iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** III **Month of publication:** March 2021

DOI: https://doi.org/10.22214/ijraset.2021.33376

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com

Lakehouse: A Unified Data Architecture

Divyansh Jain[1], Samay Jain[2]
[1]*(Data Engineer) Yash Technologies, India*
[2]*(Jnr. Data Engineer) Yash Technologies, India*

Abstract: ***There is no doubt that the industries are going ablaze with the huge eruption of data. None of the sectors have remained untouched by this drastic change in a decade. Technology has crept inside each business arena and hence, it has become an essential part of every processing unit. However, the data these companies are dealing with is of a different type, size and is coming with high velocity. Well, there are so many challenges with the traditional data lake and thus companies are looking for a solution to get rid of the traditional lakes. Hence, to overcome these challenges, Lakehouse came as a Solution.***

I. TRADITIONAL DATA LAKE

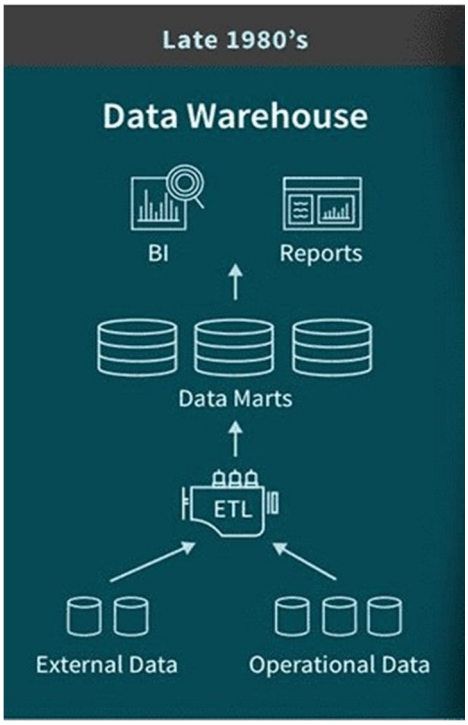
A. Data Warehouse

In the early 1980s, when the data warehouse was evolving there was a mass majority of the companies using Data Warehousing which helped companies taking advantage of Massive Parallel Processing to process a high volume of data.

While Data Warehouse only supports Structured Data, it became challenging for organizations to handle Unstructured & SemiStructured Data. Even it is not possible to apply DataScience, Machine learning Stuff for analytics. Thus, Data Warehouse is not suited for many of these use cases & is not very cost-efficient.

As data volumes grew even larger (big data), and as the need to manage unstructured and more complex data became more important, data warehouses had limitations:

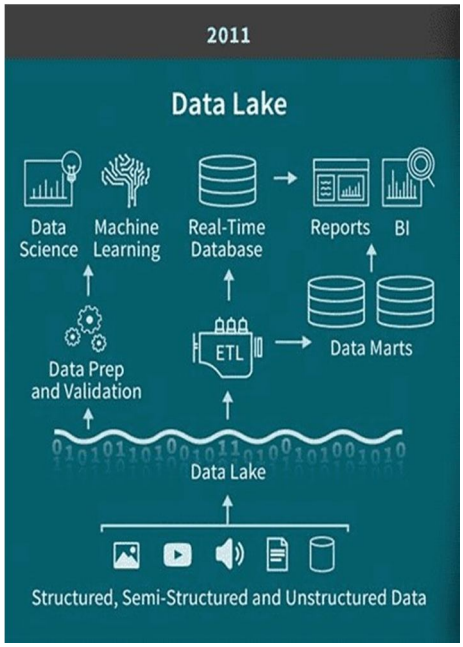
1) Data warehouses for a huge IT project can involve high maintenance costs.
2) Data warehouses only support business intelligence (BI) and reporting use cases.
3) There's no capability for supporting ML use cases.
4) Data warehouses lack scalability and flexibility when handling various sorts of data in a data warehouse.



