ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (1-8), Month: October - December 2014, Available at: www.researchpublish.com

# THUMBNAIL IMAGE CREATION IN HADOOP USING PYTHON

Devi.V.Kumar

University of Calicut, NSS College of Engineering, Kerala, India

***Abstract:*** **Apache Hadoop is a software framework that supports data-intensive distributed application under a free license. It enables applications to work with thousands of nodes and petabytes of data. Hadoop was inspired by Google's MapReduce and Google File System (GFS) papers. Hadoop was originally developed to support distribution for the Nutch search engine project. Here Hadoop is incorporated for handling large cluster of images. Moreover, Hadoop provide language independent platform which helps the programmer to remain in their comfort zone. Here I discussed the cloudera installation of Hadoop and here I present the design, implementation and evaluation of Hadoop thumbnail creation model that supports incremental job expansion. Under this model, a job consumes input as required and can dynamically govern its resource consumption while producing the required results. I have implemented the mechanism in Hadoop, and I present results from an experimental performance study of normal and Hadoop model for creating thumbnails.**

***Keyword:*** **Cloudera, SSH, MapReduce, HDFS.**

## I. INTRODUCTION

Thumbnails are reduced-size versions of images, used to help in recognizing and organizing them, serving the same role for images as a normal text index does for words. These images are usually 150x150 size and have wide scale usage in social networks .In the age of digital images, visual search engines and image-organizing programs normally use thumbnails, as do most of the operating system or desktop environments, such as Microsoft Windows, Mac OS X,KDE and GNOME. Some web designers generates thumbnail of images with the help of HTML coding that makes the user's browser degenerate the existing images .The new images thus formed will be lower in pixel ratio ,quality and clarity. In principle the display size of an image in pixels should always correspond to its actual size, in part because one purpose of a thumbnail on web pages is to reduce download time. The visual quality must be the most important factor of any images, but due to browser resizing it is also usually less than the ideal.

Displaying a significant part of the picture instead of the full frame can allow using a smaller thumbnail while maintaining the image resolution. For example, when creating thumbnails of a full-body portrait of a person, it may be better to show the face slightly reduced than an indistinct figure. This has the disadvantage that it misleads viewers about what the image contains, so it is less well suited for searching or a catalogue than for artistic presentations. So this framework provides the fastest as well as efficient way to resize your pictures .The power comes from the simultaneous generation of different sizes and ability to enlarge your images too.

Apache Hadoop provides a framework for large scale parallel processing using a distributed file system and the map-reduce programming paradigm. First step require the importing of some interesting data sets Programming model and started doing interesting projects that were previously impossible due to their massive computational requirements .Hadoop is a top-level Apache project being built and used by a global community of contributors, supported by JDK. Yahoo! has been the largest contributor to the projector, and uses Hadoop extensively across its businesses.

This paper discusses the simple installation steps of CDH3 packages. Three programs are implemented for verifying the initialization. This includes common testing programs available in CDH3 package as well as Matrix inverse program, which generate the inverse of input matrixes in a fast mode. Next step includes the testing of map-reduce python files which generates the thumbnail of input images a output file. The last step includes extrapolating the results,cpu utlization,throughput of the application and comparison with the old results.

## II. HADOOP OVERVIEW AND SYSTEM PREREQUISITE

### *A. Hadoop Overview*

Today, we’re surrounded by data. People upload videos, take pictures on their cell phones, text friends, update their Facebook status, leave comments around the web, click on ads, and so forth. Machines, too, are generating and keeping more and more data. The exponential growth of data first presented challenges to cutting-edge businesses such as Google, Yahoo, Amazon, and Microsoft. They needed to go through terabytes and petabytes of data to figure out which websites were popular, what books were in demand, and what kinds of ads appealed to people. Existing tools were becoming inadequate to process such large data sets.

Hadoop is an open source framework for writing and running distributed applications that process large amounts of data. Distributed computing is a wide and varied field, but the key distinctions of Hadoop are its properties. Hadoop runs on large clusters of commodity machines or on cloud computing services such as Amazon’s Elastic Compute Cloud (EC2). Hadoop is architected with the assumption of frequent hardware malfunctions. It can gracefully handle most such failures. Hadoop scales linearly to handle larger data by adding more nodes to the cluster. Hadoop allow users to quickly write efficient parallel code. Hadoop’s accessibility and simplicity give it an edge over writing and running large distributed programs. Even college students can quickly and cheaply create their own Hadoop cluster. On the other hand, its robustness and scalability make it suitable for even the most demanding jobs at Yahoo and Facebook. These features make Hadoop popular in both academia and industry.

