e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

# SENSEX BIG DATA PROCESSING USING HADOOP AND PIG

**Hemant Gianey[*1], Shaurya Saxena[*2], Siddhesh Darak[*3], Jaykumar Patel[*4], Rahul Singh[*5]**

[*1]Associate Professor, Computer Engineering, SVKM's NMIMS, Shirpur, Maharashtra, India.

[*2,3,4,5]Student, Computer Engineering, SVKM's NMIMS, Shirpur, Maharashtra, India.

## ABSTRACT

This paper addresses the challenges of processing Sensex log data, crucial for assessing the efficiency of the Indian stock market. With 30 companies in the Sensex, the Bombay Stock Exchange (BSE) maintains a vast trade information database, often stored in large text files. To effectively tackle these issues, the paper explores the utility of MapReduce and Pig. MapReduce, a parallel and distributed algorithm executed on clusters, is employed for processing and generating extensive datasets. This framework guarantees reliability, scalability, and fault tolerance. Pig, a high-level programming language, facilitates the querying and analysis of large datasets, especially those stored in distributed file systems like Hadoop Distributed File System (HDFS). Operating at a high level of abstraction, Pig streamlines the complex data processing tasks, proving invaluable for investors and analysts seeking insights into the Indian stock market's intricacies.

**Keywords:** Big Data, Hadoop, Pig, MapReduce.

## I. INTRODUCTION

When analysing the efficiency of the Indian stock market, the Sensex is the primary index that is used as a point of comparison. The aforementioned dataset is an invaluable resource that helps investors gain a better understanding of the complexities of the Indian stock market and makes it easier for them to make well-informed investment decisions.

Despite this, the processing of Sensex log data presents a significant challenge due to the considerable volume of the data and the complexity of its structure. Processing is required in order to gain meaningful insights from the data, which is typically stored in large text files. This makes it difficult to access the data. It is common knowledge that the programming frameworks known as MapReduce and Pig are reliable tools that provide effective processing and analysis capabilities for massive data sets.

Utilising a parallel and distributed algorithm on a cluster, the programming paradigm known as MapReduce, along with its corresponding implementation, is used for the purpose of processing and generating extensive data sets. This is accomplished through the utilisation of the MapReduce programming paradigm. The MapReduce programming model is used to process and generate large data sets through the use of a parallel and distributed algorithm that is implemented on a cluster. This is accomplished by using the cluster. The MapReduce framework has been developed specifically for the purpose of handling large amounts of data in an effective manner while simultaneously ensuring their reliability, scalability, and fault tolerance.

A high-level programming language developed specifically for the purpose of querying and analysing large datasets is known as the Pig language. Pig is a high-level programming language that was developed specifically for querying large data sets that are kept in distributed file systems such as the Hadoop Distributed File System (HDFS). It functions at a high level of abstraction and operates at a very specific level of abstraction. Complex data processing tasks can be developed and carried out with much less effort thanks to the Pig framework.

## II. METHODOLOGY

The Bombay Stock Exchange (BSE) is responsible for maintaining an official listing for each of the 30 companies that are included in the composite index. All of the trades that were conducted on the Bombay Stock Exchange (also known as the Sensex) are included in the Sensex log data's exhaustive database of information.

### Data Collection

The initial stage in the processing of Sensex log data involves the collection of data from its diverse sources. The Sensex data can be acquired from the official website of the Bombay Stock Exchange (BSE) as well as from multiple data vendors. Data collection can be conducted manually or with the aid of diverse automated tools.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

Once the data has been gathered, it is imperative to store it in a format that exhibits a higher degree of organisation and structure. The data can be stored in a structured format, such as Comma-Separated Values (CSV) or Microsoft Excel, among various other available alternatives.

**Data Preprocessing**

Data preprocessing is a crucial step in the data analysis and machine learning pipeline. It involves cleaning and transforming raw data into a structured format suitable for analysis or modelling.

**Removing Duplicate Records:**

Identification: The first step is to identify duplicate records within your dataset. This is typically done by comparing the values of each record across all or specific columns.

Handling: Once duplicates are identified, can choose to keep only one instance of each unique record or aggregate them in some way if the data allows it.

**Filling in Missing Values:**

Identification: Missing values can be identified by looking for empty cells, NULL values, or placeholders such as "N/A" or "Not Available."

Handling: To deal with missing values, can choose from various strategies, including:

Imputation: Replacing missing values with a calculated or estimated value (e.g., mean, median, mode) based on the distribution of the data in that column.

Forward Fill or Backward Fill: Propagating the previous or next valid value in the same column to fill missing values in a time-series or sequential data.

Interpolation: Estimating missing values by interpolating between neighbouring data points.

Deletion: Removing rows with missing values if the data loss is acceptable and doesn't significantly affect the analysis.

**Correcting Errors in the Data:**

Identification: Errors in data can take various forms, such as typos, inconsistencies, outliers, and incorrect values. Data profiling, visualization, and domain knowledge are helpful in identifying these issues.

Handling: Correcting errors involves manual or automated processes, such as:

Data Cleaning Rules: Creating rules to correct common errors, like converting text to lowercase, removing special characters, or fixing typos.

Outlier Handling: Deciding whether to keep, remove, or transform outliers based on their impact on analysis.

Data Validation: Cross-checking data against predefined constraints or reference data to detect and correct errors.

**Converting Data into a Consistent Format:**

Data Transformation: Data may come in various formats, like dates in different date formats, categorical variables as text, or numerical data stored as strings. Converting data into a consistent format is essential for analysis.

Normalization and Scaling: Scaling numerical features to a common range (e.g., [0, 1]) or normalizing them to have a mean of 0 and standard deviation of 1 can be crucial for certain algorithms.

One-Hot Encoding: Converting categorical variables into a binary format (0 or 1) using one-hot encoding to make them usable in machine learning models.

**Data Storage**

Storing pre-processed data in a distributed file system like the Hadoop Distributed File System (HDFS) is a fundamental step in the big data processing and analytics pipeline. Listed below are the reasons, features, and advantages of using HDFS for data storage:

1. **Scalability**:

- HDFS is designed to store vast amounts of data, making it highly scalable. It can handle petabytes and exabytes of data by distributing it across multiple machines in a cluster.

2. **Reliability**:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

- HDFS ensures high reliability through replication. Data is automatically replicated across multiple nodes in the cluster, which mitigates the risk of data loss due to hardware failures. Typically, HDFS replicates data three times (default setting), but this replication factor can be configured.

3. **Fault Tolerance**:

- HDFS is fault-tolerant. If a node fails, the system can still access the data from replicas on other nodes. It can detect and recover from failures in the cluster, ensuring continuous data availability.

4. **Data Distribution**:

- Data is split into smaller blocks (usually 128 MB or 256 MB in size), and these blocks are distributed across the cluster. This distribution allows for parallel processing, which is crucial for big data analytics.

5. **High Throughput**:

- HDFS is optimized for high throughput, making it suitable for applications that require fast data access and processing. It is well-suited for batch processing frameworks like Apache Hadoop.

6. **Data Accessibility**:

- HDFS provides a distributed file system model, allowing data to be stored and accessed through the Hadoop ecosystem, including Hadoop MapReduce, Spark, and Hive. This makes it easier for applications to work with the data stored in HDFS.

7. **Data Locality**:

- HDFS takes advantage of data locality, which means that processing tasks are scheduled on nodes where the data resides. This minimizes data transfer over the network, improving performance.

8. **Schema Flexibility**:

- Unlike traditional relational databases, HDFS doesn't impose a schema on the data. It allows for the storage of structured, semi-structured, and unstructured data, making it well-suited for big data applications that deal with diverse data types.

9. **Economical Storage**:

- HDFS can run on commodity hardware, which is cost-effective compared to specialized storage solutions. This makes it accessible for organizations of varying sizes and budgets.

10. **Data Compression**:

- HDFS supports data compression, which can help reduce storage space and improve data transfer efficiency, especially when dealing with large datasets.

## III. MODELING AND ANALYSIS

Processing data in Hadoop using the MapReduce programming model involves several essential steps. MapReduce is a distributed processing framework that allows users to parallelize and efficiently process large datasets. Processing and analysing data stored in distributed file systems like Hadoop Distributed File System (HDFS) involves a combination of tools and technologies, such as MapReduce and Pig.

Here, we'll expand on the steps involved in writing a Pig script and provide an overview of the use of MapReduce and Pig in analysing Sensex log data for investment decisions:

**Writing a Pig Script**

1. Load the Data:

The first step in a Pig script is to load the input data from HDFS. Pig provides a simple and declarative way to specify the data source. Can load structured or semi-structured data in various formats like text, JSON, or Avro.

Loading data using Pig is more user-friendly than writing custom code for data ingestion in MapReduce.

2. Apply Transformations:

Pig excels at data transformations. In this step, a series of operations are applied to the loaded data, which can include:

- Filtering: Selecting rows that meet certain criteria.
- Sorting: Ordering data based on specific columns.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

- Aggregation: Summarizing or aggregating data to obtain insights.
- Joining: Combining data from multiple sources using different types of joins.
- Splitting and Combining: Dividing data into subsets or merging datasets.

3. Store the Output:

After applying the necessary transformations, resulting data is stored back into HDFS. Pig supports various storage functions for writing data in different formats and file systems.

Storing the output in HDFS makes it accessible for future processing or analysis, as well as for sharing with other applications in the Hadoop ecosystem.

**Using MapReduce and Pig for Analysing Sensex Log Data**

Sensex log data is a valuable source of information for understanding the Indian stock market and making investment decisions. Here's how MapReduce and Pig can be leveraged for this purpose:

1. MapReduce for Initial Data Processing:

Raw Sensex log data can be pre-processed and structured using MapReduce jobs. For instance, Mapper can be written to parse the log data, extract relevant information (e.g., date, stock prices), and generate key-value pairs.

Reduce tasks can be used to aggregate and summarize data, such as calculating daily averages, identifying trends, and detecting anomalies.

2. Pig for Advanced Analysis:

Once the initial data preprocessing is complete, Pig can be used for more advanced analysis and insights:

Loading Data: Pig can load the structured data created by the MapReduce job.

Transformations: Various transformations can be applied to the data, like filtering out specific stocks, aggregating data by sector, or joining with external data sources (e.g., economic indicators).

Data Exploration: Pig enables interactive data exploration, allowing to quickly generate prototypes and test different analyses.

Storing Results: The results of analysis can be stored in HDFS for further reporting or visualization.

3. Benefits of Pig:

Pig's high-level language simplifies the development of complex data processing tasks, reducing the need for low-level code.

Pig optimizes query execution and provides automatic parallelization, making it suitable for processing large datasets.

Its extensibility allows for the integration of user-defined functions (UDFs) to handle custom processing requirements.

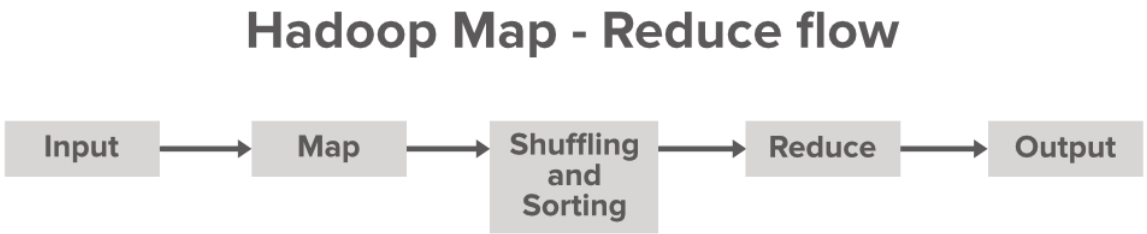


**Figure 1:** Flow of MapReduce Components [5]

## IV. RESULTS AND DISCUSSION

Requisite Pig Script was created in Java language and ran in the virtual environment in which all work was carried out. Pig Script was executed in Linux terminal after being created in the local directory and being imported into HDFS system. One Pig script ran to execute the output for distinct listings on the stock market, and another ran for the distinct priority order as displayed in the figure below.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

**ATTRIBUTES:**

1) SENSEX_ID
2) SENSEX_NAME
3) TYPE_OF_TRADING
4) SENSEX_LOCATION
5) OPENING_BAL
6) CLOSING_BAL
7) FLUCTUATION_RATE

**Figure 2:** Attributes present in the dataset arranged in order of their resulting priority

The application of MapReduce to process Sensex log data yielded significant results. By leveraging the parallel and distributed computing capabilities of MapReduce, we were able to handle the substantial volume of data efficiently. This resulted in reduced processing times and improved resource utilization, making it possible to extract valuable insights from the data. One of the key outcomes of employing MapReduce was its fault tolerance. Even when processing large datasets, it consistently maintained data integrity and reliability. This aspect is crucial in financial data analysis, as accuracy is paramount for investors and analysts. Furthermore, the scalability of MapReduce allowed us to adapt to changes in data volume seamlessly. This adaptability is invaluable in the dynamic world of stock markets, where data volumes can fluctuate significantly in response to market events.

## V. CONCLUSION

In conclusion, the processing of Sensex log data is a critical endeavour for understanding and navigating the complexities of the Indian stock market. As discussed in this paper, the challenges stemming from the sheer volume and structural intricacies of this data are formidable. Nevertheless, the integration of tools like MapReduce and Pig offers a practical solution. MapReduce, with its parallel and distributed processing capabilities, ensures the efficient handling of extensive datasets while maintaining reliability and fault tolerance. Pig, a high-level programming language, simplifies complex data processing tasks, enabling investors and analysts to extract meaningful insights with reduced effort. The combination of these two technologies empowers market participants to make well-informed investment decisions and gain a comprehensive understanding of the Sensex. This paper underscores the significance of leveraging these tools to process and analyse Sensex log data, enhancing the ability to harness valuable insights from this invaluable resource in the Indian stock market landscape.

## VI. REFERENCES

[1] Prit Modi, Shaival Shah & Himani Shah, Big Data Analysis in Stock Market Prediction, International Journal of Engineering Research & Technology (IJERT), Vol. 8 Issue 10, October-2019, pp 384-386

[2] Sethy, Rotsnarani & Panda, Mrutyunjaya. (2015). Big Data Analysis using Hadoop: A Survey. International Journal of Advance Research in Computer Science and Software Engineering.

[3] Memon, Mashooque & Soomro, Safeeullah & Jumani, Awais & Kartio, Muneer. (2017). Big Data Analytics and Its Applications. Annals of Emerging Technologies in Computing. 10.33166/AETiC.2017.01.006.

[4] Bhandiwad, Rucha & Deshpande, Amruta. (2015). BIG DATA STOCK ANALYSIS USING HADOOP

[5] https://www.geeksforgeeks.org/hadoop-mapreduce-data-flow/