International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 02, February-2018

Distributed Data Streams in Big Data Environment

Sridhar Bandavaram
Global IT Inc
Irving, TX

***Abstract*— The modern age tools facilitate the production of a large amount of data and the big data application increasingly needs to act on the data in the real time. However, the prevailing system limits the processing of the disparate data arising from different sources. Additionally, the traditional facilities increase single sourced data that is not provided in the real-time and has extensive fault repair mechanism that is difficult to maintain, restraining the traditional infrastructure. The current day technologies such as social networking input, trading, system monitoring, and the Internet of Things require powerful and flexible open source platforms such as distributed data stream. For this purpose, the distributed data streaming system has been attributed as capable of handling large-scale data being generated at high velocity across the varied portals. In the present study, an overview of distributed data streaming processors in the big data model was performed, which indicated towards its ability to decrease the latency in the big data analytics. In addition to this the various framework and architectures of the big data system were compared to elaborate on the advantages and limitations in the process of using distributed data streaming for the big data. The study established the need for distributed data streaming for the organizations requiring high-velocity data interpretation and analysis in the real time.**

Keywords: Distributed data streaming, big data, big data framework, data scalability.

I. INTRODUCTION

The existing paradigm in database management is largely developed around the model of centralized data. Centralized data elaborate the method of organizing, accessing, indexing, and holding data query in a central location using a single machine or a small group of closely linked machines. Earlier studies on the data-streaming algorithms have established the usage of centralized model computation, which is the usage of stream-processing engine as direct contact to all records of streaming data. However, the centralized stream-processing simulations overlook problems such as communication efficiency, inadequacy for prototypical data streaming application, and monitoring for IP-network and sensor net. To address the problems arising from the centralized data query sources, a majority of data streaming today has evolved to distributed data method.

II. ABOUT DISTRIBUTED DATA STREAMS

The distributed data streaming is capable of managing large-scale data collected from wide-ranging areas via nodes over multiple servers or locations. The data collected through the distributed data streaming happens at a much faster pace than in the centralized stream processing system. It is even capable of processing queries on a much larger scale in a comparable smaller time frame than the traditional centralized system. The functioning of the parallel or distributed data entails the usage of distributed query execution over multi-nodal architecture and the distributed nature of the streaming algorithm guarantees efficiency in terms of processing time and space. However, the main hindrance observed in the case of distributed data streaming arises from the physical distribution of the imposed network infrastructure which is required for managing such large-scale data. Also, communication constraints may arise during the process.

The technological advancements and increased connectivity of people via their smart devices has resulted in the availability of large data resources for the companies and government institutions. The dynamic lifestyle of the people today hasenabledlarge-scaleuser-generated data content available on the net from the omnipresence of devices such as wearable smart devices, mobile phones, and other smart devices created to facilitate the lifestyle of the people. The technologies have led to an abundance of information available for the organizations. As the data made available by the communication devices are stored by the servers to create user profile, valuable for organizations and hold information that can be only used by them for short period of time. This large-scale data is termed as big data and is required to be processed as quickly as possible to harness possible benefits from the prevailing market trends.

The distributed data streaming allows discrete inflow of the data stream where the large information gathered in form of the big data is distributed for computation by performing processing of data on arrival. The distribution allows improving the scalability of the data and also reduction of the level of fault-tolerance by allowing straggler tasks to be handled at a time. To improve the scalability of the data, the distributed data system allows several processing frameworks to be deployed on clouds utilizing the characteristic benefit of resources elasticity of the distributed system. The nodal distribution structure further allows proper allocation of resources by exploiting the facility of idle capacity match allocating additional resources available on the cloud whenever required.

III. SCOPE OF THE STUDY

The current study establishes the need for the distributed data stream in the world of technology today. The study will further highlight the advantages of real-time scalability of the distributed data streaming in increasing its advantages and applicability in the big data

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 02, February-2018

environment. Additionally, the present study also works towards identifying the importance of distributed data streaming across the various frameworks of big data to stress upon the fault tolerant and high availability features of the distributed data streaming system. The study will further highlight the strengths and shortcomings of the distributed data system and analyze the architecture and systems being followed by various companies across industries utilizing the distributed data streams.

