2018 International Conference on Information, Communication, Engineering and Technology (ICICET)
Zeal College of Engineering and Research, Narhe, Pune, India. Aug 29-31, 2018

A Study of Apache Kafka in Big Data Stream Processing

Bhole Rahul Hiraman
Assistance Professor, Department of Information Technology
Zeal College of Engineering & Research, Narhe, Pune,India
rahul.bhole@zealeducation.com

Chapte Viresh M.
Assistance Professor, Department of Information Technology
Zeal College of Engineering & Research, Narhe, Pune,India
viresh.chapte@zealeducation.com

Karve Abhijeet C.
Assistance Professor, Department of Information Technology
Zeal College of Engineering & Research, Narhe, Pune,India
abhijeet.karve@zealeducation.com

***Abstract*— Big data the name implies huge volume of data. Now a days streaming of data is more popular model which enables real time streaming data for data analytics. In current era Apache Kafka is most popular architecture used for processing the stream data. Kafka is scalable, distributed, and reliable result into high throughput. It also provides an API similar to messaging system.**

Keywords*— *Big Data, Stream data, Apache Kafka, Crypto-Currency

I. INTRODUCTION

Data is one of the new ingredient for Internet-based applications. In new trends for internet applications, data used for real time analytics is become a part of production data. Data is generated in a large volume through various activities for example, a social network platform produce from clicks, in retail data produce through order, sales & shipment etc. This types of data can be considered as stream data [1]. Stream processing is now became a popular paradigm which allow us to get result for real time & continuously for large volume of fresh data.

A. What is stream Processing?

A stream processing system refers to combination & processing of data before the data is store in storage medium. This system is built on multiple elements called as SPE (Stream Processing Element) [2], each SPE takes an input from data production perform computing & generates output.

B. What is Messaging system?

Messaging system is used for transferring of data from one application to another application, applications can focus only on data not on how data is shared. There are many traditional massing system but most of these dose not handle the big data in real time environment. Distributed messaging system focus on reliable messaging queuing. There are two types of message pattern, one is P to P (point to point) [3] [4] and second is public-subscribe. The public-subscribe which is also called pub-sub is used in massing system.

P to P messaging system (point to point)

In this system sender send the messages in queue and at the receiver end receive message in queue. The example of this system is order processing system. The figure 1 shows the P to P messaging system.

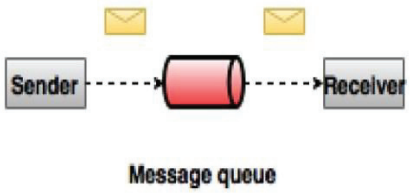

Fig. 1. P to P messaging system

Public-Subscribe (pub-sub)

In this system message sender is called as publishers & message receiver is called as subscriber. A real-life example of system is Dish TV which play different channels like movies, music, sports, news etc. here any one can subscriber to that particular Dish TV & subscribe for the available channels.

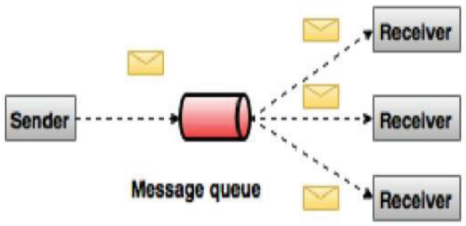

Fig. 2. Pub-sub messaging system

II. APACHE KAFKA MODEL

Apache Kafka is a platform for real time environment using distributed public-subscribed messaging system & it can handle a large volume of data which enables you to send messages at end-point. Apache Kafka is developed at LinkedIn & available as an open source project with Apache Software Foundation.

Following are some points that describe why Kafka.

1. Scalability: This framework scale easily without down time.
2. High-volume: it is designed to work with high volume of data.
3. Reliability: Kafka is partitioned, replicated, distributed & fault tolerance.

978-1-5386-5510-8/18/$31.00 ©2018 IEEE

4. Data Transformations: this frame work should provide provision for ingesting the new data stream from producer.
5. Low latency: to focus on traditional messaging, requires low latency.

A. *Apache Kafka Framework*

Apache Kafka is public-subscribed messaging system which is designed to be scalable, fast, reliable & durable. Fig 3 shows the Kafka Framework.

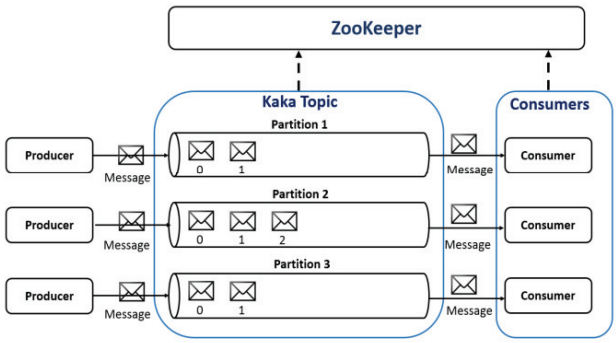

Fig. 3. Apache Kafka Framework

For knowing the Kafka framework we must have aware of some terminologies.

1. **Topic**: A topic is feeding system through which messages are stored & published, all Kafka messages are organized into topics. If you wish to read a message you read it and if you wish to send a message you send it to a specific topic. Producer applications write data to topics and consumer applications read from topics. A Kafka Topic divided into multiple partitions.
2. **Producers:** Producers are the publisher of messages to one or more Kafka topics. Producers send data to Kafka brokers. Every time a producer publishes a message to a broker. Producer can also send messages to a partition of their choice.
3. **Consumers:** It read data from brokers. Consumers subscribes to one or more topics and consume published messages by pulling data from the brokers
4. **Connectors:** It responsible for pulling stream data from Producers and delivering stream data to Consumers or Stream Processors.
5. **Stream processor:** Stream Processors are applications that transform data streams of topics to other data streams of topics in Kafka Cluster.
6. **Broker:** Kafka cluster typically consists of multiple brokers to maintain load balance. Kafka brokers are stateless, so they use Zookeeper for maintaining their cluster state.
7. **Zookeeper:** Zookeeper is used for managing and coordinating Kafka broker. Zookeeper service is mainly used to notify producer and consumer about the presence of any new broker in the Kafka system or failure of the broker in the Kafka system.

III. Related Work

Traditional message system exists from long time & play important role for data processing [5] [6] IBM WebSphere MQ allows an application to insert message into multiple queues automatically. . In JSM [7] individual messages acknowledge after processing. Recently Hedwig system [8] is available for distributed pub-sub system which is developed by Yahoo! It is scalable & offers strong durability guarantees. Apache Kafka works in combination with Hbase, spark for real-time analytics & performing streaming data. Now a days many MNC companies that are using Apache Kafka in there use cases they are as follows.

- Twitter: Twitter uses Kafka as a stream-processing infrastructure.
- LinkedIn: Apache Kafka is used at LinkedIn for the streaming data. This data uses in various product such as news feed & offline analytical system.
- Yahoo!: Kafka is used by Yahoo for their media analytic team in real time analytics.
- Netflix: Kafka used by Netflix as the gateway for data collection, this application requiring billions of messages to be processed daily.

IV. Testing Apache Kafka

We conducted the experiment on crypto-currency comparison [9] like BitCoin, Either & LiteCoin which is trending now a days. For this experiment we have written a code of publisher & subscriber in simple HTML file as Kafka uses on pub-sub messaging system.

Sample Code for Publisher

```
<html>
<head>
        <title> Crypto Publisher </title>
<script
src="https://cdn.pubnub.com/sdk/javascript/pubnub.4.18.0.
min.js"> </script>

</head>
```

Sample Code for Subscriber

```
<html>
<head>
        <title> Crypto Subscriber </title>
<!--<script
src="https://cdn.pubnub.com/sdk/javascript/pubnub.4.18.0.
min.js"> </script> -->
<scripttype="text/javascript"src="https://pubnub.github.io/e
on/v/eon/1.0.0/eon.js"> </script>
<linktype="text/css"rel="stylesheet"href="https://pubnub.git
hub.io/eon/v/eon/1.0.0/eon.css">
</head>
```

V. RESULTS

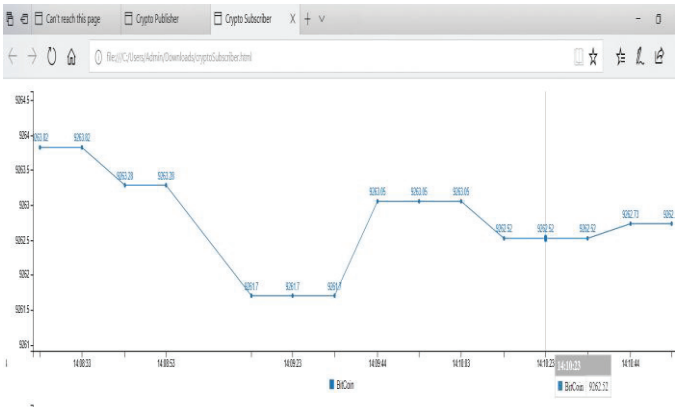

Fig. 4. Screen Shot 1

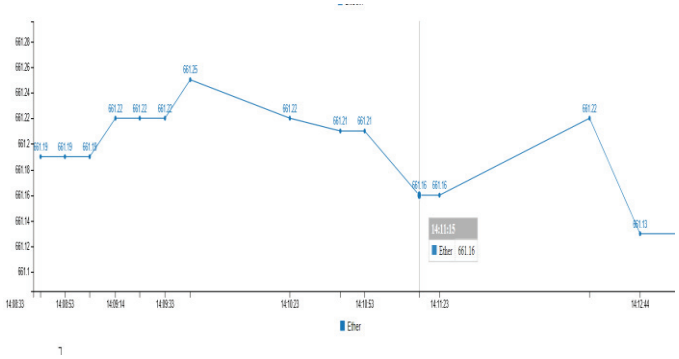

Fig. 5. Screen Shot 2

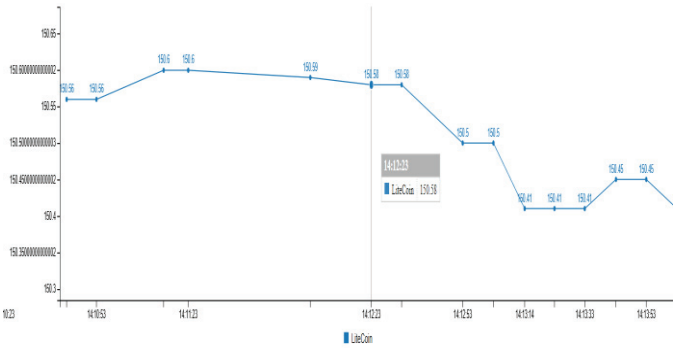

Fig. 6. Screen Shot 3

VI. CONCLUSION

In this work we focus on how to deal with Kafka & how to tune with its deployment. Kafka will help stream processing developer for effective use their big data processing architecture. Kafka defines a pull based model that allows application can consume data whenever needed.it achieves higher throughput than the traditional messaging system

REFERENCES

[1] Jay Kreps, Neha Narkhede and Jun Rao, ―Kafka: a Distributed Messaging System for Log Processing‖, LinkedIn Corp.

[2] https://www.htcinc.com/wp.../apache_kafka_event_stream_processing_solution.pdf

[3] https://www.tutorialspoint.com/apache_kafka/index.htm

[4] https://kafka.apache.org/quickstart

[5] http://activemq.apache.org/

[6] IBMWebsphereMQ: http://www01.ibm.com/software/integration/wmq/

[7] JAVA Message Service: http://download.oracle.com/javaee/1.3/jms/tutorial/1_3_1-fcs/doc/jms_tutorialTOC.html.

[8] https://issues.apache.org/jira/browse/ZOOKEEPER-775

[9] https://en.wikipedia.org/wiki/Cryptocurrency

[10] Nishant Garg, ―Apache Kafka‖ , PACKT Publishing

[11] V. Ta, C. Liu, G.W. Nkabinde, "Big Data Stream Computing in Healthcare Real-Time Analytics", 2016, IEEE International Conference on Cloud Computing and Big Data Analysis, Pages: 37 -42, doi: 10.1109/ICCCBDA.2016.7529531.

[12] A. Sotsenko, M. Jansen, M. Milrad, J. Rana, "Using a Rich Context Model for Real-Time Big Data Analytics in Twitter", 4th International Conference on Future Internet of Things and Cloud Workshops, 2014, Pages 228 -233, DOI 10.1109/W-FiCloud.2016.55.

[13] B. Yadranjiaghdam, N. Pool, N. Tabrizi, "A Survey on Real-time Big Data Analytics: Applications and Tools," in progress of International Conference on Computational Science and Computational Intelligence, 2016.