tab, in the **Application Deployment Configuration** table click on **Add New** to configure the datacenter.

Step 6: Configure the datacenters DC1 & DC2 with 50 VM's each and Memory of 1024 Mb each.

Step 7: Navigate to **Advanced** tab, to change the load balancing policy for each run.(**please refer Note below**)

Step 8: Click on **Save Configuration** to save the simulation by naming it.

Step 9: Click **Done.**

Step 11: Click on **Run Simulation** to run the experiment.

Step 12: Wait for the results to be loaded.

Step 13: Note down the average, maximum and minimum response time for the user bases & datacenter processing times.

Step 14: Note down the cost computed to run the simulation.

Note:

- For first run all the above mentioned steps remain the same except Step 8, where you have to change the load balancing policy to **Round Robin**
- For second run all the above mentioned steps remain the same except Step 8, where you have to change the load balancing policy to **Equally Spread Current Execution Load**

Output

(i) Load balancing policy: Round Robin

	Avg (ms)	**Min (ms)**	**Max (ms)**
Overall response time:	299.95	36.33	720.08
Data Center processing time:	7.84	0.07	127.79

(ii) Load balancing policy: Equally Spread Current Execution Load

	Avg (ms)	**Min (ms)**	**Max (ms)**
Overall response time:	301.10	226.26	366.52
Data Center processing time:	1.45	0.13	1.82

Outcome of experimentation

The analysis of this experiment shows that the response time and data center processing time using equally shared current execution load balancing algorithm is better than Round Robin load balancing algori

Experiment 4:

1. HADOOP MODES

Perform setting up and Installing Hadoop in its three operating modes.

- Standalone.
- Pseudo distributed.
- Fully distributed.

Configure Hadoop properties to run on all three modes of operation by setting Name Node
and Data Node.

This section describes how to set up and configure a single-node Hadoop installation so that you can quickly perform simple operations using Hadoop MapReduce and the Hadoop Distributed File System (HDFS).

Prepare to Start the Hadoop Cluster

- Download VMworkstation using the link below

  https://www.vmware.com/products/workstation-pro.html

- Hadoop training. Ova link using the link below:

  https://drive.google.com/file/d/1KaMTTNwGTd5OuCUMF3y9EQvhZyUV_O8D/view?usp=sharing

- Now, it's all ready for the running Hadoop on VMwareworkstation.

Standalone Operation

- Go to VMwareworkstation, Home page, Open Virtual Machine and specify the location of Hadoop Station.
- Click on Hadoop Training and Power on this virtual machine.
- Login to Hadoop, no password is required.
- Open terminal and start working on the instructions.
- This HadoopVM is created in such a way that, all daemons are installed in it.
- To start all daemons, open terminal and give the instruction below

