Big-Geo Data Processing using Distributed Processing Frameworks

Shruti Thakker [1], Jhummarwala Abdul [2], Dr.M.B.Potdar [2]

[1] Institute of Technology, Ahmadabad-382481, India

[2] Bhaskaracharya Institute for Space Applications and Geo-informatics Gnadhinagar-382007, India

Abstract— Geographic Information Systems (GIS) platform has been required high storage capacity and high computational power to handle and process with massive spatial data. This paper aims of processing of Geodata using Distributed processing Frameworks. GIS software like QGIS, ArcGIS, GRASS, Open JUMP etc cannot be handle and process on large volume of Geo data. Therefore, to handle and process on these types of large data it is needed to deploy Hadoop Distributed processing framework. GeoProcessing Workflow can be used to represent almost every GIS application. This paper explains the GeoProcessing Workflow for processing of image data. Also explains Hadoop Distributed File System (HDFS), MapReduce Programming Model and Spatial Hadoop architecture. This work deals with application of the KNN algorithm on large amount of spatial data. It is found that the Spatial Hadoop performs fast data processing and gives better performance compare to GIS software.

Index Terms — GIS, Geoproceeing Workflow, Distributed System, Hadoop, MapReduce, Spatial Hadoop.

1 Introduction

Today, Geographical Information System (GIS) is used to perform very well known tasks very easily like generation of map (using longitude and latitude), geographic analysis and manipulation of geographical data. The concept of GIS was first proposed by Dr. Roger Tomlinson in the 1960s, it has gone through a long process of development. In large scale applications such as Government Projects, Business and Industrial projects including real estate, community planning, natural resources, public health, climatology, and transportation etc make use of Geographical Information System. GIS is used to capture, manipulate, analyze, manage, store and present all types of spatial data [2]. Spatial data is periodically generated by special sensors (like OGC, MSS, TM, ETM+ and OLI&TIRS etc [4]) on board satellites and GPS devices. Compared to the traditional processing methods like MPI, OpenMP etc, the Hadoop distributed parallel computing enhances the computing speed when size of the dataset is very large and increasing continuously (Specially for real time data). When these types of data are used on a single machine, the power lies in its ability to scale to hundreds or thousands of nodes, each with several processing cores [1]. To efficiently distribute large amount of work with the set of nodes, distributed processing is used. In Section 1, a brief introduction to Geographical Information System (GIS) is given. In Sections 2, GeoProcessing workflow and preprocessing on Geodata are discussed. The Section 3 explains the QGIS and geoprocessing operation using it. Section 4 gives background knowledge of the distributed processing in which the architecture of HDFS, MapReduce and Spatial Hadoop are explained. Section 4 deals with the data generation and its analysis. Section 5 presents result and discussion. Section 6 presents conclusions and plans for future work.

2 GEOPROCESSING WORKFLOW

In recent years, the use of GIS applications has increased as the requirements for geo-spatial information services have grown. ArcGIS, QGIS, OPENJUMP, ArcMap, ArcObject, GRASS etc [2] are providing GUIs for development of such workflow and processing of Geodata. GeoProcessing was one of the original proposal when GIS was invented. GeoProcessing Workflow can be used to represent almost every GIS application. It integrates data and services in an interoperable way wherein each part of the workflow is responsible only for a specific task without having knowledge of the general purpose of the workflow [1]. As shown in figure 1 on GeoProcessing Workflow, first we have to load the data/image as an input file. After loading the data in GeoProcessing workflow, we have to apply common operations such as Image Enhancement, Image Classification, Image Overlapping, Image Segmentation, Image Filtering, Find farthest or nearest pair of point and Convex hull of data etc as per the output requirement. Each operation does some specific task depending on the process. As shown in figure 1, for different operations, different types of input parameters are used that generate intermediate or final output. After getting the output, we can call process N times and do the operations as per the requirement.

