Performance Assay of Big IoT Data Analytics Framework

Sandeep Bhargava, Bright Keswani, Dinesh Goyal

Abstract: Evaluation of Internet of Things (IoT) technologies in real life has scaled the enumeration of data in huge volumes and that too with high velocity, and thus a new issue has come into picture that is of management & analytics of this BIG IOT STREAM data. In order to optimize the performance of the IoT Machines and services provided by the vendors, industry is giving high priority to analyze this big IoT Stream Data for surviving in the competitive global environment. Thses analysis are done through number of applications using various Data Analytics Framework, which require obtaining the valuable information intelligently from a large amount of real-time produced data. This paper, discusses the challenges and issues faced by distributed stream analytics frameworks at the data processing level and tries to recommend a possible a Scalable Framework to adapt with the volume and velocity of Big IoT Stream Data. Experiments focus on evaluating the performance of three Distributed Stream Analytics Here Analytics frameworks, namely Apache Spark, Splunk and Apache Storm are being evaluated over large steam IoT data on latency & throughput as parameters in respect to concurrency. The outcome of the paper is to find the best possible existing framework and recommend a possible scalable framework.

Keywords : *IoT Streaming Analytics, Big IoT Data Analytics, Big Data Analytics Framework , Big IoT Data Analytics Framework, Stream Analytics Comparison*

I. INTRODUCTION

The rate at which the use of IoT machine and the amount of real time information are increasing, it is becoming more difficult to access insights that are valuable from all of this data. It is because of this the sustainability of current data analytics solutions is a concern, for eg[1]

1. What will happen if the frequency of data sending and the number of devices were to increase by a factor of ten or even hundred and many more?
2. Will the ability to meet the deadline suffer as the number of devices increases or if the frequency of data collection changes?

There is a crucial need of a data analytics framework that can process stream data of millions of events in seconds with low latency and high throughput.

This paper aims to assess the performance of existing Data Analytics frameworks namely, Splunk, Apache Spark, Apace Storm, on similar parameters and environment, with the following contributions:

- Performance issues due to big IoT data to stream analytics
- Comparative performance evaluation of all 3 with various performance parameter
- Characterization of the impact of several experimental parameters on the overall performance.

A. Characteristics of IoT Data

IoT data *is generally different from big data. In order to have better understanding needs of IoT data analysis, we need to explore the properties of IoT data and their differences from general big data [2]. IoT data has the following characteristics:*

- **Large-scale streaming data:** The infinite nodes & devices to capture data are distributed and deployed demographically and distinct and distant places, for IoT applications, which continuously generate large amounts of data streams.
- **Heterogeneity:** Various IoT data collection devices collect different types of information, resulting in data heterogeneity.
- **Time and spatial correlation:** In most IoT applications, sensor devices are connected to specific locations, so each data item has a location and timestamp.
- **High noise data:** As the tuple of information per node is small in IoT applications, many of these data may be subject to errors and noise during acquisition and transmission.

These characteristics of IoT data imposes new challenges for analytics tools to process and analysis the data at massive scales (millions of sensors, thousands of events per second) and require large computational resources to accommodate the analytics needs.

B. Streaming Analytics

It enables organizations to analyze data as soon as it is available with the nodes, before being stored in the database. Stream analysis can help organizations to discover new business opportunities, high and fast customer connections, and revenue streams to increase profits. In a Streaming Analytics environment, data is processed before it reaches the database [3]

Manuscript published on November 30, 2019.
***** Correspondence Author

Sandeep Bhargava*, Research Scholar, Suresh Gyan Vihar University, City, Country.
Bright Keswani, Professor, Computer Application, Suresh Gyan Vihar University, Jaipur, India.
Dinesh Goyal, Professor, Poornima Instiute of Engineering & Technology, Jaipur, India,

belong to the engineering and technology area. In the paper title, there should not be word 'Overview/brief/ Introduction, Review, Case study/ Study, Survey, Approach, Comparative, Analysis, Comparative Investigation, Investigation'.

