Literature Review : A Comparative Study of Real Time Streaming Technologies and Apache Kafka

Shubham Vyas
PhD Research scholar
Amity School of Engineering and Technology
Gurgaon, Haryana, India
r.shubhamvyas@gmail.com

Dr. Rajesh Kumar Tyagi
Professor, Department of Computer Science and Engineering
Amity School of Engineering and Technology
Gurgaon, Haryana, India
rktyagi@ggn.amity.edu

Dr. Charu Jain
Asst. Professor , Department of Computer Science and Engineering
Amity School of Engineering and Technology
Gurgaon, Haryana, India
cjain@ggn.amity.edu

Dr Shashank Sahu
Asst. Professor, Department of Computer Science and Engineering
Ajay Kumar Garg Engineering College
Ghaziabad, Uttar Pradesh, India
sahushank25@gmail.com

***Abstract*— For data ware housing projects, there are multiple licensed ETL (Extraction Transformation & Load) tools available in the market. To process and pass the data between two applications industry is using ETL tools like IBM Info-sphere Data Stage, Informatica, Ab Initio etc. These tools are exceptionally costly and has recurring enterprise licensees, and processed data is not available in real time, data is getting processed in batches and is available during pre-defined time intervals or on demand. Industry has started adopting the Open Source technologies to avoid the huge licensing cost and that also includes the complete end to end IT infrastructure cost. Open Source technologies and frameworks enables users to run projects with best in class performance and within the budget.**

In this literature survey paper, all possible technologies have been studied and evaluated, available in the market capable of real/" near-real-time" streaming. All licensed and open source products which are utilized and evaluated by various IT organizations and which are also evaluated by researchers have been included in this survey. There is a need of a distributed scalable technology that enables the users to ensure availability of data from one end point to another in real time with good throughput, performance and low latency. To study this, a detailed comparative survey of an open source technology Apache Kafka has been done and it compared with the other available technologies capable of doing real time streaming.

Keywords*— Data Base, Real Time Streaming, Apache Kafka, Apache Spark, Throughput, Latency, ETL (Extraction, Transformation & Load)

I. Introduction

Availability of business insights in real time has become very important to run businesses when critical and complex decisions are expected to be taken based on available data from analytics systems. If availability of analytics data is delayed even for few milliseconds, then sometimes the analytical insight becomes useless because of the delay occurred in the process. There are many applications, there success depends on the real time data for example, social media sites, data from IoT sensors and stock markets stats, fraud detection systems etc. Sometimes there is a need to analyse the incoming data even before it is getting loaded into the data base systems. Real time streaming frameworks and technologies enable users to make business decisions in real time instead of waiting for longer time. There is a need of powerful tools and technologies that can enable us to do the real time streaming, analytics and that can also process huge amount of data [5]. To meet and exceed the customer requirements real time data streaming technologies are getting developed and evolved. Apache Kafka was developed by Linked in and they have released its first version 0.6 in 2011. In 2012 this technology was graduated and adopted by Apache. Following is the time line that explains the evolution of this technology in terms of years.

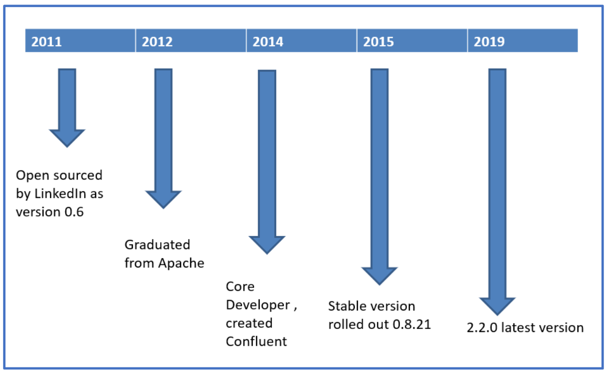
2011
2012
2014
2015
2019
Open sourced by LinkedIn as version 0.6
Graduated from Apache
Core Developer , created Confluent
Stable version rolled out 0.8.21
2.2.0 latest version