### *B. The Building Blocks of Hadoop*

On a fully configured cluster, “running Hadoop" means running a set of daemons, or resident programs, on the different servers in your network. These daemons have specific roles; some exist only on one server, some exist across multiple servers. Hadoop employs a distributed storage system called Hadoop File System, or HDFS.The NameNode is the master of HDFS that directs the slave DataNode daemons to perform the low-level I/O tasks. DataNode are constantly reporting to the NameNode. Upon initialization, each of the DataNodes informs the NameNode of the blocks it's currently storing. The Secondary NameNode (SNN) is an assistant daemon for monitoring the state of the cluster HDFS.Like the NameNode, each cluster has one SNN and it typically resides on its own machines as well. The job tracker daemon is the liaison between your application and Hadoop. Once you submit your code to your cluster, the Job Tracker determines the execution plan by determining which files to process, assigns nodes to different tasks, and monitors all tasks as they're running. As with the storage daemons, the computing daemons also follow master/slave architecture; the Job Tracker is the master overseeing the overall execution of a MapReduce job and TaskTracker manages the execution of individual tasks on each slave node.

### *C. Map Reduce Overview*

MapReduce is a programming model for data processing. The model is simple, yet not too simple to express useful programs in. Hadoop can run MapReduce programs written in various languages; in this chapter, we shall look at the same program expressed in Java, Ruby, Python, and C++. Most important, MapReduce programs are inherently parallel, thus putting very large-scale data analysis into the hands of anyone with enough machines at their disposal. MapReduce comes into its own for large datasets, so let’s start by looking at one. Map-Reduce treats data as a list of (key, value) pairs and expresses a computation in terms of two functions: map and reduce.

map (k1; v1)! list(k2; v2)

reduce(k2; list(v2)) ! list(k3; v4)

The map function, defined by the user, takes as input a keyvalue pair and outputs a list of intermediate key-value pairs. All intermediate values corresponding to the same intermediate key are grouped together and passed to a reduce function. The reduce function, also defined by the user, processes each key and the associated list of values and produces a list of keyvalue pairs that form the final output. Figure 1 shows the data flow in a Map-Reduce based execution. Input data is loaded into a file or files in a distributed file system (DFS) wherein each file is partitioned into smaller chunks, also called input splits. A Map-Reduce computation begins with a Map phase where each input split is processed in parallel by as many map tasks as there are splits. Each input split is a list of key-value pairs. A map task applies the user-defined map function to each key-value pair to produce a list of output key-value pairs. The output key-value pairs from a map task are partitioned on the basis of their key. Each partition is then sent across the cluster to a remote node in the shuffle phase. Corresponding partitions from the map tasks are merged and sorted at their receiving nodes. For each key, the associated values are grouped together to form a list. The key and the corresponding list are given to the user-specified reduce function. The resulting key-value pairs are written back to the DFS and form the final output.

