RESEARCH ARTICLE OPEN ACCESS

Big data Mining Using Very-Large-Scale Data Processing Platforms

Ms. K. Deepthi***,** Dr. K. Anuradha******
*****(Assistant Professor, SNTI, CSE Dept, Hyderabad, TS, India)
**(Professor, HOD, GRIET, CSE Dept, Hyderabad, TS, India)

ABSTRACT
Big Data consists of large-volume, complex, growing data sets with multiple, heterogenous sources. With the tremendous development of networking, data storage, and the data collection capacity, Big Data are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences. The MapReduce programming mode which has parallel processing ability to analyze the large-scale network. MapReduce is a programming model that allows easy development of scalable parallel applications to process big data on large clusters of commodity machines. Google's MapReduce or its open-source equivalent Hadoop is a powerful tool for building such applications.
Keyword **-** FastBit, GraphLab, Hadoop, MapReduce, PEGASUS.

I. INTRODUCTION

Big data sure involves a great variety of data forms: text, images, videos, sounds, and whatever that may come into the play, and their arbitrary combination. Big data frequently comes in the form of streams of a variety of types. Time is an integral dimension of data streams, which often implies that the data must be processed/mined in a timely or (nearly) real-time manner. Besides, the current major consumers of big data, corporate businesses, are especially interested in "a big data environment that can accelerate the time-to-answer critical business questions that demonstrate business values" The time dimension of big data naturally leads to yet another key characteristic of big data – speed or velocity. Gartner analysts described the dominant characteristics of big data as "three Vs" – Volume, Velocity, and Variety. Serious challenges are unfolded along each of the "V" axis. Scalability is at the core of the expected new technologies to meet the challenges coming along with big data. The simultaneously emerging and fast maturing cloud computing technology delivers the most promising platforms to realize the needed scalability with demonstrated elasticity and parallelism capacities.

Google's novel programming model, MapReduce, and its distributed file system, GFS (Google File System), represent the early groundbreaking efforts made in this line. From the data mining perspective, mining big data has opened many new challenges and opportunities. Even though big data bears greater value (i.e., hidden knowledge and more valuable insights), it brings tremendous challenges to extract these hidden knowledge and insights from big data since the established process of knowledge discovering and data mining from conventional datasets was not designed to and will not work well with big data. The cons of current data mining techniques when applied to big data are centered on their inadequate scalability and parallelism.

In general, existing data mining techniques encounter great difficulties when they are required to handle the unprecedented heterogeneity, volume, speed, privacy, accuracy, and trust coming along with big data and big data mining. Improving existing techniques by applying massive parallel processing architectures and novel distributed storage systems, and designing innovative mining techniques based on new frameworks/platforms with the potential to successfully overcome the aforementioned challenges will change and reshape the future of the data mining.

II. DATA MINING

Knowledge discovery (KDD) is a process of unveiling hidden knowledge and insights from a large volume of data, which involves data mining as its core and the most challenging and interesting step. Typically, data mining uncovers interesting patterns and relationships hidden in a large volume of raw data, and the results tapped out may help make valuable predictions or future observations in the real world. Data mining has been used by a wide range of applications such as business, medicine, science and engineering. It has led to numerous beneficial services to many walks of real businesses – both the providers and ultimately the consumers of services. Applying existing data mining algorithms and techniques to real-world problems has been recently running into many challenges due to the inadequate scalability (and other limitations) of these algorithms and techniques that do not match the three Vs of the

emerging big data. Not only the scale of data generated today is unprecedented, the produced data is often continuously generated in the form of streams that require being processed and mined in (nearly) real time. Delayed discovery of even highly valuable knowledge invalidates the usefulness of the discovered knowledge. Big data not only brings new challenges, but also brings opportunities – the interconnected big data with complex and heterogeneous contents bear new sources of knowledge and insights. Big data would become a useless monster if we don't have the right tools to harness its "wildness". We argue to consider big data as greatly expanded assets to human. All what we need then is to develop the right tools for efficient store, access, and analytics (SA2 for short). Current data mining techniques and algorithms are not ready to meet the new challenges of big data. Mining big data demands highly scalable strategies and algorithms, more effective preprocessing steps such as data filtering and integration, advanced parallel computing environments, and intelligent and effective user interaction. Next we examine the concept and big data and related issues, including emerging challenges and the (foregoing and ongoing) attempts initiated on dealing with big data.

