Abstract

In today’s era, there is a great deal added to real-time remote sensing Big Data than it seems at first, and extracting the useful information in an efficient manner leads a system toward a major computational challenges, such as to analyze, aggregate, and store, where data are remotely collected. Keeping in view the above mentioned factors, there is a need for designing a system architecture that welcomes both real-time, as well as offline data processing. Big data is data whose characteristics force us to look beyond the traditional methods that are prevalent at the time. Online news, microblogs, search queries are just a few examples of these continuous streams of user activities. Evolving data streams methods are becoming a low-cost, green methodology for real time online prediction and analysis. Heterogeneity, scale, timeliness, complexity, and privacy problems with Big Data impede progress at all phases of the pipeline that can create value from data. The problems start right away during data acquisition, when the data tsunami requires us to make decisions, currently in an ad hoc manner, about what data to keep and what to discard, and how to store what we keep reliably with the right metadata.
I. INTRODUCTION

Today, over the past five years, the authors and many others at Google have implemented hundreds of special-purpose computations that process large amounts of raw data, such as crawled documents, web request logs, etc., to compute various kinds of derived data, such as inverted indices, various representations of the graph structure of web documents, summaries of the number of pages crawled per host, the set of most frequent queries in a given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues. As a reaction to this complexity, we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. We realized that most of our computations involved applying a map operation to each logical record in our input in order to compute a set of intermediate key/value pairs, and then applying a reduce operation to all the values that shared the same key, in order to combine the derived data appropriately. Our use of a functional model with user specified map and reduce operations allows us to parallelize large computations easily and to use re-execution as the primary mechanism for fault tolerance.

Most data generated is originally streaming data. This fact is especially true for data representing measurements, actions and interactions, such as the one coming from sensor networks or the Web. In-rest data is just a snapshot of streaming data obtained from an interval of time. In the streaming model, data arrives at high speed, and algorithms must process it in one pass under very strict constraints of space and time. Streaming algorithms
use probabilistic data structures in algorithm give fast approximated answers. However, sequential online algorithms are limited by the memory and bandwidth of a single machine. Achieving results faster and scaling to larger data streams requires resorting to parallel and distributed computing. Map Reduce is currently the de-facto standard programming paradigm in this area, mostly thanks to the popularity of Hadoop1, an open source implementation of Map Reduce started at Yahoo.

Data stream real time analytics are needed to manage the data currently generated, at an ever increasing rate, from such applications as: sensor networks, measurements in network monitoring and traffic management, log records or click-streams in web exploring, manufacturing processes, call detail records, email, blogging, twitter posts and others. In fact, all data generated can be considered as streaming data or as a snapshot of streaming data, since it is obtained from an interval of time. In the data stream model, data arrive at high speed, and algorithms that process them must do so under very strict constraints of space and time. Consequently, data streams pose several challenges for data mining algorithm design. First, algorithms must make use of limited resources (time and memory). Second, they must deal with data whose nature or distribution changes over time. We need to deal with resources in an efficient and low-cost way. Green computing is the study and practice of using computing resources efficiently. A main approach to green computing is based on algorithmic efficiency. In data stream mining, we are interested in three main dimensions: Accuracy, Amount of space (computer memory) necessary. The time required to learn from training examples and to predict.

Figure 2: Peer-Research Big Data Analytics Survey

II. SURVEY REVIEW

1. Adamu Galadima describes a brief look at the Arduino microcontroller and some of its applications and how it can be used in learning. Arduino is an open-source microcontroller used in electronic prototyping. Arduino hardware and its components shall be looked at. Software and the Environment that Arduino runs on are both looked at too. Some applications will be taken as examples that can help make learning Arduino more interesting. This can be used as a major way to encourage students and others to learn more about electronics and programming.
2. Jeffrey Cohen present data parallel algorithms for sophisticated statistical techniques, with a focus on density methods. Finally, he reacts on database system features that enable agile design and flexible algorithm development using both SQL and Map Reduce interfaces over a variety of storage mechanisms.