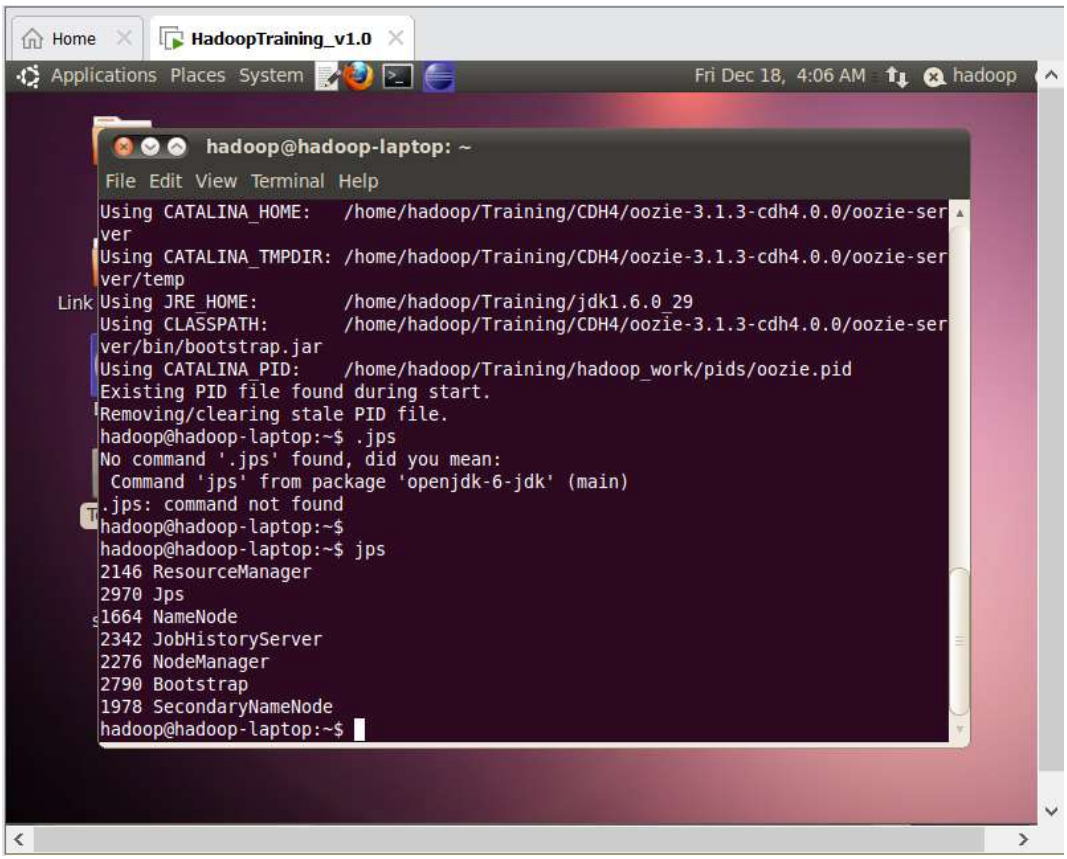
strartCDH.sh

- The above instruction will initialize all configuration files.

- To verify its working, specify the instruction and check

jps

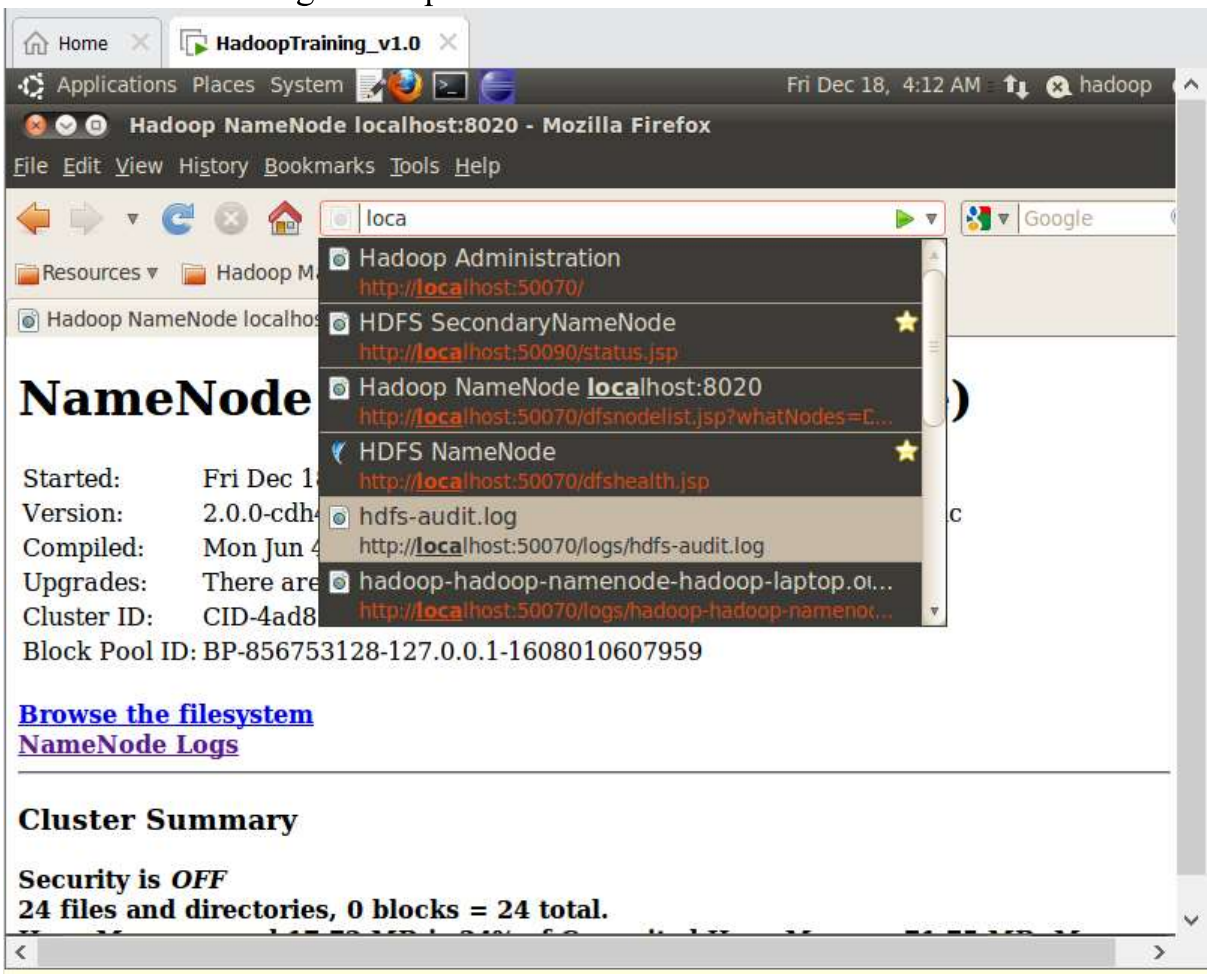
- All Resource managers will be set and executed



