Analysis of Big Data using GeoMesa

Saiyed Atufaali[1] Mr.Prashant Chauhan[2] Dr. M. B. Potdar[3]

[1]Student, GTU PG School-Ahmedabad,Gujarat,India
[2] Project Scientist, Bhaskaracharya Institute for Space Applications and Geo-Informatics, Gandhinagar,Gujarat,India
[3]Project Director, Bhaskaracharya Institute for Space Applications and Geo-Informatics, Gandhinagar,Gujarat,India

ABSTRACT

Today the term 'Big data' is coming to a fact in day to day. As the rapidly growth of data become a tangibility in many areas for that finding an efficient strategy for big data challenges and objective of improvement. Big data deals with techniques to store, analyze large amount of data which is in petabyte having velocity high. Today a massive amount of geospatial data collected, visualized and used as never before. Geospatial data is the data related to geographical location, it is usually stored as coordinates and topology and it is used for mapping. Geospatial data are first analyzed with various techniques which include its geometric structure and properties. Therefore, challenge is to process and combined the data is more complex because varied data has different properties. To analyze the geospatial data visualization tools are used. The techniques which were used before for visualizing the geospatial data are not better to handle the dataset. For finding patterns in the data, visualization makes the work easier. GeoMesa is apache open source tool which enables large amount of geospatial data on distributed environment. It has GeoServer which integrates with different range of mapping. The paper shows the challenges faced in working with GeoMesa and the Data ingestion of Vector and Raster Data in GeoMesa.

Keyword: - *Big Data, apache Hadoop, GeoMesa*

1. **Introduction**

In the today's world, Data rate are increasing due to digitalization of everything which starts from satellite images to social media posts likes comments, maintenance of medical records, police records, online shopping details, log files generated on every login of the users and data stored from many other sources for future reference. That leads the concept of big data which is one of the latest trends for future emerging technology. The demand of big data has been increasing in Government agencies, Finance, IT companies, Trade and Commerce. Big Data are characterized by 5V's Volume, Velocity, Variety, Value and Veracity. Thus, it is very important to handle big data in efficient manner and it requires special tools for proper analysis and visualization.

Therefore, data visualizing has been an integral part of every discipline particularly in GIS (Geographical Information Systems) and Remote Sensing (RS). It involves creation and study of visual representation of data. The main goal of data visualization is to communicate information efficiently through statistical and information graphics. It is one of the important steps of data analysis or data science.

In today's world of technology data visualization has become an active area of research, teaching and development. The problem of these huge data sets is to storage, analyzing, processing, sharing and visualization. The purpose of analyzing and visualizing the big data is to draw interesting patterns from these datasets. In GIS and RS, it has been found that geospatial data are playing an important role in big data visualization.

1.1 Difference Between SQL and NOSQL

In Today's world there are lots of database evolved in the industries. While having Research, world's data is exponentiality increasing day by day it is estimated that every 2 years data is doubling Some are Traditional Relational Databases, some are NoSQL Databases. Traditional Databases uses table format to represent the data structure and their relations between them. The NOSQL is the latest one that represent the relation of data other than tabular format.

Following Table 1 show the difference between SQL and NOSQL.

Table 1

Index	SQL	NOSQL
1)	Databases are categorized as Relational Database Management System (RDBMS).	NoSQL databases are categorized as Non-relational or distributed database system.
2)	SQL databases have fixed or static or predefined schema.	NoSQL databases have dynamic schema.
3)	SQL databases display data in form of tables so it is known as table-based database.	NoSQL databases display data as collection of key-value pair, documents, graph databases or wide-column stores.
4)	SQL databases are vertically scalable.	NoSQL databases are horizontally scalable.
5)	SQL databases use a powerful language "Structured Query Language" to define and manipulate the data.	In NoSQL databases, collection of documents are used to query the data. It is also called unstructured query language. It varies from database to database.
6)	SQL databases are best suited for complex queries.	NoSQL databases are not so good for complex queries because these are not as powerful as SQL queries.
7)	SQL databases are not best suited for hierarchical data storage.	NoSQL databases are best suited for hierarchical data storage.
8)	MySQL, Oracle, SQLite, PostgreSQL and MS-SQL etc. are the example of SQL database.	MongoDB, Bigtable, Redis, RavenDB, Cassandra, HBase, Neo4j, CouchDB etc. are the example of NoSQL database

