Big Data Analytics—Analysis and Comparison of Various Tools

Amit Gupta, Bhanu Prakash Dubey, Himani Sivaraman, and M. C. Lohani

Abstract Big data is the latest terminology in the computer world. The data collection is increasing day by day, and many technological changes can bring some new methods for decision-making process in many areas such as health and finance. As the complexities are increasing due to volume, veracity, variety and velocity, our focus is on the methods to calculate the value of data using various big data analytics technologies. The analytics process used with respect to big data can be improvised by using new algorithms, which enhance the analytical aspects and can be used to extract the hidden knowledge very efficiently and effectively.

Keywords Big data · Hadoop · HDFS · Spark · Map reduce

1 Introduction

Big data terminology is used for the collection of various data sets which are diverse in format and complexity. Due to its diversity, these huge data sets are very difficult to be stored and processed using traditional data processing tools or applications. Thus, we require some techniques or concepts with the help of which we can easily work on and use these data sets for various purposes. Big data analytics facilitate the collection of data from different sources, transforming them to such a format so that it becomes ready to be used by various analysts and eventually providing it to various organizations. Big data and machine learning altogether enhance the performance of

A. Gupta (✉) · B. P. Dubey · H. Sivaraman · M. C. Lohani
Department of CSE, Graphic Era Hill University, Dehradun, India
e-mail: amitgupta7920@gmail.com

B. P. Dubey
e-mail: bhanu.dubey@gmail.com

H. Sivaraman
e-mail: himanisivaraman@gmail.com

M. C. Lohani
e-mail: getmlohani@gmail.com

V. Goar et al. (eds.), *Advances in Information Communication Technology and Computing*, Lecture Notes in Networks and Systems 135,
https://doi.org/10.1007/978-981-15-5421-6_48

various industries like finance, healthcare, etc. This is because the price of data storage has been reduced and accessibility to high end and high performance computer becomes easy. Thus, various theoretical concepts of big data when implemented using machine learning tools give enhancements to many industries and business organizations.

Nowadays, the generation rate of the data is very fast. Approximately, around 90% of the data which is present in present world has been created in previous two years. In recent decades, the huge amount of data is been generated from various sources like:

1. Walmart handles more than 1 million customer transaction every hour
2. Popular social media platform Facebook uses, stores and analyzes around 30 plus petabytes of data which is all generated by its millions of users
3. Approximately, 48 h of new video are been uploaded to YouTube every hour
4. Amazon handles near about fifteen million user activity click per day that plays an important role for recommending various products to its customers
5. Various mail servers analyze around 294 billion emails to find the spam mails
6. Modern vehicles have more than 100 different types of sensors to monitor various things like fuel consumption, tire pressure, etc., and thus, every vehicle generates lots of sensor data that can be stored and processed on Cloud.

2 Big Data Characteristics

2.1 Volume

Volume means that the enormous information and data which is generated on daily basis increases in exponential, and this huge amount of data mainly represented in terabytes, petabytes or in some cases even in zetabytes. This information or data is so big that it cannot be handled, managed or controlled by using ancient methods or traditional methods of data managing techniques. For example, the size of data being generated by the interaction between humans and machines through various social media platforms.

2.2 Velocity

Velocity means the speed with which various sources will generate data on daily basis. This huge data is very enormous and continuous in nature. For example, on Facebook, there are around 1.03 billion active users daily which approximately increases around 22% each year. This concludes that how fast the number of users is increasing on social media platforms. These users are responsible for the fast growing of data on

Fig. 1 Data with missing values

Min	Max	Mean	SD
4.3	?	5.84	0.83
2.0	4.4	3.05	50000000
15000	7.9	1.20	0.43
0.1	2.5	?	0.76

daily basis. Simply, if you can handle the velocity, you will be able to generate various insights and will be able to take decision based on this real time generated data

2.3 Variety

In big data, many different types of data sources basically responsible for different types of data eventually contribute for the formation of big data. The data generated from various data sources can be structured, semi-structured or unstructured in nature. Traditionally, data was mainly stored in excel and databases, but nowadays, the data is collected in various formats like images, audio, video, sensor data, etc., and hence, this variety of semi-structured and unstructured data mainly creates problems related to storage, collecting, extracting information and analysis of data.

2.4 Veracity

Veracity [1] means that the data is in doubt or is uncertain of data availability due to the incomplete data or inconsistent data (Fig. 1).

Many a time, the data can be messy and untreatable. As in big data, the data occurs in many forms, and therefore, the quality and accuracy always remain a big problem. The volume is always responsible for the lack of quality and accuracy of data.

2.5 Value

With the volume, velocity, variety and veracity, we need to discuss one more V related to big data, this is value. It is basically the usefulness of the data. The features

and functions of big data include security, storing, analysis, exploring, visualization [2], modification and transactions. In today's world, there are various technologies and techniques [3] which can be used along with big data to perform faster and efficiently. Parallelism increases the speed at which big data can be processed, and it also increases the analyzing capabilities of the data. The usage of distributing computing [4] systems can be used for the efficient processing of big data mainly in real-time manner.

The various technologies used in Big Data which are treated as best four Apache Big Data Frameworks are described briefly in the following section.

3 Apache Hadoop

Apache Hadoop [5, 6] is basically an open source framework, which is written in Java. It is fault tolerant and scalable framework which provides batch processing techniques to be used in efficient way. It performs better than any other technique as it is capable of processing large volume of different forms of data on a group of various commodity hardware. Hadoop is mainly misunderstood as a system to store data, but instead, it is a technology or method which possesses capability for storing large volume of data along with processing of large amount of data.

Hadoop is technology that is designed to process big data which is combination of both structured and semi-structured data which is available in huge volume. It also provides analytical techniques and computational power required to work with large and diverse form of data.

Hadoop framework is an example of cluster which comprises one master node and many worker nodes. This master node is a composed of both Name Node and Job Tracker Node, whereas Worker Node can act as both Data Node—responsible for storage of data and Job Tracker—responsible for monitoring jobs. It also contains Secondary Name Node which is the replication of Name Node. The responsibility of the Secondary Name Node is to take snapshot of Primary Name Node directory information at regular interval of time. This can be used in place of Name Node to restart the faulty or failed Name Node (Fig. 2).

Hadoop consists of two main components: Hadoop Distributed File System (HDFS) and Map Reduce [7].

4 Hadoop Distributed File System (HDFS)

HDFS is used for storage and is fault-tolerant mechanism that stores large size files from terabytes to petabytes across different terminals in distributed manner. The default value of replication is 3 that can be increased according to the sensitivity of data being stored. It splits big file into large block size of 64 MB (can be changed to 128 MB) and can be stored independently on multiple nodes. Its main responsibility

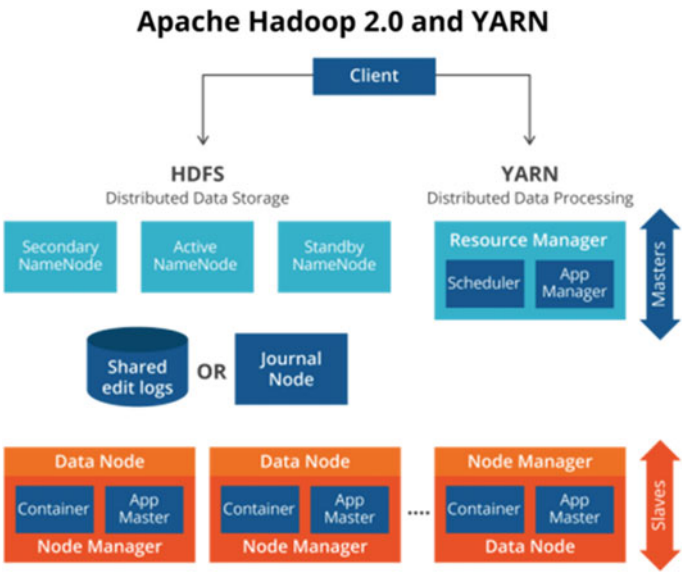

Fig. 2 Apache Hadoop and Yarn

is to ensure the availability of data even during the failures of the host machine. It is also used to store immediate processing results. HDFS is mainly suitable for the distributed storage and processing. Hadoop provides a command interface to interact with HDFS for the streaming access to file system data (Fig. 3).

HDFS provides an automatic fault detection mechanism which improvises its mechanism of recovery process during disaster. HDFS includes large number of

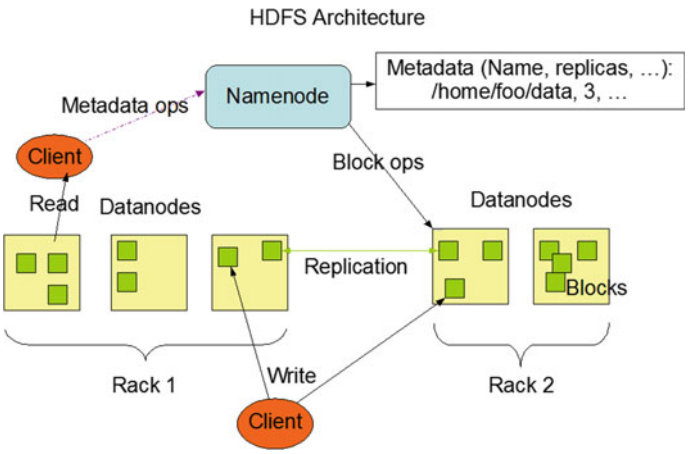

Fig. 3 HDFS architecture. *Source* hadoop.apache.org

various hardware, and thus, failure of any component is an issue. Therefore, it provides an efficient recovery system to facilitate efficient working of the Hadoop system. The processing methodology of HDFS is such that it always selects node for processing local node to reduce network traffic and increase throughput.

