Journal of Theoretical and Applied Information Technology
31st March 2024. Vol.102. No 6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

STORAGE STRUCTURES IN THE ERA OF BIG DATA: FROM DATA WAREHOUSE TO LAKEHOUSE

MOHSSINE BENTAIB[1], ABDELAZIZ ETTAOUFIK[2], ABDERRAHIM TRAGHA[3], MOHAMED AZZOUAZI[4]

Laboratory of Information Technology and Modeling, Faculty of Sciences Ben M'SIK

Hassan II University, Casablanca, Morocco

E-mail: [1]mohssine.bentaib@univh2c.ma, [2]abdelaziz.ettaoufik@univh2c.ma,

[3]abderrahim.tragha@univh2c.ma, [4]azouazii@gmail.com

ABSTRACT

The amount of data that is available to enterprises today comes from many different sources, including social networks, sensors, and IoT devices. In order to discover trends, draw conclusions, produce projections, and make informed decisions, this enormous amount of data needs to be stored across a variety of platforms for processing and analysis. The capacity of conventional EDs is surpassed by the quantity and quality of data that is being collected. To accomplish this, businesses with current data warehouses must pick a storage architecture with enough storage and processing power for this kind of data. They must choose one of the following options: The data warehouse can either (i) develop into a big data warehouse, (ii) be replaced by a data lake, or (iii) be combined with a data lake to create a data LakeHouse. In this article, we aim to find the best choice for the storage of varied and voluminous data. To do this, we examine the big data warehousing literature. After doing a comparison of the various architectures put forth, we draw a conclusion outlining the optimum storage practice.

Keywords: *Data Warehouse, Big Data, Big Data Warehouse, Data Lake, Data Lakehouse*

1. INTRODUCTION

Huge volumes of heterogeneous data have been produced as a result of the widespread usage of new technologies [1]. Organizations must deal with massive amounts of data from many sources and in various formats as a result. In order to create predictions, come to conclusions, and take wise judgments, they must process and analyze all the data, which necessitates a platform with the required capabilities and features [2]. Data warehouses are utilized mostly with massive datasets produced in various legacy systems using relational data, and they constitute a traditional domain of relational databases [3]. They get analytical data via analysis and reporting tools and are fed from various data sources via ETL. Because of the limitations imposed by data warehouses, analytical tools fall short of what analysts demand in terms of high availability and quick responses to queries [4].

Due to these restrictions, organizations are forced to move to a big data platform that offers unlimited storage capacity and supports a variety of data formats.

Because of this obligation, we ask ourselves the following questions: What role will the data warehouse play in the age of Big Data? Should the company permanently stop using the data warehouse? What is the impact of investing in a data warehouse even if the organization already has a big data platform? An in-depth analysis of the different solutions offered by companies that currently have a data warehouse is necessary to find the answers to these questions.

Numerous architectures are found in the literature. The data warehouse has been replaced by the big data warehouse, the data warehouse has been abandoned in favor of the data lake, and the two have been combined into a new tool called the LakeHouse [5].

In this paper, we answer these questions by presenting a comparative study of the new architectures that are replacing the traditional data warehouse.

The state of the art for the data lake, large data warehouse, and LakeHouse is presented in the following section. In the third section, we outline the many designs that outline excellent storage methods and offer a comparison of their individual traits. In section 4 follows we provide a synthesis, and we discuss open research's difficulties and potential

future prospects in the fifth section. The final section is when we put our labor to rest.

2. LITERATURE REVIEW

The many structures used to store, process, and analyze vast amounts of data are highlighted in this section. It provides Data Lake, Big DW and LakeHouse literature as well as a study that details and contrasts the various platforms' varied features.

2.1 Data lake

The literature on data lakes is a little murky and lacking, and the numerous implementation strategies that have been proposed do not completely address the topic of data lakes or provide a detailed design and implementation strategy [6]. The available literature discusses certain details and tangible traits of data lakes, but it does not offer a consistent idea or overarching implementation plan.

Studies show that customers save all of their unprocessed, raw data—whether it's unstructured, semi-structured, or structured data—in a single, central location called the data lake [5].

James Dixon, the Chief Technology Officer (CTO) of Pentaho, first used the phrase in 2010 to describe the idea of a single repository collecting practically infinite amounts of raw data for analysis or indefinite future usage [7], [8]. Consumers can use specifically created schemas to query the pertinent data, resulting in a smaller collection of data that can be studied to help answer queries because each data entity in the data lake is connected with a unique identifier and a substantial amount of metadata [9].

When producing or analyzing data, data models and schemas are employed, but not when storing information [10]. The data lake is described by Terrizzano I. et al. [7] as a collection of central repositories housing substantial amounts of raw data in various forms, described by metadata, arranged into recognizable datasets, and accessible on demand. Similar to this, Hai et al. [11] define data lakes as big data repositories that hold raw data and offer on-demand integration features utilizing metadata descriptions.

Data lakes in the context of big data provide extensive and flexible data storage that may accept many data formats. In spite of the trade-offs made while storing data in conventional designs like a data warehouse, they store nearly accurate or even exact copies of the source format to give an unpolished view of the data [12]. There is no attempt made to model or integrate the data before storage. The goal of the data lake is to make data available to other organizations for use in the future, like data analysis [13]. It can serve as a setting for the development of in-depth analyses with the goal of making quick, accurate decisions based on raw data. Additionally, it is the perfect response to issues with data integration and accessibility.

There are several benefits for using data lakes to store raw data. Four benefits are highlighted by Marilex R. L. [8]: enhanced data collecting, quick access to raw data, reduced initial effort through data storage, and data preservation. The main uses case of data lakes is as experimentation platforms for data scientists or analysts, staging areas or sources for data warehouses, and as a direct source for self-service BI. Figure 1 shows data lake architecture.

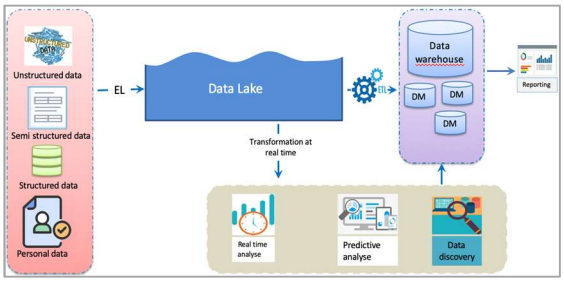

Figure 1. Data lake architecture

2.2 Big data warehouse

The traditional Data Warehouse and Hadoop are combined in the hybrid design known as Big Data Warehouse, which can be a substantial benefit in terms of data processing, multidimensional processing, and decision-making maturity [14].

Although data warehouse and big data are two distinct concepts, they work quite well together. Data warehouses are architectural descriptions of how data is organized, whereas big data is a technology connected to the storage and management of vast and varied amounts of data.

A big data warehouse is a key component of the organization and management of big data. It is a hybrid system that employs both big data technologies currently available and data warehouse design. A Big Data warehouse can be implemented as a first step in enhancing an organization's data analytics infrastructure and starting to apply Big Data technologies [15]. By fusing data warehouse analysis with big data analysis, the big data warehouse makes it possible to quickly analyze a lot of data.

Many businesses across a variety of industries are currently working to modernize their data analytics infrastructures for this new era by switching from the traditional data warehouse (DW) concept to a new

notion of data warehouse (BDW) based on a more dynamic data model [15], [16].

The large data warehouse, according to Forrester, is "A specialized and consistent set of data repositories and platforms used to support a wide variety of analytics run on-premises, in the cloud, or in a hybrid environment" [17]. The big data warehouse makes use of both established and emerging technologies, including Hadoop, columnar and row data warehouses, streaming, ETL, and elastic storage as well as in-memory frameworks [17].

Data types and formats are a significant issue right now since they contradict the core tenets of data warehouse operations. In fact, spatial data, photos, videos, and simple text cannot be stored in data warehouses. The design and implementation of big data warehouses is developing into an important area of research as a result of the contemporary conceptual, technological, and organizational setting. The literature on this subject is divided into three sections.

The first category includes works that address big data warehouses' physical design [18]–[23], the second category includes works that address big data warehouses' query processing and optimization [24]–[28], and the third category includes works that address both axes simultaneously [16], [29]. In order to demonstrate the significance of big data warehouses in information systems, we quote a few studies in this document that deal with the topic.

A new data placement approach for Hadoop's distributed data warehouses, dubbed Smart Data Warehouse Placement (SDWP), is proposed in the work published in [30]. On the other hand, in [31], the authors suggest a useful tool for heterogeneous data warehouses' data administration and integration. They go over the technologies and architectural frameworks necessary for large data processing, the back-end application that carries out the data migration from the RDBMS to the NoSQL data warehouse, the structure of the NoSQL database, and how it can be useful for upcoming data analysis.

In contrast, another study uses partitioning and compartmentalization techniques to construct a denormalized model-based Big Data warehouse [29]. For their part, authors propose in [25] a novel strategy for data integration in Big Data warehouse. This method, known as Mapping-ELT (M-ELT), is founded on the processing of fundamental ELT operations and takes semantic heterogeneity into consideration.

The work released in [32] makes a fresh suggestion for the conception and application of big data warehouses in the context of smart cities. The suggested method considers the gathering, preparing, and enrichment of data that arrives in batches and via flow mechanisms, as well as the output of data mining algorithms and simulation models [32]. In a different study [28] a strategy for query optimization in massive data warehouses is adopted. The suggested method chooses a group of materialized views to target the physical structure of big data warehouse.

Nuno Silva et al. [15] chose a strategy to implement large ED in the supply chain, on the other hand. They provided the technological and logical architectures required for its implementation.

The authors provide a novel framework for data warehouse queries that consists of a storage model and a tailored query processing model, despite the fact that a query in ED can be broken up into a huge number of separate subtasks and managed by a large-scale computing cluster. To optimize OLAP queries with star joins, Y. Ramdane et al. [30] suggested a data storage model in Hadoop. The selected model offers a fresh approach to data placement in the Apache Hadoop environment that enables a star join operation to be completed in a single Spark transaction.

Currently, the design of big data warehouses places equal emphasis on the logical and physical layers, which are represented by the data models and infrastructure, respectively [33], [34]. The concept is new, as evidenced by the state of the art. The article introduces two modeling approaches for storage and processing in the context of large data warehouses [33], [35], [36]. The first approach, dubbed "lift and shift," entails expanding the capabilities of conventional data warehouses using big data technologies like Hadoop and NoSQL databases. The second tactic, known as "rip and replace," suggests a scenario in which big data technologies totally replace a conventional data warehouse. Figure 2 illustrates the general architecture of the big data warehouse proposed in [37].

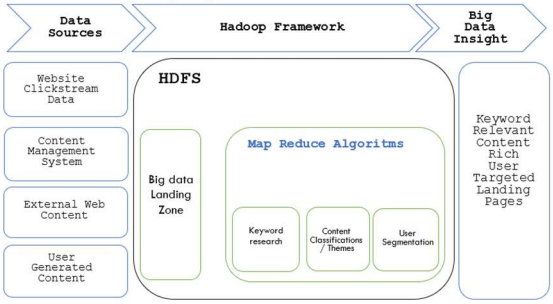

Figure 2. Conceptual big data warehouse architecture [37]

3. BENCHMARKING STUDY

The continual generation of vast amounts of data by today's digital information systems necessitates the implementation of platforms with the ability to store and handle massive data, while also taking into account its volume, speed, diversity, and validity [38]. Data management technologies have therefore progressed from structured databases to big data storage systems, massive data warehouses, and data lakes, but each solution has advantages and disadvantages.

We give a comparison of these various architectures in the section that follows.

3.1 Data lake vs. data warehouse

Although data lakes and warehouses are used to store an organization's data, each has benefits and drawbacks.

- Data. Data lakes and data warehouses store different types of data and analyze it in different ways. Data lakes contain data in its unprocessed state, devoid of any schema or structure. This makes it possible to store a wide range of data types in a single location, including social network posts, log files, pictures, and videos. Additionally, a lot of enterprise data is unstructured, which data warehouses cannot handle.
- Architecture. Data lakes employ a flat design that makes it easier to add and remove such a data source, in contrast to data warehouses that push data to the user in the form of data marts in accordance with a predefined format. There are metadata tags and a specific identification for each data element. Although the precise specified structure for handling various types and forms of data is not required for data lakes, the order of data arrival time must be maintained [39]. While, the presence of a comprehensive collection of substantial metadata ensures the agile and effective management of the data stored in it. These enable the utilization of the data contained in the data lake in a very flexible and simply adjustable manner [40].
- Processing. Schema-on-write refers to a behavior in which data that is destined for the data warehouse must be processed in order to assign it to a structure in accordance with a specified model. Schema-on-read refers to the practice of processing and modeling data at read time while it is still in its raw form and intended for the data lake [39].
- Access. Although data lakes are open to all users, only data scientists are equipped to do in-depth analyses on the lake's data. Data warehouses, on the other hand, are utilized by specialized business users to create reports and extract analytical data, but they don't satisfy data scientists who need to venture outside the data warehouse's limits to gather additional data for analysis.
- Security. Since data warehouses have been around for close to 30 years, they are more secure thanks to their experience and maturity. They implement role-based access privileges and fine-grained security policies. This method ensures efficient user access management while enabling the construction of sophisticated user access models. Despite the fact that data lake security is still under development, these assurance gaps result from the fact that current data lakes concentrate on storing heterogeneous data without considering how or why data is utilized, managed, defined, or secured [41]. As a result, this subject has been the subject of various works' research [42].
- Agility. The structured data kept in the data warehouse. This results in low agility because any change that affects the data warehouse model necessitates a reconfiguration of the data warehouse. Data lakes, on the other hand, do not adhere to any structure and as a result, have a fixed configuration.
- Cost. Since data lakes don't need as much organization and structure and don't need additional hardware or software, they are typically less expensive to set up and maintain than data warehouses.

