

---

# Big Geo-Data Processing (Spatial Analysis) over Next-Gen Distributed Processing Frameworks

---

## Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Networking Technologies

Submitted By

**SHRUTI C. THAKKER**

**13MCEN25**

Guided By

**Prof. Sapan H. Mankad**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

## Certificate

This is to certify that the major project entitled ”**Big Geo-Data Processing (Spatial Analysis) over Next-Gen Distributed Processing Frameworks**” submitted by **Shruti C. Thakker (13MCEN25)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Networking Technologies) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Prof. Sapan H. Mankad,  
Guide And Assistant Professor  
Computer Science and Engineering  
Institute Of Technology,  
Nirma University, Ahmedabad

Prof. Gaurang Raval  
PG coordinator and Assistant Professor  
Computer Science and Engineering,  
Institute Of Technology,  
Nirma University, Ahmedabad

Dr. Sanjay Garg  
Professor and Head,  
CSE Department,  
Institute of Technology,  
Nirma University, Ahmedabad.

Dr K Kotecha  
Director,  
Institute of Technology,  
Nirma University, Ahmedabad

## Certificate

This is to certify that the project entitled "**Big Geo-Data Processing (Spatial Analysis) over Next-Gen Distributed Processing Frameworks**" submitted by **Shruti C. Thakker (13MCEN25)**, working as a project student at Bhaskaracharya Institute for Space Applications and Geoinformatics(BISAG), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science And Engineering (Networking Technologies) of Nirma University, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree.

Dr. M.B.Potdar  
Director of BISAG  
BISAG,  
Gandhinagar.

## Statement of Originality

---

I, **Shruti C. Thakker**, Roll. No. **13MCEN25**, give undertaking that the Major Project entitled "**Big Geo-Data Processing (Spatial Analysis) over Next-Gen Distributed Processing Frameworks**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Networking Technologies)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

---

Thakker Shruti C.

Date:

Place: Ahmedabad

Endorsed by  
Sapan H. Mankad  
(Signature of Guide)

## Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Sapan Mankad (Nirma University)** Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad and **Mr. Abdul Jhummarwala (BISAG-Gandhinagar)**, for their valuable guidance and continual encouragement throughout this work. The appreciation and continual support they has imparted has been a great motivation to me in reaching a higher goal. Their guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad and **Dr. M.B.Potdar**, Academic Director of Bisag, Gandhinagar for their kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. K. Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad and **Mr. T.P.Singh**, Hon'ble Director, BISAG, Gandhinagar for the unmentionable motivation. They has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- **Shruti C. Thakker**

**13MCEN25**

## Abstract

High-performance computing is the field of computer science dealing with the use of distributed systems to perform complex tasks by the use of low cost off-the-shelf traditional systems. Various distributed frameworks are available as of today for creating clusters and grids to perform tasks that cannot be processed locally or even on high-end servers due to their inherent computationally complex nature, memory and IO requirements. Today's geographic information systems not only serve geographic studies and remote sensing but require integration of different technologies for analysis of data, modelling and simulations. The largest amount of raw data is added by Geospatial systems to the information systems in form of images, etc. The field of GIS has huge computational demands while the traditional desktop based GIS software has not been able to keep up with the demands and the new challenges of processing Big Geo-data. Moreover, the distributed and high-performance frameworks have not been specifically designed for spatial analysis. Design and implementation of a distributed Spatial analysis framework is required for processing multi-gigabytes of geo-data (raster and vector in addition to tabular attributes). The designed framework should be scalable and also be interoperable with various OGC standards. The framework should also provide Geo-Analytics and GeoAPI to be easily integrated to be use with and is extensible by external software packages.

# Abbreviations

<b>GIS</b>	Geographic Information Systems.
<b>QGIS</b>	Quantum Geographic Information Systems.
<b>OGC</b>	Open Geospatial Consortium.
<b>WKT</b>	Well-Known Text
<b>WKB</b>	Well-Known Binary
<b>YARN</b>	Yet Another Resource Negotiator
<b>HDFS</b>	Hadoop Distributed File System

---

# Contents

Certificate	ii
Certificate	iii
Statement of Originality	iv
Acknowledgements	v
Abstract	vi
Abbreviations	vii
List of Figure	xii
List of Table	xiii
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	1
1.2 Motivation and Need of Research . . . . .	1
1.3 Project Scope and Objective . . . . .	2
<b>2 Background Knowledge of a Project</b>	<b>3</b>
2.1 What is Big Data? . . . . .	3
2.1.1 The Three V's of Big Data . . . . .	4
2.2 Introduction of Geo-Processing . . . . .	5
2.2.1 Graphical Representation of Geo-Processing Image . . . . .	6



2.3	Open Geospatial Consortium (OGC) . . . . .	7
2.3.1	WKB Format . . . . .	7
2.3.2	WKT Format . . . . .	8
2.4	GIS Data Formats . . . . .	9
2.4.1	Raster Data . . . . .	9
2.4.2	Vector Data . . . . .	10
2.5	Major Challenges To Handling Big Geo-Data . . . . .	12
2.6	What is Distributed System? . . . . .	14
2.7	Why Distributed System is used for GIS? . . . . .	14
<b>3</b>	<b>Literature Survey</b>	<b>16</b>
3.1	Hadoop . . . . .	16
3.1.1	Hadoop Distributed File System(HDFS) . . . . .	16
3.1.2	MapReduce Programming Model . . . . .	18
3.1.3	Comparison of MapReduce 1(Classic) and Yet Another Resource Negotiator(YARN) . . . . .	19
3.1.4	Yet Another Resource Negotiator(YARN) . . . . .	19
3.2	Spatial Hadoop . . . . .	20
3.3	Summary of Survey Paper . . . . .	23
3.3.1	Advancing a Geospatial Framework to the MapReduce Model	23
3.3.2	CG-Hadoop: Computational Geometry in MapReduce . . . .	24
3.3.3	Hadoop GIS: A High Performance Spatial Data Warehousing System over MapReduce . . . . .	24
3.3.4	A Demonstration of SpatialHadoop: An Efficient MapReduce Framework for Spatial Data . . . . .	25
3.3.5	Constructing Gazetteers from Volunteered Big Geo-Data based on Hadoop . . . . .	26
3.3.6	Efficient Skyline Query Processing in Spatial Hadoop . . . . .	27
<b>4</b>	<b>Implementation</b>	<b>28</b>

4.1	Geo-Processing Operation Using QGIS . . . . .	28
4.1.1	Convexhull Operation on Shape File Using QGIS . . . . .	28
4.2	Steps for Setting up Multinode Hadoop . . . . .	30
4.2.1	Snapshot of MultiNode Hadoop Installation . . . . .	33
4.2.2	Install and Configure Spatial Hadoop on top of Multi-node Hadoop Cluster . . . . .	36
4.3	Shape File Conversion . . . . .	37
4.4	How conversion of shape file will be used in distribution of file? . . .	41
<b>5</b>	<b>Result and Discussion</b>	<b>42</b>
5.1	Result . . . . .	42
5.2	Discussion . . . . .	49
<b>6</b>	<b>Conclusion and Future Scope</b>	<b>51</b>
6.1	Conclusion and Future Scope . . . . .	51
	<b>Appendix A Published Paper</b>	<b>52</b>
	<b>References</b>	<b>52</b>

# List of Figures

2.1	Schematics of GeoProcessing Workflow . . . . .	5
2.2	Process and Analysis of Image . . . . .	6
2.3	Process and Analysis of Image . . . . .	6
2.4	Representation of Raster Image . . . . .	10
2.5	Representation of Vector Image . . . . .	11
2.6	Representation of Distributed System . . . . .	15
3.1	Representation of HDFS Architecture . . . . .	17
3.2	Representation of MapReduce Architecture . . . . .	18
3.3	Representation of YARN Architecture . . . . .	20
3.4	Architecture of SpatialHadoop . . . . .	21
4.1	Display shape file using QGIS . . . . .	29
4.2	Display convexhull operation on shape file using QGIS . . . . .	29
4.3	Snapshot of starting all service using jps command on masternode . .	33
4.4	Snapshot of starting all service using jps command on slavenode1 . .	33
4.5	Snapshot of starting all service using jps command on slavenode2 . .	34
4.6	Snapshot of information about cluster . . . . .	34
4.7	Snapshot of namenode information . . . . .	35
4.8	Snapshot of jobtacker information . . . . .	35
4.9	Snapshot of fetching file . . . . .	38
4.10	Snapshot of display shape file of Point . . . . .	38
4.11	Snapshot of validation of file . . . . .	39
4.12	Snapshot of display shape file in WKT format . . . . .	39

4.13	Snapshot of display shape file of multilinestring . . . . .	40
4.14	Snapshot of display shape file of multilypolygon . . . . .	40
5.1	Snapshot of generating and indexing of 10MB file using SpatialHadoop	42
5.2	Snapshot of apply convexhull operation on 10MB file using SpatialHadoop-1 . . . . .	43
5.3	Snapshot of apply convexhull operation on 10MB file using SpatialHadoop-2 . . . . .	43
5.4	Snapshot of generating and indexing of 50MB file using SpatialHadoop	44
5.5	Snapshot of apply convexhull operation on 50MB file using Spatial-Hadoop . . . . .	44
5.6	Snapshot of generating 100MB file using SpatialHadoop-1 . . . . .	45
5.7	Snapshot of generating 100MB file using SpatialHadoop-2 . . . . .	45
5.8	Snapshot of indexing and apply convexhull operation on 100MB file using SpatialHadoop . . . . .	46
5.9	Snapshot of apply convexhull operation on 100MB file using Spatial-Hadoop . . . . .	46
5.10	Snapshot of all files which are stored in DFS . . . . .	47
5.11	Snapshot of multiple files of 500MB file generated by HDFS . . . . .	47
5.12	Snapshot of output of convexhull operation of 500MB file on browser	48
5.13	Graph of Size vs. Time for convexhull operation using SpatialHadoop	49

# List of Tables

2.1	WKT Format . . . . .	8
2.2	WKT Format . . . . .	9
4.1	Details of the Library/JAR . . . . .	37
5.1	Table of size and time values for convexhull operation using Spatial- Hadoop and QGIS . . . . .	48

# Chapter 1

## Introduction

### 1.1 Problem Statement

Due to the increasing internet users and increasing a machine to machine connection, volume of data is growing rapidly. Data is in variety of the format like text, audio, video, email, images etc. Handle these types of data is very difficult and it is challenging task for big organizations and companies. To handle big data, many problems come across such as heterogeneity of data, security, require high speed for transformation of data, scalability, manipulation of data, analysis of data and require more storage capacity of hardware. Using the parallel and distributed processing, we can solve these problems very efficiently.

### 1.2 Motivation and Need of Research

In large scale applications such as Government Projects, Business and Industrial Projects including real estate, public health, natural resources, climatology, community planning, and transportation etc make use of Geographic Information System(GIS). GIS is used for capture, manipulate, manage, store, analyze and present all types of spatial data. Information of data like as number of datasets, its size, quality and formats are changed based on requirement of organizations and companies. Also spatial data is generated periodically via special sensors (like OGC, MSS,

TM, ETM+ and OLI&TIRS etc [3]), satellites and GPS devices. It is very difficult to handle all information and do the processing of image using normal GIS software like ArcGIS, QGIS, ArcMap, ArcObject, GRASS, OpenJUMP, AGISMAP etc. All these softwares are taking more time to process on large size of data, so it is not preferable to use GIS software for Big-Geo data. Do the processing on spatial data using one single machine is challenging task, for this reason we really need to research for handling and analysis of Big Geo-data (spatial data) using more than one node. In this project, using Hadoop distributed processing framework, doing processing task on data in more than one node at the same time is framed in detail.

### 1.3 Project Scope and Objective

There are many problems to handle Big-Geo data, so that main goal of this project is to handle and process on large amount of data, which is in the form of spatio-temporal data. In this project, Big Geo-Data is used which contain the information about spatial and the size of these data is very large, so that to handle and do some operation on Big Geo-data, distributed processing framework is used which is Hadoop and for distribution of large amount of data, improved version of Map-Reduce programming model is used which is YARN and Hadoop Distributed File System(HDFS). In this project, spatial data is used and is processed by Spatial Hadoop because it is mainly used for operation on spatial data and also is a Hadoop extension which reduces the process time of large data sets using MapReduce framework.

# Chapter 2

## Background Knowledge of a Project

### 2.1 What is Big Data?

- **Big Data:** Big Data is used to define the large volume of data (in the form of structured, semi-structured, and unstructured data) on various aspects of the environment and society[6]. These data are created by millions of people constantly, in a variety of formats such as maps, videos, audios, and photos. In Big Data, the term “big” is not only used for a huge amount of data, but also used for the multiple perspectives like different topics, scale, dimension etc.
- **Geodata:** Geodata is used to define data set, which has a spatial content and is also called as “GIS data”, “Spatial data” or “Geographic data”. In Geodata, the term “Geo” is used for spatial data set that allows to georeference that described location or region on the earth. Geodata consist of information about geographic locations that is stored in a GIS formats which are raster and vector and it is used in a Geographic Information System (GIS).
- Both the terms “Big data” and “Geo data” are used together which forms “Big Geo-Data”. These terms are used for specifically large amount of spatio-



temporal data from which we can acquire the information about geographic locations.

### 2.1.1 The Three V's of Big Data

- **Volume:** In Big Data, volume is used for the size of data sets as well as their inter-linkage which creates a global graph of linked data. These large volume of data is getting from location-based social networks, geographic information, smart dust and sensor networks in general, radar, satellite, complex transportation simulations and historical records, high resolution remote sensing data also increasing the internet user and more device to device communication. For storing these all data, there is requirement of multi bytes of the storage system.
- **Variety:** Big-data is stored in multiple format. For example database access, excel, csv or in a simple text format. Also it can be in different format like SMS, video, audio, pdf, email etc. All organizations need to arrange, manage and govern different variety of data and make it meaningful. If data is in the same format, it will be easy to do all task, but all the data in the same format is not possible in each and every case. Also the real world data are in different formats and this is the challenge to overcome with the Big Data. This variety of the data represent Big Data.
- **Velocity:** Velocity indicate the speed of data creation, streaming, and aggregation. Real time data is generated constantly and that it impossible to handle for traditional systems. Big Data is not only about large amount of data but also the speed at which data is created and updated. Real-time data poses new challenges for managing, storing, analysing and updating data. Velocity is mainly considered to know how quickly the data is generated and stored, and also how quickly it can be retrieved. In Big Data, velocity is also be applied in speedy transformation of data from one device to other device via communication link.

## 2.2 Introduction of Geo-Processing

- Geo-processing is the process of geographic information and is used to define, manage, storage, manipulate and analyze geographic information to make decision for GIS data. As shown in below figure for different operations, different types of input parameters are used that generate intermediate or final output. We can also go back to the process more than once (0 to N) after getting the output and do the different or same operations and generate final output. Geo-processing operations are geographic feature overlay, feature extraction and image/data analysis, topology processing, data conversion, image enhancement, image filtering, image classification etc.

