Real-time Analytics Data Management with Data Lake: Benefits, and Challenges

Kafayat Adeoye
Computer Science Department
University of Nottingham
Enhanced Research Module COMP4098
psxka7@nottingham.ac.uk

***Abstract*—The utilization of data lake methodologies has gained significant traction in the realm of real-time analytics. However, it is imperative to prioritize responsible data management for the storage and analysis of streaming data. This study is presented to examine the advantages, best practices and potential difficulties associated with implementing a data lake strategy for real-time analytics, and showcase a use case for weather sensor data with a hybrid data lake structure called delta lake that combines the functionalities of traditional data warehouse and data lake, the data is ingested to the storage system through Apache Kafka and Apache Structured Streaming. This paper can serve as a guide in understanding the importance of the use of a data lake to store and maintain heterogonous large streaming datasets.**

***Index Terms*—data lake, big data, real-time analytics, data warehouse, delta lake, data ingestion, data storage, benefits, challenges, responsible, data management**

I. Introduction

Real-time analytics has become crucial in various domains, including the Internet of Things and social media, where immediate insights and decisions are required. The data lake approach provides the scalability and flexibility to handle large volumes of streaming data for real-time analytics. Big data lakes can be conceptualised as an ecosystem that involves numerous big data repositories that are characterized, processed, and analyzed using established big data technologies. The primary goal of a data lake is to facilitate refined big data analytics procedures. In the modern era, a multitude of big data repositories, influence diverse data types such as text, spreadsheets, web, relational, social, sensor, and real-time data etc [5]. Hence, this paper aims to explore the prevalence of having responsible data management and considering the advantages and challenges of the data lake approach for real-time analytics with the following objectives i) highlight the benefits offered by the data lake and best practices ii) identify the potential challenges associated with implementing a data lake strategy iii) describe data lake methodology and iv) show a sample case study using sensor data to validate the use of data lake on streaming data in real-time analytics settings.

A. Data Lake Versus Data Warehouse

The concept of a data lake was initially presented by James Dixon, in 2010. Dixon suggested that data lakes would consist of vast collections of centralized repositories, whether structured or unstructured, that users could readily access for the purposes of sampling, mining, or analysis [2]. The data lake, within this particular context, functions as a data-centric system that is integrated into the decision-making framework the content comprises raw data in its original format, which can be accessed by proficient users through specific tools [2]. In contrast to data warehouses, data lakes are required to possess agility and flexibility, the objective is to extract meaningful insights from the accumulated data through the processes of data exploration, statistical model training, and data transformation for utilisation in various services and applications [13]. Traditional methodologies that prioritise schema design before data ingestion, such as those used for relational databases, data warehouses, and Extract Transform Load procedures, may not be suitable for managing data in a dynamic and adaptable environment. In the realm of data lakes, indicates that tasks such as schema definition, integration, or indexing ought to be executed solely when required during data retrieval [19]. A data lake is expected to possess the capability to receive data from diverse data sources into a single homogenous data management system. This enables data owners to impose universal data governance and get over the limitations of diverse and isolated data silos [16]. However, to prevent its transformation into a 'data swamp', it is vital to establish guidelines for data governance that encompass the administration of data quality, data security, data lifecycle, and metadata [2].

Due to their set schema, data warehouses must be fed via a method known as "schema-on-write" which extracts the raw data from its source, modify it, such as by cleaning it into the predetermined schema, and then load it into the Data Warehouse, therefore, Extract-modify-Load (ETL) operations are required [19]. The biggest disadvantage of adopting this method is the information lost during the transformation of the data to suit the predetermined schema, despite the fact that there are certainly recognised difficulties [20]. The data lake is kept in its original format here, in contrast to the schema-on-write technique of a Data Warehouse, and a schema is only deduced when a future operation reads the data, an approach known as schema-on-read [20]. Data lakes are designed to assist analysts in conducting more effective and efficient data analysis by merging a variety of current data sources, processing methodologies, and analytical tools. However, it is challenging to achieve that while employing a data lake without a metadata governance system that eventually capitalizes on all the completed analytical experiments [21].