Despite their tremendous workload, relational data warehouses have long dominated analytics and decision-making. However, the development and variety of big data have outpaced its structured data integration approach. Due to their nature of design and poor tolerance for human error, these systems are therefore very dependent on IT. Data lakes are an addition to or a replacement for data warehouses, not the other way around. Data lakes should be viewed as extensions of the BI infrastructure as a result [8].

3.2 Big data warehouse vs Data warehouse

The volume, diversity, and velocity of big data severely restrict the utilization of traditional data warehouses. Emerging methods and technologies are made possible by their rigid relational nature, expensive scalability, and occasionally ineffective performance. The idea of big data warehousing is currently growing in acceptance because it provides fresh approaches to tackling big data problems [35]. It has also drawn considerable interest from the scientific community, highlighting the necessity of

redesigning the conventional data warehouse in order to achieve new features applicable in big data environments [34].

Despite the fact that both the data warehouse and the big data warehouse are used to store data, they are distinct in the following ways:

- Data. Only consistent data that is organized using a certain model is stored in traditional data warehouses. Big data warehouses, on the other hand, hold both structured data and heterogeneous raw data, including sensor data, audio, video, image, and json files.
- Volume. The volume of data that each type of warehouse can hold is one of the key distinctions between them. The volume, diversity, and velocity of big data are too great for traditional data warehouses, which are made to manage vast amounts of structured data. massive data warehouses are made to manage massive data from many sources and have storage capacity that exceeds petabytes.
- Analytics. Big data warehouse architecture makes advantage of cutting-edge AI-based analytics. By evaluating data from many sources, it gives organizations a thorough and in-depth perspective of their business, enabling them to make the necessary predictions and enhance system performance. Traditional data warehouses provide analytical data as well, but only on the basis of sparse data. As a result, they do not permit the use of sophisticated instruments that need a substantial amount of data, which means that the analytical data generated falls short of fully revealing the company's business process.
- Flexibility: Both kinds of data warehouses give stakeholders access to analytical data, but big data warehouses are favored because they generate insights that transcend the enterprise and address many categories of decision-makers.
- Cost: Compared to a big data warehouse, a standard data warehouse may be more expensive to install and operate. Traditional data warehouses need specialized gear and software, and before the data can be used, it must be converted and structured. On the other hand, because it doesn't need the same amount of structure and organization, a massive data warehouse is typically less expensive to setup and operate.

3.3 Big data warehouse vs Data lake

Organizations utilize data lakes and big data warehouses as storage areas and ways for handling enormous amounts of data to aid in data analytics and decision-making. While both approaches have advantages, organizations must weigh the pros and downsides of each before deciding which is best for their information systems.

- data processing. A data lake is a centralized location created for the large-scale archival of organized, semi-structured, and unstructured data. The fact that data in a data lake is often kept in its original format. Because of this, data lakes are perfect for businesses that need to store and analyse massive amounts of data from many sources while also keeping the raw data for usage in the future. A big data warehouse, on the other hand, is a particular kind of data warehouse created to manage big data. It is described as an ETL process that involves erasing, customizing, reformatting, integrating, and inserting data into a conventional data warehouse [43]. As a result, it is designed to store and handle huge amounts of organized, semi-structured, and unstructured data. In contrast to a data lake, a big data warehouse often transforms, purifies, and organizes data to support analysis.
- Data control. The degree of control over the data is one of the key contrasts between a data lake and a big data warehouse. There is less control over the data because it is stored in its raw form in a data lake. Because of this, data lakes are perfect for businesses that need to store a lot of data, but they are less appropriate for mission-critical applications where data integrity and dependability are crucial. A big data warehouse, on the other hand, offers a high level of control over the data through access control, data governance, and clearly defined data architectures. Big data warehouses are therefore the best option for businesses that require structured and organized data for analysis and decision-making.
- Flexibility and scalability. Their levels of flexibility and scalability are another contrast between the two. With less control over the data, data lakes give greater freedom and autonomy. In contrast, big data warehouses typically have more strict data management policies, including access control, data governance, and clearly specified data architecture. Big data warehouses are often more expensive to create and maintain than data lakes. This is because less specialized gear and software are needed for data lakes, and the data is not processed before usage. Contrarily, big data warehouses typically cost more to set up and maintain because they need specialized hardware and software and because

the data must first be converted and sorted before it can be used.

– Extraction velocity. The speed of data extraction between the two is another important distinction. Because the data must first be converted and organized before it can be evaluated, data extraction from a data lake can be slower than data extraction from a massive data warehouse. On the other hand, since the data has already been processed, cleansed, and organized, extracting data from a massive data warehouse is typically quicker.

The decision between a data lake and a big data warehouse ultimately comes down to the particular requirements of the enterprise. Organizations that need to store significant amounts of data from numerous sources and want to keep corporate data are best served by data lakes. Both disk big data warehouses are excellent choices for businesses that need to rely on a lot of data to make decisions.

3.4 Data LakeHouse

Data LakeHouse is a newly coined phrase in the realm of massive data processing and storage [44]. It is a data management system that relies on low-cost direct access storage as well as typical DBMS administration and performance features including ACID transactions, management data versions, auditing, indexing, caching, and query optimization [44]. Data LakeHouses are seen to be especially well suited for cloud systems with independent computation and storage capabilities, where various compute applications can operate on entirely different compute nodes as needed [44].

Data LakeHouse serves as a central location for handling and storing huge amounts of unstructured and organized data. It provides a new design that fuses the finest qualities of data lake systems, such scalability and flexibility, with those of data warehouses, like structuring and organizing. It keeps data in its unprocessed state as well as data that has been transformed, cleaned up, and structured, making it easier to utilize and analyze for decision-making. The LakeHouse, therefore, serves as an alternative to both systems as shown in figure 3.