III. BIG DATA

Organizations have already started to deal with petabyte-scale collections of data and they are about to face the exabyte scale of big data and the accompanying benefits and challenges. Technology revolution has equipped millions of people the ability to quickly generate tremendous stream data at any time and from anywhere using their digital devices besides, remote sensors have been ubiquitously installed and utilized to produce continuous streams of digital data. Massive amounts of heterogeneous, dynamic, semi-structured and unstructured data are now being generated from great diverse sources and applications such as mobile-banking transactions, calling-detail records, online user-generated contents (e.g., tweets, blog posts, keeks videos), online-search log records, emails, sensor networks, satellite images and others.

3.1 Big Data Mining

The goals of big data mining techniques go beyond fetching the requested information or even uncovering some hidden relationships and patterns between numeral parameters. Analyzing fast and massive stream data may lead to new valuable insights and theoretical concepts. Comparing with the results derived from mining the conventional datasets, unveiling the huge volume of interconnected heterogeneous big data has the potential to maximize our knowledge and insights in the target domain.

However, this brings a series of new challenges to the research community. Overcoming the challenges will reshape the future of the data mining technology, resulting in a spectrum of groundbreaking data and mining techniques and algorithms. One feasible approach is to improve existing techniques and algorithms by exploiting massively parallel computing architectures (cloud platforms in our mind).

Big data mining must deal with heterogeneity, extreme scale, velocity, privacy, accuracy, trust, and interactiveness that existing mining techniques and algorithms are incapable of. The need for designing and implementing very-large-scale parallel machine learning and data mining algorithms (ML-DM) has remarkably increased, which accompanies the emergence of powerful parallel and very large scale data processing platforms, e.g., Hadoop MapReduce. NIMBLE is a portable infrastructure that has been specifically designed to enable rapid implementation of parallel ML-DM algorithms, running on top of Hadoop. Apache's Mahout is a library of machine learning and data mining implementations. The library is also implemented on top of Hadoop using the MapReduce programming model. Some important components of the library can run stand-alone. The main drawbacks of Mahout are that its learning cycle is too long and its lack of user-friendly interaction support. Besides, it does not implement all the needed data mining and machine learning algorithms. BC-PDM (Big Cloud-Parallel Data Mining) , as a cloud-based data mining platform, also based on Hadoop, provides access to large telecom data and business solutions for telecom operators; it supports parallel ETL process (extract, transform, and load), data mining, social network analysis, and text mining. BC-PDM tried to overcome the problem of single function of other approaches and to be more applicable for Business Intelligence. PEGASUS (Peta-scale Graph Mining System) and Giraph both implement graph mining algorithms using parallel computing and they both run on top of Hadoop. GraphLab is a graph-based, scalable framework, on which several graph-based machine learning and data mining algorithms are implemented. The reported drawback of GraphLab is that it requires all data fitting into memory.

3.2 Issues and Challenges

The following are key issues and challenges: heterogeneity (or variety), scale (or volume), speed (or velocity), accuracy and trust, privacy crisis, interactiveness, and garbage mining. In the past, data mining techniques have been used to discover unknown patterns and relationships of interest from structured, homogeneous, and small datasets (from today's perspective). Variety, as one of the essential characteristics of big data, is resulted from the

phenomenon that there exists a nearly unlimited different source that generate or contribute to big data. This phenomenon naturally leads to the great variety or heterogeneity of big data. The data from different sources inherently possesses a great many different types and representation forms, and is greatly interconnected, interrelated, and delicately and inconsistently represented. Mining from such a gigantic and heterogeneous dataset, which is typically a tremendous network of interrelated data elements of diverse types, such as an academic social network consisting of authors, papers, conferences, universities, and companies, containing links such as work-at, write, written-by, appear-in, and present, etc. Mining such a dataset, the great challenge is perceivable and the degree of complexity is not even imaginable before we deeply get there. Heterogeneity in big data also means that it is an obligation (rather than an option) to accept and deal with structured, semi-structured, and even entirely unstructured data simultaneously. While structured data can fit well into today's database systems, semi-structured data may partially fit in, but unstructured data definitely will not. Both semi-structured and unstructured data are typically stored in files. This is especially so in data-intensive, scientific computation areas. Nevertheless, though bringing up greater technical challenges, the heterogeneity feature of big data means a new opportunity of unveiling, previously impossible, hidden patterns or knowledge dwelt at the intersections within heterogeneous big data. We shed a little more light on the implied challenge and the opportunity by looking into the examples from a familiar scenario in the following. First, as a classic data mining example, we consider a simple grocery transaction dataset that records only one type of data, i.e., goods items. Examples insights that might be mined from this dataset may include, e.g., the famous association of "beer and diapers" showing a strong linkage between the two items, and popular items like milk that are almost always purchased by customers, showing strong linkage of milk to all other items. In contrast to that, big data mining must deal with semi-structured and heterogeneous data. Now we generalize the aforementioned simple example by extending the scenario to an online market such as eBay. The dataset now is a richer network consisting of at least three different types of objects: items, buyers, and sellers (still this scenario may not be considered complex enough to demonstrate the complexity in big data mining). Interrelation may broadly exist, e.g., between commodity items in the form of "bought with", between sellers and items in the form of "sell" and "sold by", between buyers and items in the form of "buy" or "bought by", and between buyers and sellers in the form of "buy from" and "sold to". This data network has different types of objects and relationships (indicating a light shade of heterogeneity). We speculate that existing data mining techniques would not (if applicable at all) maximally uncover the hidden associations and insights in this data network.

For a heterogeneous set of big data, trying to construct a single model (if doable at all) would most likely not result in good-enough mining results; thus constructing specialized, more complex, multi-model systems is expected . An interesting algorithm following this spirit is proposed in that first determines whether the given dataset is truly heterogeneous, and if so, it then partitions the set into homogeneous subsets and constructs a specialized model for each homogeneous subset. Partitioning, as an intuitive approach, would speed up the process of knowledge discovery from heterogeneous big data. However, potential patterns and knowledge may miss the opportunity of being discovered after partitioning if important relationships (often implicit) crossing distinct homogeneous regions are not adequately retained. Mining from heterogeneous information networks is a promising frontier of current data mining research. Relational databases have been used to capture the heterogeneous information networks and new methods for in-depth network-oriented data mining and analysis have been proposed. However, the degree of the heterogeneity captured does not reflect the real degree of the inherent heterogeneity existing in the big data. Mining hidden patterns from heterogeneous multimedia streams of diverse sources represents another frontier of data mining research. The output of this research has broad applicability such as detection of spreading dangerous diseases and prediction of traffic patterns and other critical social events (e.g., emerging conflicts and wars). Like data mining, the process of big data mining shall also starts with data selection (from multiple sources). Data filtering, cleaning, reduction, and transformation then follow. There emerge new challenges with each of these preprocessing steps. With data filtering, how do we make sure that the discarded data will not severely degrade the quality of the eventually mined results under the complexity of great heterogeneity of big data? The same question could be adapted and asked to all other preprocessing steps and operations of the data mining process.

3.3 Speed/Velocity

For big data, speed/velocity really matters. The capability of fast accessing and mining big data is not just a subjective desire, it is an obligation especially for data streams (a common format of big data) – we must finish a processing/mining task within a certain period of time, otherwise, the processing/mining results becomes less valuable or even worthless. Exemplary applications with real-time requests include earthquake prediction, stock market prediction and agent-based autonomous exchange

(buying/selling) systems. Speed is also relevant to scalability – conquering or partially solving anyone helps the other one. The speed of data mining depends on two major factors: data access time (determined mainly by the underlying data system) and, of course, the efficiency of the mining algorithms themselves. Exploitation of advanced indexing schemes is the key to the speed issue. Multidimensional index structures are especially useful for big data. For example, a combination of R-Tree and KD-tree and the more recently proposed FastBit (developed by the data group at LBNL) shall be considered for big data. Besides, design of new and more efficient indexing schemes is much desired, but remains one of the greatest challenges to the research community.

An additional approach to boost the speed of big data access and mining is through maximally identifying and exploiting the potential parallelism in the access and mining algorithms. The elasticity and parallelism support of cloud computing are the most promising facilities for boosting the performance and scalability of big data mining systems. It is interesting to note that the MapReduce parallel computing model is applicable to only a rather limited class of data-intensive computing problems. Therefore, design of new and more efficient parallel computing models besides MapReduce is greatly desired, but calls for really creative minds.

