Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018)
IEEE Xplore Compliant - Part Number: CFP18BAC-ART; ISBN:978-1-5386-1974-2

A Streamlined Approach for Real-Time Data Analytics

Shruti Arora
Department of Computer Sc. & Engg.,
Thapar Institute of Engineering and Technology,
Patiala, INDIA
shrutiarora1103@gmail.com

Rinkle Rani
Department of Computer Sc. & Engg.,
Thapar Institute of Engineering and Technology,
Patiala, INDIA
raggarwal@thapar.edu

***Abstract*—Complex and diverse data is continuously being generated every microsecond because of computational intelligence in every field. Leveraging the huge amount of sensory information is a key issue to realize the IoT solutions in many areas. Traditional approaches reported in the literature for ingesting and processing data in real-time become in-efficient sometimes or may result in inadequate results. The faster one can manipulate information from data, the greater the value and cost of operational data become. This paper explains the cost-efficient method of processing and analyzing data in real time along with tools and techniques used for certain applications.**

Keywords—Internet of Things, Stream Data Analysis, Event Hub, Storage Blobs.

I. INTRODUCTION

The term "Big Data" is coined for large datasets that become cumbersome to process and manage using traditional approaches such as Relational Database Management System (RDBMS), Data warehousing, etc. Everything connected through internet gives rise to the new field of research which is 'Internet of Things (IoT)'. The connected product engineering becomes development area in industries. The industries could be benefitted in realizing the trends and making amendments accordingly if the data is available to them in real-time. To handle the massive amount of data various novel approaches have been implemented but some approaches have real-time data processing, managing, storing or analyzing constraints. With real-time in mind, applications such as click-stream [1], intrusion detection system (IDS), financial market prediction and fraud detection [2] come into the picture.

This paper aims a streaming data analytics solution for the various applications and also literature reported traditional frameworks [3] and their limitations. The paper focuses on platforms and techniques for modeling, analysis, and storage of data streams. The paper categorizes the available platforms and tools for capturing and processing real-time data, proposed approach for the specific application of fraud detection, and analysis output [4]. This is implemented by referring to published research papers and contributions in the field in recent years.

Firstly, the background of Big Data, data stream processing [5] is discussed for the introduction of data science in recent years. Then the novel platforms, technologies, and tools available for the processing and storage of data in real-time are explored. Section IV addresses the proposed approach for easy and efficient processing and storage of real-time data.

II. BACKGROUND

A. Big Data and its challenges

Big Data is categorized into structured and unstructured data. It is not about the volume of data that matters to any Business but what really matters is how much data is beneficial for the strategic growth and business moves. The characteristics [6] of Big Data termed as three V's are Volume, Variety, and Velocity with new V's recently added is Value and Veracity. Many researchers have worked on extracting useful information (value), feature extraction of useful attributes from a variety of data and the quality of captured data for analysis (veracity).

There are many open challenges [7], [8] in this area of research such as real-time processing, storage, timeliness of data available to analytics for decision making. As Big Data covers structured and unstructured data [9], it becomes difficult to store and display data for the applications which require near real-time accuracy.

B. Traditional vs Efficient Approaches

Legacy systems such as Batch Processing [10] for the processing of huge unstructured data turned out to be inefficient for the business analytics to do decision modeling for their business in real-time. Today developers are analyzing terabytes and petabytes of data on the rolling window. Batch processing is the traditional approach for processing large data whereas Stream Processing is the latest research area which is efficient to handle data for the applications which require analysis to be carried out in real-time or near real-time. Some of the key differences between Batch processing and Stream Processing are coined below:

Batch Processing: It is carried out where the processing of blocks of data those have already been stored over a period of time. For example, processing all the records that have been generated by a major firm in a stipulated period of time. The data may contain millions of records for a day that can be stored as a file or record. The particular file will undergo processing at the end of the day for various analyses that firm wants to do and thus would turn out to be time inefficient approach. Hadoop MapReduce [10] is the best framework that is built for Batch Processing.