1.2 Apache Hadoop

Apache Hadoop is a java-based platform which is open source distributed storage. It is distributed processing of large amount of data sets on computer clusters. If any hardware failure occurs, it is designed in such a way it handles automatically. Many organization used Hadoop to process and analyze the huge amount of data. Hadoop framework consists of two components HDFS-storage and MapReduce-processing

Files in Hadoop is split to smaller pieces (called blocks) its size is 64 MB or 128 MB, which are distributed over nodes in the cluster. To take the advantage of data processing locality, Hadoop transfers processing to the nodes according to the data to be processed in parallel. [2] The accessed data in the nodes are manipulated to allow faster

processing of the dataset and more efficiently than worked in a conventional architecture with high-speed networking. [1]

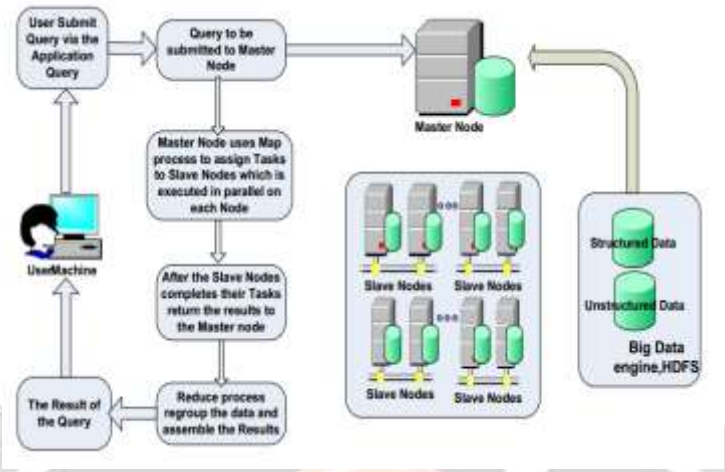

Fig. 1. Hadoop Architecture

Benefits of apache Hadoop

One of the main reason that organization started to used Hop is to that Hadoop store and process large amount of data. With increase in data volume from satellite image to social media benefits include:

Computing power: Its distributed computing model quickly processes big data. The more computing nodes you use, the more processing power you have.

Flexibility: Unlike traditional relational databases, you don't have to created structured schemas before storing data. You can store data in any format, including semi-structured or unstructured formats, and then parse and apply schema to the data when read.

Fault tolerance: Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. And it automatically stores.

Scalability and Performance: distributed processing of data local to each node in a cluster enables Hadoop to store, manage, process and analyse data at petabyte scale.

Reliability: large computing clusters are prone to failure of individual nodes in the cluster. Hadoop is fundamentally resilient – when a node fails processing is re- directed to the remaining nodes in the cluster and data is automatically re- replicated in preparation for future node failures.

Low Cost: unlike proprietary software, Hadoop is open source and runs on low- cost commodity hardware.

1.2.1 Hadoop Distributed File System

Hadoop distributed file system was derived/developed from google file system (GFS) in 2004.In the year 2004 HDFS was derived from GFS Google File system. HDFS manages to store the tremendous amount of data in efficient manner. HDFS used multiple machine to store massive files.Hadoop used HDFS for storage purpose.

HDFS is designed in such a way it provides faults tolerances.It provides high rate data transfer of data between nodes and Hadoop.

1.2.2 Map Reduce

MapReduce is a framework to process large amount of data. It is the model to compute distributed nodes. Mainly there are two important components of MapReduce Algorithm. As a input in map data set is taken and then data set is converted into another data set, splits into smaller element of data. The output of map phase is produce as key value pair. In the reduce phase shuffling and sorting process takes place to produce the result, every pair accepts only one key per time then process data per key to yield key and value pairs. [1] .The Advantage is to use map reduce is, it processes the large amount of data at the scale of multiple nodes.in the model the terminologies is use is mapper and reducer.