Fig 1

Evolution of Apache Kafka

(Captured from kafka.apache.org, dated March 22, 2021)

II. Review Methodology

The review methodology followed as reviewing major studies done so far for the various real time data streaming technologies like Apache Storm, Apache Flume, Apache Spark and various MQ (Messaging queue) technologies developed by IBM. A variety of infrastructures that enables us to do real time data streaming e.g. Hadoop cluster, IoT devices, social media websites etc have been included. The review methodology is broadly divided into three steps:

- In the first step relevant papers and material have been collected and from this material facts and contributions by researchers have been identified.

978-1-6654-2392-2/21/$31.00 ©2021 IEEE
DOI 10.1109/CICT53244.2021.00038

- Analysis is the second step where parameters have been identified based on which this survey will be conducted.
- The third step is the review writing.

The review methodology is designed by keeping in mind the review questions that form the basis for conducting this literature review. The review questions are:

1. Real time data streaming technologies reviewed in this survey supports modern frameworks, technologies and programming languages like Python, Java, Microservices etc?
2. What is the benefit of using the technology in terms of throughout, latency, scalability, and performance?
3. What are the different areas that are still open for research and further improvement?
4. What are the different datasets available and what are the sources of data collection?
5. What are the results if evaluation of Apache Kafka in comparison of all other technologies reviewed in this paper?

Following parameters has been considered to evaluate the real time stream technologies and frameworks:

- Latency: It's the amount of time required by a real time streaming framework to complete the processing of one message from producer to consumer and generate the final output.
- Throughout: Number of messages per second a real time streaming service can process.
- Fault Tolerance: In case system or infrastructure failures, streaming service can continue the processing without impacting the user expectation and able to recover itself.
- Scalability: If there is unprecedented growth in input stream of data by producer then system should be capable off to add more Infrastructure (Storage/CPU etc) with small efforts and configuration changes to meet the user requirements.

III. CRUX OF THE PAPERS REVIEWED

IEEE Transaction with title "Evaluation of Stream Processing Frameworks "published in 2020 author Giselle van Dongen evaluated real time streaming technologies with experiments based on data burst at the start up and periodic un precedented data inputs at frequent time intervals. This includes Apache Flink, Apache Kafka, Apache Spark [22]. Researchers has evaluated the throughput and latency based on the variety of loads. Researchers has proved that a problem exists when we face data loss in case of heavy loads but did not explained that how to manage the rejection of data systematically.

Paper published by Han Vu et al. with tile "A Reactive Batching Strategy of Apache Kafka for Reliable Stream Processing in Real Time" in 2020 evaluated the impacts of batch size on throughput and latency in Kafka processing [1]. This paper explains the Apache Kafka load processing and its configurations very well. As a research gap more, experiments on real world scenarios for this method can be explored as all experiments in this paper were done on some mocked-up data.

Paper published by Bhole Rahul Hiraman et al. in 2018 with the title "A Study of Apache Kafka in Big Data Stream Processing" has provided very basic overview of Apache Kafka with one example of cryptocurrency including experimental results. This paper has taken Big Data for real time streaming but did not explored the impacts of changing data volume, need to explore the impact of changes in configuration parameters while using Apache Kafka [2].

Research paper proposed in 2016 by Van-Dai Ta et al. with title "Big Data Stream Computing in Health care Real-Time Analytics" proposed a new big data architecture for health care systems analytics by using open source technologies Hadoop, Apache Storm, Kafka and NoSQL Cassandra. Importance of Bigdata & Kafka also explained for healthcare systems [3]. This paper is not touching the scalability management for big data systems which is one of the important requirements when we are dealing with huge amount of data.

Research Paper published by Jiwon Bang et al. in 2018 explained the scenario when data load is changing suddenly during stream processing and how the streaming process is impacted by this. A random load shedding engine has been proposed by researchers in details [4]. As a research gap Semantic load shedding is not explained where random load shading itself a problem that is presented as a solution.

Research paper published by Vikash, Lalita Mishra et al. in 2019 with the title "Evaluation of Real-Time Stream Processing for Internet of Things Applications" has provided a use case and architecture level details of using Kafka with IoT devices [5]. In this specific use case, researchers may further explore and evaluate if proposed system is fault tolerant or not. For example, in case if broker is down or there are other system or infrastructure related failures, how proposed system will react.

Paper published by Rishika Shree et al. in 2017 titled "KAFKA: The Modern Platform for Data Management and Analysis in Big Data Domain" has provided a comparative study of Apache Flume and Apache Kafka and how to develop application using capabilities of both the technologies together [6]. There is a research gap where Open source technologies and controlled by various configuration parameters, such details are missing in this implementation.

Paper published by Han Wu et al. in 2020 with title "Learning to Reliably Deliver Streaming Data with Apache Kafka" presented a weighted key performance indicator for Kafka users to come up with accurate values of configuration parameters and explained how to write reliable application using Apache Kafka on various configurations and network conditions [7]. Here scalability of the producer is explained in detail, but the scalability of consumer is also critical for the overall system performance which is missing in the paper. Researchers has published another paper with title "Performance Prediction for the Apache Kafka Messaging System". In this paper researchers have explained the model to predict the configuration parameters [8]. The details and evaluation of the replication factor which is a key parameter for Kafka processing is missing in the experiments performed.

Research Paper published by Anveshrithaa S et al. in 2020 with title “Real-Time Vehicle Traffic Analysis using Long Short-Term Memory Networks in Apache Spark”, here researchers have proposed a real time traffic analysis application using Hadoop, Spark and Kafka [9]. This paper has used Apache Spark Streams as primary technology for achieving the goals, Kafka is used only for storage purposes.

Research Paper published by Ameer B. A. Alaasam et al. in 2020 explained a case study for manufacturing data analysis using Apache Kafka and Micro services [10]. This paper has given a new perspective for the code management using micro services. Fault tolerance, Scalability and applicability of this model can be explored further.

Paper published by Haidong Lv et al. in year 2020 with title “The Development of Real-time Large Data Processing Platform” has proposed a method for real time data streaming for big data systems using combination on Apache Spark and Apache Kafka [11]. As a research gap we have identified that the performance evaluation for the proposed code with respect to the volume can be analyzed further.

Paper published by Han Wu et al. in 2019 with title “TRAK-A Testing Tool for Studying the Reliability of Data Delivery in Apache Kafka” proposed a tool that ensures the delivery of data in case of network failures or other infrastructure issues [12]. Data volume is not considered in this evaluation and starvation problem is also not discovered in terms of the data overload at consumer side.

Research paper published by HaiTao Mei et al. with title “A Java-Based Real-Time Reactive Stream” has proposed an application known as reactive streams using java 9 stream processing framework and did a comparative analysis with regular Java frameworks like Storm, Samza, Heron, StreamFlex & Mattheis [13].

Research paper published by Abderrahmane et al. in 2018 with the title “Application of machine learning model on streaming health data event in real-time to predict health status using Spark.” In this paper researchers have shared the knowledge about the decision tree-based model this also based on machine learning techniques to predict statues of health in real time [14]. Here solution is very complex to implement, same can be implemented using Apache Kafka with 10% to 50% performance improvements.

Paper published by Gautam Pal et al. in 2018 with title “Big Data Real Time Ingestion and Machine Learning” explores the techniques to integrate machine learning with real time data streaming processing. This paper explained the process to analyse stream data based on machine learning, linear regression and K-Means clustering [15]. Here plugging inn, a new consumer or target system with this type of implementation will require a lot of efforts and huge changes at application level.

Paper published by Erum Mehmood et al. in 2020 with the title “Challenges and Solutions for Processing Real-Time Big Data Stream: A Systematic Literature Review” explained the challenges implementation of data ware houses in real time and streaming in big data systems and have found that more solutions are available for structured data but there are very less algorithm’s available for non-structured data [16]. In this paper Challenges for heterogeneous data sources are not covered.