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com

B. Data Lake

With the advent of big data, traditional architectures like the data warehouse must be reconsidered. With data coming from different sources, in different formats, and usually in a bigger volume, a new paradigm needed to emerge to fill this gap. In a data lake, the data is stored in its raw format and it's only queried when a business question arises, retrieving relevant data that can then be analyzed to help answer the question. The data is stored in cloud storage like Amazon S3, which has become one of the largest and most cost-effective storage systems in the world as it makes it possible to store practically limitless amounts of data in its native format at a low cost.



Parameters	Data Warehouse	Data Lake
Data Capturing	Structured Data	Structured, Semi-Structured, Unstructured Data
Processing	ETL (Extract Transform Load)	ELT (Extract Load Transform)
Storage Cost	Costly	Relatively inexpensive
Schema	Schema on Write	Schema on Read
Users	Ideal for operational users, because of well structured and easy to use	Data Analysis, Data Science

Data Warehouse v/s Data Lake

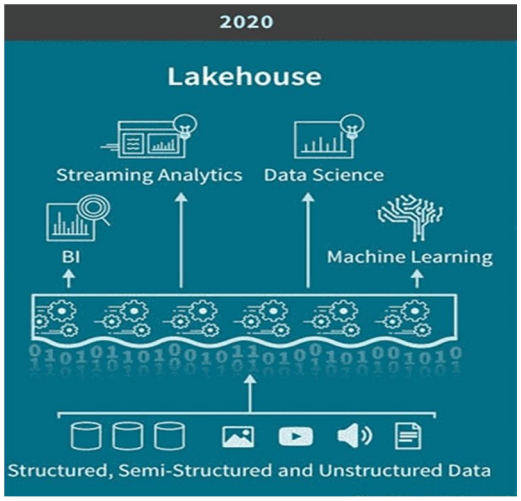
Well**,** a common approach is to use multiple systems – a data lake, several data warehouses, and other specialized systems such as streaming, time-series, graph, and image databases. Having a multitude of systems introduces complexity and more importantly, introduces delay as data professionals invariably need to move or copy data between different systems which causes delay and increases the cost.

Considering all these challenges, databricks came up with a solution called Lakehouse, which combines the best of these two worlds i.e Data Lake & Data Warehouse.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com

C. LakeHouse

A Data LakeHouse is a data solution that combines the best of both Data Lake and Data Warehouse.

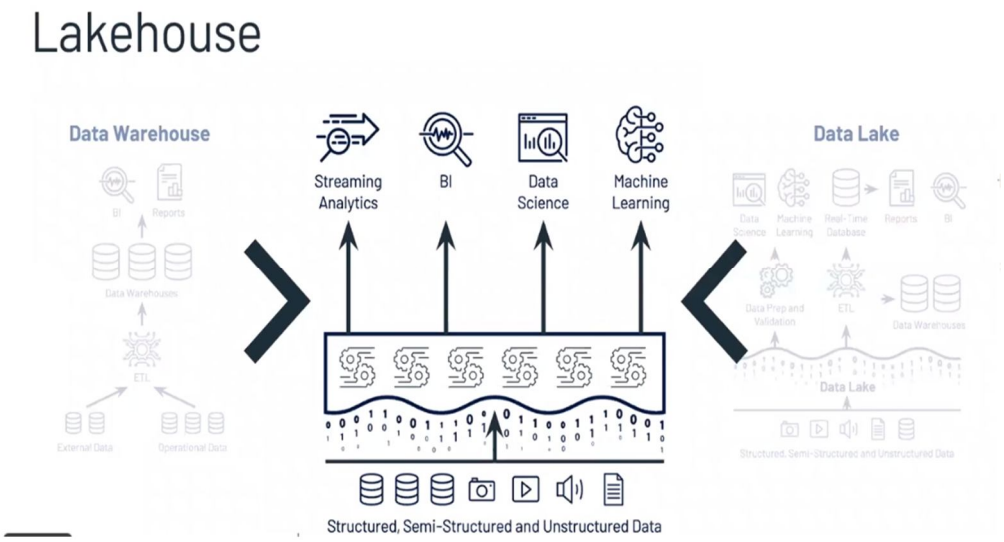


II. INTRODUCTION

Data Lakehouses is enabled by a new open and standardized system design: implementing similar data structures and data management features to those in a data warehouse, directly on the kind of low-cost storage used for data lakes. They are what you would get if you had to redesign data warehouses in the modern world, now that cheap and highly reliable storage.

Data LakeHouse provides processing power on top of Data Lakes such as S3, HDFS, Azure Blob, etc. In other words, it doesn't have to load the data onto any of the data warehouses to process and get the analysis or Business intelligence requirement done.

A. Why Lakehouse?



Earlier data scientists have to learn and understand the ETL/ELT pipelines so that they can have good visibility on how the data transformed, loaded and validated which was an extra job for data scientists. But Data Lakehouse provides us with a query engine to query the data directly from raw data which will help data scientists to build their transformation logics and cleaning techniques after understanding basic statistical insights and quality of the raw data.

B. Features of Lakehouse

1) *Transaction Support:* There are scenarios where multiple data pipelines simultaneously read and write data in an enterprise lakehouse which can lead to an issue. But, Lakehouse handles it carefully as it supports ACID transactions which maintain consistency when multiple users read or write data at a time.
2) *Schema Enforcement and Governance:* The Lakehouse should have a way to support enforcement and evolution of schema, supporting paradigms of DW schema such as star/snowflake-schemes. The framework should be capable of thinking about data privacy and should have robust processes of governance and auditing.
3) *BI Support:* Data LakeHouses allow the direct use of BI tools on the source data. This reduces staleness, increases recency, decreases latency and reduces the cost of having two operational copies of data in the data lake and data warehouse.
4) *Decoupled Storage from Compute:* Separated storage & compute, so these systems can scale to many more concurrent users and larger data sizes. This feature very well helps in reducing cost by using the resources based on the requirement meaning can use compute and storage independently whenever needed. Though this property is present in some of the modern data warehouses already.
5) *Openness:* Store files as Parquet, which is an open and standardized format, and provides an API so that a variety of tools and engines can directly access the data effectively, including machine learning and Python/R libraries.
6) *Support for Diverse Workload:* Data science, ML, SQL and analytics are included. To support all these workloads, multiple tools might be needed, and they all rely on the same repository.
7) *End-to-end Streaming:* Real-time streaming scenarios are widely used. Hence, Streaming support removes the need for different systems dedicated to support applications for real-time data.

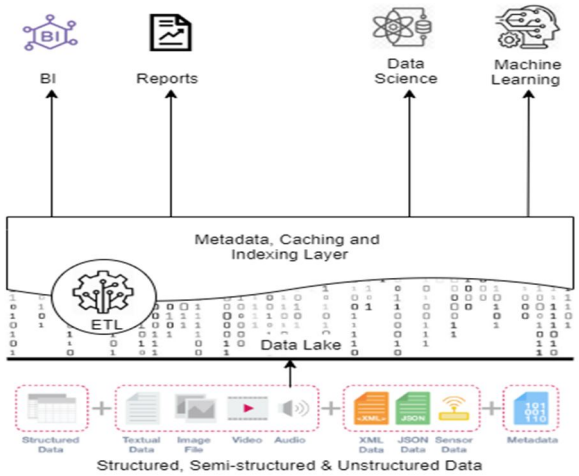
C. Lakehouse Architecture

As guaranteed, Data lakes are well known for low cost and flexibility and Data Warehouses are known for reliability & performance. So combining these features of both Data Lake & Data Warehouse introduces the concept of Lakehouse Architecture.

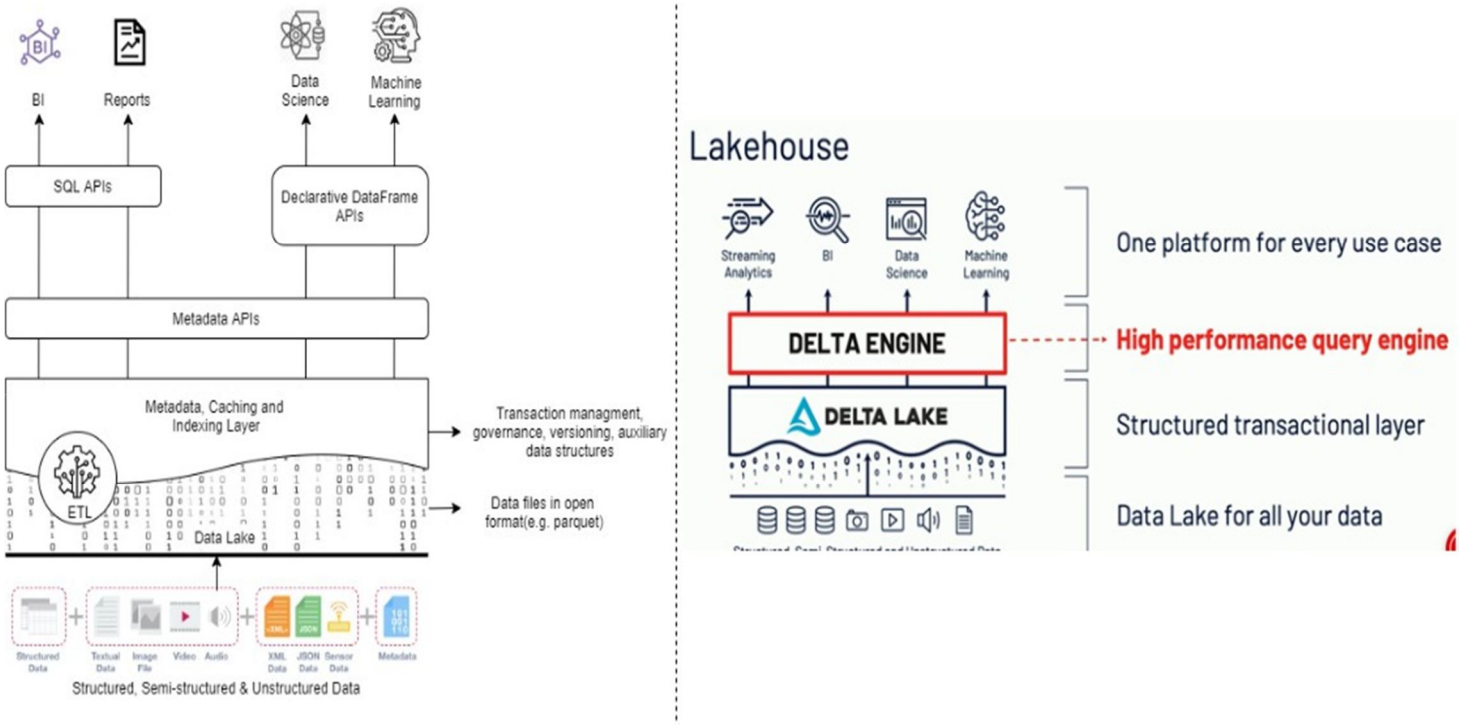
However, some of the key features include:

1) Scalable, Low-Cost Storage & Open Format
2) ACID Transactions
3) Metadata, Caching & Indexing Layer for manageability & performance while querying
4) SQL APIs, Declarative Dataframe APIs for ML and BI support

The initial idea for implementing a Lakehouse is to store data in low-cost storage i.e S3 in Parquet format, but implement a transactional metadata layer on top of the object store that defines which objects are part of a table version. Delta Lake, Hudi and many such systems have added transaction logs in this fashion.
Even though the metadata layer added management capabilities but still the SQL performance was not that good. Data Warehouse started using SSDs to store Hot data and also used indexes for better accessibility.



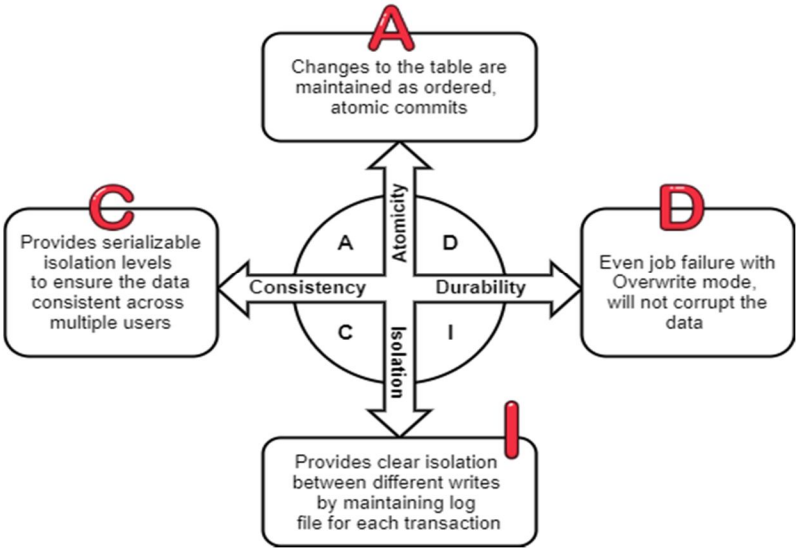
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com



Hence, Databricks came up with Lakehouse using Delta Lake, Delta Engine & Databricks ML runtime projects. And to speed up advanced analytics workload and better data management, lakehouse includes Declarative Dataframe API. However, most of these systems support a DataFrame API for data processing which not only is very easy for developers to implement but also has many optimization techniques. These APIs can thus leverage the new optimization features in a Lakehouse, such as caches and auxiliary data, to further accelerate ML.

D. ACID Transaction

Data lakes typically have multiple data pipelines reading & writing data concurrently and it may lead to data loss. Thus, a data engineer has to follow a tiring process to ensure data integrity. Therefore, Lakehouse with Delta Lake makes your data lake **ACID-compliant** meaning data stored inside the data lake has guaranteed consistency. Hence, Delta Lake is considered to be a robust data store, whereas a traditional data lake is not.

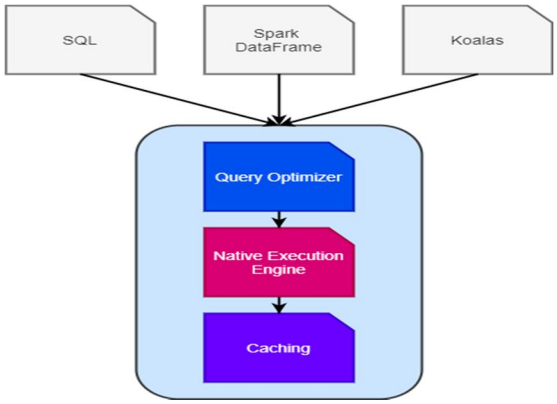


E. Delta Engine

Delta engine is a new high-performance query optimizations engine that is based on the apache spark 3.0 framework and it features a range of efficient ways to process data. Delta Engine speeds up data lake operations, enabling a number of workloads from large-scale processing of ETLs to ad-hoc, interactive queries. Many of these optimizations take place automatically in the data lake by delta engine.

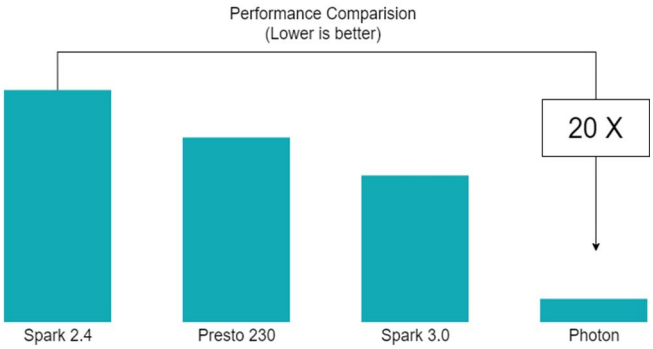
Delta engine uses three different components to accelerate and enhance the performance of delta lake for different workloads that are

1) Query optimizer
2) Native Execution Engine
3) Caching layer



F. Query Optimizer

The query optimizer is used for cost-based optimization, adaptive query execution and runtime filters. It is already included in spark 3.0 with greater advanced statistics to deliver. It enhances the performance up to 18 times for star schema workload.



G. Native Execution Engine

The Native Execution Engine is also called Photon. It is an engine within an engine. It is a completely new engine built for Databricks to enhance the performance of the modern cloud hardware. It was built from scratch in C++ to maximize control over the underlying hardware. It helps to maximize the performance of all the workload types and is completely compatible with spark.

H. Caching Layer

The Caching layer chooses the input data to cache for the user automatically. It does not just dump the bytes cache that caches the raw data, it transcodes the data into a more secure and efficient format which helps the CPU to efficiently decode the query and process the data. This helps us with enhancing the performance with 5-times for all the virtual workloads.

As we now have a better understanding of Lakehouse, let us understand some of its advantages and how to make the best out of it?

III. ADVANTAGES

1) *Elimination of simple ETL jobs:* The data must be loaded into the Data Warehouse to query or to perform analysis in the Data Warehousing technique. For example, loading data to the Data warehouse from Data lake and cleaning and transforming it using ETL/ELT tools. Using the Data LakeHouse tool, the ETL process will be eliminated by directly connecting the query engine to your Data Lake.
2) *Reduced Data Redundancy:* Data LakeHouse removes Data Redundancy. For example, you have data on various tools and platforms, such as cleaned data for processing in the data warehouse, some Business Intelligence tool meta-data, temporary ETL tool data, etc. In order to prevent any data sanity issues, such data need to be monitored and maintained continuously. So, if you use a single tool to process your raw data, data redundancy issues can be solved easily.
3) *Ease of Data Governance:* The organizational overhead of handling data governance on multiple tools can be removed by Data LakeHouse. You have to be careful when transferring data from one tool to another if you are handling sensitive data so that each tool can maintain proper access controls and encryption. But using a single Data LakeHouse tool, data governance can be managed from a single point.
4) *BI Tools Connectivity:* Data LakeHouse enables tools such as Apache Drill to connect directly to some common BI tools, such as Apache Drill (Tableau, PowerBI, etc.). The time taken from raw data to visualization is gradually decreased by exponential times.
5) *Cost Reduction:* Data must be stored in several locations in the Data Lake and warehousing paradigm, and the cost of storage is also high. Comparatively, we can store data in cheap storage such as S3, blob, etc in Data LakeHouse.