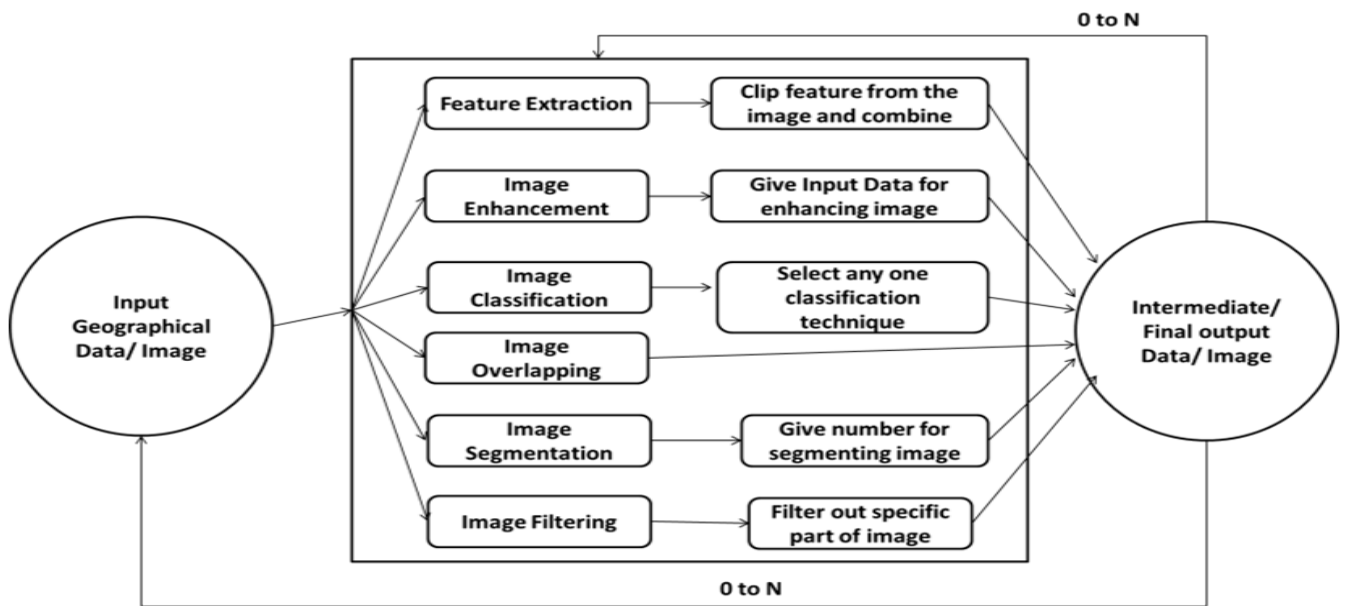


Figure 2.1: Schematics of GeoProcessing Workflow

### 2.2.1 Graphical Representation of Geo-Processing Image

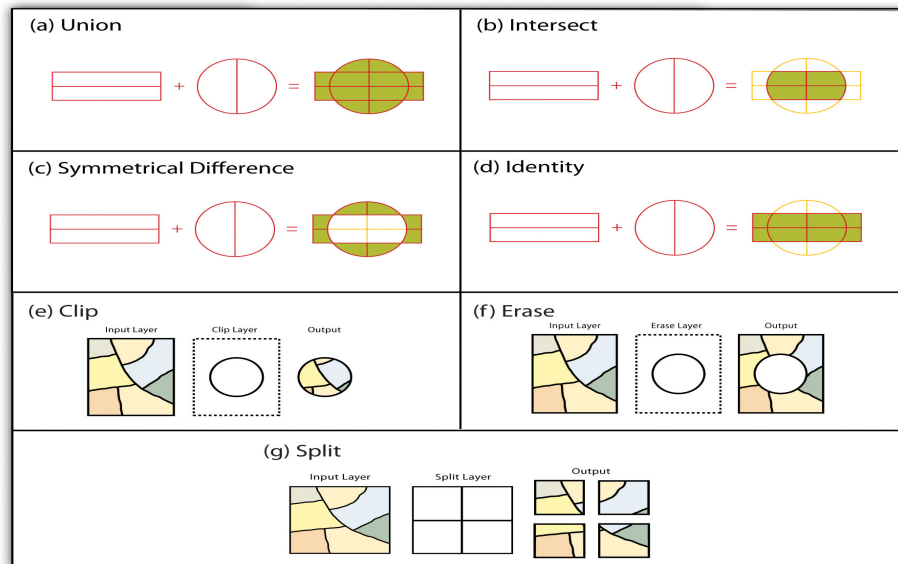


Figure 2.2: Process and Analysis of Image

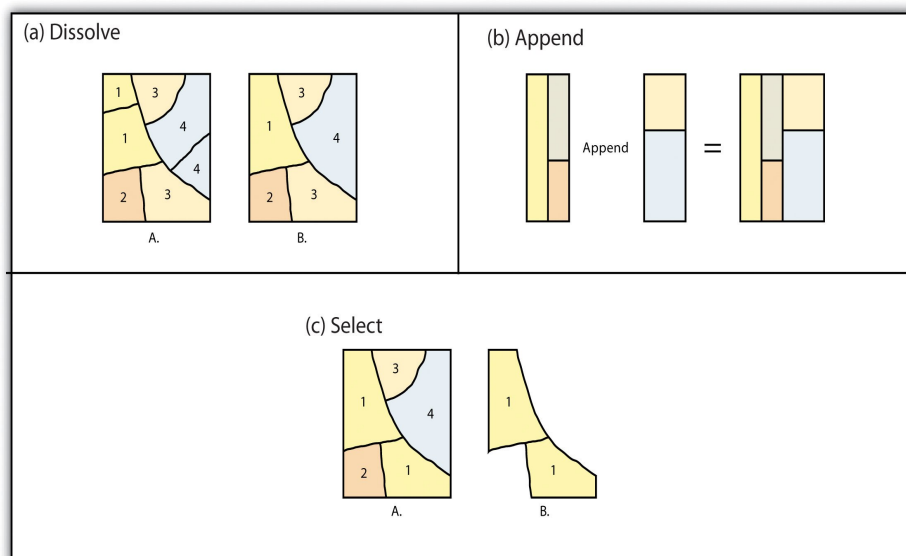


Figure 2.3: Process and Analysis of Image

## 2.3 Open Geospatial Consortium (OGC)

- OGC standard [9]: The Open Geo-spatial Consortium (OGC) is an international industry consortium of 387 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OpenGIS Standards support interoperable solutions that "Geo-enable" the Web, wireless and location-based services, and mainstream IT. It is Not-for-profit, international standards development consortium.
- Two main Standard use by OGC in GIS: WKT (Well Known Text), WKB(Well Known Binary).

### 2.3.1 WKB Format

- Full form of WKB is Well-Known Binary, Representation of geographical and geometrical data using WKT format. Well-known binary (WKB) format is represented using hexadecimal strings[10].
- WKB uses 1-byte for unsigned integers of byte order, 4-byte for unsigned integers of WKT type, and 8-byte for double-precision numbers of coordinate value .The first byte denote the order of byte. 00 for big endian, or 01 for little endian.
- For example, A POINT(2,4) is denoted using WKT format, then it consists of sequence of 21 bytes (each represented using two hex digits):

0101000000000000000000F03F4010000000000000

- The sequence is to be break into these four components:

Byte order : 01

WKB type : 01000000

X : 4000000000000000

Y : 4010000000000000

### 2.3.2 WKT Format

- Full form of WKT is Well-known Text. It is a text markup language, Which is use for representing vector geometry Objects such as point, linestring, polygon using a map or image. Geometries coordinates are in the form of 2D (x, y), 3D (x, y, z), 4D (x, y, z, m), where m is a part of linear referencing system. In WKT format, coordinate points are separated by comma(.). For more than one points, lines and polygon use the multipoint, multilinestring and multipolygon respectively. If geometries are not contain any coordinates , then it can be indicate using EMPTY symbol after the type name[10]. Below table represent a WKT format??:

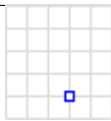
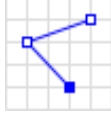
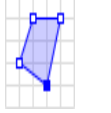
	– point	– POINT(30 10)
	– Line	– Line(40 20, 30 50)
	– Polygon	– POLYGON ((20 50, 10 10, 30 50, 20 30, 40 20))

Table 2.1: WKT Format




	– MultiPoint	– MULTIPOINT ((20 10), (50 20), (30 30), (40 20))
	– MultiLine	– MULTILINESTRING ((20 20, 30 30, 20 50),(50 50, 40 40, 50 30, 40 20))
	– MultiPolygon	– MULTIPOLYGON (((40 30, 55 50, 20 50, 40 30)),((25 15, 30 20, 20 30, 15 20, 25 15)))

Table 2.2: WKT Format

## 2.4 GIS Data Formats

To store and retrieve geographical data, GIS is use two type of formats:

- 1) Raster Data
- 2) Vector Data

### 2.4.1 Raster Data

Raster data are used for storing data that varies continuously, as in an aerial photograph, a satellite image, a surface of chemical concentrations or an elevation surface. Raster data is represented by pixels (or cells). Pixels are arranged in rows and columns with equal size and collation of single or multiple bands. Each pixel is assigned a value. If a pixel in a matrix have same values or colour, it represent the same type of geographic feature. In raster format, all data is represented as a grid of cells (usually square). OGC introduced GML (Geography Markup Language), WKT

(Well Known Text) and WKB (Well Known Binary) as interoperable alternatives to various types of non-structured data formats.

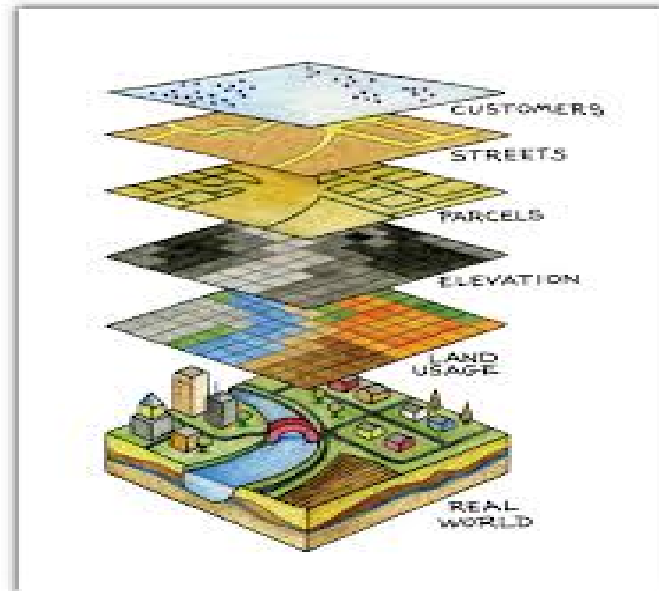


Figure 2.4: Representation of Raster Image

### Use of Raster Image

- Raster graphics are mainly used in web design where photography is required.
- When the cell is very small in information array, raster format is more efficient.
- If we want to add an effect like a texture, design, blur or any other manipulation in image, raster graphics are used.

### 2.4.2 Vector Data

Vector data is used for representation of the features of the image like points, lines, polygons and complex shapes. If the data is containing different types of the boundaries for different features, like streets, country borders, road and land parcels etc. This data consists of points, which (2D and 3D data) are stored using co-ordinates of (X,Y) or (X,Y,Z) pairs. The points are joined in a particular order which create lines or joined into closed rings which create polygons. Vector data contains lists of

co-ordinates and these coordinates are used to define vertexes. Using these vertexes, we can determine the shape.

Vector Data are stored in Three files:

- 1) A file, consisting of location information using longitude and latitude value.
- 2) A file, consisting of information about the attribute.
- 3) A file, consisting of information needed to link positional data with their attributes.

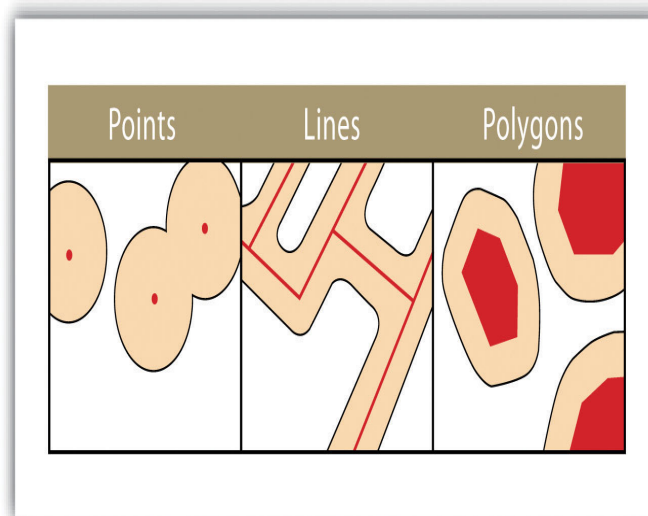


Figure 2.5: Representation of Vector Image

### Use of Vector Image

- Vector Data are used to store the information about points, lines, or boundaries enclosing areas.
- Vector data is used to design something for large scale such as vehicle wraps, banners, signage and other large format items.
- Vector graphics are used to design logo, symbol, business identity print work, promotional posters.
- When require the comparison of information whose geographical shape and sizes are different, vector formats work very efficiently.



## 2.5 Major Challenges To Handling Big Geo-Data

Due to recent advances in remote sensing (RS) and computer techniques, growth of remote sensing (RS) data is very explosive. The observation data streaming from the spacecrafts in current active missions of NASA would approximately be 1.73 gigabytes [6]. Data/Image is gathered by a single satellite data center, and it is dramatically increasing by several terabytes per day. These data is giving the information about geographic locations, which is stored in a raster or vector format and it is used in geographic information system (GIS), so handling these type of Big Geo-data(Spatial Data), we are facing some problems, which are:

- **Required More Storage Capacity:**Size of Data set is grows rapidly because they are gathered by ubiquitous information-sensing mobile devices, cameras, aerial sensory technologies, wireless sensor networks, logs, microphones, radio-frequency identification readers and so on. For storing and capturing large volume of data, we require more storage hardware and also require higher I/O speed to meet the challenges.
- **Managing Data:**Management of data is the most difficult problem in big data. The sources of collecting data are varied by temporarily, spatially and in the form of format. The data can be store in different way such as documents, drawings, sound and video recordings, models, pictures etc – with or without describing what, where, who, when, why and how it is collected[7]. Managing all these information about the large volume of data is very difficult.
- **Effective Analysis of Data:** Data is very huge and store in different place (if use cloud for storing data ) and it takes a lot of understanding to get data in the right format and process on it. For example, if the data comes from any social media like TV, Internet,radio etc , then we need to know who are the users such as a customer which use particular set of products – and understand what it is trying to visualize out of the data and based on the requirement of

the data, we have to analyze it and for big amount of data it require more time and more computational task.

- **Security:** Security problems include data security protection, property protection, personal privacy protection, commercial secrets and financial information protection. Most developed and developing countries have already made related data protection laws to enhance the security. It is necessary to know, where data is stored and process data, to make sure that they are in compliance with the rules and regulations. For Big Data related applications, data security problems are more awkward for several reasons. Firstly, the size of Big Data is extremely large, channelling the protection approaches. Secondly, it also leads to much heavier workload of the security. Otherwise, most Big Data are stored in a distributed way, and the threats from networks also can aggravate the problems.[8]
- **Scalability:** Size of the big data is very large(more than Tera byte) and it keeps on changing very frequently. Managing large and rapidly increasing volumes of data has been a challenging issue for many decades. In the past, the challenge was mitigated by processors getting faster, but the volume is increasing faster than CPU speeds and other compute resources, it is difficult to handle and not possible to process on data using one node, so that we have to migrate on parallel data processing technique and processors are being built with increasing numbers of cores.
- **Heterogeneous Data:** Heterogeneous data is data from any number of sources, largely unknown and unlimited, and in many varying formats. Big-data is stored in multiple format. For example database access, excel, csv or in a simple text format. Also now data is not even in the traditional format as we assumed, it may be in the form of SMS, video, audio, pdf, email etc. And all organizations need to arrange, manage and govern different varieties of data and make it meaningful. To process on heterogeneous data, require different techniques,

different softwares and different technologies.

- **Required High Speed:**The data is in very large amount and in different format, so accessing it require more bandwidth and low latency. The bandwidth and latency are two main features which is effective in the communication between the clients and the cloud server[8]. Huge amount of time is required for processing of Big Data, and to overcome these problems, we use hadoop distributed system and HDFS.

## 2.6 What is Distributed System?

- A distributed system is a "collection of independent computers that appear to the users of the system as a single coherent system" also it has been stated as "It is a system in which components of the system are located at networked computers and communicate with each other via message passing and any other communication technique". Example of distributed system are Web(WWW), Email, Sensor Network, DNS, Distributed File System(NFS) etc.
- Distributed system is collection of distributed hardware, distributed data and its control ( MPI, OpenMP, RPC etc).
- We can represent distributed computing and parallel computing as shown in figure:

## 2.7 Why Distributed System is used for GIS?

- To provide accurate and up-to-date data for analysis in distributed system, GIS is used. For analysis the design of distributed system, professional persons (like as engineers and scientist ) have exported data from GIS data sets to third party software.
- Applications and functionality of data analysis require different model and operating scenarios, which was not fully recognized in GIS software. So GIS

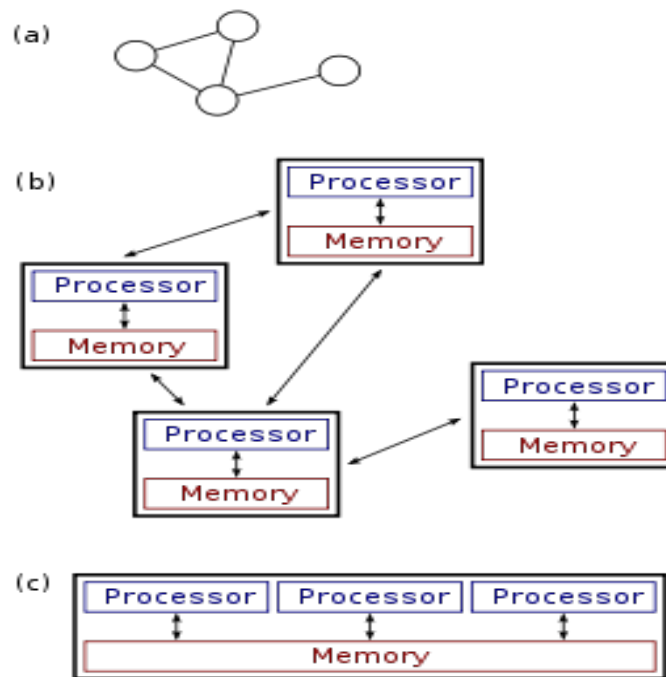


Figure 2.6: Representation of Distributed System

environment has to use this data with advanced modelling and planning tools and also need to switch distributed or parallel processing system.

- So that, for analysis Of data very Fast and provide scalability and good computational work we used distributed system For GIS.

# Chapter 3

## Literature Survey

### 3.1 Hadoop

In our day to day life with extreme demand of the resources, single system could not provide high performance and high efficiency very easily. To achieve these parameters we have to move on distributed system. Distributed System is good in terms of cost, performance, scalability, reliability, and inherent distribution. Hadoop is open source software framework for processing of large amount of data in distributed manner. Two techniques are used by Hadoop: Hadoop Distributed File System (HDFS) and MapReduce.

#### 3.1.1 Hadoop Distributed File System(HDFS)

HDFS is the distributed file system used by the Hadoop project. HDFS is designed to be deployed on low-cost hardware and highly fault-tolerant. HDFS provides high throughput access to data for applications that have large data sets [14]. For high scalability, reliability and capability of storing very large files, HDFS is best choice to handle it. HDFS use two type of node: One Name node and Several Data nodes. Name node is a manage the file system namespace and store the information about the file like access rights, size of file, and location of file [3]. Every node has assign one or more data nodes in the cluster, which store actual file (Record) [3]. In HDFS, A

large file is split into one or more Blocks/Chunks (Normally 64MB) and these blocks are stored in a Data Nodes [3]. Name node executes file system namespace operations like opening, closing, and renaming files and directories [14]. In HDFS, Name node is unique and it is used for managing and storing data in memory. Here data is metadata so that according to available memory of each node, Limited number of files can be stored by the system. It also determines the mapping of blocks to Data Nodes [14]. The Data Nodes is used for read and write requests which are getting from the clients. As per the instruction given by Name node, Data nodes are perform operation like as replication, deletion, creation of a file [14].