IV. NEED OF DISTRIBUTED DATA STREAMS

The distributed data streaming process is the need for the modern information system as data stream essentially helps the data mining process. It solves the difficulties faced by the static methods of the data mining process. Distributed data stream essentially decreases memory consumption and increases the computational efficiency through the single linear scan and low space complexity. The distributed data streaming speeds up the data mining process as it has the ability to adapt dynamically to the changing flow of the data. The distributed structure of the data streaming further enables it to effectively handle the null values and noise in the data. Further, the summarized data structure generated through the distribution system is versatile as the data is constructed with the help of classifiers that are automatically adjusted. The automatic adjustment helps in maintenance of the accuracy of data as the data arises from a large number of sources. The distributed data streaming system is relevant to the nature of data arising from its sources, which is its relevance is time dynamic. Distributed data stream allows efficient flow of data their classified algorithms. The process analyzes multidimensional data and processes data timely, comprehensively, and accurately. The distributed data streams are applied for enterprise and ISP network security enhancement that increases organizations ability to track network-wide traffic patterns. Distributed data streaming are required in the field of detecting anomalies and protect network infrastructure for malicious attacks.

The big data infrastructure needs streamlining by an analytical system that allows important timely mining of crucial information. The distributed system of data streaming further enables the companies to scale through a large volume of data and minimizes the time spent in the processing pipeline on single data item frame in the process. Organizations these days are using distributed infrastructure components that are built on an asynchronous network that are engineered with Java Virtual Machine. Such system allows real-time integration of data widely used by the organizations to construct user profile such as clients billing information, user clicks, and even unstructured data contents such as text messages and images.

The collected data is further used by the organizations to maintain materialized view for future database and initiate Graphic User Interface (GUI) such as trending topics as in the case of Twitter. The nodal information structure and distributed processing allows cost-effective cluster processing for the companies and are the largest enablers for big data analytics, as it allows the facility of pre-processing of the large background data. The data is cleansed as per the requirement and the redundancies and contradictions can be eliminated in the foreground through the system to finalize the transformation of data into the most appropriate result raised by the ad-hoc queries(Fujitsu, 2016).

V. FRAMEWORKS

There are primarily three frameworks in the big data, Batch-only frameworks or the Apache Hadoop, Stream-only frameworks that comprise of Apache Storm and Apache Samza, and Hybrid frameworks such as Apache Spark and Apache Flink. Hadoop framework uses Hadoop Distributed File System (HDFS) to synchronize the storage and repetition of data across cluster nodes even in the conditions for host failure that is inevitable. Stream processing model are based on event-structured processing of data and the results of the queries are provided immediately. The query runs through the stream processing continuously and is ended by user instructions only. Apache storm allows near to real-time processing and allows large workload of data to be processed with extremely low latency. Stream processing model Samza allows topic streaming by data entering the Kafka system. The topics are distributed among nodes to facilitate efficient data mining.

The hybrid processing system of batch and stream processing is combined in the Spark and Flink frameworks. The spark model uses Resilient Distributed Datasets (RDD) to work with datasets. Spark allows immutable structures with memory setup to trace the lineage of the data through distributed nodes that allow high memory usage and processing of data at incredibly high speed. The Flink framework reverses the process of Spark that is stream processing is done first also known as Kappa architecture followed by a batch process. Flink processes data on entry to entry basis to minimize latency and high throughput.

Thus, for data structures required by the organization, batch-only workload system such as Hadoop is used for non-time sensitive data. For stream only workloads in the organization, storm is used. However, it can deliver duplicate data. Samza, on the other hand, provides system flexibility and easy multi users as well. Spark on the other hand is useful for mixed workloads, and micro-batch processing for streaming data. Flink model provides optimized stream and batch processing support. The models allow organizations to choose their best fit on the amount of data to be processed and time bound for the query. Architecture and systems followed by companies for distributed data streams in different industries

The Apache Hadoop system is distributed among two servers, these are master and slave. The system functions on the basis of MapReduce framework, where the master node is accountable for accepting jobs from the customer that is further divided into smaller tasks and worker node then execute the tasks assigned. The Hadoop systems are used by Cloudera and IBM.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 02, February-2018

