Geo-spatial Big Data Mining Techniques

Mazin Alkathiri
Bhaskaracharya Institute for Space Applications and Geo-Informatics,
Gandhinagar 382007, India

Jhummarwala Abdul
Bhaskaracharya Institute for Space Applications and Geo-Informatics,
Gandhinagar 382007, India

M.B. Potdar, PhD
Bhaskaracharya Institute for Space Applications and Geo-Informatics,
Gandhinagar 382007, India

ABSTRACT
As stated in literature by several authors, there has been literally big-bang explosion in data acquired in recent times. This is especially so about the geographical or geospatial data. The huge volume of data acquired in different formats, structured, unstructured ways, having large complexity and non-stop generation of these data have posed an insurmountable challenge in scientific and business world alike. The conventional tools, techniques and hardware existing about a decade ago have met with the limitations in handling such data. Hence, such data are termed as big data. This has necessitated inventing new software tools and techniques as well as parallel computing hardware architectures to meet the requirement of timely and efficient handling of the big data. The field of data mining has been benefitted from these evolutions as well. This article reviews the evolution of data mining techniques over last two decades and efforts made in developing big data analytics, especially as applied to geospatial big data. This is still a very actively evolving field. There will be no surprise if some new techniques are published before this article appears in print.

Keywords
Data mining, Distributed Computing, Hadoop, Big Data, Geospatial, Radoop, SpatialHadoop, Hadoop-GIS, Pigeon

1. INTRODUCTION
At the end of last century, the data were commonly stored in relational databases and Structured Query Languages (SQLs) were used for information extraction from such databases. They were used extensively for development of decision support systems for managing businesses profitably as well as by governments in planning and execution of people friendly developmental programs. The data stored in the databases and data warehouses have grown fast with the increase in size of the storage media over last decade or so resulting in requirement of new techniques which can surmount the limitations of the traditional analysis techniques. Consequently, this has led to the development of new data warehousing and data mining techniques for analysis of large volumes of data. These techniques have enabled retrieval of interesting and useful knowledge.

With the scaling up of the storage and the processing capabilities, the time was ripe for emergence the concept of parallel computing. Several architectures were proposed and frameworks were developed to take advantage of recent developments of hardware. Among them, the most noteworthy are Network Compuing, Hadoop Framenwork for distributed computing, Cloud computing platforms, CUDA Computing using the array of processors in GPUs manufactured by NVIDIA, and recently developed OpenMP programming model based on Intel Xeon Phi co-processors. They use distributed data model wherein it will be possible to access many storing and processing units.

Wang et al.(1996) reported on the development of a software system for Data Mining based on the RDBMS, named as DBMiner, by integrating the three parts, viz. database, OLAP and data mining technologies. This system incorporated several interesting data mining techniques, interactively performed data mining at multiple levels of abstraction of any user specified set of data in a database or a data warehouse (Fig. 1). Efficient implementations of techniques were explored using different data structures, including multiple-dimensional data cubes having generalized relations. The data mining processes utilized user or expert defined set-grouping as well as schema-level concept hierarchies. The DBMiner tightly integrated a relational database system with a concept hierarchy module.

2. SPATIAL DATA
The Geo-Spatial data generated from are in general multi-dimensional in spatial, spectral and temporal domains. The information content in these dimensions represents real world features. The data represent either radiance or reflectance or any other physical quantity

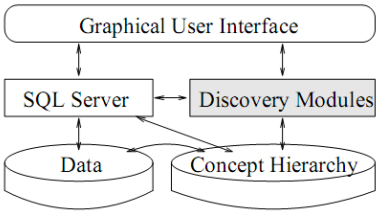

Fig. 1: Schematics Layout of DBMiner (Wang et al. 1996)

associated with a ground resolution element, called pixel. Such data are known as raster data (fig. 2). The theme based data files generated from the raster data or any other collateral data are known as vector data. Spatial data are referenced by their location co-ordinates (e.g. latitude and longitude) in the Geographic Information System (GIS) for geo-spatial analysis. The common applications of spatial data usage are for:

- Proximity assessment.
- Entity identification and estimation of likeness or similarities.
- Geometric computation and geo-spatial relationships.
- Digital representation of elevation data.
- Topological matching and pattern analysis.
- Multidimensional data representation.

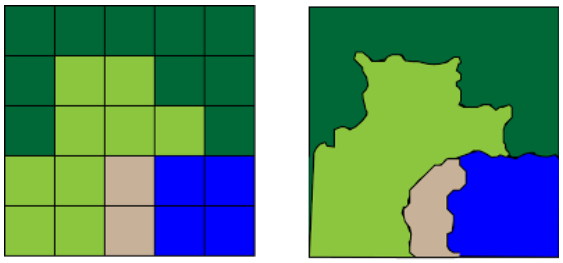

Fig.2: Representation of raster and vector data and vectorization of raster data.

3. SPATIAL DATA ANALYSIS AND MINING

The standard image processing tools and techniques are applied to extract the spatial and spectral information from the raster data and to characterize the physical, chemical, biological, geological as well as geophysical processes using multi-temporal images. There are many old examples showing that it was possible to visualize the geospatial data in a good helpful way before the computer came into existence and were subject to the data mining (DM) techniques. The Mapanalysis of Napoleon's Russian campaign (Burch and Grudnitski, 1989) and the work of Dr John Snow at the time of the great cholera epidemic in London in 1854, are well known early examples of information extraction from maps and cause effect analysis of them without the aid of computer. The Geographic Information Science (GISc) deals with the relationships among the spatial patterns and processes and their temporal dynamics. With the massive increase in the data volume due to global coverage by sensors on-board satellite systems, since the late nineteen seventies, the GISc has come to age. The massive increasing of the geographical data due to long legacy in space technology and advancements in satellite based sensor technology, satellite telemetry on one hand and the complexity of data storage and retrieval technologies, handling and analysis of large volume of data through networking of computers and parallel processing software technologies on the other hand, the issues of geographic science can now be addressed. One of such software technology field, Data Mining, offers a good solution to help converting the geographical data into information and extract knowledge out it.