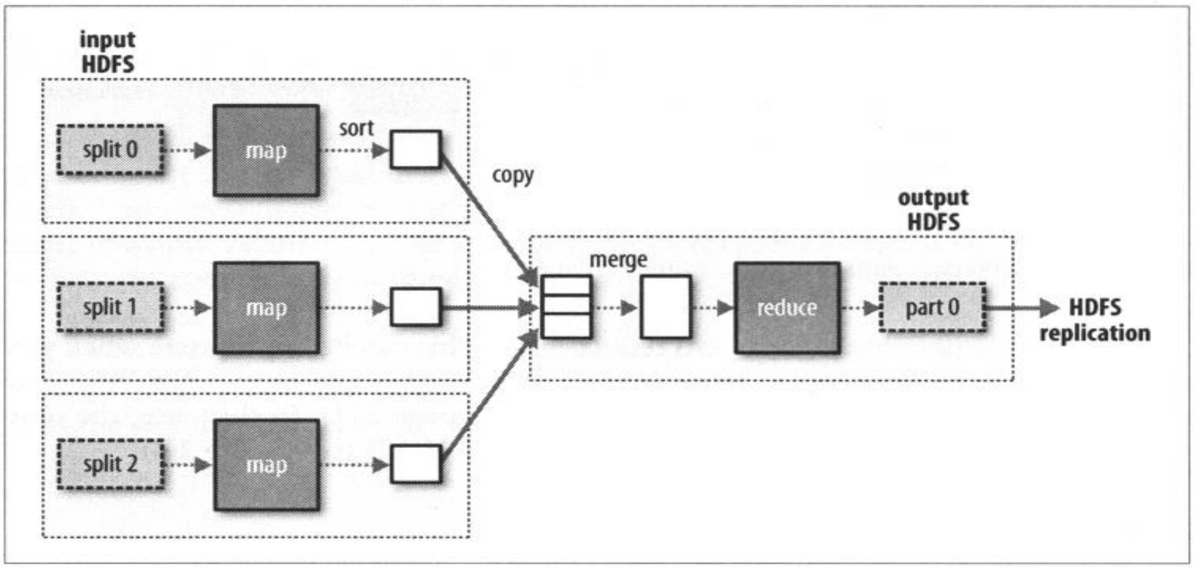


**Fig 1:** Map-Reduce dataflow diagram

### C. Inefficiencies in Existing Methods for Thumbnail creation

Thumbnail is a term used by graphic designers and photographers for a small image representation of a larger image, usually intended to make it easier and faster to look at or manage a group of larger images. For example, software that lets you manage a number of images often provides a miniaturized version of each image so that you don't have to remember the file name of each image. A thumbnail is also used to mean a small and approximate version of an image or a brochure layout as a preliminary design step. Web sites with many pictures, such as online stores with visual catalogs, often provide thumbnail images instead of larger images to make the page download faster. The user has the power to decide which images need to be seen in full size.

The basic idea of the thumbnail is that you can offer a page full of small images, and each one is linked to the full version of the image, giving your readers the option to preview any images they think they'd like to see in their full splendor, thus reducing hugely the download time of a page. There are different programming methods to generate the thumbnail of a images. The programming methods include HTML, C++, C, Java etc. All these method include either resize or crop of the image. Resizing the image gives an overview of everything in the picture, but they miss out on detail .With the cropped version, they get a much better feel for what the full picture will contain, and there is also a sense of wanting to know what else is in the picture that is not being shown. But all these methods are only processing very small size of data and throughput is very low and overall very time consuming.

## III. TOWARDS BETTER PERFORMANCE WITH HADOOP

Hadoop is a software framework that demands a different perspective on data processing. It has its own file system, HDFS that stores data in a way optimized for data-intensive processing. We need specialized Hadoop tools to work with HDFS, but fortunately most of those tools follow familiar Unix or Java syntax. For Hadoop installation, first the system must be configured. The Linux environment provide the reliable environment which high scalability, flexibility and compatibility .The implementation of the application with the help of Hadoop start with designing of Hadoop cluster favorable for the application. The next phase includes setting up the nodes and installing the required software's. On a fully configured cluster, "running Hadoop" means running a set of daemons, or resident programs, on the different servers in our network. These daemons have specific roles; some exist only on one server, some exist across multiple servers. Hadoop is a open source, which supports different programming languages. This facilitates an independent platform to the users. Though most of the coding in Hadoop happens in Java other languages such as Python, C# can be in cooperated. When setting up a Hadoop cluster , we need to designate one specific node as the master node. this server will typically host the NameNode and JobTracker daemons. It'll also serve as the base station contacting and activating the DataNode and TaskTracker daemons on all of the slave nodes. As such, we need to define a means for the master node to remotely access every node in our cluster. Hadoop uses passphrase less SSH for this purpose. SSH utilizes standard public key cryptography to create a pair of keys for user verification—one public, one private. The public key is stored locally on every node in the cluster, and the master node sends the private key when attempting to access a remote machine. With both pieces

of information, the target machine can validate the login attempt. The first step is to check whether SSH is installed on our nodes. We can easily do this by use of the `"which"` UNIX command. Having verified that SSH is correctly installed on all nodes of the cluster, we use `sshkeygen` on the master node to generate an RSA key pair . Be certain to avoid entering a passphrase, or we'll have to manually enter that phrase every time the master node

attempts to access another node. The first thing you need to do is to specify the location of Java on all the nodes including the master. In hadoop-env.sh define the `JAVA_HOME` environment variable to point to the Java installation directory. On our servers, we've it defined as:

```
export JAVA_HOME=/usr/share/jdk
```