- All the resources are executed now.

Pseudo-Distributed Operation

- Go to Browser and give the path like



- This instruction will give all the local files and its existence in the system.

- In the browser give the instructions like 50070, which executes Local Administration Page, it also gives different Pages that are residing on local host.

Pseudo-Distributed Operation

- Execute above instructions on another system.
- Go to browser, enter the address of another system in this browser, then the page will be executed.

- Thus Pseudo Distributed mode is executed.

Experiment 5:

2. FILE MANAGEMENT IN HADOOP

Implement the following file management tasks in Hadoop

- Adding Files and directories.
- Retrieving files
- Deleting files

Run in Hadoop environment how to create a new file in HDFS environment, to list files in HDFS, upload and download files in HDFS as well other properties copy file, move files and remove file operations in HDFS.

Inserting Data into HDFS

Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system. Follow the steps given below to insert the required file in the Hadoop file system.

Step 1

You have to create an input directory.

$ $HADOOP_HOME/bin/hadoop fs -mkdir /user/input

Step 2

Transfer and store a data file from local systems to the Hadoop file system using the put command.

$ $HADOOP_HOME/bin/hadoop fs -put /home/file.txt /user/input

Step 3

You can verify the file using ls command.

$ $HADOOP_HOME/bin/hadoop fs -ls /user/input

Retrieving files from HDFS

Assume we have a file in HDFS called outfile. Given below is a simple demonstration for retrieving the required file from the Hadoop file system.

Step 1

Initially, view the data from HDFS using cat command.

$ $HADOOP_HOME/bin/hadoop fs -cat /user/output/outfile

Step 2

Get the file from HDFS to the local file system using get command.

$ $HADOOP_HOME/bin/hadoop fs -get /user/output/ /home/hadoop_tp/

Deleting files

Hdfs dfs -rm -r hdfs://path/ to / file

Experiment 6:

MAP REDUCE PROGRAM

A scenario that multiples files exist with some contents in it, write a Map-Reduce program to count the number of words present in different files.

Execute these programs on Eclipse platform by creating three files.