Geospatial Data Mining should be understood as a special type of DM that seeks to carry out generic functions similar to those of conventional DM, thoroughly modified to safeguard the spatial aspects of geo-information. There are many definitions for spatial data analysis. Anselin (1993) definition is "the statistical study of phenomena that manifest themselves in space". Another definition by (Bailey, 1994) is "a general ability to manipulate spatial data into different forms and extract additional meaning as a result". From these definitions we get the recognition of space as a source of explanation for the patterns presented by the different phenomena within it.

The geo-spatial data mining is based on the foundation laid by the First Law of Geography, which is stated as: "*everything is related to everything else, but near things are more related than distant things*" (Anselin, 1993). Therefore, to a great extent, it is not possible to use the traditional methods of classical statistics to analyze spatial data, given the important role that location plays in understanding phenomena observed in space. Han et al. (1997) stated that the Spatial data mining as that aspect of data mining which deals with the extraction of knowledge, spatial relationships, or any hidden patterns not explicitly stored in spatial databases. The DBMiner package described above did not support this feature of spatial data mining. To deal with the huge amount of geo-spatial data, a need arose for advanced and special purpose data mining system which can extract important knowledge from both spatial and **non-spatial** objects in large database. Also there is no unique set of data mining algorithms that can be used in all application domains. But we can apply different types of the data mining algorithms as an integrated architecture or hybrid models to data sets to increase the robustness of the mining system.

GeoMiner, a spatial data mining system prototype was developed on the top of the DBMiner system(Han et al., 1997). In this system, the non-spatial data were handled by the DBMiner system, while the functions for mining spatial data and the relationships between spatial and non-spatial data were handled by GeoMiner functions. A query language, Geo-Mining Query Language (GMQL), was designed as an extension to Spatial SQL. It considered knowledge discovery from only single thematic map. Due to its obvious limitations, the need was felt to enhance its capability to deal with a very large amount of data, and handle streaming data as well.

The next stage was the development of Spatial Data Analysis and Modelling (SDAM) software system (fig. 3) that executed programs developed in different environments (C, C++, MATLAB) through a unified control and a simple Graphical User Interface (Lazarevic, Fiez, & Obradovic, 2000). The SDAM could be run on a local machine and also remotely. Data security was ensured by using passwords. The users could use learning algorithms remotely to build prediction models for each remote data set. They also presented a more advanced distributed SDAM for including model for data management over distributed sites and more complex methods for combining data classifiers.

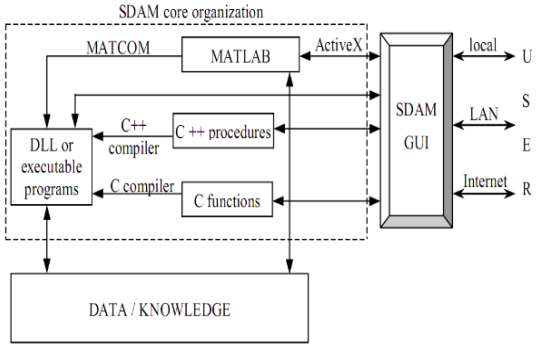

Fig. 3: SDAM under unified GUI (Lazerevic et al. 2000)

(Bação, 2006), in his thesis, proposed to divide the spatial Data Mining (DM) models used in science into three fundamental types:

- Deterministic models
- Parametric models
- Non-parametric models.

For the development of a robust theoretical framework to serve as a basis for developing a new GISc, there are two perspectives:

A. Regard DM as a "black box" within GISc 9, and

B. Use DM, but assign an important role to the pre-processing stage.

(Bação, 2006) also introduced the issues of Geospatial Data Mining (GDM) which fitted it into the broader setting of GISc and provide the framework for Geo-Spatial Data Mining (GSDM). GSDM included three generations of GIS data models are existing, viz. (i) the CAD data model, (ii) the Geo-relational data model, and (iii) the object–relation data model. The last one abstracts geographical entities as object of classes with attributes, behavior and the associated rules and relationships between objects. Some of the benefits of the Geo-database model are that they are close to human understanding and are expressions of real world objects. They also have a better capacity of expansibility, and the relationships between spatial data are fully expressed. All of spatial data together with attribute data can be stored and centrally managed in a DBMS. There was a provision to edit geographic data simultaneously by many users. Yin and Su (2006) implemented a Model for Geospatial Database as National Fundamental Geographic Information which used the DBMS Oracle9i to store geospatial objects, and the Geo-database model was applied to describe and organize the geospatial entities and their relations.

The spatial features of a particular location can be stored in the geographic databases, in which each one of those features is usually located or stored in a different relation. The process of data preparation, storing and analyzing is a time consuming task which is an issue of concern in the spatial data mining systems. One of the obvious solutions is to automate this process. Bogorny et al. (2007) presented a package named "Weka-GDPM+", which was an extension of Weka to support the automation of the spatial data preparation, storing and analyzing for mining from geographic data. The fig. 4 shows the different geographic data storage methods under the geographical database management systems (GDBMS) following the Open Geo-spatial Consortium (OGC) specifications. It also contains a knowledge repository for storing well known geographic associations extracted from geographic database schemas, Geo-Ontologies, and those provided by the users. It can seen that the first part of the diagram contains different data mining algorithms for the task of extracting the needed knowledge out of the database. The second part of this diagram is a new one which takes care of the spatial data preparation and it is located in the center to solve the problem that exists between the GDBMS and the data mining tools. The JDBC is used to reach to the data. The data is retrieved, preprocessed, and transformed into a single table format according to the user specifications.

On the top of this framework there are:

1. The Metadata retrieval module: for retrieving all relevant information from a database,
2. The Dependence **Elimination module:** for verifying all associations between the target feature type and all relevant feature types,
3. The **Spatial Join module:** for computing and materializing the user-specified spatial relationships, and
4. The **Transformation module:** for transposing as well as discretizing the Spatial Join module output into the single table representation, understandable by data mining algorithms

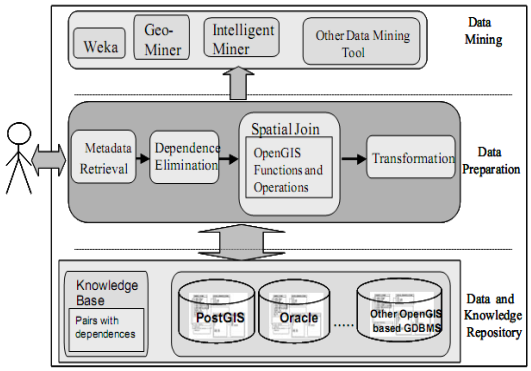

Fig. 4: Integrated spatial data mining framework (Bogorny et al. 2007)

4. GEO-SPATIAL DATA AS BIG DATA

According to some authors there have been big-bang explosion of data in recent years. The data acquisitions at the rates of terabytes per day are quite common. The spatial data on mobile phone users, rail and air travelers, marketing, consumers, goods production and wide range of daily activities are producing large volume of data which are of interest for Data Mining. The remote sensing data are no exception to these. A large number of meteorological, land and ocean observations sensors on board satellites are continuously down pouring data from space. The analyses of such large data present their own challenges, even though with highly powerful processors and high speed data access possible presently.

One of the characteristics of this volume of data generated is their formats. The data may be structured, semi-structured and un-structured formats, which further complicate their analysis. As a historical study of the term "Big Data", it is noticed that it used first time in by Mashey(1997). Diebold (2000) presented the first academic paper with the words "Big Data" in the title in 2000, and Doug Laney was the first one to introduce the 3 Vs characterizing the Big Data. The Big Data data are sets whose size (volume), complexity (variability), and rate of growth (velocity) make them complex to be collected, managed, processed or analyzed by current technologies and tools (Bhosale and Gadekar, 2014**).** The Big Data analysis require ingenious steps for data analysis, characterized by its three main components: variety, velocity and volume**"** (Sagiroglu & Sinanc, 2013**).**

Sagiroglu and Sinanc (2013) compared two of the existing big data analysis platforms, viz. Hadoop framework and High Performance Computing (HPC) clusters (HPCC).The hadoop is a known open source Java based framework, which includes, firstly, a system of several main parts starting with the distributed file system, and the secondly the system that manages the workflow and the distribution of the processes in the system. Apache Hadoop is an open source implementation of Google's MapReduce framework; it comes with its distributed file own system (HDFS), which was derived from Google File System (GFS). The HDFS has many benefits such as low cost of storing the data, data redundancy by replication, huge storage abilities, balanced utilization of storage, high fault-tolerance, high throughput rate, and scalability. These two main parts of Apache hadoop are the HDFS and MapReduce are shown in Fig. 5. The Hadoop framework also provides many other sub-projects, viz. HBase, Pig, Hive, Sqoop, Avro, Oozie, Chukwa, Flume, and ZooKeeper, each with its specific focus.

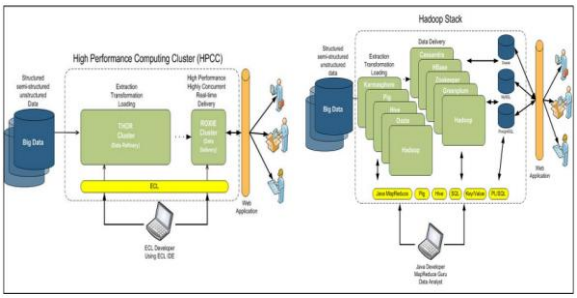

Fig. 5: Comparison of Hadoop and HPCC frameworks (Sagoroglu and Sinanc, 2013)

The HPCC Systems is also an open source, distributed, data intensive computing platform. It provides big data workflow management services. The HPCC data model is defined by the user. The three main HPCC components are: HPCC Data Refinery (Thor), HPCC Data Delivery Engine (Roxie), and Enterprise Control Language (ECL). Here, in this paper, the Hadoop framework is dealt with in more details; especially in context of geo-spatial data analysis.

5. HADOOP FRAMEWORK FOR SPATIAL DATA ANALYSIS

The main concept of the framework is segregated in two parts, viz. the Hadoop Distributed File System (HDFS) for storing data and the MapReduce programming model to process the data (fig. 6). It is a distributed data processing framework. It was first developed by Google and later it was adopted by Apache Foundation. It is capable of managing large volume data efficiently, using a large number of commodity hardware as Nodes forming a hadoop cluster. The MapReduce model of data analysis is composed of map and reduce functions. The concepts of the framework are based on divide and conquer rule, in which, the big volume data are divided into small clusters and processed parallel as SIMD or MIMD techniques. The map function reads data and converts it into key-value pairs. Later the reduce function, after shuffling the key-value pairs are converted into key-multiple values pairs. A job is divided into the following five tasks: (1) Iteration over the input data, (2) Computation of key/value pairs from each of input data, (3) Grouping of all intermediate values by key, (4) Iteration over the resulting groups, and (5) Reduction of each group.

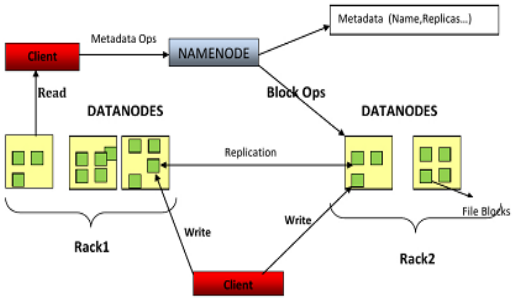

Fig. 6: Architecture of Hadoop Distributed File System (HDFS) (Source: http://hadoop.apache.org)

Hadoop can be used for many applications involving large volume of Geospatial as well as Spatio-temporal data analysis, Biomedical Imagery analysis, and even for simulation of various physical, chemical and computationally intensive application biological processes. However, in this paper, we emphasize the spatial usage of hadoop framework.

Economides et al. (2013) presented a new Spatial Database Design or Spatial Database Management System (SDBMS) named **MIGIS** which is a Hadoop based framework for handling efficiently complex spatial queries with high performance. This Hadoop framework is further extended by YSmart and RESQUE. A typical database needs an index in order to achieve fast query processing. The R-Tree based indices group nearby entities and represent them using their minimum bounding rectangle. The R-Tree can be built in three-phases, in the first phase; each of close input is divided according to the size properties. In the second phase, hadoop produces lower level R-Trees processes out of the divided data. And in the last phase, the lower level R-Trees are combined to form complete R-Tree index of the data.
Cary et al. (2009) showed how MapReduce model solved following two typical and representative spatial data processing problems fast and efficiently:

- The large scale construction of R-Trees, a popular indexing mechanism for spatial search query processing, and
- Processing digital aerial imagery and computation and storing image quality characteristics as metadata.

The key contribution of this work was presentation of a technique for large scale building R-Trees based on the MapReduce model, and also to show how MapReduce can be applied to big parallel processing of raster data. They also evaluated the algorithms performance using different matrices.

The algorithm had following three phases

(1) **Computation of partitioning function (f)**: It is a function calculated quantity using the inputs for this phase which are the data set and a positive number R, which represents the number of partitions.

(2) **R-Tree construction:** The partitioning function (f**)** calculated in the first phase is used by Mappers to divide the data set into R partitions. Also build R-Tree indices in each of the input partitions.

(3) **R-Tree consolidation**: This phase combines the R individual R-Trees, built in the second phase, under a single root node to form the final R-Tree index of the data set.

For supporting spatial queries on Hadoop, many new techniques have come to the surface. But, most of them require internal modifications of frameworks. Lee et al. (2014) have come up with a spatial index especially for big data, based on a hierarchical spatial data structure stored in distributed file storage systems. This spatial index has several advantages; such as: it can be implemented without changing the internal implementation of the existing storage systems, simple and efficient filtering, and supports to updates of spatial objects.

For this spatial indexing, a hierarchical spatial data structure, called geo-hash is used. It is a geo-coding system of latitudes and longitudes. The longer the geo-hash code the smaller the bounding. There are two categories of the existing spatial query processing techniques, viz:

- In the first, small representatives of spatial objects are used like the k-Nearest Neighbour (kNN) selection queries.

- In the second, the whole data set is used for the querying purpose. This type of query called low selectivity query.

The other basis queries for mainly spatial data are:

- **Containing:** This query returns all spatial objects containing the given search geometry.
- **ContainedIn:** This query returns all spatial objects contained by the given search geometry
- **Intersects:** This query returns all spatial objects that intersecting a given search geometry.
- **WithinDistance:** This query (or also called range query) returns all spatial objects within a given distance from a given search geometry.

The Spatiotemporal data is one type of the spatial data that also can have a very big size and is a problem while analyzing the data. Also, the very large volume of the spatiotemporal data generated by different social media networks is very important and useful in various domains like the commercial domain or more importantly in the disaster mapping or national security applications. Also much of the huge amount of data that Google generates is spatiotemporal data. To process such type of large volumes of data, there is a need for research in developing efficient management techniques and analytical infrastructure for such massive to big spatial data.

One of the main requirements of any data mining algorithm is that it should take into account the spatial and temporal autocorrelations existing, if any. The explicit modeling of spatial dependencies increases computational complexity. The Data mining primitives that explicitly model spatial dependencies are (Vatsavai et al., 2012):

- **Spatial Autoregressive (SAR)** model in which for the prediction, often spatial dependencies are modeled using regression technique,
- **Markov Random Field (MRF**) model in which the spatial dependencies in classification are often modeled as the extension of a priori probabilities in a Bayesian classification framework, and
- **Gaussian Processes Learning and Mixture** Models for modeling spatial heterogeneity in classification of large geographic regions. Used also in change detection studies**.**

There are many examples of applications which deal with big geospatial data. For example, the application on Biomass Monitoring requires high temporal resolution satellite remote sensing imagery. The MODIS instrument on board NASA's Terra satellite is providing a opportunity for continuous monitoring of biomass over large geographic regions. Since the data at global scale is difficult to handle due to its large volume and formats complexity, the data from the MODIS sensor is organized into tiles of $10^o \times 10^o$latitudes and longitudes (4800 x 4800 MODIS pixels). Also, the computational complexity of change detection algorithms is very high due to varying atmospheric contributions, varying sensor parameters, sensors look angles and scan angles, in addition, season dependent solar illumination levels.

The second example, in the above category, can be searching for complex patterns in geospatial data**.** Most of the pattern recognition as well as the machine learning algorithms are per-pixel. Such methods work well for thematic classification of moderate to high to very high resolution (pixel size of 5 meters and less) images. The Very High Resolution (VHR) images allow recognition of structures in the images.

More examples applications are:

- Recognizing complex spatial patterns in an urban areas to map informal and formal settlements,
- Recognizing critical infrastructure establishments (such as nuclear, thermal, and chemical plants, airports, shopping and sports complexes), and
- Image based content search and retrieval on applications of classification and clustering techniques.

Above tasks require feature extraction and selection, indexing, machine learning, pattern matching etc. Most of these tasks often deal with segments and objects instead of pixels. Computation parameter of similarity between image patches (Housdorff distance) is computationally expensive. The large scaling of these algorithms (to global applications) requires efficient and novel algorithmic solutions, and also require exascale computing infrastructure to support large scale to global spatiotemporal applications. In such situations, the distributed and parallel computing techniques, such as Apache hadoop framework, CUDA computing architecture and MPI based architecture etc., come handy.

There are many data mining packages are in vogue and quite popular among the data miners. To name some most popular packages are Open Office, WEKA, KNIMIE, and RapidMiner. The Rapid Miner has a clear and user friendly interface that makes it easy to learn. It enables to design the data mining work flow visually and investigate results while execution. Though, the above packages are highly rich in functionalities and visualization tools, but they have limitations in terms of size of data being handled and efficiency of large scale data processing.

6. RADOOP

Prekopcsak etal.(2011) added Hadoop extension to RapidMiner, calling the extended version as **Radoop** (fig. 7). The extension was designed to achieve a close integration of hadoop and rapid miner and provide hadoop functionalities commonly used in memory-based RapidMiner processes. Also it keeps the convenient RapidMiner features like metadata transformations and breakpoints.

The user interface is one important part of any platform or framework, and it should be a user friendly and make the interaction of the user with the platform easier and faster. In the case of the Hadoop revolution many distributed analytic systems have been developed that are strong and fault-tolerance and support many other features. But these systems are usually very hard to use and interact with them.

Radoop creates a process by adding the RadoopNest meta-operator containing general settings for the cluster (such as the IP address of the Hadoop master node), and all other Radoop operators are used inside this meta-operator. Prekopcsak et al. (2011) describe data handling and the several possibilities of uploading the data to a cluster for processing as well as the data preprocessing and modeling operators of Radoop.

For the data handling in RapidMiner, the data tables are objects of ExampleSet and normally stored in memory. In Radoop, the data tables are stored in Apache Hive and use HadoopExampleSet object to describe the data stored. The HadoopExampleSet not only stores several pointers and

settings, but all data is stored in Hive in the distributed file system resulting in no significant memory consumption during Radoop based data processing. Also, the Reader and Writer operators are implemented to enable transfer of large data files from right inside RapidMiner. These operators work with parsing every row in the dataset which has an overhead. So the big CSV (Comma Separated Variable) formatted files can be uploaded to HDFS and also loaded to Hive using the command line interface. The Store and Retrieve are extremely powerful operators to write and read back intermediate results.

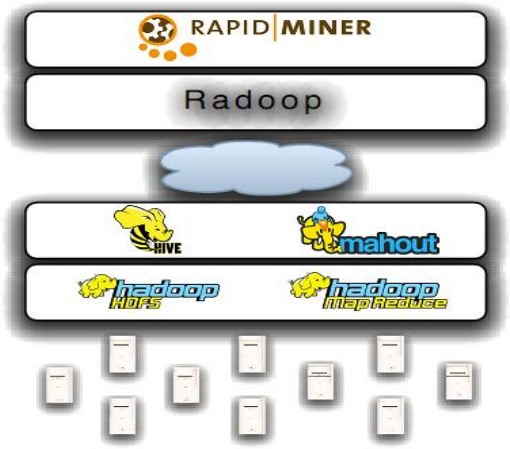

Fig. 7: Radoop Architecture (Prekopcsak et al. 2011)

Radoop has a built in excellent powerful mechanism for the data transformations using views in Hive Query Language (HiveQL). It was expensive data writing in the results table to the distributed file system only when needed. Radoop supports many data transformations, such as selecting attributes, generating new attributes, filtering examples, sorting, renaming, type conversions, and many more, that can be expressed as HiveQL scripts.

7. HADOOP-GIS

The next major development brought about by Aji et al. (2013) is the development of Hadoop-GIS, basically integrating the Geographic information System (GIS) with HDFS file system for distributed data processing. The architecture of Hadoop-GIS is shown in fig. 8. It underpins various sorts of spatial queries on Hadoop-MapReduce system through spatial partitioning, adjustable Real-time Spatial Query Engine **(**RESQUE). It uses global partition and adaptable local spatial indexing to accomplish the efficient query processing. It incorporates Hive to boost declarative spatial inquiries with an integrated construction modeling and accessible as an arrangement of library for handling spatial inquiries.

Some of the salient features of the Hadoop-GIS system are given here. There are five major categories of queries which can be summarized as follows:

1. **Feature aggregation queries** (non-spatial queries), for example, queries for finding mean values of attributes or distribution of attributes,
2. **Fundamental spatial queries,** such as point based queries, containment queries and spatial joins,
3. **Complex spatial queries,** such as spatial cross-matching or overlay (large scale spatial join) and nearest neighbor queries,
4. **Integrated spatial and feature queries,** for example, feature aggregation queries in a selected spatial regions, and
5. **Global spatial pattern queries,** include queries on finding high density regions, or queries to find directional patterns of spatial objects.

Traditional Methods for Spatial Queries have the following limitations:

1. Managing and querying spatial data on a massive scale,
2. Reducing the I/O bottleneck by partitioning of data on multiple parallel SDBMSs disks,
3. Optimzing for computationally intensive operations such as geometric computations,
4. Locking of effective spatial partitioning mechanism(s) to balance data loads and task loads across database partitions, and
5. High data loading overheads.

7.1 Resque

It stands for **RE**al-time **S**patial **QU**ery **E**ngine. It takes advantage of global tile indexes and local indexes which exist to support more efficient spatial queries. It is completely advanced engine. It underpins the data compression and low overheads on data loading. It is compiled as a shared library which can be effectively conveyed in a cluster environment. To handle boundary objects in a parallel query processing situation two methodologies are proposed, viz. (i) Assignment and (ii) Multiple Matching. The multiple task methodology is easy to implement and fits well in the MapReduce programming model.

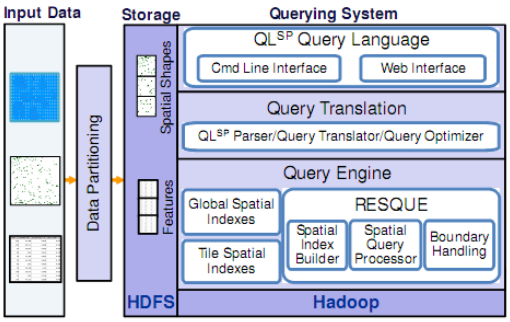

Fig. 8: Architecture of Hadoop-GIS Hive (Aji etal. 2013)

7.2 Data Partitioning

It is a starting step to define, produce and show partitioned data. In spatial data partitioning, focus is on dividing high density parts into smaller density ones. For boundary intersecting objects, taking multiple assignments based approach, in which objects are replicated and assigned to each intersecting tile, a model is proposed as follows:

1. First Count the number of objects in each tile, and sort them based on the counts.
2. Define a threshold Count Max (Cmax) as the maximal count of objects allowed in a tile.

3. Pick all tiles with object counts larger than Cmax.
4. Split each of them into two equal half-sized tiles.
5. This process is repeated until all tiles have counts lower than Cmax.

After partitioning, each object in a tile is assigned a corresponding tile Unified ID (UID). Maximum Bounding Regions (MBRs) of tiles are maintained as a global spatial index. Before loading into HDFS, all tiles are merged into a large file instead of storing each tile as a separate file. For objects across tile boundaries, i.e. intersecting boundaries, the multiple assignment approach is adopted, wherein an object intersecting with the tile boundary is assigned multiple times to all the intersecting tiles. The query results are normalized through additional boundary handling process.

7.3 Integration with Hive

To provide an integrated query language and unified system on MapReduce, the Hive is extended to HiveQL, with spatial query support for spatial constructs, spatial query translation and their execution. The HiveQL is an integrated version of Hadoop-GIS. There are several core components in HiveQL to provide spatial query processing capabilities, such as:

1. Spatial Query Translator, which parses and translates SQL queries into an abstract syntax tree,
2. Spatial Query Optimizer, which takes an operator tree as an input and applies rule based optimizations, and
3. Query Engine to support the following infrastructure operations: spatial relationship comparison, spatial measurements, and spatial access methods for efficient query processing.

7.4 Query Processing

Hive uses the traditional "plan-first-execute-next" approach for query processing. This consists of three steps: (1) query translation, (2) logical plan generation, and (3) physical plan generation. The main differences between Hive and Hive^{SP} are in the logical plan generation. If the query contains spatial operations, the logical plan is regenerated with special handling of spatial operators. Specifically, following two additional steps are performed to rewrite such a query.

1. First step: The operators involving spatial operations are replaced with internal spatial query engine operators at tile level query processing.
2. Second step: The serialization/de-serialization operations are added before and after the spatial operators, respectively, to prepare Hive query for communicating with the spatial query engine.

8. SPATIALHADOOP

It is a parallelly developed hadoop infrastructure for handling spatial data (Eldawy, 2014). Three system prototypes of the spatial hadoop are proposed: (i) **Parallel-Second:** It is a parallel spatial DBMS that uses Hadoop as a distributed task scheduler, (ii) **MD-HBase:** This extends HBase, a non-relational database which runs on top of Hadoop to support multidimensional indexes, and (iii) **Hadoop-GIS:** To extend Hive, a data warehouse infrastructure built on top of Hadoop with a uniform grid index to range queries and self-join queries. However, the main drawback of the systems discussed above is the lack of integration with the core of Hadoop (**Eldawy, 2014).**

The **SpatialHadoop** has been built-in Hadoop for improving the query processing. It is achieved by introducing spatial constructs into the core of Hadoop. It introduces standard spatial indexes and MapReduce components allowing researchers and developers to implement new spatial operations efficiently in the system, and providing more options including Grid File, R-tree and R+tree. One of the main features is that it deal with Pigeon; an extended version of the Pig for dealing with the spatial data.

In the core of SpatialHadoop, the Grid File, R-tree and R+tree indexes are adapted to the Hadoop Distributed File System (HDFS) by building them as a global index by partitioning data across nodes and multiple local indexes to organize records inside each node. The new indexes are made accessible to MapReduce programming model through two new components, viz. the SpatialFileSplitter and the SpatialRecordReader. SpatialHadoop is also used in three live systems, viz. (1) MNTG, a web service for generating traffic data, (2) TAREEG, a web-based extraction tool for OpenStreetMap data, and (3) SHAHED, a system for analyzing satellite data from NASA(Eldawy & Mokbel, 2013).

