THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Comparative Analysis of Big Data Projects For Solving Computational Challenges In Data

Mamoon Rashid[1], Amanpreet Kaur[2], Mir Mohammad Yousuf[3], Rajeev Puri[4]

[1, 2, 3]Assistant Professor, School of Computer Science & Engineering, Lovely Professional University, India.

[4]Assistant Professor, Department of Computer Science, DAV College, Jalandhar, India.

Abstract- More than 75% of the inventions are made to satisfy the business requirements worldwide. Most recent invention are used to satisfy one more business process i.e. targeting customer which is Big Data. Although, there is not any particular definition for Big Data but still in this paper an effort is made to define Big Data and it's respective technologies like Hadoop, Flume, Sqoop, Hive, Pig, Mahout, Oozie etc. In this paper, the authors has given an outline of big data technologies along with their solution for data faster computations. The authors have structured the data generated in terms of generations so as to predict the nature and various challenges and later provided their solutions in terms of various big data projects.

I. Introduction

Decades ago **Sir Timothy John Berners-Lee** and IBM were working on a special technology HTML which later on lead the foundation of World Wide Web. Technology like Search Engine, Web Crawling, and Social Media has made Internet a huge phenomenon [1]. Everything is available at the click of a button in any part of the world. But now, in today's world this phenomenon has gave rise to a new problem i.e. DATA DATA EVERYWHERE. Today in the silicon age where each and every machine is getting its electronic versions and everything is getting digital. Each and every individual as well as machines connected to each other leads to a new kind of explosion, DATA explosion [2].International Data Corporation(IDC) estimatesthe size of the "Digital Universe" at 0.18 Zetta Bytes in 2006 and forecasted a tenfold growth by 2020 to 40 ZB.

II. How Storage Data is a Problem

Imagine you have 1GB of data that you need to process. The data are stored in a relational database in your desktop computer and this desktop computer has no problem handling this load. Then your company starts growing very quickly, and that data grows to 1GB, and then 100GB.And you start to reach the limits of your current desktop computer. So, you scale-up by investing in a larger system and you are then OK for a few more months. When your data grows to 10TB, and then 100TB approaching to the limits of your system. Moreover, you are now asked to feed your application with unstructured data coming from sources like Facebook, Twitter, RFID readers, sensors, and so on which doesn't get fit into your relation databases but we need to derive information from both the relational data and the unstructured data as soon as possible [3].

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

III. Failure of Traditional Large-Scale System

Traditionally, computation is processor bound [4]. For decades, the primary push was to increase the computing power of a single machine i.e. Faster processor, more RAM But it can be done in a limit only and it costs much also

Let us take an example of conventional computation system to process a file of 1000 GB stored in a SATA hard disk with a speed of 100MBpS with 4 I/O chunks.

So, it will access data as:

100 MB in 1 sec

1000 MB< => 1 GB (approx.) in10 seconds

Therefore, 250 GB in 2500 sec

Because of 4 I/O chunks it will access 250*4=1000 GB i.e. 1 TB in 2500 secs approx. 41 minutes

Now imagine the scenario in which we have to process PB'S of data with conventional computing systems and this just for the access of the data afterwards we need to do some kind of processing also using this data.

This lead to the foundation of **BIG DATA.**

Now consider same scenario with Big Data architecture.

Just split the data to different nodes which takes 100 MB of data as each on the same commodity hardware.

So, the current Equation will be:

Time will reduce to 10 times i.e. 2500/10 =25 secs each node and now imagine the scenario where millions of nodes are working together.

According to Bernard Marr Big Data is the digital trace generated in this digital era while using digital technology which can be used and analysed to become smarter. The driving force behind this is the access to the ever-increasing data and the technology to mine that data [5].

Most of the people miss interpret big data by considering only its one parameter i.e. volume. But in actual there are total 4 main parameters by which big data can be understood and these parameters are called V's of Big Data [6].
These are Volume, Velocity, Variety and Veracity, wherever Volume refers to the amount of the data i.e. is being generated. According to a report, by 2020 world's total data may cross the 35 ZB where

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

each day we are generating the 2.5 Quintillion of data.Velocity defines the pace at which the data is produced from its source, analysis shows that nearly 400 hours of video is being uploaded to YouTube per 60 seconds. Whereas world's total current data is 7.9 ZB and it will increase 343% to become 35 ZB till 2020.Variety defines the type of data i.e. structured, semi-structured or unstructured like text, pictures, videos, audios, data from sensors or machines etc. Veracity deals with the Ambiguous data i.e. whether the data available data is trustable or not and if yes up to how much extent e.g. while chatting on social media text like hmm, lol which do not have any meaning are common.