Many businesses employ multiple data warehouses and a huge data lake nowadays, removing redundant data from the two systems and enhancing data quality. Since the data lake and ETL tool are linked in this instance, a pipeline is created between the unsorted lake layer and the integrated warehouse layer. Two different routes can be taken by the heterogeneous data that has amassed in the data lake: (i) Data handled in real time by intelligent tools like machine learning and data science. (ii) Data that has been imported into the ETL, processed, and transformed into analytical data.

There are many benefits of using data LakeHouse. For data management, it offers a lone source of truth. In contrast, raw data is stored in a traditional data lake, which might result in data silos and inconsistent data. Data in a traditional data warehouse is controlled and structured, but as data volume increases, it can be challenging to scale and manage. Data silos are eliminated, and data consistency is guaranteed, thanks to a LakeHouse's unified platform for managing data.

Performance-wise, the data LakeHouse offers quick and effective data extraction. This is supported by the fact that the data is of high quality and suitable for usage and analysis due to its recent transformation, purification, and organization. According to John Kutay [45], LakeHouses have the following qualities: lessened data redundancy, cost-effectiveness, support for a larger range of workloads, simplicity of versioning, governance, and data security.

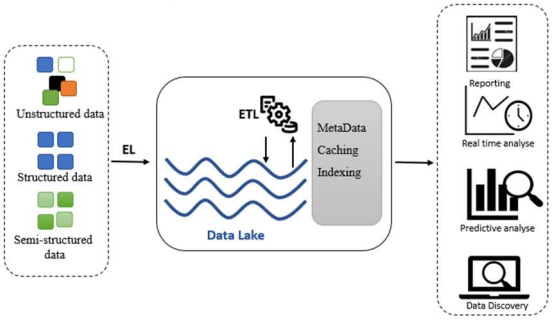

Figure 3. Data LakeHouse architecture

3.4.1 An overview of systems supporting the LakeHouse architecture

a. Delta lake

Delta Lake is a new generation of data storage options that transforms the data lake into a LakeHouse. It combines continuous and batch data processing on existing data lakes, including S3, ADLS, GCS and HDFS, and stores transaction logs and data files in a single directory. It also offers scalable metadata management and supports ACID transactions [46]. It stores information on object stores for further processing by Apache Spark. The analytical capabilities of data warehouses are combined with quick processing and inexpensive storage in this approach. Data from Delta tables is typically kept in the data lake as Parquet files, which streamlines selection queries. For considerably faster ACID properties, time travel, and metadata operations for huge tabular datasets, Delta Lake also makes use of a compacted transaction log in Apache Parquet format [47]. The Delta engine is a

component of Delta Lake, which enhances the efficiency of Spark SQL, Databricks SQL, and DataFrame operations and optimizes queries for big data.

b. Apache Iceberg

Iceberg is a high-performance format for massive analytical tables. By bringing the dependability and simplicity of SQL tables to Big Data, Iceberg enables engines like Spark, Trino, Flink, Presto, Hive, and Impala to work securely with the same data, at the same time [48]. For huge tables, Iceberg was made. In real-world applications, where a single database may hold tens of petabytes of data, it is employed because even extremely large tables can be read without the aid of a distributed SQL engine. Iceberg was created to address potential consistency issues in cloud object stores.

Using immutable file formats like Parquet, Avro, and ORC, Iceberg describes how to manage extensive analytical spreadsheets. All data is kept in many files, including: 1) The snapshot metadata file includes information on the table, including the table schema, section details, and the manifest list path. 2) There is an entry for each manifest file connected to the snapshot in the manifest list. 3) The manifest file includes a list of the locations of the linked data files. 4) The information is stored in a physical data file that is written in formats like Parquet, ORC, and others [49].

c. Apache Hudi

Similar to Apache Iceberg and Delta Lake, Apache Hudi (Hadoop Upserts Deleted Incrementals) is a framework made to speed up incremental processing on top of data file systems. In situations where only data collected over a period of time should be recovered, Apache Hudi focuses on stream data optimization and capturing data changes to speed up streaming data intake and analysis. By processing just fresh data and avoids reprocessing old data, incremental processing aids in improving query performance [50].

Hudi offers two methods for changing data tables: copy on write and merge on read [51]:

The Copy-On Write (CoW) technique locates the records that need to be updated in the files and eagerly rewrites them to new files with the changed data, resulting in a high write amplification but no read amplification.

Merge-On-Read (MoR) technique doesn't require rewriting of any files. Instead, it delays the reconciliation until query time and sends out information about record-level changes in other files, resulting in little write amplification.

These three storage options address a number of issues that arise frequently while working with data lakes [52]: (i) atomic transactions, which make sure that the data is not left in an inconsistent state if an operation fails; (ii) consistent updates, which stop reads from inconsistent states; and (iii) scalability for the data and metadata. Furthermore, they all provide comparable functionality including upserts, deletes, transaction support, time travel, SQL read/write, streaming ingestion, metadata scalability, and many more. Since Apache Spark is the main need of the platform, all of these storage systems are essentially comparable in that they allow write and read operations from Spark.

Despite the many similarities between these three storage systems, the Delta Lake has consistently come out on top in comparison studies, especially in terms of performance and integration [51], [53]–[55]. Table 2 lists the findings of a few earlier investigations.

Table 2: Comparison Between Delta Lake, Apache Iceberg And Apache Hudi.

Feature	**Delta Lake**	**Apache Iceberg**	**Apache Hudi**
Data model	Log-based	Table-based	Log-based
Storage formats	Paquet	Paquet, ORC	Paquet, AVRO
Upsert Support	Basic	Basic	Advanced
ACID Compliance	Yes	Yes	Yes
Time Travel	Yes	Yes	Yes
Integration	Very good	Limited	Good
Compaction	Yes	No	Yes
Object storage	Yes	Yes	No
Caching	Yes	No	No
Evolution	Yes	Yes	Yes
Performance	Best	-	-

3.4.2 Data LakeHouse Frameworks

Plusieurs frameworks peuvent être utilisées pour les Data LakeHouse. Bien que le HDFS représente le framework le plus largement utilisé, il existe d'autres systèmes plus flexibles, comme Amazon S3.

