On the Role of Distributed Computing in Big Data Analytics

Alba Amato

1 Introduction

Distributed paradigm emerged as an alternative to expensive supercomputers, in order to handle new and increasing users needs and application demands [1]. Opposed to supercomputers, distributed computing systems are networks of large number of attached nodes or entities connected through a fast local network [2]. This architectural design allows to obtain high computational capabilities by joining together a large number of compute units via a fast network and resource sharing among different users in a transparent way. Having multiple computers processing the same data means that a malfunction in one of the computers does not influence the entire computing process. This paradigm is also strongly motivated by the explosion of the amount of available data that make necessary the effective distributed computation. Gartner has defined big data as "high volume, velocity and/or variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation" [3]. In fact the huge size is not the only property of Big Data. Only if the information has the characteristics of either of Volume, Velocity and/or Variety we can refer the area of problem/solution domain as Big Data [4].Volume refers to the fact that we are dealing with ever-growing data expanding beyond terabytes into petabytes, and even exabytes (1 million terabytes). Variety refers to the fact that Big Data is characterized by data that often come from heterogeneous sources such as machines, sensors and unrefined ones, making the management much more complex. Finally, the third characteristic, that is velocity that, according to Gartner [5], "means both how fast data is being produced and how fast the data must be

A. Amato (✉)
Department of Industrial and Information Engineering, Second University of Naples, Caserta, CE, Italy
e-mail: alba.amato@unina2.it; albaamato@gmail.com

S. Mazumder et al. (eds.), *Distributed Computing in Big Data Analytics*, Scalable Computing and Communications, DOI 10.1007/978-3-319-59834-5_1

processed to meet demand". In fact in a very short time the data can become obsolete. Dealing effectively with Big Data "requires to perform analytics against the volume and variety of data while it is still in motion, not just after" [4]. IBM [6] proposes the inclusion of veracity as the fourth big data attribute to emphasize the importance of addressing and managing the uncertainty of some types of data. Striving for high data quality is an important big data requirement and challenge, but even the best data cleansing methods cannot remove the inherent unpredictability of some data, like the weather, the economy, or a customer's actual future buying decisions. The need to acknowledge and plan for uncertainty is a dimension of big data that has been introduced as executives seek to better understand the uncertain world around them [7]. Big Data are so complex and large that it is really difficult and sometime impossible, to process and analyze them using traditional approaches. In fact traditional relational database management systems (RDBMS) can not handle big data sets in a cost effective and timely manner. These technologies are typically not enabled to extract, from large data set, rich information that can be exploited across of a broad range of topics such as market segmentation, user behavior profiling, trend prediction, events detection, etc. in various fields like public health, economic development and economic forecasting. Besides Big Data have a low information per byte, and, therefore, given the vast amount of data, the potential for great insight is quite high only if it is possible to analyze the whole dataset [4]. The challenge is to find a way to transform raw data into valuable information.

So, to capture value from big data, it is necessary to use next generation innovative data management technologies and techniques that will help individuals and organizations to integrate, analyze, visualize different types of data at different spatial and temporal scales. Basically the idea is to use distributed storage and distributed processing of very large data sets in order to address the four V's. There come the big data technologies which are mainly built on distributed paradigm. Big Data Technologies built using the principals of Distributed Computing, allow acquizition and analysis of intelligence from big data. Big Data Analytics can be viewed as a sub-process in the overall process of insight extraction from big data [8].

In this chapter, the first section introduces an overview of Big Data, describing their characteristics and their life cycle. In the second section the importance of Distributed Computing is explained focusing on issue and challenges of Distributed Computing in Big Data analytics. The third section presents an overview of technologies for Big Data analytics based on Distributed Computing concepts. The focus will be on Hadoop.[1] which provides a distributed file system, YARN[2], a resource manager through which multiple applications can perform computations simultaneously on the data, and Spark,[3] an open-source framework for the analysis of data that can be run on Hadoop, its architecture and its mode of operation in comparison to MapReduce.[4] The choice of Hadoop is due to more elements. First

[1] hadoop.apache.org.

[2] https://hadoop.apache.org/docs/current/hadoop-yarn.html.

[3] spark.apache.org/.

[4] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.

of all it is leading to phenomenal technical advancements. Moreover it is an open source project, widely adopted with an ever increasing documentation and community. In the end conclusion are discussed together with the current solutions and future trends and challenge.

2 History and Key Characteristics of Big Data

Distributed computing divides the big unmanageable problems around processing, storage and communication, into small manageable pieces and solves it efficiently in a coordinated manner [9]. Distributed computing are ever more widespread because of availability of powerful yet cheap microprocessors and continuing advances in communication technology. It is necessary especially when there are complex processes that are intrinsically distributed, with the need for growth and reliability.

Data management industry has been revolutionized by hardware and software breakthroughs. First, hardware's power increased and hardware's price decrease. As a consequence, new software emerged that takes advantage of this hardware by automating processes like load balancing and optimization across a huge cluster of nodes.

One of the problems with managing large quantities of data, has been the impact of latency that represents an issue in every aspect of computing, including communications, data management, system performance, and more. The capability to leverage distributed computing and parallel processing techniques reduced latency. It may not be possible to construct a big data application in a high latency environment if high performance is needed. It is necessary to process, analyse and verify this data in near real time. With the aim of reducing latency various distributed computing and parallel processing techniques have been proposed by researchers and practitioners from time to time.

Frequently problems are also related to high likelihood of hardware failure, improportionate distribution of data across various nodes in cluster and security issues due to the data access from anywhere.

The solution of those problems are typically based on distributed file storage (such as HDFS,[5] OpenAFS,[6] XtreemFS,[7]...), cluster resource management (such as YARN, Mesos,[8]...), and parallel programming model for large data sets and analysis model (such as MapReduce, Spark, Flink[9]).

The term Big Data is a broad and evolving term that refers to any collection of data so wide as to make it difficult or impossible to store it in a traditional software

[5] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

[6] https://www.openafs.org/.

[7] www.xtreemfs.org/.

[8] mesos.apache.org/.

[9] https://flink.apache.org/.

system, as RDBMS (Relational Database Management System). Although the term does not refer to any particular amount, usually it is possible to talk about Big Data from couple of Gigabytes of data, that is, when the data can not be easily processed by a single process. Big Data solutions are ideal for analysing not only raw structured data, but semistructured and unstructured data from a wide variety of sources [4]; Big Data solutions are ideal when all, or most, of the data needs to be analysed versus a sample of the data; or a sampling of data is not nearly as effective as a larger set of data from which to derive analysis; Big Data solutions are ideal for iterative and exploratory analysis when measures on data are not predetermined.

The collection of data streams of higher velocity and higher variety brings several problems that can be addressed by big data technologies. Thanks to big data technology it is possible to build an infrastructure that delivers low, predictable latency in both capturing data and in executing simple and complex queries; it is also possible to handle very high transaction volumes, often in a distributed environment; and supports flexible, dynamic data structures [10]. When dealing with such a high volume of information, it is relevant to organize data at its original storage location, thus saving both time and money by not moving around large volumes of data. The analysis may also be done in a distributed environment, where some data will stay where it was originally stored and be transparently accessed for required analytics such as statistical analysis and data mining, on a wider variety of data types stored in diverse systems; to scale for extreme data volumes and deliver faster response times. Most importantly, the infrastructure must be able to integrate analysis on the combination of big data and traditional enterprise data. New insight comes not just from analyzing new data, but from analyzing it within the context of the old to provide new perspectives on old problems [10]. Context-aware Big Data solutions could focus only on relevant information by keeping high probability of hit for all application-relevant events, with manifest advantages in terms of cost reduction and complexity decrease [11]. Obviously the results of big data analysis are only as good as the data being analyzed.

In last two decades, the term database is used in several contexts and is usually used as synonymous with SQL. Recently, however, the world of data storage has changed and new and interesting possibilities are now based on NoSQL. NoSQL stands for "Not Only SQL" and this emphasizes that the NoSQL technology is not entirely incompatible with SQL (Structured Query Language), it describes a large class of databases which are generally not queried with SQL. NoSQL data stores are designed to scale well horizontally and run on commodity hardware. NoSQL is definitely not suitable for all uses and is not a replacement of the traditional RDBMS database, but it can assist them or in part replace, and its main advantages make it useful, if not essential, in some occasions. NoSQL can significantly reduce development time because it eliminates the need to address complex SQL queries to extract structured data. The NoSQL database, if used properly, return the data in a timely way than a traditional database. This factor is really important with web and mobile applications. NoSQL data stores have several key features [12] that help them to horizontally scale throughput over many servers, replicate and distribute data over

many servers, and dynamically add new attributes to data records [12]. NoSQL Data Models can be classified in:

- Key-value data stores (KVS). They store values associated with an index (key). KVS systems typically provide replication, versioning, locking, transactions, sorting, and/or other features. The client API offers simple operations including puts, gets, deletes, and key lookups.
- Document data stores (DDS). DDS typically store more complex data than KVS, allowing for nested values and dynamic attribute definitions at runtime. Unlike KVS, DDS generally support secondary indexes and multiple types of documents (objects) per database, as well as nested documents or lists.
- Extensible record data stores (ERDS). ERDS store extensible records, where default attributes (and their families) can be defined in a schema, but new attributes can be added per record. ERDS can partition extensible records both horizontally (per-row) or vertically (per-column) across a datastore, as well as simultaneously using both partitioning approaches.

Another important category is constituted by Graph data stores. They [13] are based on graph theory and use graph structures with nodes, edges, and properties to represent and store data. Key-Value, Document based and Extensible record categories aim at the entities decoupling to facilitate the data partitioning and have less overhead on read and write operations, whereas Graph-based category take the modeling the relations like principal objective. Therefore techniques to enhancing schema with a Graph-based database may not be the same as used with Key-Value and others. The graph data model fits better to model domain problems that can be represented by graph as ontologies, relationship, maps etc. Particular query languages allow querying the data bases by using classical graph operators as neighbour, path, distance etc.

Because for many Big Data use cases, the data does not have to be 100 percent consistent all the time, applications can scale out to a much greater extent. Eric Brewer's CAP theorem [14], formalized in [15], which basically states that is impossible for a distributed computing system to simultaneously provide all three of the following guarantees: Consistency, Availability and Partition Tolerance (from these properties the CAP acronym has been derived). Where:

- Consistency: all nodes see the same data at the same time
- Availability: a guarantee that every request receives a response about whether it was successful or failed
- Partition Tolerance: the system continues to operate despite arbitrary message loss or failure of part of the system that create a network partition

Only two of the CAP properties can be ensured at the same time. Therefore, only CA systems (consistent and highly available, but not partition-tolerant), CP systems (consistent and partition tolerant, but not highly available), and AP systems (highly available and partition-tolerant, but not consistent) are possible and for many people CA and CP are equivalent because loosing in Partitioning Tolerance means a lost of Availability when a partition takes place.

There are several other compute infrastructures to use in various domains. MapReduce is a programming model and an associated implementation for processing and generating large datasets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as show in [16]. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. This allows programmers without any experience with parallel and distributed systems to utilize the resources of a large distributed system easily. Ather key concepts related to Big Data Analytics are:

Bulk synchronous parallel processing [17] is a model proposed originally by Leslie Valiant. In this model, processors execute independently on local data for a number of steps. They can also communicate with other processors while computing. But they all stop to synchronize at known points in the execution; these points are called barrier synchronization points. This method ensures that deadlock problems can be detected easily.

Large data streaming generated by thousands of data sources at high velocity, in high volume. It contains valuable potential insights and need to be processing real time to capture and pipe streaming data, but also to enrich, add context, personalize, and act on it before it becomes data at rest. These high-velocity applications require the ability to analyze and transact on streaming data.[10]

Large scale In memory computing, necessary to meet the strict real-time requirements for analyzing mass amounts of data and servicing requests within milliseconds an in-memory system/database that keeps the data in the random access memory (RAM) all the time [1].

High availability (HA) that is the ability of a system to remain up and running despite unforeseen failures, avoiding unplanned downtime or service disruption. HA is a critical feature that businesses rely on to support customer-facing applications and service level agreements.[11]

3 Key Aspects of Big Data Analytics

In recent years data, data management and the tools for data analysis have undergone a transformation. We have seen a significant increase in data collected by users thanks to web applications, sensors, etc. Unlike traditional systems, the type and the amount of data sources are varied. There is no longer just dealing with structured data, but also unstructured data from social networks, sensors, from the web, smartphones, etc. The acquisition of Big Data can be done in different ways, depending on the data source. The means for the acquisition of data can be divided into four categories: Application Programming Interface: the APIs are protocols used as a

[10] https://www.voltdb.com/fast-data.

[11] https://www.mapr.com/resources/high-availability-mapr.

communication interface between software components. Examples of APIs are the Twitter API, the Facebook Graph API and API offer by some search engines like Google, Bing and Yahoo! and the weather API. They allow, for example, to get the tweets related to specific topics (Twitter API) or examining the advertising content based on certain search criteria in the case of the Facebook Graph API. Web Scraping where data are simply taken by analysing the Web, i.e. the network of pages connected by hyperlinks. This has given rise to the term Big Data, that has become very popular, but its meaning often takes on different aspects. In general, we can summarize its meaning as a way to treat large volumes of data constantly increasing [7], an action that requires instruments for collecting, storage and analysis different from the traditional ones. In particular we refer to datasets that are so large to be not manageable by traditional systems, such as relational DBMS running on a single machine. In fact, when the size of a dataset is more than few terabytes, it is necessary to use a distributed system, in which the data is partitioned across multiple machines. Several technologies to manage Big Data have been created that are able to use the computing power and the storage capacity of a cluster, with an increase in performance proportional to the number of machines present on the same cluster. Those technologies provide a system for storing and analysing distributed data. Using redundancy of data and sophisticated algorithms, can work even in the event of failure of one or more machines in the cluster, transparently to the user. Distributed systems provide the basis for those systems. In fact a distributed architecture is able to serve as an umbrella for many different systems.

4 Popular Technologies for Big Data Analytics Utilizing Concepts of Distributed Computing

In the subsections below we discuss few popular open source Big Data technologies those are wideliy used to day across various industries.

4.1 Hadoop

The Hadoop Distributed File System (HDFS) [18] is a distributed filesystem written in Java designed to be run on commodity hardware, in which the data stored are partitioned and replicated on the nodes of a cluster. HDFS is fault-tolerant and developed to be deployed on low-cost machines. Hadoop is just one example of a framework that can bring together a broad array of tools such as (according to Apache.org): Hadoop Distributed File System that provides high-throughput access to application data; Hadoop YARN for job scheduling and cluster resource management; Hadoop MapReduce for parallel processing of big data. Hadoop, for many years, was the leading open source Big Data framework but recently the newer and more advanced

Spark has become the more popular of the two Apache Software Foundation tools. Hadoop can run different applications, including MapReduce, Hive and Apache Spark. Through redundancy of data and sophisticated algorithms, Hadoop can work even in the event of failure of one or more machines in the cluster, transparently to the user. Hadoop is an open-source software system used extensively in this area, offering both a distributed file system for storing information that one for their computing platform. The module supports multiple software for the analysis of data, including MapReduce and Spark. The substantial difference between these two systems is that MapReduce obliges to store the data to disk after each iteration, while Spark can work in main memory, exploiting the disc only in case of need. The Spark system, which is a high-level framework, provides a set of specific modules for each scope.

4.2 Yarn

YARN (Yet Another Resource Negotiator) is a main feature of the second version of Hadoop. Before YARN, the same node of the cluster, on which he was running the Job Tracker, took care of both of the cluster resource management is the scheduling of the task of MapReduce applications (which were the only possible ones). With the advent of YARN the two tasks were separated and were held respectively by the ResourceManager and AppliationMaster.

4.3 Hadoop Map Reduce

Hadoop MapReduce is a programming model for processing large data sets on parallel computing systems. A MapReduce Job is defined by: the input data; a procedure Map, which for each input element generates a number of key / value pairs; a phase of shuffle network; It reduces a procedure, which receives as input elements with the same key and generates a summary information from such elements; the output data MapReduce guarantees that all elements with the same key will be tried by the same reducer, since the mapper all use the same hash function to decide which reducer send the key / value pairs.

4.4 Spark

Apache Spark is a project that otherwise to Hadoop MapReduce does not require the use of your hard disk, but may enter directly into the main memory managing to offer performance even 100 times on specific applications. Spark offers a broader set of primitive compared to MapReduce, greatly simplifying programming.

5 Conclusion

A distributed computing system consists of number of processing elements interconnected by a computer network and co-operating in performing certain assigned tasks. When data becomes large, the database is distributed into various sites. The distributed databases need distributed computing to store, retrieve, and update data in a well coordinated way [9]. The advent of Big Data has led in recent years in search of new solutions for storing them and for their analysis. To manage Big Data, technologies have been created that are able to use the computing power and the storage capacity of a cluster, with an increase in performance proportional to the number of machines present on the same. In particular big data analytics is a promising area for next generation of innovation in the field of automation, with the ever increasing need of extracting value from data in several field of application. With that objetcive in mind various technologies/system have been evolved in last decade or so. The most used of these systems is Hadoop, which provides a system for storing and analyzing distributed data. YARN is a main feature of the second version of Hadoop, born to solve common problems. Hadoop Map Reduce, is designed for processing large data sets with a parallel and distributed algorithm on a cluster, and Spark performs in-memory processing of data. In this chapter an overview of technologies for Big Data analytics based on Distributed Computing concepts have been presented. With the increasing amount of data, the analytics will be ever more important in the decision-making process in several sectors allowing the discovery of new opportunities and increasing the quality of information.

References

1. Gartner. Hype cycle for big data, 2012. Technical report (2012) On the role of Distributed Computing in Big Data Analytics 11
2. Afgan, E., Bangalore, P., Skala, K. Application information services for distributed computing environments. Future Generation Computer Systems 27 (2011) 173–181
3. Cattell, R. Scalable sql and nosql data stores. Technical report (2012)
4. Brewer, E.A. Towards robust distributed systems (abstract). In: Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing. PODC ‘00, New York, NY, USA, ACM (2000) 7-.
5. Nessi: Nessi white paper on big data. Technical report (2012)
6. Dean, J., Ghemawat, S. Mapreduce: simplified data processing on large clusters. In: Osdi04: Proceedings Of The 6th Conference On Symposium On Operating Systems Design And Implementation, Usenix Association (2004)
7. IBM, Zikopoulos, P., Eaton, C. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. 1st edn. McGraw-Hill Osborne Media (2011)
8. Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., Tufano, P. Analytics: The real-world use of big data. Ibm institute for business value – executive report, IBM Institute for Business Value (2012)
9. Gilbert, S., Lynch, N. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News 33 (2002) 51–59

10. Zhang, H., Chen, G., Ooi, B.C., Tan, K.L., Zhang, M. In-memory big data management and processing: A survey. IEEE Transactions on Knowledge and Data Engineering 27 (2015) 1920–1948
11. Valiant, L.G. A bridging model for parallel computation. Commun. ACM 33 (1990) 103–111
12. Oracle: Big data for the enterprise. Technical report (2013)
13. Robinson, I., Webber, J., Eifrem, E. Graph Databases. O'Reilly Media, Incorporated (2013)
14. White, T. Hadoop: The Definitive Guide. 1st edn. O'Reilly Media, Inc. (2009)
15. Grover, P., Johari, R. Bcd: Bigdata, cloud computing and distributed computing. In: Communication Technologies (GCCT), 2015 Global Conference on, IEEE (2015) 772–776
16. Gartner: Pattern-based strategy: Getting value from big data. Technical report (2011)
17. Gandomi, A., Haider, M. Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management 35 (2015) 137–144
18. Amato, A., Venticinque, S. In: Big Data Management Systems for the Exploitation of Pervasive Environments. Springer International Publishing, Cham (2014) 67–89
19. Afgan, E., Bangalore, P., Skala, T. Scheduling and planning job execution of loosely coupled applications. The Journal of Supercomputing 59 (2012) 1431–1454