The hadoop-env.sh file contains other variables for defining your Hadoop environment, but `JAVA_HOME` is the only one requiring initial modification. Hadoop file commands take the form of `hadoop fs -cmd <args>` where cmd is the specific file command and <args> is a variable number of arguments. The command cmd is usually named after the corresponding UNIX equivalent. For Example, the command for listing files is 2

```
hadoop fs -ls
```

This application requires handling images and generation of corresponding thumbnails. Hadoop provides the perfect potpourri of components to foster this application. As it provide us with the independent platform to work with any languages. Hadoop supports other languages via a generic API called Streaming. In practice, Streaming is most useful or writing simple, short MapReduce programs that are more rapidly developed in a scripting language that can take advantage of non-Java libraries. This Streaming help us to develop the python codes for the application

## IV. IMAGE THUMBNAIL CREATION

Hadoop has no method to read images, which was the main challenge for us. To implement we had to develop mapper and reducer code in python. The images whose thumbnails have to be generated are stored in a separate folder. The UNIX shell scripting method is used to run the commands. The mapper and reduce logic are shown in Algorithm1 and Algorithm2 respectively.

**Algorithm1:** Image Thumbnail Creation: Map Logic

```
inp <= sys.stdin, arr   <= [ ]
for all line in input do
    line <= line.strip ()
    if line <=’  ‘then
      l <= len (arr)
      if l > zero then
          Call get_image_from_string (contents) into img
          Generate img.thumbnail (150,150)
```

```
        Opp_arr <= get_string_from_image (img).split
          else
            arr.append (line)
        endif
    endif
```

Algorithm1 explains how images are taken as input. The mapper under Streaming reads a split through `STDIN` and extracts each line as a record. The mapper can choose to interpret each input record as a key/value pair or a line of text. Here images are taken in JPEG format. We have a IMG shell script to convert input images into string, which can be read by Hadoop. This IMG is called for all the image functions. File objects corresponding to the interpreter's standard input, output and error streams. stdin is used for all interpreter input except for scripts but including calls to input () and raw input(). The input is read from input folder, it is resized to (150X150) format. In mapper program the images are again converted back to string format, its resized and stored in a intermediate file, Opp_arr .

**Algorithm2:** Image Thumbnail Creation: All.sh and Reduce Logic

```
All.sh:   Call generate_inputs.sh
          Call run.sh
          Call generate_op_images .sh
Run.sh:   hadoop dfs -rmr $INPUT
          hadoop dfs -rmr $OUTPUT
          hadoop dfs -copyFromLocal input
          $INPUT hadoop jar streaming.jar -mapper $FOLDER/mapper.py reducer
          $FOLDER/reducer.py -input $INPUT -output $OUTPUT
          rm -rf op
          hadoop dfs -copyToLocal $OUTPUT op
Reducer: for all line in sys.stdin do
            In <= line.split ('\t ' )
            Print In.strip ()
           end
```

Algorithm2 explicates the final part of the application.Run.sh has commands removes input and output directory if exists. Copy the images to INPUT folder and run mapper and reducer code .Finally copy output from HDFS to local machine. Here strings are again converted back to images. The images will be of thumbnail size. In order to reduce the time for running the code we have made a final shell script All.sh.Fig1,2 show how the output of all.sh where all the image as matrix and its size.

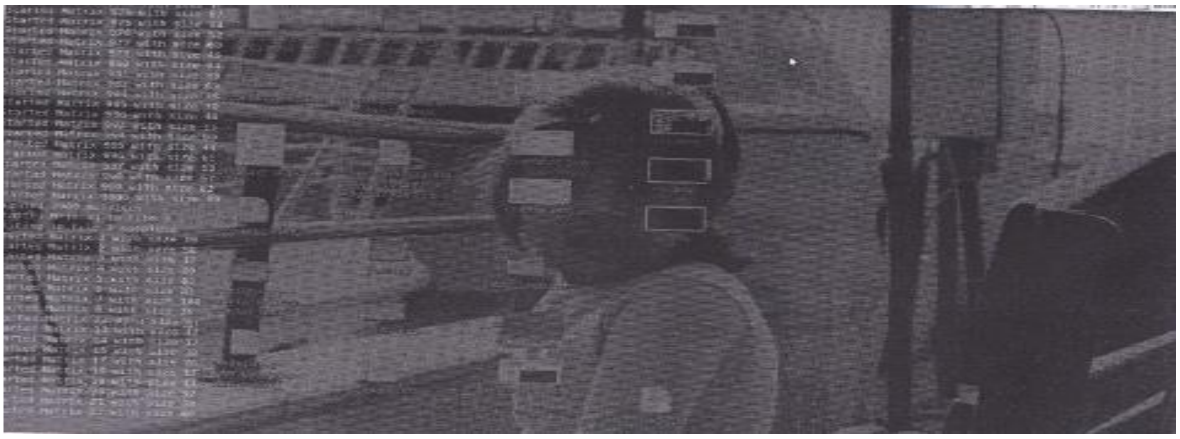


**Fig 2, 3:** The output of all.sh

## V. EXPERIMENTAL EVALUVATION

Experimental evaluation is a dynamic technique of verification and validation, which ensures that the software meets the expectations of the user. It involves executing an implementation of the software with test data. Test data is the set of data that the system will process as a normal input. System testing verifies that whole set of programs hang together. It makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate/non-testing leads to errors. The whole system has been tested and found to be working.

System testing follows the logical conclusion that is all the part o the system are tested and found to be working properly under all kinds of situations, then the system is achieving its goal of processing the data perfectly according to the user rules and requirements. A good test should be neither too simple nor too complex. A good test has high probability of finding an error and it should be "Best of Breed"..Two main types of testing include unit testing and integration testing.

The first level of testing is called Unit testing .Here all modules are tested separately. The nodes are checked with the sample programs, i.e., calculation of pi, word count given by Hadoop itself and are executed with the help of jar file. The first program was executed successfully. The input to second program was a number of text files and the output was each word in the file followed by its number of occurrences. The second level of testing is integration testing. The integration testing is to confirm that the single nodes will also show perfect result when run as a cluster. This testing activity can be considered as the design and emphasizes on testing modules interaction. Here single nodes are linked to form a cluster and the same programs executed in individual systems are once again executed in individual systems are once again executed in the master, and verified the result.

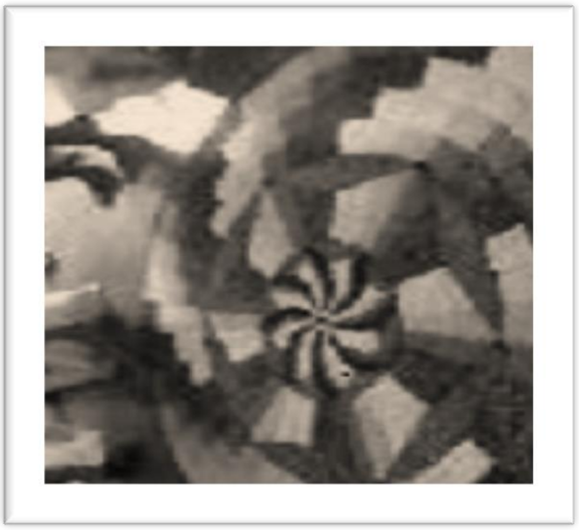


**Figure 3, 4:** The original image and it thumbnail

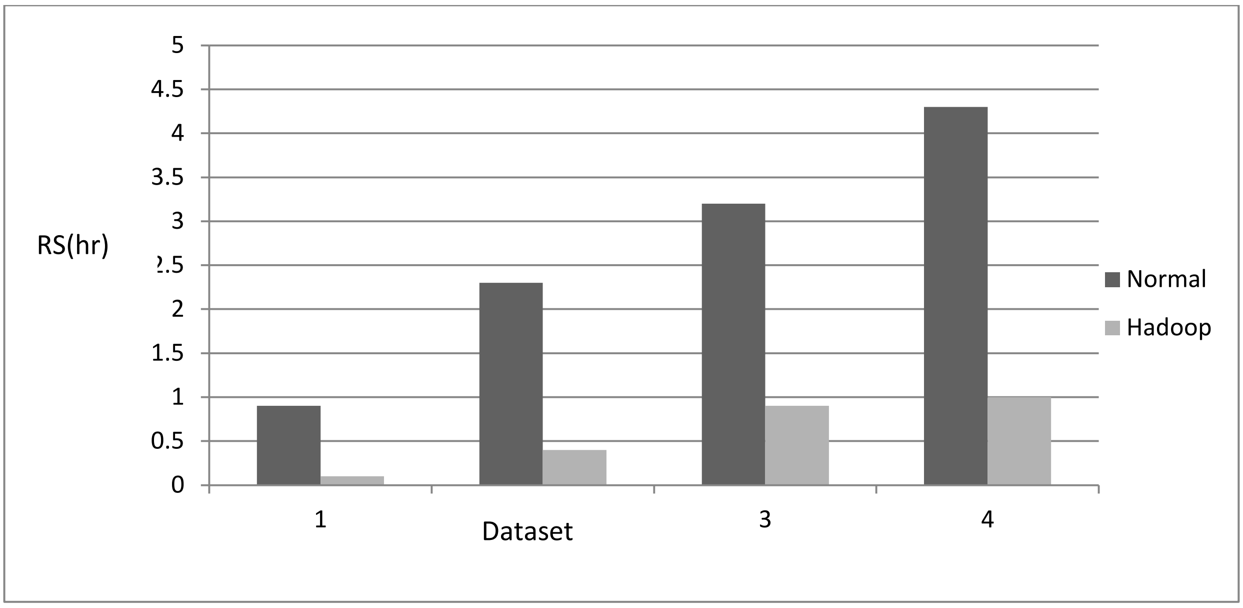


**Figure 5:** The x-axis is response time (RS (hours)) and y-axis is the size of dataset (MB)

After setting up multimode cluster, we have done the application, and a number of images are given as input to the master, and verified the output which was the thumbnails of input images. Then we analyzed the normal method with this adaptive method and the results are extrapolated in Figure5.

## VI. CONCLUSION

Depending on our data processing needs, Hadoop workload can vary widely over time. We may have a few large data processing jobs that occasionally take advantage of hundreds of nodes, but those same nodes will sit idle the rest of the time. We may be new to Hadoop cluster and we may own a startup that needs to conserve cash and wants to avoid the capital expense of a Hadoop cluster. In these and other situations, it make more sense to rent a cluster of machines rather than buy it. Cloud infrastructure is a great place for running Hadoop, as it allows us to easily scale to hundreds of nodes and gives us the financial flexibility to avoid upfront investments.

The rapid adoption of Hadoop at Facebook, Flicker, and Yahoo has been aided by a couple of key decisions. First, developers are free to write map-reduce programs in the language of their choice. Second, we have embraced SQL as a familiar paradigm to address and operate on large data sets. Most data stored in Hadoop’s file system is published as Tables. Developers can explore the schemas and data of these tables much like they would do with a good old database. When they want to operate on these data sets, they can use a small subset of SQL to specify the required datasets. Operations on datasets can be written as map and reduce scripts or using standard query operators or as a mix of the two. Over time, they have added classic data warehouse features like partitioning, sampling and indexing to this environment. This in-house data warehousing layer over Hadoop is called Hive and they are looking forward to releasing an open source version of this project in the near future.

Thumbnails are reduced-size versions of pictures, used to help in recognizing and organizing them, serving the same role for images as a normal text index does for words. In the age of digital images, visual search engines and image-organizing programs normally use thumbnails, as do most modern operating systems or desktop environment ,such as Microsoft Windows, Mac OS,X,KDE and GNOME .Some web designers produce thumbnail image on a web image is to reduce download time. The visual quality of browser resizing is also usually less than ideal. Displaying a significant part of the picture instead of the full frame can allow using a smaller thumbnail while maintaining recognizably. For example, when creating thumbnail of person, it may be better to show the face slightly reduced than an indistinct figure. This has the disadvantage that it misleads viewers about what the image contains, so it is less well suited for searching or a catalogue than for artistic presentations. Hadoop provides the fastest way to resize your pictures and images. The power comes from the simultaneous generation of different sizes and the ability to enlarge your images too.

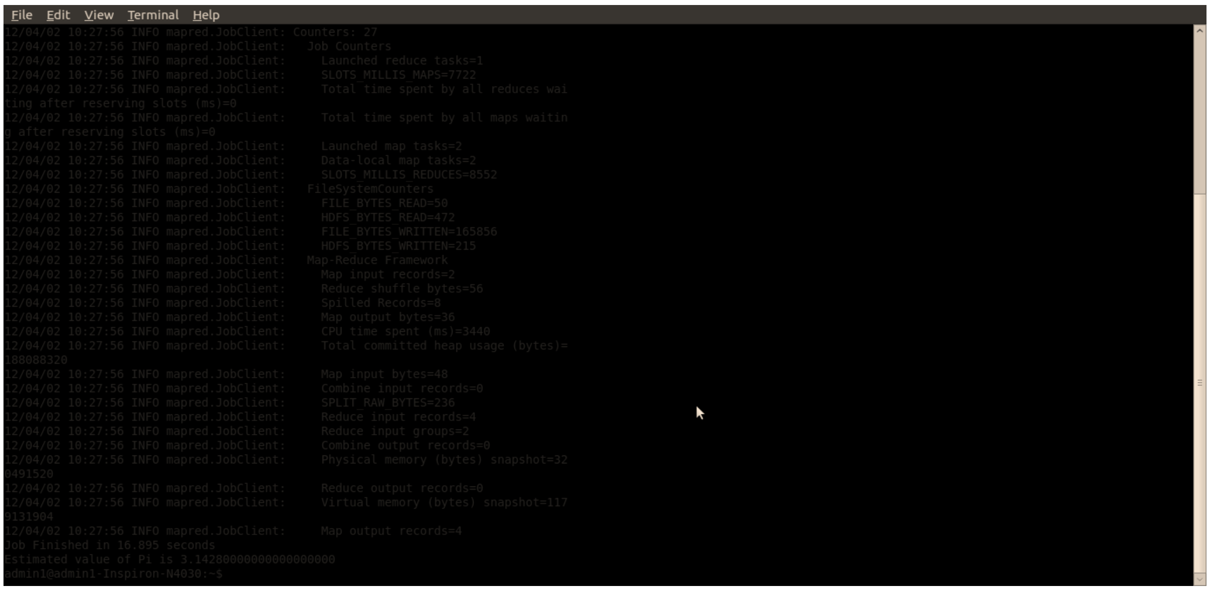
## ACKNOWLEDGEMENT

The project is supported by Computer Science Department of University of Calicut. The faculty members especially Mrs. Usha Dileep who was the guide of my research work and provided her full support. The assistant technicians helped to configure the hardware equipments and helped me in case of software failure I would like to thank all my batch mates for their help to complete this work.

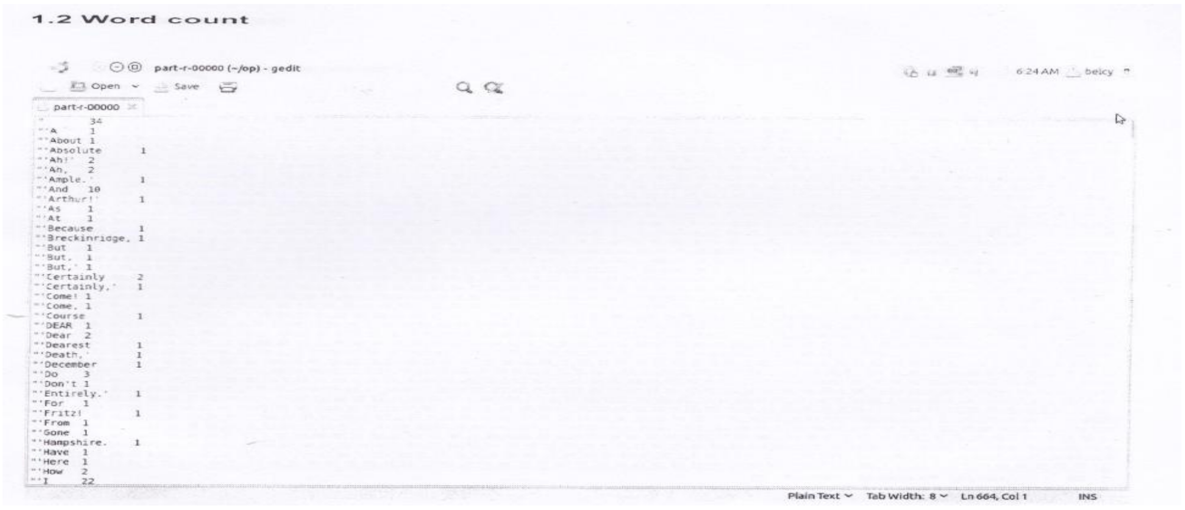
## APPENDIX - A

The screen shot of all the output I got during my testing. It includes testing of single-node Hadoop cluster and implementing Coudera sample programs.

