ISSN No : 2454-423X (Online)



International Journal of Research in Advanced Computer Science Engineering

> A Peer Reviewed Open Access International Journal www.ijracse.com

## **Image Clustering Using Hadoop Image Processing Interface**

Damarla Haritha

Department of Computer Science and Engineering MVSR Engineering College, Hyderabad, Telangana - 501510, India.

#### Abstract

Huge amounts of images are uploaded into the internet daily. As large image collections cannot be processed efficiently on one computer, image processing often requires distributed computing. Image processing can be very computationally demanding due to the large amount of images to process. Image Processing with Parallel computing is an alternative way to solve image processing problems that require large times of processing or handling large amounts of information in "acceptable time". The main idea of parallel image processing is to divide the problem into simple tasks and solve them concurrently, in such a way the total time can be divided between the total tasks (in the best case).

As there is no such distributed framework for processing of images, HIPI (Hadoop Image Processing Interface) allows image processing on distributed framework. Hadoop provides distributed computing framework for data but not for images. Images cannot be processed using hadoop, inorder to analyse large datasets of images, there is an interface for image i.e hadoop image processing processing interface(HIPI). HIPI is an interface over the hadoop file system which helps the users to run various image processing algorithms and analyse the images on a distributed framework. Hipi provides such a mechanism to process huge data sets of images. The Analysed images are then clustered using the em clustering algorithm

*Key words: Image processing, Parallel Computing, Distributed Computing, HIPI* 

#### **INTRODUCTION**

Distributed computing is a method of computer processing in which different parts of a program are run simultaneously on two or more computers that are communicating with each other over a network. Distributed computing is a type of segmented or parallel computing, but the latter term is most commonly used to refer to processing in which different parts of a program run simultaneously on two or more processors that are part of the same computer. While both types of processing require that a program be segmented divided into sections that can run simultaneously, distributed computing also requires that the division of the program take into account the different environments on which the different sections of the program will be running.

A Framework in which large problems can be divided into many small problems which are distributed to many computers. Later, the small results are reassembled into a larger solution. Distributed computing [1] is a natural result of using networks to enable computers to communicate efficiently.

Distributed Computing framework is used for creating and using compute clusters to execute computations in parallel across multiple processors in a single machine (SMP) [3-5], among many machines in a cluster, grid or cloud.

Distributed computing is the process of aggregating the power of several computing entities, which are logically distributed and may even be geologically distributed, to collaboratively run a single computational task in a transparent and coherent way, so that they appear as a single, centralized system [2]. Parallel computing is the



simultaneous execution of the same task on multiple processors in order to obtain faster results. It is widely accepted that parallel computing is a branch of distributed computing, and puts the emphasis on generating large computing power by employing multiple processing entities simultaneously for a single computation task.

These multiple processing entities can be а multiprocessor system, which consists of multiple processors in a single machine connected by bus or switch networks, or a multicomputer system, which consists of several independent computers interconnected by telecommunication networks or computer networks. Besides in parallel computing, distributed computing has also gained significant development in enterprise computing. The main difference between enterprise distributed computing and parallel distributed computing [4] is that the former mainly targets on integration of distributed resources to collaboratively finish some task, while the later targets on utilizing multiple processors simultaneously to finish a task as fast as possible.

#### LITERATURE STUDY Distributed Computing Framework

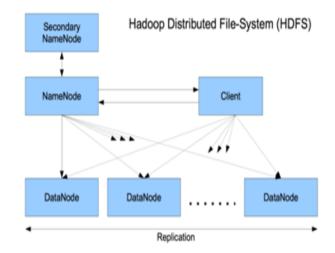
Distributed computing is a method of computer processing in which different parts of a program are run simultaneously on two or more computers that are communicating with each other over a network.

Distributed computing is a type of segmented or parallel computing, but the latter term is most commonly used to refer to processing in which different parts of a program run simultaneously on two or more processors that are part of the same computer. While both types of processing require that a program be segmented divided into sections that can run simultaneously, distributed computing also requires that the division of the program take into account the different environments on which the different sections of the program will be running. A Framework in which large problems can be divided into many small problems that are distributed to many computers is called distributed framework. Later, the small results are reassembled into a larger solution. Distributed computing is a natural result of using networks to enable computers to communicate efficiently.