Conventional data warehouses that employ a structured format are not equipped to manage diverse data types with varying latency requirements. According to [9], in the framework of big data, the utilisation of a Data Lake may have the potential to address the challenges of managing large volumes and diverse types of data, provided that those issues are effectively addressed. The fundamental principle of preserving data in its native format within a data lake facilitates a wide range of use cases and enhances data reusability [10]. To avert inaccessible, invisible, or unreliable data, this data lake must enable the ingest of various data sources, the implementation of preprocessing prior to analytics, the provision of access and consumption of this preprocessed data, and the proper governance of the data throughout the previous steps [22].

II. Related Work

As stated [16] with the rise of big data, data warehouses are no longer equipped to store the vast amounts of unstructured data generated by businesses. As a result, data lakes were envisioned, which essentially facilitate the storage of huge data by storing the data in its raw format. The paper examines the notion of traditional data warehousing and explains why most businesses are choosing data lakes in the current era of big data and machine learning. Modern businesses are switching from data warehouse to data lake architecture for their business intelligence operations because data lakes offer effective and scalable storage, and data lakes offer low-cost analytical platforms for both batch and real-time data. Additionally, the raw data saved in the data lake is able to make use of the machine learning and artificial intelligence procedures that are already integrated into the cloud storage environment, making the data analytics much more user-friendly. Thus, to maximise the effectiveness of their total company processes, modern businesses choose to employ data lakes rather than data warehouses.

[21]The authors investigate the data lake paradigm to address the problems associated with the storage of heterogeneous data and to give the capability of quick data processing. The most recent DataLake systems are presented in this document, together with a list of their main benefits and limitations and suggest a fix for the Data Lake System. The authors, concentrate on the challenges involved in converting a company's legacy information system to a data lake, implementation, administration, and exploitation are the concerns at hand. Since it is expensive, it is difficult to convert all of the data into a data lake at once. A service-oriented architecture is necessary for businesses to manage access to external data together with the company's internal databases without permanently keeping it, and provide a method that builds interfaces for each source of information. To do metadata migration they focus on extracting, if feasible, the metadata of each source. One global schema will be created by combining these schemata the benefit of this strategy is that it avoids the need to switch the entire system's information to a single server at once.

An efficient architecture is required to supply substantially higher data volumes and kinds for effective storage and analyses. The architecture for data storage and analytics using Spark was suggested in the [23] study for the big data lake of electricity use. An existing system's historical data was transferred to Apache Hive using Apache Sqoop for processing. To guarantee the integrity of the streaming data, Apache Kafka was utilized as the input source for Spark to stream data to Apache HBase. The authors leverage the Data Lake's Hive and HBase principles as search engines for Hive and HBase in order to combine the data. Separately, Apache Impala and Apache Phoenix are utilized. Moreover, Apache Spark is used in this work to analyze electricity use and power outages. This project's visualizations are all displayed in Apache Superset. Also, the HoltWinters method is used to display the use of forecast comparison. In this work, we advocated the use of Spark for large data lake analytics and data storage related to power use. In this instance, we design an architecture that can transfer existing data storage systems to Data Lake's Big Data platform and offer data storage, analysis, and visualisation applications for big data.

According to [24], although the use of data lakes in firms has not yet been the subject of empirical research, big data innovation can aid businesses in their business intelligence processes. By describing the concept, functional architecture, stages of development, and a variety of research problems and guidelines, the paper provides a preliminary review of data lake implications which will increase the effective use of the data lake approach in enterprises.

The data lake is a notion for more adaptable and potent data analytics that has just lately come into existence [26] literature mentioned that research available on data lakes, however, is relatively hazy and unfinished, and the numerous execution methods that have been suggested neither completely address the topic nor offer a plan for its design and realization. Consequently, businesses that create data lakes confront a variety of difficulties, such as governance or data models. The authors' paper looks at the current state of the field for data lakes and any open research problems that need to be solved before a data lake can be effective. The following contributions are made in the paper to support the broad data lake concept's present condition, current design and realization issues, and identify issues and outstanding research needs for data lakes.

[18] mentioned Enterprises' capacity to ingest the many data kinds that could reside in organizational silos has been outpaced by the data explosion. This study examines the benefits and drawbacks of creating data lakes, a potential method for using data as a strategic asset for business decision-making.

[1]in their research examines the issue of integrating geographical data into current databases and information systems. In light of Big Data, they draw attention to the shortcomings of traditional strategies like geographical expansions and data warehouses. The authors propose a data architecture

”Lakehouse” to get around these restrictions. The Lakehouse combines the adaptability and scalability of Data Lakes with the dependability of Data Warehouses. In order to provide an effective method for handling geographical big data, it strives to combine the complexity of spatial data with operational and analytical systems. The study presents a review of the literature on conventional methods for managing geographic data and highlights their drawbacks in relation to spatial big data. It describes the Lakehouse as an alternative strategy, its components, and best practices for creating an effective Lakehouse architecture for geographical data.