MapReduce's works:

Data goes to following steps:

Input splits: In the Input splits the data set is divided to smaller size parts.

Mapping: In the execution of map reduce program this phase is first.to produce the output values each split is passed to mapping function. From input splits the function of mapping is to count the occurrence of words and the result is produce in form of <word, value>.

Shuffling: This phase relocates the data. The data from mapping phase output is taken as input and centralized the relevant data split.

Reducer: This phase the whole data set is summarized. The values from the shuffling phase is combined and result is generated as single value.

1.3 Apache Accumulo

Apache Accumulo is a key Value Store Database Based On Google Big Table. It is likely to build on Model on top of Hadoop, Zookeeper.it has cell level security. And, server-side programming model.

1.4 Apache Spark

Apache Spark is technology which is designed for fast computation. It is basically build on Hadoop MapReduce. The extended part of MapReduce is spark which includes more types of computation, iterative query and stream processing. The main feature of apache spark is memory clustering which increase the speed of data processing.

2.Spatial indexing in RDBMSs

A spatial database is the database that is used for storing and querying data the represents objects defined in a geometric space [3]. Example is to find k nearest point in the space geometry. In traditional RDBMSs, the entries are stored in a table, and an additional spatial index is built separately. This index can be referenced by the database system and provides the chief means of efficiently answering queries which contain a geometric predicate [4]. Many traditional RDBMSs employ R-trees or Quadtrees for indexing, so we will recall their basic details next. PostGIS adds R-tree support to PostgreSQL.

3.Geospatial Data

Geospatial data involves the things which are related to space and time also latitude and longitude coordinates. This data is collected from the satellite imaginary and remote sensing. The increasing data rate hints upon the challenges of analyzing, managing, storing, processing and visualized the large amount of data. Geospatial data collection has been shifting from a data sparse to a data rich paradigm. [] the data sets are categorized as vector data and raster data.

3.1.1 Spatio-temporal Big Data

This section presents both vector and raster data, briefly discussing what makes each type of data challenging.

3.1.2 Vector data

Vector data comes into three types: points, lines, and polygons. The real-world features are provided by vector data within GIS environment. Due to vast increase in the number of location of remote sensing. Moving object, the majority rate grows in vector data.

3.1.3 Raster Data

Raster data is based on collection of cell in which the cell is identified by a coordinate. Raster data often used these coordinates which includes a location. Also raster data can be used as 2D or 3D for visualization.

4.GeoMesa

GeoMesa is based on Geo spatial and it is determined as data management system. GeoMesa is an open source, Spatio-temporal database and distributed system. It is built on top of cloud storage on distributed node which includes HBase, Accumulo, Cassandra and Kafka. It provides high parallelized indexing. The main of GeoMesa is to provide efficient data manipulation and querying on spatial data to key value pair data stores same as PostGIS query to the Postgres. With the integration of Geoserver with GeoMesa provide wide range of mapping clients standard OGC(Open Geospatial Consortium) APIs and protocols. Protocols defined by Geoserver such as WMS and WFS. For Analysis purpose GeoMesa facilitates support of Apache Spark for distributed data analytics.

The interface with Geoserver serve the data for analytics purpose such as analysis of time, performance evaluation, histograms, heat maps and query processing. GeoMesa provides effectively distributes the huge amount of data for the indexing with utilizing the cloud access for query processing. This indexing supports both temporal and spatial data. The novel Approach of GeoMesa which make it lead as the one of the good solution for the challenges over the big data problem in the distributed environment.

4.1 GeoServer

GeoServer is a Dynamic server which give feature for analyzing, sharing and editing the data such as Geospatial data from the different sources. The services provided by Geoserver can integrated with different applications for better visualization of maps and GIS data. It is open source server. GeoServer is the reference implementation of the Open Geospatial Consortium (OGC) Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS)

4.2Raster Support

The challenges faced by storing and managing of Billions of pixels. So GeoMesa introduced to handle the raster data by providing the storage, indexing and getting the large amount of data set and extracting the feature based on the location, geographic and spatial temporal data set.

5 Challenges

For faster execution of data, parallelization is required. The challenge over here is to break the entire big problem into many small problems so that all of them can run simultaneously. These are the following challenges which are facing by while working with GeoMesa.