IV. HADOOP ARCHITECTURE

At a high-level, Hadoop operates on the philosophy of pushing analysis code close to the data it is intended to analyze rather than requiring code to read data across a network. As such, Hadoop provides its own file system, aptly named Hadoop File System or HDFS. When you upload your data to the HDFS, Hadoop will partition your data across the cluster (keeping multiple copies of it in case your hardware fails), and then it can deploy your code to the machine that contains the data upon which it is intended to operate.

Like many NoSQL databases, HDFS organizes data by keys and values rather than relationally. In other words, each piece of data has a unique key and a value associated with that key. Relationships between keys, if they exist, are defined in the application, not by HDFS. And in practice, you're going to have to think about your problem domain a bit differently in order realize the full power of Hadoop (see the next section on MapReduce).
The components that comprise Hadoop are:

HDFS: The Hadoop file system is a distributed file system designed to hold huge amounts of data across multiple nodes in a cluster (where huge can be defined as files that are 100+ terabytes in size!) Hadoop provides both an API and a command-line interface to interacting with HDFS.

Map Reduce Application: Map Reduce is a functional programming paradigm for analyzing a single record in your HDFS. It then assembles the results into a consumable solution. The Mapper is responsible for the data processing step, while the Reducer receives the output from the Mappers and sorts the data that applies to the same key.

Partitioner: The partitioner is responsible for dividing a particular analysis problem into workable chunks of data for use by the various Mappers. The HashPartioner is a partitioner that divides work up by "rows" of data in the HDFS, but you are free to create your own custom partitioner if you need to divide your data up differently.

Combiner: If, for some reason, you want to perform a local reduce that combines data before sending it back to Hadoop, then you'll need to create a combiner. A combiner performs the reduce step, which groups values together with their keys, but on a single node before returning the key/value pairs to Hadoop for proper reduction.

Input Format: Most of the time the default readers will work fine, but if your data is not formatted in a standard way, such as "key, value" or "key [tab] value", then you will need to create a custom InputFormat implementation.

Output Format: Your MapReduce applications will read data in some InputFormat and then write data out through an Output Format. Standard formats, such as "key [tab] value", are supported out of the box, but if you want to do something else, then you need to create your own Output Format implementation.
Additionally, Hadoop applications are deployed to an infrastructure that supports its high level of scalability and resilience. These components include:

Name Node: The NameNode is the master of the HDFS that controls slave DataNode daemons; it understands where all of your data is stored, how the data is broken into blocks, what nodes those blocks are deployed to, and the overall health of the distributed file system. In short, it is the most important node in the entire Hadoop cluster. Each cluster has one NameNode, and the NameNode is a single-point of failure in a Hadoop cluster.

Secondary Name Node: The Secondary NameNode monitors the state of the HDFS cluster and takes "snapshots" of the data contained in the NameNode. If the NameNode fails, then the Secondary

NameNode can be used in place of the NameNode. This does require human intervention, however, so there is no automatic failover from the NameNode to the Secondary NameNode, but having the Secondary NameNode will help ensure that data loss is minimal. Like the NameNode, each cluster has a single Secondary NameNode.

DataNode: Each slave node in your Hadoop cluster will host a DataNode. The DataNode is responsible for performing data management: It reads its data blocks from the HDFS, manages the data on each physical node, and reports back to the NameNode with data management status.

JobTracker: The JobTracker daemon is your liaison between your application and Hadoop itself. There is one JobTracker configured per Hadoop cluster and, when you submit your code to be executed on the Hadoop cluster, it is the JobTracker's responsibility to build an execution plan. This execution plan includes determining the nodes that contain data to operate on, arranging nodes to correspond with data, monitoring running tasks, and relaunching tasks if they fail.

TaskTracker: Similar to how data storage follows the master/slave architecture, code execution also follows the master/slave architecture. Each slave node will have a TaskTracker daemon that is responsible for executing the tasks sent to it by the JobTracker and communicating the status of the job (and a heartbeat) with the JobTracker.

