International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

Modernizing Legacy Big Data Platforms Using Cloud-Native Lakehouse Architectures

Karthikeyan Selvarajan

Senior Staff Software Engineer, San Francisco, California, United States

karthik.selvarajan83@gmail.com

ABSTRACT: The capacity to scale up the infrastructure, intricacy of the infrastructure, utilization of resources, governance of data, and compatibility with new analytics are just a few issues with the old models of the Hadoop-based big data systems of the past. This research paper suggests the implementation of cloud-based lakehouse architecture to upgrade the outdated big data systems implemented on Hadoop that has been integrated into the existing data and analytics without affecting the existing data resources. The suggested solution will result in a scaled and expandable data management space in cloud object storing methods, Apache Spark, Databricks, distributed data processing, an open table format, and centralized governance. A measured mode of modernization, which includes an assessment of the workload, data migration, transformation of the layers of processing, the addition of metadata, governance, and optimization of the workload, is planned. It also allows the storage and compute capabilities to be independently scaled as well as having the ability of running batch processing, streaming analytics, machine learning, and business intelligence on the same platform. The framework adds controls on data quality and metadata, security policies, lineage Garnier, cost-sensitive deployment of resources, to the functionality of the traditional Hadoop deployments. Architectural considerations show that the proposed lakehouse model has the ability to ease the management of the infrastructure, enhance resource elasticity, reduce platform fragmentation, and allow various analytical workloads. The paper demonstrates how the ideas of the lakehouse that are cloud native provide an organized guideline to organizations that are planning on upgrading their past big data infrastructures without sacrificing the interoperability, governance and operational continuity.

KEYWORDS: Hadoop Modernization, Data Lakehouse, Cloud Migration, Databricks, Apache Spark, Cloud-Native Architecture

I. INTRODUCTION

A huge volume of data exchanged between enterprise applications, connected devices and digital services and business processes has become a crucial component of the information systems of the present days because this growth has blistered. However, the standard Hadoop systems have offered companies the ability to realize the scale performance of distributed storage and processing operations, and has allowed companies to build large scale data analytics. The issue with older deployments, however, is that extending it is now hard due to complexity of the infrastructure, integration of different technologies, overhead operations and non-elastic. Meanwhile, organizations need such systems too that can provide business intelligence, real-time analytics, machine learning, and mass-scale data engineering in one platform. All these are demands that have prompted the idea of adopting cloud native data architectures instead of adopting the previous Hadoop systems.

The scale allowed organizations to keep heterogeneous structured data, semi-structured and unstructured data in one location so that the data lakes could become a novel highly meaningful alternative to much more strict data warehousing domains. Nonetheless, the obstacles that would come with the classical data lakes may be in the state of data quality, data administration, metadata handling, discoverability and trusted transactional information processing [1]. The lakehouse architecture will help remedy some of these weaknesses by guaranteeing the scalability and flexibility of data lakes with features outlined like those in data warehouses. It offers a single common platform, on which common data resources can be operated to do data engineering, including business intelligence, machine learning, and advanced analytics [1]. This model is especially relevant to those companies which wish to modernize the legacy platforms avoiding developing different sets of systems to accommodate the various workloads of analytics.

The cloud object storage has emerged as one of the eminent pillars to this modernization. Cloud-native architecture supports the ability to scale each layer independently as required by its workload, unlike traditional tightly coupled,

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

storage and compute architecture where the two components are not independent of each other. In this model, transactional storage technologies supporting ACID, schema control, versioning and trustworthy data operations of cloud object stores also come in handy [2]. These abilities can break significant shortcomings of traditional data lakes and give a more reliable base of enterprise analytics.

The actual challenges concerning implementing the concept of the data-lake also confirm that the notion of the phenomenon of modernization cannot be reduced to the idea of shifting the existing datasets to the cloud-store. The environment of business intelligence needs to be provided with accessible, good, regulated and well formatted data [3]. Legacy to legacy migration migration would then rely on the functionality of data movement and data processing workloads, metadata, governance, security, quality and access of the analytics. The evolution of the overall concept of the data lakes has also followed the complaints of data structure, metadata management, quality control, and data search to the severe architectural issues [4].

This flexibility of data lakes is also appropriate since organizations keep getting data of various types and forms. When data is initially ingested, heterogeneous data can be stored in a data-lake-based architecture without strict schema definition [5]. The infinity elasticity however makes the picture more complex, when datasets are required to be scaled up in future to support reporting and analytical workloads to the enterprise. The modern lakehouse then demands a system, which encompasses the ability to flexibly ingest, as well as limited schema, transactional coherence, management and effective analytical access accountability.