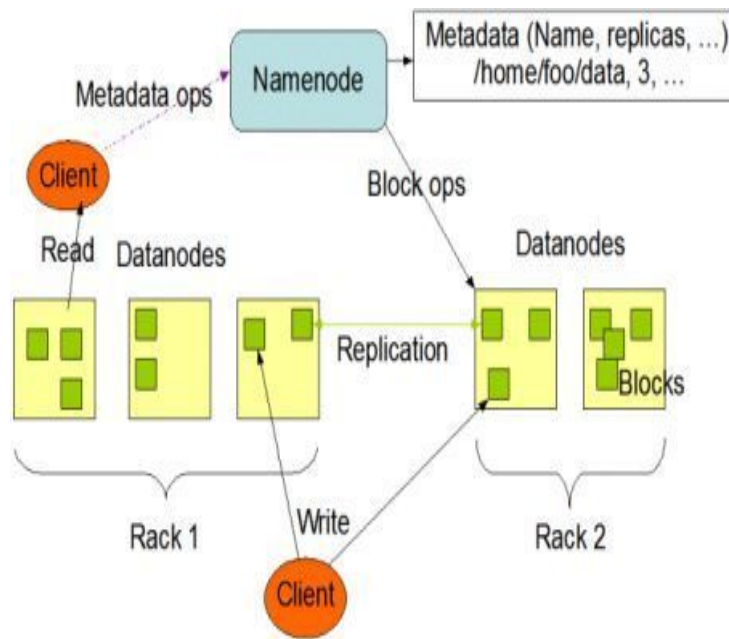


Figure 3.1: Representation of HDFS Architecture

Secondly, HDFS uses multi copy strategy so that data can be stored in many nodes in addition to have replicated data[6]. Now, when data node need to call original copy of data and if it suffers failure then name node can call the replicate data on the other machine, and this multi copy data storage strategy can effectively improve the reliability and availability of data storage.

### 3.1.2 MapReduce Programming Model

MapReduce programming model is used for execute programs very efficiently on large clusters using data distribution and parallel processing technique. These model is motivate by functional programming and it contains two user define functions, Map and Reduce. The user is write the functions and these is be covered by the Map and Reduce functions. In application, each job is contain exactly one Map function and which is followed by Reduce function, and the order of the execution cannot be changed. Also if an algorithm requires more than one Map and Reduce steps, then it can be implemented via applying a separate jobs for each, and generated output is to be transferred from one job to the next job using the file system. The map function maps a single input record (in form of  $(K1, V1)$  Pair) to a set of intermediate key value pairs  $(K2, V2)$  while the reduce function takes all values associated with the same intermediate key and produce the final output  $(K3, V3)$  Pair[13].

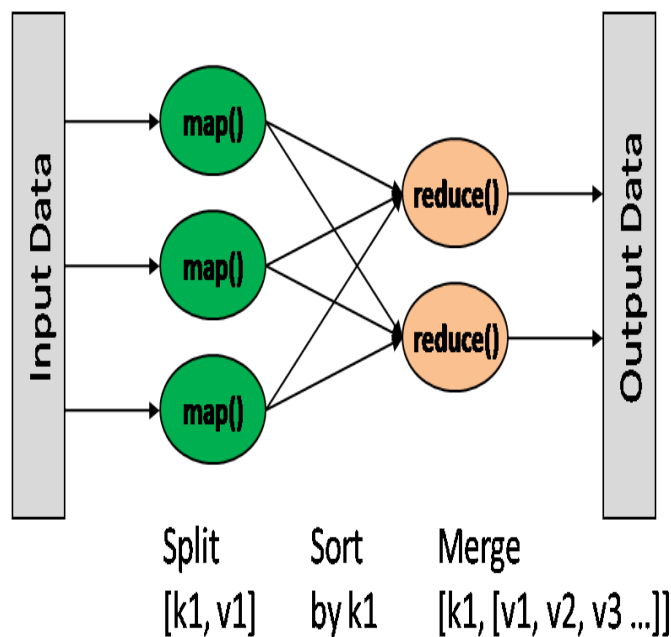


Figure 3.2: Representation of MapReduce Architecture

A MapReduce program is inherently parallel and can greatly decrease the computational time when processing huge dataset. MapReduce mainly used in wide

range application like machine learning, graph processing, web application in traversal etc.

### **3.1.3 Comparison of MapReduce 1(Classic) and Yet Another Resource Negotiator(YARN)**

YARN is the improve version of the MapReduced 1, its goal is to allow the system to serve as a general data processing framework. YARN is characterized as a large-scale, distributed operating system for big data applications. Due to some limitations of MapReduce version 1(Classic) like hard partition of resources into map and reduce slot, limited size of cluster node(max 4000 node[17]) and limited concurrent task (max 40,000 task), scalability, capacity of processing huge amount of data, fault-tolerance, query execution time can often be several hours we have to switch over to YARN [15]. Since, YARN is support MapReduce and Non-MapReduce application on the same cluster, all node have its own resources (like memory, CPU) which are allocated to application when request is needed, improve cluster utilization, more scalable compare ( max 45,000 nodes[17]) to MapReduce classic etc[15]. A YARN is software that rewrites and decouples MapReduce's resource management and schedules the capabilities from the data processing component.

### **3.1.4 Yet Another Resource Negotiator(YARN)**

There is no changes in the programming model or in HDFS of YARN. To remove the bottlenecks of the master-slave architecture, YARN is only redesign run time system. In YARN architecture, JobTracker is split into two different processes which are the ResourceManager and the ApplicationMaster[15]. ResourceManager is responsible for a distribute resources to the different applications which are running in the cluster. Instead of static Map/Reduce task, resources is distributed using the notion of containers. Based on status of available memory, disk and CPU capacity, each container is configure. The scheduler schedules the task based on the requirements of resources of each application. It also has a pluggable scheduler, which can



use different strategies to assign tasks to available nodes[15]. Application master is use for negotiating resource containers from the scheduler, monitoring their progress and tracking their status. TaskTracker is replaced with the NodeManager. Node Manager is the per-machine slave, which is responsible for launching the applications containers, monitoring their resource usage such as CPU, memory, disk, network and reporting to the ResourceManager[15].

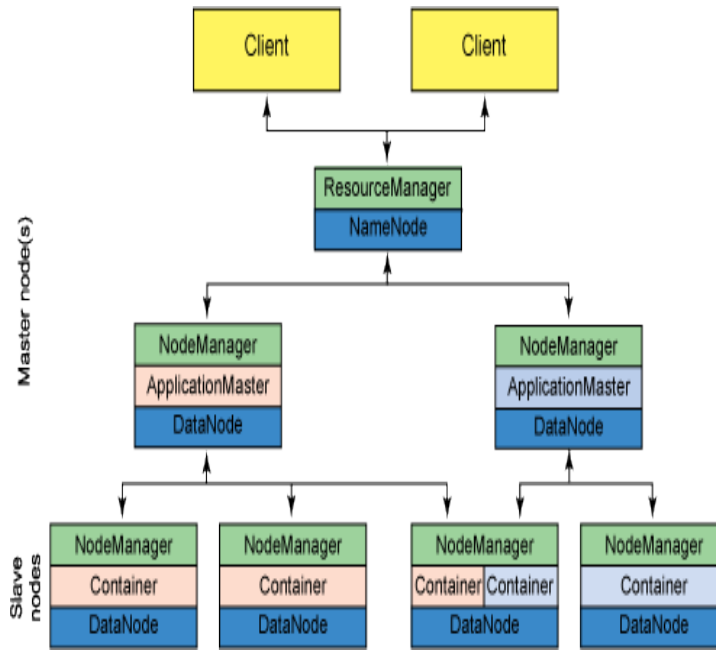


Figure 3.3: Representation of YARN Architecture

## 3.2 Spatial Hadoop

Spatial Hadoop is a comprehensive extension of Hadoop which is specially used for spatial data. In Spatial Hadoop, each layer of Hadoop namely language, storage, Mapreduce and operations layer, is aware of spatial data. Hadoop does not use spatial indexes before used for process any spatial data, so it has to scan whole dataset to generate a result, which take very time compare to spatial Hadoop and also give very bad performance.

Type of user in SpatialHadoop

1. **Simple User/Client:** Who access SpatialHadoop for process their datasets using a spatial language.
2. **Developers :** Who has a deeper knowledge of the system and can add or implement a new spatial operations like segmentation, registration, conversion etc.
3. **Administrators:** Who can manage the system by adjusting system parameters in the configuration files.

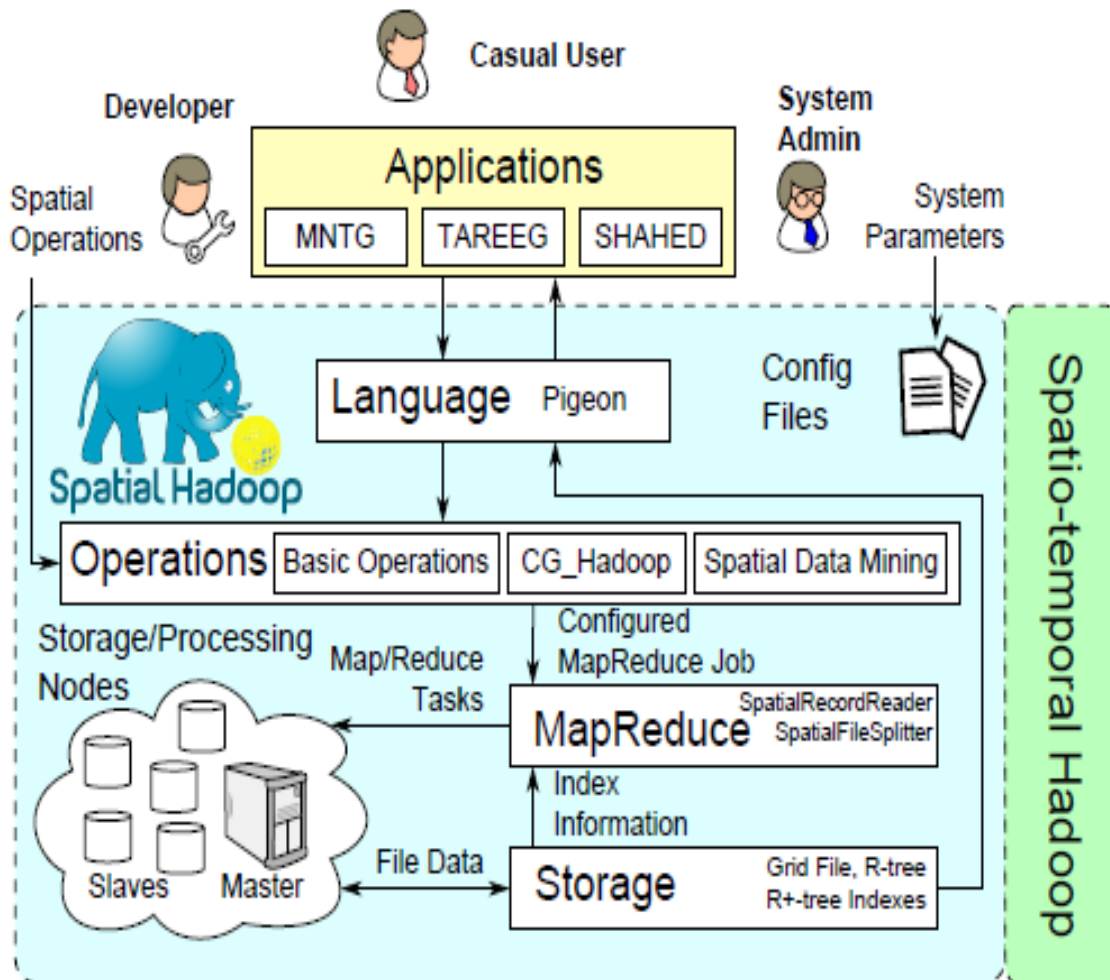


Figure 3.4: Architecture of SpatialHadoop

As explain below, Four types of layers are in Spatial Hadoop:

**1. Language Layer:** In Hadoop, program is written Pig latin language, which is not reliable for spatial data. But in Spatial Hadoop uses pigeon language which is uses spatial data types and spatial function.

**2. Storage Layer:** SpatialHadoop employs spatial index structures within Hadoop Distributed File System (HDFS) as a means of efficient retrieval of spatial data. Indexing in SpatialHadoop is the key point in its superior performance over Hadoop[19]. SpatialHadoop employs a two-level index structure of global and local indexing. The global index partitions data across computation nodes while the local indexes organize data inside each node. SpatialHadoop uses the proposed structure to implement three standard indexes, namely, Grid file, Rtree and R+-tree.

**3. MapReduce Layer:** The MapReduce layer in SpatialHadoop is the query processing layer that runs MapReduce programs. However, contrary to Hadoop where the input files are non- indexed heap files, SpatialHadoop supports spatially indexed input files. SpatialHadoop enriches traditional Hadoop systems by two main components: (1) SpatialFileSplitter an extended splitter that exploits the global index(es) on input file(s) to early prune file blocks not contributing to answer, and (2) SpatialRecordReader which reads a split originating from spatially indexed input file(s) and exploits the local indexes to efficiently process it.[19]

**4. Operation Layer:** The combination of the spatial indexing in the storage layer with the new spatial functionality in the MapReduce layer gives the core of SpatialHadoop that enables the possibility of efficient realization of a myriad of spatial operations[19]. SpatialHadoop contain geometry operations e.g. kNN, Spatial join, ConvexHull, Overlay, Farthest-pair, Nearest-pair etc.

### 3.3 Summary of Survey Paper

#### 3.3.1 Advancing a Geospatial Framework to the MapReduce Model

**Authors** Roberto Giachetta

**Publication Year:**2011

Roberto Giachetta [4] is evaluate AEGIS spatio-temporal framework using cloud based spatial data processing. For this Data process used MapReduce via Apache Hadoop implementation. This framework is improve data storage capabilities and the processing model allowing the previously implemented algorithms to be easily adapted to distributed execution without the need of any transformation. Using this framework , it allow the processing of complex geospatial data including remotely sensed imagery in the Hadoop environment.

AEGIS framework is geospatial toolkit, which used OGC standard and state of art programming methodology. AEGIS supports both vector geometry and remotely sensed images based on the well known simple feature access standard.

In Data management Component of the framework used three services: Data Import/Export, Data Maintenance service and catch manger.

In this paper [4], operations are performed using the MapReduce model in the framework

- 1)In case no merging of results is required, the operation, or multiple operations can be performed using a single Map function, and the Reduce functionality is omitted.
- 2)In case merging of results is required, the operation is performed using the Map function and an aggregation is performed in the Reduce function.

In this paper[4], using given framework the modification may be completely performed on system side, without the need of reimplementing algorithms.

### 3.3.2 CG-Hadoop: Computational Geometry in MapReduce

**Authors:**Ahmed Eldawy , Yuan Li , Mohamed F. Mokbel, Ravi Janardan

**Publication :**ACM, 2013

Ahmed Eldawy , Yuan Li , Mohamed F. Mokbel, Ravi Janardan [3] is used CG-hadoop for fundamental computational geometry problem. In this paper use scalable and efficient MapReduced algorithm for Geometry Operation, Which are Polygon Union, Skyline, Convex hall , Farthest pair and closest pair. For each computational geometry operation.

CG-Hadoop used two versions:

- 1) Apache Hadoop system
- 2) SpatialHadoop system

The main idea behind all algorithms in CG-Hadoop is to take advantage of the divide and conquer nature of many computational geometry algorithms.

In the meantime, SpatialHadoop algorithms significantly outperform Hadoop algorithms and it is beneficial for indexing of spatial data.

In Paper[3], Extensive experimental results on a cluster of 25 machines of dataset up to 128GB show that CG-Hadoop achieves up to 29x and 260x better performance than traditional algorithms when using Hadoop and SpatialHadoop systems, respectively.

### 3.3.3 Hadoop GIS: A High Performance Spatial Data Warehousing System over MapReduce

**Authors:**Ablimit Aji1, Fusheng Wang, Hoang Vo1 Rubao Lee, Qiaoling Liu1 Xiaodong Zhang, Joel Saltz

**Publication:**ACM, 2013

Ablimit Aji1, Fusheng Wang, Hoang Vo1 Rubao Lee, Qiaoling Liu1 Xi-

aodong Zhang and Joel Saltz [2] are present scalable and high performance spatial data warehousing system, which is use for running multitype of large scale spatial queries on Hadoop. Hadoop GIS is used Hive for spatial query and after execution the experiment result is show that Hadoop GIS is giving fast response and high scalable for spatial quires. For experiment, Hadoop GIS is used as standalone library and integrated version of Hive. In this Paper[2],Exploring the results of analysis with complex queries like as global spatial pattern discovery, overlay of multiple set of object etc. Using map-reduce parallization, Hadoop GIS is partition a spatial. This paper, we mainly focus on a subset of cost-intensive queries which are commonly used in spatial warehousing applications[2]. This paper[2], comparing the performance of Hadoop GIS and DBMs-X on PI data for Join, Containment and Aggregation query and the results, join query is perform better in Hadoop GIS compare to DBMS-X, containment query has similar result for both and in aggregation query DBMS-X perform better task compare to Hadoop-GIS. Also measure the scalability of Hadoop-GIS. Using all the experiment result, It is easy to say Hadoop-GIS provide scalable and efficient solution for analytical spatial queries over large scale spatial data-sets[2].