The data consumed by a workflow varies both in terms of volume and variety [5]. For improving the performance of GeoProcessing Workflow, different types of techniques are adopted. Pallickara and Malensek (2011) [13] mainly focused on Geospatial data flow for processing, which enabled interactions between the data and the computation/analysis. This is used for processing on Geodata as per the requirement of output. The input data totally depends on the rate at which the structure of data changes. Data is collected from a variety of distributed resources and converted into a specific format according to user requirements such as transfering into OGC standards. The Well Known Text (WKT) and Well Known Byte (WKB) are very popular standards of OGC. Generally before processing on Geographical data in GeoProcessing Workflow, some of the tasks are applied like Data collection from observational instruments, Data capture, Visualization, Data analysis [6]. For example, to perform atmospheric measurements, the collections of data from observing instruments like in-situ measurements and remote sensing instruments are required. Data capture is a challenge which involves extraction of the large amount of data being generated for computations and to collect data nodes for further processing such as monitoring, analysis, and archival [6]. This process is known as data capturing and used for parallel processing technique at I/O layer. To visualize large volume of data using observation, simulation or experiment result and transforming them into image, thematic maps, statistical charts, and map overlays for further processing based on GeoProcessing workflow because visualization is related to data model and representation.

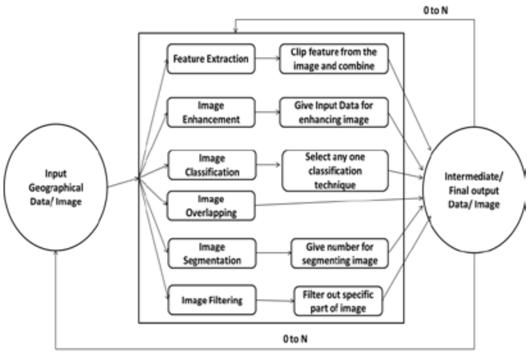
0 to N
Feature Extraction
Clip feature from the image and combine
Image Enhancement
Give Input Data for enhancing image
Input Geographical Data/ Image
Image Classification
Select any one classification technique
Image Overlapping
Intermediate/ Final output Data/ Image
Image Segmentation
Give number for segmenting image
Image Filtering
Filter out specific part of image
0 to N

Fig. 1. Schematics of GeoProcessing Workflow [1]

3 GEO-PROCESSING OPERATION USING QUANTUM GIS (QGIS)

QGIS is used for doing GeoProcessing operations on Geo data. QGIS has an optional scripting support using python language. Using python plug-in in QGIS, we can add our own geo algorithms through script and apply them on data for processing. GeoProcessing tools are required for all the raster or vector data related to operations such as image manipulation, segmentation, analysis, filtering, geometry, feature extraction, creation, addition etc. In QGIS, Spatial query apply spatial operations like as Equals, Disjoint, Within, Overlaps, Intersects etc and store result in .qgs files. After applying some operation on raster or vector data and creating any map, it can then be stored using a project file. Therefore, a user can open the project again and do changes based on the requirement of the output. 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.

