

Image Retrieval Using Map-Reduce Algorithm

A Project Report
submitted in fulfillment of the requirement
for the degree of Bachelor of Technology
in

Computer Science & Engineering

under the Supervision of

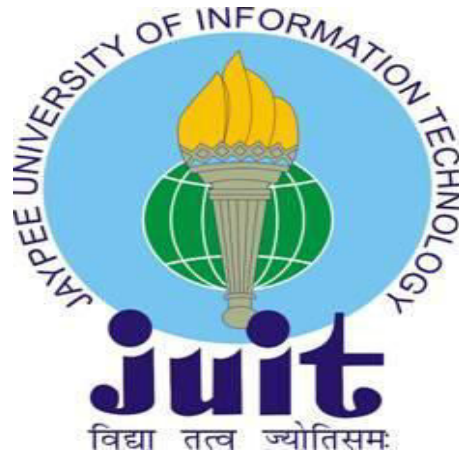
Dr. Hemraj Saini

By

Jatin Mittal (131227)

Rishabh Jain (131228)

To



Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **Image Retrieval Using Map-Reduce Algorithm**” in fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to April 2017 under the supervision of Dr.Hemraj Saini .

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Jatin Mittal (131227)
Rishabh Jain (131228)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Hemraj Saini
Assistant Professor
Computer Science & Engineering
28-04-2017

Acknowledgement

I would like to express my sincere gratitude towards Prof. Hemraj Saini, my research guide, for his patient guidance, enthusiastic encouragement and useful critiques on this research work. Without his invaluable guidance, this work would never have been a successful one.

The in-time facilities provided by the Computer Science department throughout the project development are also equally acknowledgeable.

Last but not least, my sincere thanks to all my teachers and my friends who directly or indirectly helped me to learn new technologies and complete my work successfully.

Jatin Mittal
Rishabh Jain

Table of Contents

Serial Number	Topics	Page Numbers
1	Chapter-1. Introduction	1
2	1.1 Introduction	1
3	1.2 Image Retrieval	2
4	1.2.1 Content Based Image Retrieval	2
5	1.3 Objectives	4
6	1.4 Methodology	4
7	1.4.1 Map-Reduce	9
8	Chapter-2 Literature Survey	10
9	2.1 PIRMAP	10
10	2.1.1 PIR	10
11	2.1.2 Map Reduce in PIRMAP	10
12	2.1.3 Working Of PIRMAP	11
13	2.1.4 Future Scope	12
14	2.2 CBIR	12
15	2.2.1 System Model	12
16	2.2.2 Future Scope	13
17	2.3 Oblivious Retrieval	13
18	2.3.1 System Model	14
19	2.3.2 Future Scope	15
20	2.4 Distributed Retrieval	15
21	2.4.1 System Model	15
22	2.4.2 Future Scope	15
23	2.5 Salient Orientation	16
24	2.5.1 System Model	16
25	2.5.2 Conclusion	16
26	Chapter-3 System Development	17
27	3.1 Analysis	17
28	3.1.1 Scope of Research	18
29	3.1.2 System Requirements	18
27	3.2 System Design	19
28	3.3 Mathematical Model	20
29	3.4 Hadoop Setup	21
30	3.5 Approach for task	22
31	3.5.1 Map Task Execution	23
32	3.5.2 Reduce Task Execution	24

33	3.6.a. Single Node	30
34	3.6.b. Multi Node	32
35	3.7 Final System Output	33
36	Chapter-4 Performance Analysis	35
38	4.1 Experiments and Testing	36
39	4.2 Result Analysis	38
40	Chapter-5 Conclusion	39
41	5.1 Conclusions	39
42	5.2 Future Scope	40
42	5.3 Applications Contributions	41
43	5.4 References	42

List of Tables

1. Table 4.1	Cluster Configuration	36
--------------	-----------------------	----

List of Figures

1. Figure1.4	Hadoop Features	5
2. Figure1.4.1	Map-reduce	8
3. Figure 3.2	System Architecture	19
4. Figure 3.6.1	hdfs And daemons	25
5. Figure 3.6.2	yarn	26
6. Figure 3.6.3	Directory	27
7. Figure3.6.4	Inout File	28
8. Figure 3.6.5	Execution(120Images)	29
9. Figure 3.6.6	Output Files	30
10. Figure 3.6.7	Output	31
11. Figure 3.6.8	Master & Slave	32
12. Figure 3.7.1	Input Images	33
13. Figure 3.7.2	Search Image	34
14. Figure 3.7.3	Output Images	34
15. Figure 4.2.1	Single Node	37
16. Figure 4.2.2	Multinode	38

Abstract

As Internet is growing, production of information over it is likewise expanding exponentially. Maximum a portion of this information contains pictures. Today extensive measure of collective picture information is delivered through computerized cameras, cell phones, and software like photoshops etc. These pictures are private to specific client. This advanced picture information ought to be secured with the end goal that it should not be accessed by others aside from the client.

There are a few areas which utilizes content based image retrieval applications. These applications ought to be sufficiently quick to complete client functionalities efficiently and secure with a specific end goal to ensure client information. If by chance that application is to be conveyed on cloud, then it ought to be backed by cloud structure. One innovation which has support from cloud and does distributed parallel computing is "Map Reduce".

The advancement of web causes the touchy development of pictures. It is impractical to handle that measure of pictures utilizing the ordinary strategy. So there is a requirement for new strategy that can deal with an immense measure of information and gives the more precise outcomes to the client. Hence, we are presenting another technique for recovering picture called as "Picture Retrieval Using Hadoop Map diminish". The primary goal of this is to handle an enormous measure of information with the guideline of parallel preparing.

Chapter 1

INTRODUCTION

1.1 Introduction

Today a large portion of information over Internet lies into database servers which are associated with Inter-net. In the event that a client makes a question to database, he/she ought to get required questioned data. There are numerous inquisitive clients who makes question to database deliberately or unintentionally to get private data of different clients. Parcel of research is given to shield databases from these sorts of inquisitive clients. Private data recovery is a strategy or an exploration range where security as for this sort of interruption has been considered.

A private data recovery (PIR) convention permits a client to recover a thing from a server possessing a database without uncovering which thing he/she is recovering.

Over period distinctive changes in PIR has been proposed which makes PIR quicker, more efficient and more secure. Execution like negligent exchange is likewise actualized in PIR. This limits the client from getting data about other database things . PIR additionally keeps up protection of the inquiries from database.

1.2 Image Retrieval

A image retrieval framework is intended to pursue, look and recover pictures from huge database of computerized pictures. Most customary and basic strategies for image retrieval

use technique for including metadata, for example, inscribing, watchwords, or portrayals to the pictures with the goal that recovery can be performed over the comment words. Manual picture comment is tedious, arduous and costly; to address this, there has been a lot of research done on programmed picture explanation . Explained pictures are sought in light of Images' metadata.

Picture Meta look: Search of pictures in light of related metadata, for example, watchwords, content, and so on.

Content-based picture recovery (CBIR): The utilization of PC vision to the picture recovery. CBIR goes for maintaining a strategic distance from the utilization of literary portrayals and rather re-trieves pictures in view of likenesses in their substance (shape, Textures, shapes and so forth.) to a client provided question picture or client specified picture highlights.

1.2.1 Content Based Image Retrieval

Since 1970's, Image Retrieval has been a dynamic research zone. First and foremost, research was concentrated to content based pursuit as it were. It was new system around then, which utilized names of picture les as a pursuit criteria. In this structure, pictures were should be commented on rst and after that from database administration framework, pictures were recovered . This system was having two confinements rstly size of picture information that can t into database and besides broad manual explanation work. There was have to do investigate chip away at recovery which would not have confinement of manual passage and enormous size information. It prompt to Content Based Image Retrieval strategy.

Content-based Image Retrieval (CBIR), otherwise called inquiry by picture content (QBIC) . Content based picture recovery (CBIR) is a strategy in which substance of a picture is utilized as coordinating criteria rather than picture's metadata, for example, catchphrases, labels, or any name connected with picture. This gives most inexact match when contrasted with content based image retrieval.

The term 'content' in this setting may allude to hues, shapes, surfaces, or whatever other data that can be gotten from the picture itself. Image substance can be ordered into visual and semantic substance. Visual substance can be extremely broad or space specific. General visual substance incorporate shading, surface, shape, spatial relationship, and so forth. Area specific visual substance, similar to human countenances, is application subordinate and may include space learning. Semantic substance is gotten either by literary comment or by complex induction methods in light of visual substance .

COLOR: Image recovery in light of shading really implies recovery on shading descriptors. Most normally utilized shading descriptors are the shading histogram, shading intelligence vector, shading correlogram, and shading minutes . A shading histogram identifies the extent of pixels inside a picture holding specific values which can be utilized to find likeness between two pictures by utilizing similitude separate measures .It tries to distinguish shading extent by locale and is autonomous of picture size, configuration or introduction . Shading minutes contains estimation of the first arrange (mean), the second (change) and the third request (skewness) of a picture .

TEXTURE : Surface of a picture is really visual examples that a picture has and how they are spatially defined. Surfaces are spoken to by texels which are then set into various sets, contingent upon what number of surfaces are distinguished in the picture. These sets define the surface, as well as where in the picture the surface is found. Statistical technique, for example, co-event Matrices can be utilized for quantitative measure of the course of action of powers in an area.

SHAPE : Shape in picture does not mean state of a picture but rather it implies that state of a specific area or a question. Division and edge identification are noticeable techniques that can be utilized as a part of shape discovery .

There are a few segments that are ascertained from picture substance, for example, color intensity, entropy, mean of picture and so forth which are valuable in making of feature vector. Feature vector of each picture is ascertained and is put away in database. At the point when client needs to recover set of pictures, he inquiries framework through a picture. Feature vector of required picture is compared with vectors of required picture and pictures with vector similar to vector of questioned picture are recovered. Client can choose required picture from set of displayed pictures.

1.3 Objectives

The objective of this project is to introduce a new method for retrieving images from a large database of images with high efficiency and in fashion so that it gives the desired output as per the given requirements. This will be implemented using the technique of Hadoop Map-reduce Algorithm. This new method will be able to handle a huge amount of data and provides the more accurate results to the user. The main objective of this is to handle a huge amount of data in form of images with the principle of parallel processing. This method be called as “Content Based Image Retrieval Using Hadoop Map reduce”.

1.4 Methodology

Hadoop

The Apache Hadoop programming library is a structure that permits the distributed handling of expansive informational indexes crosswise over groups of PCs utilizing Easy programming models. It is intended to scale up from single servers to a huge number of machines, each offering local calculation and capacity. Instead of depending on equipment to convey high-accessibility, the library itself is intended to recognize and handle failures at the application layer, so conveying a very accessible administration on top of a bunch of PCs, each of which might be inclined to failures.

A few modules of Hadoop are recorded underneath :

1. Hadoop Common: These regular utilities are java libraries which will be used to start hadoop and also support the other Hadoop modules.
2. Hadoop Distributed File System (HDFS): A dispersed file framework providing maximum-throughput path to data. Files are divided into blocks and stored at nodes.
3. Hadoop YARN : A system for job scheduling & managing the cluster.
4. Hadoop MapReduce: A framework used to do parallel handling of data and information using key value pairs.
5. HBase : An adaptable, widespread database that backups organized information of data stored in forms of tables.

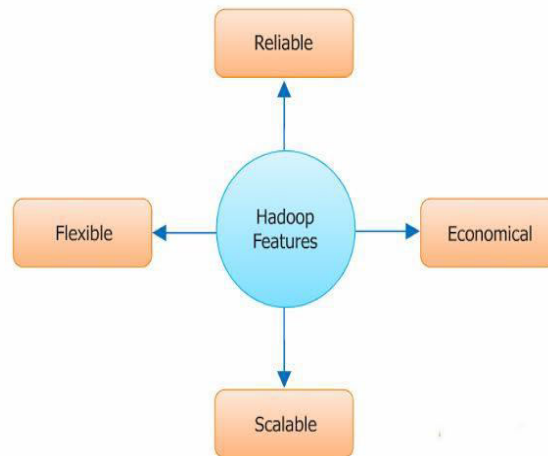


Figure 1.4

Data Types :

Structured Data

Before getting into unstructured information, you need a comprehension for its organized partner. Organized information (as clarified concisely in Big Data Republic's video) is data, more often than not message records, showed in titled sections and lines which can without much of a stretch be requested and handled by information mining instruments. This could be envisioned as an impeccably sorted out file organizer where everything is distinguished, marked and simple to get to. Most associations are probably going to be acquainted with this type of information and right now utilizing it adequately

Unstructured Data

Unstructured information, normally double information that is exclusive, is what has no identifiable inner structure. It can be pictured as a level 5 hoarder's lounge room; it's a monstrous sloppy aggregate of different articles that are useless until distinguished and put away in a composed manner. When this association procedure has occurred (using specific programming), the things can then be looked through and sorted (to a degree) for getting bits of knowledge. While information mining instruments won't not be prepared to parse data in email messages (however sorted out it might be), you may have justifiable reason motivation to gather and order information from this source. This represents the significance and conceivable broadness of unstructured information.

The Big Data industry is expanding, however, the issue of unstructured information going unused has been recognized by most associations. Even better, innovations and newer developments are being created in response.

Darin Stewart of InformationWeek said in a current article in regards to enormous information, "The period of data overburden is gradually attracting to a nearby. Endeavors are at long last getting settled with overseeing enormous measures of information, substance and data. The pace of data creation keeps on quickening, yet the capacity of framework and data administration to keep pace is going inside sight. Enormous Data is currently viewed as a gift instead of a revile.

Generally, organized data alludes to data with a high level of association, to such an extent that incorporation in a social database is consistent and promptly searchable by basic, clear web crawler calculations or other hunt operations; though unstructured information is basically the inverse. The absence of structure makes accumulation a period and vitality expending undertaking. It is helpful to an organization over all business strata to discover a system of information examination to decrease the costs unstructured information adds to the association.

Through our "information wrangling" strategies, Trifacta Wrangler empowers both organized and unstructured information arrangement, examination, and perception. Trifacta's natural interface engages everybody—even the most non-specialized of clients—to intuitively investigate and get ready basic and complex information sources keeping in mind the end goal to execute information examination.

Experts can without much of a stretch consolidate their current likely organized information with unstructured information, for example, mapping online networking with client and deals computerization information, for instance. Regardless of the many-sided quality and change, Trifacta Wrangler grants clients to influence the information they require at an early stage with a specific end goal to produce the correct yields for better basic leadership.

1.4.1 MapReduce

MapReduce is a parallel processing procedure and a program show for widespread registering in view of java. The MapReduce calculation contains two imperative assignments, in particular Map and Reduce. It outlines an arrangement of information and converts it into another arrangement of information, where singular components are separated into tuples (key/value sets). Furthermore, reducer task, which takes the output from a map as an input and consolidates those information tuples into a little arrangement of tuples. As the arrangement of the name MapReduce infers, the lesser errand is constantly performed after the map work.

The significant preferred standpoint of MapReduce is that it is anything but difficult to scale information preparing over numerous processing nodes. Under the MapReduce display, the information handling primitives are called mappers and reducers. Breaking down an information preparing application into mappers and reducers is once in a while nontrivial. Yet, once we compose an application in the MapReduce frame, scaling the application to keep running more than hundreds, thousands, or even a huge number of machines in a bunch is only a setup change. This straightforward versatility is the thing that has pulled in numerous developers to utilize the MapReduce display.

The MapReduce Algorithm

For the most part MapReduce Paradigm depends on sending the PC to where the information lives!

MapReduce program executes in three phases, in particular map stage, shuffle stage and reduce stage.

- Map organize : The guide or mapper's occupation is to handle the information. By and large the input information is as document or index and is put away in the Hadoop file framework (HDFS). The info document is passed to the mapper work line by line. The mapper forms the information and makes a few little pieces of information.
- Reduce organize : This stage is the blend of the Shuffle arrange and the Reduce organize. The Reducer's occupation is to prepare the information that originates from the mapper. Subsequent to preparing, it creates another arrangement of yield, which will be put away in the HDFS.

- During a MapReduce work, Hadoop sends the Map and Reduce assignments to the proper servers in the group.
- The structure deals with every one of the subtle elements of information passing, for example, issuing assignments, checking errand , and duplicating information around the group between the hubs.
- Most of the processing happens on nodes with information on nearby plates that lessens the system activity.
- After completion of the given undertakings, the cluster gathers and reduces the information to shape a proper outcome, and sends it back to the Hadoop server.

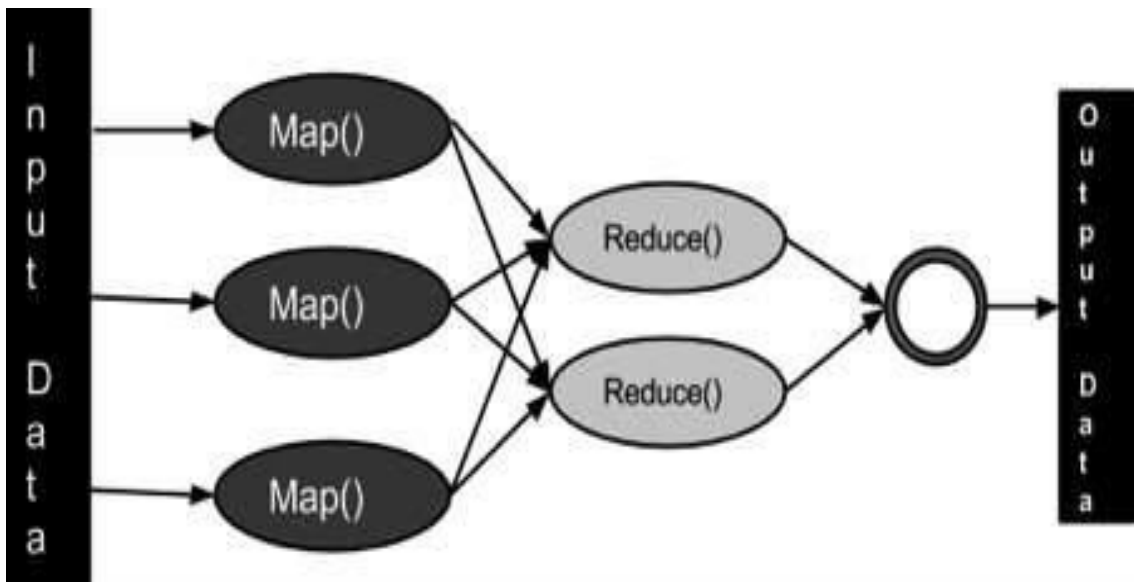


Figure 1.4.1

Example.

The map function emits each word plus an associated count of occurrences (just 1 in this simple example). The reduce function sums together all counts emitted for a particular word.

```
map_Word(String key _le, String_value line):
```

```
// key le: text le name, value line: text le contents for each  
word m in value line: EmitIntermediateval(m, \1");
```

```
reduce_Wordx(String key _le, Iterator value line):
```

```
// key le: a word from text le, value line: a-list of counts of each word int total  
count = 0;  
for each word t in value line: total  
count += ParseInt(t);  
Emit nalval(AsString(total count));
```

Chapter 2

Literature Survey

2.1 PIRMAP

Private Information Retrieval (PIR) takes into account recovery of bits from a database in a way that conceals a client's get to design from the server. This paper presents PIRMAP, a practical, profoundly efficient convention for PIR in MapReduce, a generally upheld distributed computing API. PIRMAP concentrates particularly on the recovery of huge files from the cloud, where it accomplishes ideal correspondence many-sided quality ($O(1)$ for recovery of a 1 bit file) with inquiry times significantly speedier than past plans.

2.1.1 PIR

At the point when client interfaces with a cloud, there is arrangement of cooperations between the client and the server. Client associates with cloud to recover his private data which is put away. The data can be as differentless.

Presently client needs to recover x number of files out of his n files of his decision furthermore server ought not find out about his selection of files. Presently client of cloud needs to give such inquiry to cloud to such an extent that client will get his required number of files from cloud and this exchange ought to be secured.

2.1.2 MapReduce in PIRMAP

This data recovery system utilizes delineate to parallelize the procedure which will decrease the computational time to recover the l from cloud and will make it simple to utilize.

The first stage is known as the "Map" stage. MapReduce will naturally part input calculations, similarly among accessible hubs in the cloud server farm, and every hub will then run a capacity called outline their individual pieces (called InputSplits). Note that the part really happens when the information is transferred into the cloud. This implies each "mapper" hub will have neighborhood access to its InputSplit when calculation is begun and you keep away from a long replicating and conveying period. The guide work runs a client defined calculation on each InputSplit and yields (discharges) various key-esteem combines that go into the following stage.

The second stage, "Reduce", takes as info the greater part of the key-esteem sets radiated by the guide pers and sends them "reducer" hubs in the server farm. Specifically, every reducer hub gets a solitary key, alongside the grouping of qualities discharged by the mappers which share that key. The reducers then take every set and consolidate it somehow, discharging a solitary esteem for every key.