Distributed Computing framework is used for creating and using compute clusters to execute computations in parallel across multiple processors in a single machine (SMP), among many machines in a cluster, grid or cloud.

#### **Apache Hadoop**

The Apache Hadoop MapReduce project is a distributed computing framework that enables developers to write applications which run reliably on a large number of unreliable machines with the goal of processing Terabyte and larger data sets in paralleling clusters consisting of thousands of nodes [3]. Hadoop is an open source implementation of the MapReduce framework inspired by Google MapReduce, and the Google File System (GFS)[12], although the two systems are very different. The Hadoop Distributed File System, inspired by GFS from Google [2], is a distributed filesystem which runs on low cost commodity hardware in a fault tolerant manner to redundantly store Terabyte and larger data sets.



Volume No: 2 (2017), Issue No: 9 (February) www. IJRACSE.com

February 2017



#### Hazelcast

Hazelcast[2] is an open source in-memory data grid based on Java. The Hazelcast[2] company is funded by venture capital. In a Hazelcast[2] grid, data is evenly distributed among the nodes of a computer cluster, allowing for horizontal scaling both in terms of available storage space and processing power.

JPPF[3] is a distributed parallel processing framework based on a master/worker architecture. A JPPF grid is made of 3 sorts of components that communicate with each other: clients which submit the work to the grid, nodes which execute the work, and servers which receive the work from clients and distribute it to the nodes in parallel.

#### **Apache spark**

Apache Spark[7] is an in-memory distributed data analysis platform that speeds up task processing. It provides high-level APIs in Java. It also supports a rich set of higher-level tools, including Shark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming [8], helping the development of parallel applications.

## HIPI (HADOOP IMAGE PROCESSING INTERFACE)

#### **The HIPI Framework:**

HIPI[15] is an open-source Hadoop Image Processing Interface that aims to create an interface for Image Processing(IP) with Map Reduce technology. HIPI abstracts the highly technical details of Hadoop's system and is flexible enough to implement IP algorithms.

The following goals of HIPI are:

1. HIPI provides an open, extensible library for IP using Map Reduce technology.

2. With the help of HIPI, images are stored into the HDFS easily.

- 3. HIPI allows simple filtering of set of images.
- 4. Simple and unambiguous interface for IP in hadoop.
- 5. Enhances parallel processing of images.

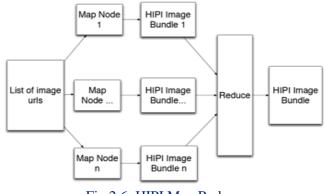
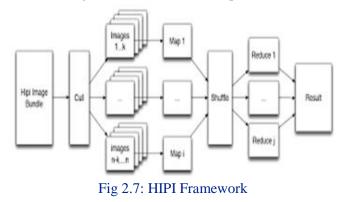


Fig 2.6: HIPI Map Reduce

The following work flow of HIPI is represented as:



#### **Data Storage:**

A HIPI Image Bundle data type that stores many images in one large file so that MapReduce jobs can be performed more efficiently. A HIPI [4] Image Bundle consists of two files: a data file containing concatenated images and an index file containing information about the offsets of images in the data file as shown in the below fig.

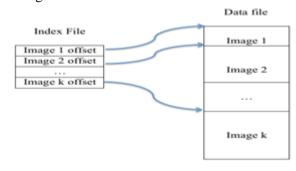


Fig 2.8: A depiction of the relationship between the index and data files in a HIPI Image Bundle



Volume No:2, Issue No:9 (February-2017)

ISSN No: 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

> A Peer Reviewed Open Access International Journal www.ijracse.com

#### **Downloading Images:**

The following steps are used for downloading the images. They are:

1. The list of images should be stored by their image urls and should be present in a text file with exactly one image url per line.

2. For parallel execution, the input list of image urls are split and the number of nodes used to download these images are specified.

3. Download the images from the internet using the url's.

4. The downloaded images are stored in the image bundle (HIB).

#### **XHAMI (Extended HDFS and Map Reduce Interface for Image Processing Applications)**

Image processing applications deal with processing of pixels in parallel, for which Hadoop and MapReduce can be effectively used to obtain higher throughputs. XHAMI [4] offers extended library of HDFS [5] and MapReduce to process the single large scale images with high level of abstraction over writing and reading the images. XHAMI [4] has an API to implement Image processing that is two phase extensions to HDFS and MapReduce programming model.