II. PERFORMANCE CHALLENGES OF STREAM ANALYTICS

Big data flow analysis is especially important when it comes to getting useful knowledge from current events so that organizations can react quickly to problems or find new trends that help improve their performance [4]. However, due to the nature of the big data streams, there are some challenges such as scalability, integration, fault tolerance, timeliness, consistency, heterogeneity, load balancing, privacy issues and accuracy.

1. **Scalability:** It is the ability of the system to adapt to the ever-increasing demands of data processing. To support big data processing, different platforms use different forms of file extension. One of the main challenges in big data stream analysis is the scalability issue. Big data streams are experiencing exponential growth in a much faster way than computing resources. Therefore, research is needed to develop scalable frameworks to accommodate data flow computing models, effective resource allocation strategies and parallelization issues to address the growing data scale. [4].

   a. **Scalability issue**

      i. Whenever there is point of scalability of any IoT analytics platforms there are certain constraints in the product itself like powerBi and Apache Splunk products support distributed horizontal scaling of resources under a single processing and analytics engine. For example if an analytics engine can process 10 thousand instructions per second, no matter whatever resources required by the engine to process data, that means after reaching the product limitation there will be no relevance of increasing hardware resource beyond the limitation

      ii. Whenever there is a single processing engine on top of distributed scalable environment, it will not increase latency and throughput of analytics engine even if we increase the hardware resources.

      iii. Even if there are free resources in the hardware and the analytics engine is not exceeded by the limit still if we increase the hardware resources it will not optimize the latency and throughput for better processing and analytics of stream data.

2. **Memory:** While processing real time data, memory consumption vary with respect to programming language that we choose to for Big data processing and analytics of different kernel platforms.

3. Concurrency: Huge IoT Data can be handled by increasing the number of concurrent consuming resources, which will be a problem for single analytics engine because a single analytics engine needs more CPU threads and cores to accept the data concurrently.

III. EXPERIMENT SETUP

A. In response to the growing demand for big data stream analysis, open source communities and enterprise technology vendors have developed many alternative big data flow solutions. The top 3 big data stream analytics tools, which are being tested in this paper are:

   a. **Apache Spark**

   b. **Splunk**

   c. **Apache Storm**

B. To compare the performance of above 3, intended Distributed Stream Analytics Framework (DSAF) we measure end-to-end latency *(Latency is an important measure for IoT analytics to evaluate the "real-time" nature of processing, as with other streaming applications. Lower the latency, faster the ability to process and make fast decisions)* for each tuple in variation with concurrency, and calculate the average latency [5] in desired time slices. We generate the latency results obtained by Storm, Splunk and Spark Streaming respectively; using 2, 5 and 8 worker nodes. Two parameters which are being analyzed are Latency and Concurrency

C. **Assumption:**

- Concurrency will vary from 10 packets to 40 packets
- Latency monitoring time is varying from 20 Sec to 5 Minute.

D. **Machine Configuration:**

- Processor: Intel Xeon CPU E5-2670 running at 2.40GHz
- CPU has 4 cores, each core has 2 threads (hyper-threading).
- 8GB memory.
- Operating System RHEL 7.5
- Data Size: Live streamed data
- Data source: Spark.org, Splunk.com, Hadoop.com, Twitter.com

IV. PERFORMANCE EVALUATION

A. Latency Evaluation

To compare the performance of intended DSAFs in terms of latency, we measure end-to-end latency for each tuple, and calculate the average latency in desired time slices. Stream Data used for latency analysis is obtained from various logs of Streaming Data, for a particular input rate (1 k tuple/s) we have performed experiments on all three DSAFs by different number of worker nodes. In Figs. 1 and 9 we can see the latency results obtained by Storm, Splunk and Spark Streaming respectively; using 2 and 8 worker nodes.