Dans [56], les auteurs réalisent une étude comparative entre les deux Framework de stockage en comparant le coût, l'élasticité, la SLA (disponibilité et durabilité), la performance et l'écriture transactionnelle. Les auteurs concluent que le stockage S3 et cloud offre une élasticité, avec une disponibilité et une durabilité d'un ordre de grandeur supérieures et des performances 2 fois supérieures, à un coût 10 fois inférieur à celui des clusters de stockage de données HDFS traditionnels. Cependant, avec S3, toutes les lectures doivent passer par le réseau, ce qui interdit l'optimisation des performances, ce qui représente un sérieux inconvénient.

3.5 Synthesis

We have performed a literature review on various data storage, processing, and analysis architectures in the previous section. The properties of the different storage architectures were then compared. The decision to use such a design ultimately depends on the specific needs and goals of the information systems, because each, despite its ability to store, process, and analyze data, has unique advantages and disadvantages.

According to the comparative study presented in Table 1, the lakehouse and the big data warehouse have the same characteristics, which means that the lakehouse can be considered as a big data warehouse. On the other hand, LakeHouse offers the scalability and flexibility of data lakes while maintaining the structure and control of data warehouses. Therefore, LakeHouse remains the best choice for businesses that need to process, store and analyze huge amounts of structured, semi-structured and unstructured data in light of the comparative data shown in Table 1.

4. DATA LAKEHOUSE OPTIMIZATION TECHNIQUES

Organizations can store vast amounts of unstructured, structured, and semi-structured data in a Lakehouse, which combines aspects of data lakes and data warehouses and enables quick, scalable analytics. Additionally, it inherits robust governance and auditability from data warehouses as well as streaming workloads from data lakes [57].

It is essential to optimize performance in a Lakehouse setting, which entails enhancing query performance, cutting expenses, and ensuring that data-driven insights are readily available [57].

Several methods are employed in this situation; we list the most popular ones here:

- Management of Metadata [58]. In order to assure affordable storage without sacrificing governance and management features, The LakeHouse uses a transactional metadata storage layer on top of the cloud object store [58]. Metadata includes details on the data stored in Lakehouse, such as statistics, data graphing, and schema. Data discovery and accessibility are made easier by a well-structured data catalog with adequate metadata tagging and search capabilities. Users may find the information they require fast, saving time spent looking for pertinent data.
- Indexing. By removing unnecessary data, indexing primarily aims to reduce the time it takes for queries to execute. Global and local indexes are the two types of indexes that are employed. Because the two types are unrelated, a system may contain both or just one index, and the index types may vary depending on the level [59]. Some systems solely use local indexes located in the slave nodes in a distributed environment, while others decide to add a global index located in the master node to speed up local query processing and reduce the number of trips to the master node [53]. By enabling direct access to particular rows or columns without having to scan the entire dataset, indexes can dramatically improve query performance. Thoughtful analysis of the tradeoffs between query performance and storage costs is necessary for index administration.
- Data compaction and pruning [60]. Data lakes can amass a considerable volume of historical data over time. By eliminating or consolidating duplicate or obsolete data, data pruning and compaction procedures assist save storage costs and enhance query performance. Data preservation regulations and temporal division are two common techniques.
- Caching. Keeping frequently accessed data or query results in memory helps speed up responses to repeated searches. When cached files are still accessible for reading can be simply determined by running transactions. A transcoded format for the cache is another option, which is more effective for query engine execution [57]. In order to do this, in-memory caching can be especially useful for workloads that involve a lot of reading because it reduces the need to contact the underlying storage.
- Parallelism. greatly facilitates the creation and administration of query workloads on clusters.

To prevent resource conflicts and provide constant performance, managing concurrent queries and resource allocation is essential.

- Performance monitoring. Gathering metrics on how queries are executed and query profiling might reveal performance bottlenecks and potential areas for improvement. It assists in locating resource-intensive or lengthy queries that can benefit from optimization.
- Partitioning. One of the most used approaches of optimization is partitioning. It provides easy accessibility, better scalability, and less CPU resource usage. When partitioning data in a distributed environment, it is important to consider the types of frequent queries that will be applied to the data as well as the processing demands [53]. Since each data partition will be assigned to a compute node that will perform a specific portion of the query, this will also minimize inter-node exchanges and lower the quantity of data visited throughout the processing phase.

  There are two different types of partitioning described in the literature: space partitioning and data partitioning [61]. The analysis performed on these data typically focuses more on geometric objects than qualities, therefore the first type entails combining spatial data that is geographically close together into the same partitions. The data distribution pattern in the cluster is carried by the disk as well as the data partitioning. Three data partitioning techniques, STR, STR+, and K-d Tree, were mentioned by the authors in [61].

We carried out our experiments with Delta Lake in a distributed environment. For this, we used an AWS S3 object store, using the 8.3 runtime based on Scala 2.12, Spark 3.1.1. We configured 4 workers including the supervisor (driver). We loaded an 800GB csv file for testing.

First, we loaded and executed a load of 100 different queries and measured performance before and after partitioning. Then we varied the query load to assess the impact of incremental partitioning on performance.

The figure shows the results obtained.

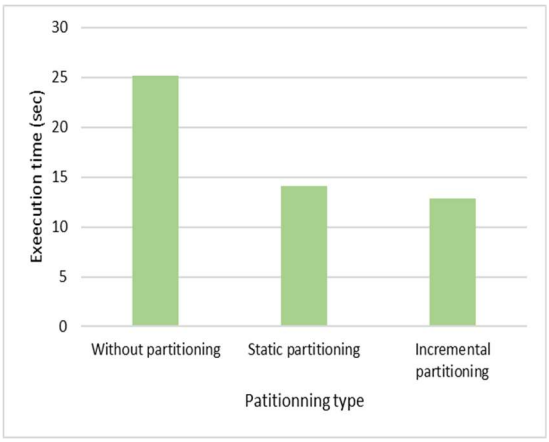

Figure 4. Load query performance peer partitioning type

5. OPEN RESEARCH CHALLENGES AND FUTURE DIRECTIONS

Big Data has properties that are beyond the scope of conventional approaches, especially when data is kept in a distributed setting that necessitates the use of parallel processing tools like the MapReduce paradigm. Due to these restrictions, new methodologies with specific features and enhanced capabilities have emerged, such Hadoop Distributed File System (HDFS), Cassandra, and MongoDB. High availability and large-scale data processing are also capabilities of these scalable systems. Processing is a crucial component of the Big Data universe at the storage stage. It entails processing the data necessary to get it ready for the following step. New processing technologies like Hadoop and Spark have been created in response to the functional limitations of conventional systems. These solutions allow businesses to swiftly, effectively, and concurrently process enormous amounts of data.