GeoProcessing operation is applied on spatial data using GIS software like QGIS. Overlap of two small size of shape files using QGIS is very simple task in QGIS. All the data which is in multiKB size can be easily opened and all the processes can be easily performed on it. The result of the overlapping of two shape file is generated in within 5 second. But there are some limitations in QGIS such as the size of data of about 50 MB or more cannot be opened and sometime its create some problem in software due to of which it gets hanged, only supports python plug-in for adding Geo-algorithm, after adding some layers on map area- QGIS becomes slow and freezes. But now a day, the data is generated periodically and in very large amount due to increase in number of internet users and machine to machine connections. In 1990, normal storage capacity of hardware was 1400 MB, transfer speed of data was 4.5Mbps and the entire drive could be read in 5 minutes. In 2010, storage capacity of hardware was 1 TB, transfer speed of data was 100MB/s and it requires 3 hours to read. Based on this we can say that storage capacity has grown exponentially but reading speed has very negligible effect. To overcome the above mentioned problem, distributive processing system is used and Hadoop is one of the example. To handle large Geodata in variety of format, it is not easy and feasible for any single node and many problems are encountered such as heterogeneity of data, security issues and require high speed for transformation of data, scalability, manipulation of data, analysis of data, continuous requirement of more storage capacity and hardware etc [1]. Also using available tools of GIS like QGIS, ArcGIS, OpenJUMP etc, the large amount of data cannot be processed due to limitation of resources. A typical large size GeoProcessing of a raster or vector data, it takes several minutes and such repeated attempts lead to system crash. Using distributed processing; these problems can be resolved very efficiently. As explained in section 2, for distribution of file we use Hadoop Distributed File System (HDFS) and using MapReduce programming model, we can process these files at a same time and generate final output as per the user requirement in reasonable time.

4. BACKGROUND KNOWLEDGE OF DISTRIBUTED PROCESSING

In our daily life with high demand of the resources, single system cannot provide high performance and high efficiency. The data, which is generated by a spatial sensor, device and satellite, is very large in a size so it is not give efficient result using GIS software. But distributed nature of geographic data for Distributed processing frameworks, GeoProcessing Workflow provides very easy processing of highly distributed and complex data for a wide variety of applications. To achieve these, we have to move on distributed system. OpenMP, MPI and MapReduce are the most widely recognized parallel or distributed programming frameworks. Distributed System is good in terms of Cost, Performance, Scalability and Reliability. OpenMP is mainly used for shared memory systems, MPI is a standard for distributed memory systems and MapReduce is a standard on framework for big data processing. In GIS, the data are usually generated in very large amount and in variety of formats. We can very easily and efficiently process the Geodata. Compared to MPI and OpenMP, The Hadoop is open source software framework for processing of large amount of data in distributed environment using two components, the Hadoop Distributed File System (HDFS) and MapReduce

4.1 Hadoop Distributed File System (HDFS)

HDFS is designed to be deployed on low-cost hardware and highly fault-tolerant system. Main goal of HDFS is to provide high throughput for access data which are very large in a size. Using Hadoop Distributed File System, a large data set is distributed dynamically among a number of nodes. HDFS use two types of Node: One Name node and Several Data nodes. Name node is on a master server that manages the file system namespace and stores the information of file like size, access rights, and location [3]. The Data Nodes, usually one per node in the cluster, store actual file records [3]. HDFS exposes a file system namespace and allows user data to be stored in files. A large file is divided into one or more Blocks/Chunks (Normally 64MB size) and these blocks are stored in Data Nodes. The Name Node works as a main data sever and it executes file system namespace operations like opening, closing, and renaming files and directories. Name node also determines the mapping of blocks to Data Nodes [7]. The Data Node is used for serving read and writes requests from the clients file System. They are also performing operations like block creation, deletion, and replication as per instruction from Name Node [7]. In HDFS, the access to file is either directly or using proxy server. In direct access, client can retrieve metadata such as block's locations, size and access information from Name node. Also, Client can directly access Data node(s). For accessing data, Java and C++ APIs are normally used by MapReduce. In proxy based access, client is communicating through a proxy. These proxy servers are packaged with Hadoop like Thrift, WebHDFS REST and Avro etc. Secondly, The HDFS uses multi copy strategy so that data can be stored in many nodes by replication [8]. This strategy can effectively improve the reliability, data security and availability of data storage.

4.2 MapReduce Model

The Hadoop framework employs MapReduce programming Model, which uses two functions *Map* and *Reduce* in this sequence. It is mainly used to adopt divide and conquers strategy for distribution of data and do parallel processing on it. These two functions are User-defined. The *map* function maps a single input record (in form of (Key, Value) = (K1, V1) Pair) to a set of intermediate key value pairs (K2, V2), while the *reduce* function takes all values associated and the Intermediate key k3 and produce corresponding value of (K3, V3) Pair [6].