- Installation of GeoMesa is most complex as it required different integration of system. GeoMesa is the most complex, built on multiple systems with many sources of latency which can be difficult to measure.
- As GeoMesa is in developing every time updates are coming.
- GeoMesa should be used in narrow cases not in enlargement cases.
- While working in the data ingestion we faced many challenges. As the data is insert into Accumulo ,the configuration of Accumulo should be properly configured otherwise error of Accumulo occur.

6 System Configuration

For this experiment we used GeoMesa 1.2.6 and GeoMesa 1.3.3 on top of Hadoop 2.7.3, Accumulo 1.6.6 and indexed only the space and time attributes of the records. By default, GeoMesa 1.2.6 creates Z2, Z3, and record tables for points; and XZ2, XZ3, and record tables for tracks. When setting up Accumulo and ingest the data using the GeoMesa. command line tools.

TABLE . ACCUMULO CONFIGURATION

Hadoop	
Zookeepers	1
Hadoop DataNode Memory	1 GB
Accumulo TabletServer HeapSize	4 GB
Accumulo Block Cache	4 GB
Accumulo Index Cache	1 GB
EBS Disk Storage	1 TB
Instance Type	r3.xlarge

6.1 Analysis

Based on experimentation, GeoMesa displayed a distinct advantage in the number of records and latency when performing larger spatio-temporal queries. As the number of records increased. First, GeoMesa's underlying datastore Accumulo, has lexical range partitioning, which enables a high-performance scalable space-filling curve index. As data volumes within an index increase, Accumulo (and other BigTable deriva- tives) are able to split the lexicographic key ranges in that index preventing any one part of the index from becoming too large and unwieldy. As tablets grow and split into smaller tablets, the key ranges are stored in the Accumulo metadata table and can be quickly looked up by querying the Accumulo master. As data scales into the terabyte range, each tablet can be operated on independently, allowing for greater parallelism across subsections of the index in a manner corresponding to the distribution of the data within the index keyrange.

Second, GeoMesa provides an 3-dimensional composite index on both space and time. This process is generally less efficient than traversing a single composite index such as GeoMesa's Z3 index, which interleaves bits of longitude, latitude, and time. By creating a single composite index, GeoMesa is able to satisfy a 3-dimensional index query over both space and time with a single pass through the 1-dimensional Z3 keyspace.

Lastly, the partitioning and composite index combine toprovide GeoMesa with more throughput from the servers to the client. Because the Z3 index in GeoMesa can be queried in a single pass and tablets operates independently of

each other, clients can return data out of order from multiple servers using multiple threads in parallel using an Accumulo BatchScanner.

6.2 Indexing

The main aim to observed query performance of GeoMesa. We analysis 4 things 1) how quickly queries came back based on the number of hits (records in the result set); 2) how many records/second were returned based on the number of hits; 3) how quickly queries came back relative to the area of the query polygons; and 4) how quickly queries came back based on the size of the temporal window used in the query filter. GeoMesa queries return much faster. even though GeoMesa's Accumulo database contained more records. Regardless, the GeoMesa run times are consistently about 1 order of magnitude faster.

We analysis the vector data, how the data are managed by GeoMesa, generate analytic result and give the performance for ingest, query and analysis. We analyses first using csv format and set the convertors in configuration file and then ingest the data in accumulo using GeoMesa command line tool. The ingestion is completed successfully. And results is shown in web interface of Accumulo.

Also we ingest the vector data in the Shape file format first ingest the data in accumulo using the GeoMesa commnad line tool.we have ingest small data and large data in case of small data the GeoMesa querying time is less within seconds But if we ingest large data it takes time in querying.

6.2.1 Ingest performance

The data uses Hadoop for storage in HDFS using MapReduce ingest in Backend GeoMesa uses GeoTool also. The ingest uses Accumulo BatchWriters instead r-file based file output format. Ingest time into Accumulo is dependent on many factors including hardware, schema design, cluster usage, and configuration. Two common methods of improving ingest speed are pre-splitting tables and introducing additional ingest clients per node. The ingest rates depend on the Features of spatial data.We did not perform an exhaustive experiment to quantify the performance of various split methodologies but instead used a generic method. It should be noted that ingest rates are extremely variable depending on the schema configuration and attribute indexing configuration.