### *1) Computation of pi*



```
12/04/02 10:27:56 INFO mapred.JobClient: Counters: 27
12/04/02 10:27:56 INFO mapred.JobClient:   Job Counters
12/04/02 10:27:56 INFO mapred.JobClient:     Launched reduce tasks=1
12/04/02 10:27:56 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=7722
12/04/02 10:27:56 INFO mapred.JobClient:     Total time spent by all reduces wai
ting after reserving slots (ms)=0
12/04/02 10:27:56 INFO mapred.JobClient:     Total time spent by all maps waitin
g after reserving slots (ms)=0
12/04/02 10:27:56 INFO mapred.JobClient:     Launched map tasks=2
12/04/02 10:27:56 INFO mapred.JobClient:     Data-local map tasks=2
12/04/02 10:27:56 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=8552
12/04/02 10:27:56 INFO mapred.JobClient:   FileSystemCounters
12/04/02 10:27:56 INFO mapred.JobClient:     FILE_BYTES_READ=50
12/04/02 10:27:56 INFO mapred.JobClient:     HDFS_BYTES_READ=472
12/04/02 10:27:56 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=165856
12/04/02 10:27:56 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=215
12/04/02 10:27:56 INFO mapred.JobClient:   Map-Reduce Framework
12/04/02 10:27:56 INFO mapred.JobClient:     Map input records=2
12/04/02 10:27:56 INFO mapred.JobClient:     Reduce shuffle bytes=56
12/04/02 10:27:56 INFO mapred.JobClient:     Spilled Records=8
12/04/02 10:27:56 INFO mapred.JobClient:     Map output bytes=36
12/04/02 10:27:56 INFO mapred.JobClient:     CPU time spent (ms)=3440
12/04/02 10:27:56 INFO mapred.JobClient:     Total committed heap usage (bytes)=
188088320
12/04/02 10:27:56 INFO mapred.JobClient:     Map input bytes=48
12/04/02 10:27:56 INFO mapred.JobClient:     Combine input records=0
12/04/02 10:27:56 INFO mapred.JobClient:     SPLIT_RAW_BYTES=236
12/04/02 10:27:56 INFO mapred.JobClient:     Reduce input records=4
12/04/02 10:27:56 INFO mapred.JobClient:     Reduce input groups=2
12/04/02 10:27:56 INFO mapred.JobClient:     Combine output records=0
12/04/02 10:27:56 INFO mapred.JobClient:     Physical memory (bytes) snapshot=32
0491520
12/04/02 10:27:56 INFO mapred.JobClient:     Reduce output records=0
12/04/02 10:27:56 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=117
9131904
12/04/02 10:27:56 INFO mapred.JobClient:     Map output records=4
Job Finished in 16.895 seconds
Estimated value of Pi is 3.14280000000000000000
admin1@admin1-Inspiron-N4030:~$
```


1.2 Word count

```
"       34
"'A     1
"'About 1
"'Absolute      1
"'Ah!'  2
"'Ah,   2
"'Ample.'       1
"'And   10
"'Arthur!'      1
"'As    1
"'At    1
"'Because       1
"'Breckinridge, 1
"'But   1
"'But.  1
"'But,' 1
"'Certainly     2
"'Certainly,'   1
"'Come! 1
"'Come, 1
"'Course        1
"'DEAR  1
"'Dear  2
"'Dearest       1
"'Death,'       1
"'December      1
"'Do    3
"'Don't 1
"'Entirely.'    1
"'For   1
"'Fritz!        1
"'From  1
"'Gone  1
"'Hampshire.    1
"'Have  1
"'Here  1
"'How   2
"'I     22
```

## REFERENCES

[1] http://hadoop.apache.org

[2] http://en.wikipedia.org/wiki/Hadoop

[3] http://cloudera.com

[4] Raman Grover,"Extending MapReduce for efficient predicate based sampling",pg 1150-1152