Fraud Tolerance in Hadoop Data Processing

Dr. K.R. Viswa Jhananie[1]
Assistant Professor, Seshadripuram Academy of Business Studies, Kengeri Satellite Town, Bangalore-60

Abstract - Data which are extremely large are called as big data. Hadoop is an open source software where large amount of different data like structured, unstructured or semi structured can be stored and processed faster than the legacy databases with the support of more computing and powerful processing nodes.
Stored data can be processed for many reasons. This paper uses one of the data processing types called streaming to achieve fraud tolerance in banking sector using Apache Spark Streaming.

Keywords - Hadoop, distributed file system, manual data processing, mechanical data processing, electronic data processing.

I. INTRODUCTION

Data is analyzed constantly in Bigdata for producing efficient data which opens new opportunities and business models to the world. Hadoop is the technology which can store and process large-scale data efficiently by distributed framework [1]. Hadoop is used by most of the companies to handle big data. It processes data according to user's need. Adding a new node/machine to the Hadoop cluster is very simple and it is main advantage of it. Hadoop is capable of provides massive storage, enormous processing power and limitless concurrent tasks [2]. Since parallel processing is used, every node is processed in parallel. It uses a distributed file-system called **H**adoop **D**istributed **F**ile **S**ystem to store data, **Y**et **A**nother **R**esource **N**egotiator for cluster resource management and **M**ap**R**educe for batch processing [3]. In this paper, fraud tolerance is implemented using Hadoop.
This paper is carried over in this fashion. Different types of data processing are discussed in section-2. Real time systems and streaming are defined in section-3. Streaming processing components are discussed in section-4. Implementation is given in section-5. Future enhancement and conclusion is given in section-6 and section-7 respectively.

II. TYPES OF DATA PROCESSING

Manual Data Processing - It uses human for data entry process. The expected errors during the process is delay in processing [4]. It demands manual labor with high cost for equipment and supplies, rent, etc.,
Mechanical Data Processing - It reduces the errors count and process data in less time. Calculator is an example of mechanical data processing.
Electronic Data Processing - It is also is called as Information Services/ Management Information Services/Systems processes data were programs in an environment uses electronic communication. This method was created to replace the punched cards and paper reports.
Real Time Processing - RTP processes data in the stipulated time period. For example, when a customer purchases a product in online portal, it can suggest relevant products for further purchase in short duration.
Batch Processing - In this, data are collected, then entered, then processed and the results are produced by separate programs respectively. It makes large data processing efficient. Example: Analysis monthly or daily purchases of a super market.

III. REAL TIME VS STREAMING

Real time systems give result in tight deadlines e.g. Televisions where continuous computation (streaming) are happens as data flows through the source. No stipulated deadlines are defined. The data is processed immediate once it comes in. For example, words count present in a paragraph. The constraint in this processing is the output rate should be faster or equal to the input rate. Input needs enough memory to store inputs while processing.

IV. HADOOP BATCH, REAL TIME AND STREAMING PROCESSING COMPONENTS

Apache Spark - Spark is a batch processing framework with stream processing and Machine Learning analytics capability. It can be programmed in Java, Scala, Python, and R. It uses Hadoop YARN for cluster management and HDFS for storage.
It uses a read-only multiset of data items called **R**esilient **D**istributed **D**ataset (data structure) which is distributed over the entire Hadoop cluster. It can access different data sources like HDFS, Cassandra, HBase etc., for distributed storage.
The basic I/O functionalities are handled by the foundation called Spark Core. The restricted forms are shared variables, broadcast variables and accumulators [5]. Other elements of Spark Core are: Spark SQL to manipulate DataFrames. Spark Streaming for streaming and batch analytics. Spark MLlib for machine-learning analytics and GraphX for distributed graph processing.
Apache Storm - Storm provides distributed real-time processing. It is designed as a topology. In which graph verticals are Spouts and bolts. Small and discrete operations are composed into a topology to transform data as a pipeline.
Unbound data that continuously arrive to the storm as streaming. Sprouts are at the edge of the topology as data stream sources. Bolts does process and applies an operation

to those data streams which are coming in. The graph edges (streams) move data from one node to another node [6]. Bolts and Sprouts allow batch and distributed processing of real-time streaming data.

Samza - Samza provides real-time and asynchronous framework to do distributed stream processing [7]. Transformations create new streams for other components to consume it without any changes [8]. It uses Apache Kafka for messaging and YARN for fault tolerance and resources management.