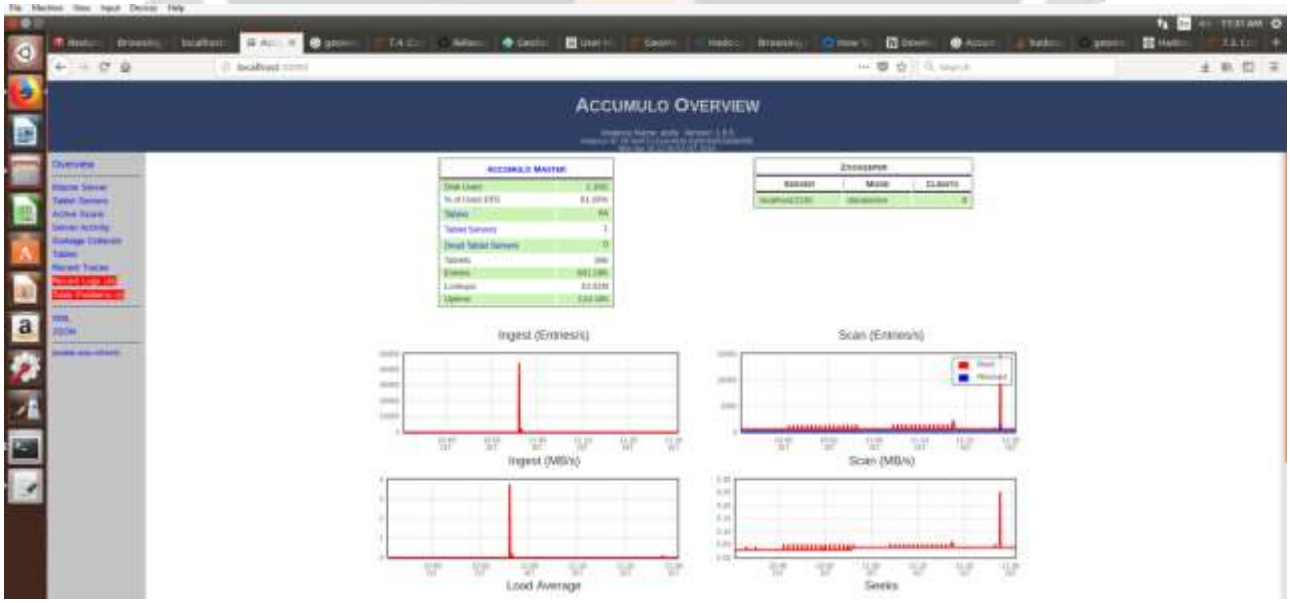

Fig. 4 Ingest performance

6.3 Ingestion of Raster data

GeoMesa's raster support is used to provide a central, scalable raster database capable of providing the multiple data nodes, each of copies of data. The request of imaginary data set parallelly evaluate the data and generation proceeds. The data are persisted as a sparse image pyramid, collections of image tiles that are grouped by image resolution. Uniquely for this application, the tiles persisted per database row are identical to the tiles requested during the rendering process, so each request scans and returns a single Accumulo entry. As a consequence

6.3.1 Performance

To ingest the raster data into GeoMesa ,we first produced the image pyramid from the file gdal_retile.py.Once the pyramid is created ,ingest the tiles into GeoMesa command line. The ingest is repeated per level of pyramid generated. Queries for individual 128 x 128 tiles were performed one thousand times to arrive at representative average timing measures for query planning, scanning, and returning the image data through WCS requests to GeoServer. To view ingested pyramid into GeoServer.