Besides of these four V's, there is also fifth V called Value which refers to the ability to find hidden and useful value out of the random data.

A Big problem related to big data is 75-80 percent of the data available is unstructured i.e. which cannot be stored in the conventional relational Databases. Therefore, a different technology must emerge in order to serve as solution to this problem.

Apache Hadoop is an open source project designed specifically for managing Big Data. Hadoop can be differentiated with its two main versions i.e. Pre-Hadoop 2.2 and Hadoop 2.2 [7].

Pre-Hadoop 2.2 can be taken as the implementationof initial solution which is suggested to manage the big data. It has two mains component which are a Distributed File System i.e. Hadoop Distributed File System(HDFS) to store the data and a Programming model i.e. MapReduce to process that data.

Architecture suggests to use the commodity hardware which makes it much cost effective. A large number of processing systems are connected together in form distributed systems collaborating with each other in order to process the data where each system is termed as a node. Collection of 30-40 nodes is called a rack and collection racks is termed as cluster. Network bandwidth between two nodes in same rack is greater than the nodes in different racks [8].

HDFS runs on top of each individual node and performs well with larger files as compared to the smaller files and does sequential data access rather than the random data access. The main reason behind these is Seek time[2]. More the size of the file less will be the seek time and alsosequential data access takes less seek time rather than the random data access. Less seek time makes Hadoop more cost effective. Blocks are used to store data in HDFS where default size of each block is 64 MB which can be altered as according[3]. The main advantage of storing data into blocks that if in case data is larger than the disk size then it can be stripped into parts and then stored to different blocks on different disks and also if in case data to be stored is smaller in size than the block size then rest of the space will not be wasted but will be made available to the system. Due to use of commodity hardware it is very much prone to the hardware failure which may lead to the data loss. For this, replication of data blocks is done to the other nodes. By default, data is replication is done at 3 different nodes [9].

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

There are two types of nodes in HDFS, Name Node and Data Node. Whereas Name node can only be one per cluster but the data node can be many. Name Node stores the metadata of Data nodes (which block is situated to which Data node etc.) and namespaces. Expensive hardware can be used for Name node even for replication. Data nodes stores the actual data in form of blocks into it. There can be many data blocks situated into a single Data node. Record of each block in a Data node is stored in Name node as metadata. Each Data node pings to Name node periodically to ensure its healthy presence. Commodity hardware is used for all Data nodes and software level replication is done.

MapReduce in Pre-Hadoop 2.2 also known as MapReduce version 1 has its own built in own resource manager and scheduler. It is software framework which is capable easy application development which can process huge amount of data. It has two types of nodes, Job Tracker and Task Tracker. Job Tracker manages the MapReduce version 1 and can only be one per cluster. Its primary operation is to take the job from client, scheduling Map and Reduce tasks to the Task Trackers, Monitoring the failed task and reschedule it[4]. Task tracker act as the descendant of JVM while executing the Map and Reduce tasks by communicating with Job tracker and Task tracker [10].

After the success of Pre-Hadoop 2.2, Hadoop 2.2 comes as its upgraded version.The main feature of Hadoop 2.2 is that its resource manager and scheduler is external to framework which becomes its third primary component called YARN extended as Yet Another Resource Negotiator. Due to YARN, there is no need of Job Tracker and Task tracker. The main features of YARN are resource management, generic and efficient scheduling and workload balance.

Because the hardware capability of each node might be different from the another one in a cluster, so in Pre-Hadoop 2.2 number of slots have to be assigned to Map and Reduce tasks as according to the hardware capabilities but due to YARN it can be automatically done through a Node Manager which runs on the top of each node. When a task is assigned to the node, it invokes the Application Master which is part of each node and it negotiates with resource manager for resources, fetch the resources to the local container of the node. Only Name node is a single point of failure but in Hadoop 2.2, two Name nodes are facilitated, one active and another with standby. Only one Name node can remain active and this decision is taken by Journal Nodes which are always odd in number with default three in count. They decide together that active one is lost and now back up should be take over.

IV. Big Data Projects

There are certain other projects which support Hadoop for its efficient operation. Due to open source behaviour of Hadoop these features are developed by different companies for their interest and later accepted by other communities like HBase, Pig, Hive, Zookeeper, Flume, Sqoop, Oozie,.
A brief introduction for these components is discussed here.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Pig: It is developed by Yahoo in 2006 but later by Apache Software Foundation in 2007. Pig offers PigLatin as High-Level Language which is a Data Flow Language in nature i.e. programmer connects different components to each other showing the flow of data. Main feature of pig is that it can operate on complex data structure also. Unlike SQL it doesn't require any schema which makes it suitable for Unstructured Data although inputs from structured schema is still supported. It also supports the relational algebra operations like cross product etc [11].

Hive: It is developed by Facebook which treats Hadoop as a Data Warehouse. As it has similar dialects as SQL queries so it is a declarative language. It's working mechanism is little opposite too Pig as rather than mentioning data flow, user mentions the inputs and Hive itself generates the data flow which will produce the same output as mentioned by client. Schema is required for Hive but it is not limited to single schema[12].

Eventually both Pig and Hive are little similar in nature as both converts the High-level language to the Map and Reduce jobs. It helps programmers working with MapReduce jobs in java and other language as program's code length becomes much smaller.

Flume: It is open software program developed by Cloud Era. It provides services for moving data around cluster with large volume as soon as it is generated by its source. One example can be gathering of Log files from a machine or sensor and then storing it to some storage system like HDFS [13].

Sqoop: It is an open source software application which is used to transfer data between relational database and Hadoop using JDBC. First it accesses the database system in order to understand the available schema of database system. Then it generates the MapReduce application to import and export the data. After development of application a java class is generated whose function is to encapsulate each row imported individually. Source code for this can be accessed to develop additional MapReduce application [14].

Oozie: An open source job control component whose function is to control the Hadoop jobs. Its working mechanism is based upon some set of actions mentioned in the form of acyclic graph. Workflow is written in HPDL i.e. XML ProcessDefinition Language and stored in workflow.xml.

V. Conclusion

In this paper, the authors have structured data generation from various sources in systematic manner and provided solutions of big data challenges in terms various Apache projects. The authors have made finding that it is computation of data which is challenge at present times in comparison to huge storage of data. The authors believe that this work will help beginner researchers as a pathway for various problems where research needs to be done.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

References:

[1] Mahrt, M., &Scharkow, M. (2013). The value of big data in digital media research. Journal of Broadcasting & Electronic Media, 57(1), 20-33.

[2] Rubin, G. D. (2000). Data explosion: the challenge of multidetector-row CT. *European journal of radiology*, *36*(2), 74-80.

[3] Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., &Indragandhi, V. (2015). Unstructured data analysis on big data using map reduce. *Procedia Computer Science*, *50*, 456-465.

[4] Abelson, H. (1978, October). Lower bounds on information transfer in distributed computations. In *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)* (pp. 151-158). IEEE.

[5] Marr, B. (2015). *Big Data: Using SMART big data, analytics and metrics to make better decisions and improve performance*. John Wiley & Sons.

[6] Sheth, A. (2014, March). Transforming big data into smart data: Deriving value via harnessing volume, variety, and velocity using semantic techniques and technologies. In 2014 IEEE 30th International Conference on Data Engineering (pp. 2-2). IEEE.

[7] Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R. &Saha, B. (2013, October). Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing* (p. 5). ACM.

[8] Lee, S. J., Sharma, P., Banerjee, S., Basu, S., & Fonseca, R. (2005, March). Measuring bandwidth between planetlab nodes. In *International Workshop on Passive and Active Network Measurement* (pp. 292-305). Springer, Berlin, Heidelberg.

[9] Dittrich, J., Quiané-Ruiz, J. A., Jindal, A., Kargin, Y., Setty, V., &Schad, J. (2010). Hadoop++: Making a yellow elephant run like a cheetah (without it even noticing). *Proceedings of the VLDB Endowment*, *3*(1-2), 515-529.

[10] Mackey, G., Sehrish, S., Bent, J., Lopez, J., Habib, S., & Wang, J. (2008, November). Introducing map-reduce to high end computing. In 2008 3rd Petascale Data Storage Workshop (pp. 1-6). IEEE.

[11] Fuad, A., Erwin, A., &Ipung, H. P. (2014, September). Processing performance on apache pig, apache hive and MySQL cluster. In Proceedings of International Conference on Information, Communication Technology and System (ICTS) 2014 (pp. 297-302). IEEE.

[12] Barbierato, E., Gribaudo, M., &Iacono, M. (2013, December). Modeling apache hive based applications in big data architectures. In Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools (pp. 30-38). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[13] Hoffman, S. (2013). Apache Flume: distributed log collection for Hadoop. Packt Publishing Ltd.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

[14] Chen, L., Ko, J., & Yeo, J. (2015). Analysis of the influence factors of data loading performance using Apache Sqoop. KIPS Transactions on Software and Data Engineering, 4(2), 77-82.

[15] Islam, M., Huang, A. K., Battisha, M., Chiang, M., Srinivasan, S., Peters, C., ...&Abdelnur, A. (2012, May). Oozie: towards a scalable workflow management system for hadoop. In Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies (p. 4). ACM.