2.1.3 Working of PIRMAP

PIRMAP is an expansion of the PIR convention by Kushilevitz and Ostrovsky, focusing on the recovery of substantial l s in a parallelization-collection calculation system, for example, MapReduce. We will begin by giving an outline of PIRMAP which can be utilized with any additively homomorphic encryption plot.

Transfer: In the accompanying, we accept that the cloud client has as of now transferred its l s into the cloud utilizing the interface gave to them by the cloud supplier.

Question: with regards to standard PIR documentation, our information set holds n l s, each of which is l bits long. There is likewise an extra parameter k which is the piece size of the picked figure. For simplicity of presentation, we will consider the situation where all l s are a similar length, yet PIRMAP can without much of a stretch be reached out to suit variable length l s by cushioning or prepending every l with a couple of bytes that determine its length.

2.1.4 Future Scope/Issues

This usage is specifically for les which contains printed information, it doesn't sup-port for sight and sound information, for example, picture, video or sound les.

2.2 Map/Reduce in CBIR Application

Toward the begin, picture recovery is for the most part reliant on the content based recovery of a picture. This innovation is broadly utilized, the ebb and flow standard web crawlers Google, Baidu, Yahoo, and so forth basically utilized this strategy to inquiry picture. In this strategy, name of picture le was utilized to look at and recover the mage from database.

Be that as it may, it has downsides, analysts frequently need to physically stamp with content for all pictures, and this check content can't equitably and precisely portray the visual data of the pictures.

After the 1990s, there has been Content-Based Image Retrieval, the accompanying unified call CBIR, and different from TBIR, it can separate visual components from picture naturally, and afterward recovery picture by their visual elements. Since CBIR instinctive, e cient, and can be generally utilized as a part of data recovery, medicinal finding, trademarks and licensed innovation assurance, wrongdoing anticipation and different zones, it has a high connected esteem.

2.2.1 System Model

Selecting an Algorithm

The procedure utilized as a part of this paper utilizes Color component based picture recovery. This chiefly contains the accompanying calculations:

Shading histogram, tint histogram, shading minutes, shading entropy and so forth.

Considering the shading histogram calculation is broadly utilized, its element extraction and comparability coordinating are more less demanding, this technique will be utilized as our objective shading calculation.

Shading Histogram

Shading Histogram of a picture gives broad data about its structure. It gives number of pixels, its hues in RGB design and so on. So it is anything but difficult to figure mean, entropy, middle of a picture which can be utilized as components of a picture in the event of highlight extraction. This element can be utilized to contrast with questioned picture's element and with recover every single comparative picture from database.

Include Calculation and Similarity Matching

Include vector of each transferred picture is put away in database. This element vector is match with highlight vector of an info picture. Both component vectors are utilized to ascertain similarity coefficient. This is called as Pearson connection coefficient. Outline is utilized as a part of similitude coordinating stage just to give quick outcomes.

Include vector of info picture is ascertained in the question and is coordinated with the pictures from picture library and pictures for which pictures are coordinating are recovered.

2.2.2 Future scope/Issues

map reduce method is utilized at comparability coordinating phase of framework. map reduce technique can be efficiently utilized as a part of highlight extraction handle by part picture into parts at guide stage and afterward joining it at lessen level.

2.3 An Oblivious Image Retrieval Protocol

Today numerous site suppliers have huge gathering of pictures. For e.g. google.com, confront book.com have substantial database of pictures where every client transfers his private information to the server farm of site. This information ought to be shielded from outer or inward risk. The risk might be of taking of picture information, modification of the information, erasing private information of an individual, or abusing it for specific reason and so forth.

As the span of this information is expanding step by step, organizations find it difficult to oversee and secure this sort of huge information. Their server farm servers are getting over-burden because of gigantic information. To lessen strain from their capacity servers they are searching for another choice. This choice is of utilizing outer stockpiling servers.

These outside capacity servers are kept up and overseen by different organizations. i.e. information is presently outsourced by the organizations to different organizations to lessen overhead of their interior servers. For this situation it turns out to be imperative to ensure and secure the information of a client over outer outsourced database servers. Unaware Image Retrieval Protocol comes into picture for outsourced picture databases. It is a strategy to safely inquire picture database for required picture information and recover coordinated pictures from database. This recovered information ought to likewise safely get exchanged to the client.

2.3.1 System Model

Framework accepts that all component vectors of the considerable number of pictures are as of now in database. This element information is put away in encoded form into the database. All inquiry operations are done on scrambled information. This for the most part contains two sections one is security saving questioning system and unmindful exchange of the decoding keys.

Security safeguarding convention of paper recommends taking after. Initially, to inquire picture set, client produces an encryption of the question highlight vector set utilizing a homomorphic open key encryption strategy and sends it to the database server. The question highlight vector is contorted by the client with a steady irregular vector to keep any factual deduction by the database server. Second, the database server utilizes the scrambled inquiry include vector and plays out a homomorphic subtraction with an irregular element vector. The server plays out a subtraction of the database picture highlights with a similar arbitrary component vector. Before sending the outcomes back to the client, the database server plays out a change of the subtracted include vectors so that the client is not ready to take in the relative ordering structure of the database pictures. The server incorporates pseudo-identifiers

for the permuted pictures to permit the client to distinguish his decisions. Third, upon getting the server reaction, the client expels the contorting irregular consistent from the server reaction and ascertains the Euclidean standard of the numerical difference the question picture highlight vector and the database picture include vectors

2.3.2 Future Scope/Issues

The oblivious image retrieval technique can be implemented using map reduce technique, map reduce can be used at content based image retrieval stage.

2.4 Distributed Image Retrieval System Based on MapReduce.

A Distributed Image Retrieval System(DIRS) is a framework in which pictures are recovered in a substance based manner, and the recovery among enormous picture information stockpiling is conveyed parallelly by using MapReduce widespread figuring model.

2.4.1 System Model

In this framework, pictures are transferred to HDFS in one guide diminish handle. This guide lessen prepare extricates elements of pictures by utilizing LIRE library and stores pictures and their component vectors to Hbase table.

Pictures are likewise recovered parallelly. In the element coordinating procedure, CBIR framework calculates the similitude between the specimen picture and the objective pictures, and afterward gives back those pictures coordinating the example picture generally nearly.

Before MapReduce employment is begun, the example picture ought to be added to Distributed Cache to empower each guide undertaking to get to it. Picture is perused from Distributed reserve, extricate its visual components, and after that contrast the element vectors and highlight vectors of target pictures from HBase. Every single coordinated picture are recovered back to client.

2.4.2 Future Scope/Issues

The distributed image retrieval technique needs some secure technique to protect data.

2.5 Content based image retrieval using salient orientation histograms

2.5.1 SYSTEM MODEL

Contentbased image retrieval is a vital topic these days, when the measure of computerized picture information is exceedingly expanding. Existing sketch based picture recovery (SBIR) frameworks perform at a diminished level on genuine pictures, where foundation information may bend picture descriptors and recovery comes about. To stay away from this, a pre processing step is acquainted in this paper with recognize closer view and foundation, utilizing incorporated saliency location. To construct the descriptor just on the most important pixels, introduction highlight is separated at notable Modified Harris for Edges and Corner (MHEC) keypoints utilizing an enhanced edge outline, in a Salient Orientation Histogram (SOH). The proposed SBIR framework is additionally expanded with a division venture for question recognition. The technique is tried on the THUR15000 database, containing irregular web pictures. Picture recovery and protest discovery both give promising outcomes contrasted with other best in class techniques.

2.5.2 Conclusion

In this paper, a novel SBIR framework is presented, utilizing a remarkable keypoint based orientation histogram (SOH). The proposed technique initially separates the striking picture locale in view of surface uniqueness, trailed by a Modified Harris for Edges and Corners (MHEC) intrigue point discovery. Along these lines the most important pixels of the picture are chosen to manufacture an introduction histogram on an enhanced edge outline, of applying Canny edge delineate prior SBIR frameworks. The edge guide is additionally adjusted for division. Generally, the proposed descriptor accomplishes superior on the THUR15000 dataset, and it likewise gives a proficient protest recognition strategy. Future work will research the enhanced joining of saliency in SBIR frameworks.

Chapter 3

System Development

3.1 Analysis

With the explosive development of computerized media information, there is a colossal interest for new instruments and frameworks. It ought to empower normal client to pursuit, get to, process, oversee, creator and share these computerized media substance more efficiently and all the more successfully. This prompts to change individuals' enthusiasm from work to work with entrainment.

Illustrations:

Business Phones to Smartphones,

Watching motion pictures/T.V.,

Listening music,

Web-based social networking: interface and share

There are diverse sorts of media substance exhibit today.

A sight and sound data framework can store and recover content, 2D dim scale and shading pictures, 1D time arrangement, digitized voice or music, video and there is have to propose an efficient, hearty, secure answer for recover this sort of interactive media data.

We are thinking about picture recovery over sound or video data recovery since today picture recovery framework has parcel of uses. Picture recovery methods have gone under a few changes and are turning out to be increasingly best in class over the timeframe. In early years of picture recovery, pictures were looked on content based where every picture must be clarified with specific content. This method was more troublesome in view of its intricacy.

Today content based picture recovery procedure gives broad pursuit office over database which recovers every single related picture which are comparable in substance with questioned picture.

The CBIR innovation has been utilized as a part of a few applications, for example, fingerprint identification, biodiversity data frameworks, computerized libraries, wrongdoing avoidance, pharmaceutical, recorded research, among others. This CBIR covers wide range of different areas. Today numerous sites like google.com, facebook.com and so forth has parcel of pictures on their servers. Numerous clients get to these sites consistently. They transfer, seek download pictures from them. There is a need of quick and secured method to transfer, seek and recover these pictures on client request.

3.1.1 Scope of Research

Scope of this research is not limited to one particular domain. It spans over multiple domains. Domains in which extensive research can be done are Private Information Retrieval, Content Based Image Retrieval and Oblivious transfer i.e. security. Research papers in this area suggests content based image retrieval technique over cloud computing as well as secured transfer techniques for the images but not a single paper talks about fast (over cloud computing) as well as secured retrieval of images form servers using Private Information Retrieval Technique.

This system can be evolved in many ways. By changing CBIR algorithm, by using different encryption techniques, by using different storage methods over HDFS, this system can be moved to new more efficient and secure level

3.1.2 System Requirements

HARDWARE:

Least one framework with Intel's or AMD's processor and 2Gb of RAM. Most recent innovation will give better outcomes. i.e. Group of 5 Clusters having Intel's center arrangement processor(i3,i5,i7) of high recurrence with 4Gb RAM will perform superior to anything 5 node cluster having more seasoned processor variant taking a shot at less recurrence and less RAM. Group ought to have nodes having same setup. In the event that it has nodes hving distinctive RAM limit, processor recurrence then cluster won't perform up to capacity.