The analysis phase is the last step, where data analysis is done in order to make informed conclusions. In this context, a variety of analysis tools are employed, including capabilities that let analysts create interactive dashboards that give businesses a holistic view of the market.

Researchers are faced with a challenge while trying to improve any of the three phases outlined above.

Although big data analysis also uses machine learning and artificial intelligence (AI), we intend to propose a new architecture for the optimal storage, processing and processing of big data. To do this, we intend to create an intelligent architecture that merges LakeHouse's capabilities with machine learning and artificial intelligence. Our vision will enable AI-based incremental partitioning of LakeHouse data and metadata.

6. CONCLUSION

The data warehouse continues to play a key role in business intelligence (BI), even as big data technologies drive data processing. As a result, it is possible to create a variety of hybrid designs, such as Data LakeHouse, by combining Big Data technologies with traditional data warehouses. This new technology integrates two key components, data processing and BI maturity.

We have discussed various big data storage and processing architectures. We also compared the main characteristics of the different architectures. Our comparative study allows us to conclude that data LakeHouse today represents the best choice for companies needing to process, store and analyze enormous quantities of raw, structured and semi-structured data.

In our experimental study, we demonstrated the remarkable impact of data partitioning on system performance. We also studied two types of partitioning techniques, namely static and incremental partitioning.

In our future work, we intend to include optimization techniques to improve the performance of Data LakeHouse which may degrade as a result of the exponential increase in the volume of data injected into Data LakeHouse.

REFERENCES:

[1] M. Bala, O. Boussaid, et Z. Alimazighi, « A Fine-Grained Distribution Approach for ETL Processes in Big Data Environments », *Data Knowl. Eng.*, vol. 111, p. 114-136, sept. 2017, doi: 10.1016/j.datak.2017.08.003.

[2] W. X. B. Granda, F. Molina-Granja, J. D. Altamirano, M. P. Lopez, S. Sureshkumar, et J. N. Swaminathan, « Data Analytics for Healthcare Institutions: A Data Warehouse Model Proposal », in *Inventive Communication and Computational Technologies*, G. Ranganathan, X. Fernando, et Á. Rocha, Éd., in Lecture Notes in Networks and Systems. Singapore: Springer Nature, 2023, p. 155-163. doi: 10.1007/978-981-19-4960-9_13.

[3] Z. Bicevska et I. Oditis, « Towards NoSQL-based Data Warehouse Solutions », *Procedia Comput. Sci.*, vol. 104, p. 104-111, janv. 2017, doi: 10.1016/j.procs.2017.01.080.

[4] L. Oukhouya, A. E. Haddadi, B. Er-raha, et H. Asri, « A generic metadata management model for heterogeneous sources in a data warehouse », *E3S Web Conf.*, vol. 297, p. 01069, 2021, doi: 10.1051/e3sconf/202129701069.

[5] A. Nambiar et D. Mundra, « An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management », *Big Data Cogn. Comput.*, vol. 6, nº 4, Art. nº 4, déc. 2022, doi: 10.3390/bdcc6040132.

[6] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, et B. Mitschang, « Leveraging the Data Lake: Current State and Challenges », in *Big Data Analytics and Knowledge Discovery*, C. Ordonez, I.-Y. Song, G. Anderst-Kotsis, A. M. Tjoa, et I. Khalil, Éd., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, p. 179-188. doi: 10.1007/978-3-030-27520-4_13.

[7] I. G. Terrizzano, P. M. Schwarz, M. Roth, et J. E. Colino, « Data Wrangling: The Challenging Yourney from the Wild to the Lake. », in *CIDR*, Asilomar, 2015.

[8] M. R. Llave, « Data lakes in business intelligence: reporting from the trenches », *Procedia Comput. Sci.*, vol. 138, p. 516-524, janv. 2018, doi: 10.1016/j.procs.2018.10.071.

[9] C. Walker et H. Alrehamy, « Personal Data Lake with Data Gravity Pull », in *2015 IEEE Fifth International Conference on Big Data and Cloud Computing*, août 2015, p. 160-167. doi: 10.1109/BDCloud.2015.62.

[10] R. L. Grossman, « Data Lakes, Clouds, and Commons: A Review of Platforms for Analyzing and Sharing Genomic Data », *Trends Genet.*, vol. 35, nº 3, p. 223-234, mars 2019, doi: 10.1016/j.tig.2018.12.006.

[11] R. Hai, S. Geisler, et C. Quix, « Constance: An Intelligent Data Lake System », in *Proceedings of the 2016 International Conference on Management of Data*, San Francisco California USA: ACM, juin 2016, p. 2097-2100. doi: 10.1145/2882903.2899389.

[12] G. Phillips-Wren, L. S. Iyer, U. Kulkarni, et T. Ariyachandra, « Business Analytics in the Context of Big Data: A Roadmap for Research », *Commun. Assoc. Inf. Syst.*, vol. 37, 2015, doi: 10.17705/1CAIS.03723.

[13] J. Ziegler, P. Reimann, F. Keller, et B. Mitschang, « A Graph-based Approach to Manage CAE Data in a Data Lake », *Procedia CIRP*, vol. 93, p. 496-501, janv. 2020, doi: 10.1016/j.procir.2020.04.155.

[14] M. E. Houari, M. Rhanoui, et B. E. Asri, « Hybrid big data warehouse for on-demand decision needs », in *2017 International Conference on Electrical and Information*

Technologies (ICEIT), nov. 2017, p. 1-6. doi: 10.1109/EITech.2017.8255261.

[15] N. Silva *et al.*, « Advancing Logistics 4.0 with the Implementation of a Big Data Warehouse: A Demonstration Case for the Automotive Industry », *Electronics*, vol. 10, nº 18, p. 2221, sept. 2021, doi: 10.3390/electronics10182221.

[16] V. M. Ngo, N.-A. Le-Khac, et M.-T. Kechadi, « Designing and Implementing Data Warehouse for Agricultural Big Data », in *Big Data – BigData 2019*, vol. 11514, K. Chen, S. Seshadri, et L.-J. Zhang, Éd., in Lecture Notes in Computer Science, vol. 11514. , Cham: Springer International Publishing, 2019, p. 1-17. doi: 10.1007/978-3-030-23551-2_1.

[17] « The Next-Generation EDW Is The Big Data Warehouse », Forrester. Consulté le: 21 août 2023. [En ligne]. Disponible sur: https://www.forrester.com/report/The-NextGeneration-EDW-Is-The-Big-Data-Warehouse/RES128005

[18] L. Sautot, S. Bimonte, et L. Journaux, « A Semi-Automatic Design Methodology for (Big) Data Warehouse Transforming Facts into Dimensions », *IEEE Trans. Knowl. Data Eng.*, vol. 33, nº 1, p. 28-42, janv. 2021, doi: 10.1109/TKDE.2019.2925621.