3. Brian Dolan present the design philosophy, techniques and experience providing MAD analytics for one of the world's largest advertising networks at Fox Audience Network, using the Green plum parallel database system. We describe database design methodologies that support the agile working style of analysts in these settings.

4. R. P. Singh explain why a cloud-based solution is required, describe our prototype implementation, and report on some example applications we have implemented that demonstrate personal data ownership, control, and analytics. He address these issues by designing and implementing a cloud-based architecture that provides consumers with fast access and fine-grained control over their usage data, as well as the ability to analyse this data with algorithms of their choosing, including third party applications that analyse data in a privacy preserving fashion.

5. Jeffrey Dean describes the basic programming model and gives several examples. Many real world tasks are expressible in these models. Implementation of Map Reduce runs on a large cluster of commodity machines and is highly scalable: a typical Map Reduce computation processes many terabytes of data on thousands of machines. Programmers and the system easy to use: hundreds of Map Reduce programs have been implemented and upwards of one thousand Map Reduce jobs are executed on Google's clusters every day.

6. Panagiotis D. Diamantoulakis implements the Big Data Analytics for Dynamic Energy Management in Smart Grids. The smart electricity grid enables a two-way flow of power and data between suppliers and consumers in order to facilitate the power flow optimization in terms of economic efficiency, reliability and sustainability. This infrastructure permits the consumers and the micro-energy producers to take a more active role in the electricity market and the dynamic energy management (DEM). The most important challenge in a smart grid (SG) is how to take advantage of the users’ participation in order to reduce the cost of power.

7. L. Aniello explores the idea of a framework leveraging multiple data sources to improve protection capabilities of CIs. Challenges and opportunities are discussed along three main research directions: i) use of distinct and heterogeneous data sources, ii) monitoring with adaptive granularity, and iii) attack modelling and runtime combination of multiple data analysis techniques.

III. BIG DATA TECHNOLOGY

Big-Data Technology: Sense, Collect, Store and Analyze. The rising importance of big-data computing stems from advances in many different technologies:

1. Sensors: Digital data are being generated by many different sources, including digital imagers (telescopes, video cameras, MRI machines), chemical and biological sensors
(microarrays, environmental monitors), and even the millions of individuals and organizations generating web pages.

2. **Computer networks:** Data from the many different sources can be collected into massive data sets via localized sensor networks, as well as the Internet.

3. **Data storage:** Advances in magnetic disk technology have dramatically decreased the cost of storing data. For example, a one-terabyte disk drive, holding one trillion bytes of data, costs around $100. As a reference, it is estimated that if all of the text in all of the books in the Library of Congress could be converted to digital form, it would add up to only around 20 terabytes.

4. **Cluster computer systems:** A new form of computer systems, consisting of thousands of "nodes," each having several processors and disks, connected by high-speed local-area networks, has become the chosen hardware configuration for data-intensive computing systems. These clusters provide both the storage capacity for large data sets, and the computing power to organize the data, to analyze it, and to respond to queries about the data from remote users. Compared with traditional high-performance computing, where the focus is on maximizing the raw computing power of a system, cluster computers are designed to maximize the reliability and efficiency with which they can manage and analyze very large data sets. The "trick" is in the software algorithms – cluster computer systems are composed of huge numbers of cheap commodity hardware parts, with scalability, reliability, and programmability achieved by new software paradigms.

5. **Cloud computing facilities:** The rise of large data centers and cluster computers has created a new business model, where businesses and individuals can rent storage and computing capacity, rather than making the large capital investments needed to construct and provision large-scale computer installations. For example, Amazon Web Services (AWS) provides both network-accessible storage priced by the gigabyte-month and computing cycles priced by the CPU-hour. Just as few organizations operate their own power plants, we can foresee an era where data storage and computing become utilities that are ubiquitously available.