Another important factor of modernization is the processing performance. Lakehouse platforms Sharing with modern, lakehouse providing is currently executing the analysis workload on large datasets in accelerated query engines [6]. Other distributed processing systems like the Apache Spark also offer greater functionality when it comes to the capability to process large scale data, analytics and pipeline development [7]. The technologies also open the possibilities of configuring the traditional Hadoop processing pipelines to be more elastic into cloud-based processing pipelines and of both the batch oriented and advanced loads of analytics.

Smart data management is also important to successful modernization. There may be thousands of datasets that are distributed by domains, sources and form, and which can be located on large-scale cloud platforms. Metadata-management and data-discovery mechanisms should thus be availed to get the data assets in a maintainable and usable state [8]. The importance of the interaction between storage, ingestion, processing, governance, and access functions as one unit in a platform can also be additional to the influence of the architectural strategies on data lakes [9]. In addition, data locality and data movement will also be taken into consideration as massive migration can also introduce network overhead, network performance bottlenecks, and can inscribe additional infrastructure costs [10].

Regardless of these developments, any organization with a history of Hadoop systems must still have a clear roadmap, which will make sure that it switches to new cloud-based systems. The current solutions are grounded on preoccupations like how lake houses should be designed, transactional storage, control over information lakes, or even the mass processing but not on the overall view of modernization. The paper thus suggests that one can use cloud-native lakehouse architecture to upgrade outdated big data architecture, both in terms of cloud object storage and moving workloads, Apache Spark, lakehouse storage, scale compute, governance, metadata management, and moving workloads. It has been aimed to offer an architectural concept, which can be used to achieve the highest scalability, operational flexibility, analytical as well as resource efficacy, and allow an organization to humbly contemporary urbanize existing large-scale data environment.

II. RELATED WORK

The pushes to scale-up big data platforms have, primarily, gone beyond the conventional data warehouses to Hadoop-supporting and cloud-native data lakes to join up integrated and scaled components of data platforms. The initial generation data lake tools encompassed the storing of the big data in somewhat scalable data stores that were prone to store a variety of data. However, the increasing need of legitimate transactions, management, analytical capability, and delivery of diverse workloads has demonstrated deficiencies with standard deployment of data lakes. New research therefore concentrates on the lakehouse designs that unify the dynamism quality of data lakes and integrity in addition to insightfulness of data warehouses.

The foundation of the integration of data engineering, business intelligence and advanced analytics into a platform is the Lakehouse paradigm. In [1], the article provided the lake house as open data-management architecture to address

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

the challenges of lack of integration between data-lake and data warehouse. The architecture will be oriented at cost-effective storing of objects, it will be able to access the various types of data directly, high reliability in transactions, good data governance and ability to support machine learning and business intelligence workloads. This is especially the case in modernization where it offers an avenue through which the organizations are beginning to lay down their back heritage data-processing service islands, and begin to join up into common analytical environments. But when one decides to introduce this type of architecture into the already existing installations of Hadoop one needs to be very careful regarding the problems of migration, interoperability and manipulation of workloads as well as months of operating shortages.

The other nice thing that would help to upgrade the past big data systems would be transaction management. The approach described in [2] opened the idea of Delta Lake, or storage layer, which offers the ACID transactions, scalable metadata processing, schema enforcement, and time-travel characteristics to the cloud object storage. These functions run under some of the limitations of traditional data lakes where simultaneous data alterations, inconsistency between data set and schema change may complicate data analysis functions. With legacy Hadoop platforms, these features provide mechanisms of reliability improvement, separation of storage and compute resources. This is especially notable in cloud-native workflows where processing units can be autonomously expanded based on the load demands.

In [3], the hands-on data lake based business intelligence is also learned. The paper has touched upon organizational stories of implementation of data-lake, and has identified the gap that may exist between the technical potential of data-lake and its application in reporting settings. Governance, integration, issues with data quality and data accessibility among other problems are some factors that other than rendering a data lake less efficient lack proper architecture controls. These perceptions are critical in the elements of modernization since simple data transfer of available Hadoop data at the cloud storage cannot be equated to better outputs of the analysis. Modernization to date should thus be inclusive of governance, metadata control and data and measures workload optimal.

In [4] the larger trend of development was covered and the issues of the data-lake structures were discussed. The article acknowledges that data lakes constitute one of the useful ways of managing heterogeneous and bulky data pointing out challenges associated with metadata, organization, data quality and information search. All these issues are overdramatized as the volumes of data platforms are increased to numerous diverse loads of analysis and operations. One of the ways a cloud-native lakehouse can alleviate these risks is via the combination of integrated centralized governance and semi-structured and structured data management. This transition however, comes with some architectural options to deal with the forms of storage, processing engines, metadata services, security and lifecycle management.

In [5], a conceptual basis of data lakes was developed with the data lake being offered as an alternative to the inflexible traditional data-management architecture. It has a consumption orientation towards a variety of datasets and elaboration of the heavy requirements of aggregation. The flexibility is suitable in organizations which handle fast varying sources of data and heavy workloads. It also results in the issues with data quality and discoverability, though, owing to the unlimited flexibility. Current trends of modernization architectures necessitate trade-off in the schema flexibility and governance in that light.

The other significant contributing factor of adoption of lakehouses is performance optimization. The authors have suggested Photon in [6] that is workload unique high-performance query engine in lakehouses. The focus on running efficient queries can be leveraged to show how the current analytic engines can enhance performance of process without organizations necessarily having to trade-off the compute as well as the storage infrastructure. The use of optimized query engine can be utilized to augment distributed processing structure when modernizing a legacy platform as well as friendliness and interactive load in analytical processing and conventional batch processing.

Apache Spark has been considered as a platform of processing the data-lake as well. The location as reported in [7] demonstrates that you can create an instance using Spark as a data lake that has the capability of addressing huge data-processing needs. This distributed execution nature, together with the batch and analytical workload has helped Spark to be used in the migration strategy where Hadoop processing loads are slowly being transformed to the present-day cloud-based pipelines. But Spark-based modernization should keep in mind the compatibility of applications, the distribution of resources, the migration of data-formats, and optimization of the current processing processes.

The research of intelligent control of data lakes has been undertaken in the Constance system in [8]. The system is able to solve the problem of data discovery and data management, as it proposes smart ways of arranging and retrieving

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

heterogeneous data. These are applicable in cloud-native modernization since the migration process normally leads to more than one datasets, source, schema and analytical consumer. Less complexity in the operations can therefore be achieved by good metadata and discovery thereby making the modernized platform more consumable.

The approach of recommended architectural advice to data lakes is given in [9] and the storage, ingestion, processing, and governance as well as access layers should be visualized. The legacy modernization is open to this multi-dimensional solution because nowadays the Hadoop-based environments can be characterized by the large amount of highly-integrated units and workload-related dependencies. Decoupling these duties in a cloud-native design facilitates more elasticity and allows a smaller-scale migration, instead of a wholesale change in platform.

Finally, in [10], the authors have examined the notion of personal data lakes and data gravity and the impact that the physical location of data and its movement have on data-management architectures. The concept most applicable is data gravity, which in the process of cloud migration is most relevant, though it can also present issues of network, performance and cost to the relocation of huge legacy data sets across environments. All of these assist migration strategies which measure the variables of locality of data, where it is stored, the dependency of the work load and the necessity of processing.

All in all, the analyzed literature prepared the technological ground to modernization on a lakehouse but offers an opportunity of a specific framework that could deal with the legacy Hadoop-to-cloud migration. The existing literature has focused primarily on single aspects of lakehouses, transactional storage, data-lake management, query performance or Spark processing. Much of the proposed study is anchored on these foundations with the additions of cloud-native storage, distributed process, lakehouse, governance and workload migration as well as operations modernization combined in one architecture.

III. CLOUD-NATIVE LAKEHOUSE MODERNIZATION ARCHITECTURE

The architecture proposal offers a roadmap based on which the existing infrastructure of Hadoop could be upgraded to scalable lakehouse infrastructure as a cloud service. The architecture does not view modernization as data narrowing of data on-premises infrastructure to cloud infrastructures, but rather desegregations storage, computation, ingestion, governance, metadata and analytical services into on-premises over layers, which are readily supported. This kind of disconnection gives organizations the ability to progressively update workloads on-premise, without impacting business continuity and data access. The system unites the impact of cloud object storage, Apache Spark, and lakehouse table management, metadata services, and governance mechanisms, and analytics interfaces into a single platform.

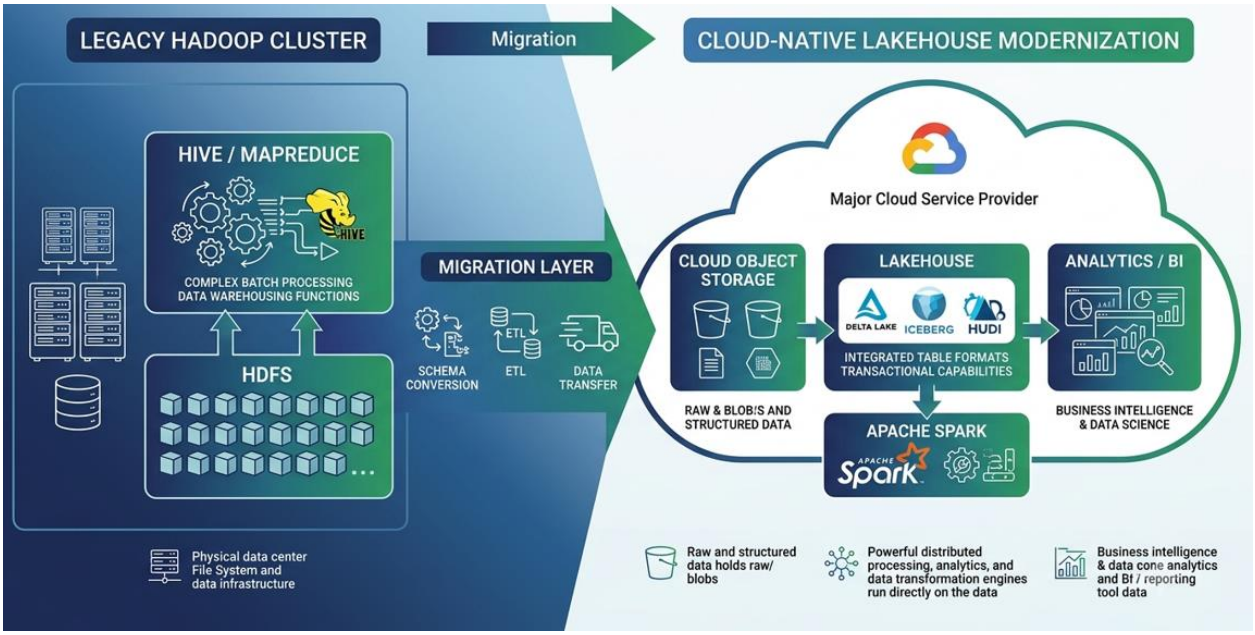

Figure 1 — Legacy Hadoop to Cloud-Native Lakehouse Modernization

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

3.1 Legacy Platform Assessment and Migration Layer

The Hadoop modernization project begins with the assessment of the existing Hadoop ecosystem. Legacy environments have HDFS, MapReduce jobs, Hive tables, Spark workloads, Sqoop pipelines, streaming elements, and custom scripts, as well as, enterprise reporting applications. All the workloads are segmented according to the data dependency, processing pattern, business importance, the performance required and complexity of the migration.

This has workload inventory, to identify the datasets and applications which can be directly migrated or modified to cloud-native pipelines or stored temporarily in the legacy environment. Dependency analysis can be quite essential since Hadoop workloads can be filled with dependencies between storage locations, processing, metadata storage and downstream locations. This layer, thus, forms a measured transition point between the current infrastructure and the desired lakehouse.

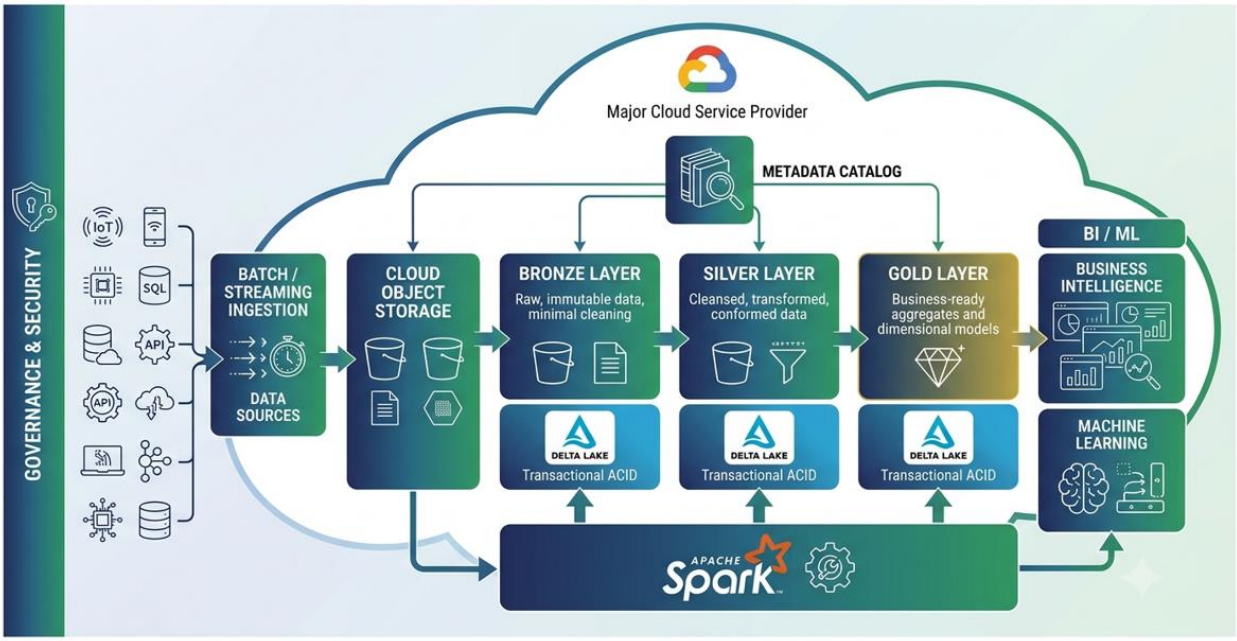

Figure 2 — Proposed Cloud-Native Lakehouse Architecture

3.2 Cloud-Native Ingestion Layer

Ingestion layer bridges the gap between the outdated system and the new platform by connecting the enterprise sources of data to the updated system. It is capable of supporting structured, semi-structured, and unstructured information, which is sourced both out of relational databases and enterprise applications, APIs, files, IoT systems and logs as well as the existing Hadoop repositories.

All the ingestion mechanisms are included in processing in Form of a batch and streaming processing to suit varying work load needs. Unlike streaming pipelines, the transfer of historical and transactional data via batch pipeline can be periodically done, but new events can be triggered continuously with a streaming pipeline. Incremental synchronization of the legacy databases and cloud storage can also utilize the change-data-capture mechanisms.

One of the significant architectural practices is incremental migration. It can also migrate datasets based on business priority and technical dependency as opposed to the whole of the legacy environment. Not only is this an efficient way of minimising the risk of migration, but also a way of ensuring that it can carry old and new workloads at the same time during the transition.

3.3 Elastic Lakehouse Storage Layer

It is a design that comprises of the main center of a lakehouse over the foundation of a cloud object-store. The scaled-out object storage can be scaled to extreme levels and can be maintained and the data capacity and compute resources decoupling. The data are then structured into rational areas like the raw, refined and curated layers.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

Raw layer contains data that is ingested either in raw form or in a barely processed form. The clean layer introduces the validation, cleansing, standardization, and schema controls. The pre-prepared machine learning workload includes business-ready datasets which are capable of enabling reporting, analytics and machine-learning performance.

A transactional lakehouse storage layer is placed on top of object storage, to offer services like ACID, schema, version management and concurrent access. These characteristics negate many of the constraints which are conventionally associated with traditional data lakes and allow the platform to equally support both the workload of data engineering tasks as well as the workload of analytical tasks.

3.4 Distributed Processing and Transformation Layer

Spark is Apache that will be a processing capability of materials in the proposed system. It provides distributed implementation of numerous scale change, aggregation, characteristic preparation and analysis processing. These workloads of the old MapReduce can be checked and moved over to processing in Spark at its need step by step.

The processing layer supports scheduled batch, and continuously flowing, data. This is possible with workload adjustment as resource allocation can be done with workload adjustment, as the workload is not involved with the persistent storage through the decoupling of the computation. Then compute resources do not necessarily need to be of the same capacity as the underlying data.

The optimized SQL query engines can also be mixed with the architecture to obtain interactive analytical workloads. It may allow a single, powerful information foundation list to power facts engineering, SQL software, machine learning and business insight without divergent copies of the same information collections.

3.5 Metadata, Catalog, and Governance Plane

A governance plane is not a component and thus has cross-relations across the architecture. The centralized catalog retains all the information on datasets, ownership, classifications, relationship and usage. Metadata is used to inform the users and applications of the important datasets without necessarily searching the underlying storage mechanisms.

The flow and transformation of ingested, processed and consumed data can be tracked using the data lineage mechanisms. This is especially critical where it involves the process of modernization because one will have to have a feel of how it will turn out in pivoting the heritage datasets into the new environment by organizations.

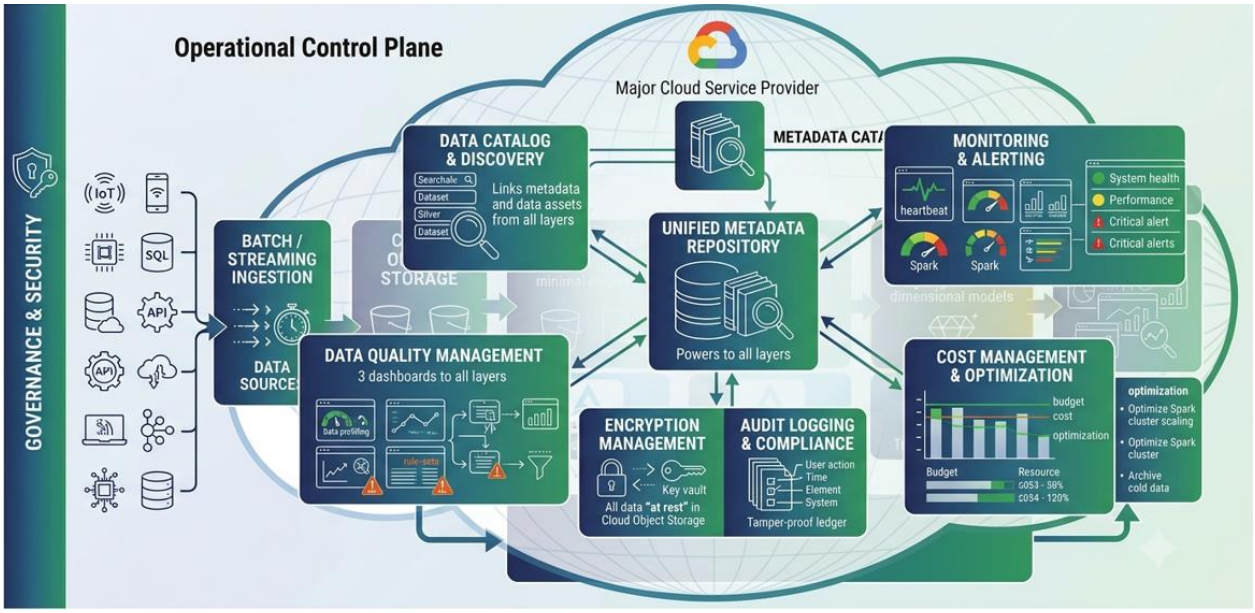

Figure 3 — Governance and Operational Control Plane

There is a governance policy that restricts the user, role, application and sensitivity to information. To guard information in the enterprise, encryption, authentication, authorization, auditing, retention policies and rules on data-

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

quality can be included. Schema validation and quality checks are established in processing and absorbing data to discourage the circulation of unreliable data to the refined levels in analysis.

3.6 Lakehouse Analytics and Consumption Layer

The consumption layer is in control of the modernized data platform. Curated datasets can be used to query dashboards, reporting and decision support applications using business intelligence application. Governed datasets can be accessed by data scientists in workflows of machine-learning, whereas data engineers can create and observe pipelines of transformations.

Interestingly, it is due to the common shared controlled data resources of the different analytical consumers as opposed to data silos being isolated. This not only prevents redundant duplication, but also makes operations reporting, analytical processing and machine-learn applications to be uniform.

3.7 Observability, Security, and FinOps

The existing platforms based on clouds necessitate constant control over its activities. This empowered architecture then entangles a layer of observability, which oversees the break-even of the pipeline, processing latency, storage-utilization, query-performance, data-quality metrics and resource-utilization.

These security restrictive measures are enforced at the application interfaces, networking and storage and processing interfaces. Identity-based access control (UNDatabase management System Access control) and least-privilege ensure that only authorized persons operate the system and audit logs enable tracing of the significant data operations.

Cost management is the other important aspect of cloud modernization. As storage and compute are independent to scale, the organization can scale dynamically the resources of processing with the demand of the workload. Cloud spending can also be put under control by automated shutdown of idle resources, scheduling of workloads, storage lifecycle policies and consumption monitoring.

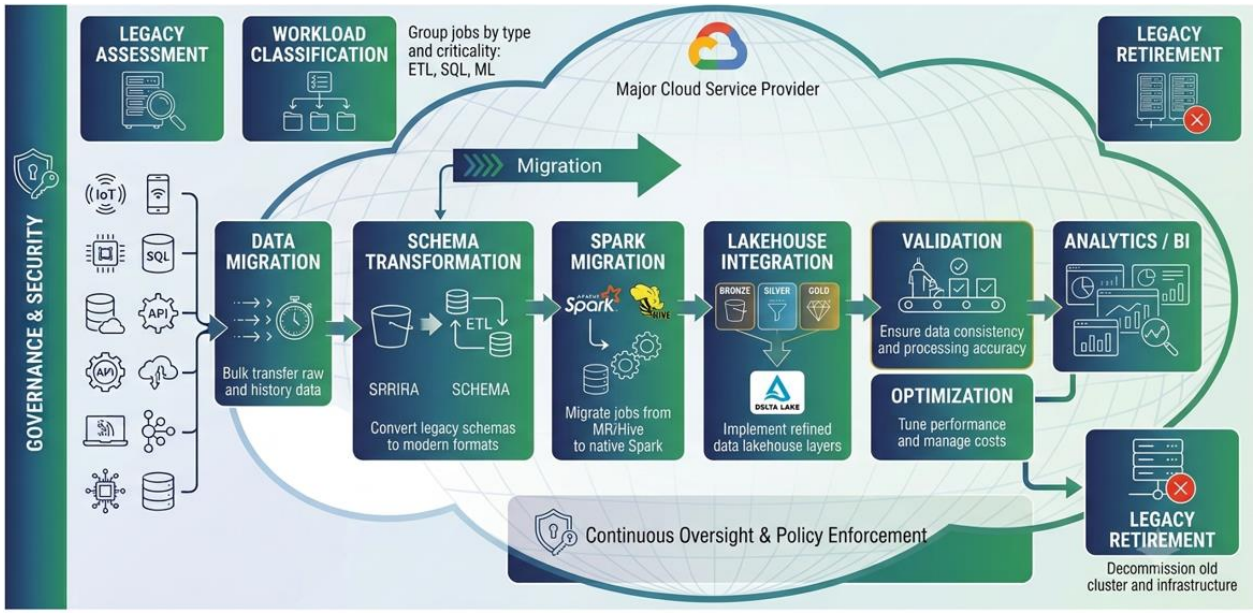

Figure 4 — Phased Hadoop Modernization Workflow

IV. MODERNIZATION VALIDATION AND ARCHITECTURAL ASSESSMENT

The analyzed cloud-native lakehouse architecture is based on an analysis of the architecture in terms of scalability, processing efficiency, migration, governance, reliability, and the operational control. This will be evaluated by the degree of working around the perceived constraint inherent in the legacy Hadoop environment, and the one serving the current analytical loads.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

4.1 Scalability and Elasticity

An advantageous scaling benefit is in storage and scaling of computers. Cloud object storage need not be proportional increment of processing infrastructure as it is able to accommodate the increasing amount data. Scale Compute resources can be scaled depending on workload requirements, and batch workloads, interactive analytics and machine-learning workloads can be scaled autonomously. The model minimizes the reliance on permanently provisioned Hadoop clusters and dynamically replicates the enterprise workloads in a more efficient way.

4.2 Processing and Query Performance

Spark is a distributed processing framework in order to support the large volumes of changing and analysis. It migrates the previous workflows of processing in Spark pipelines step by step due to its architecture. Optimized lakehouse query engines can also additionally support interactive SQL workloads. Processing latency, query response time, throughput, resource utilization and workload completion time are some of the performance assessment that can be taken into account. The networked processing and scalable compute infrastructure can enable a scaled underpin of heterogeneous tasks.

4.3 Migration Continuity and Interoperability

Migration flexibility is quantified in terms of capability of the architecture in order to facilitate in transitioning between Hadoop and cloud infrastructure in a gradual migration. Existing datasets and workloads can be classified according to the complexity of the migration of the dataset and the business priority and technical dependencies. Simultaneous support of a legacy and migrated workload assists in testing migrated workloads, which can then be moved to production. Storage formats that are interoperable and distributed processing frameworks can minimize the reliance of a single processing environment and ease gradual modernization.

4.4 Data Governance and Reliability

An additional centralized metadata, lineage, access control, schema management and data-quality mechanisms are provided in the architecture. These aspects are evaluated in regards to the discoverability of the data features, traceability, policy and consistency. Transactional lakehouse storage provides the reliability of data operations running concurrently and controlled mechanism of schema evolution. The architecture also meets functionality criteria of governance and reliability that are hard to achieve with loosely controlled traditional data lake.

4.5 Operational Efficiency and Cost Management

Operation evaluation takes into consideration the infrastructure operation and resource utilization, monitoring and cloud usage. Scaling of compute and storage independently, can minimize the need to have constantly running infrastructure. The utilization can further be improved with the help of automated workload scheduling, lifecycle control and resource monitoring. To aid in proactive operation management, observability mechanisms make the usage of pipelines observable in a failed state, processing latency, query responsive, and resource using.

4.6 Overall Architectural Assessment

The analysis shows that the proposed architecture will enable the adherence to a structured modernization course instead of having to replace the infrastructure directly. Its most significant characteristics are; elastic scaling, the ability to have a single data management system, incremental migration, controlled access and flexing workloads. However, the modernization continues to depend on challenges such as the intricacy of migration, compatibility of the former applications, requirements of data transfer, capability to utilize the clouds, maturity of governance and optimization of workload. This ought to be part of the implementation planning in that the better results in the operations by operation modernization of technical modernization would be achieved in a sustainable operation.

V. CONCLUSION AND FUTURE WORK

The re-invention of older forms of Hadoop-based big data systems is increasingly becoming relevant as companies outsource scalable, heterogonous and controllable environments to be capable of performing a variety of analytical operations. This research paper came up with a lakehouse architecture built in the clouds and includes the cloud objects storage, sparks Apache server, transactional lakehouse, which includes metadata management, governance, in security and analytical service. Separating storage and processing in structure allows its allocation to dynamically and place constraints on the infrastructure requirement of the conventional deployments of the Hadoop. Slow modernization approach based on evaluation, categorization of workload, gradual transition, adaptation and validation followed by optimization allows offering pursuant pathfinder in ensuring even further modernization of the legacy environments without introducing any disruption in the operation of same. According to the architectural review, the suggested

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March-April 2024||

DOI:10.15662/IJRPETM.2024.0702008

implementation was in a position to accommodate the heterogeneous workloads, enhance accessibility of the information, back governance and offer a central repository of business intelligence, data engineering and advanced level analytics. However, along with the interest are the complexity of migration, data compatibility, movement and cost of cloud and organizational preparedness.

The architecture will be extendable in the future by advancing AI-aided migration planning, automation in the area of workload dependency discovery, and intelligent reprocessing of retired Hadoop and MapReduce jobs in native pipeline processing on clouds with the help of AI. Substantial improvements can be made in the future by exercising autonomous resource optimization via workload-aware scheduling and predictive scaling. Its seamless approach in the multi-cloud environment and the hybrid-cloud can enhance the mobility and decrease the reliance on some cloud environments. This could be further explored in the automated testing of data-quality, the creation of semantic metadata, lineage intelligence, and policy-conscious governance. Lastly, systematic validation of many workloads deployed to an enterprise could offer greater insight into the effort required to migrate, relative performance of the processing when compared to the extent of operational efficiency, resource utilization, and the overall cost of ownership.

REFERENCES

[1] M. Armbrust *et al*., "Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics," *Proceedings of the VLDB Endowment*, 2021.
[2] M. Armbrust *et al*., "Delta Lake: High-performance ACID table storage over cloud object stores," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 3411–3424, 2020.
[3] M. R. Llave, "Data lakes in business intelligence: Reporting from the trenches," *Procedia Computer Science*, vol. 138, pp. 516–524, 2018.
[4] H. Decker *et al*., "Data lakes: Trends and perspectives," *Proceedings of the VLDB Endowment*, 2019.
[5] P. P. Khine and Z. S. Wang, "Data lake: A new ideology in big data era," *Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud)*, 2018.
[6] A. Behm *et al*., "Photon: A fast query engine for lakehouse systems," *Proceedings of the VLDB Endowment*, 2022.
[7] T. Y. Chen *et al*., "On construction of a power data lake platform using Spark," *International Journal of Computer Science and Information Security*, 2019.
[8] R. Hai, S. Geisler, and C. Quix, "Constance: An intelligent data lake system," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2016.
[9] A. Laplante *et al*., "Architecting data lakes," *IEEE Software*, 2016.
[10] C. Walker *et al*., "Personal data lake with data gravity pull," *Future Generation Computer Systems*, 2015.