Fig 3.1: Flowchart of environmental set up of Hadoop, HIPI,Gradle

### ALGORITHM TO COMPUTE GREY LEVEL HISTOGRAM OF AN IMAGE

#### Algorithm:

Input: image, no. Of bins

Output: histogram values.

Steps:

1. Read the Input image .

2. Obtain the histogram of the input image by traversing image pixel data in raster-scan order and update running average in the bins.

#### Usage:

Haritha@ Haritha-Vostro-3549:~/work/hipi\$ hadoop jar build/libs/helloWorld.jar tigers.hib hist1

# ALGORITHM TO COMPUTE MEAN VALUES OF AN IMAGE

#### Algorithm:

**Input**: image **Output**: average values.

#### Steps:

1. Read the Input image.

2. Obtain the mean values of the input image by averaging image pixel data and update running averages cumulatively.

#### Usage:

Haritha@ Haritha - Vostro-3549:~/work/hipi\$ hadoop jar build/libs/helloWorld.jar tigers.hib average

#### EM CLUSTERING ALGORITHM

The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin. They pointed out that the method had been "proposed many times in special circumstances" by earlier authors. In particular, a very detailed treatment of the EM method [6] for exponential families was published by Rolf Sundberg in his thesis and several papers following his collaboration with Per Martin-Löf and Anders Martin-Löf.

#### **Algorithm Overview**

An expectation-maximization (EM) algorithm is an



iterative method for finding maximum likelihood or maximum a posteriori (MAP) [7] estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and maximization (M) step, which computes parameters maximizing the expected loglikelihood found on the E step. These parameterestimates are then used to determine the distribution of the latent variables in the next E step.

#### **EM Clustering**

EM is an algorithm for maximizing a likelihood function when some of the variables in your model are unobserved (i.e. when you have latent variables).

To maximize a function, why don't we just use the existing machinery for maximizing a function. Well, if you try to maximize this by taking derivatives and setting them to zero, you find that in many cases the first-order conditions don't have a solution. There's a chicken-and-egg problem in that to solve for your model parameters you need to know the distribution of your unobserved data; but the distribution of your unobserved data is a function of your model parameters.

E-M tries to get around this by iteratively guessing a distribution for the unobserved data, then estimating the model parameters by maximizing something that is a lower bound on the actual likelihood function, and repeating until convergence - The EM algorithm Starts with guess for values of your model parameters.

#### **E-step**

For each data point that has missing values, use your model equation to solve for the distribution of the missing data given your current guess of the model parameters and given the observed data (note that you are solving for a distribution for each missing value, not for the expected value).

Now that we have a distribution for each missing value, we can calculate the expectation of the likelihood function with respect to the unobserved variables. If our guess for the model parameter was correct, this expected likelihood will be the actual likelihood of our observed data; if the parameters were not correct, it will just be a lower bound.

#### **M-step**

Now that we've got an expected likelihood function with no unobserved variables in it, maximize the function as you would in the fully observed case, to get a new estimate of your model parameters. Repeat until convergence.

#### **Applications**

EM is frequently used for data clustering in machine learning and computer vision. In natural language processing, two prominent instances of the algorithm are the Baum-Welch algorithm [8] (also known as forwardbackward) and the inside-outside algorithm for unsupervised induction of probabilistic context-free grammars.

In psychometrics, EM is almost indispensable for estimating item parameters and latent abilities of item response theory models.

With the ability to deal with missing data and observe unidentified variables, EM is becoming a useful tool to price and manage risk of a portfolio [9].

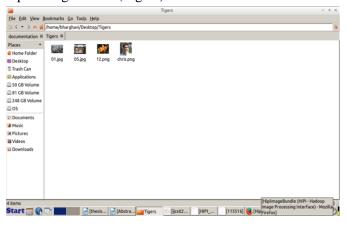
The EM algorithm (and its faster variant Ordered subset expectation maximization) is also widely used in medical image reconstruction, especially in positron emission tomography and single photon emission computed tomography. See below for other faster variants of EM.