Stream Processing: It is the key if one wants to process and get analytics reports in near-real time. Data ingestion, processing, storage and analytics reports are displayed in real-time. It is an approach that is crucial in applications such as Fraud Detection. The wide range of available platforms [11] for Stream Processing is Spark Streaming, Flume, Kafka and Azure Stream Analytics.

III. TOOLS AND TECHNOLOGIES

Big Data processing platforms offer a pool of services such as Ingesting, processing, storage, and analytics. The traditionally available platforms are Hadoop, Spark, Storm, Flume, and Kafka. All these platforms are open source, fault-tolerant and scalable. Hadoop is mostly built for applications where voluminous data processing is required and time-constraint is not considerable [12]. Spark, Storm, Flume, Kafka, on the other hand, are built for real-time data processing and are implemented in JVM based programming languages such as Java, Scala, and Clojure. All these platforms are considered efficient for stream processing. This paper aims to complement existing analytics with streaming data with one more platform which offers all services for real-time streaming data processing with an integration of analytical dashboard which has ease of use and interactive interface available. This stream processing platform is developed by Microsoft and is known as Azure Stream Processing framework with Power BI [13] analytical dashboard integration.

A. Azure

Azure Stream Analytics is a cloud-based and efficient streaming data management service for ingesting high-velocity data streaming from devices, sensors, applications, Web sites, and other multiple data sources. Unlike other stream processing, it has a support for a SQL-like query language that works over dynamic data streams. It makes querying data as simple as querying on static data. Azure has enhanced dashboard added services and user-friendly notification alerts on the interface as well as mobile devices. It is very user-friendly and economic. Azure services can be used in scenarios where data-processing and analysis are required in real-time. Fraud detection, identity-theft protection, optimizing the allocation of resources (think of an Ola-like transportation service that sends drivers to areas of increasing demand before that demand peaks), click-stream analysis on Web sites, recommendation systems in retail-shopping and countless others. The ability to process the data on the fly, reliable support services, fault tolerance, low-cost cloud storage services makes Azure one of the best platform for Stream data Analytics.

a) The Architecture of Azure Stream Analytics

Stream Analytics starts with a streaming data source. The data is ingested into Azure from sensor or device producing high volume data streams using an Azure event hub or IoT hub. The data is pulled from a data store known as Azure Blob Storage. The stream is examined by creating a Stream Analytics job that specifies from where the data is obtained. The job also specifies a transformation; how to look for data, patterns or relationships. The transformation operations for Stream Analytics are taken care by a SQL-like query language that helps to sort, filter, aggregate and join streaming data over a time frame. Azure has integration with Power BI tool for analytical reporting framework. It has an interactive and business-friendly interface which provides ease of analysis to business decision-makers.

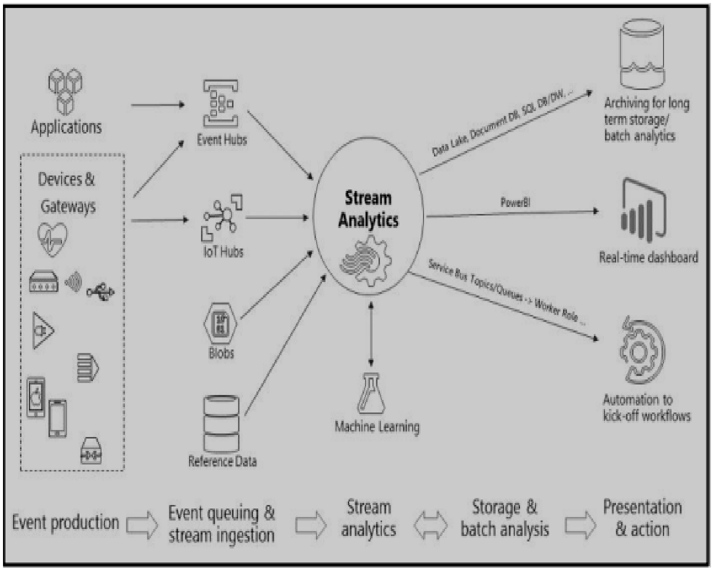


Figure 1. The Architecture of Azure Stream Analytics

b) Components of Azure Stream Analytics Platform

Azure has five main components: Event production, event queuing and stream ingestion, stream analytics, storage and presentation and action [14].