### 3.3.4 A Demonstration of SpatialHadoop: An Efficient MapReduce Framework for Spatial Data

**Authors:**Ahmed Eldawy And Mohamed F. Mokbel

**Publication:**ACM, 2013

Ahmed Eldawy and Mohamed F. Mokbel [1] are demonstrate Spatial-Hadoop using MapReduce framework with for spatial dataset. In this paper, 20 nodes of Amazon EC2 cluster deploy real system prototype of SpatialHadoop. Amazon EC2 cluster is access by two front end and back end, and show how to handle spatial data file and fire query and visualize the result for selected file. SpatialHadoop architecture has mainly four layer which are language, storage, MapReduce and operation. For non technical user, the language layer use for high level language, Which provide spatial

data analysis. Storage layer is used two type of index(Local and Global) for storing information about spatial data index which is used to build R-tree, grid index or R+-tree. For accessing index file as input, MapReduce layer is used two component which are SpatialFileSplitter(for global index) and SpatialRecordReader (for local Index). The operations layer is consist a number of spatial operations such as KNN, range query and spatial join. Also implemented using the indexes and add new components in the MapReduce layer. So using the spatialhadoop architecture, this paper[1] is indicate that spatialhadoop is better than simple hadoop for spatial or non-spatial data.

### 3.3.5 Constructing Gazetteers from Volunteered Big Geo-Data based on Hadoop

**Author:** Song Gao, Linna Li, Wenwen Li, Krzysztof Janowicz, Yue Zhang

**Publication:**IEEE, 2014

Song Gao, Linna Li, Wenwen Li, Krzysztof Janowicz, Yue Zhang[12] are use space and place, which are associated through gazetteers for geospatial applications. In this paper[12] demonstrate a Big data-driven approach by mining VGI sources to create a crowd sourced gazetteer. Take a three examples of different types (point, polyline, polygon) of geographic features and extract, analyze and visualize these three features. Also present a provenance-based user reputation model for the trust evaluation. This semi-automatic construction of a crowd-sourced gazetteer can be facilitated by using high-performance computing resources because it involves the process of mining large-volumes of geospatial data. This paper is designed and established a Hadoop-based processing platform (GPHadoop) to show the promise of using VGI and cloud computing in gazetteer research and GI-Science in general. In particular, approach has the following three merits:

- Using the examples of the spatial join operation to the increasing number of points in different geographic scales, we demonstrate that the MapReduce based algorithm has a higher efficiency to process such Big Geo-Data analysis com-

pared to a traditional desktop PC-based analysis.

- The MapReduce algorithm of counting co-occurrence words makes it possible to rapidly extract parts of a place semantics and popular tags to characterize a place.
- The platform enables scalable geoprocessing workflow to solve geospatial problems based on the Hadoop ecosystem and Esri GIS tools, which make contributions in connecting GIS to a cloud computing environment for the next frontier of Big Geo-Data analytics.

### 3.3.6 Efficient Skyline Query Processing in Spatial Hadoop

**Author:**Dimitris Pertesis, Christos Doukeridis

**Publication:**IEEE, 2014

Dimitris Pertesis and Christos Doukeridis [11] are studies the problem of computing the skyline of a vast sized spatial data set in Spatial Hadoop, an extension of Hadoop that supports spatial operations efficiently. This paper is propose a scalable and efficient framework for skyline query processing that operates on top of SpatialHadoop, and can be parametrized by individual techniques related to filtering of candidate points as well as merging of local skyline sets. In next,this paper introduce two novel algorithms that follow the pattern of the framework and boost the performance of skyline query processing. The algorithms employ specific optimizations based on effective filtering and efficient merging, the combination of which is responsible for improved efficiency. This paper is also compare solution against the state-of-the-art skyline algorithm in SpatialHadoop. The results show that this, techniques are more efficient and out perform the competed or significantly, especially in the case of large skyline output size.



# Chapter 4

## Implementation

### 4.1 Geo-Processing Operation Using QGIS

First I tried to do the geoprocessing operation using GIS software like QGIS. This software is used for create, edit, visualise, analyse and publish geospatial information. It works with raster and vector data and geo-algorithm on it has been applied. The data has been converted in different formats like .shp to .csv, .csv to WKT, .shp to WKT, .shp to tiff, RGB to PTC, PTC to RGB etc and it also connected with database software. As shown in the following figure apply convexhull operation on a 10MB size of shape file using QGIS is very simple task is displayed. All the data which is in multiKB size can be easily opened and all the processes can be easily performed on it. But the size of data of about 50 MB or more can not be opened and sometime its create some problem in software because of which it gets hanged.

#### 4.1.1 Convexhull Operation on Shape File Using QGIS

In below Snapshot, display a convexhull operation on shape files which contain Point, Line and Polygon features individuals. Output of convexhull operation show boundary and using it cover whole area as shown below using QGIS software:

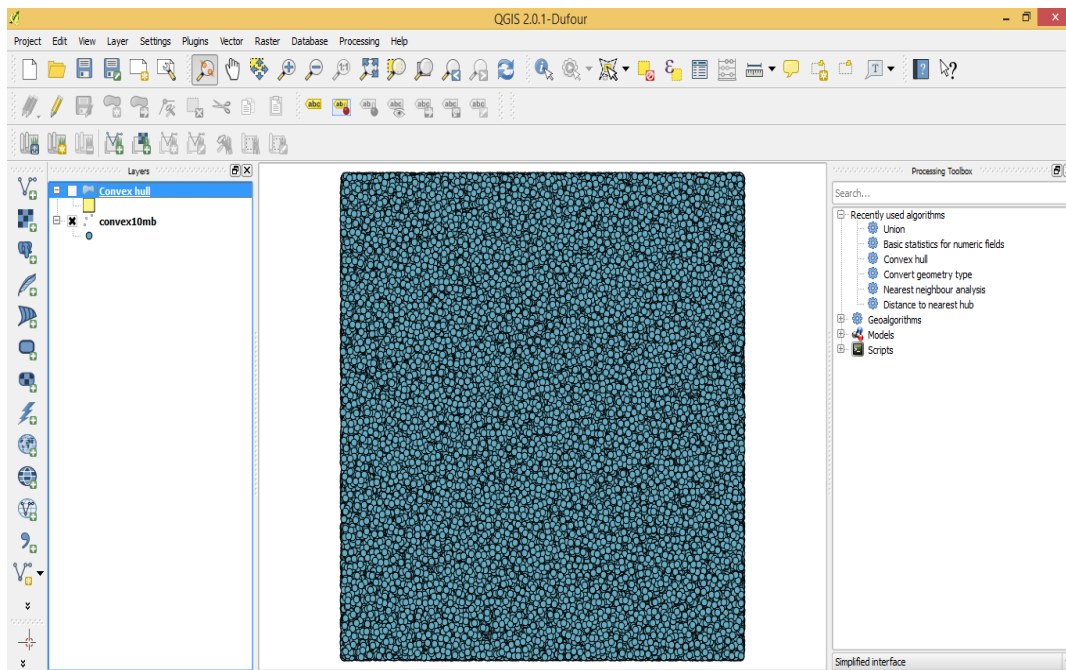


Figure 4.1: Display shape file using QGIS

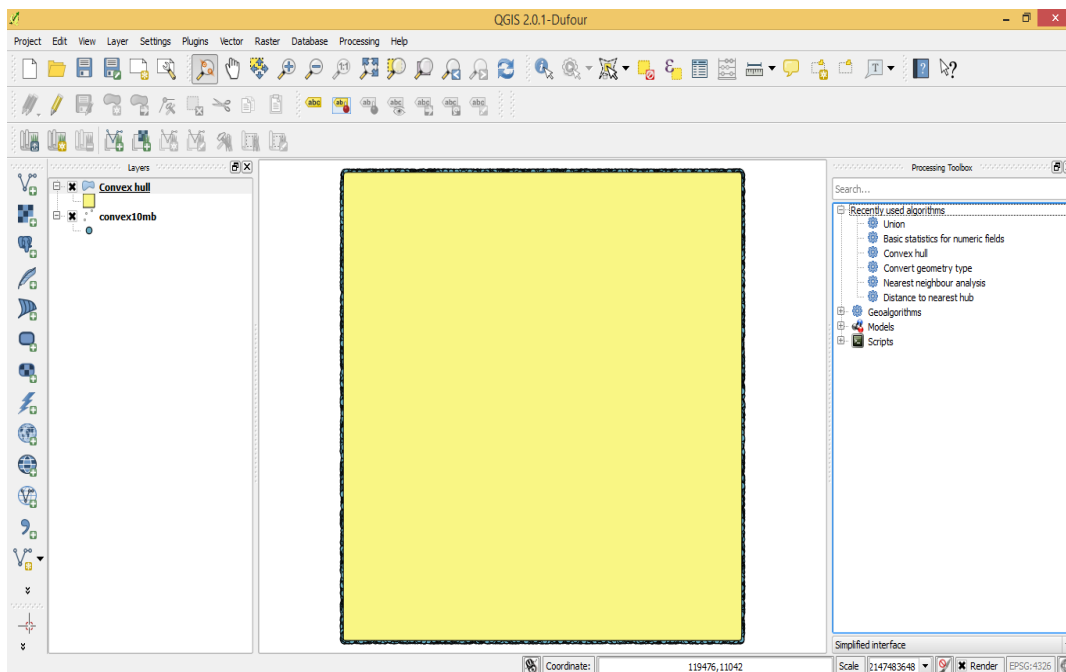


Figure 4.2: Display convexhull operation on shape file using QGIS

## 4.2 Steps for Setting up Multinode Hadoop

As mentioned before due to some limitations of GIS software, It was necessary to get the proper result so the work was switched over to hadoop. This project elaborates a proposed solution of Hadoop MapReduce framework. This implementation will be efficient and suitable for the problem of handling large data sets. We setup a Hadoop-1.2.1 in Ubuntu 14.10 (Utopic Unicorn) Operating System. Hadoop 1.2.1 is a latest version hadoop which supports YARN. YARN is a next version of a MapReduce classic programming model and uses REST APIs, which supports write/modify operations.

In this project, Hadoop-1.2.1 through multinode setup has been installed. To create cluster of multinode it uses the Oracle Virtual Box. One Master node and three slave nodes are used for crate multinode hadoop cluster.

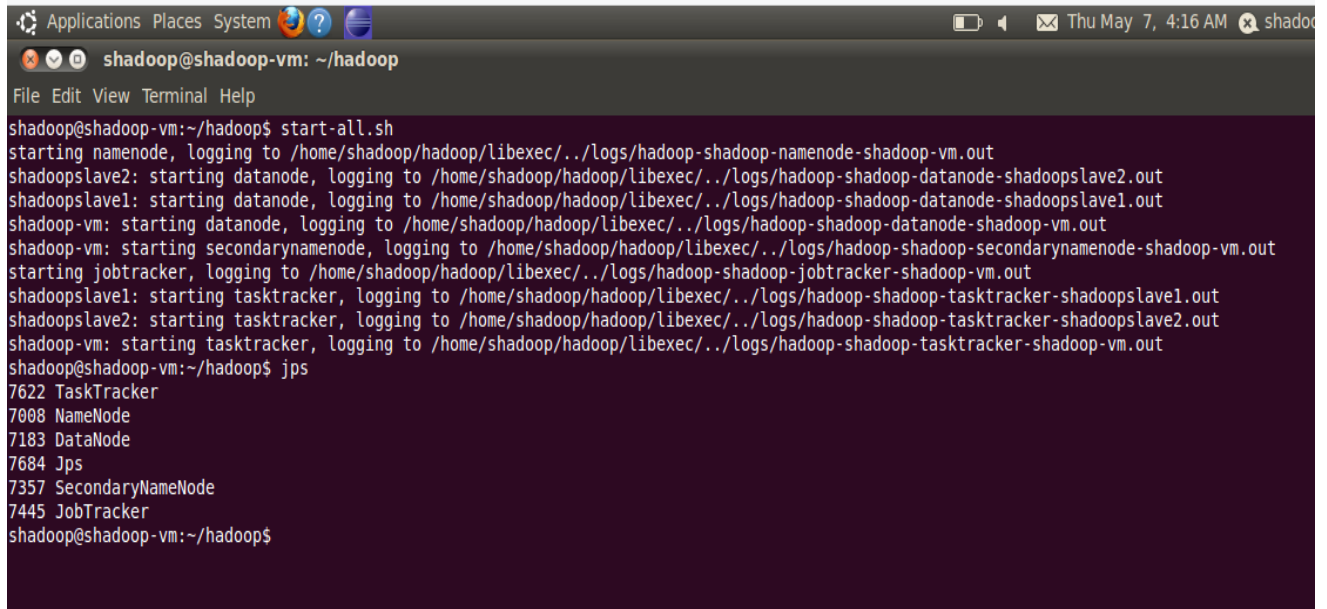
- Prerequisites:
  - Java must be installed
    - \* **Command:** apt-get update
    - \* **Command:** apt-get install default-jdk
  - Check Java version:
    - \* **Command:** java -version
  - SSH must be installed and must be running to use the Hadoop to manage remote Hadoop datanode
    - \* Create and setup SSH using below command:
    - \* `ssh-keygen -t rsa -P ''`
    - \* `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
- Installation of Hadoop
  - Fetch and Install Hadoop
    - \* Fetch Hadoop (Stable Version)

- *wget* `http://apache.tradefbit.com/pub/hadoop/common/current/hadoop-2.6.0.tar.gz`
- Extract File : `tar xzf hadoop-2.6.0.tar.gz`
- Move to local (Runing Directory) : `mv hadoop-2.6.0/usr/local/hadoop`
- Edit and Setup Configuration Files (`conf/*-site.xml`):
  - \* `home/.bashrc`
  - \* `conf/core-site.xml`
  - \* `conf/mapred-site.xml`
  - \* `conf/yarn-site.xml`
  - \* `conf/hdfs-site.xml`
- Formatting the HDFS filesystem via the NameNode
  - **Command:** `hdfs namenode -format`
- Starting the all services of node
  - Run the command `/bin/start-all.sh` on the machine you want the (primary) namenode to run on. This will bring up HDFS with the namenode, datanode, secondarynameNode and YARN with nodemanager, resorecemanager on node
- Using `jps` command we can check the working service on node. (which is shown in following snapshot)
- Stopping all services using `/bin/stop-all.sh` on the machine

Single node cluster setup using above step, moving to the next step in selecting the master node and slave node. After selection of node, starting the multinode cluster is done in following two steps. First, the HDFS daemons are started: the NameNode daemon is started on master, and datanode daemons are started on all

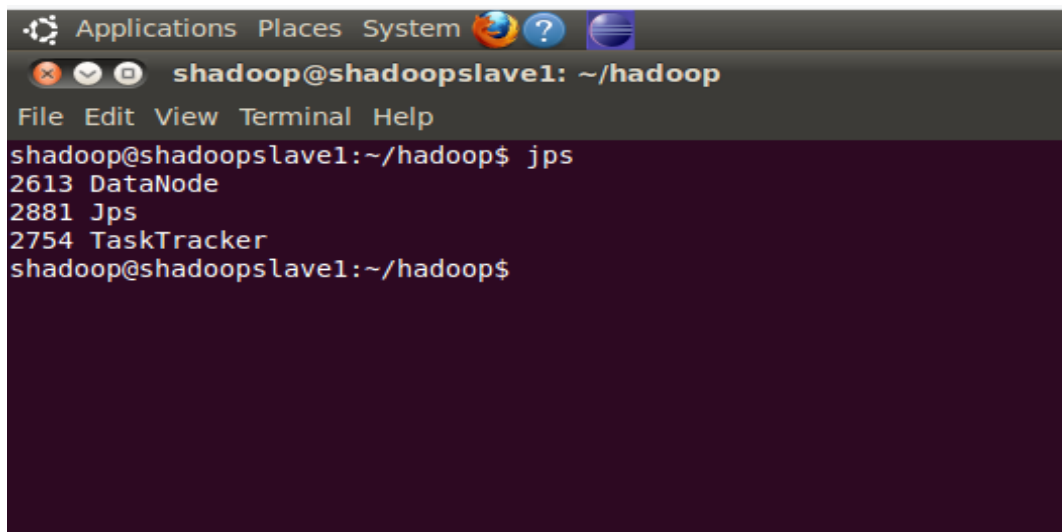
slaves. Typically masternode in the cluster is designated as the namenode and ResourceManager. The rest of the machines, which are slaves in the cluster act as both datanode and NodeManager.

