GeoProcessing Workflow Models for Distributed Processing Frameworks

Shruti Thakker
Institute of Technology, Nirma University,
Ahmadabad-382481

Jhummarwala Abdul
Bhaskaracharya Institute for Space Applications and Geo-informatics (BISAG),
Gnadhinagar-382007

M. B. Potdar, Ph.D.
Bhaskaracharya Institute for Space Applications and Geo-informatics (BISAG),
Gnadhinagar-382007

ABSTRACT
Geographic Information Systems (GIS) platforms are used to implement and deal with the massive spatial data, especially image data. Therefore, these platforms require high storage capacity and high computational power to process them. This paper aims of processing of Geodata using Distributed processing Frameworks. Large volume of Geodata cannot be processed using desktop GIS tools such as QGIS, ArcGIS, GRASS, OpenJUMP etc. Therefore, to handle and process on these types of large data, use of Hadoop Distributed processing framework needs to be deployed. GeoProcessing is a GIS operation used to manipulate spatial data. It is one of the original proposal in GIS development. Almost every GIS application is represented by a GeoProcessing Workflow. This paper explains the GeoProcessing Workflow for processing of image data. Also explains Hadoop Distributed File System (HDFS), MapReduce Programming Model and Yet Another Resource Negotiator (YARN) architecture, useful in large spatial data handling and analysis at fast rate**.**

General Terms
GeoProcessing, Distributed System.

Keywords
GIS, Geoproceeing Workflow, Distributed System, Hadoop, YARN, MapReduce Classic.

1. INTRODUCTION
Generation of map (using longitude and latitude) and geographic analysis are very well known task in the remote sensing field. The concept of GIS (Geographical Information System) was first proposed by Dr. Roger Tomlinson in the 1960s, it has gone through a long process of development. Using GIS, we can perform these tasks very easily. Before the GIS technology, only a few people had the skills necessary to use geographic information to help decision making and problem solving. Today, GIS is employed by thousands of organizations over world covering large number of Industries. It is now playing an important role in many areas like Government Project, Business and Industry with application including real estate, public health, natural resources, climatology, community planning and transportation etc. It is used to capture, manipulate, analyze, manage, store and present all types of spatial data [1]. Spatial data is periodically generated by special sensors (like OGC, MSS, TM, ETM+ and OLI&TIRS etc [3]), on board satellites and GPS devices. Spatial data is information that identifies the geographic location and characteristics of natural or manmade features and boundaries on the earth and/or complex geographic features [1]. These features can be represented by points, lines, polygons using different colors and different types of the patterns. Compared to the traditional processing methods like MPI, OpenMP etc, Hadoop distributed parallel computing enhances the computing speed when size of the dataset is very large and increase data. Continuously (Especially for real time data). When these types of data are used on a single machine, there power lies in its ability to scale to hundreds or thousands of nodes, each with several processing cores. To efficiently distribute large amount of work with the set of nodes, distributed processing is used. In Section 1, after brief introduce to Geographical Information System (GIS). Section 2 gives background knowledge about the distributed processing in which explain architecture of HDFS, Mapreduce, YARN and the comparison of MapReduce classic and YARN is given. In Sections 3, GeoProcessing workflow and preprocessing on Geodata are discussed. Section 4 presents conclusions and plans for future work.

2. KNOWLEDGE OF DISTRIBUTED PROCESSING
In our daily life with high demand of the resources, single system cannot provide high performance and high efficiency. 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 use 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 which are 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.

2.1 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. 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 store the information of file like size, access rights, and location [2]. The Data Nodes, usually one per node in the cluster store actual file record [2]. 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 [5]. 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 [5]. 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 is 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.

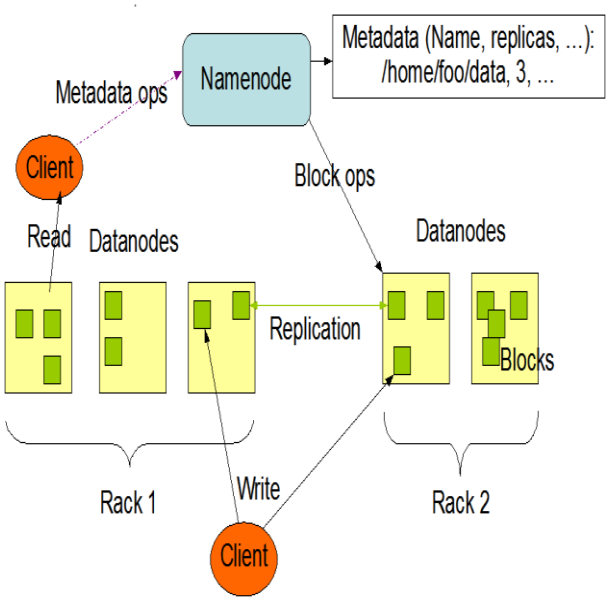

Figure 1: HDFS Architecture [3]

Secondly, The HDFS uses multi copy strategy so that data can be stored in many nodes by replication [6]. When data node needs 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, data security and availability of data storage.