Event production: The devices and gateways produce data streams, which are generally transactions or dynamic rate of change of values every microsecond. The data produced is very complex and has variable dimensions which further needs fabrication for the processing to take place, which is further taken care by Event or IoT Hubs. Azure Event Hubs is a hyper-scale telemetry ingestion service which collects, transforms and stores millions of events.

Event Queuing and Stream Ingestion: Advanced Message Queuing Protocol (AMQP) [15] is used by Event hubs to enable interoperability and compatibility across platforms, while also supporting HTTPS for data ingress. Azure Services including Stream Analytics and Virtual Machines can be integrated while building rich server-less workflows [16].

Stream Analytics: Stream Analytics on direct connection with Azure IoT Hubs and Azure Event Hub for stream ingestion, and Azure Blob for storage service facilitates real-time analytics on multiple IoT or non-IoT streams of data.

Storage: Azure Storage is highly secure, scalable, durable and flexible storage cloud service managed by Microsoft. Azure provides 3 storage services [17] i.e. Blob, File and Queue storage.

- Blobs service are basically simple files which may be .doc, .xls, log files, etc. which are stored in folders known as containers and the whole setup is called blob storage which is accessible in a distributed system through URLs, REST interface [18] or Azure Software Development Kit (SDK) storage libraries.
- Files Service uses standard Server Message Block (SMB) protocol for accessing largely available network file.
- Queue service is used for storage and retrieve of messages. Queue messages allow up to 64 KB queue in size, and a queue can accommodate millions of messages. This service is generally used for asynchronous storage of lists of messages.

All the storage services have encryption feature for secure storage of data. The encryption methods involved are Encryption at rest and Client-side Encryption [19].

Presentation and Action: Azure has an integration of Power BI dashboards for displaying the analysis with various statistical features. It is built mainly for live reporting of events occurring in microseconds. Business Analyst or the client's primary requirement is to get the real-time reports and is not concerned about what programming model is providing the required output.

c) Applications

The Azure framework very efficiently handles real-time applications such as fraud detection, click-stream analysis, intrusion detection and many others. This paper focuses on fraud detection of calls in the telecommunication server industry. To accomplish this task a telecom server is built for research purposes in Node.js programming language, which populates the call record data when the script is executed.

B. Power BI

Power BI is a compilation of connectors, apps, and software services that work together to amplify unrelated sources of data into coherent and visually immersive. Whether the data is a collection of cloud-based and on-premises hybrid data warehouses or a simple Excel spreadsheet, Power BI is used for connection of data sources and visualization. It is an online SaaS (*Software as a Service).*

IV. IMPLEMENTATION

The area of Fraud detection in real-time is quite complex as detection and inquisitive action is required on the fly as it a critical case. The fulfillment of an application is achieved by building a stream analytics job which ingests data from IoT Event Hub generated by Node.js telecommunication server script.

A. Proposed Process Model

The proposed process model depicted in Fig 2. aims at fraudulent call detection and displaying the analysis report on Power BI interface. The design is very simple, low cost and efficient. The data generated by telcodatagen.js node script is the dataset of calls captured in a particular timeframe. The data is ingested into event hub after being authenticated from authentication server with the connection string. The event hub passes the data to a streaming job which processes it and stores into Azure Storage. The SQL-like query language support facilitates querying of fraudulent calls. The query once fired can be saved and invokes whenever a streaming job is started. The notification and alerts are real-time for each and every action taking place for the Azure streaming job. The observations and results are displayed below for this fraud-detection application.

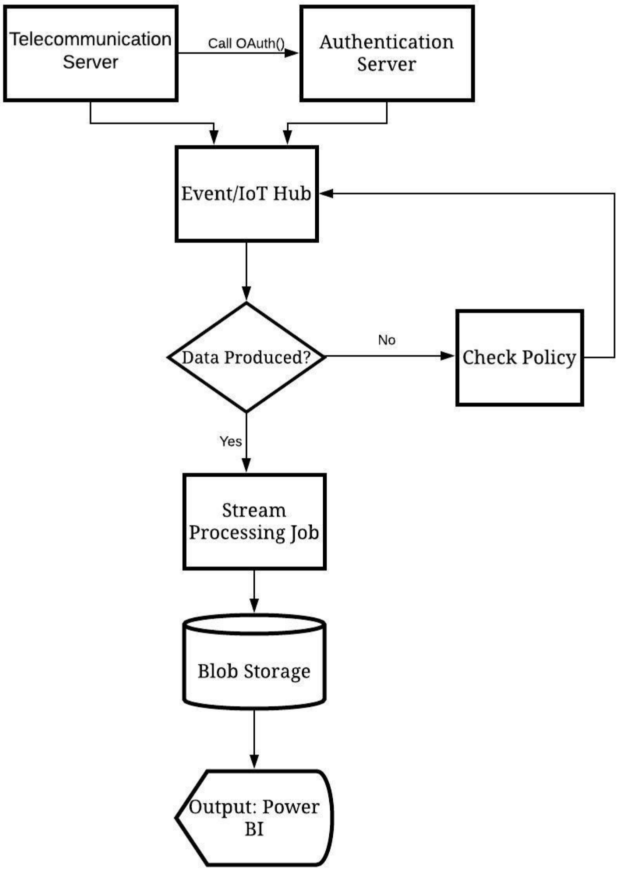


Figure 2. Proposed Process Model for Fraudulent Call Detection

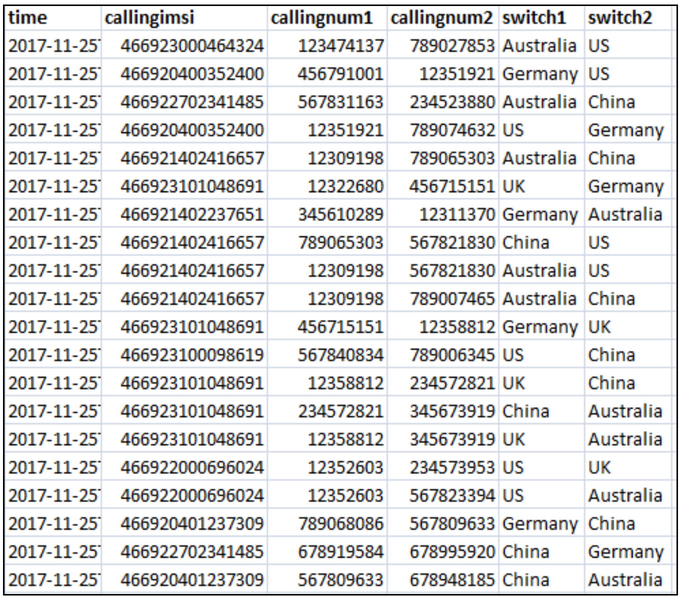
B. Dataset and Result

Table 1. depicts the attributes of the telecommunication server data from which fraudulent calls were detected [20].

The data produced by telecommunication server is displayed in a snapshot in Fig. 3 where the application is configured as:

- Number of CDRs per hour.
- SIM Card Fraud Probability
- Duration in hours: The number of hours that the application should run.

The data ingested from event hub was processed by streaming job and the data set was set to be sampled from sliding window of 10 minutes so that the call stream data from a particular window frame is queried for fraudulent calls and displayed on Power BI analysis graph. An optimized self-join query on dataset displays all the fraudulent calls in Azure output interface as well as in Power BI integration in the form of statistical graphs as shown in Fig. 4



time	callingimsi	callingnum1	callingnum2	switch1	switch2
2017-11-25	466923000464324	123474137	789027853	Australia	US
2017-11-25	466920400352400	456791001	12351921	Germany	US
2017-11-25	466922702341485	567831163	234523880	Australia	China
2017-11-25	466920400352400	12351921	789074632	US	Germany
2017-11-25	466921402416657	12309198	789065303	Australia	China
2017-11-25	466923101048691	12322680	456715151	UK	Germany
2017-11-25	466921402237651	345610289	12311370	Germany	Australia
2017-11-25	466921402416657	789065303	567821830	China	US
2017-11-25	466921402416657	12309198	567821830	Australia	US
2017-11-25	466921402416657	12309198	789007465	Australia	China
2017-11-25	466923101048691	456715151	12358812	Germany	UK
2017-11-25	466923100098619	567840834	789006345	US	China
2017-11-25	466923101048691	12358812	234572821	UK	China
2017-11-25	466923101048691	234572821	345673919	China	Australia
2017-11-25	466923101048691	12358812	345673919	UK	Australia
2017-11-25	466922000696024	12352603	234573953	US	UK
2017-11-25	466922000696024	12352603	567823394	US	Australia
2017-11-25	466920401237309	789068086	567809633	Germany	China
2017-11-25	466922702341485	678919584	678995920	China	Germany
2017-11-25	466920401237309	567809633	678948185	China	Australia

Figure 3. Data produced by Telecommunication Server

TABLE I. ATTRIBUTES OF DATASET

Record	Definition
CallrecTime	The call start timestamp
SwitchNum	The telephone switch used to connect the call. The switches are strings that represent the country (US, China, UK, Germany, or Australia)
CallingNum	The contact number of the caller
CallingIMSI	The International Mobile Subscriber Identity (IMSI) which is the Unique identifier of the caller
CalledNum	The contact number of the recipient
CalledIMSI	International Mobile Subscriber Identity (IMSI) which is the unique identifier of the call recipient

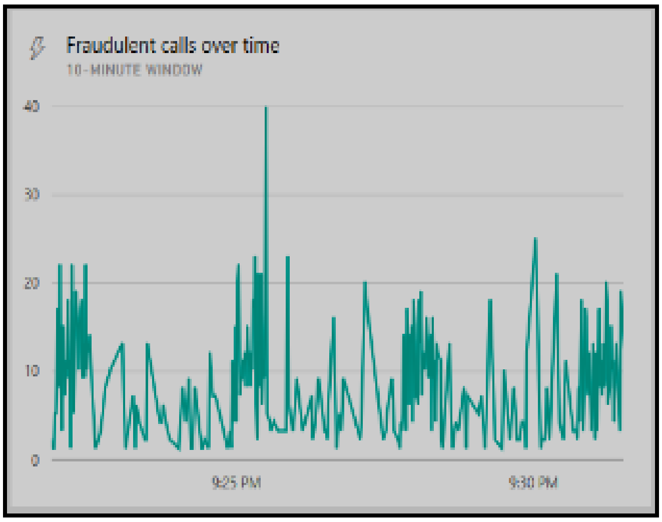


Figure 4. Output of Fraudulent Calls in Power BI desktop

V. CONCLUSION

This paper basically aims to explain the framework available for stream processing of big data. Although many other platforms such as Hadoop, Spark, Storm, Kafka, and Flume are available, Azure is found to be more user-friendly and efficient platform for on the fly analysis and storage. The Data Storage Lake and many other storage services like Blob, File, and Queue provide storage for continuously flowing multidimensional data. Real-time applications such as Fraud Detection, Click-Stream Analysis, and Stock Market prediction can be analyzed using the Azure framework. This paper is mainly aimed at Fraud Detection of calls where the data is generated by telecommunication server. Azure's interface and notification services make it the most user-friendly interface platform. Hadoop and other platforms do not have SQL-like query language support, they have database systems like Hive, NoSQL, MongoDB etc. whereas Azure has SQL query support for transformations on the acquired dataset. An overall architecture and process model have been outlined in this paper.

REFERENCES

[1] S. Senecala, J. Kalczynski, J. Nantel, "Consumers' decision-making process and their online shopping behavior: a clickstream analysis," *in Journal of Business Research*, vol. 58, no. 11, pp. 1599-1608, 2004.

[2] M. Stonebraker, S. Zdonik, "The 8 requirements of real-time stream processing," *in SIGMOD Record*, vol. 34, no. 4, pp. 42-47, 2005.

[3] S. Petrovic, M. Osborne and V. Lavrenko, " Streaming first story detection with application to Twitter," *in the Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181-189, 2010.

[4] M. Kiran, P. Murphy, I. Monga, J. Dugan, "Lambda architecture for cost-effective batch and speed big data

processing," *in the Proceedings of IEEE International Conference on Big Data*, pp. 2785–2792, 2015.

[5] D. M. C. Dissanayake, K. P. N. Jayasena, "A cloud platform for big IoT data analytics by combining batch and stream processing technologies," *in the Proceedings of National Information Technology Conference (NITC)*, pp. 40-45, 2017

[6] S. Atta, B. Sadiq, A. Ahmad, S.N. Saeed, E. Felemban, "Spatial-crowd: A big data framework for efficient data visualization, " *in the Proceedings of IEEE International Conference on Big Data*, pp. 2130-2138, 2016.

[7] C.L.P. Chen, CY. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, " *Journal of Information Sciences*, vol. 275, pp. 314-347, 2014.

[8] S. Kaisler, F. Armour, J. A. Espinosa and W. Money, "Big Data: Issues and Challenges Moving Forward, *" in the Proceedings of 46th Hawaii International Conference on System Sciences*, pp. 995-1004, 2013.

[9] H. Baars, H.G. Kemper, "Management Support with Structured and Unstructured Data: An Integrated Business Intelligence Framework" *in Information Systems Management*, pp.132-148, 2008.

[10] R.Uzsoy, "Scheduling a single batch processing machine with non-identical job sizes," *International Journal of Production Research*, vol. 29, no. 1, pp.1615-1635, 2007.

[11] S. Chintapalli *et al*., "Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming," *in the Proceedings of IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Chicago, pp. 1789-1792, 2016.

[12] X. Lin, P. Wang and B. Wu, "Log analysis in cloud computing environment with Hadoop and Spark," *in the Proceedings of 5th IEEE International Conference on Broadband Network & Multimedia Technology*, pp. 273-276, 2013.

[13] C. Pirnau, N.I. Marinescu, L.D. Ciocardia, "Business intelligence development with Power BI applied in nonconventional technologies," *in Nonconventional Technologies Review/Revista de Tehnologii Neconventionale*, vol. 21, no. 4, pp.18-26, 2017.

[14] P. Paakkonen, D. Pakkala, "Reference architecture and classification of technologies, products and services for Big Data systems," *Big Data Research*, vol. 2, no.4, pp. 166-186, 2015.

[15] S. Vinoski, "Advanced Message Queuing Protocol," *IEEE Internet Computing*, vol. 10, no. 6, pp. 87-89, 2006.

[16] Li, Jin. "Serverless peer-to-peer multi-party real-time audio communication system and method." U.S. Patent No. 7, pp. 460-495, 2008.

[17] B. E. A. Calder, "Windows azure storage: A highly available cloud storage service with strong consistency," *in the Proceedings of 23rd ACM Symposium of Operating System Principles*, pp. 143–157, 2011.

[18] Haupt, Florian, et al. "A framework for the structural analysis of REST APIs" *in the proceedings of IEEE International Conference on Software Architecture (ICSA),* pp.55-58, 2017.

[19] J. Xu, E. Chang, J.Zhou, "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage, " *in Proceedings of the 8th ACM SIGSAC Symposium on Information, computer and communications security*, pp.195-206, 2013.

[20] V. Jain, "Perspective analysis of telecommunication fraud detection using data stream *analytics* and neural network classification based data mining," *International Journal of Information Technology*, vol.9, no.3, pp.303–310, 2017.