ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 13 (2023) No. 2, pp. 17-30
DOI:10.33168/JSMS.2023.0202

A Scalable Smart Farming Big Data Platform for Real-Time and Batch Processing Based on Lambda Architecture

Mohamed El Mehdi El Aissi, Sarah Benjelloun, Younes Lakhrissi, Safae El Haj Ben Ali

[1] Sidi Mohamed Ben Abdellah University, Faculty of Science and Technology,
SIGER Laboratory Fes, Morocco
mohamedelmehdi.elaissi@usmba.ac.ma (Corresponding author)

Abstract. In recent years, the rise of big data technologies, the Internet of Things (IoT), and cloud computing has led to significant advances in data-driven strategies. However, despite the emergence of the concept of smart farming, the vast amounts of data generated by IoT devices remain largely underutilized due to a lack of effective data management systems. To fully exploit this data and take agriculture to the next level, it is crucial to implement a dedicated big data architecture that can capture and analyze data in real-time. In this paper, we propose a new architecture for managing big data in smart farming that is based on the data lake concept and the lambda architecture. Our proposed architecture aims to address the challenges of collecting, processing, and analyzing smart farming data in both batch and real-time modes. By combining smart farming and big data, our approach offers the possibility of real-time farm monitoring for agri-ecosystems stakeholders, which can help maximize productivity and quality while minimizing effort. Overall, our proposed architecture represents a significant step forward for smart farming and has the potential to revolutionize the agricultural industry. With the ability to capture and analyze data in real-time, our approach will enable farmers to make more informed decisions and optimize their operations for greater efficiency and profitability.

Keywords: Batch Processing, Big Data Platform, Data-driven Strategy, Data Lake, Lambda Architecture, Real-Time Processing, Smart Farming.

1. Introduction

According to the United Nations reports, the global population is growing by 81 million annually. Additionally, it is predicted that by 2030, there will be 8.5 billion people on the planet, and by 2050, there will be 9.7 billion [1]. This means the food consumption rate is growing rapidly since the agricultural field already produces approximately 17% more than it used to produce just three decades ago. However, about 821 million people worldwide face the risk of food security [2]. Additionally, to meet the food demands of the massively increasing world population, the food and agriculture organizations (FAO) affirmed that agriculture production should be increased by 70% by 2050 [1]. On the other hand, it is about feeding people and providing them with highly nutritious food without harming the environment.

Along with this perspective, the agricultural field has multiple challenges in providing farmers with advanced systems, allowing analysis and prediction operations to meet the increasing population demands with height efficiency and quality, and protecting the environment and sustainability.

Data-driven strategy is the approach of making strategic decisions based on analyzing data. It allows companies to organize their data and extract valuable information by concluding it to serve the consumer better. On the other hand, precision farming is observing, measuring, and taking adequate actions for a better farming life cycle. Indeed, the data-driven farming strategy extends the precision farming approach by considering data as a cornerstone and building all the operational processes and decision-making on top of it.

At this level, it is primordial to shed light on the Internet of Things (IoT) technologies for smart farming as it becomes widely adopted by farmers, such as drones and sensors. Moreover, according to Business Insider Intelligence, more than 11 million agricultural sensors will be implanted by 2023 [2]. Hence, those IoT devices generate massive quantities of data every second. Moreover, IBM estimates that the average farm generates half a million data records [1].

Big data analysis has been successful in various industries, including banking, insurance, healthcare, business, and marketing [3]. Big data has been extremely successful in the industries mentioned earlier but has not yet been broadly adopted in the agricultural sector [4]. So, using Big Data-related technologies to address the issues of productivity, environmental impact, food security, and sustainability became inevitable [5]. This may be accomplished by offering a productive data-driven agriculture approach by giving farm stakeholders in-depth knowledge of the entire ecosystem.

The complexity resides in designing a dedicated smart farming big data architecture to allow:

- Handling 5 V's (volume, variety, veracity, velocity, value) generated data
- Processing IoT and sensors data in both real-time and batch mode using highly efficient technologies
- Performing advanced analytics such as predictive or prescriptive analytics
- Exposing the analyzed data in dashboards and reports

Despite the increasing interest in using big data analytics in agriculture, there is still a significant literature gap in existing smart agriculture architectures. One of the primary issues is the lack of integration of data from multiple sources, which limits the development of actionable insights for farmers. While some studies have focused on individual data sources, such as weather or soil moisture sensors, the integration of satellite imagery or social media data is often missing. Additionally, the lack of effective models for data analysis hinders the ability of farmers to make informed decisions about their operations. Therefore, there is a need for research to develop comprehensive smart agriculture architectures based on big data that can integrate multiple data sources and provide actionable insights for farmers.

This paper proposes a dedicated big data architecture for handling data from various sources focusing on the different data processing approaches. The rest of this paper is structured as follows. After the introduction, the second section reviews the existing smart farming systems. The third section describes the three different data processing architectures: Lambda architecture, Kappa architecture,

and Hybrid architecture. The fourth section presents big data processing technologies that can serve in implementing a dedicated data solution for smart farming. In the fifth section, we propose a smart farming big data platform allowing batch and real-time data processing. In the sixth section, we present and discuss the obtained results of the platform performance check. Finally, in the seventh section, we draw a conclusion.

2. Literature Review

In the literature, we find multiple papers related to big data applications in the agricultural field. Namely, we can cite precision farming, yield prediction, risk prediction, sustainable farming, and supply chain management [6-7-8-9]. As well as that, we distinguish between three categories of big data applications in agriculture: (i) advanced sensor technology systems, (ii) prediction and risk management systems, and (iii) agricultural management systems [10]. Since our work aims to provide big data processing architecture for data-driven agriculture strategies, we shed light on the third category in this section.

Lately, studies have been focused on creating dedicated agricultural data platforms to handle generated massive data and provide smart agriculture systems with highly performant decision support tools. However, in most cases, those systems are designed by private companies or public-private partnerships [11]. For instance, the Monsanto company developed an integrated farming platform that collects multiple data and provides valuable information to farmers [12]. The Climate FieldView platform gathers data from heterogeneous sources and transforms them to provide diagnostics to farmers [11]. Even so, from 2018 until now, there have been only four market leaders: DowDuPont, Syngenta-ChemChina, BASF, and Bayer-Monsanto [13].

On the other hand, some public-private partnerships have engendered advanced agriculture systems to enhance the adoption of modern technologies and explore the benefit of the collected data. To illustrate, the Barto platform results from the collaboration of multiple actors from the private and public sectors to build a smart-farming platform. The main goal behind Barto's platform is to digitize generated farm data to speed up on-farm processes and reduce manual tasks. It must be noted that Barto's platform is developed on top of 365FarmNet, which provides farm management software. To clarify, the platform is available as SaaS (Software as a Service) and offers the possibility of collecting and managing generated data by farms. Unfortunately, Barto's platform is not open access, therefore we could not find the technical architecture.

We found much research related to smart farming systems at the academic level. For example, the PLATEM system collects data from various sources and provides real-time decision support for farmers [14]. Another example is the SmartDairyTracer system, which allows easy state monitoring of dairy cattle and feeds in real-time and digitizes production processes. SmartDairyTracer's architecture is designed using the global edge computing architecture, which is split into three main layers: (i) IoT, (ii) Edge, and (iii) Business solution layers [15].

3. Big Data Processing Architectures for Data Driving Smart Farming

The amount of data generated by smart farming is increasing exponentially due to the adoption of IoT and network technologies. Moreover, the data being generated by smart farming systems are categorized as big data since it respects all 5V's that characterize big data, which are listed below:

- Volume: The size of data produced is referred to as volume. This aspect of big data is most closely associated with its sheer size[16]. In some cases, the amount of data generated can be staggering, with large organizations amassing Terabytes or even Petabytes of information stored on various devices and servers. The greater the volume of data, the more opportunities there are for obtaining valuable insights and uncovering meaningful patterns.

- Velocity: The rate at which data is produced and how swiftly it travels is called velocity. This element is crucial for organizations as it determines data availability for prompt, informed decision-making.
- Variety: Dealing with diverse types of data gathered from multiple sources is a part of big data's variety feature [16]. Data is typically divided into three categories: structured, semi-structured, and unstructured. Data is structured in a set format, length, and size. Semi-structured data complies with a specific data format to some extent. Conversely, unstructured data is disorganized and does not adhere to conventional data formats. For instance, relational database data or CSV files are examples of structured data; JSON, XML, or other markup languages are examples of semi-structured data; and photos, videos, and social media data are examples of unstructured data.
- Value: The value of data is determined by its potential to impact an organization positively. Simply gathering large volumes of data is not enough; merely storing and aggregating data does not lead to added value. What is essential is the meaningful insights that can be gleaned from the data [17]. By utilizing advanced data analytics, organizations can extract valuable insights from the collected data, which can then be used to inform decision-making. To determine if the investment of time and effort into big data is worth it, a cost-benefit analysis can be performed to assess if big data analytics will bring any value to the organization.
- Veracity: or validity, measures the quality and trustworthiness of collected data. The data must be credible for the insights derived from this data to be reliable enough to inform decision-making. Dirty data, also known as rogue data, is false, erroneous, or inconsistent information. Cleaning dirty data helps improve its veracity and reduces the risk of making misinformed decisions. Duplicate, out-of-date, non-compliant, incomplete, and erroneous data are typical types of dirty data. Organizations can perform data health assessments with data providers to avoid dirty data, blend data sources (first-party, third-party, and intent data), regularly cleanse data, fill gaps, and engage in ongoing data management.

This huge data should be managed and transformed to extract valuable information. For this purpose, generated data undergo a pipeline of extraction, loading, and transformation, also known as the ELT process, to integrate into Data Lake. To be clear, a data lake is a centralized storage system for managing large amounts of data of many forms, including structured, semi-structured, and unstructured data [18]. Since it does not require a preexisting schema or model to respect the acquired data, the data lake gives great flexibility in managing and dealing with data [19]. In addition, data should be processed to provide in-depth insights to support data-driven smart farming strategies. To do so, data should be processed in batch and real-time modes. From this perspective, three state-of-the-art architectures are often used to process big data: Lambda, Kappa, and Hybrid architecture.

3.1. Lambda Architecture

Lambda architecture is defined as a scalable, fault-tolerant, batch-processing, and stream-processing architecture [20]. As illustrated in Figure 1, it has two branches, one for the batch-processing layer where data is appended to the master dataset area, then using scheduled jobs with defined frequency, data is aggregated, and results are provided to the batch views. The second layer is the stream-processing layer, where data is processed in real-time, and the output is stored in incremental views. Due to the advanced aggregation of the massive data generated, the batch-processing layer is time-consuming. As a result, the output is usually not up to date.

For this reason, the stream-processing layer minimizes processing time by performing aggregations and transformations on small data. As a result, the queries are performed on recent data. Finally, the serving layer merges the output views of batch-processing and stream-processing layers in a single view the end-user uses to query data.