The findings of exploratory research conducted to better understand how the data lake concept is used in organizations are presented in the [12] article. The author conducted an interview with 12 experts who had used this strategy in different businesses and also outlines a number of potential advantages and disadvantages of the data lake strategy. The author’s research was guided by the following research questions: What benefits does integrating a data lake into a BI infrastructure provide? How do data lakes impact an organization’s BI architecture? What are the advantages and difficulties of adding a data lake to a BI architecture?

The paper authored by [7] highlights the emergence of data warehouses and enterprise data lakes as a solution to the challenge of managing data silos and extracting value from the vast and diverse data types available. These technologies offer improved scalability and versatility, enabling organizations to leverage their unlimited data resources more effectively. The presented study centres on the crucial function of enterprise Data Lake in addressing big data obstacles and propelling the digital transformation epoch. It also delves into the potential of constructing an enterprise data lake to revolutionize the business paradigm. The text delves into the market landscape and expansion of data lakes, while also exploring their potential for future development, as well as the challenges and opportunities that accompany the proliferation of enterprise data lakes.

The paper authored by [12] examines how alterations in the data paradigm have given rise to novel architectures for data analytics and management. The presented study focused on the storage of data with high volume, velocity, and variety in their raw formats, utilizing a data storage framework known as a data lake. Initially, introduce their investigation concerning the constraints of conventional data warehouses in managing contemporary alterations in data paradigms. This paper presents data analytics and management infrastructure that can enable intricate multilayered predictive analytics for live streaming data from a multitude of sources for efficient storage, analysis, and querying of high-velocity data necessitating the utilisation of a data lake.

The related work review emphasizes the crucial role of managing real-time big data analytics through a data lake, highlighting its ability to accommodate the dynamic nature of big data, including velocity, variety, veracity, and volume. The review also discusses the potential challenges associated with data lake architecture and compares it with traditional data warehouses, which rely on relational databases. Additionally, the review explores the combined capabilities of data warehouse and data lake in the development of data lakehouses, such as Delta Lake, and its application in statistical analysis.

III. Data Lake Architecture

The architecture of Data Lake has undergone significant evolution since its inception in early 2010. Originally, the purpose of Data Lakes was to store large quantities of diverse data in its unprocessed and unaltered state. From its inception, Data Lakes were positioned as an advancement or potential substitute for Data Warehouses, a prevalent framework utilised for analytical objectives and informed decision-making. Data Lakes have the capability to store and process data in near real-time, thereby presenting an opportunity to serve as a support system for analytics over operational data. Diverse architectural models have surfaced, each offering distinct advantages for data retention, analysis, and user utilisation [8]. The concept of data lake architecture encompasses the entirety of a data lake’s design, including its infrastructure, data storage, data flow, data modelling, data organization, data processes, metadata management, data security and privacy, and data quality [10]. This paper will touch on the four functional layers of the architecture of a data lake that can effectively handle large-scale data sets is outlined as follows;

- ***Ingestion layer***: Data can be ingested into the data lake either in real-time or through the traditional batch format [16] from various sources and its subsequent integration into the Data Lake. The preliminary extraction of metadata from structured and semi-structured data is carried out through automated means [8] through metadata extractor extracts the maximum amount of metadata feasible from the data source, such as schemas from relational or XML sources, and subsequently stores this information in the metadata storage of the data lake [2]
- ***Storage layer***: comprises the metadata repository and repositories for raw data. A repository for raw data offers comprehensive assistance for various formats and configurations of data, encompassing both file-based and record-based storage [8]. The interface offers users the ability to query data without exposing the intricate details of the underlying storage complexity [16].
- ***Transformation layer***: has the potential to enable scalable execution of various operations, including data cleansing, data transformation, and data integration [20] and is responsible for the comprehensive management of the data lake [24]
- ***Interaction layer***: facilitates the provision of access to both the metadata repository and the data generated in the transformation layer for end-users [23] Users are able to retrieve this data for the purpose of conducting data exploration, generating and implementing analytical queries, and employing diverse visualisation tools to display the data that has been stored [25].

IV. Benefits of Data Lake

Big data has characteristics such as volume, velocity, variety, truthfulness, variability, value, and visibility and can be organised. Big data processing comes in three forms: hybrid, streamed in real-time, and batching. A data lake contains a considerable quantity of unprocessed data that is unprocessed in its initial form either structured, unstructured, or semi-structured, [14] viewed in accordance with the need for reuse until necessary as well as computing devices (engines) that can consume data without affecting the data structure with the unique requirements for data lakes in terms of the nature of the managed data. Security, accessibility, flexibility, transparency, efficiency, quantity, integrity, durability, maintenance, and dependability are the essential requirements [21]. Significant cost savings can be achieved through the integration of software and hardware, enabling the consolidation of all data types into a single repository for streamlined data management.

The methodology involves combining disparate data sets into a singular location, thereby focusing on the perceived advantages of data lakes [18]. The concept of minimising the initial effort through data storage, improved data acquisition, expedited access to unprocessed data, and data retention has led to the identification of three primary rationales for utilising data lakes. These include their utility as staging grounds for data warehouses, their capacity to serve as a platform for experimentation by data scientists and analysts, and their potential as a direct source for self-service business intelligence [12]. The utilization of data lakes has been shown to enhance operational efficiency and reduce transaction costs. Enhancing data quality and accelerating the transmission of information, augmenting scalability in data handling [7].

The primary advantage of a data lake lies in its ability to consolidate information from diverse origins into a centralized repository. The vast amount of tables or records and files that can be contained within a data lake necessitates the need for scalable solutions for data storage, management, and analytics [3]. Upon aggregation within the data lake, data originating from various sources can be effectively correlated, integrated, and analyzed through advanced big data analytics and search methodologies that would have otherwise been unfeasible [11]. The main utilization scenario for a data lake is to execute preliminary processing and extract, transform, and load conversion of data, in order to facilitate subsequent exploration by other systems [13]. The schema-on-read approach refers to the process of defining the data structure at the time of its usage. This approach, adopted by data lakes, eliminates the need for intricate and expensive data modelling and integration efforts [6].

A. Responsible Data Management

In the absence of establishing relationships between datasets within a data lake, there exists a potential for the data lake to merely function as an assortment of disparate information silos, rendering it ineffective in the long run. Metadata holds significant importance in the process of querying data lakes [15]. The reason for this is that querying a data lake differs from querying a data warehouse, as the former does not rely on a predetermined schema or strict constraints. Instead, querying a data lake requires an exploratory approach to identify the pertinent data sources that contain the desired information [2]. The concept of data lakes pertains to the quality of data. Similar to data warehouses, a multitude of issues pertaining to data quality emerge when the data is utilised in a context that differs from its intended purpose. The usefulness of legacy data beyond its typical context may be restricted due to the presence of particular semantics and workarounds that have been hard-coded into the applications [5].

Without the ability to define relationships between the datasets in a data lake, there is the risk that the data lake is just a collection of independent information silos and that it becomes useless over time, The metadata is also important for querying data lakes [26]. This is because, instead of querying a relational style with a predefined schema and precise constraints as in data warehouses, querying a data lake will also involve an exploration process to detect the data sources which are relevant for a piece of certain information that is needed [20]

As opposed to conventional schema-on-write approaches, a data lake infrastructure must possess the capability to accommodate diverse data types, including those lacking a pre-existing schema [23]. The concept of schema-on-read refers to the process of generating a schema at the time of query execution, potentially involving deferred indexing, view generation, or on-demand query planning [6]. The rationale behind utilising the schema-on-read approach is that the optimal method of organising the schema is contingent upon the specific query or analysis being performed, and therefore may not be ascertainable during the loading phase [4].

V. Challenges in Data Lake Approach

One of the primary obstacles in constructing and implementing a data lake for an organisation is ensuring data reliability. The optimal utilisation of data present in the data lake can be challenging for both data scientists and end-users in the absence of advanced analytical tools [16]. The enforcement of data validation techniques in a data lake setting can prove to be an issue in the absence of expert assistance. Managing the capture and update of a substantial volume of historical data concurrently can be challenging due to the dual processing of batch and real-time data in data lakes [3]. The presence of numerous small files in the data lake may result in decreased performance due to limits in the input/output throughput [14] and the interactive query performance may be limited due to the vast amount of data it contains [19].

The matter of security has been a persistent concern, however, commendable strides have been taken by both the open-source community and vendors in order to facilitate an organization's security and privacy prerequisites [12]. The preservation of data privacy is neglected in a conventional data lake, as the primary objective of creating a data lake is for business purposes, and multiple users of the data lake access a shared data repository [21]. The challenges associated with data lakes can be summarized as follows;

- The incapacity to ascertain the quality of data or the provenance of data discoveries [18].
- The Data Lake system lacks oversight and governance, allowing for the acceptance of any data without proper descriptive metadata or a mechanism for maintaining metadata [9]. This can result in the accumulation of unorganized and unstructured data, commonly referred to as a "data swamp." It is necessary to conduct a comprehensive analysis of data anew on each occasion, as there is no assurance of performance [17].
- The security of data lakes encompasses aspects such as privacy, regulatory compliance, and access control is a critical concern [12].

VI. Delta Lake Deployment

Delta Lake is a suitable solution for a majority of data lake applications that would have previously utilized structured storage formats like Parquet objects, as well as numerous conventional data warehousing workloads [3]. The utilization of Delta Lake enables the modification of the schema of Spark tables and provides additional functionalities. Facilitating the capability of updating data in a Data Lake without necessitating traversal of the complete Data Lake repository [17]. The Delta table is composed of partitioned data in Parquet files, which facilitates efficient data skipping, particularly in selective queries aimed at specific data subsets. The delta transaction log is a constituent of the delta table, as it maintains a record of all the commits executed on the table, thereby ensuring ACID-atomicity, consistency, isolation, and durability compliance [4]. The Delta Engine offers optimization techniques to enhance the efficiency of parallel processing through Apache Spark including optimization of the data layout [1].

A. Weather Sensor Streaming Data

The objective of this case study is to apply delta lake to store data streams sourced from Open Weather's Air Pollution Hourly Forecast. The data pertains to the levels of polluting gases in the city of Nottingham, which is geographically located at latitude 52.950001 and longitude -1.150000. The subsequent section explains the process of data ingestion and the deployment of the data in delta lake storage. This data can subsequently be used for a machine learning model prediction.

1) Set up Apache Spark: The *SparkSession* can be regarded as the mixture of various contexts, which also encompasses the *StreamingContext*. The establishment of this foundation serves as the basis for constructing Structured Streaming. The SparkSession serves as the primary interface for connecting PySpark code to the Spark cluster. The code below instantiates a local spark session called Kafka Stream as demonstrated below where the * alongside the 'import' command means all the methods of the module should be imported to be used in the code function. *getOrCreate()* is to create the spark UI interface where execution mode can be visualized;

```
//import spark functions, data types
and spark sql to initiate SparkSession
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *

// create the session with getOrCreate()
spark = SparkSession.builder \
.appName("KafkaStream") \
.getOrCreate()
```

2) Ingestion Layer: The collected data API response from open weather is in JSON format from the API request; the best way to extract JSON objects is to use the requests and json python libraries

```
// import python json and request function
import requests
import json
```

Open weather required to create of an API key on the website which is included in the API url with the coordinate of the location where the air pollution data will be extracted

```
// create a URL with the open weather
API request to fetch data from it

url = "http://api.openweathermap.org/data/
2.5/air_pollution/forecastlat=52.950001&
lon=-1.150000&appid=
2d557f868d673c2659a653d495a004dd"

// use the requests.get()
function to parse the data stream

response = requests.get(url)

// create an if statement to get data
when the site is available with code 200
for normal site operation and print
an error message if the site is unavailable

if response.status_code == 200:
    data = response.json()
else:
    print("Error fetching data from API")
```

Kafka Ingestion: Setting up *KafkaProducer()* function with the broker servers called bootstrap servers indicates the host machine address 127.0.0.1 on port address 9092 and serializes data to the appropriate json encoded format. The value serializer variable refers to the process of converting data structures into a particular format in this case JSON with the JSON.dumps() function and encoded to `utf-8` in order to efficiently manage the format of the data the origin and the destination. The *producer.send()* function publishes the formatted json weather data or event to the Kafka topic *'weather data'*.

```
//import KafkaProducer producer is
used to fetch data from the API response
from kafka import KafkaProducer

// creating Kafka producer to extract
data from open weather API
producer = KafkaProducer(bootstrap_servers=
[’127.0.0.1:9092’],
value_serializer= lambda x: json.dumps(x)
.encode(’utf-8’))

// send data to Kafka topic with
the producer.send()
producer.send(’weatherdata’, data)
producer.flush()
```

3) Extraction from Kafka to Apache Structured Streaming: The structured streaming consumes data as a data frame df from the Kafka streaming producer; the *spark.readStream()* method read the data from the specified source with format() indicating that the data source is been derived from the Kafka producer and the option() provides the configurations of the producer source to subscribe to the created Kafka topic and load() checks and returns the streaming data values as a data frame which is an asset of rows of a dataset with the defined schema.

- bootstrap.servers: kafka server ports
- subscribe: subscribe specify the Kafka topic that holds the data
- startingOffsets: is a policy applied to begin data extraction from the earliest generated.

```
// structured streaming acquiring streaming
data from Kafka producer

    df = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers",
    "127.0.0.1:9092") \
    .option("subscribe", "weatherdata") \
    .option("startingOffsets", "earliest") \
    .load()
```

4) Importing Data from Apache Structured Streaming to Delta Lake: Structured Streaming can be utilized to write data into a Delta table as well. Delta Lake leverages the transaction log to ensure the delivery of exactly-once processing, even in the presence of concurrent streams or batch queries being executed on the table. The integration of Delta Lake with Spark structured streams is extensive, facilitated by the employment of readStream and writeStream. The code following steps demonstrates first creating a query table called `my_query’ just as shown in the code above but in this case it sinks into memory and is extracted from there to pass the data frame into the variable called stream with sql command ‘select’.

```
// create a streaming query data table

stream_query = df_n.writeStream \
    .outputMode("append") \
    .format("memory") \
    .queryName("my_query") \
    .start()

stream = spark.sql("SELECT * FROM my_query")
```

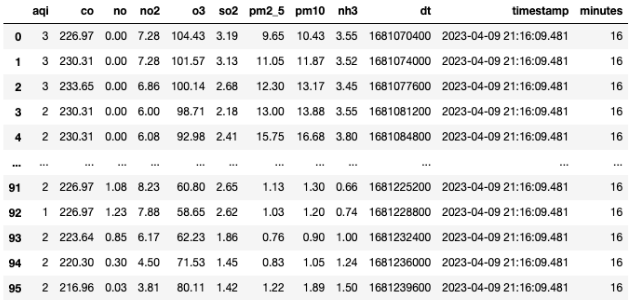
Then infer from the query table to the delta lake table with the stream.write.format() in append mode and save() as a new delta table. The format() method specifies the sink mode which is ‘delta’ and Figure 1 displays the structured dataset in delta lake.

```
// import the delta table function
from delta.tables import *

// write the stream to Delta Lake
stream2 =stream.write.format("delta")
.mode("append")
.save("/tmp/df_n_table")

// read the streaming from Delta Lake
and display as pandas table
stream3 = spark.read.format("delta")
.load("/tmp/df_n_table")

// load Delta table as DataFrame
display(DeltaTable.forPath(spark,
"/tmp/df_n_table").toDF().toPandas())
```



	aqi	co	no	no2	o3	so2	pm2_5	pm10	nh3	dt	timestamp	minutes
0	3	226.97	0.00	7.28	104.43	3.19	9.65	10.43	3.55	1681070400	2023-04-09 21:16:09.481	16
1	3	230.31	0.00	7.28	101.57	3.13	11.05	11.87	3.52	1681074000	2023-04-09 21:16:09.481	16
2	3	233.65	0.00	6.86	100.14	2.68	12.30	13.17	3.45	1681077600	2023-04-09 21:16:09.481	16
3	2	230.31	0.00	6.00	98.71	2.18	13.00	13.88	3.55	1681081200	2023-04-09 21:16:09.481	16
4	2	230.31	0.00	6.08	92.98	2.41	15.75	16.68	3.80	1681084800	2023-04-09 21:16:09.481	16
...	...	...	...	...	...	...	...	...	...	...	...	...
91	2	226.97	1.08	8.23	60.80	2.65	1.13	1.30	0.66	1681225200	2023-04-09 21:16:09.481	16
92	1	226.97	1.23	7.88	58.65	2.62	1.03	1.20	0.74	1681228800	2023-04-09 21:16:09.481	16
93	2	223.64	0.85	6.17	62.23	1.86	0.76	0.90	1.00	1681232400	2023-04-09 21:16:09.481	16
94	2	220.30	0.30	4.50	71.53	1.45	0.83	1.05	1.24	1681236000	2023-04-09 21:16:09.481	16
95	2	216.96	0.03	3.81	80.11	1.42	1.22	1.89	1.50	1681239600	2023-04-09 21:16:09.481	16

Figure 1 Delta Table for Weather Sensor Data

VII. Discussion and Limitations

The experience with data lake - Delta Lake shows that it can be implemented for big data real-time analytics workload that supports the functionalities such as data acquisition, metadata catalogue, data storage, data exploration, data aggregation, data lifecycle and data quality. The design provides advanced functionalities such as autonomous data layout optimisation, upserts, caching, and audit logs. At present, Delta Lake exclusively offers serializable transactions within an individual table, owing to the fact that each table possesses its own transaction log and this limitation can be eliminated by distributing the transaction log among several tables. In situations involving high transaction volumes, it may be valuable to employ a coordinator to facilitate write access to the log such as Apache structured streaming as depicted in the ingestion layer in the weather sensor data use case, this coordinator would operate independently of the read and write processes for data objects. The delta lake features adopted as stated by [1] ACID transactions which refers to a set of properties that ensure reliability and consistency, integration of batch and streaming processing, cache and optimisation of Data Layout [5] In the context of streaming tasks, the performance of Delta Lake is constrained by achieving millisecond streaming latency [3], still, the data workload in the use case which was executed as parallel tasks through Apache Spark has low latency while utilizing Delta Lake tables which was considered adequate.

VIII. Conclusion

This paper explains the importance of using data lake storage for integration, processing and managing real-time big data considering the dynamic characteristics of this type of data stream, potential challenges, best practices and the use case that was also described to evaluate the performance of the system. Data Lake presented a suitable solution to deal with the massive amount and velocity at which streaming data can be generated. Delta Lake is a hybrid data management system combining the functionalities of a data lake and data warehouse to showcase the implementation of real-time data with weather sensor data as an example. However, the undefined flow of stream processing and the need for precise responses in real-time will ask for definite specifications in handling the data in Data Lake [9]. As security and metadata management are weak in Data Lake, data veracity that is data integrity cannot be assured so it is recommended that researchers and IT professionals focus on these problems in upgrading the storage framework to support the streaming data analytics lifecycle in order to ensure the quality of insights derived from the acquired data.

References

[1] Ait Errami, S., Hajji, H., Ait El Kadi, K., I& Badir, H. (2023). Spatial big data architecture: From Data Warehouses and Data Lakes to the Lake-House. In Journal of Parallel and Distributed Computing (Vol. 176, pp. 70–79). Academic Press Inc. https://doi.org/10.1016/j.jpdc.2023.02.007

[2] Anne Laurent Dominique Laurent Cédrine Madera. (2020). Data Lakes.

[3] Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Łuszczak, A., Świtakowski, M., Szafrański, M., Li, X., Ueshin, T., Mokhtar, M., Boncz, P., Ghodsi, A., Paranjpye, S., Senster, P., … Zaharia, M. (2020). Delta lake. Proceedings of the VLDB Endowment, 13(12), 3411–3424. https://doi.org/10.14778/3415478.3415560

[4] Belov, V., I& Nikulchev, E. (2021). Analysis of Big Data Storage Tools for Data Lakes based on Apache Hadoop Platform. Article in International Journal of Advanced Computer Science and Applications, 12(8), 2021. https://doi.org/10.14569/IJACSA.2021.0120864

[5] Cuzzocrea, A. (2021). Big data lakes: Models, frameworks, and techniques. Proceedings - 2021 IEEE International Conference on Big Data and Smart Computing, BigComp 2021, 1–4. https://doi.org/10.1109/BIGCOMP51126.2021.00010

[6] Fang, H. (2015). Managing data lakes in the big data era: What's a data lake and why has it become popular in the data management ecosystem. 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015, 820–824. https://doi.org/10.1109/CYBER.2015.7288049

[7] Glory Johny, M., Pillai, S. (2022). Analyzing the vital role of an enterprise data lake in the era of digital transformation. 2519, 30059. https://doi.org/10.1063/5.0109834

[8] Hlupic, T., Orescanin, D., Ruzak, D., I& Baranovic, M. (2022a). An Overview of Current Data Lake Architecture Models. 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology, MIPRO 2022 - Proceedings, 1082–1087. https://doi.org/10.23919/MIPRO55190.2022.9803717

[9] Khine, P. P., I& Wang, Z. S. (2017). Data lake: a new ideology in big data era. https://doi.org/10.1051/itmconf/20181703025

[10] Liu, H., Quix, C., Blomer, J., I& Wieder, P. (2022). Toward data lakes as central building blocks for data management and analysis. https://flume.apache.org/

[11] Liu, R., Isah, H., I& Zulkernine, F. (2020). A Big Data Lake for Multilevel Streaming Analytics. 2020 1st International Conference on Big Data Analytics and Practices, IBDAP 2020. https://doi.org/10.1109/IBDAP50342.2020.9245460

[12] Llave, M. R. (2018). Data lakes in business intelligence: Reporting from the trenches. Procedia Computer Science, 138, 516–524. https://doi.org/10.1016/J.PROCS.2018.10.071

[13] Mathis, C. (2017). Data Lakes. Datenbank-Spektrum 2017 17:3, 17(3), 289–293. https://doi.org/10.1007/S13222-017-0272-7

[14] Miloslavskaya, N., I& Tolstoy, A. (2016). Big Data, Fast Data and Data Lake Concepts. Procedia Computer Science, 88, 300–305. https://doi.org/10.1016/J.PROCS.2016.07.439

[15] Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., I& Arocena, P. C. (2019). Data lake management. Proceedings of the VLDB Endowment, 12(12), 1986–1989. https://doi.org/10.14778/3352063.3352116

[16] Prakash, S. S., I& Tech, M. (2020). Evolution of Data Warehouses to Data Lakes for Enterprise Business Intelligence International Journal of Innovative Research in Computer and Communication Engineering Evolution of Data Warehouses to Data Lakes for Enterprise Business Intelligence. An ISO, 4, 1038. www.ijircce.com

[17] Saddad, E., El-Bastawissy, A., Mokhtar, H. M. O., I& Hazman, M. (2020). Lake Data Warehouse Architecture for Big Data Solutions. IJACSA) International Journal of Advanced Computer Science and Applications, 11(8). www.ijacsa.thesai.org

[18] Shepherd, A., Kesa, C., Cooper, J., Onema, J., I& Kovacs, P. (2018). OPPORTUNITIES AND CHALLENGES ASSOCIATED WITH IMPLEMENTING DATA LAKES FOR ENTERPRISE DECISION-MAKING. Issues in Information Systems, 19(1), 48–57. https://doi.org/10.48009/1_iis_2018_48-57

[19] Terrizzano, I., Schwarz, P., Roth, M., I& Colino, J. E. (2015). Data Wrangling: The Challenging Journey from the Wild to the Lake.

[20] Haddadi, O. El, Hamlaoui, M. El, Taoufiq, D., I& Nassar, M. (2020). Data Lake and Digital Enterprise. https://doi.org/10.5220/0009415604230429

[21] CHERRADI, M., I& HADDADI, A. EL. (2022). A Scalable Framework for data lake ingestion. Procedia Computer Science, 215, 809–814. https://doi.org/10.1016/J.PROCS.2022.12.083

[22] Dang, A. R. F.; V.-N., Zhao, Y., Megdiche, I., I& Ravat, F. (2021). A Zone-Based Data Lake Architecture for IoT, Small and Big Data. https://doi.org/10.1145/3472163.3472185

[23] Yang, C. T., Chen, T. Y., Kristiani, E., I& Wu, S. F. (2021). The implementation of data storage and analytics platform for big data lake of electricity usage with spark. Journal of Supercomputing, 77(6), 5934–5959. https://doi.org/10.1007/S11227-020-03505-6/FIGURES/26

[24] Singh, J., Singh, G., I& Bhati, B. S. (2022). The Implication of Data Lake in Enterprises: A Deeper Analytics. 8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022, 530–534. https://doi.org/10.1109/ICACCS54159.2022.9784986

[25] Stach, C., Bräcker, J., Eichler, R., Giebler, C., I& Mitschang, B. (2021).Demand-Driven Data Provisioning in Data Lakes. ACM International Conference Proceeding Series, 187–198. https://doi.org/10.1145/3487664.3487784

[26] Giebler, C., Gröger, C., Hoos, E., I& Schwarz, H. (2019).Leveraging the Data Lake Current State and Challenges Institute for Parallel and Distributed Systems / AS. https://doi.org/10.1007/978-3-030-27520-4_13