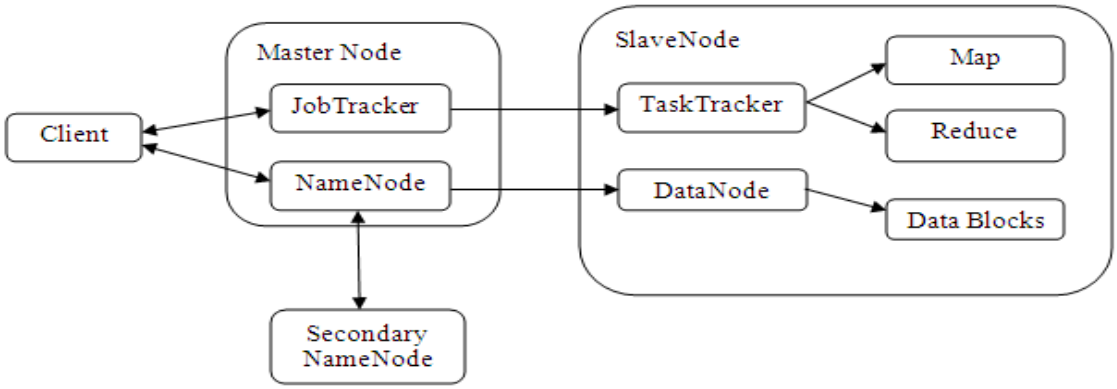

Figure 1: Hadoop Architecture

In Fig 1, the master node contains two important components: the NameNode, which manages the cluster and is in charge of all data, and the JobTracker, which manages the code to be executed and all of the Task Tracker daemons. Each slave node has both a Task Tracker daemon as well as a DataNode: the Task Tracker receives its instructions from the Job Tracker and executes map and reduce processes, while the DataNode receives its data from the NameNode and manages the data contained on the slave node. And of course there is a Secondary NameNode listening to updates from the NameNode.

4.1 MapReduce

MapReduce is a functional programming paradigm that is well suited to handling parallel processing of huge data sets distributed across a large number of computers, or in other words, MapReduce is the application paradigm supported by Hadoop and the infrastructure presented in this article. MapReduce, as its name implies, works in two steps:

Map: The map step essentially solves a small problem: Hadoop's partitioner divides the problem into small workable subsets and assigns those to map processes to solve.

Reduce: The reducer combines the results of the mapping processes and forms the output of the MapReduce operation.

My Map definition purposely used the work "essentially" because one of the things that give the Map step its name is its implementation. While it does solve small workable problems, the way that it does it is that it maps specific keys to specific values. For example, if we were to count the number of times each word appears in a book, our MapReduce application would output each word as a key and the value as the number of times it is seen. Or more specifically, the book would probably be broken up into sentences or paragraphs, and the Map step would return each word mapped either to the number of times it appears in the sentence (or to "1" for each occurrence of every word) and then the reducer would combine the keys by adding their values together.

4.2 Accuracy, Trust, and Provenance

In the past, data mining systems were typically fed with relatively accurate data from well-known and quite limited sources, so the mining results tend to be accurate, too; thus accuracy and trust have

never been a serious issue for concern. With the emerging big data, the data sources are of many different origins, not all well-known, and not all verifiable. Therefore, the accuracy and trust of the source data quickly become an issue, which further propagates to the mining results as well. To (at least partially) solve this problem, data validation and provenance tracing become more than a necessary step in the whole knowledge discovery process (including data mining). History has repeatedly proven that challenges always comes hand-in-hand with opportunities (sometimes unnoticeably). In the case of big data, the copious data sources and gigantic volumes provide rich sources to extract additional evidences for verifying accuracy and building trust on the selected data and the produced mining results.