The SpatialHadoop's core consists of the following four layers:

- **Language layers:** These are simple high level language provided to simplify spatial data analysis for non-technical users.
- **Storage layers**: In this a two-layered spatial index structure is provided.
- **MapReduce layers**: These have two new components added to allow the MapReduce programs to access indexed files as input, viz. SpatialFileSplitter and SpatialRecordReader.
- **Operations layers**: These contain a number of spatial operations (range query, kNN and spatial join) implemented using the above mentioned indexes and new components in the MapReduce layer. Other spatial operations can be added in a similar way.

Basic Operations of the SpatialHadoop are:

- **Range Query:** It takes a set of spatial records R and a query area A as inputs, and returns the records that overlap with A.
- **Spatial Join:** It takes two sets of spatial records, e.g. R and S, and a spatial join predicate θ (e.g., touches, overlaps, or contains) as input, and returns the set of all pairs hr, si where $r \in R$, $s \in S$, and the join predicate θ is true for hr, si.
- **CG_Hadoop:** It is a suite of computational geometry (CG) operations for MapReduce supporting following five fundamental computational geometry operations, viz:
  - polygon union,
  - skyline,
  - convex hull,
  - farthest pair, and
  - closest pair.

8.1 Pigeon

Many SQL-like high level languages, such as Pig Latin, HiveQL and Y-Smart were introduced to simplify big data processing with Hadoop. To analyze large amount of spatial data on hadoop, many spatial operations were implemented as map-reduce functions; such as range query, kNN, spatial join and kNN join. But it is still not easy for Hadoop to analyze the big geospatial data, because the SQL-like languages which can't deal with spatial data types derive results from spatial data.

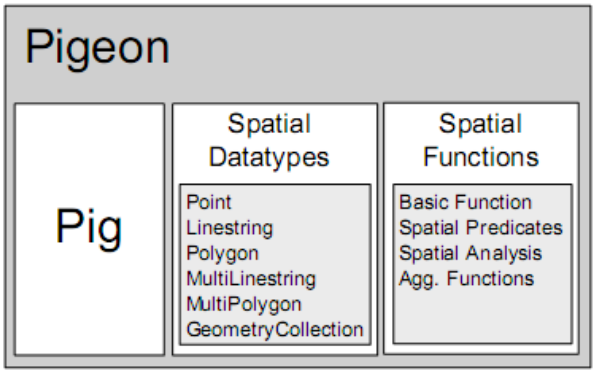

Fig. 9: Pigeon as an extension of Pig by adding spatial data types and functions (Eldawy and Mokbel, 2014)

Pigeon, as a spatial extension to Pig, introduces the spatial functionality in Pig for Hadoop and shown in fig. 9 (Eldawy and Mokbel, 2014). That is achieved through user defined functions (UDFs) making it easy to reuse and compatible with all recent versions of Pig. This also allows it to be integrated with existing non-spatial functions and allowed to use existing operations such as Filter, Join and GroupBy. Pigeon provides support to the standard OGC (Open Geo-spatial Consortium) data types, and can import spatial objects stored in the OGC compatible Well-Known Text (WKT) and Well-Known Binary (WKB) formats. To export the value of a spatial data, it provides three methods: AsBinary, AsText and AsHex to convert a spatial object back to one of the three standard formats.

8.1.1 Spatial Capabilities

Pigeon is based on the extensibility of Pig to execute the spatial tasks as UDFs which makes it simple to install and use in a current Pig. Pigeon gives four classifications of spatial capabilities:

(1) **Basic spatial functions** for retrieving basic information of a single spatial object (such as the perimeter length or area).

(2) **Spatial predicates** takes one or two spatial objects and returns a Boolean value based on the relationship of the input object(s)

(3) **Spatial analysis** for performing some spatial transformation on spatial objects. Some functions are unary (e.g., Centroid) while others are binary (e.g., Intersection).

(4) **Aggregate functions** takes a set of spatial objects and returns a single value that summarizes all the input; (e.g. the function ConvexHull returns one polygon that represents the minimal convex hull of all input spatial objects).

8.2 APACHE MAHOUT

The word mahout means elephant driver in Hindi. It was a subproject of Apache's Lucene project in 2008 providing the well-known open source search engine. It provided advanced implementations of search, text mining, and information-retrieval techniques. In 2010, Mahout became a top-level Apache project with Java based framework meaning it has a platform-independent feature. So, it can be used with any platform that can run an updated JVM. Apache Mahout has following three defining qualities:

1. It has an Open source machine learning library.
2. It is Scalable when the collection of data to be processed grows very large
3. It has a Java library.

However, it doesn't provide a user interface or a prepackaged server or an installer.

Some of the techniques and algorithms Mahout includes are still under development or in a trail phase. Three core functions are available, viz. recommender engines (collaborative filtering), clustering, and classification. These techniques works best when provided with a large amount of good input data (Owen et al. 2011). Hadoop implements Mahout with MapReduce paradigm. It does the following:

- manages storage of the input data,
- manages partitioning,
- data transfers between node machines, and
- the detection and recovery from individual machine failures.

8.3 MRPrePost

It is parallel algorithm developed by Liao et al. (2013) for mining big data. For mining big data, there are two groups of algorithms; the first one works by repeatedly scanning the database to prune candidate sets (called Apriori-like algorithm), and the other group is called Frequent Pattern Growth (FP-growth) algorithm. But both are not well adapted with large data. A hybrid of the Dis-Eclat (Moens et al., 2013) basis Pre-Post algorithm, which is for mining big data and is based on hadoop called MRPrePost as proposed by Liao et al., (2014). The PrePost process needs to scan a database twice to construct a PPC-Tree. In the mining process, it only needs to intersect the merger N-list. N-List is a data structure introduced in MRPrePost algorithm by Liao et al. (2013). Each element of N-list composed by PrePost-Code (PP-Code) is called after the sequence encoding the preamble. PPCTree node consists of five components; viz. item-name: representing the node name, count: representing the node count, children-list: representing collection of the child nodes, pre-order: representing order of node, and post-order: representing the order of node when post order.

The Data mining process is divided into three stages by MRPrePost algorithm: (l). Statistic stage, where each node independently perform map function, reduce function combined statistical results, (2). Using a method similar to building FP-Tree to build PPC Tree, traverse and generate N-List frequent one set; and (3). The search space division stage where the N-list is distributed to each worker nodes, in order to ensure the cluster load balance

.

9. SUMMARY

Over last two decades, the techniques of data mining have undergone big transformation to realize the value of big data, including geo-spatial data. In the late nineties, efforts were made to extend database mining techniques to spatial data. Extension of DBMiner to GeoMiner and SDAM are the examples. Radoop, an extension of RapidMiner is another example. Later, efforts were also been made to extend the primitive data mining software to take advantage of parallel processing paradigm. Currently, the emphasis is on using the High Performance Computing (HPC) systems using a computing network and distributed computing techniques, such as Hadoop framework. Also, the GPUs and array processors like Intel Xeon Phi co-processor are put to use to achieve the parallel processing. The new techniques of data warehousing (Apache Hive), querying (Resque, Pig and Pigeon) and mining (Apache Mahout) of distributed file systems have been either under active development or are in the near mature states. Mahout on Spatial-Hadoop, CG-Hadoop and MRPrePost extend the parallel processing to spatial data. Mahout offers some of the most important data mining functionalities. But, lot need to done to develop Mahout into a full-fledged data mining package.

10. ACKNOWLEDGMENTS

We are grateful to Shri T. P. Singh, Director, BISAG for his keen interest in and support to this work.

11. REFERENCES

[1] Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X., & Saltz, J. (2013). Hadoop GIS: a high performance spatial data warehousing system over mapreduce. Proceedings of the VLDB Endowment, 6(11), 1009-1020.

[2] Bação, F. L. (2006). Geospatial Data Mining. ISEGI, New University of Lisbon.

[3] Bhosale, H. S., & Gadekar, D. P. (2014). A Review Paper on Big Data and Hadoop.

[4] Bogorny, V., Kuijpers, B., Tietbohl, A., & Alvares, L. O. (2007). Spatial data mining: From theory to practice with free software. Paper presented at the Proc. of WSL International Workshop on Free Software (WSL'07).

[5] Cary, A., Sun, Z., Hristidis, V., & Rishe, N. (2009). Experiences on processing spatial data with mapreduce. Paper presented at the Scientific and statistical database management.

[6] Diebold, F. (2000). Big data dynamic factor models for macroeconomic measurement and forecasting. Discussion read to the 8th World Congress of the Econometric Society, Seattle, August.

[7] Economides, G., Piskas, G. and Siozos-Drosos, S. (2013). Spatial Data and Hadoop Utilization.

[8] Eldawy, A. (2014). Spatialhadoop: towards flexible and scalable spatial processing using mapreduce. Paper presented at the Proceedings of the 2014 SIGMOD PhD symposium.

[9] Eldawy, A. and Mokbel, M. F. (2013). A demonstration of spatialhadoop: an efficient mapreduce framework for spatial data. Proceedings of the VLDB Endowment, 6(12), 1230-1233.

[10] Eldawy, A., & Mokbel, M. F. (2014). Pigeon: A spatial mapreduce language. Paper presented at the Data Engineering (ICDE), 2014 IEEE 30th International Conference on.

[11] Han, J., Koperski, K., & Stefanovic, N. (1997). GeoMiner: a system prototype for spatial data mining. Paper presented at the AcM sIGMoD Record.

[12] Lazarevic, A., Fiez, T., & Obradovic, Z. (2000). A software system for spatial data analysis and modeling. Paper presented at the System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference.

[13] Lee, K., Ganti, R. K., Srivatsa, M., & Liu, L. (2014). Efficient spatial query processing for big data. Paper presented at the Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.

[14] Liao, J., Zhao, Y., & Long, S. (2014). MRPrePost—A parallel algorithm adapted for mining big data. Paper presented at the Electronics, Computer and Applications, 2014 IEEE Workshop.

[15] Mashey, J. R. (1997). Big Data and the next wave of infraS-tress. Paper presented at the Computer Science Division Seminar, University of California, Berkeley.

[16] Moens S., Aksehirli E., Goethals B., 2013, Frequent Itemset Mining for Big Data, IEEEInt. Conf. on Big Data, IEEE, 2013, 111-118.

[17] Owen, S., Anil, R., Dunning, T., & Friedman, E. (2011). Mahout in action: Manning Shelter Island.

[18] Prekopcsak, Z., Makrai, G., Henk, T., & Gaspar-Papanek, C. (2011). Radoop: Analyzing big data with rapidminer and hadoop. Paper presented at the Proceedings of the 2nd RapidMiner Community Meeting and Conference (RCOMM 2011).

[19] Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. Paper presented at the Collaboration Technologies and Systems (CTS), 2013 International Conference on.

[20] Vatsavai, R. R., Ganguly, A., Chandola, V., Stefanidis, A., Klasky, S., & Shekhar, S. (2012). Spatiotemporal data mining in the era of big spatial data: algorithms and applications. Paper presented at the Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data.

[21] Wang, JHYFW., Koperski, JCWGK., Li, D., Stefanovic, YLARN., & Zaiane, BXOR. (1996). DBMiner: A system for mining knowledge in large relational databases. Paper presented at the Proc. Intl. Conf. on Data Mining and Knowledge Discovery (KDD'96).

[22] Yin, H.-m., & Su, S.-w. (2006). Modeling for geospatial database of national fundamental geographic information. Paper presented at the Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on.