Research Paper published by Thandar Aung et al. in 2020 with the title “Coordinate Checkpoint Mechanism on Real-Time Messaging System in Kafka Pipeline Architecture” proposed a solution to make Apache Kafka a Fault Tolerant system by using a fixed check point to overcome the problem of reduction in the lost messages and time system takes to recover in occasion of failures at server side [17]. There is one gap that we have identified is Latency and throughput which are key performance indicators for the real time processing are not evaluated in this implementation.

Paper published by Buqing Shu et al. in 2017 explains that how performance gains can be achieved using Apache flume by reducing the overhead of disk I/O and network. This paper has explained that the performance can be improved by 10% to 50% in various data and configurations [18]. Here load balancing has been implemented based on network nodes, which can be further improved by including the data volume measurements with the network nodes.

Research paper published by Gautam Pal et al. in 2018 proposed a method based on Cassandra to implement streaming system when patterns are kept on changing and evaluated the scenarios on low latency and high latency. A comparative analysis of throughput is also done for Kafka, Spark Stream and Cassandra [19]. There is a gap identified in this paper where data lose is not evaluated in any of the proposed scenarios to make sure data is successfully delivered from source to the target.

Research paper published by D.Surekha et al. in 2016 with title ‘Real Time Streaming Data Storage and Processing using Storm and Analytics with Hive” have proposed a method to implement the streaming using Storm and Apache Hive. A comparative analysis with Kafka and Storm is also captured [20]. We came across with a gap where data extraction from Hive is difficult and time consuming as Hive does not support the indexes and data retrieval always takes a lot of time compare to other data storage/data base solutions.

Research paper proposed by Alramzana et al. in 2019 with title “Real-Time Data Streaming Algorithms and Processing Technologies: A Survey” explained a detailed survey on six streaming data algorithms on a variety of real time and artificial data sets that includes data sets which are structured and non-structured [21]. Data Quality and Data lose is not evaluated during the survey.

The literature review table summarizes some of the most recent studies done on real time streaming technologies and captures the contribution from the authors and details of the gaps for future improvements.

Table 1: Literature Review Table

Author, Year, Publisher	Title of Paper	Contribution	Gaps for further Improvements
Han Vu Zhihao Shang Guang Peng Katinka wolter 2020, IEEE	A Reactive Batching Strategy of Apache Kafka for Reliable Stream Processing in Real Time	This paper has evaluated the impacts of batch size on throughput and latency in Kafka processing.	• This paper has taken some sample up data to explain the problem and the proposed solution. • More experiments on real world scenarios with this method can be explored.
Bhole Rahul Hiraman Et al. 2018, IEEE	A Study of Apache Kafka in Big Data Stream Processing	This paper has provided very basic overview of Apache Kafka with one simple example including experimental results.	• Introductory details for Apache Kafka have been provided. • Impact of changes in configuration parameters can be explored when using Apache Kafka for real time streaming.
Van-Dai Ta, Et al. 2016, IEEE	Big Data Stream Computing in Healthcare Real-Time Analytics	Researchers have proposed a method to analyze healthcare data based on big data technologies.	• This paper is not touching the scalability management for big data systems which is one of the important requirements when dealing with Big Data. • Fault tolerance of the proposed system also need to be explored and evaluated.
Jiwon Bang, Siwoon Son, Hajin Kim, Yang-Sae Moon and Mi-Jung Choi 2018, IEEE	Design and Implementation of a Load Shedding Engine for solving Starvation Problems in Apache Kafka	This paper explained the problem of data starvation and its solution using of random load shading.	• Semantic load shedding is not explained where random load shading itself a problem that is presented as a solution. • Need to explore further how system will react when unprecedented stream of data is ingested.
Vikash, Lalita Mishra, Shirshu Varma 2019 IEEE	Evaluation of Real-Time Stream Processing for Internet of Things Applications	This paper has provided a use case and architectural details of using Apache Kafka with IoT devices.	• Fault tolerance can be further evaluated. • if broker is down or if there are infrastructure related issues, how proposed system will react.
Rishika Shree, et al. 2017 IEEE	KAFKA: The Modern Platform for Data Management and Analysis in Big Data Domain	Researchers has given a comparative study of Apache Flume and Apache Kafka and how to develop application using capabilities of both the technologies' together.	• Open source technologies and controlled by various configuration parameters, such details are missing in this implementation. • Value of Replication factor which is used is not provided for Apache Kafka.
Han Wu Et al.	Learning to Reliably Deliver Streaming Data with Apache Kafka	Here researchers have explored details on predefined reliability metrices and explained how to write reliable application using Apache Kafka on various configurations and network conditions.	• Here scalability of the producer is explained in detail, but the scalability of consumer is also critical for the overall system performance which is missing in the paper. • Retry mechanism can be further explored.
Han Wu, Zhihao Shang, Katinka Wolter 2019 IEEE	Performance Prediction for the Apache Kafka messaging System	In this paper researchers have explained the model to predict the configuration parameters	• Details and evaluation of the replication factor which is a key parameter for Kafka processing is missing in experiments. • More tests need to be conducted to prove that predictions are accurate

			when actually processing the data based on predefined parameters.
Anveshrithaa S Lavanya K 2020 IEEE	Real-Time Vehicle Traffic Analysis using Long Short-Term Memory Networks in Apache Spark	In this paper researchers have proposed a real time traffic analysis application using Hadoop, Spark and Kafka.	• This paper has used Apache Spark Streams as primary technology for achieving the goals of stream processing, Kafka is used only for storage purposes. • More tests can execute to validate the performance of Apache Kafka on Hadoop cluster.
Ameer B. A. Alaasam Gleb Radchenko Andrey Tchernykh 2020 IEEE	Stateful Stream Processing for Digital Twins: Microservice-Based Kafka Stream DSL	In this paper researchers have explained a case study for manufacturing data analysis used Apache Kafka and Microservices.	• Fault tolerance, Scalability and applicability of this model need to explore further. • Microservices can be tested further to see if its working as expected when data need to be processed sequentially.
Haidong Lv ,et al. 2020 IEEE	The Development of Real-time Large Data Processing Platform Based on Reactive Micro- Service Architecture	In this paper researchers have given a real time streaming solution for big data systems using combination on Apache Spark and Apache Kafka.	• Performance evaluation for the proposed code with respect to the volume can be analysed further. • Impacts of replication factor with periodic burst can be tested further.
Han Wu, et al. 2020, IEEE	TRAK- A Testing Tool for Studying the Reliability of Data Delivery in Apache Kafka	In this paper researchers have developed a tool that ensures the delivery of data in case of network failures or other infrastructure issues.	• In this paper starvation problem is not discovered in terms of the data overload at consumer side. • Unprecedented volume at the start of stream processing and at frequent intervals can be tested further.
HaiTao Mei, et al. 2016 IEEE	A Java-Based Real-Time Reactive Stream	In this paper researchers have built an application using java 8 stream processing framework and did a comparative analysis with regular Java.	• Fault tolerance cannot be guaranteed using the method that exists in Apache Kafka. Apache Kafka provides the configurations to build a fault tolerant system. • Microservices architecture can be used for the implementation which is more suitable.
Abderrahmane Ed-daoudy Khalil Maalmi 2018 IEEE	Application of machine learning model on streaming health data event in real-time to predict health status using Spark.	In this paper researchers have shared the knowledge about the decision tree-based model that works using machine learning technologies to predict the health status in real time.	• Solution is very complex to implement, same can be implemented using Apache Kafka with 10% to 50% improvements in performance. • As it's a bug data solution, Integration of Hive can be explored further to store the results and future usages.
Gautam Pal Et al. 2018 IEEE	Big Data Real Time Ingestion and Machine Learning	This paper explained the process to analyse stream data based on machine learning, K-Means clustering and linear regression.	• Plugging inn, a new consumer or target system with this type of implementation will require a lot of efforts and huge changes at application level. • Configuration parameters can be provided to add multiple consumers as per the requirements.
Erum Mehmood and Tayyaba Anees 2020 IEEE	Challenges and Solutions for Processing Real-Time Big Data Stream: A Systematic Literature Review	In this paper researchers has explained the challenges in populating data into data warehouses in real-time and doing streaming for big data systems and have found that more solutions are	• Challenges for heterogeneous data sources are not covered and there is more focus on ETL (Extraction, transformation, and load) based applications. • Un structured data can also be included for testing this solution.

		available for structured data but there are very less algorithm's available for non-structured data.	
Thandar Aung, et al. 2020 IEEE	Coordinate Checkpoint Mechanism on Real-Time Messaging System in Kafka Pipeline Architecture	Researchers have proposed a solution to make Apache Kafka a Fault Tolerant system by fixing a check point to lower the intensity of the lost messages and time it takes to recover in occasion of failures at server side.	• Fixed check point interval is considered which is not configurable, variable checkpoint interval can be further explored. • Latency and throughput which are key performance indicators for the real time processing are not evaluated in this implementation.
Buqing Shu, et al. 2017 IEEE	Dynamic Load Balancing and Channel Strategy for Apache Flume Collecting Real-time Data Stream	In this paper researchers have given a method that improves the performance of systems doing real time streaming using Apache flume by reducing the overhead of disk I/O and network. After various tests researchers were able to see the performance gains between 10 to 50%	• Load balancing has been implemented based on network nodes, load balancing can be more effective and accurate by considering the data volume measurement's with node configuration. • More details can be provided about the configuration parameters and environment when the performance gain is 50%.
Gautam Pal Et al. 2018 IEEE	Near Real-Time Big Data Stream Processing Platform Using Cassandra	Here researchers have provided a method based on Cassandra to implement streaming system when patterns are kept on changing and evaluated the scenarios on low latency and high latency. A comparative analysis of throughput is also done for Kafka, Spark Stream and Cassandra.	• Data lose is not evaluated in any of the proposed scenarios to make sure data is successfully delivered to the target. • This work can be evolved further by implementing the Lambda architecture which is a hybrid solution of real time streaming and batch processing.
D.Surekha, et al. 2016 IEEE	Real Time Streaming Data Storage and Processing using Storm and Analytics with Hive	In this paper a method has been proposed to implement the streaming using Storm and Apache Hive. A comparative analysis with Kafka and Storm is also captured.	• Data extraction from Hive is difficult and time consuming as Hive does not support the indexes and data retrieval always takes a lot of time compare to other data storage/data base solutions. • Instead of Hive other Data Storage solutions like MongoDB can be used.
Alramzana Nujum Navaz Saad Harous Mohamed Adel Serhani Ikbal Taleb 2019 IEEE	Real-Time Data Streaming Algorithms and Processing Technologies: A Survey	This is a survey on 6 streaming data algorithms on a variety of real time and artificial data sets.	• Data Quality and Data lose is not evaluated during the survey. • This paper has provided a basic overview and details of performance parameters for all 6 mentioned algorithms.
Giselle van et al. 2020 IEEE	Evaluation of Stream Processing Frameworks	Researchers has evaluated many real time streaming technologies with many experiments and presented their results. This includes Apache Flink, Apache Kafka, Apache Spark. Researchers has evaluated the throughput and latency based on the variety of loads.	• This comparison can be further done with joining the static datasets. We can also involve other frameworks like Apax,Beam and Storm. • There are two other major aspects of fault tolerance and scalability which are not studied here. For example, replication factor can be considered while studying the Apache Kafka which is missing in this transaction.

IV. EXTENDED DISUCSSION

After reviewing the literature Apache Kafka has been identified as a best suitable framework and technology to implement real time data streaming which is cost effective and scalable as per requirements. When we compare Apache Kafka with other real time streaming technologies on different performance and quality parameters like throughput, latency and data integrity, Kafka has higher throughput and Kafka was developed for distributed systems so its easily scalable to many clusters as per the requirements. There is one more major difference the way Kafka treats the messages compare to other streaming platforms, its stores the messages into files instead of keeping them into memory so it performs better as memory is not utilized for actual messages and it can be utilized for providing a good performance to the system [4]. So, in many cases Apache Kafka has been identified a very suitable framework for real time streaming of data in terms of performance and scalability.

Apache Kafka is developed for distributed systems which are highly scalable, and it works based on publish-subscribe method. It can process high volume of data and data can be passed from one source (publisher) to many targets (subscribers). As a streaming service it has three main capabilities:

1. Provide publish and subscribe capability to stream of data like a messaging queue.
2. It can store the data stream in fault-tolerant way
3. Publish the records to the target system as they arrive.

Following are the basic terminologies we should understand when are using Apache Kafka:

Producer: An application that sends messages to Kafka.
Consumer: Service that can read data from producer
Broker: Broker is a Kafka Server.
Topic: A category of messages belongs to a topic; it works as a data storage in Kafka processing.
Partitions: As Kafka is a distributed system so data or messages can be scattered among multiple partitions. This id is assigned as the message arrives in partition.
Consumer groups: A group of consumers acting as a single logical unit.

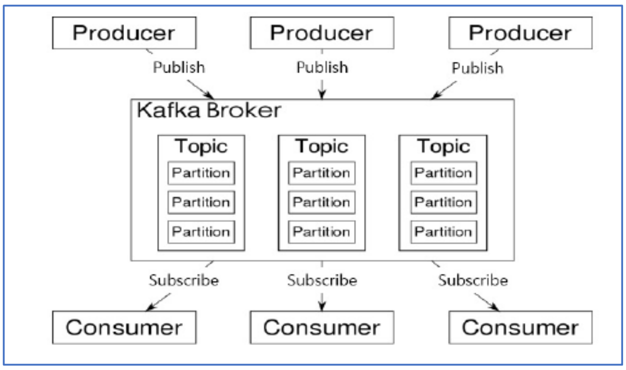

Fig. 2
Kafka Architecture [4]

. V.CONCLUSION & FUTURE SCOPE

Aim of software engineering is to develop software's which meets the customer requirements and can be built within the expected budget. Emergence of open source technologies made all this possible where there are technologies available which are free to use and providing best in class performance. To stream data in real time on distributed environments this statement is very true. After this survey, evaluation and comparative study, Apache Kafka is proved as one of the best candidates when it comes to get best in class performance within budget to stream data in real time.

Apache Kafka supports all programming languages and modern coding frameworks and practices like Microservices Architecture. As a future scope and work researchers can further explore on how Apache Kafka can be utilized to transform and evolve traditional batch applications into Real Time stream applications. A comparative study can be performed between traditional ETL batch applications and Apache Kafka to see if Apache Kafka is more suitable to traditional batch applications in terms of budget and performance.

REFERENCES

[1] Han Vu, Zhihao Shang,Guang Peng ,Katinka wolter ,A Reactive Batching Strategy of Apache Kafka for Reliable Stream Processing in Real Time,2020 IEEE 31st International symposium on Software Reliability Engineering (ISSRE)

[2] Bhole Rahul Hiraman,Chapte Viresh M.,Karve Abhijeet C., A Study of Apache Kafka in Big Data Stream Processing,2018 International Conference on Information, Communication, Engineering and Technology (ICICET)

[3] Van-Dai Ta,Chuan-Ming Liu ,Goodwill Wandile ,Nkabinde,Big Data Stream Computing in Healthcare Real-Time Analytics,2016 IEEE International Conference on Cloud Computing and Big Data Analysis

[4] Jiwon Bang, Siwoon Son, Hajin Kim, Yang-Sae Moon and Mi-Jung Choi, Design and Implementation of a Load Shedding Engine for Solving Starvation Problems in Apache Kafka, No. 2016-0-00179, Development of an Intelligent Sampling and Filtering Techniques for Purifying Data Streams, IEEE 2018

[5] Vikash, Lalita Mishra2 et al.,"Evaluation of Real-Time Stream Processing for Internet of Things Applications", 978-1-7281-3455-0/19@IEEE

[6] Rishika Shree, Tanupriya Choudhury, Subhash Chand Gupta, Praveen Kumar, KAFKA: The Modern Platform for Data Management and Analysis in Big Data Domain, 2017 2nd International Conference on Telecommunication and Networks (TEL-NET 2017)

[7] Han Wu,Zhihao Shang,Katnka Wolter, Learning to Reliably Deliver Streaming Data with Apache Kafka, 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)

[8] Han Wu, Zhihao Shang, Katinka Wolter, Performance Prediction for the Apache Kafka Messaging System, 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems

[9] Anveshrithaa S, Lavanya K, Real-Time Vehicle Traffic Analysis using Long Short-Term Memory Networks in Apache Spark, 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)

[10] Ameer B. A. Alaasam, Gleb Radchenko, Andrey Tchernykh, Stateful Stream Processing for Digital Twins- Microservice-Based Kafka Stream

DSL, 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)

[11] Haidong Lv, Tianzhi Zhang,Zhenhan Zhao ,Jiahui Xu1,Tianyu He, The Development of Real-time Large Data Processing Platform Based On Reactive Micro-Service Architecture, 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC 2020)

[12] Han Wu, Zhihao Shang, Katinka Wolter, TRAK- A Testing Tool for Studying the Reliability of Data Delivery in Apache Kafka, 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)

[13] HaiTao Mei, Ian Gray and Andy Wellings, A Java-Based Real-Time Reactive Stream, 2016 IEEE 19th International Symposium on Real-Time Distributed Computing

[14] Abderrahmane Ed-daoudy, Khalil Maalmi, Application of Machine Learning Model on Streaming Health Data Event in Real-Time to Predict Health Status Using Spark, 00978-1-5386-7328-7/18 ,2018 IEEE

[15] Gautam Pal, Gangmin Li, Katie Atkinson, Big Data Real Time Ingestion and Machine Learning, IEEE Second International Conference on Data Stream Mining & Processing, August 21-25, 2018, Lviv, Ukraine, 978-1-5386-2874-4/18

[16] Erum Mehmood And Tayyaba Anees , Challenges and Solutions for Processing Real-Time Big Data Stream: A Systematic Literature Review,IEEE Access Digital Object Identifier 10.1109/ACCESS.2020.3005268

[17] Thandar Aung, Hla Yin Min, Aung Htein Maw, Coordinate Checkpoint Mechanism on Real-Time Messaging System in Kafka Pipeline Architecture, University of Information Technology, Yangon, Myanmar

[18] Buqing Shu, Haopeng Chen, Meng Sun, Dynamic Load Balancing and Channel Strategy for Apache Flume Collecting Real-time Data Stream, 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)

[19] Gautam Pal, Gangmin Li, Katie Atkinson Near Real-Time Big Data Stream Processing, Platform Using Cassandra, 2018 4th International Conference for Convergence in Technology (I2CT) SDMIT Ujire, Mangalore, India. Oct 27-28, 2018

[20] D.Surekha, G.Swamy, Venkatramaphanikumar, Real Time Streaming Data Storage and Processing using Storm and Analytics with Hive, 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)

[21] Alramzana Nujum Navaz, Saad Harous, Mohamed Adel Serhani,, Ikbal Taleb, Real-Time Data Streaming Algorithms and Processing Technologies: A Survey, 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) December 11,12, 2019, Amity University Dubai, UAE

[22] IEEE Transactions on Parallel and Distributed Systems, Evaluation of Stream Processing Frameworks, Giselle van Dongen and Dirk Van den Poel, VOL. 31, NO. 8, AUGUST 2020