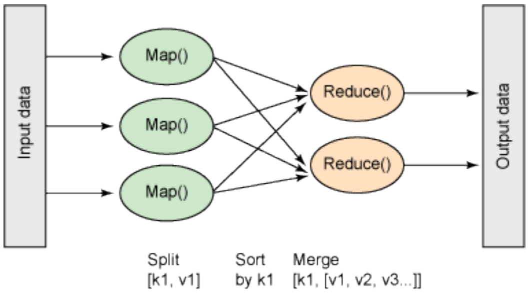

Figure 2: MapReduce Architecture

4.3 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 takes very long time compared to spatial Hadoop and also give very bad performance.

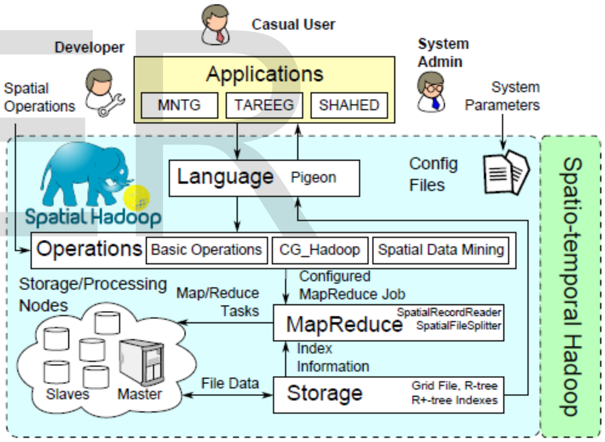

Figure 3: Architecture of SpatialHadoop

Type of Users in SpatialHadoop

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

Spatial Hadoop works with following four types of layers:

1. Language layer: In Hadoop, a program is written Pig Latin language, which is not reliable for spatial data. 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[16]. 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[16].

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[16]. SpatialHadoop contain geometry operations e.g. kNN join, RNN, overlay, Farthest-pair, Nearest-pair etc

4.4 Installation and Configuring Spatial Hadoop on top of multinode hadoop cluster

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 paper 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-1.21 in Ubuntu 14.10 (Utopia Unicorn) Operating System and using Hadoop-1.21 multinode setup has been installed. To create cluster of multinodes it uses the VMware Workstation 9. One Master node and three slave nodes are used for create multinode hadoop cluster. After setup Single node cluster using installation of Java, SSH and Hadoop-1.2.1, Configure the files and move 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.

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 does some operations as shown in below section. The steps are available on the official web page of SpatialHadoop (http://spatialhadoop.cs.umn.edu).

5 DATA AND ANALYSIS

In this paper, a file has been generated on spatial hadoop. After that the grid Indexing has been performed and applied KNN algorithm on it. For this purpose multi size of data has been used and an analysis of the result has been displayed in result section

TABLE 1
FILE AND SIZE VALUE USING SPATIAL HADOOP

Size of a File(MB) /Time (Second)	10 MB	50 MB	100 MB	200 MB	500 MB
Genertion of Data	2.86	2.70	5.77	7.52	9.24
Apply Grid Index	2.94	2.82	6.06	7.95	9.83
KNN for 50 points	1.08	1.09	1.23	1.4	1.8