[19] Y. Ramdane, O. Boussaid, D. Boukraà, N. Kabachi, et F. Bentayeb, « Building a novel physical design of a distributed big data warehouse over a Hadoop cluster to enhance OLAP cube query performance », *Parallel Comput.*, vol. 111, p. 102918, juill. 2022, doi: 10.1016/j.parco.2022.102918.

[20] S. Benkrid, L. Bellatreche, Y. Mestoui, et C. Ordonez, « PROADAPT: Proactive framework for adaptive partitioning for big data warehouses », *Data Knowl. Eng.*, vol. 142, p. 102102, nov. 2022, doi: 10.1016/j.datak.2022.102102.

[21] C.-H. Chang, F.-C. Jiang, C.-T. Yang, et S.-C. Chou, « On construction of a big data warehouse accessing platform for campus power usages », *J. Parallel Distrib. Comput.*, vol. 133, p. 40-50, nov. 2019, doi: 10.1016/j.jpdc.2019.05.011.

[22] S. Benkrid, Y. Mestoui, L. Bellatreche, et C. Ordonez, « A Genetic Optimization Physical Planner for Big Data Warehouses », in *2020 IEEE International Conference on Big Data (Big Data)*, déc. 2020, p. 406-412. doi: 10.1109/BigData50022.2020.9378196.

[23] E. Costa, C. Costa, et M. Y. Santos, « Evaluating partitioning and bucketing strategies for Hive-based Big Data Warehousing systems », *J. Big Data*, vol. 6, nº 1, p. 34, mai 2019, doi: 10.1186/s40537-019-0196-1.

[24] K. Smelyakov, A. Chupryna, D. Sandrkin, et M. Kolisnyk, « Search by Image Engine for Big Data Warehouse », in *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, Vilnius, Lithuania: IEEE, avr. 2020, p. 1-4. doi: 10.1109/eStream50540.2020.9108782.

[25] I. Hilali, N. Arfaoui, et R. Ejbali, « A new approach for integrating data into big data warehouse », in *Fourteenth International Conference on Machine Vision (ICMV 2021)*, SPIE, mars 2022, p. 475-480. doi: 10.1117/12.2623069.

[26] H. Wang *et al.*, « Efficient query processing framework for big data warehouse: an almost join-free approach », *Front. Comput. Sci.*, vol. 9, nº 2, p. 224-236, avr. 2015, doi: 10.1007/s11704-014-4025-6.

[27] B. Malysiak-Mrozek, J. Wieszok, W. Pedrycz, W. Ding, et D. Mrozek, « High-Efficient Fuzzy Querying With HiveQL for Big Data Warehousing », *IEEE Trans. Fuzzy Syst.*, vol. 30, nº 6, p. 1823-1837, juin 2022, doi: 10.1109/TFUZZ.2021.3069332.

[28] M. Kechar, L. Bellatreche, et S. Nait-Bahloul, « ZigZag+: A global optimization algorithm to solve the view selection problem for large-scale workload optimization », *Eng. Appl. Artif. Intell.*, vol. 115, p. 105251, oct. 2022, doi: 10.1016/j.engappai.2022.105251.

[29] A. Shahid, T.-A. N. Nguyen, et M.-T. Kechadi, « Big Data Warehouse for Healthcare-Sensitive Data Applications », *Sensors*, vol. 21, nº 7, p. 2353, mars 2021, doi: 10.3390/s21072353.

[30] Y. Ramdane, N. Kabachi, O. Boussaid, et F. Bentayeb, « SDWP: A New Data Placement Strategy for Distributed Big Data Warehouses in Hadoop », in *Big Data Analytics and Knowledge Discovery*, vol. 11708, C. Ordonez, I.-Y. Song, G. Anderst-Kotsis, A. M. Tjoa, et I. Khalil, Éd., in Lecture Notes in Computer Science, vol. 11708. , Cham: Springer International Publishing, 2019, p. 189-205. doi: 10.1007/978-3-030-27520-4_14.

[31] A. A. Alekseev, V. V. Osipova, M. A. Ivanov, A. Klimentov, N. V. Grigorieva, et H. S. Nalamwar, « Efficient data management tools for the heterogeneous big data warehouse », *Phys. Part. Nucl. Lett.*, vol. 13, nº 5, p.

689-692, sept. 2016, doi: 10.1134/S1547477116050022.

[32] C. Costa et M. Y. Santos, « The SusCity Big Data Warehousing Approach for Smart Cities », in *Proceedings of the 21st International Database Engineering & Applications Symposium on - IDEAS 2017*, Bristol, United Kingdom: ACM Press, 2017, p. 264-273. doi: 10.1145/3105831.3105841.

[33] P. Russom, « Evolving data warehouse architectures in the age of big data », *Data Wareh. Inst.*, 2014.

[34] M. Y. Santos *et al.*, « Evaluating SQL-on-Hadoop for Big Data Warehousing on Not-So-Good Hardware », in *Proceedings of the 21st International Database Engineering & Applications Symposium*, in IDEAS '17. New York, NY, USA: Association for Computing Machinery, juill. 2017, p. 242-252. doi: 10.1145/3105831.3105842.

[35] C. Costa et M. Y. Santos, « Evaluating Several Design Patterns and Trends in Big Data Warehousing Systems », in *Advanced Information Systems Engineering*, J. Krogstie et H. A. Reijers, Éd., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, p. 459-473. doi: 10.1007/978-3-319-91563-0_28.

[36] P. Russom, « Data warehouse modernization in the age of big data analytics », *Data Wareh. Inst.*, 2016.

[37] R. K. Bathla et S. G., « Research Analysis of Big Data and Cloud Computing with Emerging Impact of Testing », *Int. J. Eng. Technol.*, vol. 7, p. 239-243, août 2018.

[38] A. A. Harby et F. Zulkernine, « From Data Warehouse to Lakehouse: A Comparative Review », in *2022 IEEE International Conference on Big Data (Big Data)*, déc. 2022, p. 389-395. doi: 10.1109/BigData55660.2022.10020719.

[39] N. Miloslavskaya et A. Tolstoy, « Application of Big Data, Fast Data, and Data Lake Concepts to Information Security Issues », in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, août 2016, p. 148-153. doi: 10.1109/W-FiCloud.2016.41.

[40] P. Lo Giudice, L. Musarella, G. Sofo, et D. Ursino, « An approach to extracting complex knowledge patterns among concepts belonging to structured, semi-structured and unstructured sources in a data lake », *Inf. Sci.*, vol. 478, p. 606-626, avr. 2019, doi: 10.1016/j.ins.2018.11.052.

[41] P. P. Khine et Z. S. Wang, « Data lake: a new ideology in big data era », *ITM Web Conf.*, vol. 17, p. 03025, 2018, doi: 10.1051/itmconf/20181703025.

[42] « Data Lake Governance Best Practices - DZone », dzone.com. Consulté le: 23 août 2023. [En ligne]. Disponible sur: https://dzone.com/articles/data-lake-governance-best-practices

[43] S.-C. Chou, C.-T. Yang, F.-C. Jiang, et C.-H. Chang, « The Implementation of a Data-Accessing Platform Built from Big Data Warehouse of Electric Loads », in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, juill. 2018, p. 87-92. doi: 10.1109/COMPSAC.2018.10208.

[44] O. Azeroual, J. Schöpfel, D. Ivanovic, et A. Nikiforova, « Combining Data Lake and Data Wrangling for Ensuring Data Quality in CRIS », *Procedia Comput. Sci.*, vol. 211, p. 3-16, janv. 2022, doi: 10.1016/j.procs.2022.10.171.

[45] J. Kutay, « Data Warehouse vs. Data Lake vs. Data Lakehouse: An Overview of Three Cloud Data Storage Patterns », Striim. Consulté le: 23 août 2023. [En ligne]. Disponible sur: https://www.striim.com/blog/data-warehouse-vs-data-lake-vs-data-lakehouse-an-overview/

[46] Z. Chen, H. Shao, Y. Li, H. Lu, et J. Jin, « Policy-based access control system for delta lake », in *2022 Tenth International Conference on Advanced Cloud and Big Data (CBD)*, IEEE, 2022, p. 60-65.

[47] M. Armbrust *et al.*, « Delta lake: high-performance ACID table storage over cloud object stores », *Proc. VLDB Endow.*, vol. 13, nº 12, p. 3411-3424, août 2020, doi: 10.14778/3415478.3415560.

[48] « Apache Iceberg ». Consulté le: 3 septembre 2023. [En ligne]. Disponible sur: https://iceberg.apache.org/

[49] V. Belov et E. Nikulchev, « Analysis of Big Data Storage Tools for Data Lakes based on Apache Hadoop Platform », *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 12, nº 8, Art. nº 8, 31 2021, doi: 10.14569/IJACSA.2021.0120864.

[50] « Hello from Apache Hudi | Apache Hudi ». Consulté le: 3 septembre 2023. [En ligne]. Disponible sur: https://hudi.apache.org/

[51] P. Jain, P. Kraft, C. Power, T. Das, I. Stoica, et M. Zaharia, « Analyzing and Comparing Lakehouse Storage Systems », CIDR, 2023.
[52] L. Gagliardelli *et al.*, « A big data platform exploiting auditable tokenization to promote good practices inside local energy communities », *Future Gener. Comput. Syst.*, vol. 141, p. 595-610, avr. 2023, doi: 10.1016/j.future.2022.12.007.
[53] S. Ait Errami, H. Hajji, K. Ait El Kadi, et H. Badir, « Spatial big data architecture: From Data Warehouses and Data Lakes to the LakeHouse », *J. Parallel Distrib. Comput.*, vol. 176, p. 70-79, juin 2023, doi: 10.1016/j.jpdc.2023.02.007.
[54] DataBeans, « Delta vs Iceberg vs hudi : Reassessing Performance », Medium. Consulté le: 4 septembre 2023. [En ligne]. Disponible sur: https://databeans-blogs.medium.com/delta-vs-iceberg-vs-hudi-reassessing-performance-cb8157005eb0
[55] F. Hellman, « Study and Comparsion of Data Lakehouse Systems », 2023.
[56] « Top 5 Reasons for Choosing S3 over HDFS », Databricks. Consulté le: 24 septembre 2023. [En ligne]. Disponible sur: https://www.databricks.com/blog/2017/05/31/top-5-reasons-for-choosing-s3-over-hdfs.html
[57] M. Armbrust, A. Ghodsi, R. Xin, et M. Zaharia, « Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics », in *Proceedings of CIDR*, 2021.
[58] « Data Lakehouse Architecture and AI Company », Databricks. Consulté le: 20 septembre 2023. [En ligne]. Disponible sur: https://www.databricks.com/
[59] A. Eldawy et M. F. Mokbel, « The era of big spatial data », in *2015 31st IEEE International Conference on Data Engineering Workshops*, avr. 2015, p. 42-49. doi: 10.1109/ICDEW.2015.7129542.
[60] A. Behm *et al.*, « Photon: A Fast Query Engine for Lakehouse Systems », in *Proceedings of the 2022 International Conference on Management of Data*, Philadelphia PA USA: ACM, juin 2022, p. 2326-2339. doi: 10.1145/3514221.3526054.
[61] A. Eldawy, L. Alarabi, et M. F. Mokbel, « Spatial partitioning techniques in SpatialHadoop », *Proc. VLDB Endow.*, vol. 8, nº 12, p. 1602-1605, 2015.

Table 1: Comparison between DW, Big DW, Data lake and Data LakeHouse

Feature	Data Lake	Data Warehouse	Big Data Warehouse	Data LakeHouse
Data Storage	Raw data in original form	Structured data	- Structured data - Unstructured data	- Structured data - Unstructured data
Schema	On-read	On-write	On-read/On-write	On-read/On-write
Data Integration Tools	EL	ETL	EL/ETL	EL/ETL
Data Processing	-Batch data processing -Real-time data processing	Batch data processing	-Batch data processing -Real-time data processing	-Batch data processing -Real-time data processing -Data management and governance
Data Integration	Data silos	Eliminates data silos	-Eliminating data silos -Ensuring data consistency	-Eliminates data silos -Data consistency and accuracy
Data Management	Less control	Highly control	Highly control	Highly control
Scalability	Highly scalable	Limited	Extremely scalable	Extremely scalable
Cost	Less	Expensive	More expensive	More expensive
Queries	Ad hoc	Predefined	- Ad hoc - predefined	- Ad hoc - predefined
Use Cases	Storage of a large amounts of raw data for later use	-Structured data storage -Data reporting and analysis	-Structured and unstructured data storage -Processing and analyzing	-Structured and unstructured data storage -Processing and analyzing -Focus on data quality and consistency.