IV. DATA ANALYSIS

This component analyzes the data and provides as outputs information about (i) how to adapt the grain of the monitoring, (ii) what protection actions should be performed on the CI. Starting from our past experiences on attack modelling and data analysis, we consider the following functional blocks.

**Data Processing.** Collected raw data typically contain useless or redundant information that can undermine the goodness of performed analysis. The first analysis step to be performed is thus to polish raw data, adopting filtering or event coalescence techniques, such as the ones analyzed in [5].

**Attack Modelling.** This functional block provides tools to define and statically analyzed attack models. The attack model used in this block must be capable of: (i) providing a high degree of flexibility in representing many different security scenarios in a compact way; (ii)
allowing the specification of various kinds of constraints (e.g., temporal) on possible attacks; (iii) representing attack scenarios at different abstraction levels, allowing to “focus” the conformance checking task in various ways. Typed temporal graph-based attack models [3] appear to be good options for the above requirements. They are rich in terms of temporal constraints that can be expressed. In addition, it is relatively easy to handle the definition of generalization/specialization hierarchies among event types.

V. DATA PROCESSING UNIT

In data processing unit (DPU), the filtration and load balancer server have two basic responsibilities, such as filtration of data and load balancing of processing power. Filtration identifies the useful data for analysis since it only allows useful information, whereas the rest of the data are blocked and are discarded. Hence, it results in enhancing the performance of the whole proposed system. Apparently, the load-balancing part of the server provides the facility of dividing the whole filtered data into parts and assign them to various processing servers. The filtration and load-balancing algorithm varies from analysis to analysis; e.g., if there is only a need for analysis of sea wave and temperature data, the measurement of these described data is filtered out, and is segmented into parts.

VI. MULTIPLE DATA SOURCES

The idea of using distinct and heterogeneous data sources available in today’s CIs can help to draw a clearer picture of the system to protect and of the threatening activities being carried out. The aim is to improve the protection of future CIs exploiting the (hidden) value of data: they are already available but not fully exploited in today CIs.

However, as the size and complexity of systems increase, the amount of information that can be collected by data sources skyrockets. For example, in the 1300-nodes data centre we target as case study (see Section III-D) the monitoring system produces about 16.6 GB of data per day, with observed traffic peaks of about 240000 pkt/s. This is a consequence of multiple factors: (i) the increasing availability of cheap HW probes, (ii) the ubiquitousness of communication infrastructures (either wired or wireless) and the Internet, and (iii) the novel algorithmic approaches that today’s making handling huge amounts of data practical.

A further important aspect is that the heterogeneity of collected data is going to increase as well: new data sources are connected to monitoring systems to collect and analyze different kinds of data as this could potentially provide useful insights on current system statuses. This mix of factors marks the shift from a mostly human controlled distributed monitoring model (think, for example, about how railway companies in the past controlled the status of their infrastructures through hundreds of people deployed on the territory along their tracks to locally monitor and then report to their bosses) to fully automated IT infrastructure for monitoring that tries to relieve as much as possible from humans the burden of analyzing data to infer high-level information. Making this new model practical in scenarios where huge amounts of heterogeneous data are available calls for the research of new algorithmic and architectural solutions able to withstand these new challenges.
VII. CONCLUSION

In this review paper, finally all the survey is related to real-time Big Data analysis for remote sensing application. The algorithms review in this paper for each unit and subunits are used to analyze remote sensing data sets, which helps in better understanding of land and sea area. This architecture welcomes researchers and organizations for any type of remote sensory Big Data analysis by developing algorithms for each level of the architecture depending on their analysis requirement. The better analysis of the large volumes of data that are becoming available, there is the potential for making faster advances in many scientific disciplines and improving the profitability and success of many enterprises. We have only begun to see its potential to collect, organize, and process data in all walks of life. A modest investment by the federal government could greatly accelerate its development and deployment.

VIII. REFERENCES


To Cite This Article