### 4.2.1 Snapshot of MultiNode Hadoop Installation



```
shadoop@shadoop-vm: ~/hadoop
shadoop@shadoop-vm:~/hadoop$ start-all.sh
starting namenode, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-namenode-shadoop-vm.out
shadoopslave2: starting datanode, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-datanode-shadoopslave2.out
shadoopslave1: starting datanode, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-datanode-shadoopslave1.out
shadoop-vm: starting datanode, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-datanode-shadoop-vm.out
shadoop-vm: starting secondarynamenode, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-secondarynamenode-shadoop-vm.out
starting jobtracker, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-jobtracker-shadoop-vm.out
shadoopslave1: starting tasktracker, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-tasktracker-shadoopslave1.out
shadoopslave2: starting tasktracker, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-tasktracker-shadoopslave2.out
shadoop-vm: starting tasktracker, logging to /home/shadoop/hadoop/libexec/../logs/hadoop-shadoop-tasktracker-shadoop-vm.out
shadoop@shadoop-vm:~/hadoop$ jps
7622 TaskTracker
7008 NameNode
7183 DataNode
7684 Jps
7357 SecondaryNameNode
7445 JobTracker
shadoop@shadoop-vm:~/hadoop$
```

Figure 4.3: Snapshot of starting all service using jps command on masternode



```
shadoop@shadoopslave1: ~/hadoop
shadoop@shadoopslave1:~/hadoop$ jps
2613 DataNode
2881 Jps
2754 TaskTracker
shadoop@shadoopslave1:~/hadoop$
```

Figure 4.4: Snapshot of starting all service using jps command on slavenode1

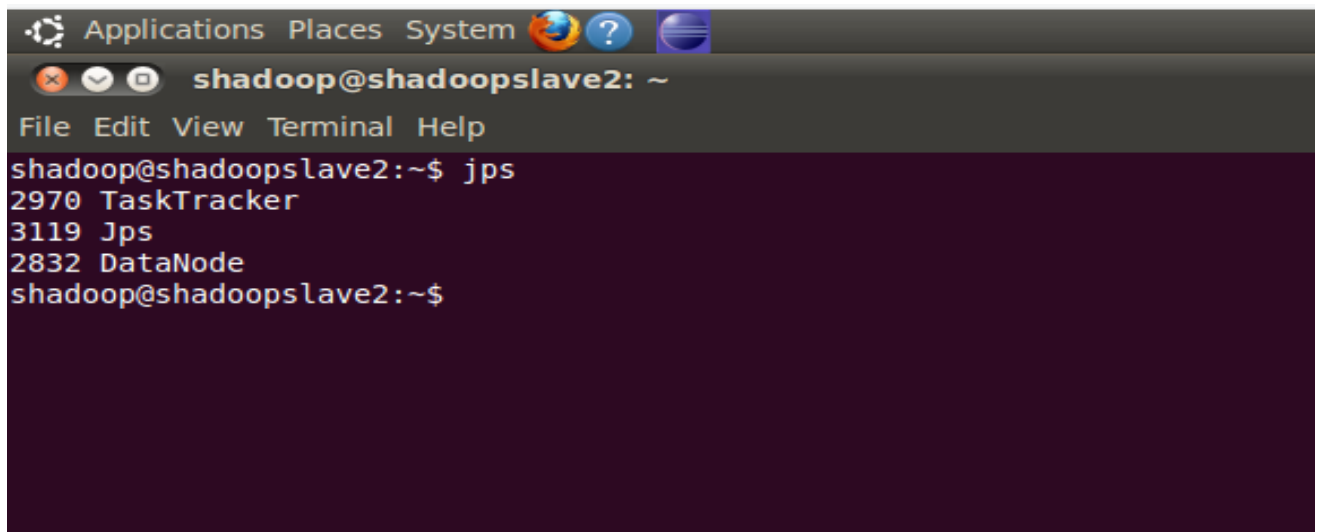


Figure 4.5: Snapshot of starting all service using jps command on slavenode2

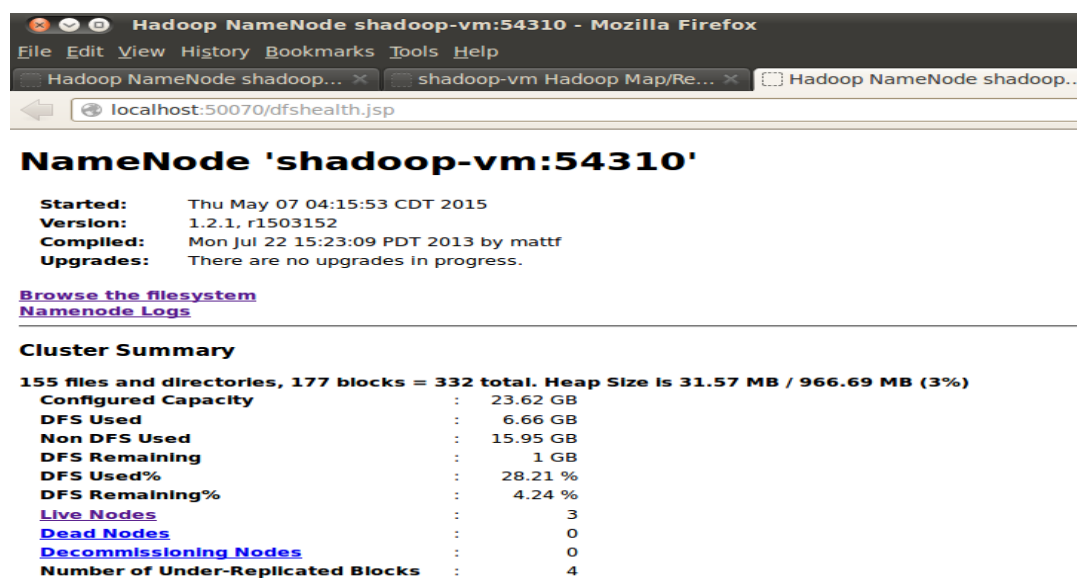


Figure 4.6: Snapshot of information about cluster

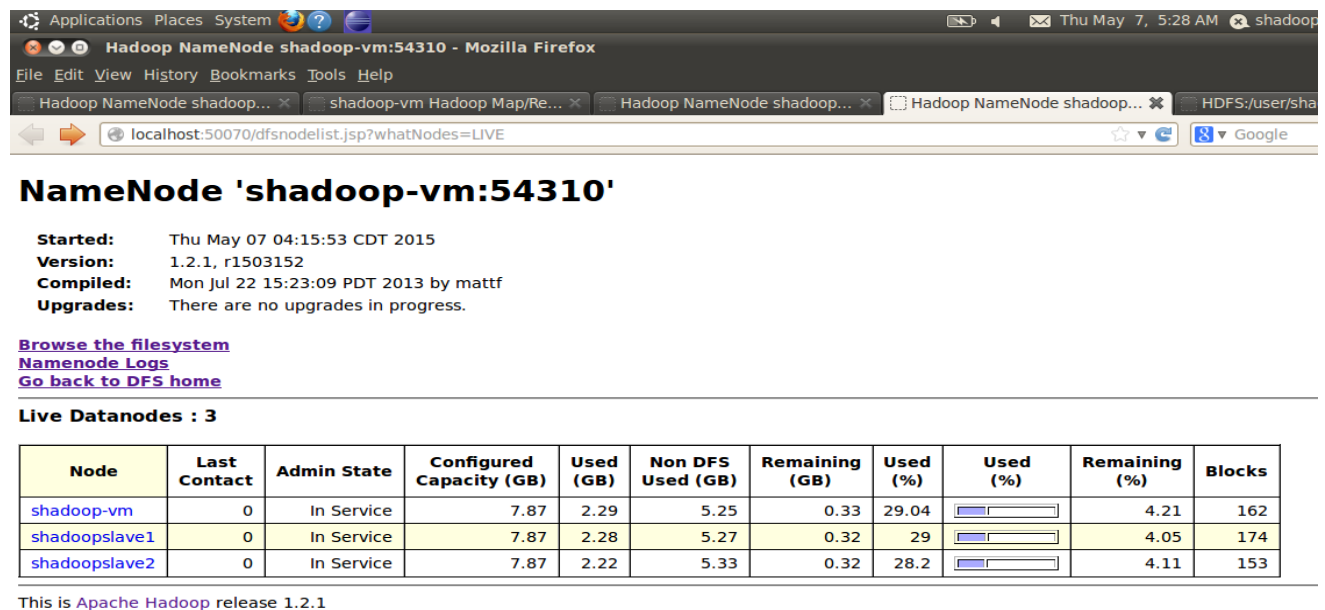


Figure 4.7: Snapshot of namenode information

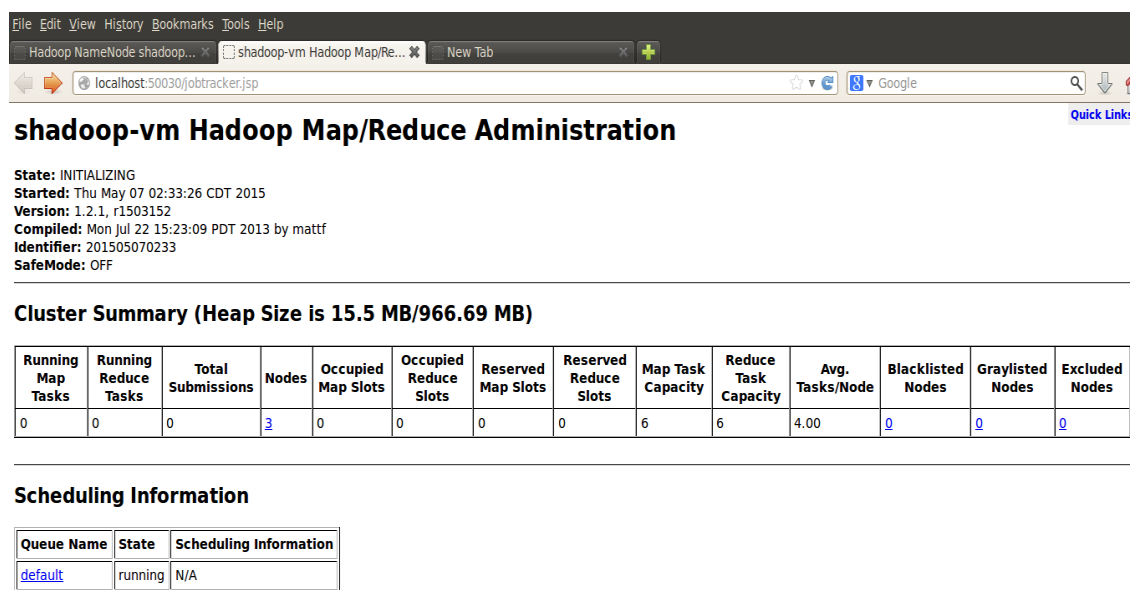


Figure 4.8: Snapshot of jobtacker information



### 4.2.2 Install and Configure Spatial Hadoop on top of Multi-node Hadoop Cluster

There was an indexing problem in hadoop for a spatial data, so to overcome this limitation we switch over to spatial hadoop. To install SpatialHadoop, the first step is to download the binaries as a compressed file and decompress it to the local disk. Then, the installation is configured by editing some configuration files and a directory in to hadoop folder. Also download and compile a Ant and IVY tools for managing spatialhadoop project dependency. After that, the SpatialHadoop server is started and do some operations as shown in below section. The steps are available on the official web page of SpatialHadoop (<http://spatialhadoop.cs.umn.edu/>).

### 4.3 Shape File Conversion

In spatial Hadoop, before starting a distribution of shape file, we have to convert it into specific format so that data or content of a file can not be lost and after distribution we can get accurate result. So that Shape file is converted into WKT and WKB format of OGC standard as a result of which data in the form of international standard format can be obtained. For conversion of file, Netbean 8.0 and maven build automation tool in Java programming language is used. Using shapefilereader and geotool library in program, we can see and read the shape file content. Vivid solutions library/JAR is used for converting the shape file into WKT format. The following table display the libraries/jar among which the first one is of vivid solutions and the others are of Geo tools. The following table also gives information of their versions and reference sites from where they have taken.

Library/JAR	Version	Name Of The Site
jts	1.1.3	<a href="http://www.oracle.com">www.oracle.com</a>
gt-shapefile	2.7-M2	<a href="http://www.osgeo.org">www.osgeo.org</a>
gt-swing	2.7-M2	<a href="http://www.osgeo.org">www.osgeo.org</a>
gt-api	2.7-M2	<a href="http://www.osgeo.org">www.osgeo.org</a>
gt-main	2.7-M2	<a href="http://www.osgeo.org">www.osgeo.org</a>
gt-render	2.7-M2	<a href="http://www.osgeo.org">www.osgeo.org</a>