4.3 Privacy Crisis

Data privacy has been always an issue even from the beginning when data mining was applied to real-world data. The concern has become extremely serious with big data mining that often requires personal information in order to produce relevant/accurate results such as location-based and personalized services, e.g., targeted and individualized advertisements. Also, with the huge volume of big data such as social media that contains tremendous amount of highly interconnected personal information, every piece of information about everybody can be mined out, and when all pieces of the information about a person are dug out and put together, any privacy about that individual instantly disappears. You might ask, how could this be possible? Well, it is already a reality that every transaction regarding our daily life is being pushed to online and leaves a trace there: we comminute with friends via email, instant message, blog, and Facebook; we do shopping and pay our bills online too; and yet, credit card companies hold our confidential identity information; your payroll office has your personal information, too; your home phone number and address are listed in the region's directory that everyone can access; last month, you had a birthday party that disclosed your exact birthday to the circle of your friends, and some of them posted your birthday party in blogs, ... Thanks goodness, everyone so far has the righteous sense of protecting your confidential personal information, but the possibility of unintended leaking cannot be ruled out once and forever, and no leaking today does not guarantee impermeable tomorrow. As time goes, every piece of your personal information will be scattered here or there (hopefully not all available from one location). Well, we have desperately wanted and are diligently working toward powerful mining tools capable of mining a great portion or even the whole Web. So you shall not doubt such powerful mining tools or systems one day will be able to find confidential information of you (and actually of everyone else) – it's now just a matter of time. Everyone would easily gain the privilege of using such powerful tools (via SaaS on the cloud), mine your privacy, and see you entirely "naked". Without the shield of any privacy protecting you, a bad guy could open a new credit card account in your name, and transfer your hard-earned money away from your bank account... Everything seems becoming possible! Imagine how big a social disaster it would be when everyone in the US, for example, can access everyone else's social security number and other identity information, name, address, birthday, birthplace, phone numbers, etc. Even credit card companies do not ask for all this information when one requests to open a new account on the phone. So we definitely run the risk of living transparently or "naked" in an era of no privacy. Should we be proud to say that one day, we will live in a world that everyone can perfectly pretend to be any other one? Well, when anybody can "become" another body as s/he wishes, we get completely separated from our true identities. Now we need most seriously ask ourselves: would we rather to wear the "the emperor's new clothes"? The answer is certainly "no" as we all believe. Then what are the possible countermeasures? Apparently, we urgently need proper policies and approaches to manage sharing of personal data, while legitimate data mining activities shall still be granted facilitated.

V. CONCLUSION

Big data discloses the limitations of existing data mining techniques, resulted in a series of new challenges related to big data mining. Big data mining is a promising research area, still in its infancy. In spite of the limited work done on big data mining so far, we believe that much work is required to overcome its challenges related to heterogeneity, scalability, speed, accuracy, trust, provenance, privacy, and interactiveness. This paper also provides an overview (though limited due to space limit) of state-of-the-art frameworks/platforms for processing and managing big data as well as platforms and libraries for mining big data. More specifically, we originally pointed out and analyzed the risk of privacy crisis which is deteriorated by big data and big data mining and first time proposed and formulated garbage mining – a critical issue in the big data era that has not been realized by others nor addressed anywhere else. As our future work, we are at the stage of seriously planning a research project on cyberspace garbage mining to make the cyberspace a more sustainable environment.

REFERENCES

[1] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu,and G. Fox, "Twister: A runtime for iterative MapReduce," inProc. 19th ACM Int. Symp. High Perform. Distrib. Comput., 2010,pp. 810–818.

[2] Amazon Elastic MapReduce. (2013). [Online]. Available: http://aws.amazon.com/ elasticmapreduce/

[3] R. Ahmed and G. Karypis, "Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks," Knowledge and Information Systems, vol. 33, no. 3, pp. 603-630, Dec. 2012.

[4] M.H. Alam, J.W. Ha, and S.K. Lee, "Novel Approaches to Crawling Important Pages Early," Knowledge and InformationSystems, vol. 33, no. 3, pp 707-734, Dec. 2012.

[5] S. Aral and D. Walker, "Identifying Influential and Susceptible Members of Social Networks," Science, vol. 337, pp. 337-341, 2012.

[6] A. Machanavajjhala and J.P. Reiter, "Big Privacy: Protecting Confidentiality in Big Data," ACM Crossroads, vol. 19, no. 1,pp. 20-23, 2012.

[7] S. Banerjee and N. Agarwal, "Analyzing Collective Behavior from Blogs Using Swarm Intelligence," Knowledge and InformationSystems, vol. 33, no. 3, pp. 523-547, Dec. 2012.

[8] E. Birney, "The Making of ENCODE: Lessons for Big-DataProjects," Nature, vol. 489, pp. 49-51, 2012.