[i] Latency calculation, on Storm Framework with variable note size

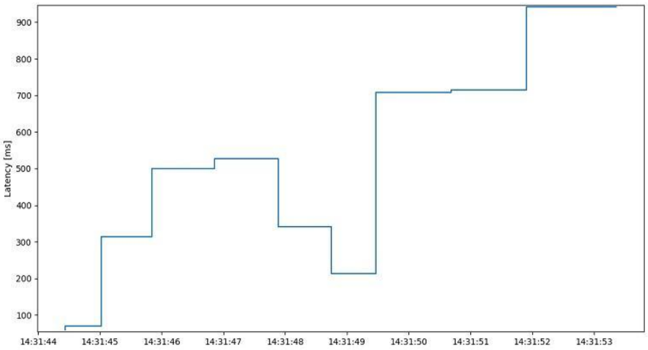


Fig. 1: Latency on Storm Framework with 2-Node

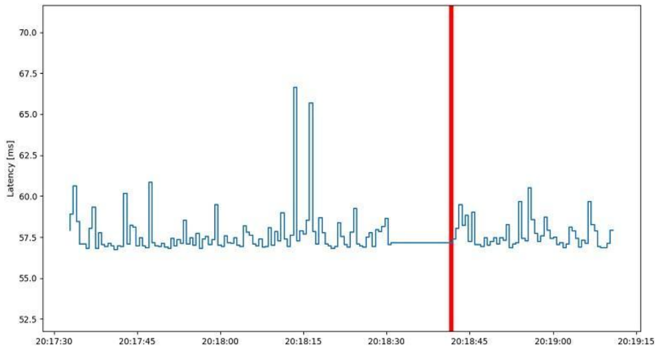


Fig. 2: Latency on Storm Framework with 5-Node

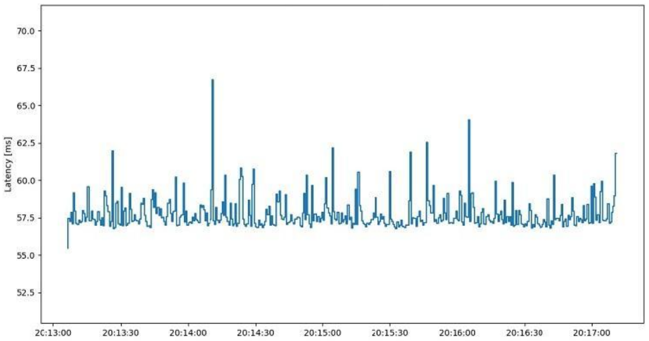
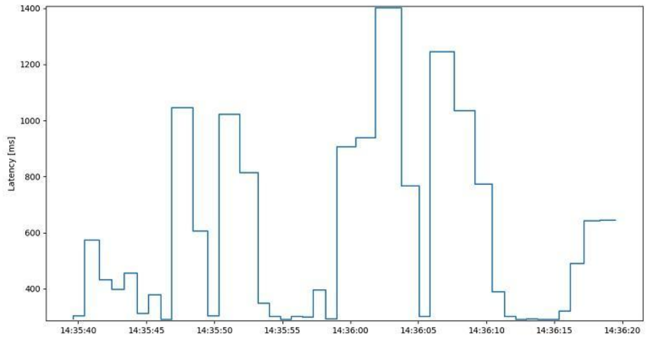