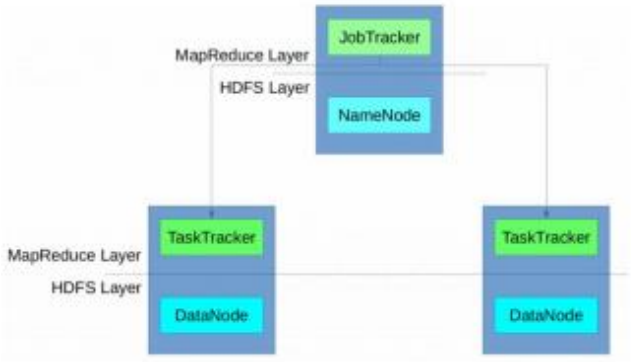

Figure 1 Architecture of Hadoop

The Apache Spark is evolved form of MapReduce framework from the Hadoop structure; it is set of apparatuses and software structured in a defined architectural format. The system is being used by companies such as MyFitnessPal and Conviva. The architecture of Spark enables it to work on a wide data range at the same time dividing the task between the nodes for performing in-memory analysis.

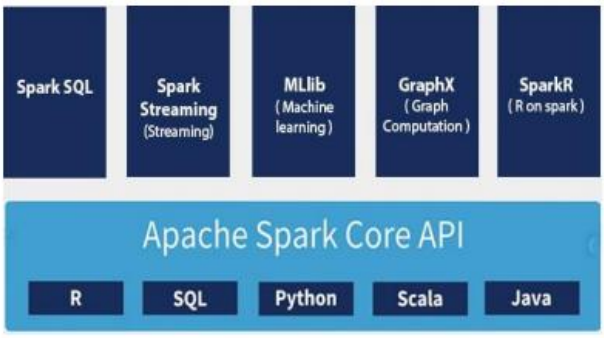

Fig. 5. Spark framework ecosystem.

Figure 2 Spark Network Architecture

The Apache Storm architecture cluster has three distinct nodules “Nimbus” that is equivalent to Job Tracker in the Hadoop structure, “Supervisor” initiates and terminates the processes within the structure, and “Zookeeper” node that shares the coordinated services in the storm cluster. In Storm architecture, MapReduce jobs are replaced with topologies of data streams.

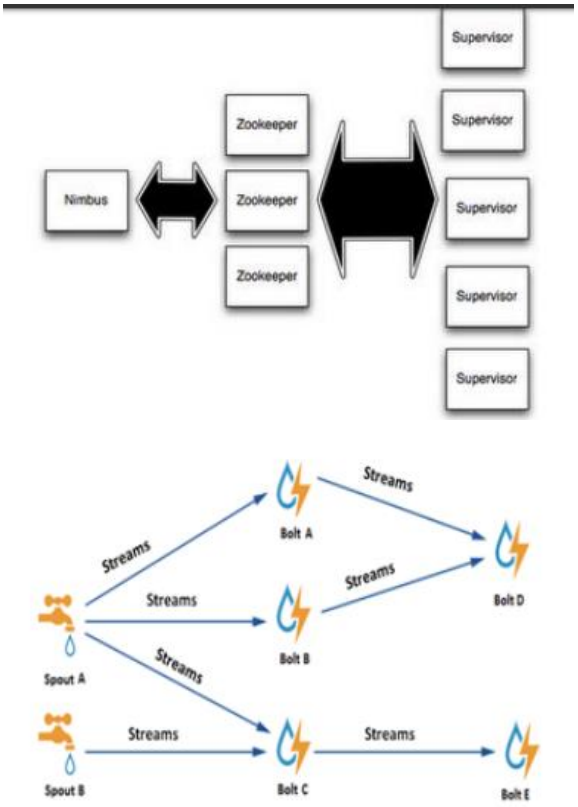

Figure 3 Architecture and topology of Storm Arrangement

Different industries for real-time processing of data use two basic forms of architecture that are Lambda and Kappa. The lambda architecture combines batch processing and real-time processing in a single framework. The architecture provides better results with low latency and is made of three layers in the lambda structure represented in the figure below

The batch layer stores the data sets and initiates arbitrary batch views, next is the serving layer that integrates between the batch and the speed layer, which processes data with updated services layer.