#### **RESULTS**

This chapter shows the results of an image dataset of 1000 images and then the images are clustered based on



#### their grey level histogram. Input image folder (Tigers):



#### Input image in HDFS (Tigers.hib):

Hadoop Overv	iew Datanodes	Snapshot Startup	Progress Utiliti	es -			
Browse [	Directory						
/user/bharghavi							Go
Permission	Owner	Group	Size	Replication	Block Size	Name	
-09-0-0-	bharghavi	supergroup	56 B	1	128 MB	Tigers.hib	
-09-0-0-	bharghavi	supergroup	1.43 MB	1	128 MB	Tigers.hib.dat	
drwxr-xr-x	bharghavi	supergroup	08	0	0.8	hist3	
-fW-ff	bharghavi	supergroup	87.04 KB	1	128 MB	hist3.txt	
-FW-FF	bharghavi	supergroup	7.84 KB	1	128 MB	images.hib	
-FW-FF	bharghavi	supergroup	28.55 MB	1	128 MB	images.hib.dat	
Hadoop, 2014.							

#### **Grey level Histogram Results:**

The following shows the results of 4 images that are given as input to the histogram algorithm and the histogram values of the images are obtained respectively.

#### Table 6.1: List of Grey level Histogram values

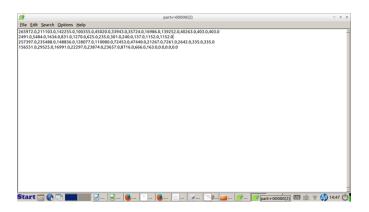
image	Histogram values
1	25329.00,21389.00,12901.00,12628.00,7554.00,8215.00,9730.00,517.00,40.00,1.00,0.00,0.00
2	58090.00, 14104.00, 8136.00, 5624.00, 3218.00, 2950.00, 5686.00, 427.00, 61.00, 8.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.
3	6565.00, 12484.00, 12100.00, 7669.00, 5671.00, 7977.00, 8007.00, 36120.00, 1666.00, 45.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0
4	11801.00,22788.00,11363.00,10993.00,20241.00,5418.00,11950.00,2812.00,865.00,67.00,6.00,6.00

#### Mean value Results:

The following shows the results of 4 images that are given as input to the mean algorithm and the mean values of the images are obtained respectively.

#### Table 6.2: List of Mean values

image	Mean values
1	265972.0,211103.0,142255.0,100355.0,45020.0,33943.0,35724.0,16986.0,139252.0,40263.0,
	403.0,403.0
2	2491.0, 5484.0, 1634.0, 831.0, 1270.0, 625.0, 235.0, 301.0, 240.0, 137.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 1152.0, 11
3	257397.0, 235488.0, 148836.0, 128077.0, 110080.0, 72453.0, 47440.0, 21267.0, 7261.0, 2642.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 128077.0, 1280777.0, 128077.0, 128077.0, 1280777.0, 128077.0, 128077.0, 128077.0,
	335.0,335.0
4	156551.0, 29525.0, 16991.0, 22297.0, 23874.0, 23657.0, 8716.0, 666.0, 163.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,



#### **EM Clustering**

The following shows the results of Clustering 1000 images on 10 clusters using EM Clustering algorithm on a single node.

The 1000 images are put into a folder on the physical system

\$mkdir /home/bharghavi/Desktop/images

Volume No:2, Issue No:9 (February-2017)

ISSN No: 2454-423X (Online)



## International Journal of Research in Advanced Computer Science Engineering

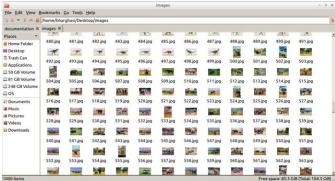
A Peer Reviewed Open Access International Journal www.ijracse.com

documentation #	images #												
Places •			and the same	420	100	Contra Co	and the second	100	100	25	P.C.A	1	
Home Folder	0.jpg	1.00	2.00	3.jpg	4.jpg	Sjpg	6.300	7.109	8.jpg	9.jpg	10.109	11.jpg	
Trash Can	10		TAT	1	被	Sile	1	-	1	1	1990 C		
Applications 50 GB Volume	12.300	13.300	14.00	15.100	16,00	17.jpg	18.jpg	19.jpg	20.00	21.100	22.309	23.100	
81 GB Volume	-	C.M.M.	1	100	24	10	in the	100	and a	-	4	10	
248 GB Volume	24.jpg	25 jpg	26.jpg	27.jpg	28.jpg	29.jpg	30.jpg	31,000	32.jpg	33.jpg	34.jpg	35./pg	
Documents	10.20	and the second	200	Dist.		-		-		Sec	and the second	10	
Music	36.jpg	37.jpg	38.jpg	39.jpg	40.jpg	41.jpg	42.jpg	43.00	44.jpg	45 jpg	46.jpg	47.jpg	
Pictures Videos	20	Fighter of	-	1	THE .	1	1	-	100	ALC: N	-	and a	
Downloads	48.jpg	49.jpg	50.jpg	S1.jpg	52.jpg	53.jpg	54.jpg	ssjog	56.jpg	S7 jpg	SBjpg	59.jpg	
149102/020	1200		201	1	1000	1	1	and the	1	1	100	38	
	60.jpg	61.jpg	62.jpg	63.jpg	64.jpg	65.jpg	66.jpg	67.300	68.jpg	69.jpg	70.jpg	71.jpg	
	1	2	12	-	1	Ser.	1	4.			12	1000	
	72.jpg	73.000	74.jpg	75.00	76.jpg	77.jpg	78.jpg	79.00	80.00	81.jpg	82.jpg	83.109	

2						images							- +
Elle Edit View B	ookmarks §	Go Tools I	Help										
1 < + > A @	/home/bhan	rghavi/Desk	top/images										
documentation ¥													
Places *	84.jpg	85.jpg	86.jpg	87.jpg	88.jpg	89.jpg	90.jpg	91.jpg	92.jpg	93.jpg	94.jpg	95.jpg	
Home Folder		100	2.00	and the	CONTRACT OF	4	1	-		@	-	-	
Desktop Trash Can	96,00	97.jpg	98.jpg	99.jpg	100.jpg	101.jpg	102.jpg	103.jpg	104.jpg	105.jpg	106.jpg	107.jpg	
Applications	1. Al	100				1000	100	and the second	100	-	and the lot	-	
50 GB Volume	108.jpg	109.jpg	110,jpg	111.jpg	112.jpg	113.jpg	114.jpg	115.jpg	116.jpg	117.jpg	118.jpg	119.jpg	
81 GB Volume				-	The second second	_		-		_	100		
248 GB Volume		-	-	1-2	-	1000	Sec. 1	1.00	-	00000		-	
los	120.jpg	121.jpg	122.jpg	123.jpg	124.jpg	125.jpg	126.jpg	127.jpg	128.jpg	129.jpg	130.jpg	131.jpg	
Documents Music	Const.		-	10.0		-		100	de P	-	-		
Pictures	132.jpg	133.jpg	134.jpg	135.jpg	136.jpg	137.jpg	138.jpg	139.jpg	140.jpg	141.jpg	142.jpg	143.jpg	
I Videos	250		-	100	100		0.000			STATES.		1	
Downloads	144.jpg	145.jpg	146.jpg	147.jpg	148.jpg	149.jpg	150.jpg	151.jpg	152.jpg	153.jpg	154.jpg	155.jpg	
	and a	100	-		the state	in the	- 25			Sec.			
	156.ipg	157.jpg	158.jpg	159.jpg	160.jpg	161.jpg	162.jpg	163.jpg	164.jpg	165.jpg	166.jpg	167.jpg	
				_			-	_	_				
	<b>B A</b>		10	<u>_</u>	-	100	100	10.25	100	194	100	100	
000 items	168.ioa	169.ipg	170.ipg	171.ioa	172.ioa	173.ipg	174.ing	175 ipg	176 ipg	177.ipg	178.ing	179 inn iB (Total: 10	