Flink - This framework provides stream processing and batch tasks management to achieve high-throughput and low-latency. Can be implemented in Java and Scala. Its runtime allows the execution of batch processing and iterative algorithms. Event-time processing and state management are also support.

Streams, sources, sinks and operators are components of Flink. Streams are immutable. Operators are used to create other streams. Sources are the entry points. Streams flow out of the Flink system by Sinks.

Spark Stream Processing - It provides high-throughput, scalable and fault-tolerant processing of live data (streams). Streams can be ingested from sources like Kafka, Flume, TCP sockets etc., Complex algorithms can also be used for data processing. Processed data can be written out to filesystems and/or databases and then to dashboards (graphical reports) from it.

V. IMPLEMENTATION

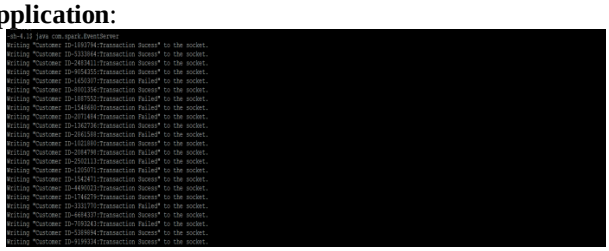
The implementation is carried over in four steps. They are

1. Starting data generating server to generate events.
2. Running Streaming application to connect to data generator.
3. Push events from data generator to Streaming application.
4. Receive events and process as needed to act on fraud transactions.

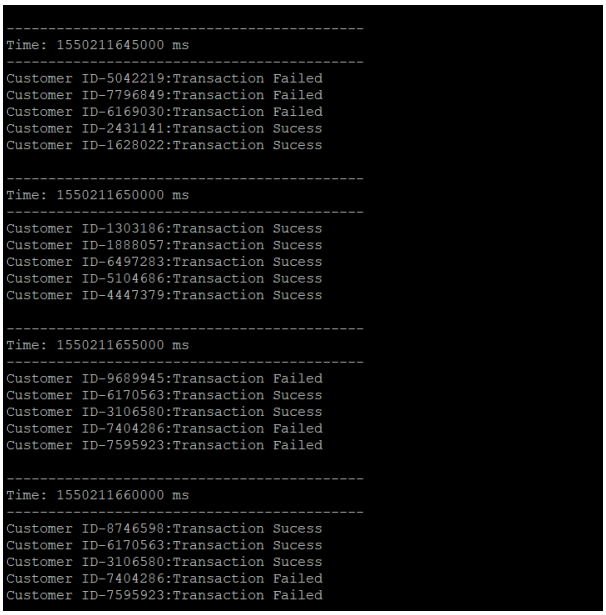
Starting data generating server to generate events:

Running Streaming application to connect to data generator:

Push events from data generator to Streaming application:



Receive events and process as needed to act on fraud transactions:



VI. FUTURE ENHANCEMENT

1. Ingest data from other sources like Kafka, Flume, Kinesis
2. Spark's MLli and GraphX can be used.

VII. CONCLUSION

In web world, fraud tolerance has become key need for any business. Because of high volume of data (like bank transactions), Hadoop is needed to process huge data to identify frauds immediately by real or near real time or streaming. This paper showed how the Fraud tolerance can be achived with Hadoop.

VIII. REFERENCES

[1]. B. Saraladevi, N. Pazhaniraja, P. Victer Paul, M.S. Saleem Basha and P. Dhavachelvan, "Big Data and Hadoop – A Study in Security Perspective", Elsevier, Procedia Computer Science 2015, pg – 596-601.

[2]. Rahul Beakta, "Big data and Hadoop: A Review Paper", RIEECE, Vol-2, issue-2, 2015.

[3]. Sara Landset, Taghi M. Khoshgoftaar, Aaron N. Richter and Tawfiq Hasanin, " A Survey on Open Source Tools for Machine Learning with Big Data Hadoop Ecosystem", Journal of Bigdata, 2015.

[4]. Revathi V, Rakshitha K.R, Sruthi K and Guruprasaath S, "Big Data with Hadoop – For Data Management, Processing and Storing", IRJET, Vol-4, Issue-8, 2017.
[5]. Varsha B, "Survey Paper on Big Data and Hadoop", International Research Journal of Engineering and Technology", Vol-3, Issue-1, 2016.
[6]. Apache storm official website. http://storm.apache.org/
[7]. Apache Samza official website. http://samza.apache.org/
[8]. Dr.Sanjay Srivatsa, Swami Singh, Viplav Mandal, "The Big Data Analytics with Hadoop", International Journal of Research in Applied Science and Engineering Technology, Vol-4, Issue- 3, 2016.