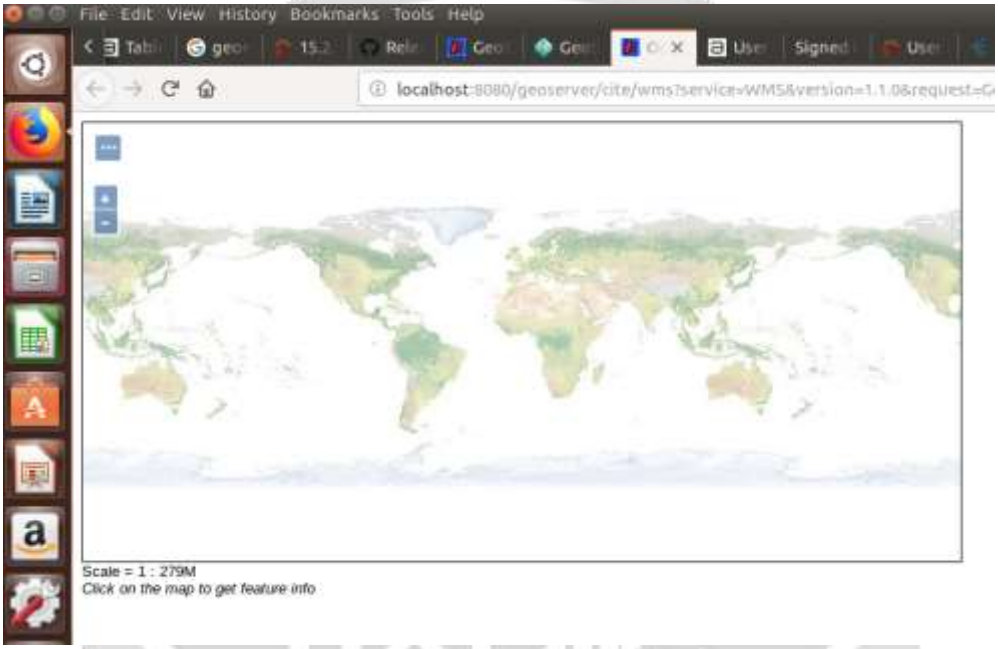

Fig. 3 Visualization of Data

7 Conclusions

The use of big data in this world of growing data is a tremendously increasing. As the rapidly growth of data become a tangibility in many areas for that finding an efficient strategy for big data challenges and objective of improvement. Big data deals with techniques to store, analyze large amount of data which is in petabyte having velocity high. In this paper, the technique used is analyze Data is GeoMesa. For the purpose of querying the high-volume data in the terabytes range, the advantages of using the GeoMesa is provides high parallelized indexing. For the high efficiency data streaming, the data storage Accumulo have been used to handle the large amount of load while mainting low latency query performance. The GeoMesa architecture handles the large amount of spatio temporal querying the data with large scale of visualization and analytics. In the future we plan to work we other database including Hbase, Kafka, Cassandra for the performance. Also, to compare the performance evaluation and query latency with other analytics tool.

Acknowledgments

We are thankful to Shri T. P. Singh, Director, BISAG, for providing infrastructure and encouragement to carry out this project at BISAG and Special thanks to Abdul Zummervala, Research Scholar, BISAG for permitting to carry out the project at BISAG

References

[1] Mohamed, Ehab, and Zheng Hong. "Hadoop-MapReduce Job Scheduling Algorithms Survey." *Cloud Computing and Big Data (CCBD), 2016 7th International Conference on*. IEEE, 2016.
[2] Fox, Anthony, et al. "Spatio-temporal indexing in non-relational distributed databases." Big Data, 2013 IEEE International Conference on. IEEE, 2013.
[3]Bhosale, Harshawardhan S., and Devendra P. Gadekar. "A review paper on Big Data and Hadoop." *International Journal of Scientific and Research Publications* 4.10 (2014): 1-7.
[4]"Hadoop Releases". apache.org. Apache Software Foundation. Retrieved 2014-12-06.
[5] "Hadoop Releases". Hadoop.apache.org. Retrieved 2015-07-29.
[6] "Welcome to Apache™ Hadoop®!". hadoop.apache.org. Retrieved 2015-09-20.
[7]"What is the Hadoop Distributed File System (HDFS)?". ibm.com. IBM. Retrieved 2014-10-30.
[8]Malak, Michael (2014-09-19). "Data Locality: HPC vs. Hadoop vs. Spark". datascienceassn.org. Data Science Association. Retrieved 2014-10-30.