-						images							- +
File Edit View B	ookmarks :	Go Tools	Help										
36-344	/home/bhai	ghavi/Deskl	top/images										_
documentation ¥	images #												
Places •		(Barrow)		Sec. 1	-	Contract of	(Constant)	Concession in which the	-	-	The second second		
Home Folder	A REAL PROPERTY AND A REAL	2	No.	100			1000	-	-		<b>_</b>		
III Desktop	372.jpg	373.jpg	374.jpg	375.jpg	376.jpg	377.jpg	378.jpg	379.jpg	380.jpg	381.jpg	382.jpg	383.jpg	
🗄 Trash Can	-		-			<b></b>			-	-	-	<b>C</b>	
Applications	384.jpg	385.jpg	386.jpg	387.jpg	388.jpg	389.jpg	390.jpg	391.jpg	392.jpg	393.jpg	394.jpg	395.jpg	
50 GB Volume										-		-	
248 GB Volume	<b>1</b>	1			75	×	A	18	A	TRK	*	1	
os.	396.jpg	397.jpg	398.jpg	399.jpg	400.jpg	401.jpg	402.jpg	403.jpg	404.jpg	405.jpg	406.jpg	407.jpg	
Documents	-	R	*	A	m.	7	X	T	-	1	m	T	
A Music	408.jpg	409.jpg	410.jpg	411.jpg	412.jpg	413.jpg	414.jpg	415.jpg	416.jpg	417.jpg	418.jpg	419.jpg	
Pictures	×	T	T	×	R	×	1	T	×	*	T	×	
Downloads	420.jpg	421.jpg	422.jpg	423.jpg	424.jpg	425.jpg	426.jpg	427.jpg	428.jpg	429.jpg	430.jpg	431.jpg	
	1	*	T	- met	XX-	T	-	T	-	F	R	-	
	432.jpg	433.jpg	434.jpg	435.jpg	436.jpg	437.jpg	438.jpg	439.jpg	440.jpg	441.jpg	442.jpg	443.jpg	
	×	T	A	Y	×	m	*	-	-	×	-m	*	
	444.jpg	445.jpg	446.jpg	447.jpg	448.jpg	449.jpg	450.jpg	451.jpg	452.jpg	453.jpg	454.jpg	455.jpg	

Start 📷 🕐 🗁 📰 💽 the... 🕞 Abs... 📷 ima... 👘 ficsi... 👔 IHP... 🧧 [11... 📄 the... 💻 [bh... 🗮 201... 🔶 🖽 👘 10:11 ტ



1000 kens Start 👕 🕐 🗅 📰 💽 the... 🖸 Abs... 📷 ima... 🖆 (jicsl... [PHP... ] [11... ]] the... 💻 [bh... ]] 201... 4 💷 (b) 😤 (b) (cda: 104.3 chb)

-						images						
File Edit View B	ookmarks	Go Tools	Help									
34.344	/home/bhar	rghavi/Desk	top/images									
documentation ¥	images #											
Places • Home Folder	588.jpg	589.jpg	590.jpg	591.jpg	592.jpg	593.jpg	594.jpg	595.jpg	596.jpg	597.jpg	598.jpg	<b>4</b> 599.jpg
Trash Can Applications	600.jpg	601.jpg	602.jpg	603.jpg	604.jpg	605.jpg	606.jpg	607.jpg	608.jpg	609.jpg	610.jpg	611.jpg
81 GB Volume 248 GB Volume OS	612.jpg	613.jpg	614.jpg	615.jpg	616.jpg	617.jpg	618.jpg	619.jpg	620.jpg	621.jpg	622.jpg	623.jpg
Documents Music Pictures	624.jpg	625.jpg	626.jpg	627.jpg	628.jpg	629.jpg	630.jpg	631.jpg	632.jpg	633.jpg	634.jpg	635.jpg
E Videos Downloads	636.jpg	637.jpg	638.jpg	639.jpg	640.jpg	641.jpg	642.jpg	643.jpg	644.jpg	645.jpg	646.jpg	647.jpg
	648.jpg	649.jpg	6S0.jpg	6S1.jpg	652.jpg	653.jpg	654.jpg	655.jpg	656.jpg	657.jpg	658.jpg	659.jpg
	660.jpg	661.jpg	662.jpg	663.jpg	664.jpg	665.jpg	666.jpg	667.jpg	668.jpg	669.jpg	670.jpg	671.jpg



Volume No: 2 (2017), Issue No: 9 (February) www. IJRACSE.com

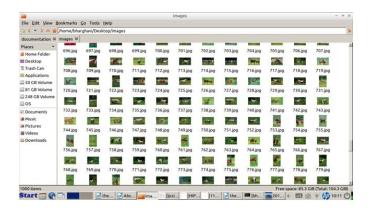
February 2017

Volume No:2, Issue No:9 (February-2017)

ISSN No: 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal www.ijracse.com



-						images							- +
File Edit View B	ookmarks :	Go Tools	Help										
34-344	/home/bhai	ghavi/Deskl	top/images										_
documentation ¥	images #												
Places •	200	1	15	3	-	-	1	100	10	-	0	38	
Home Folder					1			0.					
E Desktop	924.jpg	925.jpg	926.jpg	927.jpg	928.jpg	929.jpg	930.jpg	931.jpg	932.jpg	933.jpg	934.jpg	935.jpg	
Trash Can	1	0	9	22	10.00		<**S				5	1	
Applications	936.jpg	937.jpg	938.jpg	939.jpg	940.jpg	941.jpg	942.jpg	943.jpg	944.jpg	945.jpg	946.jpg	947.jpg	
81 GB Volume		Æ	-	100	<b>S</b>			Ser.		1		-	
248 GB Volume		949.jpg		347							958.jpg		
os	948.jpg		950.jpg	951.jpg	952.jpg	953.jpg	954.jpg	955.jpg	956.jpg	957.jpg		959.jpg	
Documents	1 m	2	100	600	1		100	1990			100	A STATE	
a Music	960.jpg	961.jpg	962.jpg	963.jpg	964.jpg	965.jpg	966.jpg	967.jpg	968.jpg	969.jpg	970.jpg	971.jpg	
Pictures	1		1		2	2	1		S.	10	0	1	
Videos  Downloads	972.jpg	973.jpg	974.jpg	975.jpg	976.jpg	977.jpg	978.jpg	979.jpg	980.jpg	981.jpg	982.jpg	983.jpg	
DOWING BUS	NO.	1000		-								1000	
	Mark.		1	<b>C M</b>	大学	্ৰেন্দ্ৰ	90	-			9	-	
	984.jpg	985.jpg	986.jpg	987.jpg	988.jpg	989.jpg	990.jpg	991.jpg	992.jpg	993.jpg	994.jpg	995.jpg	
	0	100	19 A										
	996.jpg	997.ipg	998.ipg	999.ipg									
1000 items												iB (Total: 104	_

These 1000 images are saved into the hdfs as images.hib. The screen shot is given below:

Hadoop one	rview Datanodes	File information - images.hib	×		
			- 10		
Browse	Directory	Download			
/user/bharghavi		Block information - Block 0			
Permission	Owner	Block ID: 1073744071	ze	Name	
-09-66	bharghavi	Block Pool ID: BP-616774020-127.0.1.1-1461940402318 Generation Stamp: 3247		Tigers.hib	
-fW-f=-f=-	bharghavi	Size: 8024		Tigers.hib.dat	
drwxr-xr-x	bharghavi	Availability:			
-110-11-1	bharghavi	<ul> <li>bharghavi-Vostro-3549</li> </ul>		hist3.txt	
-fW-ff	bharghavi			images.hib	
-19-1-1-	bharghavi		_	images.hib.dat	
			Close		
Hadoop, 2014.			_		

#### Conclusion

HIPI is an image processing map-reduce framework that is designed to hide the complex details of Hadoop's powerful Map-reduce framework for processing the images. HIPI provides a format for storing images for efficient access within the Map Reduce pipeline, and simple methods for creating and storing image files of the float type and is stored in the form of HIB. HIPI interface brings about a new level of simplicity for creating large-scale vision applications that use the mapreduce framework for the processing of images. The features of the images are extracted by HIPI framework and given to the EM clustering algorithm in Hadoop for clustering.

#### **Future Work**

The future work includes implementation of more complex Image processing algorithms on the proposed system with the integration of OpenCV tool on a multinode setup of HIPI.

#### REFERENCES

[1] Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. "The hadoop distributed file system." In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pp. 1-10. IEEE, 2010.

[2] Vemula, Sridhar, and Christopher Crick. "Hadoop Image Processing Framework." 2015 IEEE International Congress on Big Data. IEEE, 2015.

[3] Zaharia M, Das T, Li H, et al. (2012) Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. Proc. 4 th Edition.

[4] Apache Storm. https://storm.apache.org/. Accessed 27 Dec 2014.

[5] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. SIGOPS Oper. Syst. Rev., 37(5):29–43, 2003.

[6] Kune, Raghavendra, et al. "XHAMI--Extended HDFS and MapReduce Interface for Image Processing Applications." 2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). IEEE, 2015.

Volume No: 2 (2017), Issue No: 9 (February) www.IJRACSE.com



[7] Sweeney, Chris, et al. "HIPI: a Hadoop image processing interface for image-based mapreduce tasks." Chris. University of Virginia (2011).

[8] Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. "The hadoop distributed file system." In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pp. 1-10. IEEE, 2010.

[9] Tom White. Hadoop: The Definitive Guide. O'Reilly Media, Inc., 3rd edition, 2009.