Data Ingestion using a Novel Method: H-Stream Framework

Gunturi S. Raghavendra, Shanthi Mahesh, M. V. P. Chandrasekhara Rao

Abstract— Current huge volumes of data is generated from wide variety of data sources and there is lot of demand for processing, this data. Apache Hadoop is designed for batch processing. Though Hadoop is used for batch processing there is lot of requirement for real time stream processing and querying on unstructured data. Data ingestion tools of Hadoop are playing a key role in processing of streamed log data. With the increase of volume of the data performance of data ingestion tools goes down linearly. In this paper we discuss solutions for performance issues of data ingestion tools, capturing and processing of streamed multimedia data along with real-time stream processing with the help of frame work known as H-Stream framework. (Abstract)

Keywords—Ingestion, Kera, Frame work, Lambda, H-stream

I. INTRODUCTION

The traditional methods we are using will collect high volumes of data which is unmovable i.e. static data. But analysis of streaming data is necessary to get deep insights present in the data and it must be analyzed whenever it is available.

A. Batch Processing

The group of the jobs will be executed at a time in offline fashion and hence is known as offline processing. Hadoop is an example of batch processing approach[1]. The disadvantage of batch processing is it can't be used for processing of data immediately whenever it is available. [1]

B. Real-time processing

Most of current domains and applications started calling of using real-time response on Big Data for accurate and rapid decision support.Examples of real time data applications: social media, mobile data, stock market data.[1]

C. Challenges of real-time processing systems

1) Data gathering: It is tedious task to handle a huge stream of dynamic data. The velocity with the data is coming will be a great challenge for the processing system to adapt to. Another challenge is type of data whether it is structured or unstructured. [1]

2) Data Storing: persistently storing large volume of data is another challenge. Data storage changes with the requirement of application. [1]

3) Data Modelling: Real time processing applications need in-memory processing to have low-latency. Better modelling and algorithms are needed to process these data. [1]

Revised Manuscript Received on January 10, 2020.

Gunturi S. Raghavendra*, Department of CSE, Atria Institute of Technology, Bangalore, India. E-mail: raghavendragunturi@gmail.com

Dr. Shanthi Mahesh, *, Department of ISE, Atria Institute of Technology, Bangalore, India. E-mail: shanthi.mahesh@atria.edu

Dr. M. V. P. Chandrasekhara Rao*, Department of CSE, R.V.R & J.C College of Engineering, Guntur, India. E-mail: manukondach@gmail.com

II. INGESTION OF DATA: METHOD 1

Applications are slowly moving from batch processing approach to streaming approaches to acquire valuable patterns. For these approaches, Streaming processing pipeline plays a key role is ingestion of data. [2]

A. Process

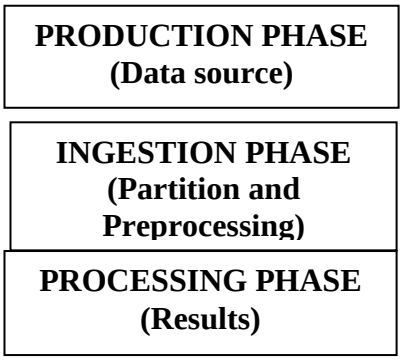
This method overcome the disadvantages of fixed stream divisioning by using an approach of variable divisioning scheme to improve throughput, latency and scalability.[2]

Big Data stream processing can be described in three phases

1)Production phase: Data from various sources are gathered.

2)Ingestion phase: Data is collected from source and portioned and preprocessing will be done. [2]

3)Processing phase: Processing of data for accurate results are done. [2]



B. Static partitioning Kafka

Utilize a static divisioning scheme where the streams are partitioned in to a fixed number of units, every one of which is an unbounded, requested. Every Kafka producer is answerable for one or numerous units/allotments. Producer gain records in fixed measured batches, every one of which is annexed to one segment. To decrease correspondence overhead, the producer batch together numerous clumps that compare to the segments of a solitary intermediary in a solitary solicitation. Every consumer is allotted to at least one segments. Each parcel doled out to a single consumer. [2]

Drawback The application needs a priori knowledge about the optimal number of partitions.in real-life situations it is difficult to know the optimal number of partitions a priori, both because it depends on a large number of factors (number of brokers, number of consumers and producers, network size, estimated ingestion and processing throughput target, etc.) [2]

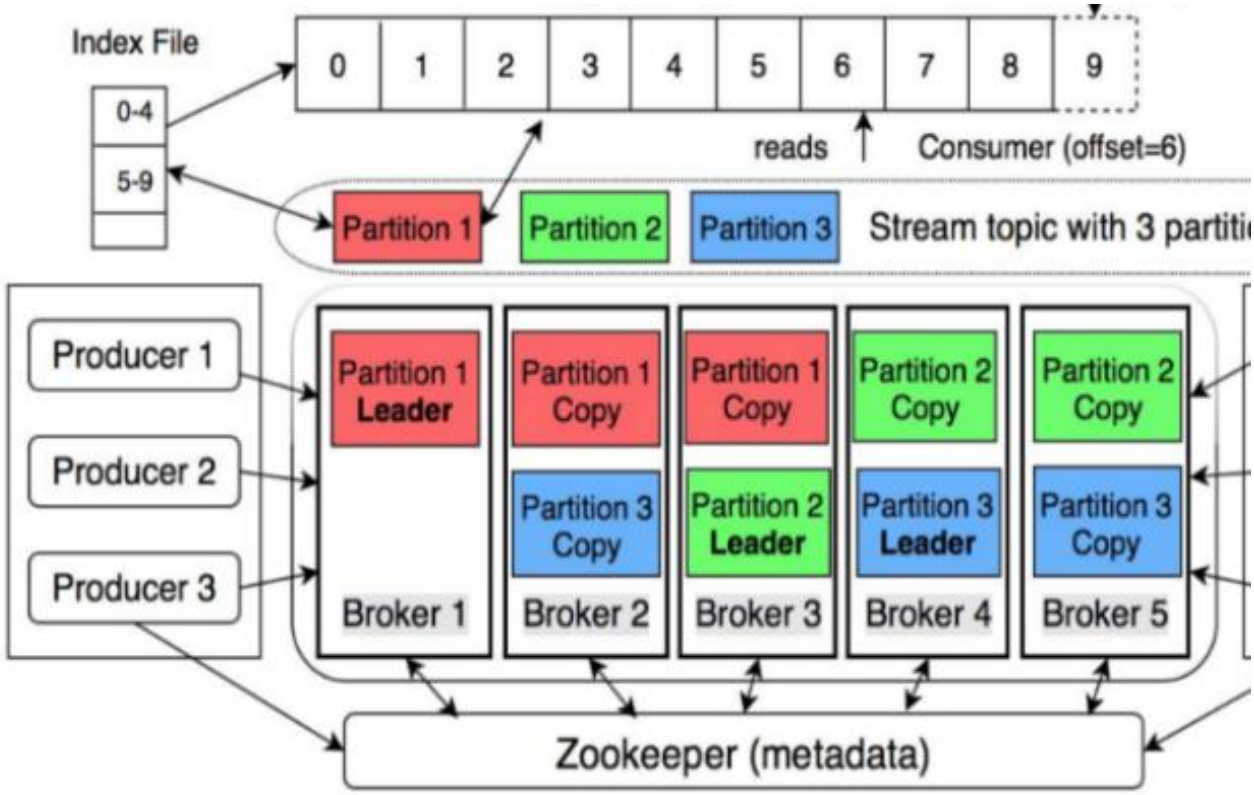


Fig 1: Architecture of Kafka

III. APACHE FLUME LOG PROCESSING

Apache Flume, a reliable framework used for processing, has some limitations and drawbacks on load balancing and storage. [3]

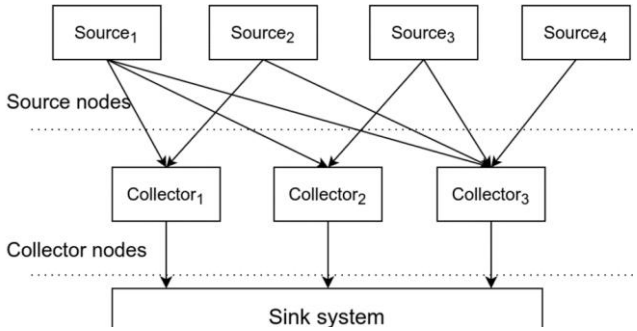


Fig 2: Architecture of Flume

According to the native system architecture with common three-tier pattern shown in Figure 2, the source node publishes information and sends information into Flume's information stream over RPCdepending on the Flume Client SDK, whilethe collector node gets information from source data nodes, at that point stores them in internal channels briefly and move them to the sink framework. The structure of sink framework isn't depicted in light of the fact that it could be any distributed storage (for example HDFS) or ongoing handling framework (Storm) with different designs. [3] Data will be separated into a few batches during the age. Each batch will be sent to a specific gatherer hub by source hub as indicated by load adjusting technique. Be that as it may, local flume just supports some essential calculation without the thought of conditions of the whole group like cooperative effort and hash. Clearly, the fundamental Flume methodology plans to send a similar amount to every hub, which just functions admirably on the reason that all authority hubs never crash and hold similar assets [3]. Be that as it may, practically speaking, the accident of hub is unavoidable and raises more system overhead because of re-association and information re transmission. [3] Also, the presentation is very extraordinary among hubs with various assets particularly memory. So the hub with poor assets will be the bottleneck of the batch before the hub with rich assets gets over-burden. [3]

IV. DATA INGESTION ARCHITECTURE

A. Lambda Architecture

Lambda Architecture (LA) [5is a standard system for overseeing enormous information which empowers blending of real-time information with cluster information. The fundamental design of lambda offers three layers: speed layer for constant information, batch layer for enormous volume of static authentic information pool and serving layer that incorporates continuous and batch jobs. Lambda Architecture coordinates low dormancy constant system with high throughput Hadoop batch structure. [5] information from Kafka message line gets ingested to both and stream processor structures. While stream processors can break down information, batch module stores the ingested information pool into HDFS over the time before preparing. [5] Apache Storm and Hadoop Map Reduce structures are utilized at stream and group modules separately. [5] A NoSQL information store (Cassandra) joins the batch and ongoing perspectives at the serving layer.

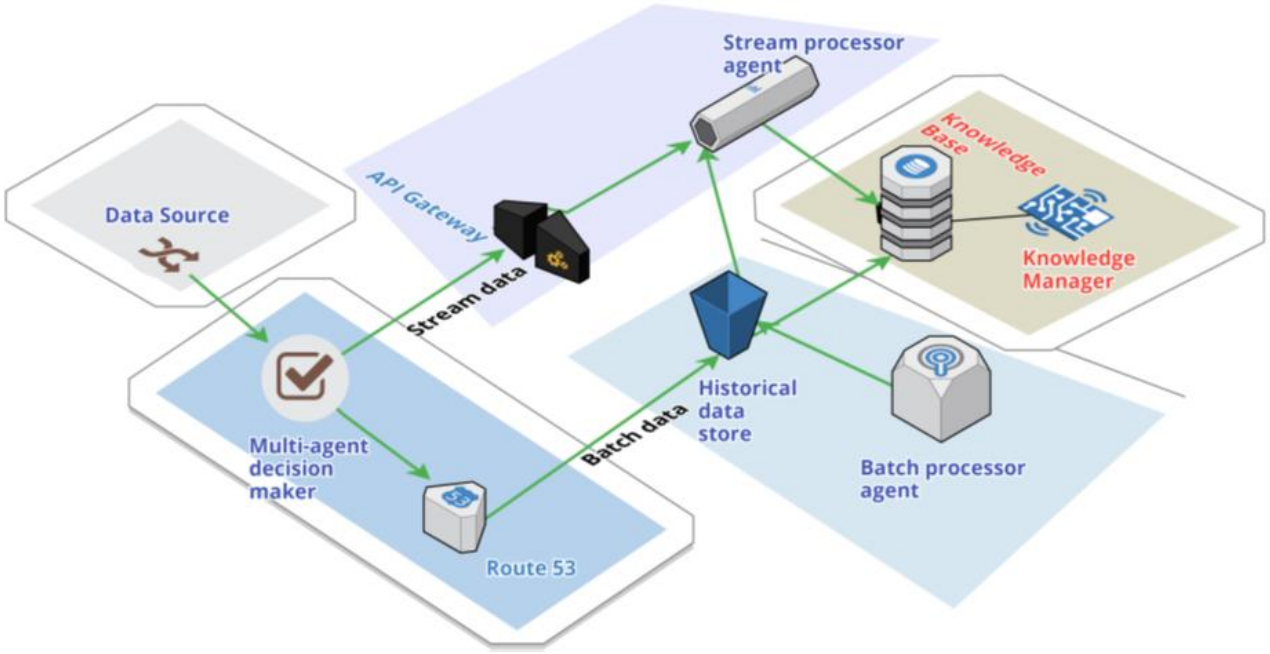


Fig 3: N Phase Lambda Architecture of Data Ingestion

B. Stream Processing Engine it instates itself with prepared model produced from group put away into HDFS. Stream motor uses the batch information as beginning balance to begin with and expands over it, steadily at a predetermined window interim. The fundamental test spilling layer faces is to process in-flight high speed of ingested information without first putting away into a document framework or a database. *[5]*

C. Data Miner has two sub segments: Distributed storage and preparing by Map Reduce. HDFS or a NoSQL information stores like Cassandra, MongoDB or Base can be a capacity alternative for batch oriented jobs *[5]*

D. Knowledge Miner and Knowledge Base Knowledge Minder (KM) is the cutting off layer to the end client. It joins the perspectives from group and stream to give a deep rooted learning system. KM perseveres the outcomes into the Knowledge Base (KB). KM likewise performs information filtration, gives the system health and execution insights for information administration and monitoring purposes. *[5]*

V. DATA REAL TIME INGESTION AND MACHINE LEARNING

A. Streamed Machine Learning

Batch machine learning is applied for fixed arrangement of information. Normally, these systems are likewise iterative, and we play out various ignores preparing information to join to an ideal arrangement. In opposite, web based learning predicts on each progressing window of time span. [6] In a steady manner the model persistently refreshes as new data is gotten. Be that as it may, web based learning model can be utilized alongside batch setting. Like we can utilize stochastic slope plummet (SGD) enhancement to prepare arrangement and relapse model after each preparation model. [6]

B. Streaming Regression

Training: Takes the labelled data points. Model gets trained on every batch of the input stream. It can be called repeated time to train on different stream. [6]

Predict: It also take labelled data points and tells the model to make prediction on the input stream. [6]

C. Streaming K-Means Clustering

In streaming K means clustering, model is refreshed with each passing window utilized on a blend between group focuses figured from the past batchs and the present cluster. Calculation begins with allotting information focuses to their closest batch. [6] For each new emphasis, when new information comes, register new group focuses, at that point update each batch utilizing following recipe [6]

VI. PERFORMANCE COMPARIOSN OF DATA INGESTION TOOLS

SNO	DATA INGESTION TOOL	DESCRIPTION
1	Apache Kafka	Message Broker System.Peformance lags with size of data
2	Apache Nifi	Provides directed graph of data routings.it is system mediation logic
3	Wavefront	Used for data ingesting ,visualizing and alerting metric data
4	Amazon Kinesis	Cloud Based data ingestion system.
5	Apache Samza	Message API, It maintains snapshotting and restoration of stream processor state.
6	Apache Flume	Low end data ingestion system only works well with small data with high latency
7	Apache Sqoop	Static data ingestion system, work only with data bases

VII. LIMITATIONS OF ABOVE INGESTION FRAME WORKS

A. Data Acquiring It is absurd to expect to deal with voluminous stream of streaming information. The framework must be fit for adjusting with the speed of approaching information and furthermore with assortment of information. The Processing of organized information goes about as an ideal contribution for direct frameworks, while the unstructured information requires parcel of information pre-preparing like separating, extraction and association into organized configuration. The dormancy of the stream preparing framework shifts with organized and unstructured information. The right portrayal of information and information securing procedures rely upon the application based on the highest point of stream handling frameworks.

B. Data Handling Second challenge is to appropriately deal with huge volume of information. The application requires examining the affectability of information, which need to store into persevering stockpiling. A few applications just require putting away the combined prepared outcomes while different applications require putting away sifted and fundamentally composed handled information for later utilization and investigation. The information taking care of and persevering stockpiling of information design changes with the application necessity. It should be appropriately evaluated by stream preparing frameworks.

C. *Data Modelling* The preparing frameworks require in-stream handling capacities to have a low idleness. Thinking about the volume, assortment, speed and veracity of information, the stream preparing framework requires prescient models and effective calculations to extricate application connected to significant occasions from monstrous information streams. It likewise requires information models to perform extensive examination by joining every single accessible datum.

VIII. H-STREAM FRAMEWORK

Real time processing of streaming data by using H-Stream frame work is done in two phases.

A. Phase 1

In this phase developing of a frame work known as H-Stream is done. The importance of this Frame work it can handle any type of data and can process data better than previous techniques

The tool may process data as following

1.Input data can be given from various sources like social media, YouTube etc

2.Then the data may first have compressed by using Map-Reduce version 2 (YARN).

3. HIPI is a picture preparing library intended to be utilized with the Apache Hadoop Map Reduce system. It gives an answer for how to store an enormous assortment of pictures on the Hadoop Distributed File System (HDFS) and make them accessible for productive appropriated handling. The essential info item to a HIPI program is a Hipi Image Bundle (HIB). A HIB is an assortment of pictures spoke to as a solitary record on the HDFS. The HIPI conveyance incorporates a few valuable apparatuses for making HIBs, including a Map Reduce program that fabricates a HIB from a rundown of pictures downloaded from the Internet. The main handling phase of a HIPI program is a separating step that permits sifting the pictures in a HIB dependent on an assortment of client characterized conditions like spatial goals or criteria identified with the picture metadata. This usefulness is accomplished through the Culler class. Pictures that are winnowed are rarely completely decoded, sparing preparing time. The pictures that endure the separating stage are doled out to singular guide errands such that endeavours to augment information territory, a foundation of the Hadoop Map Reduce programming model.

B. Phase 2

In this phase the data which will be the output of H-Stream may be analysed by using machine learning algorithm like KNN, SVM. After successful categorization of data. Then data is analysed by using of Hadoop tools like Hive, pig. Advanced data visualization techniques like isoclines, iso-surface, Oracle Visual Analyser, Microsoft Power BI for 2D, 3D visualization of processed data are used.

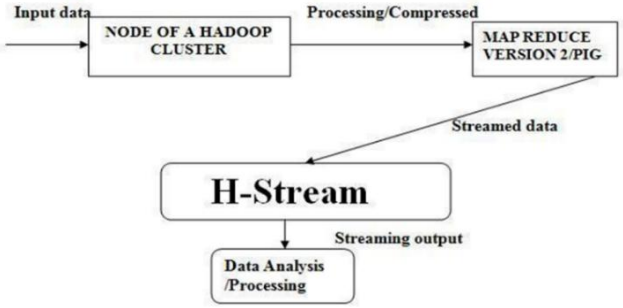


Fig 4: Architecture of H-Stream Framework

C. Implementation of H-Stream Framework

H-Stream work is a framework used for capturing large voluminous data of any type. The frame work may be developed by using existing advanced software tools like Coda, IntelliJ Diffmerge etc. It contains components: Mutiple sources, multiple channels connected in network, sinks, and visualize/decision supporter.

IX. EXPECTED RESULTS OF THE PROPOSED METHOD

1. Real time querying helps users to take accurate instant decision support.
2.Need for additional hardware and tools for processing of large data can be decrease
3. Highly robust frame work which makes querying/processing easy.
4. They may be no need of separate tools for capturing streaming data processing and visualization since everything is encapsulated as a single frame-work.
5. Works on both structured and unstructured data.

X. CONCLUSION

The H-Stream Framework will be well suited for those applications which require and demand for real-time processing. Traditional methods of ingestion don't support real time low latency processing. In future This frame work can be extended for all multimedia applications which require real time analysis and frame work.

REFERENCES

1. Adesh Chimariya B. Professor Mika Mäntylä, "Streaming Data AnalyticsBackground, Technologies, and Outlook," Master's Thesis, University of Oulu
2. Ovidiu-Cristian Marcu , Alexandru Costan , Gabriel Antoniu , Mar´ıa S. Perez-Hern ´ andez ´ Bogdan Nicolae† , Radu Tudoran, Stefano Bortoli 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) ,pp.1480–1485.
3. UnGyu Han and Jinho Ahn, "Dynamic Load Balancing Method for Apache Flume Log Processing," in Advanced Science and Technology Letters, Vol.79 (IST 2014), pp.83-86
4. Yang Ruan, Zhenhua Guo, Yuduo Zhou, Judy Qiu, Geoffrey Fox, "HyMR: a Hybrid MapReduce Workflow System," ACM 978-1-4503-1339-1/12/06.
5. Gautam Pal, Gangmin Li , Katie Atkinson "Multi-Agent Big-Data Lambda Architecture Model for E-Commerce Analytics " ,,mdpi ,pp.1-15.
6. Gautam Pal, Gangmin Li , Katie Atkinson "Big Data Real Time Ingestion and Machine Learning", IEEE Second International Conference on Data Stream Mining & Processing,pp.25-

AUTHORS PROFILE

Mr. G. S. Raghavendra, is a research scholar in Department of Computer Science and Engineering, His Area of research is Big Data Analytics. He has done 5 publications.

Dr. Shanthi Mahesh, is working as Head and professor of Department of Information Science and Engineering at Atria Institute of Technology, she has 18 Publications and received 5 Lakhs research Grants.

Dr. M. V. P. Chandrasekhara Rao, is working as professor of Department of Computer Science and Engineering at R.V. R & J.C. College of Engineering. he has 25 Publications and he is the Life Member of Computer Society of India.