2.2 Mapreduce Model

The Hadoop framework employs MapReduce programming Model, which uses two functions *Map* and *Reduce* in 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 [4]. A MapReduce program is inherently parallel and can greatly decrease the computational time when processing huge datasets. Mapreduce mainly used in wide range of applications like machine learning, graph processing, web application in traversal etc.

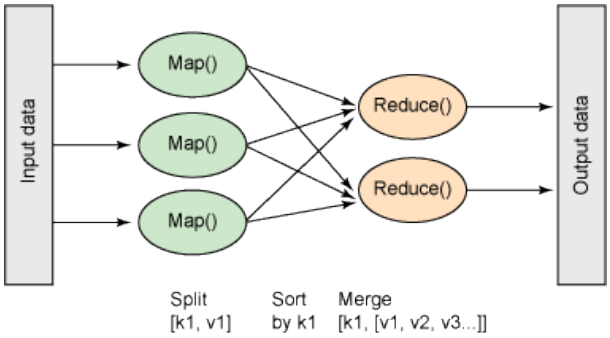

Figure 2: MapReduce Architecture

The MapReduce functioning can be illustrated as follows:

- Assume a file containing (key, value) pair (City, Temperature), with data {(Toronto, 20), (Chicago, 27), (NewYork, 15), (New York, 22), (Rome, 32), (Toronto, 4), (Rome, 33), (New York, 18), (Rome,38)} . In such a file same key represented multiple times. The objective to find maximum temperature for each city.
- Using the MapReduce framework, break the task into three map tasks, where each mapper works on one of the three files and the mapper task goes through the data and returns the maximum temperature for each city. For example, the results produced from one mapper task for the data is: {(Toronto, 20), (Chicago, 27), (New York, 15)}.
- The four mapper tasks produce intermediate results: {(NewYork, 22), (NewYork, 18),(NewYork,15),} {(Toronto, 20), (Toronto, 4)}, { (Rome, 38),(Rome,32)}, { (Chicago, 27)}.
- The output stream is then fed into the reduce tasks, which combined the input data and output a single value for each city, producing a final result as {(New York, 33), (Toronto, 20), (Rome, 38), (Chicago, 27),}. This example shows how to map and reduce tasks work for the large amount of data.

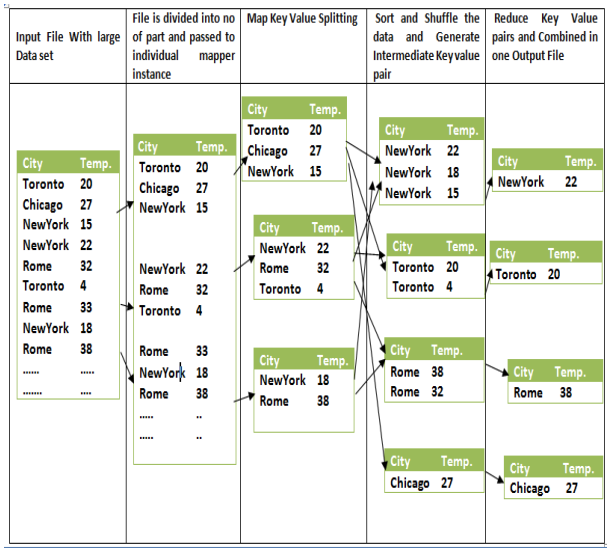

Figure 3: Graphical Representation of Mapreduce Example

2.3 YARN (Yet another Resource Negotiator) Architecture

In the MapReduce version 1 architecture, the cluster is managed by a service called the JobTracker. TaskTracker services live on each node and would launch tasks on behalf of jobs. The JobTracker serves information about completed jobs. In YARN (MapReduce version 2), the function of the JobTracker is split into two domains, Resource manager and Application Master [6]. TaskTracker has been replaced with the NodeManager. It is responsible for launching containers, each of which can house a map or reduce task [7]. The ResourceManager and the NodeManager form a new generic system for managing applications in a distributed manner [9]. The ResourceManager is managing resources among all applications in the system. The ApplicationMaster is a framework-specific entity that negotiates resources from the ResourceManager and works with the NodeManager to execute and monitor the component tasks [7]. The figure 4 represents architecture of YARN:

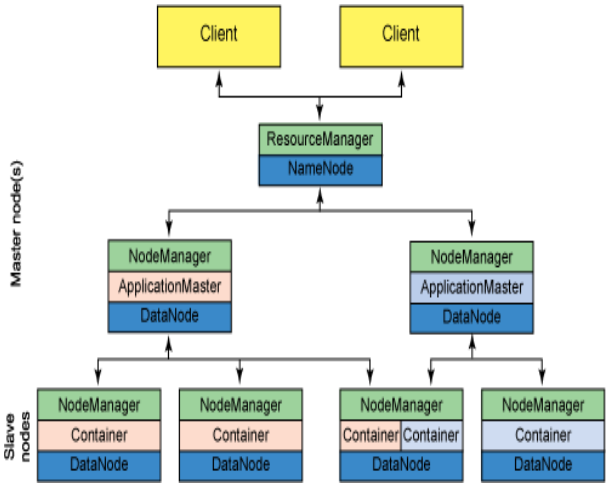

Figure 4: YARN Architecture

- ResourceManager: It has a scheduler, which is responsible for allocating resources to the various applications running in the cluster, according to constraints such as queue capacities and user limits [7]. This scheduler schedules the resources based on requirements of each application.
- ApplicationMaster: Each of has responsibility for negotiating appropriate resource containers from the scheduler, tracking their status, and monitoring their progress [9].
- NodeManager: It is allocated per-machine on slave node, which is responsible for launching the applications containers, and monitoring their resource usage such as CPU, memory, disk, network and reporting to the ResourceManager.

2.4 Comparison of YARN (Yet another Resource Negotiator) and Classical

YARN is the improved version of the MapReduced classical. It is characterized as a large-scale, distributed operating system for big data analysis. Due to some limitations of MapReduce version 1(Classical) such as hard partition of resources into map and reduce slots, limited size of cluster node (max 4000 [8]) and limited concurrent tasks (max 40,000), scalability, fault-tolerance and capability of processing vast amounts of data, query execution time can often be several hours. The YARN supports MapReduce and Non-MapReduce applications on the same cluster, all nodes have their own resources (like memory, CPU) which are allocated to applications when request made improved cluster utilization, more scalability compared (max 45,000 [8]) to MapReduce classic etc[9]. A YARN is software that rewrites and decouples MapReduce's resource management and schedules the capabilities from the data processing component. The MapReduce and YARN are compared in table 1:

Table 1: Comparison of MapReduce 1(Classic) and YARN:

Features	MapReduce 1 (Classic)	YARN
Scalability	Less Scalable	More Scalable
Maximum No of Node	4000	>40,000
Limitation of no of Concurrent Tasks	40,000	> 3,00,000
Supported Application	MapReduce Applications in same Cluster	Both MapReduce & Non-MapReduce Applications in same Cluster
Cluster Utilization	Less Utilization	More Utilization
Architecture	Two Components: Job Tracker and Task Tracker. Two type of Node: One Name node and Many Data Node.	Two Components: Job Tracker and Task Tracker. JobTracker i Split into two domains Resource Manager and Application Master.
Resources	Shared in all Nodes. Therefore a partition of the resources is very hard.	Each node has its own Resources (Like CPU, Memory etc.) so that partition of resources is easy.

3. GEOPROCESSING WORKFLOW

In recent years, the use of GIS applications has increased as the requirements for geo-spatial information services have grown. Due to this growth, there is a need of standard based geospatial data management and processing services for delivering data in organization. These standards are mainly driven by International Standards Organization (ISO) and Open Geospatial Consortium (OGC) for geospatial industries. ISO standards are mainly work in ranging from agriculture, to mechanical engineering, to information technology. And OGC is mainly focuses on standards which are used for Geospatial Information Technology. Open Geospatial Consortium (OGC) [15] is providing some standard web services such as Coordinate Transformation Service, WMS (for map representation), WCS (for images and terrain), WFS (for

retrieve and update features), WPS (for spatial models, to Interface Standard provides rules for standardizing inputs and outputs (requests and responses) for geospatial processing services.), Catalog Service (provide common interface to discover, browse, query, resources etc). ISO is providing mainly two standards for GeoProcessing viz ISO 19130 (defines sensor and data models for imagery and gridded data) and ISO 19115 (defines the schema required for describing geographic information and services.) [14].Using these standards, GIS and web client applications can directly access the web services and begin exploiting the data services over the internet. GeoProcessing is mainly used to manipulate spatial data for GIS tasks and creates geospatial information specifications based on user's specific decision making requirements. Before processing on spatial data, in some cases, a chain of OGC web service is needed for standardization of data. For example, OGC Web Coordinate Transformation Service (WCTS) and OGC Web Image Classification Service (WICS) are chained together using Business Process Execution Language (BPEL) or XML Process Definition Language (XPDL) specification.

ArcGIS, QGIS, OPENJUMP, ArcMap, ArcObject, GRASS etc [1] are providing GUIs for development of such workflow and processing of Geodata. As shown in figure 5 of GeoProcessing Workflow, first load data in the form of image which are obtained from the satellite, remote sensors and GPS devices. After loading an image in GeoProcessing workflow, we have to apply operations such as Feature Extraction, Image Enhancement, Image Classification, Image Overlapping, Image Segmentation, Image Filtering etc as per output requirement. Each operation does some specific task depending on the process. Using QGIS, something related to GeoProcessing operation. QGIS has an optional scripting support using python language. Using python plugins in QGIS, can be written/generate make own script for any operation. Also in GeoProcessing tools require all the raster-vector data related operations such as image manipulation, segmentation, analysis, filtering, geometry, feature extraction, creation, addition etc based on the given input data. Query processing using query builder and allow to define a subset of a table as like SQL – like EQUAL, WHERE etc and it results in display on main window and query result can be saved as a new vector layer or project file. In QGIS, Spatial query apply spatial operations like as equals, Disjoint, Within, Overlaps, intersects etc and store result in .qgs files. Also it can convert the image into vector and raster format. Using QGIS, read and store image/data. After applying some operation on raster or vector data and create any map then it can be stored using project file, therefore if user can open the project again and do changes based on the requirement of the output. As shown in figure 5, 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

The data consumed by a workflow is varies both in terms of volume and variety [12]. GeoProcessing was one of the original proposed when GIS was invented. Almost every GIS application is represented using GeoProcessing Workflow. It integrates data and services in an interoperable way where in each part of the workflow is responsible for only a specific task without having knowledge of the general purpose of the workflow.

To process the data which are generated at very high speed due to availability of high network bandwidth, more resources, the server side processing capacity and low latency, it is required to use web service technology. For standardization of this technology, it needs to process Geodata over the web. Schaeffer and Baranski(2012) [10] combines Grid computing and GeoProcessing workflow. They first wrap GIS application with OGC Web Processing Service (WPS) and implementation of application using grid computing for distribution of large data and convert parallel processing on it for efficient execution. They also show that wrapping GeoProcessing functionality with WPS interface is a promising concept to reuse existing and reliable functionality in a web service context.

Due to the distributed nature of geographic data, GeoProcessing Workflow provides very easy processing of highly distributed and complex data for a wide variety of applications. For improving the performance of GeoProcessing Workflow, different types of techniques are adopted. Pallickara, Malensek, M. (2011) [12] mainly focus on Geospatial data flow for processing, which enables interactions between the data and the computation/analysis which is used to processing Geodata as per GeoProcessing workflow.

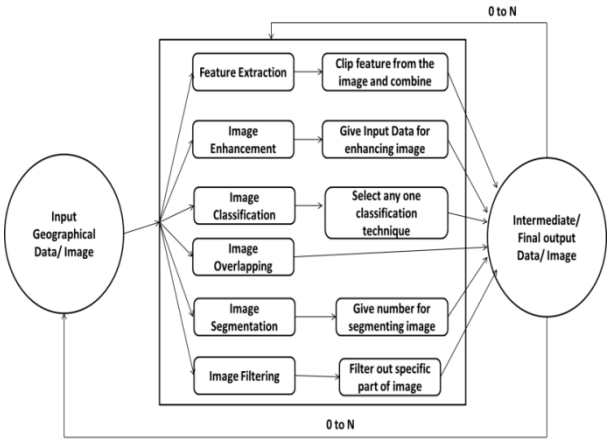

Figure 5: Schematics of GeoProcessing Workflow

The input data is used not static in structure/nature. The variation of input data depends on the rate at which the structure of data changes. Data is collected from a variety of distributed resources and transform it into a specific format according to user requirements such as transfer it into OGC standards. Well Known Text (WKT) and Well Known Byte (WKB) are very popular standards of OGC. Using Hadoop Distributed File System, a large data set is distributed dynamically among a number of nodes. Generally before process on Geographical data in GeoProcessing Workflow, some of the tasks are applied like Data collection from observational instruments, Data capture, Visualization, Data analysis [12]. For example, to perform atmospheric measurements, the collections of data from observation instruments like in-situ measurements and remote sensing instruments are required. Data capture is a challenge which involves extraction the large amount of data being generated for computations off the compute nodes at runtime and to collect data nodes for further processing such as monitoring, analysis, and archival [12]. This process is known as data capturing and use parallel processing technique at I/O layer. To visualize large volume of data using observation, simulation or experiment result and transfer 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. Malensek and Pallickara [12]. use "*VisIt*" application to implement a data flow network with parallel processing for visualization. Analysis of data use different type of parameters or variables which are related to processing of data and requirement of the output. For representing data, they are used Object based Model and Field Based Model and for multi dimensional data used N-dimensional Array Model, Which is used for geospatial data.

The data is generated very quickly 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. In Hadoop, uses 100 drivers working at the same time and hence the data of 1TB can be read in 2 minutes. 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. Also using available tools of GIS like QGIS, ArcGIS, OpenJUMP etc, the large 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 Yet Another Resource Negotiator (YARN), we can process these files at a same time and generate final output as per the user requirement in reasonable time.

Take an example of image processing for Geographical data and then apply feature extraction task on it using distributed processing framework. Mapreduce programming method can greatly reduce the processing time and therefore it can be applied to other areas of image processing as well. Tungkasthan and Premchaiswad [13] presents a framework to perform parallel processing of Content based image retrieval (CBIR) which is used for feature extraction of image. For CBIR operation, Hadoop MapReduce framework is mainly used with the intention of increasing the performance of data insertion due to large size and required more query processing on data. It is possible to perform parallel distributed processing by writing programs involving the following three steps: Map, Shuffle, and Reduce [13]. so that using Hadoop Mapreduce processing method, CBIR can be easily implemented very efficiently. As explained in section 2, for distribution of file we use Hadoop Distributed File System (HDFS) and using Yet Another Resource Negotiator (YARN), we can process these files at a same time and generate final output as per the user requirement in reasonable time.

Take an example of image processing for Geographical data and then apply feature extraction task on it using distributed processing framework. Mapreduce programming method can greatly reduce the processing time and therefore it can be applied to other areas of image processing as well. Tungkasthan and Premchaiswad [13] presents a framework to perform parallel processing of Content based image retrieval (CBIR) which is used for feature extraction of image. For CBIR operation, Hadoop MapReduce framework is mainly used with the intention of increasing the performance of data insertion due to large size and required more query processing on data. It is possible to perform parallel distributed processing by writing programs involving the following three steps: Map, Shuffle, and Reduce [13]. so that using Hadoop Mapreduce processing method, CBIR can be easily implemented.

4. CONCLUSIONS AND FUTURE SCOPE

In this paper, the distributed processing and GeoProcessing Workflow for large data set which is mostly in the form of image have been explained. For distribution of file, Hadoop Distributed File System (HDFS) and YARN help process on these files at a same time and generate final output as per the user requirement. MapReduce Classic and YARN are compared. Generic flow of GeoProcessing Framework is representing a different type of processing on image. The data which is large in a size cannot be process using GIS tools. Hadoop Distributed processing can be used to perform only image processing operations very easily. In future, we can use Spatial Hadoop specifically for the fast processing of geospatial data and handling large data sets. Spatial Hadoop has provision for indexing data before invoking HDFS process. This ensures that each node is loaded with data having same type of features and hence reduced data and time requirement of computation.

5. ACKNOWLEDGEMENT

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

6. REFERENCES

[1] 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.

[2] 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.

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

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

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

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

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

[13] Tungkasthan, A.; Premchaiswadi, W., "A parallel processing framework using MapReduce for content-based image retrieval," *ICT and Knowledge Engineering (ICT&KE), 2013 11th International Conference on* , vol., no., pp.1,6, 20-22 Nov. 2013.

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

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