Table 4.1: Details of the Library/JAR

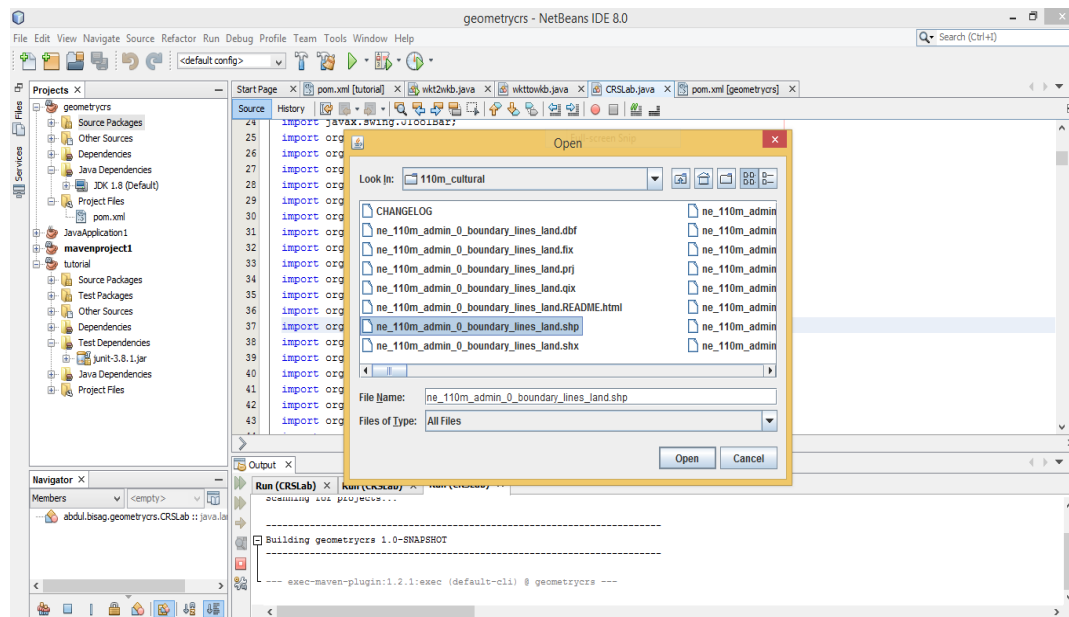


Figure 4.9: Snapshot of fetching file

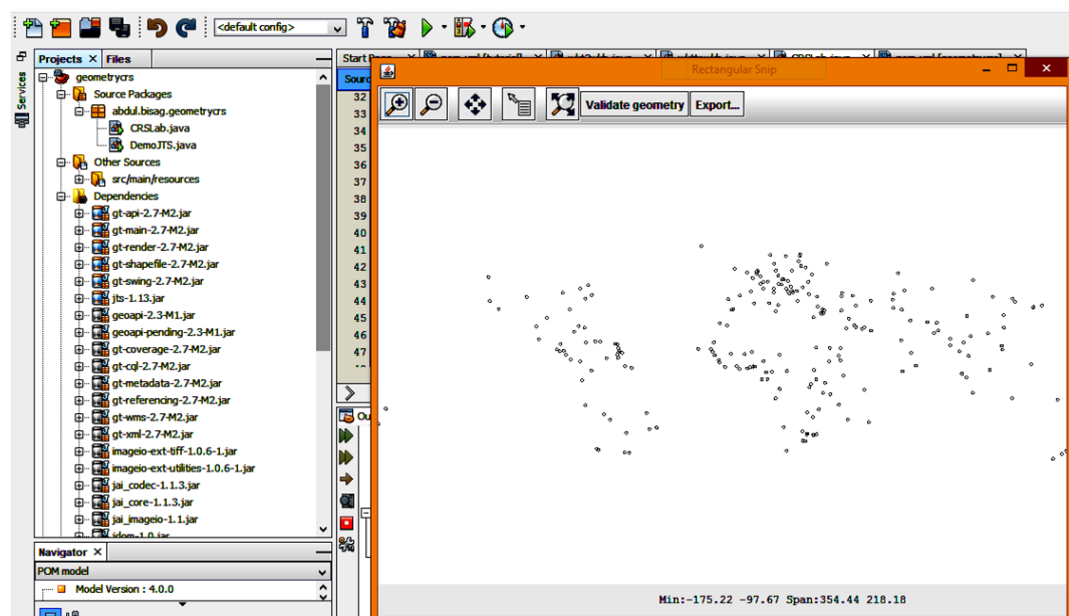


Figure 4.10: Snapshot of display shape file of Point

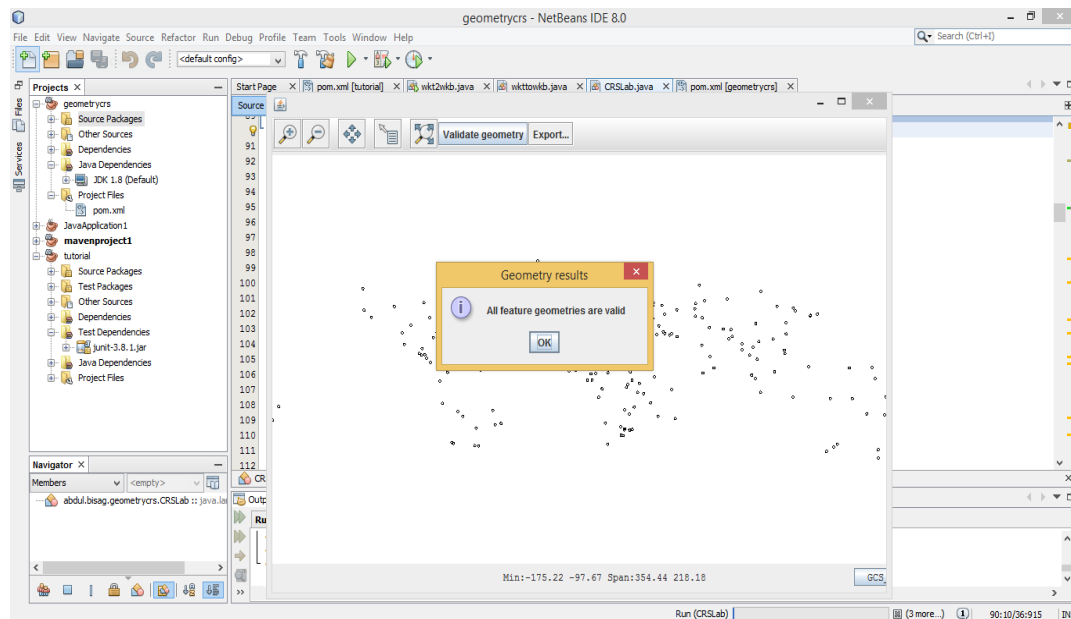


Figure 4.11: Snapshot of validation of file

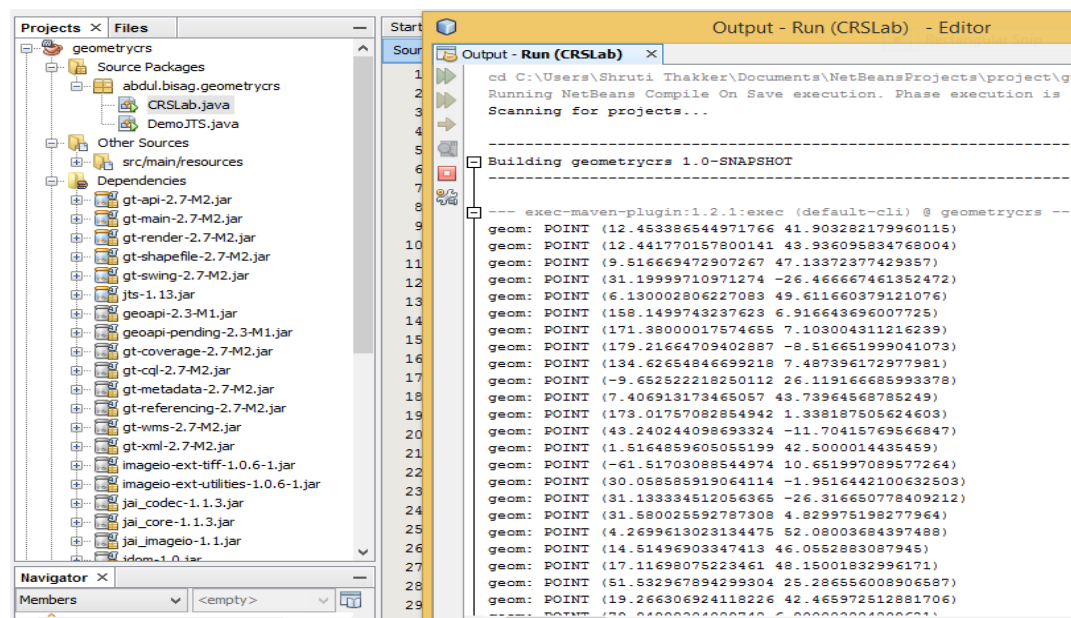


Figure 4.12: Snapshot of display shape file in WKT format

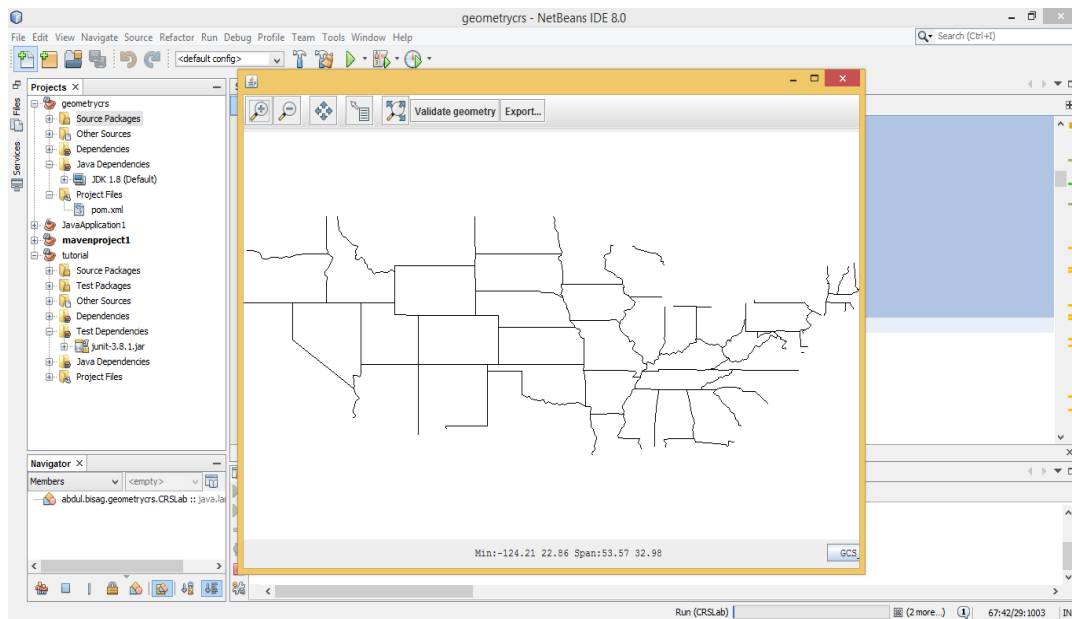


Figure 4.13: Snapshot of display shape file of multilinestring

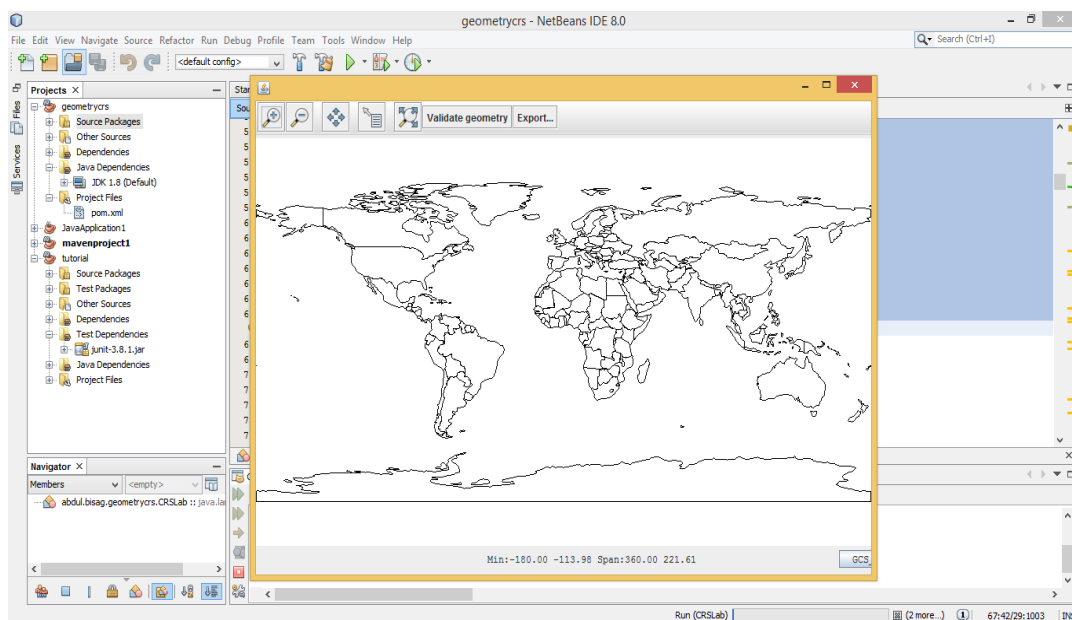


Figure 4.14: Snapshot of display shape file of multipolygon

## 4.4 How conversion of shape file will be used in distribution of file?

If the data is in the shape file format, we can not distribute into more than one file so we can not directly use in Spatial Hadoop and without distribution of a file, Spatial Hadoop is not beneficial for large dataset. So first we need to convert it into specific format. WKT and WKB is the format, which will be used to distribute file in more than one part and used into distributed processing frameworks.

If we convert shape file into WKB format than all data is in the byte format. So we can break the file from any point without any lose of data. In WKB, we can divide the file in any number and it is the main advantage of this format.

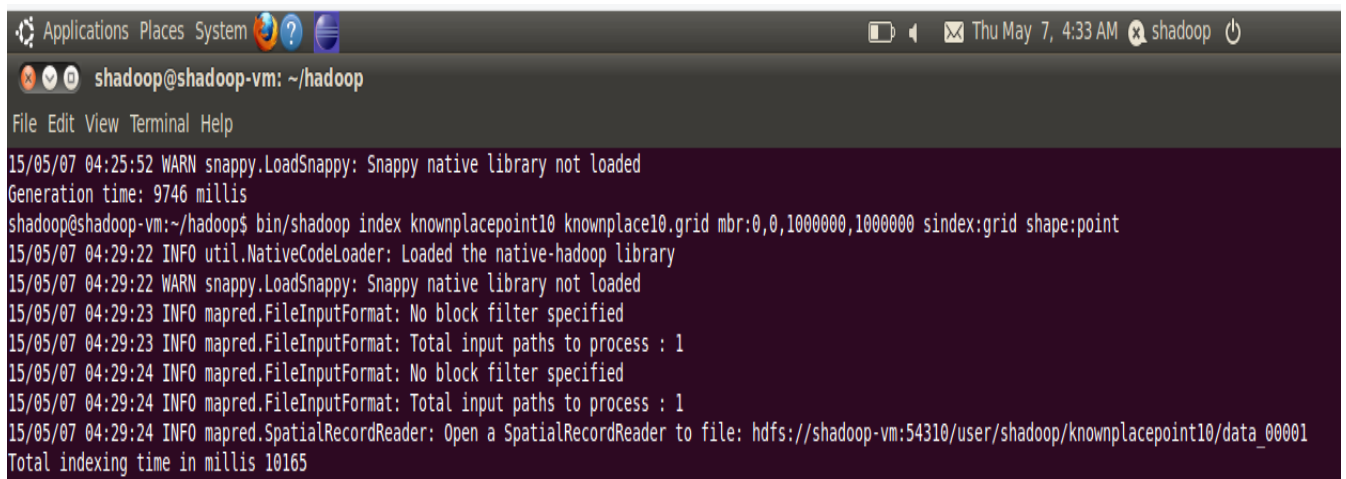
If we convert shape file into WKT format than all data is in the text format. So we can break the file based on the feature which is stored in to the file without any lose of any data. In WKT, we can divide the file based on number of features like lines, polygons which is used into file as vector data an then we can divide this file into two files. In one file it contains only line coordinator values and the other file contains only polygon coordinator values.

# Chapter 5

## Result and Discussion

### 5.1 Result

In this project work, A file has been generated on spatial hadoop, After that the grid Indexing has been performed and applied convexhull algorithm on it. For this purpose multi size of data has been used and analyses of the result has been displayed in result section.

A terminal window titled 'shadoop@shadoop-vm: ~/hadoop' with a menu bar (File, Edit, View, Terminal, Help). The terminal output shows a warning about the Snappy native library not loading, followed by the command 'bin/shadoop index knownplacepoint10 knownplace10.grid mbr:0,0,1000000,1000000 sindex:grid shape:point'. The output includes several INFO messages from mapred.FileInputFormat and mapred.SpatialRecordReader, and a final message 'Total indexing time in millis 10165'.

```
15/05/07 04:25:52 WARN snappy.LoadSnappy: Snappy native library not loaded
Generation time: 9746 millis
shadoop@shadoop-vm:~/hadoop$ bin/shadoop index knownplacepoint10 knownplace10.grid mbr:0,0,1000000,1000000 sindex:grid shape:point
15/05/07 04:29:22 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:29:22 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:29:23 INFO mapred.FileInputFormat: No block filter specified
15/05/07 04:29:23 INFO mapred.FileInputFormat: Total input paths to process : 1
15/05/07 04:29:24 INFO mapred.FileInputFormat: No block filter specified
15/05/07 04:29:24 INFO mapred.FileInputFormat: Total input paths to process : 1
15/05/07 04:29:24 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplacepoint10/data_00001
Total indexing time in millis 10165
```

Figure 5.1: Snapshot of generating and indexing of 10MB file using SpatialHadoop

```

shadoop@shadoop-vm: ~/hadoop$ bin/shadoop convexhull knownplace10.grid knownplace10.out
15/05/07 04:34:48 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplace10.grid/data_00001
15/05/07 04:34:48 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:34:48 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:34:49 INFO spatialHadoop.OperationsParams: Autodetected shape 'point' for input '62289.267652563765,18412.505494383848'
15/05/07 04:34:50 INFO operations.ConvexHull: Processing 1 out of 1 partition
15/05/07 04:34:50 INFO mapred.FileInputFormat: Spatial filter function matched with 1 cells
15/05/07 04:34:52 INFO mapred.JobClient: Running job: job_201505070416_0001
15/05/07 04:34:53 INFO mapred.JobClient: map 0% reduce 0%
15/05/07 04:35:09 INFO mapred.JobClient: map 98% reduce 0%
15/05/07 04:35:10 INFO mapred.JobClient: map 100% reduce 0%
15/05/07 04:35:20 INFO mapred.JobClient: map 100% reduce 33%
15/05/07 04:35:22 INFO mapred.JobClient: map 100% reduce 100%
15/05/07 04:35:25 INFO mapred.JobClient: Job complete: job_201505070416_0001
15/05/07 04:35:25 INFO mapred.JobClient: Counters: 30
15/05/07 04:35:25 INFO mapred.JobClient:   Job Counters
15/05/07 04:35:25 INFO mapred.JobClient:     Launched reduce tasks=1
15/05/07 04:35:25 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=18104
15/05/07 04:35:25 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/05/07 04:35:25 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/05/07 04:35:25 INFO mapred.JobClient:     Launched map tasks=1
15/05/07 04:35:25 INFO mapred.JobClient:     Data-local map tasks=1
15/05/07 04:35:25 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=11415
15/05/07 04:35:25 INFO mapred.JobClient:   File Input Format Counters
15/05/07 04:35:25 INFO mapred.JobClient:     Bytes Read=10485847
15/05/07 04:35:25 INFO mapred.JobClient:   File Output Format Counters
15/05/07 04:35:25 INFO mapred.JobClient:     Bytes Written=1157
15/05/07 04:35:25 INFO mapred.JobClient:   FileSystemCounters
15/05/07 04:35:25 INFO mapred.JobClient:     FILE_BYTES_READ=2322
15/05/07 04:35:25 INFO mapred.JobClient:     HDFS_BYTES_READ=10485965
15/05/07 04:35:25 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=120759
15/05/07 04:35:25 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=1157

```

Figure 5.2: Snapshot of apply convexhull operation on 10MB file using SpatialHadoop-1

```

shadoop@shadoop-vm: ~/hadoop$ bin/shadoop convexhull knownplace10.grid knownplace10.out
15/05/07 04:35:25 INFO mapred.JobClient: Launched map tasks=1
15/05/07 04:35:25 INFO mapred.JobClient: Data-local map tasks=1
15/05/07 04:35:25 INFO mapred.JobClient: SLOTS_MILLIS_REDUCES=11415
15/05/07 04:35:25 INFO mapred.JobClient: File Input Format Counters
15/05/07 04:35:25 INFO mapred.JobClient:   Bytes Read=10485847
15/05/07 04:35:25 INFO mapred.JobClient: File Output Format Counters
15/05/07 04:35:25 INFO mapred.JobClient:   Bytes Written=1157
15/05/07 04:35:25 INFO mapred.JobClient: FileSystemCounters
15/05/07 04:35:25 INFO mapred.JobClient:   FILE_BYTES_READ=2322
15/05/07 04:35:25 INFO mapred.JobClient:   HDFS_BYTES_READ=10485965
15/05/07 04:35:25 INFO mapred.JobClient:   FILE_BYTES_WRITTEN=120759
15/05/07 04:35:25 INFO mapred.JobClient:   HDFS_BYTES_WRITTEN=1157
15/05/07 04:35:25 INFO mapred.JobClient: Map-Reduce Framework
15/05/07 04:35:25 INFO mapred.JobClient:   Map output materialized bytes=1158
15/05/07 04:35:25 INFO mapred.JobClient:   Map input records=288546
15/05/07 04:35:25 INFO mapred.JobClient:   Reduce shuffle bytes=1158
15/05/07 04:35:25 INFO mapred.JobClient:   Spilled Records=192
15/05/07 04:35:25 INFO mapred.JobClient:   Map output bytes=4616736
15/05/07 04:35:25 INFO mapred.JobClient:   Total committed heap usage (bytes)=177016832
15/05/07 04:35:25 INFO mapred.JobClient:   CPU time spent (ms)=8080
15/05/07 04:35:25 INFO mapred.JobClient:   Map input bytes=10485743
15/05/07 04:35:25 INFO mapred.JobClient:   SPLIT_RAW_BYTES=118
15/05/07 04:35:25 INFO mapred.JobClient:   Combine input records=288546
15/05/07 04:35:25 INFO mapred.JobClient:   Reduce input records=64
15/05/07 04:35:25 INFO mapred.JobClient:   Reduce input groups=1
15/05/07 04:35:25 INFO mapred.JobClient:   Combine output records=64
15/05/07 04:35:25 INFO mapred.JobClient:   Physical memory (bytes) snapshot=226471936
15/05/07 04:35:25 INFO mapred.JobClient:   Reduce output records=32
15/05/07 04:35:25 INFO mapred.JobClient:   Virtual memory (bytes) snapshot=749342720
15/05/07 04:35:25 INFO mapred.JobClient:   Map output records=288546
Total time: 35652 millis

```

Figure 5.3: Snapshot of apply convexhull operation on 10MB file using SpatialHadoop-2



```

shadoop@shadoop-vm:~/hadoop$ bin/shadoop generate knownplacepoint50 mbr:0,0,100000,100000 size:50.mb shape:point
15/05/07 04:38:44 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:38:44 WARN snappy.LoadSnappy: Snappy native library not loaded
Generation time: 49062 millis
shadoop@shadoop-vm:~/hadoop$ bin/shadoop index knownplacepoint50 knownplace50.grid mbr:0,0,100000,100000 index:grid shape:point
15/05/07 04:40:31 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:40:31 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:40:32 INFO mapred.FileInputFormat: No block filter specified
15/05/07 04:40:32 INFO mapred.FileInputFormat: Total input paths to process : 1
15/05/07 04:40:32 INFO mapred.FileInputFormat: No block filter specified
15/05/07 04:40:32 INFO mapred.FileInputFormat: Total input paths to process : 1
15/05/07 04:40:32 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplacepoint50/data_00001
Total indexing time in millis 44142
shadoop@shadoop-vm:~/hadoop$ bin/shadoop convexhull knownplace50.grid knownplace50.out
15/05/07 04:42:08 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplace50.grid/data_00001
15/05/07 04:42:08 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:42:08 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:42:11 INFO spatialHadoop.OperationsParams: Autodetected shape 'point' for input '82170.51422791573,88839.77590144158'
15/05/07 04:42:13 INFO operations.ConvexHull: Processing 1 out of 1 partition
15/05/07 04:42:13 INFO mapred.FileInputFormat: Spatial filter function matched with 1 cells
15/05/07 04:42:14 INFO mapred.JobClient: Running job: job_201505070416_0002
15/05/07 04:42:15 INFO mapred.JobClient: map 0% reduce 0%
15/05/07 04:42:32 INFO mapred.JobClient: map 16% reduce 0%
15/05/07 04:42:35 INFO mapred.JobClient: map 26% reduce 0%
15/05/07 04:42:39 INFO mapred.JobClient: map 39% reduce 0%
15/05/07 04:42:42 INFO mapred.JobClient: map 53% reduce 0%
15/05/07 04:42:45 INFO mapred.JobClient: map 61% reduce 0%
15/05/07 04:42:48 INFO mapred.JobClient: map 76% reduce 0%
15/05/07 04:42:51 INFO mapred.JobClient: map 87% reduce 0%
15/05/07 04:42:54 INFO mapred.JobClient: map 97% reduce 0%
15/05/07 04:42:55 INFO mapred.JobClient: map 100% reduce 0%
15/05/07 04:43:05 INFO mapred.JobClient: map 100% reduce 33%

```

Figure 5.4: Snapshot of generating and indexing of 50MB file using SpatialHadoop

```

Applications Places System
shadoop@shadoop-vm: ~/hadoop
File Edit View Terminal Help
15/05/07 04:43:09 INFO mapred.JobClient: Launched map tasks=1
15/05/07 04:43:09 INFO mapred.JobClient: Data-local map tasks=1
15/05/07 04:43:09 INFO mapred.JobClient: SLOTS_MILLIS_REDUCES=11315
15/05/07 04:43:09 INFO mapred.JobClient: File Input Format Counters
15/05/07 04:43:09 INFO mapred.JobClient: Bytes Read=52428898
15/05/07 04:43:09 INFO mapred.JobClient: File Output Format Counters
15/05/07 04:43:09 INFO mapred.JobClient: Bytes Written=1383
15/05/07 04:43:09 INFO mapred.JobClient: FileSystemCounters
15/05/07 04:43:09 INFO mapred.JobClient: FILE_BYTES_READ=4488
15/05/07 04:43:09 INFO mapred.JobClient: HDFS_BYTES_READ=52429016
15/05/07 04:43:09 INFO mapred.JobClient: FILE_BYTES_WRITTEN=122457
15/05/07 04:43:09 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=1383
15/05/07 04:43:09 INFO mapred.JobClient: Map-Reduce Framework
15/05/07 04:43:09 INFO mapred.JobClient: Map output materialized bytes=690
15/05/07 04:43:09 INFO mapred.JobClient: Map input records=1442618
15/05/07 04:43:09 INFO mapred.JobClient: Reduce shuffle bytes=690
15/05/07 04:43:09 INFO mapred.JobClient: Spilled Records=285
15/05/07 04:43:09 INFO mapred.JobClient: Map output bytes=23081888
15/05/07 04:43:09 INFO mapred.JobClient: Total committed heap usage (bytes)=219021312
15/05/07 04:43:09 INFO mapred.JobClient: CPU time spent (ms)=27520
15/05/07 04:43:09 INFO mapred.JobClient: Map input bytes=52428791
15/05/07 04:43:09 INFO mapred.JobClient: SPLIT_RAW_BYTES=118
15/05/07 04:43:09 INFO mapred.JobClient: Combine input records=1442827
15/05/07 04:43:09 INFO mapred.JobClient: Reduce input records=38
15/05/07 04:43:09 INFO mapred.JobClient: Reduce input groups=1
15/05/07 04:43:09 INFO mapred.JobClient: Combine output records=247
15/05/07 04:43:09 INFO mapred.JobClient: Physical memory (bytes) snapshot=265052160
15/05/07 04:43:09 INFO mapred.JobClient: Reduce output records=38
15/05/07 04:43:09 INFO mapred.JobClient: Virtual memory (bytes) snapshot=749907968
15/05/07 04:43:09 INFO mapred.JobClient: Map output records=1442618
Total time: 57764 millis

```

Figure 5.5: Snapshot of apply convexhull operation on 50MB file using SpatialHadoop

```

shadoop@shadoop-vm: ~/hadoop
File Edit View Terminal Help
shadoop@shadoop-vm:~/hadoop$ bin/shadoop generate knownplacepoint100 mbr:0,0,100000,100000 size:100.mb shape:point
15/05/07 04:47:22 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:47:22 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:47:23 INFO core.SpatialSite: Partitioning file into 1 cells
15/05/07 04:47:25 INFO mapred.JobClient: Running job: job_201505070416_0003
15/05/07 04:47:26 INFO mapred.JobClient: map 0% reduce 0%
15/05/07 04:47:43 INFO mapred.JobClient: map 9% reduce 0%
15/05/07 04:47:46 INFO mapred.JobClient: map 13% reduce 0%
15/05/07 04:47:47 INFO mapred.JobClient: map 15% reduce 0%
15/05/07 04:47:50 INFO mapred.JobClient: map 21% reduce 0%
15/05/07 04:47:53 INFO mapred.JobClient: map 27% reduce 0%
15/05/07 04:47:56 INFO mapred.JobClient: map 33% reduce 0%
15/05/07 04:47:59 INFO mapred.JobClient: map 39% reduce 0%
15/05/07 04:48:02 INFO mapred.JobClient: map 45% reduce 0%
15/05/07 04:48:05 INFO mapred.JobClient: map 50% reduce 0%
15/05/07 04:48:06 INFO mapred.JobClient: map 53% reduce 0%
15/05/07 04:48:08 INFO mapred.JobClient: map 59% reduce 0%
15/05/07 04:48:09 INFO mapred.JobClient: map 62% reduce 0%
15/05/07 04:48:12 INFO mapred.JobClient: map 68% reduce 0%
15/05/07 04:48:15 INFO mapred.JobClient: map 73% reduce 0%
15/05/07 04:48:18 INFO mapred.JobClient: map 75% reduce 0%
15/05/07 04:48:19 INFO mapred.JobClient: map 78% reduce 0%
15/05/07 04:48:22 INFO mapred.JobClient: map 79% reduce 0%
15/05/07 04:48:26 INFO mapred.JobClient: map 81% reduce 0%
15/05/07 04:48:29 INFO mapred.JobClient: map 83% reduce 0%
15/05/07 04:48:32 INFO mapred.JobClient: map 85% reduce 0%
15/05/07 04:48:35 INFO mapred.JobClient: map 88% reduce 0%
15/05/07 04:48:38 INFO mapred.JobClient: map 90% reduce 0%
15/05/07 04:48:41 INFO mapred.JobClient: map 93% reduce 0%
15/05/07 04:48:44 INFO mapred.JobClient: map 95% reduce 0%
15/05/07 04:48:47 INFO mapred.JobClient: map 98% reduce 0%
15/05/07 04:48:50 INFO mapred.JobClient: map 100% reduce 0%

```

Figure 5.6: Snapshot of generating 100MB file using SpatialHadoop-1

```

shadoop@shadoop-vm: ~/hadoop
File Edit View Terminal Help
15/05/07 04:48:41 INFO mapred.JobClient: map 93% reduce 0%
15/05/07 04:48:44 INFO mapred.JobClient: map 95% reduce 0%
15/05/07 04:48:47 INFO mapred.JobClient: map 98% reduce 0%
15/05/07 04:48:50 INFO mapred.JobClient: map 100% reduce 0%
15/05/07 04:48:53 INFO mapred.JobClient: Job complete: job_201505070416_0003
15/05/07 04:48:53 INFO mapred.JobClient: Counters: 20
15/05/07 04:48:53 INFO mapred.JobClient:   Job Counters
15/05/07 04:48:53 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=131849
15/05/07 04:48:53 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/05/07 04:48:53 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/05/07 04:48:53 INFO mapred.JobClient:     Launched map tasks=2
15/05/07 04:48:53 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
15/05/07 04:48:53 INFO mapred.JobClient:   File Input Format Counters
15/05/07 04:48:53 INFO mapred.JobClient:     Bytes Read=0
15/05/07 04:48:53 INFO mapred.JobClient:   File Output Format Counters
15/05/07 04:48:53 INFO mapred.JobClient:     Bytes Written=104857782
15/05/07 04:48:53 INFO mapred.JobClient:   FileSystemCounters
15/05/07 04:48:53 INFO mapred.JobClient:     FILE_BYTES_READ=208
15/05/07 04:48:53 INFO mapred.JobClient:     HDFS_BYTES_READ=154
15/05/07 04:48:53 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=118381
15/05/07 04:48:53 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=104857782
15/05/07 04:48:53 INFO mapred.JobClient:   Map-Reduce Framework
15/05/07 04:48:53 INFO mapred.JobClient:     Map input records=2885303
15/05/07 04:48:53 INFO mapred.JobClient:     Physical memory (bytes) snapshot=79626240
15/05/07 04:48:53 INFO mapred.JobClient:     Spilled Records=0
15/05/07 04:48:53 INFO mapred.JobClient:     CPU time spent (ms)=91750
15/05/07 04:48:53 INFO mapred.JobClient:     Total committed heap usage (bytes)=32505856
15/05/07 04:48:53 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=748572672
15/05/07 04:48:53 INFO mapred.JobClient:     Map input bytes=104857548
15/05/07 04:48:53 INFO mapred.JobClient:     Map output records=2885303
15/05/07 04:48:53 INFO mapred.JobClient:     SPLIT_RAW_BYTES=154

```

Figure 5.7: Snapshot of generating 100MB file using SpatialHadoop-2

```

shadoop@shadoop-vm: ~/hadoop
shadoop@shadoop-vm:~/hadoop$ bin/shadoop index knownplacepoint100 knownplace100.grid mbr:0,0,1000000,1000000 index:grid shape:point
15/05/07 04:49:31 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:49:31 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:49:32 INFO mapred.FileInputFormat: No block filter specified
15/05/07 04:49:32 INFO mapred.FileInputFormat: Total input paths to process : 2
15/05/07 04:49:32 INFO mapred.FileInputFormat: No block filter specified
15/05/07 04:49:32 INFO mapred.FileInputFormat: Total input paths to process : 2
15/05/07 04:49:32 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplacepoint100/part-00000_data_00001
15/05/07 04:50:34 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplacepoint100/part-00001_data_00001
Total indexing time in millis 91934
shadoop@shadoop-vm:~/hadoop$ bin/shadoop convexhull knownplace100.grid knownplace100.out
15/05/07 04:52:01 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplace100.grid/data_00001
15/05/07 04:52:01 INFO util.NativeCodeLoader: Loaded the native-hadoop library
15/05/07 04:52:01 WARN snappy.LoadSnappy: Snappy native library not loaded
15/05/07 04:52:03 INFO mapred.SpatialRecordReader: Open a SpatialRecordReader to file: hdfs://shadoop-vm:54310/user/shadoop/knownplace100.grid/data_00002_1
15/05/07 04:52:04 INFO spatialHadoop.OperationsParams: Autodetected shape 'point' for input '17549.03119590322,17473.484220037317'
15/05/07 04:52:05 INFO operations.ConvexHull: Processing 2 out of 2 partition
15/05/07 04:52:05 INFO mapred.FileInputFormat: Spatial filter function matched with 2 cells
15/05/07 04:52:06 INFO mapred.JobClient: Running job: job_201505070416_0004
15/05/07 04:52:07 INFO mapred.JobClient: map 0% reduce 0%
15/05/07 04:52:25 INFO mapred.JobClient: map 11% reduce 0%
15/05/07 04:52:28 INFO mapred.JobClient: map 21% reduce 0%
15/05/07 04:52:31 INFO mapred.JobClient: map 32% reduce 0%
15/05/07 04:52:34 INFO mapred.JobClient: map 42% reduce 0%
15/05/07 04:52:37 INFO mapred.JobClient: map 54% reduce 0%
15/05/07 04:52:40 INFO mapred.JobClient: map 66% reduce 0%
15/05/07 04:52:43 INFO mapred.JobClient: map 72% reduce 0%
15/05/07 04:52:44 INFO mapred.JobClient: map 79% reduce 0%
15/05/07 04:52:46 INFO mapred.JobClient: map 84% reduce 0%
15/05/07 04:52:47 INFO mapred.JobClient: map 89% reduce 0%
15/05/07 04:52:48 INFO mapred.JobClient: map 91% reduce 0%
15/05/07 04:52:50 INFO mapred.JobClient: map 97% reduce 0%

```

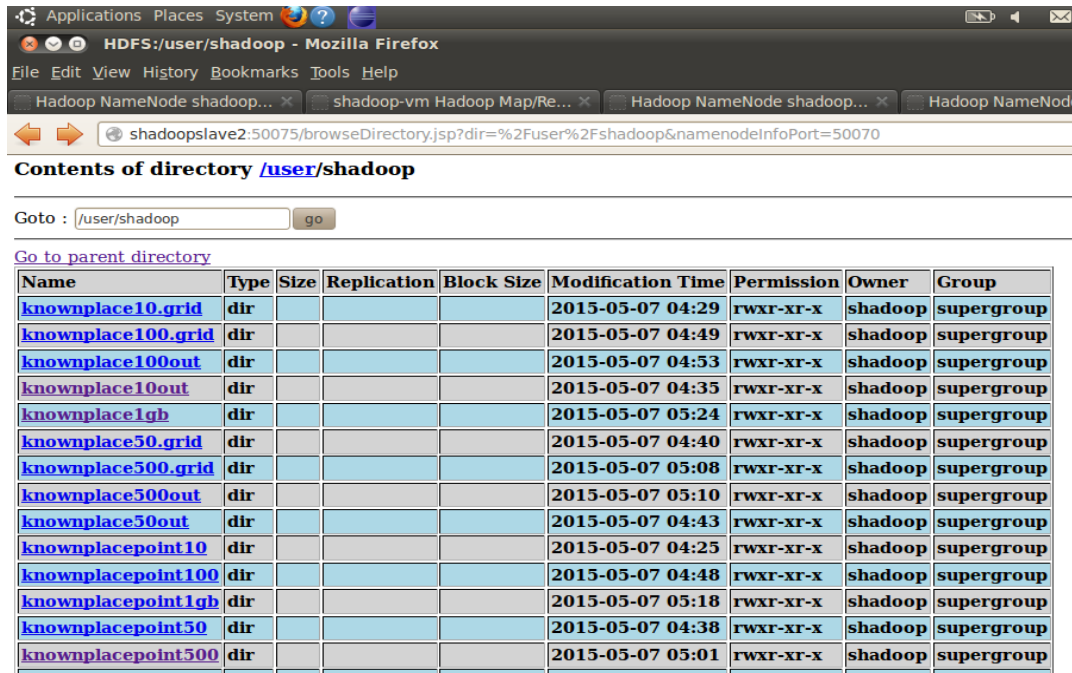
Figure 5.8: Snapshot of indexing and apply convexhull operation on 100MB file using SpatialHadoop

```

shadoop@shadoop-vm: ~/hadoop
15/05/07 04:53:02 INFO mapred.JobClient: Launched map tasks=2
15/05/07 04:53:02 INFO mapred.JobClient: Data-local map tasks=2
15/05/07 04:53:02 INFO mapred.JobClient: SLOTS_MILLIS_REDUCES=11617
15/05/07 04:53:02 INFO mapred.JobClient: File Input Format Counters
15/05/07 04:53:02 INFO mapred.JobClient: Bytes Read=104857974
15/05/07 04:53:02 INFO mapred.JobClient: File Output Format Counters
15/05/07 04:53:02 INFO mapred.JobClient: Bytes Written=1321
15/05/07 04:53:02 INFO mapred.JobClient: FileSystemCounters
15/05/07 04:53:02 INFO mapred.JobClient: FILE_BYTES_READ=8340
15/05/07 04:53:02 INFO mapred.JobClient: HDFS_BYTES_READ=104858214
15/05/07 04:53:02 INFO mapred.JobClient: FILE_BYTES_WRITTEN=185485
15/05/07 04:53:02 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=1321
15/05/07 04:53:02 INFO mapred.JobClient: Map-Reduce Framework
15/05/07 04:53:02 INFO mapred.JobClient: Map output materialized bytes=1164
15/05/07 04:53:02 INFO mapred.JobClient: Map input records=2885303
15/05/07 04:53:02 INFO mapred.JobClient: Reduce shuffle bytes=1164
15/05/07 04:53:02 INFO mapred.JobClient: Spilled Records=523
15/05/07 04:53:02 INFO mapred.JobClient: Map output bytes=46164848
15/05/07 04:53:02 INFO mapred.JobClient: Total committed heap usage (bytes)=421789696
15/05/07 04:53:02 INFO mapred.JobClient: CPU time spent (ms)=59580
15/05/07 04:53:02 INFO mapred.JobClient: Map input bytes=104857548
15/05/07 04:53:02 INFO mapred.JobClient: SPLIT_RAW_BYTES=240
15/05/07 04:53:02 INFO mapred.JobClient: Combine input records=2885698
15/05/07 04:53:02 INFO mapred.JobClient: Reduce input records=64
15/05/07 04:53:02 INFO mapred.JobClient: Reduce input groups=1
15/05/07 04:53:02 INFO mapred.JobClient: Combine output records=459
15/05/07 04:53:02 INFO mapred.JobClient: Physical memory (bytes) snapshot=486768640
15/05/07 04:53:02 INFO mapred.JobClient: Reduce output records=36
15/05/07 04:53:02 INFO mapred.JobClient: Virtual memory (bytes) snapshot=1123856384
15/05/07 04:53:02 INFO mapred.JobClient: Map output records=2885303
Total time: 57594 millis
shadoop@shadoop-vm:~/hadoop$

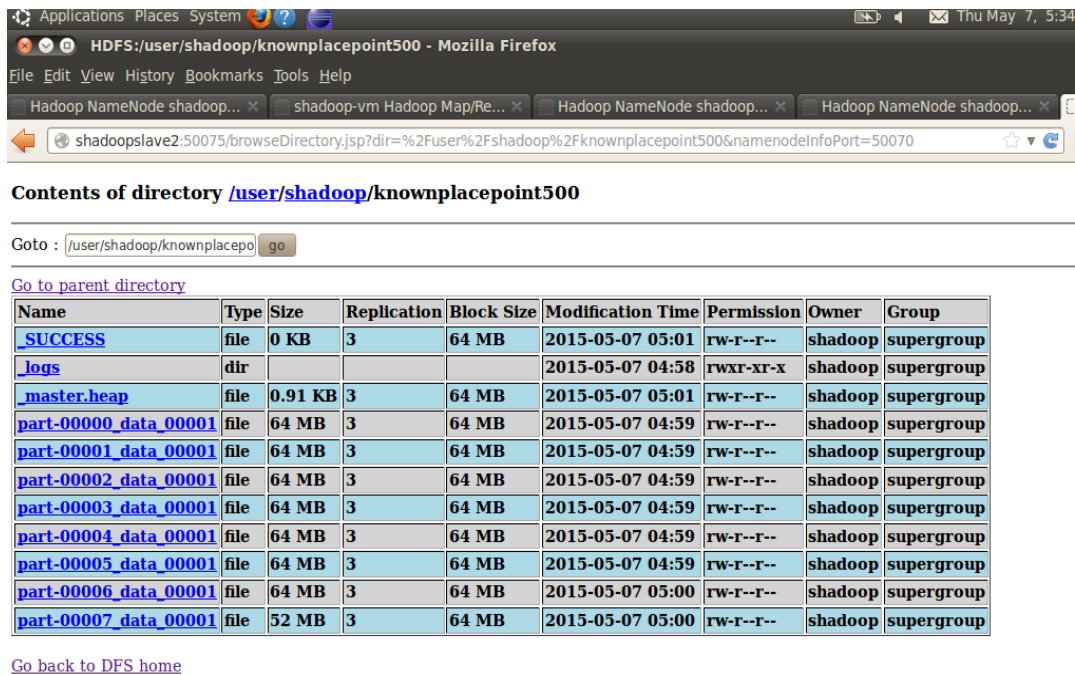
```

Figure 5.9: Snapshot of apply convexhull operation on 100MB file using SpatialHadoop



Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
<a href="#">knownplace10.gr1d</a>	dir				2015-05-07 04:29	rw-r--r--	shadoop	supergroup
<a href="#">knownplace100.gr1d</a>	dir				2015-05-07 04:49	rw-r--r--	shadoop	supergroup
<a href="#">knownplace100out</a>	dir				2015-05-07 04:53	rw-r--r--	shadoop	supergroup
<a href="#">knownplace10out</a>	dir				2015-05-07 04:35	rw-r--r--	shadoop	supergroup
<a href="#">knownplace1gb</a>	dir				2015-05-07 05:24	rw-r--r--	shadoop	supergroup
<a href="#">knownplace50.gr1d</a>	dir				2015-05-07 04:40	rw-r--r--	shadoop	supergroup
<a href="#">knownplace500.gr1d</a>	dir				2015-05-07 05:08	rw-r--r--	shadoop	supergroup
<a href="#">knownplace500out</a>	dir				2015-05-07 05:10	rw-r--r--	shadoop	supergroup
<a href="#">knownplace50out</a>	dir				2015-05-07 04:43	rw-r--r--	shadoop	supergroup
<a href="#">knownplacepoint10</a>	dir				2015-05-07 04:25	rw-r--r--	shadoop	supergroup
<a href="#">knownplacepoint100</a>	dir				2015-05-07 04:48	rw-r--r--	shadoop	supergroup
<a href="#">knownplacepoint1gb</a>	dir				2015-05-07 05:18	rw-r--r--	shadoop	supergroup
<a href="#">knownplacepoint50</a>	dir				2015-05-07 04:38	rw-r--r--	shadoop	supergroup
<a href="#">knownplacepoint500</a>	dir				2015-05-07 05:01	rw-r--r--	shadoop	supergroup

Figure 5.10: Snapshot of all files which are stored in DFS



Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
<a href="#">_SUCCESS</a>	file	0 KB	3	64 MB	2015-05-07 05:01	rw-r--r--	shadoop	supergroup
<a href="#">_logs</a>	dir				2015-05-07 04:58	rw-r--r--	shadoop	supergroup
<a href="#">_master.heap</a>	file	0.91 KB	3	64 MB	2015-05-07 05:01	rw-r--r--	shadoop	supergroup
<a href="#">part-00000_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 04:59	rw-r--r--	shadoop	supergroup
<a href="#">part-00001_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 04:59	rw-r--r--	shadoop	supergroup
<a href="#">part-00002_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 04:59	rw-r--r--	shadoop	supergroup
<a href="#">part-00003_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 04:59	rw-r--r--	shadoop	supergroup
<a href="#">part-00004_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 04:59	rw-r--r--	shadoop	supergroup
<a href="#">part-00005_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 04:59	rw-r--r--	shadoop	supergroup
<a href="#">part-00006_data_00001</a>	file	64 MB	3	64 MB	2015-05-07 05:00	rw-r--r--	shadoop	supergroup
<a href="#">part-00007_data_00001</a>	file	52 MB	3	64 MB	2015-05-07 05:00	rw-r--r--	shadoop	supergroup

[Go back to DFS home](#)

Figure 5.11: Snapshot of multiple files of 500MB file generated by HDFS



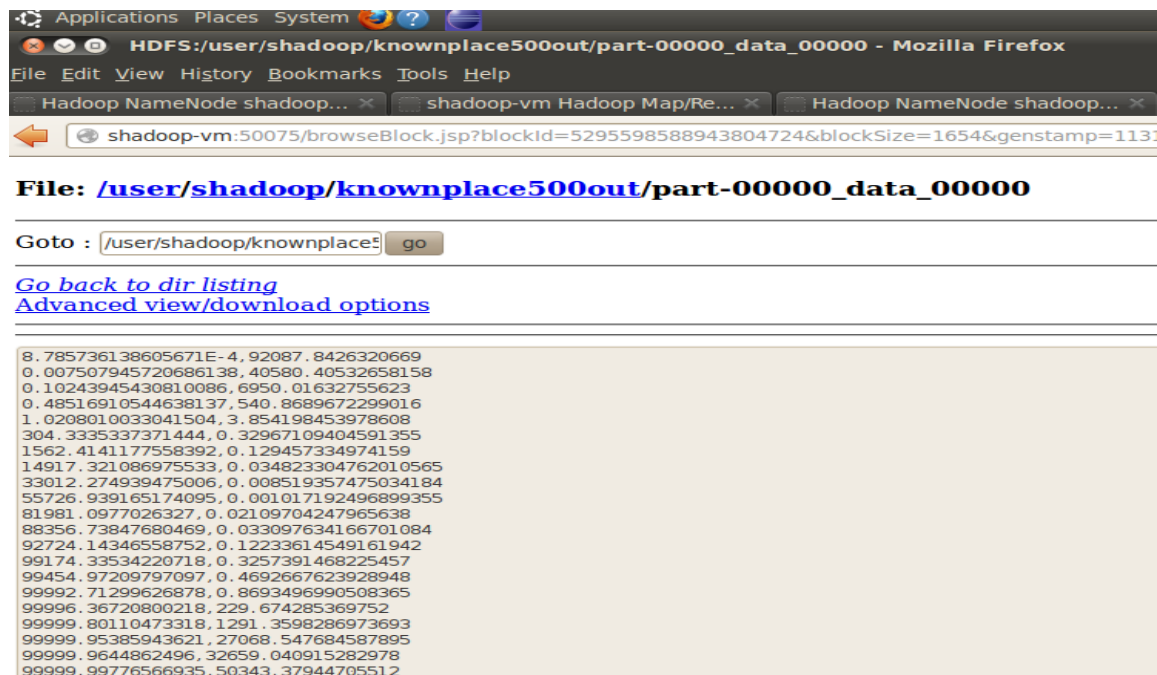


Figure 5.12: Snapshot of output of convexhull operation of 500MB file on browser

Size(MB) Vs Time(Minutes)	Generation of File	Indexing of File	Convexhull Operation Using SpatialHadoop	Convexhull Operation Using QGIS
10	0.1624	0.1694	0.5942	0.1768
50	0.8177	0.7337	0.9627	0.4825
100	1.5291	1.5325	0.9591	--
250	1.9576	2.0886	1.0713	--
500	3.9150	4.0052	2.0090	--

Table 5.1: Table of size and time values for convexhull operation using SpatialHadoop and QGIS

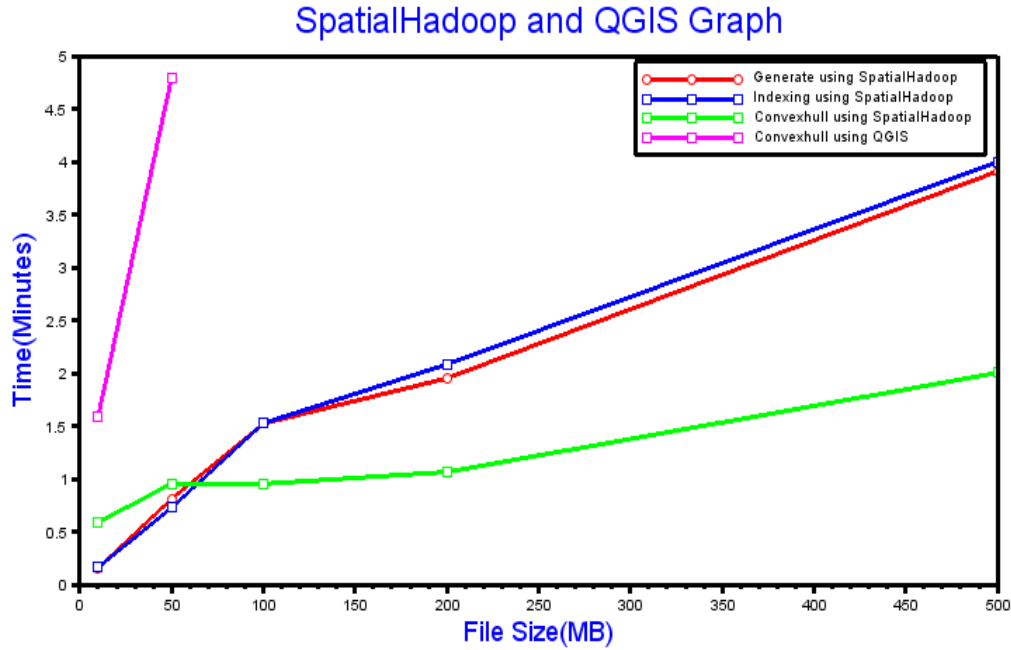


Figure 5.13: Graph of Size vs. Time for convexhull operation using SpatialHadoop

## 5.2 Discussion

There are some limitations of QGIS software like the data above 50MB size was not displayed or worked as per the requirement, it only supports python plug-in for adding Geo-algorithm, after adding some layers on map area- QGIS becomes slow and freezes, when we used "QuickWKT" plug-in from Shape file to WKT conversion then it can not handle base64 encoded WKB String( binary-to-text encoding schemes that represent binary data in an ASCII string format).

In spatial hadoop, limitations of the GIS software can be easily overcome. To overcome a problem of QuickWKT plugin of QGIS, make conversion program of Shape to WKT format and use it in Spatial Hadoop as data input. After giving data as input or generating data using Spatial Hadoop, do some indexing task and apply some geo-processing operation on it as shown above snapshot. After getting result of geo-processing operation (for this project use Convexhull algorithm for getting one polygon which contains all points of a file) calculate how much time is required

for output generation and then generate a graph of different data size (multiMB) vs. required time to process a different operations on data. As shown in the above mentioned graph, after applying Convexhull algorithm on 500MB size of data, it takes time of about 2.0090 minutes, so it is concluded that spatial hadoop is more efficient and gives better result compare to GIS software for large data.

# Chapter 6

## Conclusion and Future Scope

### 6.1 Conclusion and Future Scope

To handle Big Geo-data, scientists face many problems such as heterogeneity of data, security, scalability, manipulation of data, analysis of data, more storage capacity and hardware etc and It can not be handle using single machine and available GIS software. So that using distributed processing, these problems can be resolved very efficiently in this research work. For geo-processing on spatial data, shape file is converted into WKT and CSV file using Java Programing Language and some Library/JAR. This file is used in spatial hadoop as an input and applied Convexhull Geo-processing operation. After calculating time which is required for processing, it is concluded that Spatial Hadoop is very fast and gives better results. In these project work, Limited operations are done to achieve the required task but in future many other operations like Conversion of a file in different format, other operation on raster data (Image filtering, Image analysis etc.) can be performed using spatial hadoop.



# Appendix A

## Published Paper

Sr. No	Paper Title	Conference Name	ISBN	Status
1.	GeoProcessing Workflow models for Distributed Processing Frameworks	IJCA - 2015	International Journal of Computer Applications	Published
2.	Big-Geo Data Processing using Distributed Processing Frameworks	IJSER - 2015	International Journal of Scientific and Engineering Research	Accepted

# References

- [1] Ahmed Eldawy , Mohamed F. Mokbel "A Demonstration of SpatialHadoop: An Efficient MapReduce Framework for Spatial Data",Proceedings of the VLDB Endowment, Vol. 6, No. 12,Oct-2013.
- [2] Ablimit Aji1, Fusheng Wang, Hoang Vo1 Rubao Lee, Qiaoling Liu1 Xiaodong Zhang, Joel Saltz. "Hadoop GIS: A High Performance Spatial Data Warehousing System over MapReduce", Proceedings of the VLDB Endowment, Vol. 6, No. 11, Copyright 2013 VLDB Endowment 2150 8097/13/09.
- [3] Ahmed Eldawy ,Yuan Li , Mohamed F. Mokbel, Ravi Janardan"CG-Hadoop: Computational Geometry in MapReduce",November 05 - 08 2013, Orlando, FL, USA Copyright 2013 ACM 978-1-4503-2521-9/13/11.
- [4] Roberto Giachetta,"Advancing a geospatial framework to the MapReduce Model",2013 Artical, ACM
- [5] *[http : //www.apache.org](http://www.apache.org)*
- [6] N Skytland, "Big data: What is nasa doing with big data today." Open. Gov,open access article, 2012.
- [7] S Kaisler, F Armour, J A Espinosa, W Money; "Big Data: Issues and Challenges Moving Forward",IEEE 995-1004, Jan -2013

- [8] C.L. Philip Chen, C.-Y. Zhang "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data ", Information Sciences 275 (2014) 314–347
- [9] [http : //www.opengeospatial.org/standards](http://www.opengeospatial.org/standards)
- [10] [http : //en.wikipedia.org/wiki/Well-known\\_text](http://en.wikipedia.org/wiki/Well-known_text), 18November2014at17 : 28
- [11] D.Pertesis,C.Doulkeridis,"Efficient skyline query processing in Spatial-Hadoop",Information Systems(2014), 2014.10.003
- [12] Gao, Song and Li, Linna and Li, Wenwen and Janowicz, Krzysztof and Zhang, Yue, " Constructing gazetteers from volunteered Big Geo-Data based on Hadoop." Computers, Environment and Urban Systems (2014),2014.02.004
- [13] J. Dean and S. Ghemawat." MapReduce: Simplified Data Processing on Large Clusters." ,Communications of ACM, 51, 2008.
- [14] [http : //hadoop.apache.org/](http://hadoop.apache.org/)
- [15] Kalavri, V.; Vlassov, V., "MapReduce: Limitations, Optimizations and Open Issues," Trust, Security and Privacy in Computing and Communications (Trust-Com), 2013 12th IEEE International Conference on , vol., no.,pp.1031,1038,16-18July2013
- [16] [http : //hortonworks.com/hadoop/yarn/](http://hortonworks.com/hadoop/yarn/)
- [17] [https : //developer.yahoo.com/hadoop/](https://developer.yahoo.com/hadoop/)
- [18] G. Wei Xiang Goh;T. Kian-Lee, "Elastic MapReduce Execution", 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,2014.
- [19] Eldawy, Ahmed, and Mohamed F. Mokbel. "SpatialHadoop: A MapReduce Framework for Spatial Data." ICDE, 2015.