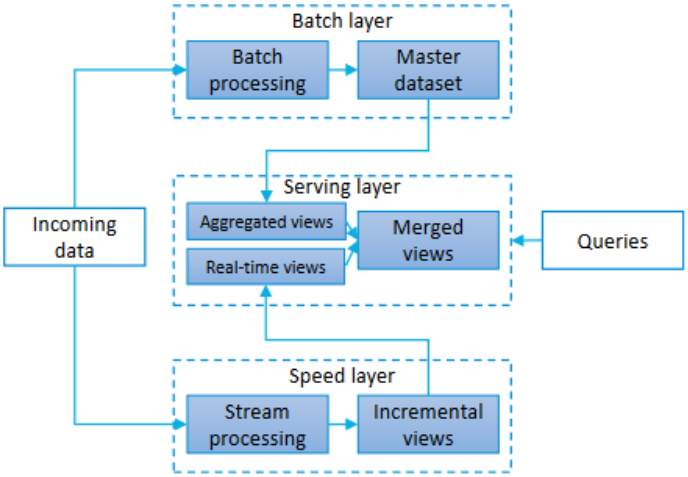

Fig. 1: Lambda architecture [12]

3.2. Kappa Architecture

The kappa architecture is designed to highlight the advantage of using a stream-processing approach as it relies only on the stream-processing layer, as illustrated in figure 2. The collected data is processed in real-time. The output results are appended in incremental views; then, the serving layer is the entry point for the end user to perform analysis on data. In addition, the Kappa architecture also offers the possibility of processing the entire data saved at the master dataset level if needed. Despite the use of one processing layer, which simplifies the systems, performing aggregations on massive data sets and real-time data in parallel may result in storage and performance issues.

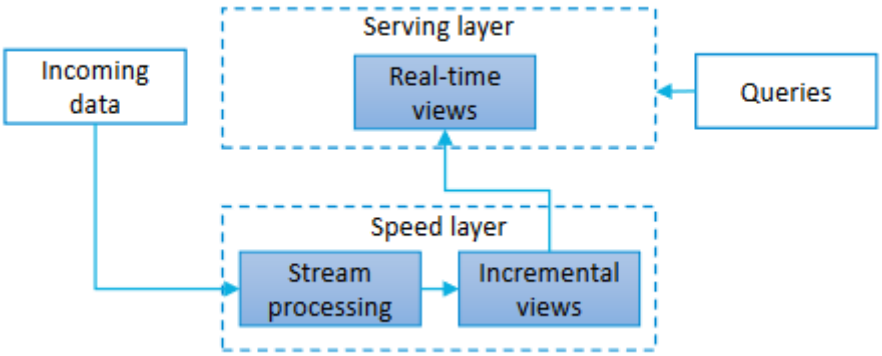

Fig.2: Kappa architecture [12]

3.3. Hybrid Architecture

Hybrid architecture is designed to use Lambda and kappa architecture advantages. The reason behind this is to hide the complexity of merging both batch-processing and stream-processing layer results manually. Since the computations are done separately on each layer, adopting a hybrid architecture data analysis is more efficient and accessible [21]. A good example of hybrid architecture is the BRAID architecture, which combines batch-processing and stream-processing layers in a shared result layer [22], as presented in figure 3. Thus, querying data is more flexible and accessible since the users do not have to care about which processing layer created the particular data. However, data may be tagged to allow metadata management [23].

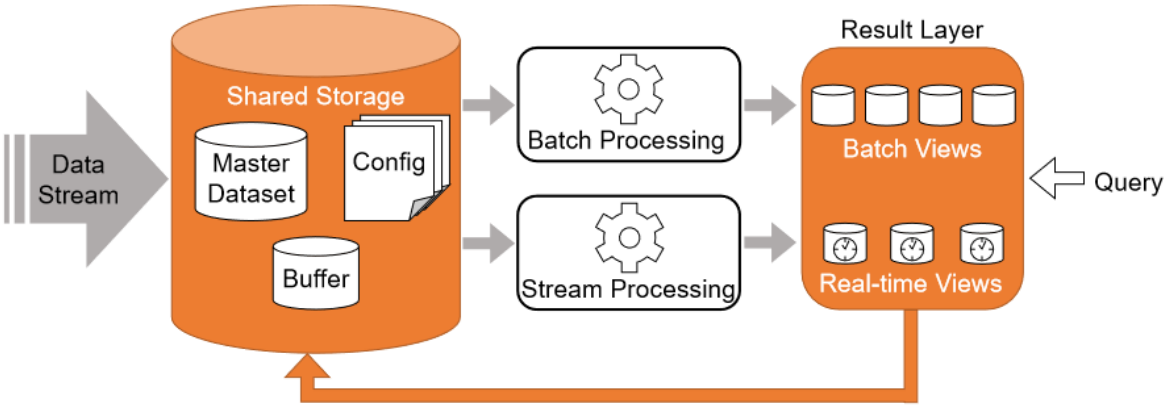

Fig. 3: Hybrid Architecture: BRAID [22]

4. Big Data Processing Technologies

This section presents the different big data technologies that could be adopted to implement a dedicated big data processing architecture for data-driven smart farming. Moreover, the following big data technologies are chosen based on multiple metrics: open source, scalability, community, and distributed.

4.1. Adopted Big Data Technologies

4.1.1 Hadoop Ecosystem

A significant user community supports Apache Hadoop, a big data platform that has been around for a while. Its main goal is to address the complexity and performance problems that come with utilizing conventional systems to analyze enormous amounts of data. Through its distributed file system, Hadoop has been developed to perform advanced data processing on large datasets efficiently. Hadoop enables speedy querying of terabytes of data with excellent failure tolerance because data is duplicated among servers to prevent loss. This is made possible by conducting data processing activities directly on the cluster where the data is stored instead of copying it into memory [24].

The Hadoop Distributed File System (HDFS) and the MapReduce (MapR) paradigm are the two main parts of the Hadoop platform. On top of Apache Hadoop, additional modules can be added to fulfill the needs of particular applications.

4.1.2 Hadoop Distributed File System (HDFS)

Apache Hadoop Distributed File System utilizes a master-slave architecture to store large data files with high scalability economically. It can accommodate structured, semi-structured, and unstructured data types. One of the key benefits of HDFS is that it minimizes network traffic in the cluster by performing computation tasks close to where the data is stored. The cluster consists of two server types: the NameNode, which oversees file system operations and acts as the master, and several Data Nodes, which are responsible for both data storage and computation tasks and serve as the slaves [25].

4.1.3 MapReduce Model (MapR)

Built on top of HDFS, the MapReduce programming model is crucial to large data management. The use of parallel processing and its two primary functions, Map and Reduce, enables the effective processing of massive data sets [26].

The dataset is divided into key-value pairs using the Map function. These pairs are then processed in parallel by the mapper throughout the cluster to produce intermediate key-value pairs. Before moving on to the Reduce phase, these intermediate pairings are sorted and organized. The intermediate key-value pairs are processed in the Reduce phase by averaging the values for each key, which summarizes the entire dataset. The finished product is kept in HDFS.

4.1.4 Data Exposition: Hive, Hbase, Elasticsearch

Built on top of HDFS, Apache Hive is a distributed and fault-tolerant data warehouse system. Its goal is to use structured tables to provide centralized storage and effective analysis of massive datasets. Each table in Hive corresponds to an HDFS directory that is further subdivided into buckets and partitions. The ability to query data using HiveQL, a language similar to SQL, is another benefit of Hive. A HiveQL query is converted into MapReduce jobs when it is executed, and these tasks have a direct impact on the HDFS data that is being processed in parallel [27].

Built on top of HDFS, Apache HBase is a distributed, column-oriented, non-relational database with a key/value model. It was developed to manage a high rate of table updates and store non-relational data [28].

Elasticsearch is a document-oriented, NoSQL database that stores information in an unstructured manner and does not support SQL queries. Since it is based on indices, searching requires indexing the entire object graph. Kibana is a proprietary data visualization tool for Elasticsearch, while Elasticsearch is a distributed search and analytics engine.

The Kibana Query Language, which supports free text or field-based filtering, allows users to examine data stored in Elasticsearch using Kibana interactively. The Elastic Stack can be used to store and analyze logs, metrics, and security event data in addition to being useful for functional data [29].

4.1.5 Data Ingestion: Apache Sqoop, Apache Flume

An open-source command-line program called Apache Sqoop is made to quickly move large amounts of data between relational database management systems like MySQL, Oracle, and Hadoop. It is an ETL application (extract, transform, load) [30].

Large amounts of unstructured data can be gathered, combined, and transferred into Hadoop using the open-source, dependable, and flexible Apache Flume technology (HDFS or HBase). Flume, a Java-based application, has a flexible design based on streaming data flows and a built-in query processing engine to modify incoming data before delivering it to the sink, its eventual destination [30].

4.1.6 Data Processing: Apache Spark

Unlike the MapReduce framework, Apache Spark is an open-source distributed data processing engine that uses in-memory caching for improved performance. Through development APIs in Scala, Python, or Java, Spark makes it possible to conduct intricate calculations on enormous datasets. The Resilient Distributed Dataset (RDD) idea, a collection of data built from a source system or another RDD kept in memory, serves as the foundation for Spark. The robustness of RDDs is increased by Spark's ability to recover an RDD in the event of failure using a Direct Acyclic Graph that depicts the order of operations. It is vital to remember that Spark automatically converts Data Frames and Datasets into RDDs before processing them when interacting with them. The Spark framework is made up of a number of the following components [31]:

- Spark Core: The Spark Core is the platform's main component and is in charge of managing memory, allocating tasks, interfacing with HDFS, and fault recovery.
- Spark SQL: Unlike the MapR approach, Spark SQL is a distributed engine that enables quick interactive querying. Additionally, it leverages HiveQL and offers the option of writing the output RDD to Hive tables.
- Spark Streaming: Spark Streaming is a real-time engine that enables streaming data analysis. It primarily relies on the idea of micro-batch data input and makes it possible to process data using logic that is quite similar to that of batch processing.

4.1.7 Stream Processing: Apache Kafka

Apache Kafka is an open-source technology with low latency and high throughput for managing real-time data flows. It offers a very effective and trustworthy method for handling data streams because it was created in Java and Scala. Kafka is excellent for creating real-time data pipelines because of its

capacity to process over 2 million writes per second and promise no data loss. Data is published to a topic on a broker in this publish-subscribe messaging system, where it is subsequently made available for use by other applications [32].

4.1.8 Data Scheduling: Apache Oozie, Apache Airflow

Oozie is a server-based workflow scheduling tool used to control Hadoop tasks. The workflows in Oozie are arranged as control flow and action nodes in a Directed Acyclic Graph (DAG) architecture and authored in XML.

On the other hand, Airflow is an open-source platform designed to manage complex data engineering pipelines. It was developed by Airbnb and is based on Python. Workflows in Airflow are authored as a series of tasks organized as Directed Acyclic Graphs (DAGs).

5. Big Data Processing Architectures for Data Driving Smart Farming

The following section proposes a dedicated big data architecture for processing generated smart farming data. The designed architecture is based on Lambda architecture. Indeed, this choice comes from the fact that batch and stream data processing are separated and cannot be merged in one job. In addition, the Lambda architecture allows linear scalability and high fault tolerance.

One of the main characteristics of a Data Lake architecture is to ensure high flexibility and easy data access. In the same perspective, we propose a multi-zone data lake architecture for handling smart farming data based on three zones, namely, (i) Raw Zone, (ii) Trusted Zone, and (iii) Access Zone.

The information is gathered from a variety of sources, including Relational Data Base Management Systems (RDBMS), CRM/ERP, flat files created by humans, Application Programming Interfaces (APIs) for weather, and Internet of Things (IoT)/Sensors for data like pH, temperature, and pressure.

5.1 Raw Zone

The entry point of the data lake is the gateway machine, which is responsible for gathering data from various sources before it is stored in the Hadoop Distributed File System. The gateway server is separate from HDFS and ensures that data files are tracked by assigning them a naming pattern based on their source and date of arrival. The data loader, a specially designed job, retrieves the files from the gateway and stores them in the data lake's file system. It utilizes the naming pattern to determine the source, domain, and timing of each batch of data and loads it into the relevant HDFS repository.

Depending on the file extension, the data loader can handle various data types, including structured, semi-structured, and unstructured. Three repositories structured, semi-structured, and unstructured are used in HDFS to store data. Data is organized into groups by source and domain inside each repository. The Raw Zone, which consists of these three HDFS repositories, stores the data in its unaltered original form.

5.2 Trusted Zone

Before entering the trusted zone, the data lake's data must travel via a gateway. Data is extracted from many sources and stored in the Hadoop Distributed File System on this gateway system. It employs a naming scheme to keep track of the data and its source. In order to identify the source, domain, and time of the data batches, the data loader uses the naming pattern to load the data into HDFS. Based on the file extension, the loader distinguishes between structured, semi-structured, and unstructured data and saves each kind in a different HDFS repository in the Raw Zone.

Hive's structured tables serve as the direct storage for structured data. While managed tables are handled by Hive, including storage and metadata, external tables in Hive are linked to a remote data directory with a specific structure. Although establishing the table is a one-time job, Apache Oozie is used to automate data feeding into the conforming zone.

Apache Spark tasks change semi-structured data, like JSON or XML files, to fit a specified structure in the conformed zone. A specific Spark job must be created for each semi-structured data source to

transform and store the data in a structured Hive table. The orchestrator is used to carry out the Spark jobs methodically.

The smart farming data lake typically uses unstructured data for data science objectives, such as image and geographic data. To handle this data, transition scripts are built to construct specific data science repositories for each use case. Using Apache Kafka and Spark, streaming data from IoT devices and sensors are added to the trustworthy zone.

5.3 Access Zone

The access zone aims to make all data stored in the data lake easily accessible for reporting, dashboard development, and data analysis tasks. For each requirement, a separate layer is made with controlled data access in order to do this. These layers may consist of Hive tables or specific directories with unstructured data. All external tools and teams can connect to the data lake and read the designated data because Apache Ranger regulations regulate data access.

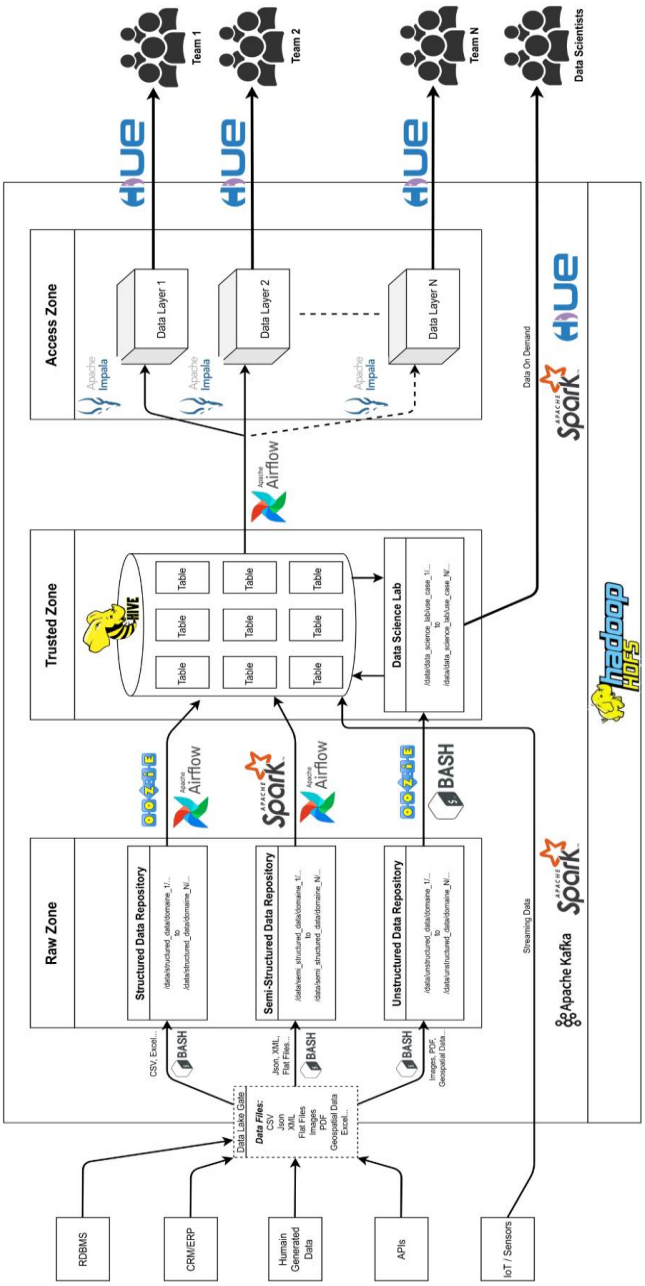

Fig. 4: Data Lake Architecture for Processing Smart Farming Big Data

6. Results and Discussion

Data-driven strategies are more than business intelligence (BI) in terms of impact on organizations. A data-driven strategy affects all the processes directly or indirectly linked to a company's operations. Within the same perspective, adopting a data-driven strategy will enhance the smart farming domain by making smart and data-based decisions. However, to have an efficient data-driven strategy for smart farming, it is imperative to have a robust data management platform capable of handling all data generated by IoTs and other sources.

The choice of the data lake solution comes from the fact that traditional data management systems, such as data warehouses, are not designed to handle big data. Even so, those systems are not offering the possibility of performing advanced analytics on data. The proposed data lake architecture aims to offer a data platform for advanced analytics and predictions. Moreover, multiple architectures, like the lambda and kappa architectures, can be deployed within a data lake architecture depending on the use case. Also, the choice of a three-zone data lake architecture is mainly to allow high flexibility and scalability.

The proposed architecture for smart farming is a dedicated lambda architecture. It contains two data processing layers: the batch-processing layer and the stream-processing layer. The optics behind this approach is to offer a separated processing level for two data flow types, batch data flow, and stream data flow, separated for more efficiency and performance. Figure 5. represents the previous lambda architecture and highlights its two layers.

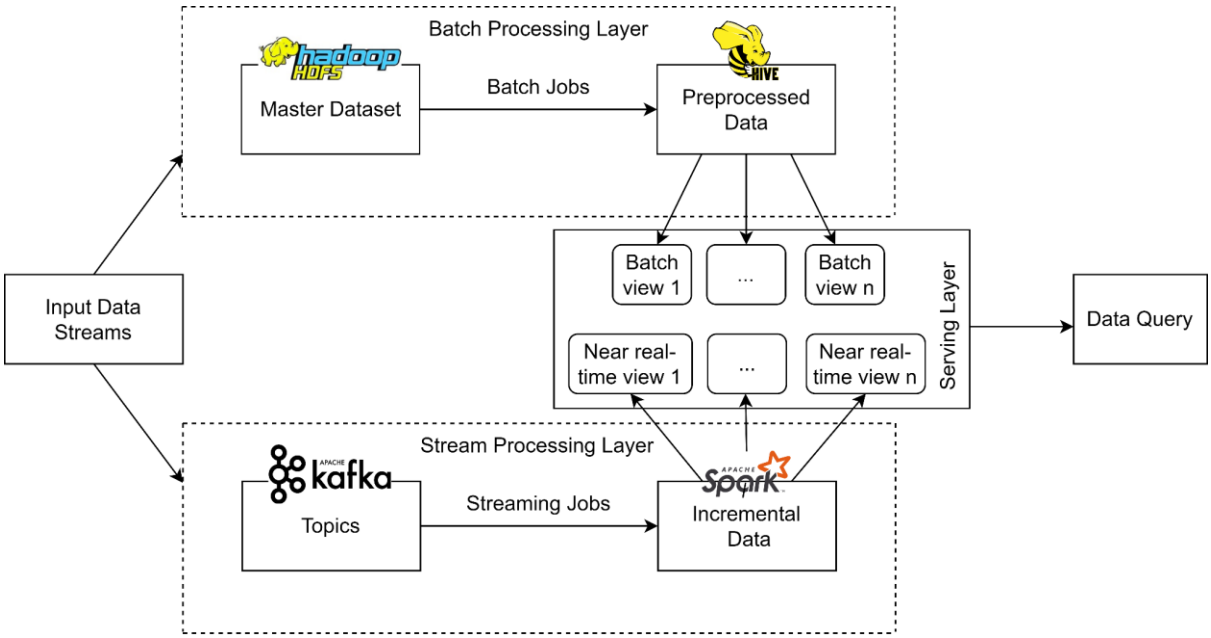

Fig.5: Lambda architecture implementation

Moreover, in this perspective, we implemented a proof of concept following the lambda architecture in figure 5. to evaluate its performance and effectiveness. We performed executions on both the stream-processing layer and batch-processing layer. For the stream-processing layer, data streams are generated by IoT devices and sent directly to Kafka topics. Then, using Spark structured streaming jobs, calculations are performed to provide near real-time KPIs. Finally, the results are resent to Kafka and stored in Hive. In parallel, at the level of the batch-processing layer, the same input data is stored at the level of HDFS to constitute the master dataset, and transformations are executed using MapR jobs running periodically.

To demonstrate, we simulate data generated by IoT devices and send it to Kafka topics. It has to be noted that the IoT data used for this proof of concept represent smart fish farming data. Collected data represents temperature, Ph, and dissolved oxygen from sensors implanted in fish farming tanks. The goal is to calculate the Water Quality Index (WQI) using these three parameters. The stream-processing layer provides near real-time WQI based on data streams. The streaming jobs consume this data from the topics and transform it via spark-structured streaming with a trigger time of 30 seconds. Then, pre-processed data is sent to Hive tables. At the same time, we capture the processing time of each streaming transformation. Furthermore, in the batch-processing layer, a job consumes all data available at the master dataset from HDFS in order to transform and store it. This job runs every six hours for seven days and, each time consumes all data available.

Figure 6 shows the processing time of the streaming job. Through these experiments, we can see that the processing time of the streaming job varies between 2.52 seconds and 5.2 seconds. Also, the mean execution time is 4.6 seconds in the speed layer. Moreover, we can notice that the processing time is relatively short and does not vary a lot. This is because we receive the same amount of data each time. Note that the reported time includes reading from the source, transforming then storing results in the storage area.

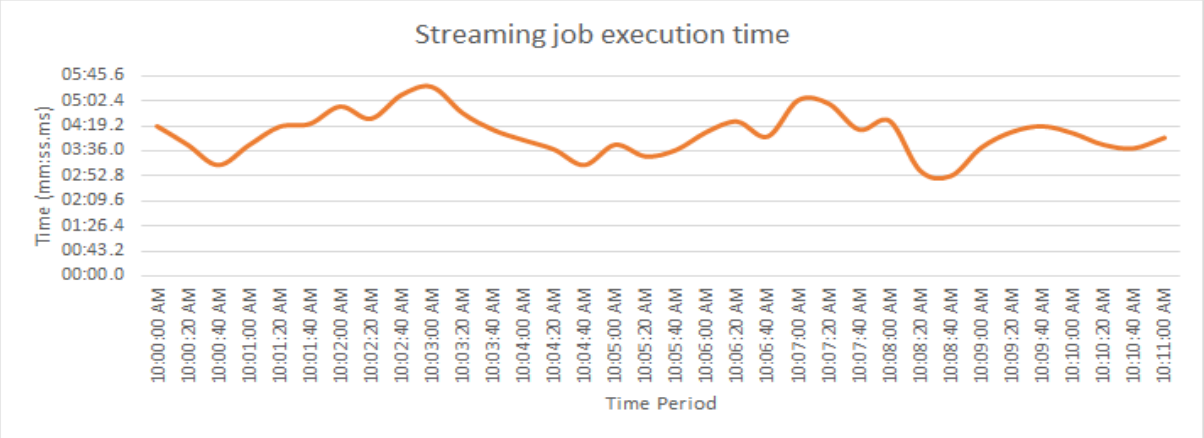

Fig. 6: Execution time for streaming job

In the same way, figure 7 shows the batch-processing job's processing time of the same consumed data. The job reads data from fish farms, conducts some transformations and saves the results in the database. This set of experiments shows that the processing time in the batch layer is considerable and increasing. Plus, the processing time varies greatly (minimum of 2:32 minutes and maximum of 9:36 minutes).

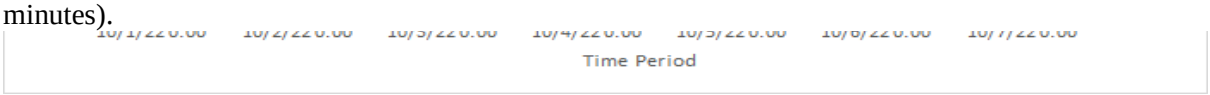

Fig.7: Execution time for batch job

The practical implications of a data-driven strategy for organizations are significant, as it affects all processes directly or indirectly linked to a company's operations. In the smart farming domain, a robust data management platform capable of handling large volumes of data generated by IoTs and other sources is imperative. Data lake solutions offer a better choice for advanced analytics and predictions, and the proposed lambda architecture with two data processing layers offers high flexibility and scalability for efficient and performant processing of data. The proof-of-concept implementation using the lambda architecture demonstrated the efficiency of processing data streams in near-real-time and the importance of architecture design for effective data processing. Theoretical implications of data-driven strategies include the ability to generate insights from large volumes of data, enabling better decision-making and innovation, while also enhancing the understanding of complex relationships between variables.

7. Conclusion

The population's rapid growth leads to thinking of ways to handle the increasing food demand. In fact, agriculture not only helps secure food around the world but also provides job opportunities. Moreover, the challenges of agricultural production are only increasing with climate change and the need for sustainability. Mortality, diseases, and soil and water quality are concerns that need to be addressed and can be helped with precision agriculture and advanced technologies. The agriculture domain deserves to be invested in, and many farms already contain IoT technology.

Furthermore, big data has proved its potential in other industries. Indeed, adopting a big data-based strategy helps to overcome productivity challenges and facilitate decision-making. With the help of big data technologies, data analytics are performed, and the results are displayed in ergonomic dashboards and published in reports. Additionally, a data lake with huge quantities of preprocessed data allows for performing artificial intelligence applications like machine learning and deep learning algorithms. These technologies uncover hidden patterns and produce predictions using preventive methodologies.

In agriculture, data is generated in significant volumes and velocity, characterized by variety, value, and veracity. These are the 5 V's of big data and, thus, the domain. Consequently, the generated data can be transformed to extract value for the domain. Adopting the right big data architecture for stream and batch data is crucial in this optic. This context has three main architectures: lambda architecture, kappa architecture, and hybrid architecture. In smart farming, data is produced in streaming and must be processed continuously to have near real-time monitoring. However, batch data is also produced and must be processed to extract value.

On top of that, the lambda architecture provides the most scalable and fault-tolerant platform. With this in mind, we propose a data lake architecture based on the lambda architecture's batch processing layer and stream processing layer. The big data technologies used for these means include Hadoop HDFS and Hadoop MapR, Hive, HBase, Elasticsearch, Sqoop, Spark, Kafka, and airflow Airflow.

The proposed big data architecture dedicated to agriculture is based on a multi-zone data lake. The three zones (raw, trusted, and access zone) ensure high flexibility and easy access to the preprocessed data. In order to test the performance of our architecture, we implemented a proof of concept with the different layers of the lambda architecture. In order to proceed, farming data is sent to the Kafka topic and HDFS simultaneously and processed by the stream-processing layer and the batch-processing layer. The results of these experiments show that the streaming jobs have a small and consistent processing time while the batch jobs process the historical data in a considerable and increasing time.

Declarations

- **Authors' contributions:** Mohamed El Mehdi El Aissi and Sarah Benjelloun worked on the conceptualization, methodology and writing. Younes Lakhrissi and Safae El Haj Ben Ali did the formal analysis and validation.
- **Funding:** Not Applicable.
- **Availability of supporting data:** Not Applicable.
- **Acknowledgments:** Not Applicable.
- **Competing interests:** The authors declare no conflict of interest.

References

Bhat, S. A., & Huang, N. F. (2021). Big data and AI revolution in precision agriculture: Survey and challenges. *IEEE Access*, 9, 110209-110222.

Perter, N., (2020). The internet of things report. Business Insider Intelligence URL: https://store.businessinsider.com/products/the-internet-of-things-report.

Rodríguez-Mazahua, Lisbeth, et al.(2016). A general perspective of Big Data: applications, tools, challenges and trends. *The Journal of Supercomputing*, 72(8), 3073-3113.

Saiz-Rubio, Verónica, and Francisco Rovira-Más.(2020). From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy*, 10(2), 207.

Astill, Jake, et al. (2020). Smart poultry management: Smart sensors, big data, and the internet of things. *Computers and Electronics in Agriculture*, 170,105291.

Kamilaris A., Kartakoullis A., Prenafeta-Boldú F.X. (2017). A review on the practice of big data analysis in agriculture Computers and Electronics. *Agriculture*, 143, 23-37.

Kunal, P., (2019). Big data in the bigger world of agriculture today. *IEEE India Info*.

O'Grady M.J., O'Hare G.M. (2017). Modelling the smart farm Information processing in agriculture, 4, 179-187.

Wolfert S., Ge L., Verdouw C., Bogaardt M.J. (2017). Big data in smart farming-a review Agricultural Systems, 153, 69-80.

Tantalaki N., Souravlas S., Roumeliotis M. (2019). Data-driven decision making in precision agriculture: The rise of big data in agricultural systemsJournal of Agricultural & Food Information, 20, 344-380.

Finger, R., Swinton, S.M., El Benni, N., Walter, A., (2019). Precision farming at the nexus of agricultural production and the environment.

Roukh, A., Fote, F. N., Mahmoudi, S. A., & Mahmoudi, S. (2020). Big data processing architecture for smart farming. *Procedia Computer Science*, 177, 78-85.

Weersink A., Fraser E., Pannell D., Duncan E., Rotz S.(2018). Opportunities and challenges for big data in agricultural and environmental analysis. *Annual Review of Resource Economics*, 10, 19-37.

Cambra Baseca C., Sendra S., Lloret J.,(2019). Tomas J.A smart decision system for digital farming. *Agronomy*, 9, 216.

Sittón-Candanedo I., Alonso R.S., Corchado J.M., Rodríguez-González S., Casado-Vara. (2019). R.A review of edge computing reference architectures and a new global edge proposal Future Generation. *Computer Systems*, 99, 278-294.

Seung-Wan Ju.(2022). A Study on the Influence of Big Data-based Quality on Satisfaction and Repurchase Intent. *Journal of System and Management Sciences*, 12(3), 286-317.

Wysel, Matthew, Derek Baker, and William Billingsley. (2021). Data sharing platforms: How value is created from agricultural data. *Agricultural Systems*, 193, 103241.

Nargesian, Fatemeh, et al.(2019). Data Lake management: challenges and opportunities. *Proceedings of the VLDB Endowment*, 12(12),1986-1989.

Naqvi, Rabab, et al.(2021). The nexus between big data and decision-making: A study of big data techniques and technologies. *The International Conference on Artificial Intelligence and Computer Vision. Springer, Cham*, 2021.

Marz, N., (2011). How to beat the cap theorem. Online article, July URL: http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html.

Giebler, C., Stach, C., Schwarz, H., Mitschang, B., (2018). Braid. *Proceedings of the 7th International Conference on Data Science, Technology and Applications, SCITEPRESS-Science and Technology Publications, Lda*, 294-301.

https://www.ipvs.uni-stuttgart.de/departments/as/publications/stachch/data_18_braid.pdf

https://csimq-journals.rtu.lv/article/view/1548

Benjelloun, Sarah, et al. (2020). Big data processing: batch-based processing and stream-based processing. *Fourth International Conference on Intelligent Computing in Data Sciences (ICDS). IEEE*, 2020.

Monteith, J. Yates, John D. McGregor, and John E. Ingram.(2013). Hadoop and its evolving ecosystem. *5th International Workshop on Software Ecosystems*,50, 2013.

Oussous, Ahmed, et al. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4), 431-448.

Condie, Tyson, et al. (2010). MapReduce online. *Nsdi*. 10(4). 2010.

Shaw, Scott, et al. (2016). Hive architecture. *Practical Hive. Apress, Berkeley, CA*, 37-48.

Prasad, Bakshi Rohit, and Sonali Agarwal. (2016). Comparative study of big data computing and storage tools: a review. *International Journal of Database Theory and Application*, 9(1).