The entire MapReduce program can be fundamentally divided into three parts:

- Mapper Phase Code
- Reducer Phase Code
- Driver Code

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{
```

```
  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context
                  ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}
public static class IntSumReducer
     extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();
  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}
public static void main(String[] args) throws Exception {
```

```
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Usage

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/hadoop com.sun.tools.javac.Main WordCount.java
$ jar cf wc.jar WordCount*.class
```

Assuming that:

- `/user/ravi/wordcount/input` - input directory in HDFS
- `/user/ravi/wordcount/output` - output directory in HDFS

Sample text-files as input:

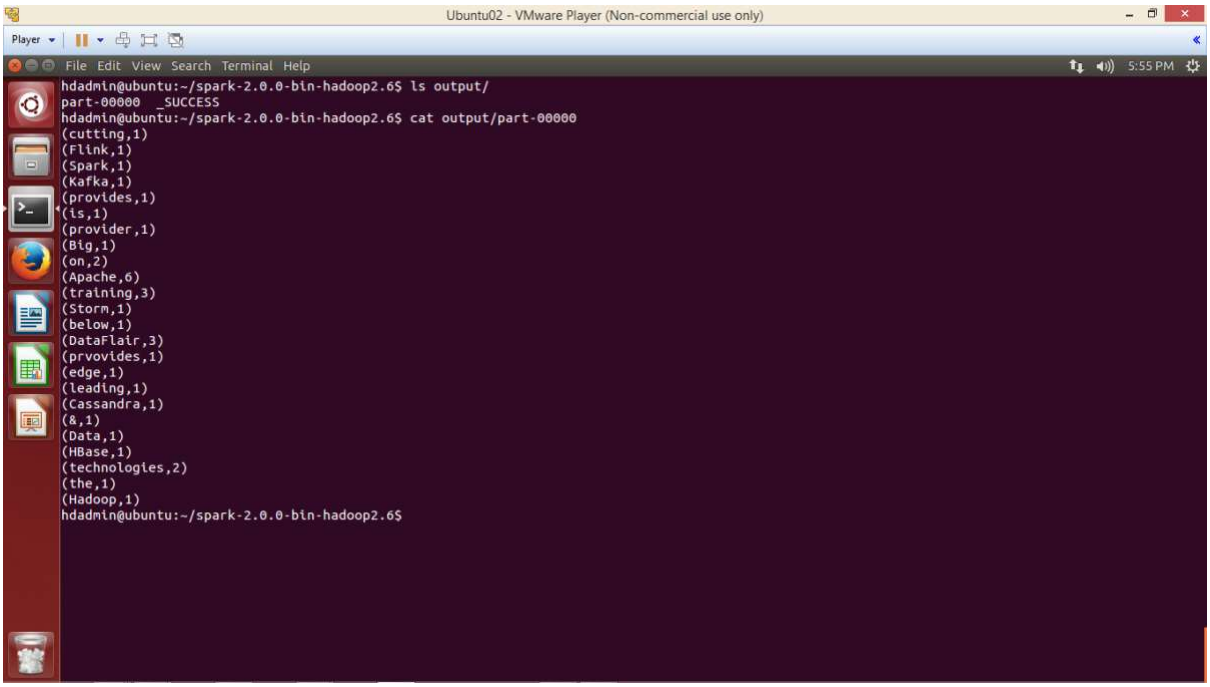
```
$ bin/hadoop fs -ls /user/ravi/wordcount/input/

/user/ravi/wordcount/input/file01

/user/ravi/wordcount/input/file02


$ bin/hadoop fs -cat /user/ravi/wordcount/input/file01

Hello World Bye World


$ bin/hadoop fs -cat /user/ravi/wordcount/input/file02

Hello Hadoop Goodbye Hadoop
```

Run the application:

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

Output:

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

Bye 1

Goodbye 1

Hadoop 2

Hello 2

World 2
```


Ubuntu02 - VMware Player (Non-commercial use only)
Player
File Edit View Search Terminal Help
5:55 PM
hdadmin@ubuntu:~/spark-2.0.0-bin-hadoop2.6$ ls output/
part-00000 _SUCCESS
hdadmin@ubuntu:~/spark-2.0.0-bin-hadoop2.6$ cat output/part-00000
(cutting,1)
(Flink,1)
(Spark,1)
(Kafka,1)
(provides,1)
(is,1)
(provider,1)
(Big,1)
(on,2)
(Apache,6)
(training,3)
(Storm,1)
(below,1)
(DataFlair,3)
(prvovides,1)
(edge,1)
(leading,1)
(Cassandra,1)
(&,1)
(Data,1)
(HBase,1)
(technologies,2)
(the,1)
(Hadoop,1)
hdadmin@ubuntu:~/spark-2.0.0-bin-hadoop2.6$