A Comparative Study based on Big Data Processing Techniques

Ajaz Ahmad Kumar
Department of Electronics and Information Technology
National Institute of Electronics and Information Technology (NIELIT),
Ministry of Electronics & Information Technology, GOI

Abstract— The concept of bigdata is based on grid computing and its essential features i.e. six (6) V's. The aim of BPT leveraging big data is to take action – to make more accurate decisions and to produce smaller data sets for analysis so quickly. The study of bigdata processing has been carried on the areas like generation and interpretation of bigdata, large volume of data beyond capacity and unstructured data. Unsupervised machine learning algorithms are being applied profusely in BPT. These unsupervised learning is used for finding the hidden structure from the unlabelled datasets. Since the datasets are not labelled, there will be no error while evaluating for potential solutions. The objective of this paper is to present a comparative study of different bigdata processing techniques.

Key words: bigdata; data sets; data processing; data storage

I. INTRODUCTION

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, data creation, search, sharing, storage, transfer, visualization, querying, updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behaviour analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set." [1] There is little doubt that the quantities of data now available are indeed large, but that's not the most relevant characteristic of this new data ecosystem."[2]

The recently study surveyed 1,061 companies from across the globe. The survey found that twenty-eight percent of the firms interviewed were piloting or implementing big data activities. IBM outlined four phases of big data adoption, which include educate, explore, engage and execute (4 E's).

- Educate. This phase focuses on knowledge gathering and market observations.
- Explore. After completing the education phase, companies will develop a strategy and roadmap based on business needs and challenges.
- Engage. During the third phase, a business will pilot big data initiatives to validate value and requirements.
- Execute. Companies in the fourth phase have deployed two or more big data initiatives and are continuing to apply advanced analytics.

Bigdata processing techniques analyze big data sets at terabyte or even petabyte scale. Offline batch data processing is typically full power and full scale, tackling arbitrary BI use cases. While real-time stream processing is performed on the most current slice of data for data profiling to pick outliers, fraud transaction detections, security monitoring, etc. The toughest task however is to do fast (low latency) or real-time ad-hoc analytics on a complete big data set. It practically means you need to scan terabytes (or even more) of data within seconds. This is only possible when data is processed with high parallelism.

II. SOME IMPORTANT DEFINITIONS

It is essential to define some important characteristics [3]-[4] related to bigdata processing:

- Volume: The quantity of generated and stored data. The size of the data determines the value and potential insight- and whether it can actually be considered big data or not.
- Velocity: The speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development.
- Variety: The type and nature of the data. This helps people who analyze it to effectively use the resulting insight.
- Veracity: The quality of captured data can vary greatly, affecting accurate analysis.
- Variability: Inconsistency of the data set can hamper processes to handle and manage it.
- Value: To create value from big data, IBM stated that a company should:
  a) Commit initial efforts to customer-centric outcomes
  b) Develop an enterprise-wide big data blueprint
  c) Start with existing data to achieve near term results
  d) Build analytical capabilities based on business priorities
  e) Create a business case based on measurable outcomes
- Machine Learning: big data often doesn't ask why and simply detects patterns [5]
- Digital footprint: big data is often a cost-free by-product of digital interaction [6]
- Hadoop: is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.
- Hadoop Comm: The common utilities and libraries that support the other Hadoop modules.
- Hadoop Distributed File System: a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
- Hadoop YARN: a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications [7]-[8]
- Hadoop MapReduce: an implementation of the MapReduce programming model for large scale data processing [12].

Apache Hadoop is a distributed computing framework modeled after Google Map Reduce to process large amounts of data in parallel. Once in a while, the first thing that comes when speaking about distributed computing is EJB. EJB is de facto a component model with remoting capability but short of the critical features being a distributed computing framework that include computational parallelization, work distribution, and tolerance to unreliable hardware and software. Hadoop on the other hand has these merits built-in. Zoo Keeper modelled on Google Chubby is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and group services for the Hadoop cluster[9]-[10]. Hadoop Distributed File System (HFDS) modeled on Google GFS is the underlying file system of a Hadoop cluster[9]-[10]. HDFS works more efficiently with a few large data files than numerous small files. A real-world Hadoop job typically takes minutes to hours to complete, therefore Hadoop is not for real-time analytics, but rather for offline, batch data processing. Recently, Hadoop has undergone a complete overhaul for improved maintainability and manageability. Something called YARN (Yet Another Resource Negotiator) is at the center of this change. One major objective of Hadoop YARN is to decouple Hadoop from MapReduce paradigm to accommodate other parallel computing models, such as MPI (Message Passing Interface) [11] and Spark.

III. STUDY OF SOME RECENT ALOGORITHIMS ADAPTED FOR BIGDATA PROCESSING TECHNIQUES

A. Move-Code-To-Data Philosophy Technique

Data flows from components to components in an enterprise application. This is the case for application frameworks (EJB and spring framework), integration engines (Camel and Spring Integration), as well as ESB (Enterprise Service Bus) products. Nevertheless, for the data-intensive processes Hadoop deals with, it makes better sense to load a big data set once and perform various analysis jobs locally to minimize IO and network cost, the so-called "Move-Code-To-Data" philosophy. When you load a big data file to HDFS, the file is split into chunks (or file blocks) through a centralized Name Node (master node) and resides on individual Data Nodes (slave nodes) in the Hadoop cluster for parallel processing[9]-[10].

B. Map Reduce Processing Technique

A centralized Job Tracker process in the Hadoop cluster moves your code to data. The code hereby includes a Map and a Reduce class. Put simply, a Map class does the heavy-lifting job of data filtering, transformation, and splitting. For better IO and network efficiency, a Mapper instance only processes the data chunks co-located on the same data node, a concept termed data locality (or data proximity). Mappers can run in parallel on all the available data nodes in the cluster. The outputs of the Mappers from different nodes are shuffled through a particular algorithm to the appropriate Reduce nodes. A Reduce class by nature is an aggregator. The number of Reducer instances is configurable to developers.[12]-[13]

1) Word Count Example

A rudimentary "word count" example exhibits how Hadoop runs. The objective is to count the number of times each word is presented in a set of text documents. As we know Hadoop and HDFS work better with a few large files; the first step is to merge all the small text files through the HDFS API to become a single big file, which is further broken down evenly into chunks (file blocks) by the Name Node. These file blocks are distributed across the data nodes in the cluster. Each Map instance takes a line from its local file blocks as input and splits it into words. It then emits a key-value pair of the word and count 1. The outputs with the same key (word) are shuffled to the same Reduce node. A Reduce node sums the counts for every word received and emits a single key-value pair with the word and the total count. While Map phase runs with high parallelism, Reduce phase reconciles the outputs from the Mappers to yield the final results.

C. Hadoop Ecosystem Technique

Hadoop API is often considered low level, as it is not easy to program with. The quickly growing Hadoop ecosystem offers a list of abstraction techniques, which encapsulate and hide the programming complexity of Hadoop. Pig, Hive, Cascading, Crunch, Scrunch, Scalding, Scoobi, and Cascalog all aim to provide low cost entry to Hadoop programming[21]-[22].

D. Pig, Crunch (Scrunch), and Cascading are data-pipe based techniques

A data pipe is a multi-stepped process, in which transformation, splitting, merging, and join may be conducted individually at each step. Thinking about a work flow in a general work flow engine, a data pipe is similar. Hive on the other hand works like a data warehouse by offering a SQL compatible interactive shell. Programs or shell scripts developed on top of these techniques are compiled to native Hadoop Map and Reduce classes behind the scene to run in the cluster. Given the simplified programming interfaces in conjunction with libraries of reusable functions, development productivity is greatly improved [21]-[22].

E. Stream Processing Technique

Twitter Storm is an open source, big-data processing system intended for distributed, real-time streaming processing. Storm implements a data flow model in which data (time series facts) flows continuously through a topology (a network of transformation entities). The slice of data being analyzed at any moment in an aggregate function is specified by a sliding window, a concept in CEP/ESP. A sliding window may be like "last hour", or "last 24 hours", which is constantly shifting over time. Data can be fed to Storm through distributed messaging queues like Kafka, Kestrel, and even regular JMS. Trident is an abstraction API of Storm that makes it easier to use. Like Twitter Storm, Apache S4 is a product for distributed, scalable, continuous, stream data processing.[17]-[18]

F. Fast/Real-Time Big Data Processing Technique

Big data OLAP (OnLine Analytical Processing) is extremely data and CPU intensive in that terabytes (or even more) of data are scanned to compute arbitrary data aggregates within seconds. Note indexing is indeed not helpful in a full "table" scan; in addition, building an index on a big data set is costly and slow [16]-[18].

G. Google Dremel (BigQuery), Cloudera Impala, Apache Drill

Cloudera Impala and Apache Drill are modeled after Google Dremel.[19] These techniques run fast in that coordination, query planning, optimization, scheduling, and execution are all distributed throughout nodes in a cluster to maximize parallelization. All three techniques favor a query-efficient columnar storage format. Before saving data directly in a columnar format in a data store (NOSQL, NewSQL, Relational, and more), you may need to transform the existing row-based data through a MapReduce job. Full "table" scan-based ad-hoc queries are offered but with certain limitations. Therefore, they are not a replacement of Hadoop. Supporting in situ (in position) data sources like GFS, BigTable, HDFS, and HBase makes data access blazing faster because of data locality (proximity). With the data source being an OLTP database (BigTable, HBase), a write made by an end user is reflected instantaneously in an analysis report. Such architecture brings you a Big Data OLAP system with typical latency in seconds range. To save resources, it is recommended to build a materialized view (cached result) on an analysis job, and return that view if no changes are expected on the result. Impala and Drill have nice integration with commercial BI/Analysis tools like Tableau, MicroStrategy, Excel, SAP etc.[19]-[20]

IV. CONCLUSION

Researchers have been studying various aspects of bigdata processing techniques over the years. In recent times it is observed that batch-based, stream-based, graph-based, DAG-based, interactive-based or visual-based techniques are specifically applied for diverse bigdata processing problems. The paper tries to capture the essence of few BPT based research works in current years. A brief description of individual's work is provided underlining their respective contributions. The problem of bigdata processing especially generation of demanded and challenging and valuable data sets for decision making is extremely difficult and requires more attention.

REFERENCES

[1] New Horizons for a Data-Driven Economy – Springer. Doi:10.1007/978-3-319-21569-3.
[2] boyd, dana; Crawford, Kate (September 21, 2011). "Six Provocations for Big Data". Social Science Research Network: A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society. doi:10.2139/ssrn.1926431
[3] Hilbert, Martin. "Big Data for Development: A Review of Promises and Challenges. Development Policy Review.". martinhilbert.net. Retrieved 2015-10-07.
[4] DT&SC 7-3: What is Big Data?. 12 August 2015 – via YouTube.
[5] Mayer-Schön berger, V., & Cukier, K. (2013). Big data: a revolution that will transform how we live, work and think. London: John Murray.
[6] "Digital Technology & Social Change".
[7] "Resource (Apache Hadoop Main 2.5.1 API)". apache.org. Apache Software Foundation. 2014-09-12. Retrieved 2014-09-30.
[8] Murthy, Arun (2012-08-15). "Apache Hadoop YARN – Concepts and Applications". hortonworks.com. Hortonworks. Retrieved 2014-09-30.
[9] Google spotlights data center inner workings | Tech news blog - CNET News.com
[10] MapReduce: Simplified Data Processing on Large Clusters
[11] http://www.mcs.anl.gov/research/projects/mpi/mpi-standard/mpi-report-2.0/mpi2-report.htm MPI 2 standard
[12] Tutorial on MPI Reduce and all reduce
[13] Google Map Reduce: Simplified data processing on large clusters.
[14] Google Big Table: A distributed storage system for managing structured data.
[15] Apache Hadoop: A reliable, scalable, distributed computing framework.
[16] Apache Cassandra: A distributed, scalable, big data store.
[17] Twitter Storm: A free and open source distributed real-time computation system.
[18] Apache S4: A general-purpose, distributed, and scalable data stream processing system.
[19] Google Dremel: Interactive analysis of web-scale datasets.
[20] Cloudera Impala: Real-time queries in Apache Hadoop.
[21] Hadoop in Action: A book on Hadoop.
[22] Hadoop in Practice: A book on Hadoop ecosystem.