SOFTWARE:

1. Operating System: Any open source Linux working framework.
2. Coding Language: Hadoop outline innovation/programming Java dialect (JDK)
3. IDE: Eclipse environment

3.2 System Design

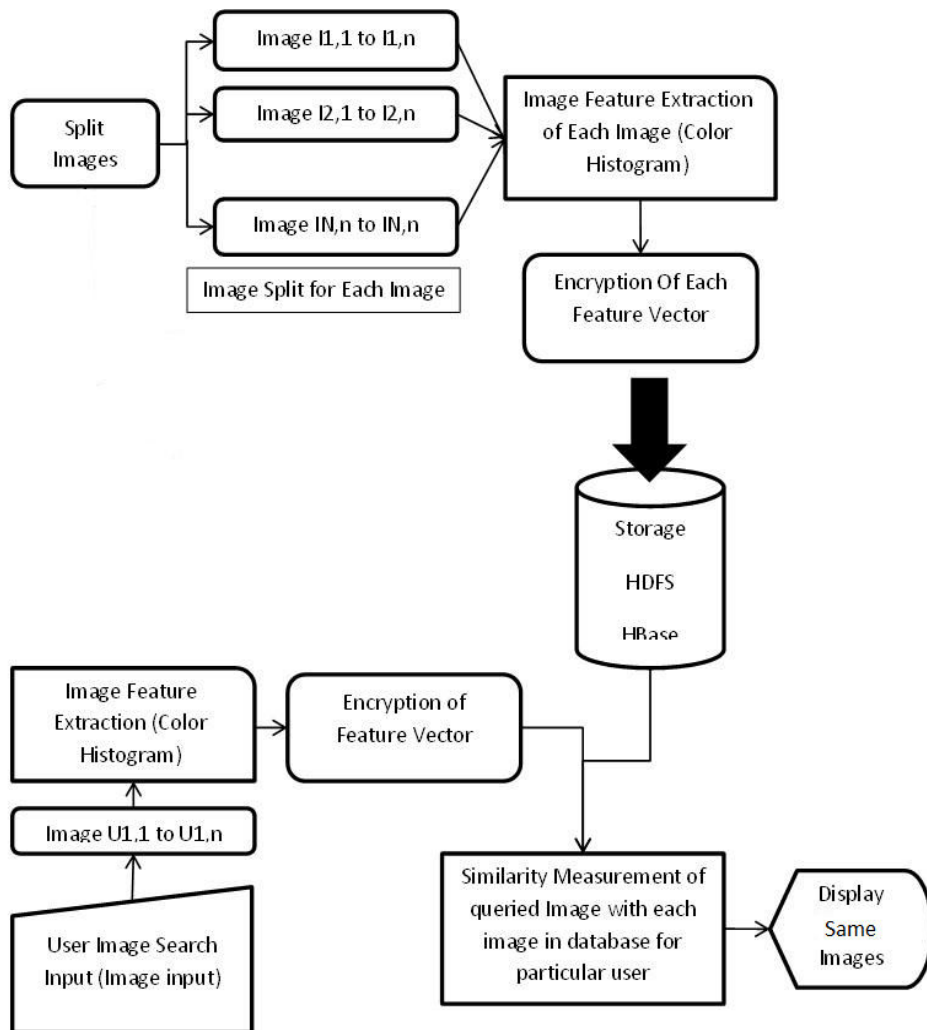


Figure 3.2.1 System Architecture

Highlight is only the substance of picture that depicts the picture. The elements are extricated from the picture as feature vector. The principal stage incorporates extraction of components from every one of the pictures in the database and stores it as feature vector in HDFS. The elements removed in the proposed strategy are shading highlight, surface component and shape include. These elements are separated parallelly by utilizing hadoop MapReduce. Keeping in mind the end goal to recover comparative pictures from the database we should have some likeness estimation system. The second stage is similitude coordinating. This stage incorporates correlation of components of question picture with the element vector in the database. Those pictures with the less separation are recovered.

3.3 MATHEMATICAL MODEL

Similarity Measurement

A similarity measurement is constantly chosen to find how comparative the two vectors are. The issue can be changed over to figuring the inconsistency between two vectors $x, y \in \mathbb{R}^d$. There are three separation estimations: Euclidean, Mahalanobis, and Chord Distances which are audited as takes after.[11]

3.3.1 Euclidean Distance

The Euclidean distance between $p, q \in \mathbb{R}^2$ is computed by [11]

$$\begin{aligned} d(p, q) = d(q, p) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \end{aligned}$$

3.3.2 Mahalanobis Distance

The Mahalanobis distance between two vectors x and y with respect to the training patterns x_i is computed by [11]

$$d(x; y)_M = \sqrt{(x - y)^t S^{-1} (x - y)}$$

where mean vector u and the sample covariance matrix S from the sample $x_i, j = 1, \dots, n$ are computed by [11]

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - u)(x_i - u)^t \text{ with } u = \frac{1}{n} \sum_{i=1}^n x_i$$

3.3.3 Chord Distance

The chord distance between two vectors x and y is to measure the distance between the projected vectors of x and y onto the unit sphere, which can be computed by [11]

$$d(x; y) = \sqrt{r^2 + s^2 - 2rs \cos \theta}, \text{ where } r = \|x\|_2 \text{ and } s = \|y\|_2$$

3.4 Hadoop Setup

Pre-installation Setup

Before installing Hadoop into the Linux environment, we need to set up Linux using ssh (Secure Shell).

Creating a User

At the beginning, it is recommended to create a separate user for Hadoop to isolate Hadoop file system from Unix file system.

SSH Setup and Key Generation

SSH setup is required to do different operations on a cluster such as starting, stopping, distributed daemon shell operations.

Installing Java

Java is the main prerequisite for Hadoop.

Downloading Hadoop

Download and extract Hadoop 2.4.1 from Apache software foundation using the following commands.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.4.1/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.1.tar.gz
# mv hadoop-2.4.1/* to hadoop/
# exit
```

3.5 Approach For Task

Map Reduce Execution will take place on a Single Node sequentially i.e one by one each part will process. First feature vector of all images are calculated and stored in secure encrypted form than the input image feature vector is calculated and stored. After that the feature vector of queried images and other images is compared one by one. After that the same images will be the output of the program . Now , time complexity of this will be higher because it doesn't utilize the map reduce algorithm since no parallel processing takes place , only the sequential processing of program gives poor time complexity for larger data.

So, we need to switch to multinodes i.e creating a cluster of computers using virtual machines so as to utilize the time complexity for large data of images using the parallel processing technique of map reduce . In this technique a master node is created along with several slaves node . The following process takes place as using the multimode :

Part 1 : Map/Reduce procedure chips away at its dispersed framework called as Hadoop Distributed File System(HDFS) which is completely worked for parallel processing. All pictures are put away into file system. Data about all pictures of each picture are put away into HBase table. HBase takes a shot at top of HDFS. Table in HBase stores way of picture and different components of that picture. [17], [13], [16].

Part 2 : These pictures are given to the image processing part. In this stage, Color Histogram of each picture is determined. This histogram will give distinctive element values, for example, entropy, p intensity, mean, median and so on. These qualities are utilized for the closeness computation of pictures i.e. remove between quired picture and pictures from HDFS.[11] Favored point of view of this will come into picture at the time of recovery. At the time of recovery when customer will request to the system then mixed component vector of quired picture will be used. This element vector which is in encoded casing will be used against total feature vector of all split picture of particular picture.

Part 3 : Here client will give input of one picture for which he needs to discover same pictures. At the point when client needs to hunt pictures in light of specific picture, he transfers it to framework. Feature vector of each picture is computed from color histogram. All element vectors of split pictures are scrambled with general society key of client. These all encoded include vectors of questioned picture are sent to server. This system of searching, comparing and recovering pictures is likewise one Map reduce process. In this procedure, when client transfers picture, encoded highlight vectors of questioned picture i.e. include vectors of all split picture records alongside all picture feature vectors in HBase table goes to Map prepare.

Now, after successful implementation of the multinode cluster , we will then analysis the results of the single node VS several other multinodes on different kind of system with different System Configurations and using different amount of datasets.

3.5.1 Map Task Execution

Every map task is doled out a part of the info record called a split. Of course, a split contains a solitary HDFS block (64MB of course), so the aggregate number of file blocks decides the quantity of guide undertakings.

The execution of a map task is partitioned into two stages.

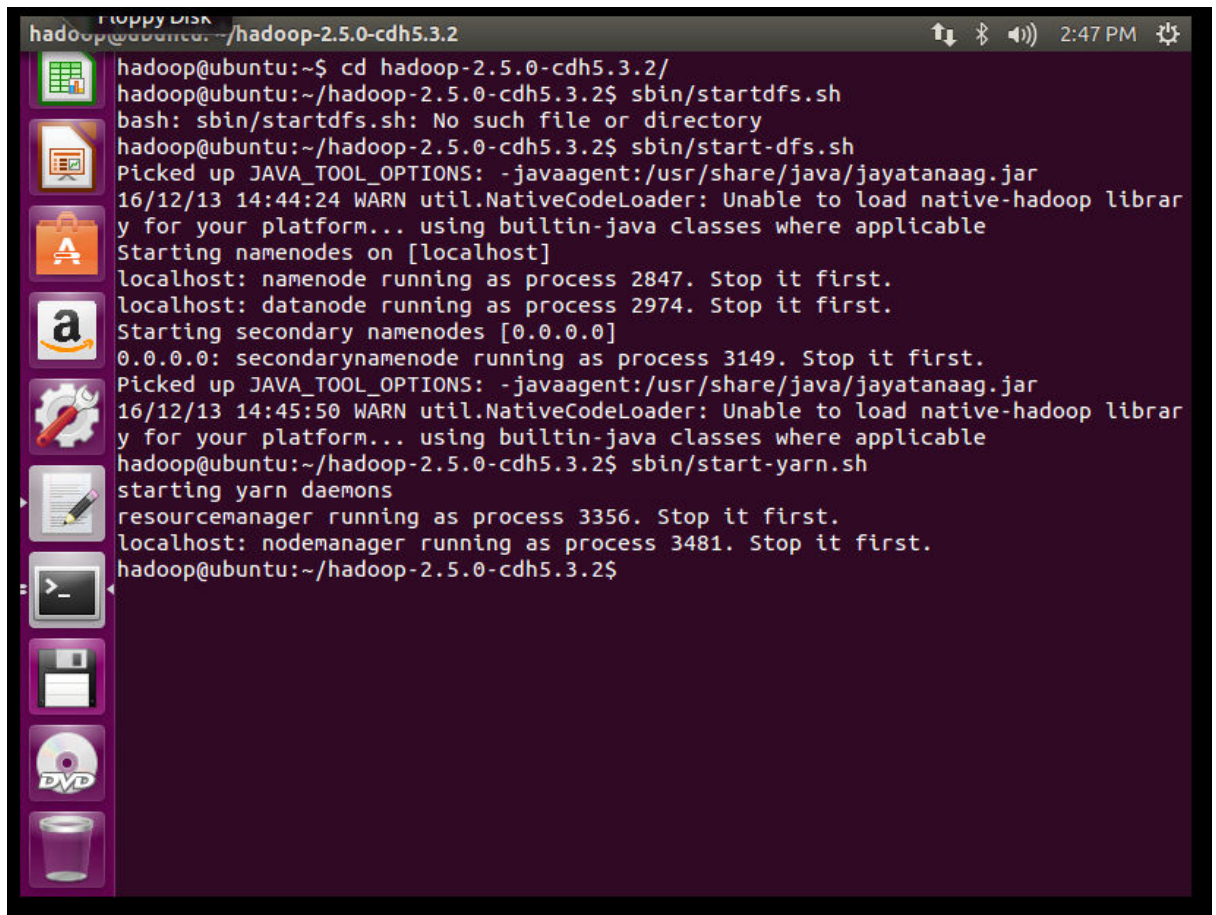
1. The map task peruses the undertaking's part from HDFS, parses it into records (key/esteem matches), and applies the map function to every record.
2. After the map function has been connected to every information record, the commit stage enrolls the last yield with the TaskTracker, which then illuminates the JobTracker that the assignment has got done with executing.
3. The third contention to the map method determines an OutputCollector occasion, which collects the yield records delivered by the map function. The yield of the map step is devoured by the reduce step, so the OutputCollector stores delineate in a configuration that is simple for reduce task to expend. Middle of the road keys are appointed to reducers by applying an apportioning capacity, so the OutputCollector applies that capacity to every key created by the guide capacity, and stores every record and segment number in an in-memory cradle. The OutputCollector spills this support to plate when it achieves limit. A spill of the in-memory cushion includes first sorting the records in the cradle by parcel number and afterward by key. The support substance is composed to the nearby document framework as a list record and an information record. The list record focuses to the balance of every parcel in the information document. The information document contains just the records, which are sorted by the key inside every segment section. Amid the confer stage, the last yield of the guide undertaking is created by blending all the spill records delivered by this assignment into a solitary combine of information and file documents. These records are enlisted with the TaskTracker before the errand finishes. The TaskTracker will read these documents when adjusting demands from lessen undertakings.

3.5.2 Reduce Task Execution

The execution of a reduce task is divided into three phases.

1. The shuffle stage gets the reduce task input information. Each reduce assignment is doled out a parcel of the key range delivered by the map step, so the reduce task must bring the substance of this segment from each map assignment's output.
2. The sort phase groups records with the same key together.
3. The reduce phase applies the client characterized reduce function to each key and appropriate list of data. In the reshuffle stage, a reduce errand gets information from each map output by issuing HTTP solicitations to a configurable number of TaskTrackers on the double (5 of course). The JobTracker transfers the location of each TaskTracker that hosts outline to each TaskTracker that is executing a reduce assignment. Take note of that a reduce task can't get the yield of a map task until the map has got done with executing and submitted its last yield to disk. Subsequent to accepting its potion from all map task, the reduce task enters the sort stage. The map output for each portion is as of now sorted by the diminish key. The reduce undertaking combines these runs to create a solitary run that is sorted by key. The assignment then enters the reduce stage, in which it summons the client characterized reduced work for each particular key in sorted request, passing it the related list of data. The yield of the reduced function is composed to an temporary location on HDFS. After the reduce work has been connected to each key in the reduce task partition, the assignment's HDFS yield document is molecularly renamed from its temparay area to its final location. In this outline, the yield of both map and reduce assignments is composed to disk before it can be devoured. This is especially costly for reduce task, on the grounds that their yield is composed to HDFS. Yield emergence improves adaptation to non-critical failure, since it lessens the measure of express that must be reestablished to consistency after a node failure. The JobTracker basically plans another task to play out an indistinguishable work from the fizzled assignment. Since an errand never sends out any information other than its last answer, no further recovery steps are required.

3.6 Running the Map Reduce (a.Single Node)

A terminal window titled 'hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2' showing the execution of Hadoop commands. The user navigates to the 'sbin' directory and runs 'startdfs.sh' and 'start-yarn.sh'. The output shows the starting of namenodes, datanodes, and secondary namenodes on localhost, along with process IDs and warnings about native Hadoop libraries.

```
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ cd hadoop-2.5.0-cdh5.3.2/
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ sbin/startdfs.sh
bash: sbin/startdfs.sh: No such file or directory
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ sbin/start-dfs.sh
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
16/12/13 14:44:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: namenode running as process 2847. Stop it first.
localhost: datanode running as process 2974. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 3149. Stop it first.
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
16/12/13 14:45:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ sbin/start-yarn.sh
starting yarn daemons
resourcemanager running as process 3356. Stop it first.
localhost: nodemanager running as process 3481. Stop it first.
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$
```

Figure 3.6.1 hdfs & daemons

The following command is used to start dfs. Executing this command will start your Hadoop file system.

```
$ start-dfs.sh
```

Use of these command is to run hdfs daemons and yarn daemons.

It is used to verify the hadoop Installation

The daemons are as follows:

- 1-namenode
- 2-datanode
- 3-secondary namenode
- 4-resource manager
- 5-node manager

```
hadoop@ubuntu: ~/hadoop-2.5.0-cdh5.3.2
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ sbin/startdfs.sh
bash: sbin/startdfs.sh: No such file or directory
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ sbin/start-dfs.sh
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
16/12/13 14:44:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: namenode running as process 2847. Stop it first.
localhost: datanode running as process 2974. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 3149. Stop it first.
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
16/12/13 14:45:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ sbin/start-yarn.sh
starting yarn daemons
resourcemanager running as process 3356. Stop it first.
localhost: nodemanager running as process 3481. Stop it first.
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ jps
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
2974 DataNode
3149 SecondaryNameNode
3356 ResourceManager
2847 NameNode
14752 Jps
3481 NodeManager
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$ hdfs dfs -mkdir /input00001
Picked up JAVA_TOOL_OPTIONS: -javaagent:/usr/share/java/jayatanaag.jar
16/12/13 14:48:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@ubuntu:~/hadoop-2.5.0-cdh5.3.2$
```

Figure 3.6.2 yarn

jps command is used to check all the Hadoop daemons like NameNode, DataNode, ResourceManager, NodeManager etc. which are running on the machine, if they are all working or not.

The use of next command hdfs dfs -mkdir

It takes the path uri's as an argument and creates a directory or multiple directories and in this directory we put our input file i.e input00001.

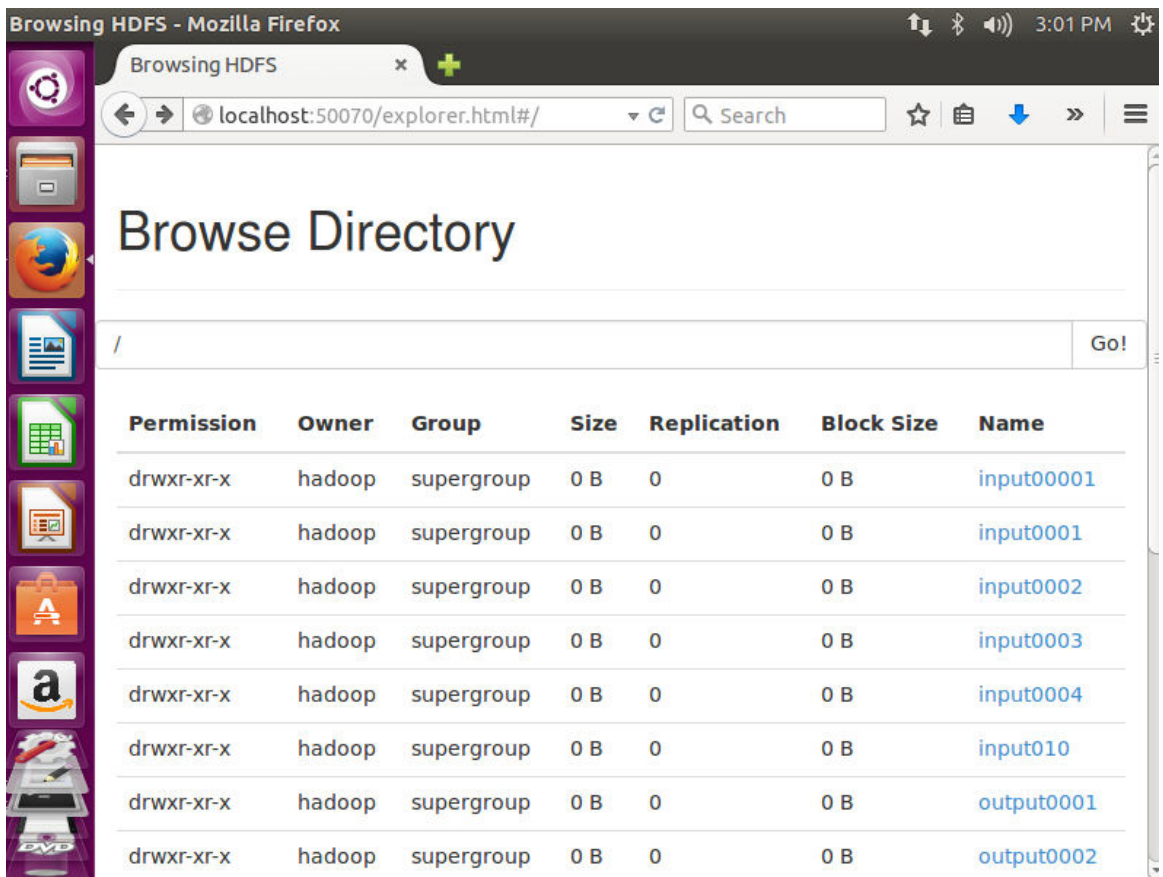


Figure 3.6.3 Directory File

This is the Localhost where hdfs works.

Our Input file input00001 can be seen here.

On the left hand side in the permission column which specifies the permissions and authentication of the files.

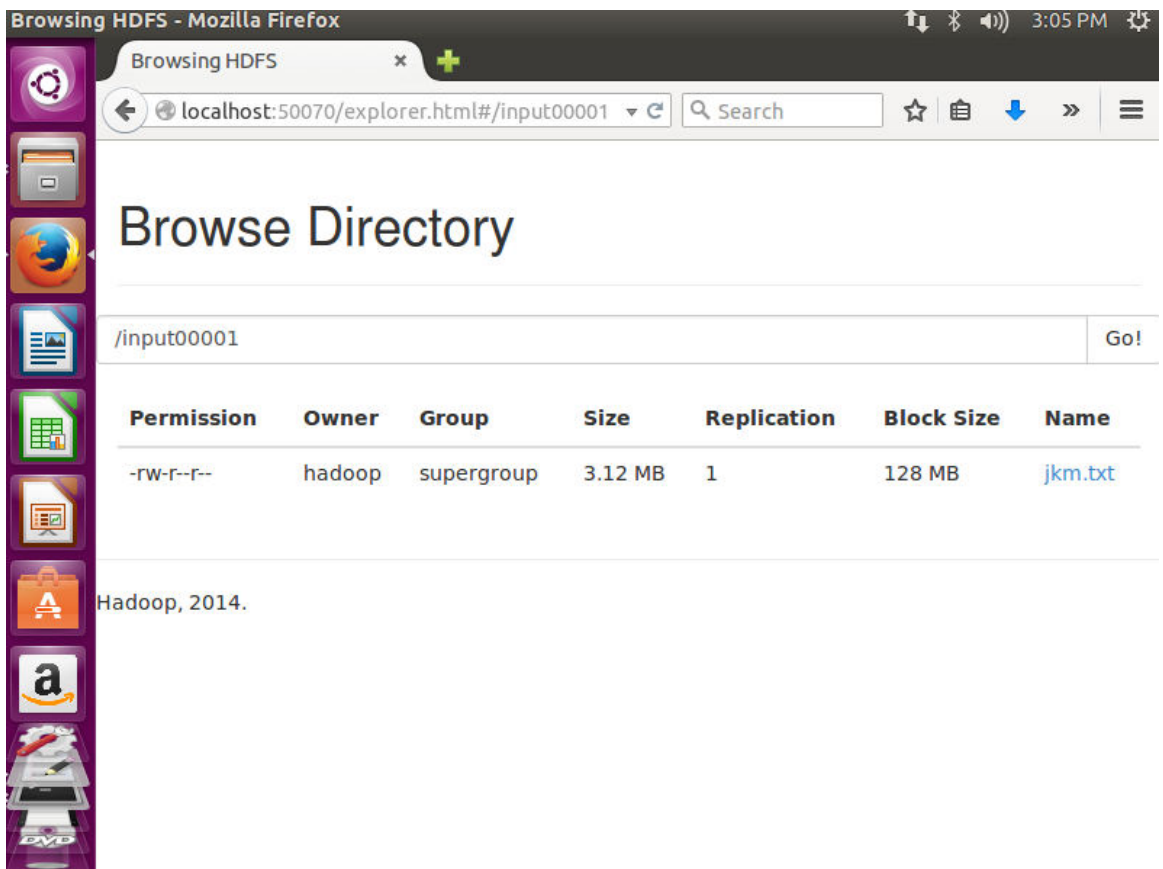


Figure 3.6.4 Input File

In directory input 00001 , we put our input file jkm.txt which contains the dataset that's need to counted using the map reduce Algorithm. The file contains images and has a size of 3.12 mb as specified above.

