Data Solutions to Meet CPRA Regulations using StreamSets

Madhusudan S C
B.E Student, Department of ECE
RV College of Engineering,
Benagluru, India

Anusha L S
Assistant Professor
Department of ECE
RV College of Engineering

Abhilash Desai
Lead Software Engineer
Epsilon, Bengaluru
India

***Abstract*— The ever-increasing volume and complexity of data in the digital age pose significant challenges for organisations in terms of efficient data processing and privacy compliance. This paper explores utilisation of streamsets, a robust data integration platform, to address these challenges within the context of the Hadoop ecosystem. The primary objective of this research is to create two pipelines for data loading and data aggregation to facilitate efficient data loading, processing, and compliance with the California Consumer Privacy Act (CCPA) and the California Privacy Rights Act (CPRA). The findings demonstrate that streamsets simplifies the design and execution of pipelines, enabling seamless usage. By adopting streamsets, organizations will be able to optimize data loading processes, harness power of their Hadoop infrastructure, and ensure to comply with privacy regulations in the era of big data.**

***Keywords*— CPRA, StreamSets, Big data, Hadoop, Pipeline**

I. INTRODUCTION

This In today's digital age, data is a ubiquitous and essential element of modern life. Data refers to any piece of information that will be recorded and stored electronically. It includes everything from text and numbers to images, video, and audio. The amount of data being generated and stored has grown exponentially in recent years, leading to the concepts of big data. Big data refers to extremely large and huge datasets that are too complex and dynamic to be easily managed and analysed using traditional data processing tools. Despite the challenges posed by big data, it’s a powerful resource that organizations will use to gain insights into their operations, customers, and competitors. As organisations gather and store more data, they must ensure that it is managed and protected in accordance with applicable laws like the California Consumer Privacy Act (CCPA) and the California Privacy Rights Act (CPRA), and build platform that can handle loaded data using tools like streamsets.

CCPA is a state law that was enforced on January 1, 2020. CPRA is an amendment to the existing CCPA revisions. These laws give California residents right to know what personal data is being collected and whether it is sold or disclosed. Residents will also be able to access their personal information (PI) and opt not to sell or completely delete it. Streamsets is a data integration platform that can help organisations meet CCPA and CPRA regulations by giving solutions for data management and protection. Using streamsets data collectors, pipelines can be created that can load available data files into platforms such as Hadoop. A pipeline describes the flow of data from the origin system to the destination system and defines how to transform the data along the way. A pipeline is made of stages, that are divided into three types: origins, processors, and destinations/executors. By leveraging specialised tools and platforms like streamsets, organisations will not only comply with regulations like CCPA and CPRA but also gain greater control and visibility over their big data and make better-informed business decisions.

Wong et al. [1] examines how large technology companies address privacy regulations like the General Data Protection Regulation (GDPR) and the CCPA in their annual regulatory filings with investors. Using a qualitative document analysis of FORM 10-Ks from nine major technology companies, the authors compared annual reports to identify ways that GDPR and CCPA are considered business risks. Paper [2] discusses the new amendments to the CCPA through CPRA regulations, providing a new definition of sensitive personal information and the processing of such data. It summarised the new rules that can be utilised by California residents. Quinto et al. [3] provide insight on data ingestion and processing into Kudu from local directories in batch and real-time. Third-party applications such as streamsets data collectors, Cask Data Application Platform, Pentagon data integration, and Talend data integration, or own applications using native tools such as Apache Spark and Kudu’s client APIs, can be used to build a pipeline. Paper [4] shows the use of Cloudera distribution to manage data from web service clients. Technologies like streamsets, Kafka, and Cloudera were used for building the model.

From the literature review, it was found that there exists no appropriate streamsets automation pipeline that could convert the datafiles into a readable table. So the objective of this project is to design a data loading pipeline that fetches the data zip files from different sources in the Hadoop Distribution File System (HDFS) and converts them into a database that will be stored in the Hadoop database. Another pipeline called the data aggregation pipeline is created to track the data loaded by creating a tracking ID in the Oracle database and updating the ID if that particular table is used to generate reports for business use.

II. METHODOLOGY

To store and manage all the big data files in Hadoop User Experience (HUE), an open-source web interface is built, consisting of an HDFS file browser, a query editor like Hive or Impala, and a Hadoop database to store the tables. Figure 1 shows an overview of the process carried out. Initially, data files in zip (.gz) format from different sources are loaded into the HDFS file browser. The data loading pipeline created in streamsets fetches the zip file and sends it into a set of shell and python scripts. These scripts extract data, convert it into tables, and store it in a Hadoop database. For tracking these activities, a few databases are created using Oracle Database, the ER model for the same with table names and attributes is shown in Figure 2. As the scripts start extracting the data, a row with a unique DATA_AGGREGATION_ID is created in the DATA_AGGREGATION table with ACTIVE_FG as N and LOAD_STATUS as R.

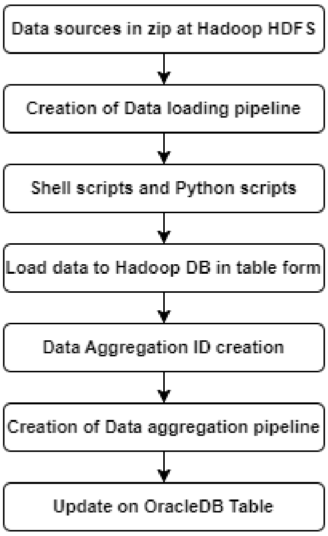
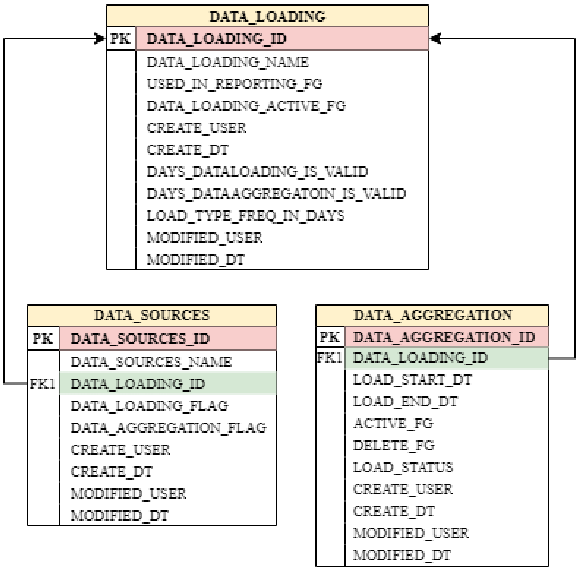

Data sources in zip at Hadoop HDFS
Creation of Data loading pipeline
Shell scripts and Python scripts
Load data to Hadoop DB in table form
Data Aggregation ID creation
Creation of Data aggregation pipeline
Update on OracleDB Table

Figure 1: Flow of processing file from data sources



	DATA_LOADING
PK	DATA_LOADING_ID
	DATA_LOADING_NAME
	USED_IN_REPORTING_FG
	DATA_LOADING_ACTIVE_FG
	CREATE_USER
	CREATE_DT
	DAYS_DATALOADING_IS_VALID
	DAYS_DATAAGGREGATOIN_IS_VALID
	LOAD_TYPE_FREQ_IN_DAYS
	MODIFIED_USER
	MODIFIED_DT

	DATA_SOURCES
PK	DATA_SOURCES_ID
	DATA_SOURCES_NAME
FK1	DATA_LOADING_ID
	DATA_LOADING_FLAG
	DATA_AGGREGATION_FLAG
	CREATE_USER
	CREATE_DT
	MODIFIED_USER
	MODIFIED_DT

	DATA_AGGREGATION
PK	DATA_AGGREGATION_ID
FK1	DATA_LOADING_ID
	LOAD_START_DT
	LOAD_END_DT
	ACTIVE_FG
	DELETE_FG
	LOAD_STATUS
	CREATE_USER
	CREATE_DT
	MODIFIED_USER
	MODIFIED_DT

Figure 2: ER Diagram of Tables created in OracleDB

Also, the load start and end time are recorded. If the data loading is successful, scripts should set the LOAD_STATUS to S else N. After the data loading pipeline finishes the job, a data aggregation pipeline is created that fetches the rows from the Oracle database that have ACTIVE_FG as N and LOAD_STATUS as S and runs an update query that makes ACTIVE_FG as Y. As shown in Figure 2, the DATA_LOADING table acts as the master table that will be used to insert the names of the available data sources in the HDFS file browser and track the validity of data loading and aggregation pipeline. The DATA_SOURCES table is created to insert data source names referencing the DATA_LOADING table and check if data loading and data aggregation are created or not for that particular data source using flags.

A. Data Loading Pipeline

The Data Loading Pipeline implemented using streamsets is shown in Figure 3. Hadoop FS Standalone block is configured to Hadoop HDFS using certain credentials and by giving file path and file pattern, data files in zip format are fetched from the HDFS. The file is sent to the shell block through a delay block where shell and python scripts are executed on the incoming zip file leads to creation of table in Hadoop database.

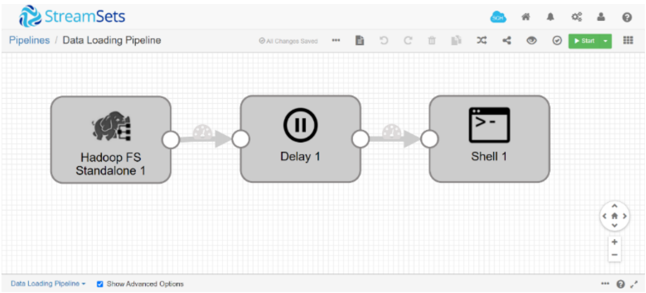


Figure 3. Data Loading Pipeline in Streamsets

As the data starts loading, a row with unique DATA_AGGREGATION_ID is inserted into DATA_AGGREGATION table in oracle database with ACTIVE_FG as F and LOAD_STATUS as R. After successful load the LOAD_Status is made S else N.

B. Data Aggregation Pipeline

As the data loading pipeline finishes the job and creates the DATA_AGGREGATION_ID, data aggregation pipeline picks this particular ID from DATA_AGGREGATION table. Figure 4 shows the design of pipeline in streamsets. JDBC query consumer is connected to oracle databasee using username and password. SQL query is written for table data aggregation that fetches DATA_AGGREGATION_ID and DATA_AGGREGATION_FLAG for the particular data file with DATA_LOADING_ID from the database that has ACTIVE_FG as N and LOAD_STATUS as S. If the value of DATA_AGGREGATION_ID is unique, the record duplicator sends it to the stream selector and stores it in memory, if repeated then discards it. The stream selector check the value of DATA_AGGREGATION_FLAG, that indicates if data aggregation pipeline is required for the particular data loading pipeline.

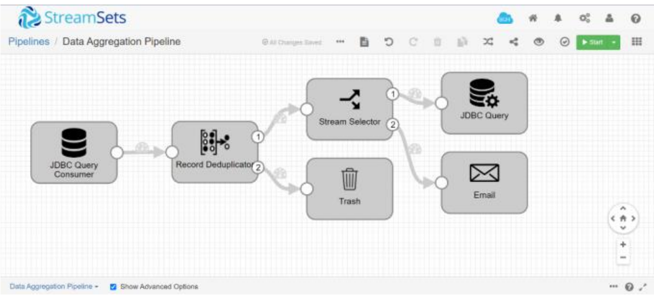


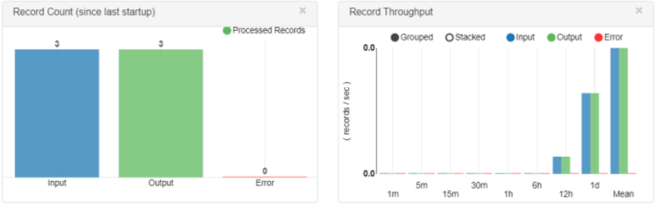
Figure 4. Data Aggregation Pipeline in Streamsets

If the DATA_AGGREGATION_FLAG is S, then JDBC query is executed the updates the ACTIVE_FG as S for the detected DATA_AGGREGATION_ID at the beginning. If the flag is N, a email is sent to user indication the ACTIVE_FG is not updated. Everytime a data file comes from the same datasource or different and starts loading into Hadoop database, DATA_AGGREGATION_ID is created in Oracle database to track that particular file and use the content of table for business purpose.

III. RESULTS AND DISCUSSION

In this study, streamsets is employed as a robust data integration platform for managing and processing big data in compliance with CCPA and CPRA regulations. The implementation of streamsets proved successful in achieving efficient data management and regulatory compliance. Streamsets demonstrated its ability to handle large volumes of data from various sources. It facilitated seamless integration with Hadoop HDFS, allowing for the data laoding and transformation of diverse data sets. The platform's data pipeline creation capabilities enabled the smooth processing and movement of data, ensuring streamlined operations throughout the compliance workflow.

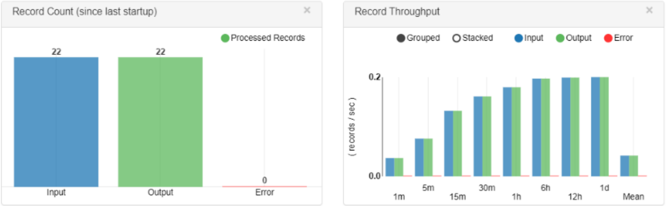
For the data loading pipeline, Figure 5 depicts the overall results, showing various parameters. The record count for pipeline is represented in Figure 5(a), where three records are processed and there are no errors. Record throughput plotted against time can be observed in Figures 5(b) showing efficiency of data loading pipeline.



(a) (b)

Figure 5: Data loading pipeline results (a) Record count (b) Record Throughput

The results for the data aggregation pipeline is shown in Figure 6. The twenty-two records that are fetched in the pipeline are processed without any error and depicted in Figure 6(a). Figure 6(b) shows the record throughput of the pipeline indicating individual time and average time taken by records.



(a) (b)

Figure 6: Data loading pipeline results (a) Record count (b) Record Throughput

IV. CONCLUSION

The adoption of streamsets for data management and compliance has yielded several key benefits. Firstly, the platform's user-friendly interface and comprehensive features enabled organisations to design and execute complex data pipelines with ease. This enhanced efficiency and reduced the effort and time for data integration tasks. The two pipelines designed work efficiently for every the test cases and deployed to use in the organisation.

In conclusion, this paper examined the utilisation of streamsets as a powerful data integration platform for managing and processing big data in compliance with the CPRA regulations. The results demonstrated the effectiveness of streamsets in enabling efficient data management and achieving regulatory compliance. Streamsets proved to be a valuable tool for handling large volumes of data from diverse sources, offering seamless integration with systems like Hadoop HDFS, Hadoop databases, and Hive. The platform's data pipeline creation capabilities facilitated smooth data processing and movement, ensuring streamlined operations throughout the compliance workflow.

ACKNOWLEDGMENT

I want to express my gratitude to Epsilon India for giving me an internship to carry out the project. I want to thank my team for helping me out during the internship.

REFERENCES

[1] Wong, Richmond Y., Andrew Chong, and R. Cooper Aspegren. "Privacy Legislation as Business Risks: How GDPR and CCPA are Represented in Technology Companies' Investment Risk Disclosures." Proceedings of the ACM on Human-Computer Interaction 7.CSCW1 (2023): 1-26.

[2] Determann, Lothar, and Jonathan Tam. "The California Privacy Rights Act of 2020: A broad and complex data processing regulation that applies to businesses worldwide." Journal of Data Protection & Privacy 4.1 (2020): 7-21.

[3] Quinto, Butch, and Butch Quinto. "Batch and Real-Time Data Ingestion and Processing." Next-Generation Big Data: A Practical Guide to Apache Kudu, Impala, and Spark (2018): 231-374.

[4] Fortino, Roberto. "Reengineering of a Big Data architecture for real-time ingestion and data analysis." PhD diss., Politecnico di Torino, 2018.

[5] Sreemathy, J., et al. "Overview of ETL tools and talend-data integration." 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1. IEEE, 2021.

[6] Song, H., Xu, C., Gao, X., & Wang, H. (2019). A secure data privacy protection scheme based on blockchain in cloud computing. Future Generation Computer Systems, 99, 1-8.

[7] Zhang, Y., Wu, J., & Huang, X. (2020). Design of a secure data privacy platform in big data era. Journal of Ambient Intelligence and Humanized Computing, 11(1), 355-362

[8] Jorge-Arnulfo Quiané-Ruiz et al, "Pipeline-as-a-Service: A Framework for Deploying and Managing Data Integration Pipelines in the Cloud" 2019

[9] Chao Wu et al."Data Integration for Heterogeneous Data Sources using a Unified Pipeline", IEEE, 2020.

[10] Sfaxi, Lilia, and Mohamed Mehdi Ben Aissa. "DECIDE: An Agile event-and-data driven design methodology for decisional Big Data projects." Data & Knowledge Engineering 130 (2020): 101862.

[11] Fortino, Roberto. Reengineering of a Big Data architecture for real-time ingestion and data analysis. Diss. Politecnico di Torino, 2018.

[12] Sinha, Sudhi R., et al. "Constructing Data Service Platform." Building an Effective IoT Ecosystem for Your Business (2017): 95-121.

[13] Yayah, Fauzy Che, Khairil Imran Ghauth, and C. Ting. "Adopting big data analytics strategy in telecommunication industry." Journal of Computer Science & Computational Mathematics 7.3 (2017): 57-67.

[14] Bhimte, Pallavi H., et al. "Hadoop Framework: Big Data Management Platform for Internet of Things." From Visual Surveillance to Internet of Things. Chapman and Hall/CRC, 2019. 175-198.

[15] Janošcová, R. E. N. A. T. A. "Mining big data in weka." 11th IWKM, Bratislava, Slovakia (2016).

[16] Quinto, Butch, and Butch Quinto. "Introduction to Kudu." Next-Generation Big Data: A Practical Guide to Apache Kudu, Impala, and Spark (2018): 7-56.