5 Map Reduce

Hadoop Map Reduce [8, 9] is basically a software framework for easily providing various processing tasks that involves huge amount of data. It basically facilitates parallel execution of application on large clusters in fault-tolerant manner. Map Reduce [10, 11] is programming structural model for writing tasks which can be executed in parallel fashion on multiple nodes. It also provides analytical capabilities for complex data. Traditional model has a limitation as it cannot provide mechanism to process huge volumes of scalable data, and on the other hand, the centralized system provides too much of a bottleneck while processing multiple files simultaneously. Google has developed an algorithm to solve this issues, and this technique is called Map Reduce [11, 12] which divides the task into small part and assigns them to different nodes. After processing, these individual results are combined to give the integrated output. The Map Reduce algorithm consists of two important activities: Map and Reduce [5, 13]. Map converts the data sets into individual elements of key, value or tuple. Reduce collects the output from each mapper and combines them. The most important benefit of using Map Reduce is that it provides an easy mechanism and method to distribute data processing on multiple and different computing nodes.

6 Apache Storm

It is a framework which mainly focuses on low latency. It provides an efficient and better option for processing which actually requires real-time processing. It provides an efficient methodology that works on huge amount of data and reduces latency in comparison with other frameworks. Storm has facilities such as real-time analytics, online machine learning, continuous computation and ETL, and it is scalable, fault tolerant, guarantees efficiently processing of data. There are certain features that make storm more powerful tool rather than Hadoop like fault tolerant, scalable, fail fast, auto restart approach, support multiple languages and Json, support for direct acylic graph (DAG) topology, etc.

7 Apache Samza

It is a stream processing framework that is strongly associated with Apache Kafka messaging system. It is designed specifically to enhance the benefits of Kafka's specific architecture. Like other technologies, it also uses fault-tolerant mechanism for buffering and storage. For the purpose of resource negotiation, it uses YARN [14] along with its rich features.

8 Apache Spark

Apache Spark [15, 16] is an example of all purpose cluster computing system which possesses huge and large number of libraries and APIs for various programming languages such as R, Python, Scala and Java. Unlike Hadoop, it is very fast and efficient in processing and accessing data from the storage. It can be implemented by using Hadoop or without Hadoop. It mainly focuses on quick execution of the task by implementing the methodology of batch processing workload in memory computation. It can be implemented as standalone cluster and can be used with Hadoop as an alternative to Map Reduce. The main component of Spark [17, 18] is driver program, cluster manager and worker node. The driver program is on the spark which starts the execution of any application. The cluster manager allocates all the resources. Lastly, Worker Node does all the processing. Some properties of the spark which makes it better than Hadoop are its high speed, high performance, high query optimization. It can run on any platform, has a large library set and data pipelining facility.

9 Apache Flink

Apache Flink is a platform which is categorized as open source; unlike any other framework, it has a flow engine for streaming data which also provides a methodology for communication, fault tolerant and distribution of data on various distributed computations over streaming data. This framework of data analytics is wholly compatible with Hadoop. Flink has the capability to execute both streaming processing and batch processing [19, 20] without any difficulty.

Because of the micro-batch architecture of spark, it is not suitable for many use cases. It is also enriched by the batch and stream processing capabilities. Apache Flink provides low latency, high throughput and real transactional processing. The architecture of Kappa forms the basis for working of Flink. The benefit of using Kappa architecture is it has only single processor—stream, which considers various input as stream, and the streaming engine present in the Kappa processes the entered

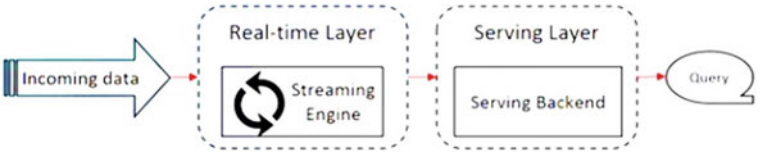

Fig. 4 Apache Flink architecture

Table 1 Comparison of various tools

	Execution model	Supported language	In-memory processing	Low latency	Fault tolerance	Enterprise support
Map reduce	Batch	Java	No	No	Yes	No
Storm	streaming	Any	Yes	Yes	Yes	No
Spark	Batch and streaming	Java, Scala, Python, R	Yes	Yes	Yes	Yes
Flink	Batch and Streaming	Java, Scala	Yes	Yes	Yes	No

data in real-time fashion. The processing of batch data is treated as a special case in Kappa. The diagram specified below gives the architecture of Flink (Fig. 4; Table 1).

10 Conclusion

This paper specifies various comparisons of the tools that can be used in big data analytics. According to the comparison chart given above, it is clear that Map Reduce technique is better only for batch processing system, whereas Spark and Flink can work efficiently on batch processing as well as on streaming data. Fault tolerance is provided in all the techniques, but again, Map Reduce does not support in-memory processing and low latency. According to the survey done on various technologies, Spark is the most efficient framework that can give efficient and accurate results.

References

1. Demchenko Y, Grosso P, de Laat C, Membrey P. Addressing big data issues in scientific data infrastructure. In: 2013 international conference on collaboration technologies and systems (CTS), San Diego, 2013. IEEE, pp 48–55
2. Cox M, Ellsworth D. Managing big data for scientific visualization. In: ACM Siggraph '97 course #4 exploring giga-byte datasets in real-time: algorithms, data management, and time-critical design, August, 1997
3. Bekkerman R, Bilenko M, Langford J (2011) Scaling up machine learning: parallel and distributed approaches. Cambridge University Press, Cambridge

4. Ni Z Comparative evaluation of spark and stratosphere. Thesis, KTH Royal Institute of Technology; 2013
5. Bu Y, Howe B, Balazinska M, Ernst MD (2010) HaLoop: efficient Iterative data processing on large clusters. Proceedings VLDB Endowment 3(1):285–296
6. Jakovits P, Srirama SN (2014) Evaluating MapReduce frameworks for iterative scientific computing applications. In: 2014 International conference on high performance computing & simulation; 2014. pp 226–33
7. Vavilapalli VK, Murthy AC, Douglas C, Agarwal S, Konar M, Evans R, Graves T, Lowe J, Shah H, Seth S, Saha B, Curino C, O'Malley O, Radia S, Reed B, Baldeschwieler E. Apache Hadoop YARN: yet another resource negotiator. In: Proceedings of the 4th annual symposium on cloud computing; 2013
8. Fernández A, del Río S, López V, Bawakid A, del Jesus MJ, Benítez JM, Herrera F (2014) Big data with cloud computing: an insight on the computing environment, MapReduce, and programming frameworks. Wiley Interdiscip Rev Data Min Knowl Discov. 4(5):380–409
9. Lin J, Kolcz A. Large-scale machine learning at twitter. In: Proceedings of the 2012 ACM SIGMOD international conference on management of data; 2012. pp 793–804
10. Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. In: Proceedings of the 6th symposium on operating systems design and implementation; 2004
11. Malewicz G, Austern MH, Bik AJC, Dehnert JC, Horn I, Leiser N, and Czajkowski G (2010) Pregel: A system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD international conference on management of data; 2010. pp 135–45
12. Attenberg J (2015) Conjecture: scalable machine learning in Hadoop with Scalding. 2014. https://codeascraft.com/2014/06/18/conjecture-scalable-machine-learning-in-hadoop-with-scalding/. Accessed 1 Jun 2015
13. Zaharia M, Chowdhury M, Das T, Dave A (2012) Fast and interactive analytics over Hadoop data with Spark. USENIX Login 37(4):45–51
14. White T (2012) Hadoop: the definitive guide, 3rd edn. O'Reilly Media, Inc., Sebastopol, CA
15. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark (2010) Cluster Computing with Working Sets. In: Proceedings of the 2nd USENIX conference on hot topics in cloud computing
16. Cai Z, Gao J, Luo S, Perez LL, Vagena Z, Jermaine C. A comparison of platforms for implementing and running very large scale machine learning algorithms. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data (SIGMOD'14) 2014, pp 1371–1382
17. Zhang H, Tudor BM, Chen G, Ooi BC (2014) Efficient in-memory data management: an analysis. Proc VLDB Endowment 7(10):6–9
18. Singh J (2014) Big data analytic and mining with machine learning algorithm. Int J Inform Comput Technol 4(1):33–40
19. Ousterhout K, Rasti R, Ratnasamy S, Shenker S, Chun B (2015) Making sense of performance in data analytics frameworks. In: Proceedings of the 12th USENIX symposium. On networked systems design and implementation (NSDI 15)
20. Shahrivari S, Jalili S (2014) Beyond batch processing : towards real-time and streaming big data. Computers 3(4):117–129