```
hadoop@ubuntu: ~
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=112902
  Total time spent by all reduces in occupied slots (ms)=40796
  Total time spent by all map tasks (ms)=112902
  Total time spent by all reduce tasks (ms)=40796
  Total vcore-seconds taken by all map tasks=112902
  Total vcore-seconds taken by all reduce tasks=40796
  Total megabyte-seconds taken by all map tasks=115611648
  Total megabyte-seconds taken by all reduce tasks=41775104
Map-Reduce Framework
  Map input records=1
  Map output records=1
  Map output bytes=2146
  Map output materialized bytes=2156
  Input split bytes=93
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
```

Figure 3.6.5 For 120 Images

Now the execution of the map reduce.

The command is used to run by taking the input files from the input directory. Wait for a while until the file is executed. After execution, as shown above, the output will contain the number of input splits, the number of Map tasks, the number of reducer tasks, etc.

Step by Step execution takes place

First the mapper job starts and finishes, only after that the reducer will start doing its work as the input for reducer is the mapper file itself.

We calculate and compare the time spent by mapper and reducer in order to execute the single node for 20, 60, 120...so on.

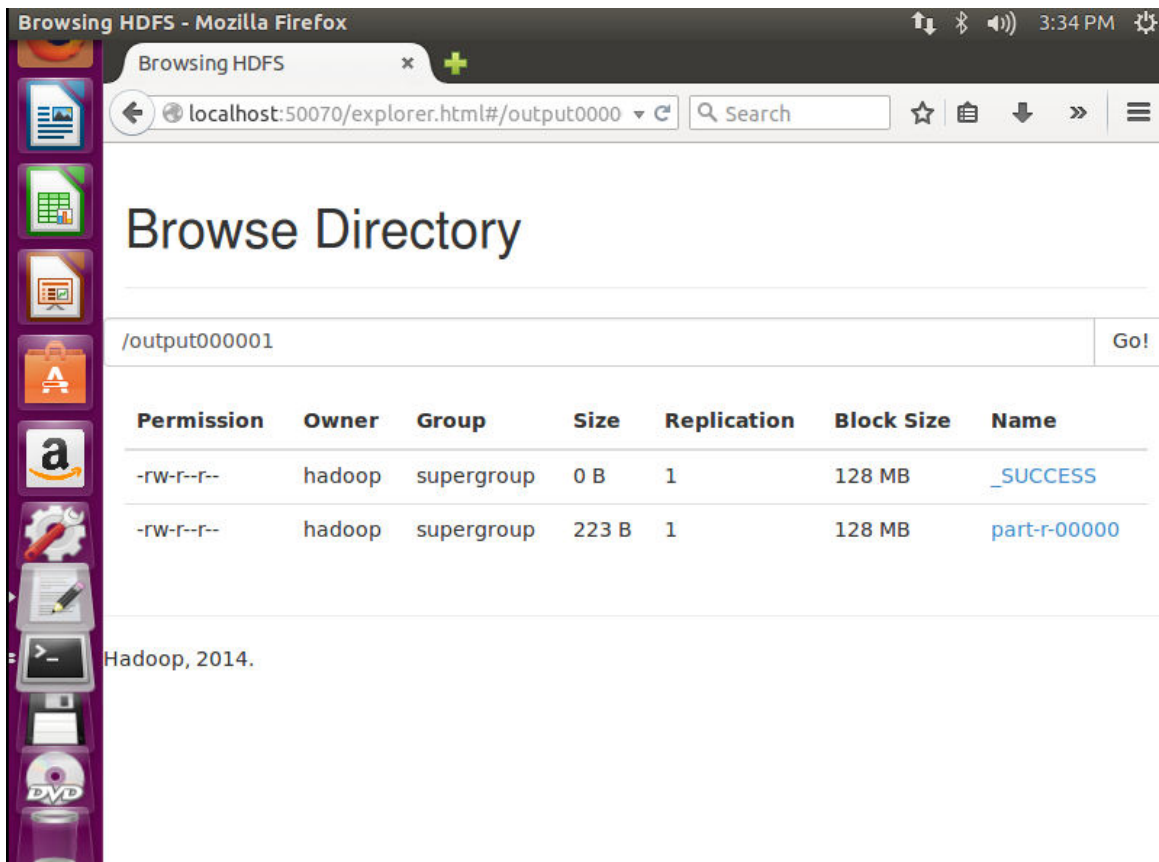


Figure 3.6.6 Output Files

The following command is used to verify the resultant files in the output folder.

```
$HADOOP_HOME/bin/hadoop fs -ls output_dir/
```

The following command is used to see the output in **Part-00000** file. This file is generated by HDFS.