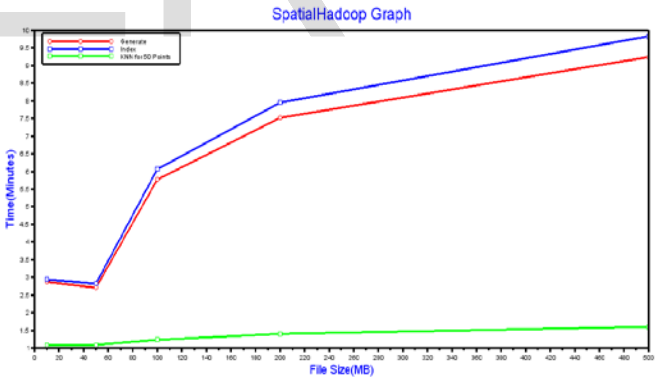

Figure: 4 Graph of Time vs. File size for KNN algorithm using Spatial Hadoop

6

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. In Spatial Hadoop, limitations of the GIS software can be easily overcome. Also in Spatial Hadoop, We can generate a file which is in size of multi gigabyte or we can directly apply a data as input file. 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 KNN algorithm for getting nearest neighbor value for a particular point) 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 KNN algorithm on 500MB size of data, it takes time of about 1.8 sec, so it is concluded that spatial hadoop is more efficient and gives better result compared to GIS software for large data

7 CONCLUSIONS 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 cannot be handled 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 Programming Language and some Library/JAR. This file is used in spatial hadoop as an input and applied KNN GeoProcessing operation. After calculating time which is required for processing, it is concluded that Spatial hadoop is very fast and gives better results. In these paper, 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.

Acknowledgment

We are thankful to Shri. T. P. Singh, Director, Bhaskaracharya Institute for Space Applications and Geoinformatics for providing infrastructure and constant encouragement

REFERENCES

[1] Shruti Thakker, Jhummarwala Abdul and M B Potdar. Article: GeoProcessing Workflow Models for Distributed Processing Frameworks. *International Journal of Computer Applications* 113(1):33-38, March 2015.

[2] Siddiqui, S.T.; Alam, M.S.; Bokhari, M.U., "Software Tools Required to Develop GIS Applications: An Overview," *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on* , vol., no., pp.51,56, 7-8 Jan. 2012.

[3] Eldawy, A., Li, Y., Mokbel, M. F., & Janardan, R. “CG_Hadoop: computational geometry in MapReduce.” In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 284-293, ACM. November 2013.

[4] Central Africa Regional Program for the Environment: http://carpe.umd.edu/geospatial/satellite_imagery_resources.php

[5] Pallickara, S. L., Malensek, M., & Pallickara, S., “On the Processing of Extreme Scale Datasets in the Geosciences.” In *Handbook of Data Intensive Computing*, pp. 521-537, Springer New York. 2011.

[6] Dean, J., & Ghemawat, S. “MapReduce: simplified data processing on large clusters.” *Communications of the ACM*, *51*, *vol.*1, 107-113. 2013.

[7] The Apache Software Foundation: http://hadoop.apache.org/

[8] Kalavri, V.; Vlassov, V., "MapReduce: Limitations, Optimizations and Open Issues," *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on* , vol., no., pp.1031,1038, 16-18 July 2013.

[9] Hadoop YARN: http://hortonworks.com/hadoop/yarn/

[10] Hadoop at Yahoo!: https://developer.yahoo.com/hadoop/

[11] Wei Xiang Goh; Kian-Lee Tan, "Elastic MapReduce Execution," *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on* , vol., no., pp.216,225, 26-29 May 2014.

[12] Schaeffer, B., Baranski, B., Foerster, T., & Brauner, J. “ A service-oriented framework for real-time and distributed geoprocessing“ ,In *Geospatial Free and Open Source Software in the 21st Century* , pp. 3-20, Springer Berlin Heidelberg. 2012.

[13] Khan, F. A., & Brezany, P. “Provenance Support for Data-Intensive Scientific Workflows.” In *Grid and Cloud Database Management* , pp. 215-234, Springer Berlin Heidelberg. 2011.

[14] ISO Standards : www.iso.org/iso/catalouge_detail.htm

[15] OGC Standards : www.opengeospatial.org/standards

[16] Eldawy, Ahmed, and Mohamed F. Mokbel. "SpatialHadoop: A MapReduce Framework for Spatial Data." ICDE, 2015.