Fig. 3: Latency on Storm Framework with 8-Node

[ii] Latency calculation, on Splunk Framework with Variable note size



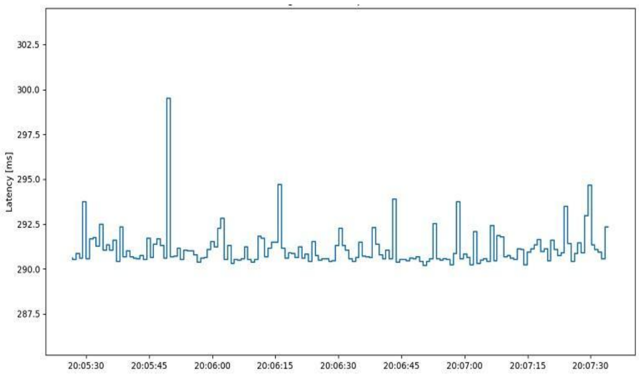


Fig. 5: Latency on Splunk Framework with 5-Node

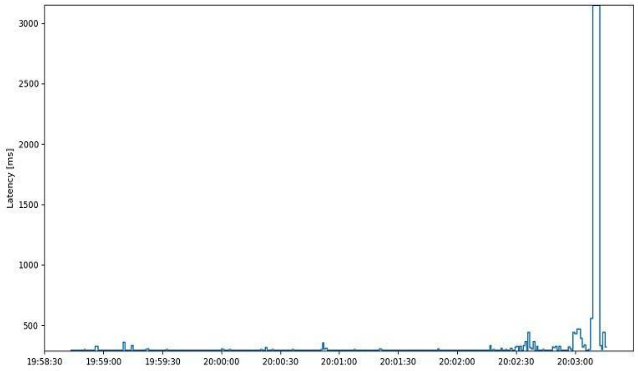


Fig. 6: Latency on Splunk Framework with 8-Node

[iii] Latency calculation, on Spark Framework with Variable note size

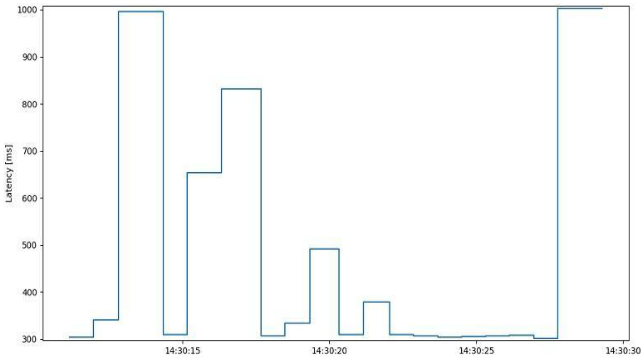


Fig. 7: Latency on Spark Framework with 2-Node

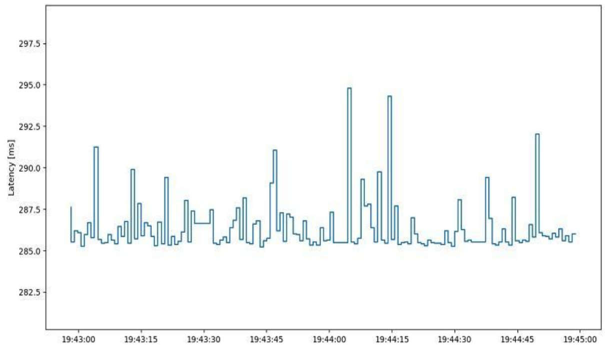


Fig. 8: Latency on Spark Framework with 5-Node

Retrieval Number: D7383118419/2019©BEIESP
DOI:10.35940/ijrte.D7383.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

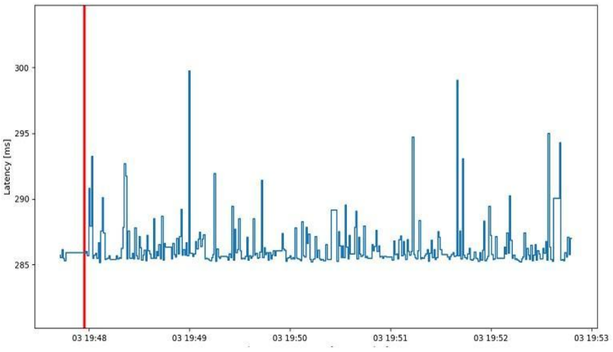


Fig. 9: Latency on Spark Framework with 8-Node

Looking at these graphs, we can see that Storm has the slowest latency, but when you increase the number of working nodes, Storm has some improvements. In large clusters, Spark Streaming outperforms Splunk in latency. In terms of scalability, both Spark and Splunk Streaming have behaviors that are nearly linear than Storm.

B. Load Scalability

To assess the ability for Apache Storm, Splunk, Apache Spark. framework to efficiently expand and contract its resources inorder to accommodate high speed streaming data, we increase the concurrency by sending random number of large concurrent stream data ranging from 1000 to 2200 from various IoT sources and monitor the latency behavior of the DSAF's over Load Scalability (Concurrency)

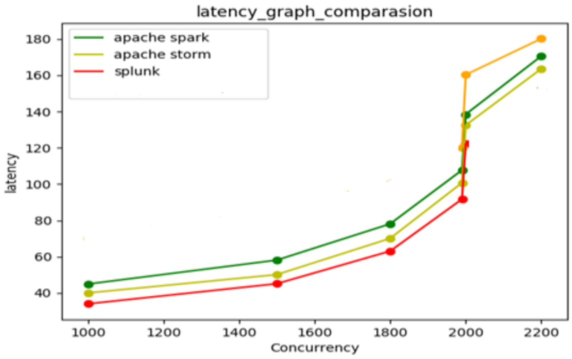


Fig. 10: Latency analysis of 3 DSAF's over Load Scalability

Latency of splunk is linear to the other two DSAF's when the load is less as the concurrent load is increased, Splunk outperforms the other DSAF's namely Storm and Spark.