```
$HADOOP_HOME/bin/hadoop fs -cat output_dir/part-00000
```

And this image displays our output file and the size of this file is 223 byte as displayed above. In order to view it, we will need to download the output file first.

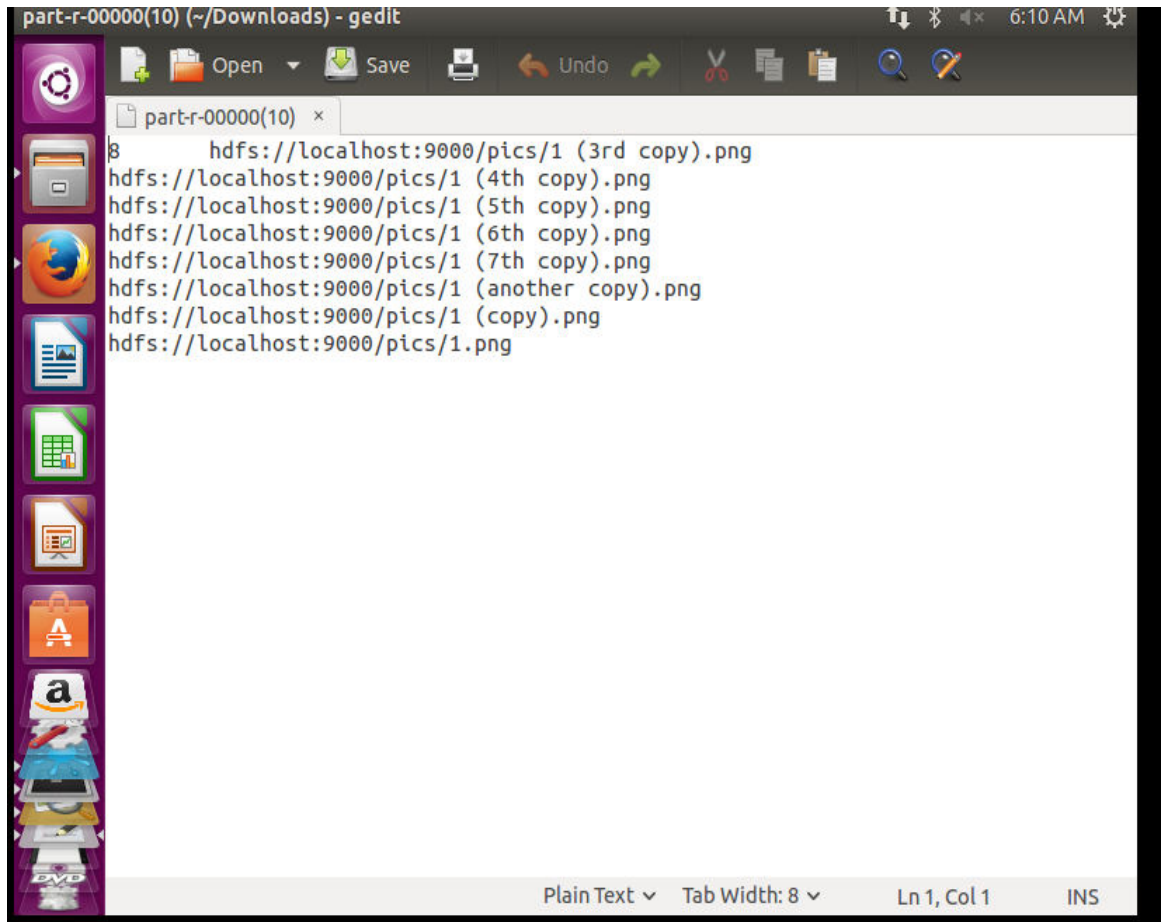
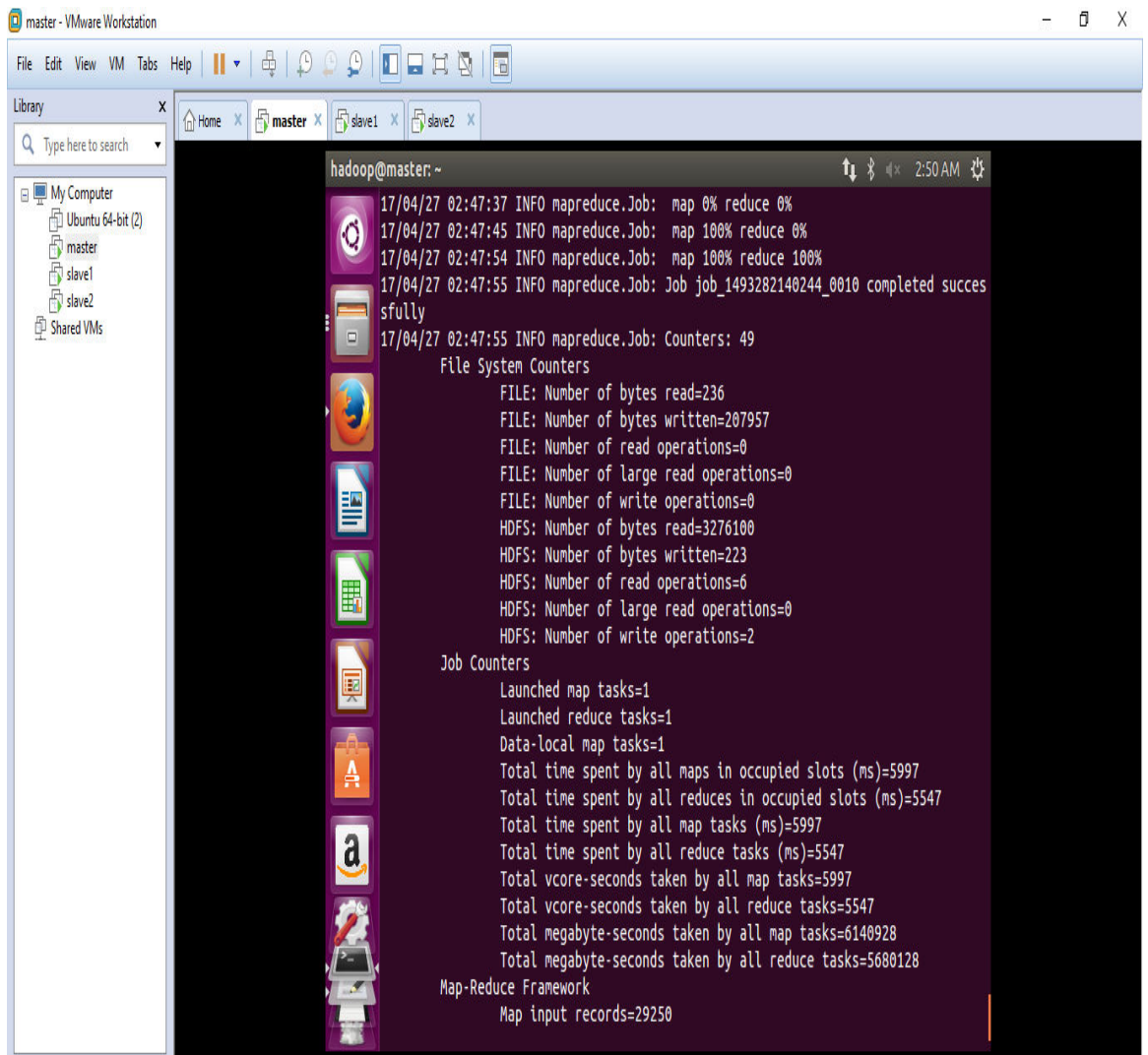


Fig 3.6.7 Output

This is the output generated by the map reduce algorithm. Using the input query image , same images are generated by the map reduce using the feature vector comparison. Output for images with same dimensions is high while for images with different dimensions have some variation.

3.6 Running the Map Reduce (b.Multi Node)



```
hadoop@master: ~
17/04/27 02:47:37 INFO mapreduce.Job: map 0% reduce 0%
17/04/27 02:47:45 INFO mapreduce.Job: map 100% reduce 0%
17/04/27 02:47:54 INFO mapreduce.Job: map 100% reduce 100%
17/04/27 02:47:55 INFO mapreduce.Job: Job job_1493282140244_0010 completed successfully
17/04/27 02:47:55 INFO mapreduce.Job: Counters: 49
File System Counters
  FILE: Number of bytes read=236
  FILE: Number of bytes written=207957
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=3276100
  HDFS: Number of bytes written=223
  HDFS: Number of read operations=6
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=5997
  Total time spent by all reduces in occupied slots (ms)=5547
  Total time spent by all map tasks (ms)=5997
  Total time spent by all reduce tasks (ms)=5547
  Total vcore-seconds taken by all map tasks=5997
  Total vcore-seconds taken by all reduce tasks=5547
  Total megabyte-seconds taken by all map tasks=6140928
  Total megabyte-seconds taken by all reduce tasks=5680128
Map-Reduce Framework
  Map input records=29250
```

Fig 3.6.8-1 master & 2 Slave

Running the Image Retrieval on multimode on. This is just a screenshot for the high configuration desktop.

This program is run on various system with different system configuration and with the use of single nodes and myltinode nodes.

3.7 Final System Output

3.7.1 Input Images

Following figure shows a example regarding the complete output that our system generates. Database of images is present in the hdf5 which is shown in Fig- 3.7.1 . It contains images of different objects with variable sizes. Images shown are just sample of the total images used in the experiment.

Following is set of images :



Figure 3.7.1 : Images in database

Image shown is an image on basis of which similar images are retrieved. This is the query image that will be sent in as the input for the program and its feature vector will be compared with the other images present in the database.



Figure 3.7.2: Input Search Image

Following is an output of a system. Depending upon mean value of image all same images from database are retrieved. System outputs all images which are having mean values different from mean of input image. Five variations from input mean is considered in the system.



Figure 3.7.3: Output of System

Chapter 4

Performance Analysis

At first, Hadoop was introduced on single framework i.e. on single node cluster which is crucially utilized for application advancement. After fruition of improvement, single node cluster is slowly expanded to two, three node cluster over different system configuration.

In single node cluster, master node and slave node are same i.e. just a single node where as in multi node cluster there is one master and different slaves. More master node can be conceivable. Every node of the cluster has establishment of Hadoop's Map Reduce. Be that as it may, HBase is restricted to master just which is utilized to store clients login qualifications.

Design of every cluster and the part it plays in the bunch is displayed in Table . At Single Node every single real process will run. These are Namenode process , resource manager , file manager and secondary namenode process and HMaster procedure of Hbase. Slave hubs will just have Datanode, Jobtracker and node manager.

S.no.	Device	Nodes	Memory(Gb)	CPU Conf.	Cpu Freq.(GHz)	Disk (Gb)	OS
1	Laptop	Single	4	Intel Core i5	2.4	48	Ubuntu 15.04
2	Laptop	3	4	Intel Core i5	2.4	48	Ubuntu 15.04(3)
3	Laptop	Single	2	Intel Core i3	1.8	40	Ubuntu 15.04
4	Laptop	3	2	Intel Core i3	1.8	40	Ubuntu 15.04(3)
5	Desktop	Single	4	Dual Core Amd	2	120	Ubuntu 15.04
6	Desktop	3	4	Dual Core Amd	2	120	Ubuntu 15.04(3)

Table4.1 Cluster Configuration

4.1 Experiments and Testing

For doing tests, you require database in this application case image database. Initially experiments were done on static information. When application fundamental procedures were developed i.e. Map Reduce for retrieval, it was tested on dataset of 10 to 15 images.

Experiments are carried out at retrieval stage with a vault of images. These images are stored in dataset of sizes 10, 20,.. to 100 then 200, 300,...1000 ,2000 ,.....5000. A number of them were repeated. All having same format .jpg and same size 640 by 480 pixel.

Again experiments were done with set of 25 images having different formats like .jpg, .gif, .bmp, .png etc. and different sizes. These procedures set aside opportunity to finish. These timings are thought about for execution of framework. Execution of a framework is figured at both transfer and recovery organize.

4.2 Result Analysis

At begin of testing, examinations were done on two machines. Both having same single node cluster setup of Hadoop, yet having different processor, speed, RAM and so forth. Readings of transfer and recovery procedures were taken from both machines for different set of pictures. After perception it is found that guide diminish handle relies on processor form, its speed/recurrence, RAM exhibit into framework. Taking after outcome can definitely highlight this reality about frameworks. Brings about the figure indicates correlation between execution of map reduce transfer process and recovery prepare on single node cluster introduced on portable workstation and desktop. Configuration of both is specified in the above table.

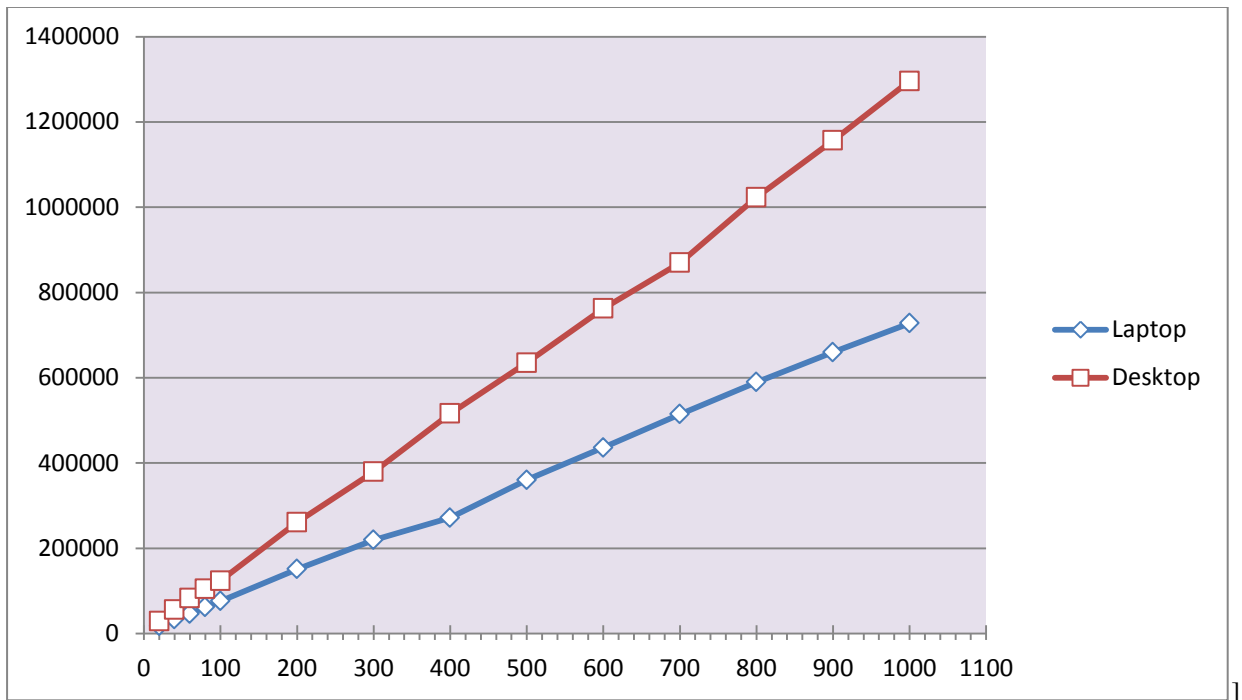


Fig 4.2.1 One Node Cluster at Laptop & Desktop

No. Of Images	Laptop Time(ms)	Desktop Time(ms)
20	17552	29215
40	33260	55626
60	46215	82651
80	62250	105126
100	76226	123621
200	151236	261023
300	219562	379562
400	271562	516220
500	360215	635189
600	435962	762514
700	514562	869541
800	589564	1023154
900	659884	1156624
1000	727859	1295651

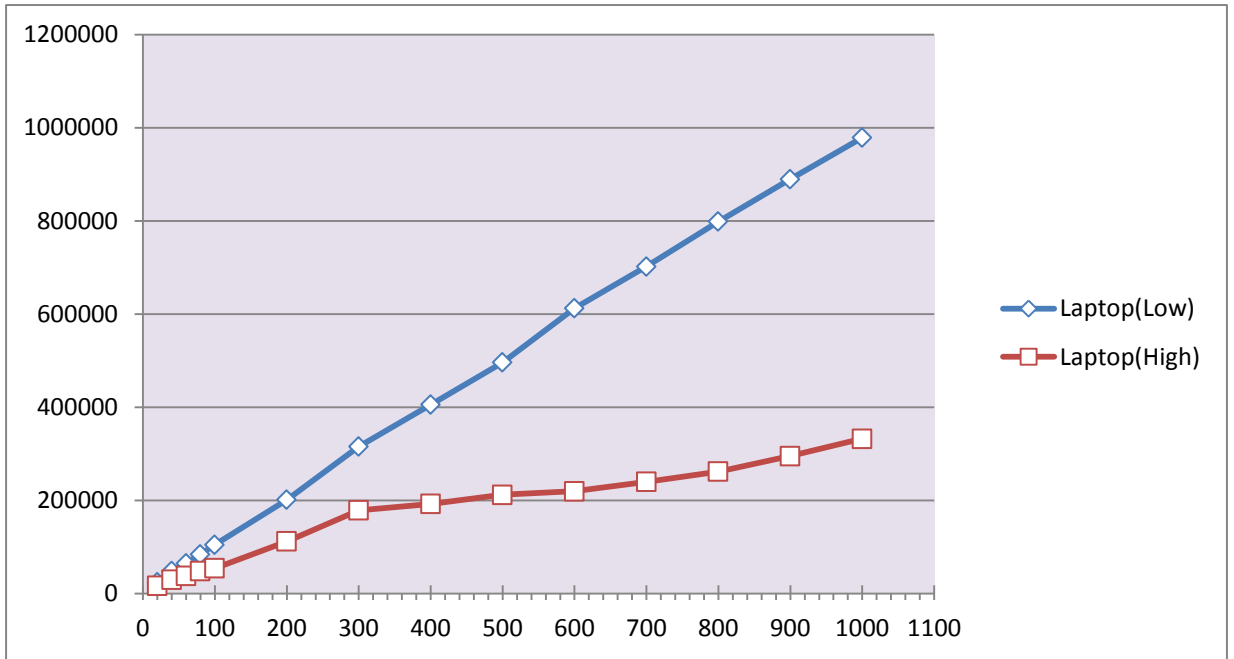


Fig 4.2.2 Multinode Cluster (3 Nodes)

No. Of Images	Laptop(Low) Time (ms)	Laptop(high) Time(ms)
20	24512	17125
40	48615	30321
60	65154	38412
80	84565	48651
100	105148	54561
200	201548	112132
300	315489	178999
400	405518	192615
500	496265	212356
600	612315	219651
700	702156	240221
800	798484	262155
900	889556	295489
1000	978451	332616

Chapter 5

Conclusions

5.1 Conclusion

We experimented and evaluated a different version of proposed system. Its similarity with the proposed system is the Size of dataset used along with the model used for the retrieval.

On similarity measurement front, application is successful in searching and retrieving similar images by precision of 100%, for same format and same dimension of data. Variation in dimension varies the output similarity precision by 20%. But application does not get affected by format change and delivers output with same high similarity.

For execution time complexity, application is tested on desktop systems with different configurations. Systems with latest processors and memory deliver high performance and increases time efficiency nearly by 50%. Throughput of the system increased due to use of latest core processors having high frequency. Main memory also played crucial role in increasing performance.

It does not lag in security frame. But it can be made more secure by using different keys, provided by some secured key management server.

Although system has provided satisfactory results, we consider this as just a first step to provide CBIR system for their use and it opens avenues for further research and developments in this.

5.2 Future Scope

After this implementation , next thing will be the proposed System for retrieval of image using the same concept of word count with different inputs & datatypes.

Image Retrieval has many other future scopes and can be used & improved over many domains. This application can be evolved in Private Information Retrieval, Content Based Image Retrieval and Oblivious transfer i.e. security domains.

Different homomorphic encryption technique can be used, similarity measurement can be changed to improve accuracy, also different feature extraction techniques like Tamura features, Shape features can be used along with color histogram to increase the accuracy of similarity measurement.

This application has large scope in cloud domain and internet world because every domain servers are using cloud technology and they want to provide their customers a new cutting edge applications for imaging which are secure, efficient and delivers with high throughput.

5.3 Application Contribution

Pictures are being produced at an always expanding rate by sources, for example, safeguard and non military personnel satellites, military observation and reconnaissance flights, fingerprinting and mug-shot-catching gadgets, logical tests, biomedical imaging, and home diversion frameworks. For instance, NASA's Earth Observing System will produce around 1 terabyte of picture information every day when completely operational. A Content based image retrieval (CBIR) framework is required to viably and productively utilize data from these picture archives. Such a framework helps clients (even those new to the database) recover significant pictures in view of their substance. Application territories in which CBIR is a key movement are various and assorted. With the late enthusiasm for interactive media frameworks, CBIR has pulled in the consideration of analysts over a few orders.

- A. Medical Applications
- B. Biodiversity Information Systems
- C. Digital Libraries
- D. Security Purposes
- E. Weather Forecasting

References

- [1] T. Mayberry, E.-O. Blass, and A. H. Chan, "Pirmap: Efficient private information retrieval for mapreduce," 2012, <http://eprint.iacr.org/>.
- [2] L. Shi, B. Wu, B. Wang, and X. Yan, "Map/reduce in cbir application," in Computer Science and Network Technology (ICCSNT), 2011 International Conference on, vol. 4, dec. 2011, pp. 2465–2468.
- [3] P. R. Sabbu, U. Ganugula, S. Kannan, and B. Bezawada, "An oblivious image retrieval protocol," in Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, ser. WAINA '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 349–354. [Online]. Available: <http://dx.doi.org/10.1109/WAINA.2011.128>
- [4] J. Zhang, X. Liu, J. Luo, and B. Lang, "Dirs: Distributed image retrieval system based on mapreduce," in Pervasive Computing and Applications (ICPCA), 2010 5th International Conference on, 2010, pp. 93–98.
- [5] Y. Rui and T. S. Huang, "Image retrieval: Current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39–62, 1999.
- [6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23–32, 1995.
- [7] J. Eakins, M. Graham, J. Eakins, M. Graham, and T. Franklin, "Content-based image retrieval," *Library and Information Briefings*, vol. 85, pp. 1–15, 1999.

- [8] Apache hadoop. [Online]. Available: <http://hadoop.apache.org/>
- [9] Mapreduce - hadoop wiki. [Online]. Available: <http://wiki.apache.org/hadoop/MapReduce>
- [10] Hdfs users guide. [Online]. Available: [http://hadoop.apache.org/docs/hdfs/current/hdfs user guide.html](http://hadoop.apache.org/docs/hdfs/current/hdfs_user_guide.html)
- [11] Apache hbase. [Online]. Available: <http://hbase.apache.org/>
- [12] Mapreduce. [Online]. Available: <http://wiki.apache.org/hadoop/MapReduce>
- [13] Eclipse. [Online]. Available: <http://www.eclipse.org/>
- [14] Java platform, enterprise edition. [Online]. Available: [http://en.wikipedia.org/wiki/Java Platform, Enterprise Edition](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)
- [15] Java server pages overview. [Online]. Available: <http://www.oracle.com/technetwork/java/overview-138580.html>
- [16] Apache tomcat. [Online]. Available: [http://en.wikipedia.org/wiki/Apache Tomcat](http://en.wikipedia.org/wiki/Apache_Tomcat)
- [17] Apache tomcat. [Online]. Available: <http://tomcat.apache.org/>

- -