The Kappa architecture, on the other hand, has a more dedicated approach for processing data than the lambda system. The Kappa system uses two layers in its architecture to reduce the complexity of structure and employ single code path layer.

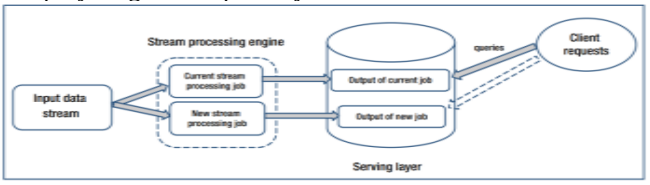

Figure 4 Architecture in Kappa System

Kappa system in companies such as Ericsson allows users to develop, check, debug, and control across Spark, HDFS, Storm, Samza, and Kafka frameworks.

Apache Kafka is another distributed streaming platform used to publish and subscribe to stream messages in clustered or single server environment. Apache Kafka has four core APIs Producer API, Consumer API, Streams API and Connector API.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 02, February-2018

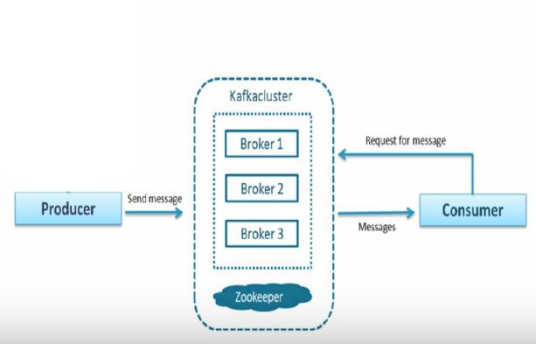

Figure 5 Apache kafka in Cluster

VI. STRENGTHS AND SHORTCOMINGS

Big data is a universal term used for the strategies and technologies that encompasses data tasks such as gathering, organizing, processing, and creating information or insight from the large data set gathered by the organization. However, in the recent times, the problem of working with data has also increased as modern technologies have allowed accumulation of large scale of data made available to the company.

Big data with distributed data system are useful for the organization as it allows huge amount of data available to the organizations to be centralized continuously through streaming techniques. The technique allows data collection and sensor net monitoring through efficient tracking of global queries and measuring collection from sensor net through clever in-network processing techniques. Big data along with distributed data system can further be employed to monitor large stream of continuous data. It further allows the organizations to detect problems such as fraud and abuse in real-time. Other issues for the companies include pervasiveness of data, computation and storage power of limited computer available to the companies, and value of computations. Big data methods or processing frameworks allow computation of data in the system and formulation of insight from the large-scale data gathered from individual data points. Harnessing big data through distributed sources can transform the working of organizations and initiate new upsurge of productive growth through proper structuring and analysis of data.

The shortcomings in the process still lie in capturing, storing, searching, transferring, analyzing, and visualization of data constraints. The distributed data streaming has to be structured to be used as analysis of unstructured data is challenging and may lead to erroneous results of the big data. The sub process of structuring the data in accordance with the need of the organization is difficult. The distributed data structure possesses the challenge of maintaining the integrity and legitimacy of the data. The distributed data structure also requires large monetary and structural resources to maintain the nodal system required for supporting the large-scale distributed data streaming.

VII. CONCLUSION

Integration of distributed data streaming within the paradigm of big data could open attractive opportunities for the government and private companies at large. The system imbalance of the traditional system restricts the probe of achievement from the big data. The traditional single source system is not sufficient to process the exponential data increase that the industries are witnessing with the integration of mobiles, smart devices, and wearable in the daily lives of the people. This large-scale data generated on a daily basis requires quick information processing tools. The query processing strategic frameworks of the big data such as Hadoop, Storm, and Flink required more than the traditional infrastructure. The query processing strategies established by the frameworks require nodal architecture of distributed data streaming to facilitate near to real-time data management and processing. The big data along with the distributed data streaming has the potential of changing the paradigm for big data usage across the industries. However, the distributed system also brings along challenges such as inconsistency arising from multiple data sources, incomplete data, timeliness, security, and scalability of the data from nodal sources. The two concepts if developed together have the characteristic of establishing a large system that is profitable in the data mining field. To enhance the integration of distributed data streaming with big data would require the companies to enhance the research front and increase their expenditure on infrastructure to support the multimodal structure and human resources in the field.

REFERENCES

[1] Clifford, S. and Hardy, Q. (2013) 'Attention, shoppers: Store is tracking your cell', New York Times. Available at: http://www.nytimes.com/2013/07/15/%0Abusiness/attention-shopper-stores-aretracking-your-cell.html.

[2] Dias De Assun Ao, M. et al. (2017) 'Distributed Data Stream Processing and Edge Computing: A Survey on Resource Elasticity and Future Directions', arXiv e-Print archive. Available at: https://arxiv.org/pdf/1709.01363.pdf (Accessed: 31 January 2018).

[3] Ellingwood, J. (2018) Hadoop, Storm, Samza, Spark, and Flink: Big Data Frameworks Compared | DigitalOcean, Digital Ocean LLC. Available at: https://www.digitalocean.com/community/tutorials/hadoop-storm-samza-spark-and-flink-big-data-frameworks-compared (Accessed: 31 January 2018).

[4] Fujitsu (2016) White paper Solution Approaches for Big Data. Germany. Available at: https://sp.ts.fujitsu.com/dmsp/publications/public/wp-bigdata-solution-approaches.pdf (Accessed: 31 January 2018).

[5] Fuqiang, Y. (2011) 'The Research on Distributed Data Stream Mining based on Mobile Agen', Procedia Engineering, 23, pp. 103–108. Available at: https://ac.els-cdn.com/S1877705811053161/1-s2.0-S1877705811053161-main.pdf?_tid=2aaacd9c-0662-11e8-9c98-00000aacb35e&acdnat=1517388113_e8809e3515438518b2cada7d641f7e6a (Accessed: 31 January 2018).

[6] Garofalakis, M. (2009) 'Distributed Data Streams', Encyclopedia of Database Systems, pp. 883–890. Available at: https://pdfs.semanticscholar.org/c068/c359a52ca0e360091c5bbf88302abce746c8.pdf (Accessed: 31 January 2018).

[7] Gruyter Oldenbourg, D. et al. (2016) 'Special Issue Real-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 02, February-2018

time stream processing for Big Data', it – Information Technology, 58(4), pp. 186–194. doi: 10.1515/itit-2016-0002.

[8] Ha, K. et al. (2014) 'Towards wearable cognitive assistance', 12th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '14, ACM, New York, USA, pp. 68–81. doi: 10.1145/2594368.2594383.

[9] Khan, N. et al. (2014) 'Big data: survey, technologies, opportunities, and challenges.', TheScientificWorldJournal. Hindawi, 2014, p. 712826. doi: 10.1155/2014/712826.

[10] Kreps, J. (2014) 'Questioning the Lambda Architecture', O"Reilly, pp. 1–10.

[11] Kune, R. et al. (2016) 'The Anatomy of Big Data Computing', Software: Practice and Experience, 46, pp. 79–105. doi: 10.1002/spe.2374.

[12] Lambda-architecture.net (2014) Lambda Architecture.

[13] Landset, S. et al. (2015) 'A survey of open source tools for machine learning with big data in the Hadoop ecosystem', J. Big Data, 2(1), p. 24.

[14] Mohamed, M. A., Nagi, M. H. and Ghanem, S. M. (2016) 'A clustering approach for anonymizing distributed data streams', in 2016 11th International Conference on Computer Engineering & Systems (ICCES). IEEE, pp. 9–16. doi: 10.1109/ICCES.2016.7821968.

[15] Ounacer, S. et al. (2017) 'A New Architecture for Real Time Data Stream Processing', IJACSA) International Journal of Advanced Computer Science and Applications, 8(11). Available at: www.ijacsa.thesai.org (Accessed: 31 January 2018).

[16] Salehi, A. (2010) 'Design and Implementation of an Efficient Data Stream Processing System', EPFL. Available at: https://infoscience.epfl.ch/record/142936/files/EPFL_TH4611.pdf (Accessed: 31 January 2018).