V. CONCLUSION

By performing the above experiments over each of the frameworks studied here with their challenges and issues, we can conclude that Spark Streaming has much low latency and it provides higher throughput compared to other analytics framework. Though it can be concluded that Spark Streaming outperforms Splunk and Storm in terms of latency. While Splunk performs slightly better than other in term of load scalability In order to address the issues of optimizing latency and concurrency we will propose a new Portable Auto Scaling Big Data Analytics Framework; there performance can be tuned in order to minimize processing time and less resource consumption with zero down time to meet the growing need of data processing and analytics of large volume data set.

REFERENCES

1. Prateep Misra, 2016, "Build a Scalable Platform for High-Performance IoT Applications" https://www.tcs.com/content/dam/tcs/pdf/discover-tcs/Research-and-Innovation/Build_a_Scalable_Platform_pdf.pdf
2. M. Chen, S. Mao, Y. Zhang, and V. C. Leung, 2014, Big data: related technologies, challenges and future prospects. Springer, [https://arxiv.org/pdf/1712.04301.pdf]
3. Harrine Freeman, 2016 "Streaming Analytics 101: The What, Why, and How" https://www.dataversity.net/streaming-analytics-101/
4. Taiwo Kolajo et. al, 2019, "Big data stream analysis: a systematic literature review" Journal of Big Data, Springeropen
5. https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-streaming-unit-conumption
6. Nasiri, Hamid et. al, 2019, "Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities", Journal of Big Data, SpringerOpen,Volume 6, Issue 1, https://doi.org/10.1186/s40537-019-0215-2
7. S Bhargava et.al,2019, "Performance Comparison of Big Data Analytics Platforms", International Journal of Engineering, Applied and Management Sciences Paradigms (IJEAM), ISSN 2320-6608, Volume 54 Issue 2 May 2019.
8. Acharjya D. P. et. al, 2016," A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 2, https://thesai.org/Downloads/Volume7No2/Paper_67-A_Survey_on_Big_Data_Analytics_Challenges.pdf
9. Agarwal S. 2016, "state of fast data and streaming applications survey". https://www.opsclarity.com/2016-state-fast-data-streaming-applications-survey/.
10. 2018 "Apache spark: Lightning-fast unified analytics engine". https://spark.apache.org/. Accessed 02 June 2018.
11. S Bhargava et.al,2018, "Big IoT Data Analytics: Literature Review, Opportunity and its Research Challenges, Journal of Emerging Technologies and Innovative Research, ISSN-2349-5162, Volume 5 Issue 11 November 2018.
12. S Kamburugamuve, G Fox. 2018"Survey of distributed stream processing. Bloomington: Indiana University

AUTHORS PROFILE

Sandeep Bhargava is a Ph.D research scholar at Suresh Gyan Vihar University, Jaipur. . His research areas is Big Data Analytics & IoT.

Dr. Bright Keswani is working a professor at Suresh Gyan Vihar University, Jaipur. Dr. Keshwani is acquired experience of 20 years in Teaching, Research and Administration. His research areas are Data Analytics, Big data & Cloud computing.

Dr. Dinesh Goyal is working as Professor at Poornima Institute of Engineering & Technology. Dr. Goyal acquired experience of 19 years in Teaching, Research and Administration. His research areas are Data Analytics, Security & Cloud computing. He has published more than 100 research papers in international publications, out which 12 are scopus indexed &amp; 1 SCI Indexed along with3 patents. He has edited 3 research books.. He has supervised eight doctoral students. He is a fellow member of CSI, IEEE, ISC, ISTE.

Retrieval Number: D7383118419/2019©BEIESP
DOI:10.35940/ijrte.D7383.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication