ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Comparative Study of Open Source Processing Frameworks for Analysis of Big Data

Sanisha
Department of Computer Science
HPU
Shimla, India

Dr. Kishori Lal Bansal
Department of Computer Science
HPU
Shimla, India

Abstract: Multiple terabytes of data is being generated every moment nowadays by various devices. Merely collecting data is worthless if not utilized for better decision making. Big Data Analytics provide the value to data and helps in improving decision making. Various data processing frameworks are available, open source as well as enterprise based, which can process data in batches as well as in real time. This paper provides an overview and comparative study of such open source frameworks highlighting their main features. The aim of the paper is to give better insight of open source frameworks and help researchers find the best framework for their application.

Keywords: Big Data; Data analytics; Big Data Analytics; Hadoop; Spark; Flink; Samza; Storm

I. INTRODUCTION

Today data being generated by us is growing at exponential rate. The amount of data generated from the beginning of time till 2003 was 5 billion gigabytes. The same amount of data was generated in every two days in 2011, and every 10 minutes in 2013. We perform 40,000 search queries every second on Google alone. Facebook users send on average 31.25 million messages [1].Approximately up to 300 hours of video are uploaded to YouTube.

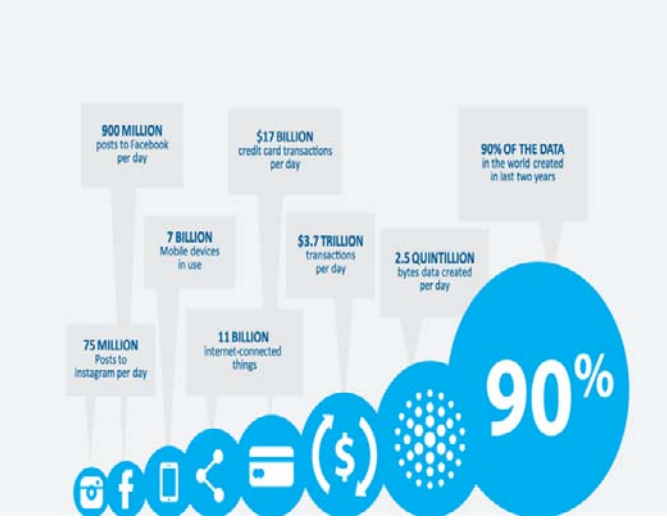

Figure 1.Sources of Big Data [2]

Merely collecting huge amount of data is worthless until it is not used for decision making process in the organizations [3]. Big Data Analytics is the process of analyzing data to discover hidden patterns, correlations and valuable information from Big Data. To extract the hidden information, the data needs to be processed. Different frameworks are available for processing the data [4]. These frameworks differ in the way of processing data. Some of them process data in batch while others process data as a continuous stream as it flow into the system. Still others can process data in either ways. Some of the open source frameworks are: Hadoop, Storm, Samza, Spark and Flink. These frameworks need to be compared to choose the best among them to suit the requirements of end user.

Further sections are divided as: section II gives a brief literature review. Section III describe the evolution of big data. Section IV provide an elaboration of data analytics and big data analytics. Section V provide an overview of data processing systems. Section VI briefly describe various open source frameworks for data computation. Section VII have comparative analysis of frameworks defined in the paper. Section VIII concluded the topic.

II. LITERATURE REVIEW

Yashika Verma, Sumit Hooda [5]: identified various issues with big data as about 80% of today's data is unstructured. Traditional approach to process data does not scale for big data. Also high cost is associated with load and store of data. They found Hadoop to be the framework to solve these issues. Its HDFS component can store structured as well unstructured data while map reduce makes analysis easier.

Samiddha Mukherjee, Ravi Shaw [6]: found the applications of Big Data in different fields such as healthcare, fraud detection, understanding customer behavior, food industry, telecom etc. They defined two approaches to deal with big data: breaking large data sets into smaller sets, powerful server with massive storage. They have discussed challenges with big data implementation in the paper.

Nada Elgendy and Ahmed Elragal [7]: proposed a framework which incorporates the big data analytics tools and methods into the decision-making process. This framework maps different Big Data storage, management, and processing tools, analytics tools and methods, and visualization and evaluation tools to the different phases of the decision making process.

T. Giri Babu, Dr. G. Anjan Babu [8]: defined Data Science as an emerging technology which deals with storing, analyzing and presenting data. They studied various tools of Data Science Technology that can be applied to Big Data Analytics.

R.A.Fadnavis, Samrudhi Tabhane [9]: proposed map reduce technique in Hadoop. Map takes key value pair as an input. Reduce takes the output of map as input and process it to generate desired result.

Ms. Vibhavari Chavan, Prof. Rajesh. N. Phursule [10]: proposed a new framework called Dache-data aware chache to eliminate the short coming of map reduce

framework. Map reduce generates large amount of intermediate data which is not utilized by the framework.

V.Srilakshmi, V.Lakshmi Chetana, T.P.Ann Thabitha [11]: studied various technologies of Hadoop and enhanced Map Reduce runtime. They found Twister to be more efficient than other runtimes.

Abdul Ghaffar Shoro & Tariq Rahim Soomro [12]: presented analysis of data by Apache Saprk using twitter streaming API and concluded that Spark analysed data with latency of few seconds only.

AnkushVerma, Ashik Husain Mansuri, Dr. Neelesh Jain [13]: reviewed various analytic tools, techniques and frameworks like Hadoop,Saprk and Storm.

Zhigao Zheng, Ping Wang, Jing Liu, Shengli Sun [14]: proposed a four layer framework for real time data processing consisting of Data, Analytics, Integration and Decision layer.

III. EVOLUTION OF BIG DATA

In the early days of computing data was stored in flat files. A flat file is a plain text file or a binary file. Most of the tasks such as storage, retrieval and search need to be done manually. These files can store only the textual data. The problems with such system was redundancy of data and time consuming search process etc. Later on E.F.Codd presented 12 rules for the development of relational database systems where a single database can store multiple tables. Data is stored in the terms of records. This type of data is called structured data. The size of database ranges from 100 MB to 100 GB. These systems were suitable for transactional processing but not optimized for analytical needs and reporting. So the concept of Data warehousing system came into existence. The size of database is from 100GB to 100 TB. These were suitable for analytic purpose. But due to advancement in technologies huge amount of unstructured data is being generated every second. This data is difficult to manage, store and process by traditional data management systems. Hence the concept of Big Data came. Big Data is a large set of different types of data-structured, semi-structured and unstructured. Most of the analysts and practitioners currently refer to data sets from 30-50 TB to multiple petabytes as Big Data [15]. These data sets contain data which is huge in size (volume), generated and processed at a very high speed (velocity), contain different formats (variety) and have hidden information (value).

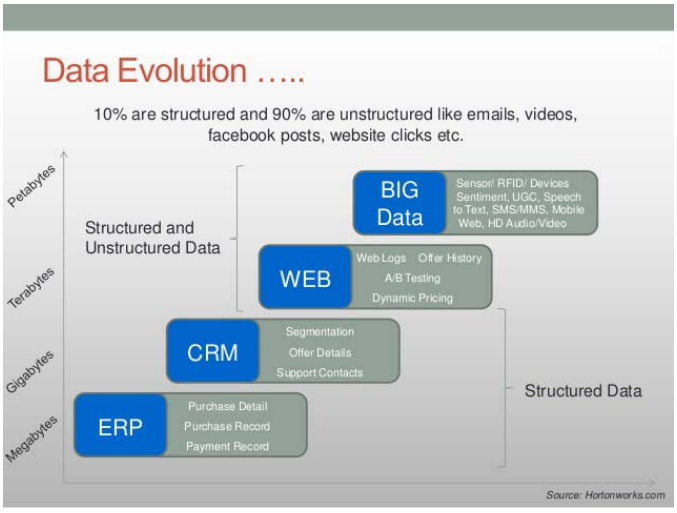

Figure 2.Big Data Evolution [16]

IV. DATA ANALYTICS AND BIG DATA ANALYTICS

Data analytics involve the process of examining data sets to draw hidden information they contain, with the help of specialized systems and software. Data analytics technologies and techniques are broadly used in commercial industries to enable organizations to make more-informed business decisions and by scientists and researchers to verify or disprove scientific models, theories and hypotheses. Data analytics make use of qualitative and quantitative analysis techniques and processes which are used to enhance productivity and business gain. Data is extracted and then categorized to identify and analyze behavioral data and patterns, and the techniques vary according to organizational requirements.

Quantitative data analysis technique is the process analyzing numeric data statistically with quantifiable variables that can be compared or measured. However the qualitative approach is more interpretive. Its emphasis is on understanding the content of non-numerical data like text, images, audio and video, including common phrases, themes and points of view.

A. Types of data analytics

Data mining: It involves sorting through large data sets to identify trends, patterns and relationships

Predictive analytics: It seeks to predict customer behavior, equipment failures and other future events.

Machine learning: is an artificial intelligence technique that uses automated algorithms to churn through data sets more quickly than data scientists can do via conventional analytical modeling.

B. Big Data Analytics

Big data analytics uses advanced analytic techniques to discover hidden patterns from big data. It apply the data mining, predictive analytics and machine learning tools to sets of big data that can be structured, unstructured and semi-structured data. It involves examining large data sets, in order to discover hidden patterns, unknown correlations, market trends, customer choice and other useful information that can help organizations make efficient business decisions.

The primary motive of big data analytics is to help companies make very effective decisions. It demands data scientists, predictive modelers and other analytics professionals to analyze large volumes of transaction data, as well as other forms of data. These data sets include server logs and clickstream data, social media contents and social network activity reports, text from customer emails, call records and data captured by sensors connected to the Internet of Things. Semi-structured and unstructured data do not fit in traditional data warehouses based on relational database model. Also data warehouses may not be able to handle the processing requirements posed by sets of big data that need to be updated frequently. For example, real-time data on the performance of mobile applications. As a result, many organizations are looking to collect, process and analyze big data resulting in new set of technologies which includes Hadoop Spark etc. These technologies are an open source software framework that support the processing of large and different data sets across clustered systems.

V. BIG DATA PROCESSING SYSTEMS

Processing frameworks are responsible for computing the data, operate over data in order to increase understanding, surface patterns, and gain insight into complex interactions. In order to get better insights from data, it needs to be

processed. The processing of data can be done in two ways depending upon the latency required- batch processing, stream processing.

A. Batch Processing Systems

Batch processing systems operate over a large, static dataset and return the result at a later time when the computation is complete.They are suitable for calculations where access to a complete set of records is required. As an example, calculating totals and averages, datasets must be treated as a whole instead of as a collection of individual records.

The datasets in Batch processing are have following properties:

bounded: batch datasets are finite collection of data

persistent: data is almost always backed up in some type of permanent storage

large: batch operations are only option for processing large sets of data

B. Stream Processing Systems

Stream processing systems compute data as it enter into the system. Its processing model is different from the batch processing. Stream processor define operations to be applied to data item as it enter the system instead of defining operations to apply to an entire dataset.

The dataset is unbounded in stream processing systems. It means that:

Total dataset is defined as the amount of data that has entered the system so far

The working dataset is limited to a single item at a time.

Processing is event-based and Results are immediately available and will be updated continuously as new data arrives.

Stream processing systems have the capability of handling large amount of data, but they can process only one record (true stream processing) or very few (micro-batch processing) items at a time. Analytics, server or application error logging, and other time-based metrics are some fields where stream processing best fits.

VI. OPEN SOURCE PROCESSING FRAMEWORKS

Different open source frameworks are there such as: Hadoop (batch only), Storm, Samza (Stream only), Flink, Spark (Hybrid).These has been detailed in the following section.

A. Hadoop

It is an Apache project founded by Doug Cutting in 2008 at Yahoo and Mike Cafarella at the University of Michigan. Hadoop is the best suit when the data can be processed in batch, and split into smaller processing jobs, and spread across a cluster, by recombining the efforts. Its efficiency comes from working with batch processes set up in parallel. Rather than moving the data through a network to a specific processing node, large problems are dealt with by dividing them into smaller problems. These problems are then solved, and results are combined to provide a final answer to the large problem.

Some features of Hadoop are:

- **Open-source**: Hadoop is an open source project. Its code can be modified according to business requirements.
- **Distributed Processing**: data is stored in a distributed manner in HDFS across the cluster and is processed in parallel on a cluster of nodes.
- **Fault Tolerance**: 3 replicas of each block is stored by default across the cluster in Hadoop and it can be changed as per the requirement. As any node goes down, data on that node can be recovered from other nodes.
- **Reliability**: Due to replication of data, it is stored on the cluster of machine despite machine failures. If the machine goes down, then also the data will be stored.
- **High Availability**: Data is highly available and is accessible despite hardware failure due to multiple copies of data. If a machine crashes, the data will be accessed from another path.
- **Scalability**: New hardware can be easily added to the nodes. It provides horizontal scalability which means new nodes can be easily added**.**

B**.** *Storm*

Storm is used for real time analytics and distributed Machine Learning. It is written in Clojure and it can be integrated in Hadoop ecosystem [17]. In Storm a graph of real time computation is designed which is called Topology. These topologies describe various transformations on data entering the system. The topologies consists of:

Streams: This is unbounded data coming into the system.

Spouts: Sources of data stream. These can be APIs, queues, etc. which produce data to be operated on.

Bolts: represent a processing step that consumes streams, applies an operation to them, and outputs the result as a stream.

Some features of Storm are:

- **Windowing management:** Storm provides support for sliding windows based on time duration and/or event count.
- **State management:** This framework automatically and periodically snapshots the state of the bolts across the topology.
- **Handling message level failures:** It supports 3 message processing semantics: at least once, at-most-once and exactly once.
- **Ease of development:** Storm provides an easy, rich APIs that describe the DAG nature of process flow (topology).
- **Language supported:** Storm applications can be created in Java,Scala and Clojure.
- **Storm Connectors:** Storm 1.0 provide support for Cassandra and MongoDB NoSQL stores

C**.** *Samza*

Samza is a stream processing framework and uses Apache Kafka's messaging system. It uses YARN for managing cluster resources. It is suitable if there is need for exactly once semantics with low latency.

Some features of Samza are [18]:

- **Simple API:** Samza provides a simple callback-based "process message" API comparable to MapReduce.
- **Managed state:** Samza can handle large amounts of state (many gigabytes per partition).It manages snapshotting and restoration of a stream processor's state. When the processor is restarted, Samza restores its state to a consistent snapshot.

- **Fault tolerance:** Samza works with YARN to transparently migrate the tasks to another machine, whenever a machine fails in the cluster.
- **Durability:** Samza make use of Kafka to guarantee that messages are processed in the order they were written to a partition, and ensure that no messages are ever lost.
- **Pluggable:** Although Samza works out of the box with Kafka and YARN, Samza also provides a pluggable API that allow it to run with other messaging systems and execution environments.
- **Processor isolation:** Samza uses Apache YARN for resource management, which supports Hadoop's security model, and resource isolation through Linux CGroups.

D. ***Flink***

Flink can handle both stream as well as batch processing. For stream processing it handle incoming data on item-by-item basis. For batch processing it reads a bounded data set off the persistent storage as a stream. It support batch, micro batch and streams.

Some features of Flink are:

- **Unified Framework:** Flink allows building a single data workflow that hold streaming, batch, SQL and Machine learning. It can also process graph using its own Gelly library and use Machine learning algorithm from its own FlinkML library.
- **Fault-tolerance** Flink uses Chandy-Lamport distributed snapshots based fault-tolerance mechanism. It allow the system to maintain high throughput rates and provide strong consistency guarantees as well.
- **Automatic cost-based optimizer:** Batch programs are automatically optimized to exploit situations where expensive operations (like shuffles and sorts) can be avoided, and when intermediate data should be cached.
- **Ease to use:** Flink APIs are easier to use than programming for Mapreduce and it is easier to test them as compared to hadoop.
- **Custom Memory Manager:** Flink has its own memory management inside the JVM. Flink do not throw an OutOfMemoy Exception and memory is allocated, deallocated and used strictly using internal buffer pool implementation.

E. ***Spark***

Spark provide full in memory computation to speed up batch processing. It uses Directed Acyclic Graphs to represent data to be processed, operations to be done and relationships between them. It does not process streams one at a time instead it slices them into small batches of time intervals [19]. These are called micro batches.

Some features of Spark are:

- **Real-Time Stream Processing**
  Spark provides real-time stream processing. It handle and process the real time data.
- **Fault Tolerance in Spark**
  Apache Spark provides fault tolerance through Resilient Distributed Datasets. Spark RDDs are designed to handle the failure of worker node in the cluster. Thus, the loss of data is reduced to zero
- **In-Memory Computation**
  Spark has DAG execution engine which provides facilitates in-memory computation. The data is being cached so there is no need to fetch data from the disk every time and thus the time is saved.
- **Reusability**
  The code of Spark can be reused for batch-processing.
- **Swift Processing**
  Apache Spark provide a high speed data processing of about 100x faster in memory and 10x faster on the disk. This is possible because of reduced number of read-write to disk.
- **Dynamic in Nature**
  We can easily develop a parallel application, as Spark provides 80 high-level operators.
- **Support Multiple Languages**
  Spark support multiple languages like java, R, Scala, Python

VII. COMPARISON OF BIG DATA PROCESSING FRAMEWORKS

There are different processing frameworks for big data. So they need to be compared in order to choose the one which suits the need of user. Table 1 shows the comparison for 5 framework on the basis of certain features.

Table I. Comparison of open source big data processing frameworks

Sr. No.	*Parameters*	*Hadoop*	*Spark*	*Flink*	*Storm*	*Samza*
1	**Distributed File Systems**	Own file system HDFS	Depends on HDFS	Depends on HDFS	GPFS, Lustre	-
2	**Scalability**	Highly Horizontally scalable	Horizontal	Horizontal	Horizontal	Horizontal
3	**Message delivery guarantee**	Exactly-once	Exactly-once	Exactly-once	At-least once	At-least once
4	**Streaming system**	Do not support streaming	Micro batching	Native	Native	Native
5	**Data processing engine**	At the core map reduce is batch processing	At the core Spark is batch processing engine	At the core Flink is stream processing engine	Stream only processing	Stream only processing

		engine				
6	**Cost**	Less expensive	More requirement of RAM increases cost	More requirement of RAM increases cost	More requirement of RAM increases cost	More requirement of RAM increases cost
7	**Data computation**	Disk-based	In memory	In memory	In memory	In memory
8	**Hardware Requirement**	commodity hardware	mid to high-level hardware .	mid to High-level hardware .	mid to high-level hardware .	mid to high-level hardware
9	**Auto-Scaling**	Yes	Yes	No	No	No
10	**Languages Supported**	Primarily Java, but other languages like C, C++, Ruby, Groovy, Perl, Python also supported using Hadoop streaming	Java, Scala, python and R	Java as well as Scala	Python, Ruby, and any JVM-based language	Scala, Java, and Python

A. ***Distributed file system:***

Distributed file system is a file system which is based on client server architecture. Data is stored at the server but can be accessed and processed by the clients as if it is local to them. Hadoop has its own distributed file system called HDFS, while the other frameworks do not have their own file systems. They rely on Hadoop's distributed file system. GPFS being used by Storm involve the licensing cost and it is based on disk sharing which means that access control will be given to the client's system and data will not be available if the client node will fail. Some of the features of HDFS are:

- It is open source.
- It is specially designed for commodity hardware.
- It have high scalability.
- It is fault tolerant.

B. ***Scalability:***

Scalability is the ability of a system to expand from the configurations it has for handling increasing amount of load. Scaling of the system can be done in two ways: i) by upgrading the existing hardware configuration (scale up) ii) by adding extra hardware (scale out). Hadoop, Spark, Flink and Storm are horizontally scalable i.e. nodes can be added to the cluster whenever required.

Hadoop have incredible scalability and can have tens of thousands of nodes. The largest known cluster of hadoop have 14000 nodes. Spark and Flink are also scalable and can have thousands of nodes in a cluster.

C. ***Message delivery guarantees:***

There are three message delivery guarantees in case of failure. Spark streaming and Flink provide exactly once delivery which means that message will neither be duplicated nor be lost and will be delivered to the recipient exactly once. Storm and Samza guarantee at-least once delivery. It mean that multiple attempts are made to deliver the message so that at least one succeeds. The message may be duplicated but will not be lost.

D. ***Streaming system***

Flink, Samza and Storm provide native streaming which means that the records will be processed immediately as they will enter into the system. However in case of micro batching, short batches of records will be created according to pre-defined time constant. Latency is very low with native streaming as record is immediately processed at its arrival.

E. ***Data processing engine***

Batch processing frameworks perform operations on large datasets which are static and return the result once computation is complete. These are suitable for processing large volumes of data at once. While Stream processing frameworks process records one by one as the data come into the system.

F. ***Cost***

Hadoop uses map reduce to do the computation task. Map reduce can run on commodity hardware and do not attempt to store everything in memory. So its cost is relatively less than other frameworks which require a lot of RAM. Due to the more requirement of RAM the cost increases.

G. ***Data Computation***

Hadoop's Mapreduce write the data back to disk after each operation. Hence full recovery can be made easily in case something goes wrong. While in other frameworks data is kept on RAM which is volatile. So the data is more prone to loss.

H. ***Hardware requirement***

Hadoop can easily run on commodity hardware which is reasonable and can be afforded easily. While the other frameworks require high level hardware which are relatively high in cost.

I. ***Auto scaling***

Hadoop and Spark support auto scaling. While rest of the frameworks do not.

J. Languages Supported

Hadoop provides support for more languages compared to other frameworks.

VIII. CONCLUSION

In this paper different open source frameworks have been compared by considering various parameters. Based upon the parameters taken, it can be concluded that Hadoop has come out to be best among these frameworks. Hadoop is more suitable as it do not require any specific hardware to run, it can easily run on commodity hardware. Also Hadoop has its own distributed file system which is highly scalable. Hadoop store the processed data immediately on the disk, thus its requiring less amount of RAM comparatively. It is less expensive to implement than other frameworks. Hadoop is more suitable to process huge amount of datasets at once thus resulting in high throughput.

IX. REFERENCES

[1] https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/.

[2] https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwivrZ3v74PUAhVFvY8KHV79CfcQjRwIBw&url=http%3A%2F%2Faoife.dbsdataprojects.com%2Ftag%2Fdata-evolution%2F&psig=AFQjCNGcm2LLGV0rVu-sQxbWig3REz7vUA&ust=1495555178940947

[3] Amir Gandomi, MurtazaHaider,"Beyond the hype: big data concepts, methods, and analytics",International Journal of Information Management, Volume 35, Issue 2, April 2015

[4] Justin Ellingwood,"Hadoop, Storm, Samza, Spark and Flink: Big Data Frameworks Compared",https://www.digitalocean.com/community/tutorials/hadoopstormsamzasparkandflinkbigdataframeworkscompared.

[5] YashikaVerma,SumitHooda," A review paper on Big Data and Hadoop",IJSRD - International Journal for Scientific Research & Development| vol. 3, Issue 02, | ISSN (online): 2321-0613, 2015

[6] Samiddha Mukherjee, Ravi Shaw, "Big Data – Concepts, Applications, Challenges and Future Scope", International Journal of Advanced Research in Computer and Communication Engineering vol. 5, Issue 2, February 2016.

[7] Nada Elgendy and Ahmed Elragal,"Big data analytics: A literature review paper",https://www.researchgate.net/publication/264555968, 2014

[8] T. Giri Babu, Dr. G. Anjan Babu," A Survey on Data Science Technologies & Big Data Analytics", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, Issue 2, February 2016

[9] R.A.Fadnavis, SamrudhiTabhane,"Big Data Processing Using Hadoop",International Journal of Computer Science and Information Technologies, vol. 6, 2015)

[10] Ms. Vibhavari Chavan, Prof. Rajesh. N. Phursule," Survey Paper on Big Data", International Journal of Computer Science and Information Technologies, vol. 5, 2014

[11] V. Srilakshmi, V.Lakshmi Chetana, T.P.Ann Thabitha," A Study on Big Data Technologies", International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, Issue 6, June 2016

[12] Abdul GhaffarShoro& Tariq Rahim Soomro, "Big Data Analysis: Ap Spark Perspective", Global Journal of Computer Science and Technology: C Software & Data Engineering vol. 15, Issue 1, Version 1.0, 2015

[13] AnkushVerma, Ashik Husain Mansuri, Dr. Neelesh Jain, "A Review on Big Data Environment on Different Frameworks, Techniques and Tools", International Journal of Core Engineering & Management (IJCEM), vol. 3, Issue 3, June 2016

[14] Zhigao Zheng, Ping Wang, Jing Liu, Shengli Sun,"Real-Time Big Data Processing Framework: Challenges and Solutions", Applied Mathematics & Information Sciences an International Journal, November 2015

[15] http://www.navint.com/images/Big.Data.pdf, March 2012

[16] https://image.slidesharecdn.com/bigdata-140128092341-phpapp02/95/big-data-6-638.jpg?cb=1390901096

[17] http://www.kdnuggets.com/2016/03/topbigdataprocessingframeworks.html

[18] https://samza.apache.org/learn/documentation/0.11/introduction/background.html

[19] https://tsicilian.wordpress.com/2015/02/16/streamingbigdatastormsparkandsamza/