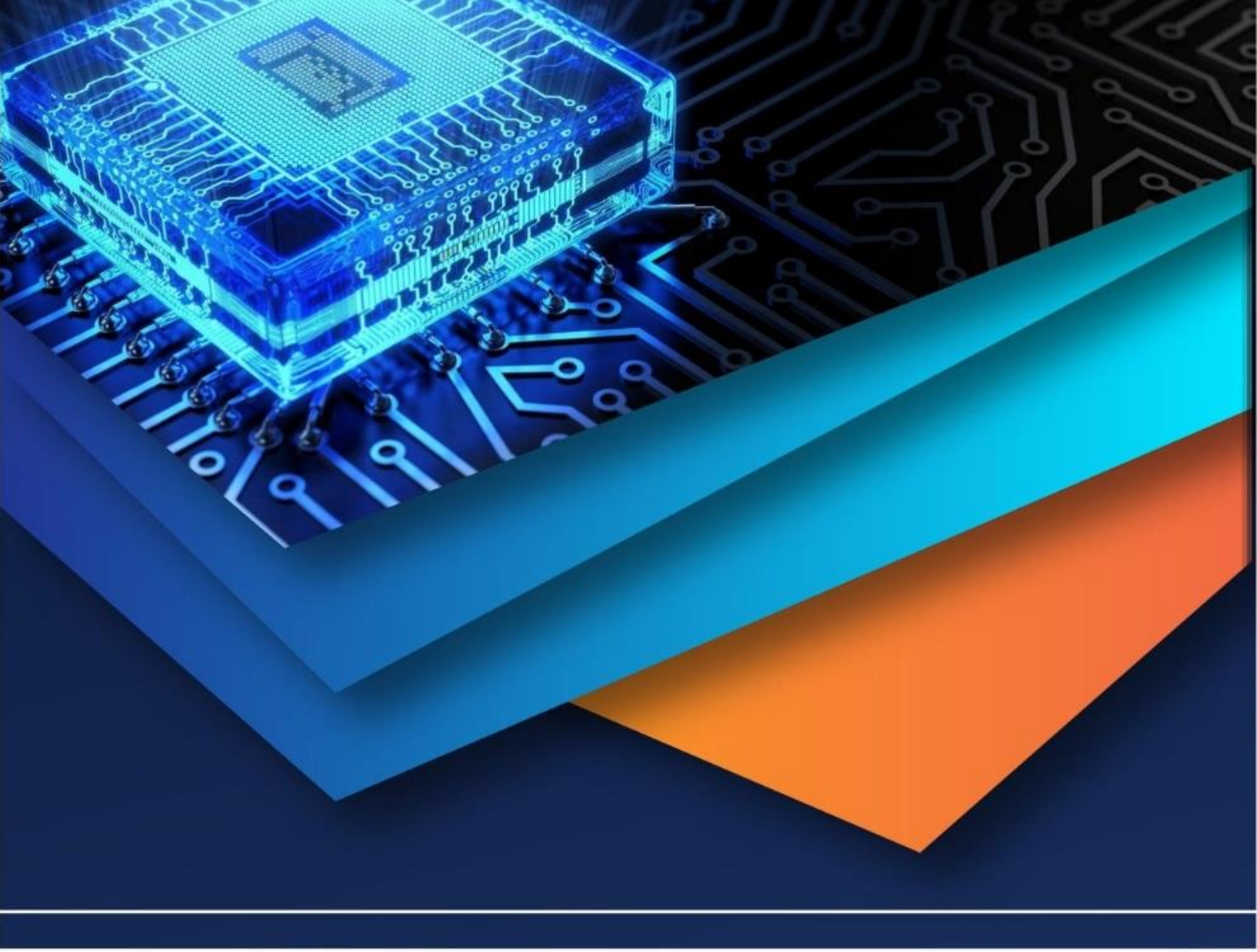
So, these were some of the advantages which help in designing our Data Pipeline better.

IV. CONCLUSION

With unified experience and overcoming the limitations of Data Lake and Data Warehouse, Lakehouse architecture best suits Modern Architecture which not only brings out the best of both the traditional worlds i.e Data Lake & Data Warehouse but also provides ACID compliance and other optimizing features for a better experience. So, to get the best out of your data, it's time to gear up and embrace the future in data processing and management.

REFERENCES

[1] Ben Lorica, Michael Armbrust, Ali Ghodsi, Reynold Xin, and Matei Zaharia - What is Lakehouse?
[2] Joel Minnick - Diving into Lakehouse
[3] Delta Lake Documentation @ https://delta.io/
[4] Databricks Delta Documentation @ https://docs.databricks.com/delta/index.html
[5] Delta Engine @ https://docs.databricks.com/delta/optimizations/index.html
[6] Databricks Official Documentation @ https://docs.databricks.com/index.html
[7] Ali Ghodsi Lakehouse Webinar - Databricks Summit 2020-2021
[8] Simon Whiteley - Achieving Lakehouse models with Spark
[9] Lakehouse @ https://databricks.com/product/data-lakehouse
[10] Spark Summit @ https://databricks.com/sparkaisummit/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY