ZKG INTERNATIONAL
ISSN / eISSN 2366-1313
Publisher BAUVERLAG BV GMBH,
AVENWEDDER STR 55, GUTERSLOH, GERMANY, 33311
Volume 25-Issue 1-(2025)
https://doi.org/10.63386/618416

OPEN ACCESS

Hadoop-Based Big Data Governance Platform for Multi-Source Heterogeneous Systems: Architecture Design and Cross-Platform Integration

Xue Teng[1,*], Xianqing Wang[1]
1, Department of Library, Guangdong Polytechnic of Science and Technology, Zhuhai, Guangdong, 519090, China.
leo1210@163.com
Corresponding author: Xue Teng, (e-mail: leo1210@163.com)

Acknowledgment:

This paper is one of the phased research results of the 2023 Zhuhai Philosophy and Social Science planning project "Research on Big Data Governance Mechanism of Public Culture in Guangdong-Hong Kong-Macao Greater Bay Area" (Project number: 2023GJ062).

Abstract

The fast growth of big data in various industries has resulted in data ecosystems made up of many different and spread-out sources. Sufficient governance in such surroundings is needed to maintain the accuracy of data, follow regulations, ensure security, and improve operations. This study explains the process of creating, deploying, and assessing a Hadoop-based Big Data Governance Platform meant for use in multi-source and heterogeneous systems. The suggested architecture relies on the modular components of Hadoop, such as HDFS, Hive, Spark, Atlas, and Ranger, to manage metadata, enforce rules, follow data lineage, and integrate data with different platforms. Hadoop allows data to be streamed in real time and in batches from relational, NoSQL, and cloud databases using Kafka, Flume, and Sqoop. Testing with ten nodes in a Hadoop cluster and a variety of datasets shows that the platform provides thorough metadata, follows all policies consistently, is easy to track data lineage, and performs queries quickly without causing much overhead. What's more, the use of visual analytics and profiling greatly supports data quality and ensures that Sisense works with PostgreSQL, MongoDB, and S3. Evidence shows that Hadoop-based management ensures that companies can handle the challenges and variety involved in big data governance.

Keywords:
Big Data Governance, Hadoop, Apache Atlas, Cross-Platform Integration, Metadata Management, Policy Enforcement, Data Quality, Heterogeneous Systems, Spark, Apache Ranger.

1. Introduction

The rise of big data is thanks to rapidly increasing data from various sources, such as mobile apps, IoT devices, systems from businesses, social media sites, and services in the cloud. When looking at data from the perspective of its growing size, speed, variety, quality, and importance (the 5Vs), it causes problems for data management, especially with data from many different systems that use various formats (Laney, 2001). Gandomi & Haider, 2015). As structured and single-center data was managed by data management systems in the past, they are now inadequate for handling today's complex and widely distributed data environments (Hashem et al., 2015).

Big data governance is important now as it brings together the methods, rules, frameworks, and technologies needed to control how data is managed in an organization. Governance ensures data assets are handled properly across the business, so the organization can comply with regulations, use data for decision-making, and build trust in what analytics shows (Otto, 2011). Khatri & Brown, 2010). Nevertheless, it is difficult to bring order to big data environments because each platform, data model, and processing workflow is usually unique (Abiteboul et al., 2012).

Hadoop is now a key technology for managing the storage and analysis of large volumes of data (White, 2015). By using Hadoop Distributed File System, Yet Another Resource Negotiator, and the MapReduce model, Hadoop enables handling large amounts of data in a scalable and fault-tolerant way. Using Apache Hive, Spark, Sqoop, Flume, and Pig, Hadoop has become a strong platform that can manage complex data processing needs and analytics tasks. The Apache Atlas and Apache Ranger recent additions give users the ability to access metadata, track data lineage, and set up policies for security (Apache Atlas, 2023). Apache Ranger, 2023).

However, most Hadoop implementations are currently bound to the Hadoop ecosystem and have difficulty handling governance in different, mixed platforms. Most enterprises utilize hybrid architectures that rely on Oracle, SQL Server, MongoDB, Cassandra, Amazon Redshift, Google BigQuery, and cloud storage such as AWS S3 and Azure Blob Storage. Integrating and governing

data from different systems needs a platform that can work together, keep track of similar information, enforce the same rules everywhere, and create detailed logs for tracking changes and activity in the data. Cheong & Chang, 2007).

Current studies have examined separate issues of handling distributed data and controlling access to that data. For example, in 2018, Ghosh et al. designed a governance approach using Apache Atlas, though their focus was on data from Hadoop systems only. Singh and Thakur (2020) mentioned data security and privacy in Big Data, stressing the need for policies and implementation, while not discussing integration issues in other systems besides Hadoop. Echoing Choo (2014), others pointed out the value of metadata in finding knowledge, although they did not explain how it applies to diverse systems.

To address this gap, the paper suggests a Big Data Governance Platform built on Hadoop that supports the platform's native features along with adding integration and governance for many different types of data sources. The architecture calls for custom ingestion tools, a multi-layered processing method, and connectors that make it easy for different systems to talk to one another. Furthermore, these solutions feature methods for managing essential data such as centrally storing metadata, retrieving data lineage, controlling who has access, and oversight of data quality to make certain the system is scalable and flexible.

2. Literature Review

The widespread use of big data technologies in recent years has resulted in growth in data governance frameworks to support data quality, security, opening up access to information, and meeting compliance requirements. Numerous studies have examined the challenging aspects of big data governance from various viewpoints, mainly in complicated settings that contain a mix of different systems.

Handling the variety in data sources is one of the key problems in big data governance. Today, data can be classified as structured data from relational models, semi-structured data found in logs and structured text, or unstructured data like images and videos on the internet and social media networks. The study by Taylor et al. (2015) shows that different types of data lead to confusion in meaning and variation in form that prevent proper data management. Effective governance,

therefore, must start with a good system for managing metadata, so that data meaning and terminology is consistent everywhere.

Many experts have argued that metadata should guide how information is managed. Jain and Thakurta (2017) claim that without metadata acting as the spine, key governance processes such as tracking lineage, controlling versions, aligning schemas, and ensuring compliance cannot function properly. They suggested a system where metadata is stored and managed across different platforms. In another study, Zhao et al. (2019) used semantic metadata methods to unite data from multiple domains, showing it led to better query performance and preparedness for audits.

Researchers often pay special attention to data quality since it is crucial for live data streams and various inputs. According to Batini and Scannapieco (2016), there are four main groups of big data quality challenges. difficulties related to intrinsic, contextual, representational, and accessibility aspects. They found that having ongoing monitoring systems and data quality dashboards is necessary to monitor the accuracy, timeliness, completeness, and consistency of the collected data. Since Hadoop values speed over security, setting up these controls is still challenging in Hadoop platforms.

Various studies have focused on ensuring policies are followed and data is protected in distributed systems. Chen and Zhao (2014) devised a security architecture that works hand-in-hand with the basic Hadoop security settings, relying on data tags and rule-based access engines. Nevertheless, they pointed out that using these controls on a larger scale and integrating them with third-party services proved to be very difficult. Bagui and Dhar (2021) recently showed that ABAC works effectively in hybrid cloud setups and is particularly useful for managing multiple users in the same environment.

The compatibility between big data and old systems has been analyzed in detail, but the focus on governance is generally lacking. Kambatla et al. (2014) focused primarily on boosting performance of Hadoop and Spark on various servers, but they did not consider data integrity, tracking audits, or overseeing compliance measures. On another point, De Oliveira et al. (2017) examined data lake governance in enterprises and found that weak integration and no central administration often led to data swamps instead of clean and organized data lakes.

There are also research efforts toward building data governance maturity models that suit big data. Khatun and Walters (2018) suggested a scale that helps assess how well an organization manages governance by evaluating it at different maturity stages. First, keep it repeatable, defined, managed, and optimized. The model included factors such as stakeholder participation, automating business processes, and looking after data. Nonetheless, it was challenging to use it in Hadoop because there was no clear guidance on how to make it work operationally.

Adopting data governance models customized for cloud computing is becoming more popular. According to Farahani and Bahadori (2021), designing government methods has to be relevant for containerized and serverless architectures. In their view, these kinds of environments must be managed with governance that is flexible and capable of changing data and infrastructure.

There is an increasing interest among academics in understanding how ontologies and knowledge graphs help in data governance. According to Wang et al. (2020), an ontology-based governance approach improves the process of finding and combining data from different fields. They showed how it could be implemented in smart cities, joining data from different sources and creating guidelines by using common definitions.

Literature discusses several open-source tools that help with governance. People often praise Apache NiFi for having a user-friendly interface and effective flow control (Fernandez et al., 2019). However, NiFi's governance capabilities are generally limited to managing flows, and do not provide comprehensive coverage for data. Just like Talend, Informatica's Axon tool is often checked in companies to see how their metadata cataloging and policy management work (Kiran & Palanisamy, 2020). Still, these tools usually find it hard to fit well in customized Hadoop systems, which leads to gaps in data governance.

Lastly, data governance covers important legal and ethical topics. Because the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) affect many industries, Lin and Esfahani (2020) have made it clear that strong automated systems are required for compliance checking, limiting data, and assisting with people's requests for their personal data. They claim that if compliance functions are not built into big data applications, companies will encounter greater risks and concerns about their reputations.

Overall, studies confirm that strong governance is crucial for addressing the risks involved in big data environments processing various forms of data. Even now, there are not many Hadoop-native tools that can govern all data, include data from outside, and work well while remaining compliant and scalable. The main objective of this study is to address that gap by offering and examining a Hadoop architecture that makes data exchange and integration possible across platforms.

3. Methodology

For this study, design science research was used to develop, implement, and assess a Hadoop-based platform that supports governance of big data from various and different sources. It includes designing the architecture, choosing the proper tools, integrating systems, and evaluating performance. This is done to make sure the proposed system can handle different kinds of data well, and also enforce policies on metadata, tracking data's history, controlling access, and monitoring data quality.

3.1 System Architecture Design

The platform is built on a layered design that separates handling ingestion, storing information, processing tasks, governing activities, and dealing with integration. By following the principles of modularity and scalability, the architecture became adaptable in multiple enterprise contexts. The layers in the framework were designed to work with particular open-source tools in Hadoop while including microservices that provided additional support for governance.

The main task of the ingestion layer is to manage data from both continuous and single event streams. Apache Sqoop was used to collect data from structured relational databases in batches, whereas Apache Kafka and Apache Flume were used to collect data from devices and application logs in real time. They allowed us to adjust to various data speeds and formats coming in from the different sources.

Unstructured and semi-structured data was saved in HDFS, and HBase was used for NoSQL storage to handle data that comes in fast and doesn't need a set structure. By using both a file and record system, it guaranteed compatibility for both simple text files and large binary or multimedia ones.

3.2 Processing Framework

Apache Hive and Apache Spark were used for the processing layer. On Hive, data analysts and compliance officers wrote SQL-like queries, while Spark handled transformation, cleansing, and advanced analytics using distributed processing. The system was set up to perform ETL processes that cleaned and normalized data as it entered. Livy helped link Spark to REST APIs so that external systems could initiate governance actions as needed.

3.3 Governance Layer Implementation

Apache Atlas was used as the main metadata repository to monitor metadata and data lineage. System administrators set up Atlas to collect Hive, HDFS, and Kafka metadata automatically afterward. To push data from sources external to Hadoop, Python and Java custom connectors were built to connect to Atlas using its REST API.

Apache Ranger was used to implement security governance. The configuration included RBAC policies for Hive tables, directories in HDFS, and Kafka topics. MySQL contained both audit trails from the rangers and policy enforcement logs that were visualized through Grafana dashboards. Thus, we were able to analyze the way data was accessed and spot any compliance issues.

A custom engine for data profiling was created using PySpark to watch over data quality. It used automated rules to check for null values, ranges, duplicates, and to ensure referential integrity. The findings were placed in Hive tables and periodically checked by a scheduler set up in Apache Oozie. The results of each quality check were recorded in the metadata repository, supporting effective management.

3.4 Cross-Platform Integration

An important point about the platform was how it interacted with other systems outside of Hadoop, such as PostgreSQL, MongoDB, and Amazon S3. Data from SQL-based sources was handled with JDBC connectors, and secure connections to NoSQL and cloud data were created using RESTful APIs and Apache Knox Gateway. Open metadata standards, such as OpenLineage and Egeria, were used to identify schemas from these platforms, in addition to working with Apache Atlas.

It also helped in translating Apache Ranger's governance rules into access policies for various outside platforms with its rule translation engine. This way, security and privacy policies were enforced in the same way no matter the location or way the data was managed.

3.5 Experimental Setup and Data Sources

The system was tested in a laboratory using ten machines loaded with Hadoop 3.3, Spark 3.0, Hive 3.1, Kafka 3.0, and relevant governance tools. All nodes featured 64GB of RAM, along with 16-core processors that ran on Ubuntu Server 20.04. Both real and artificial data were used in combination to simulate the complex data mix typically found in practice. Random customer profiles, fake transactions, and sensor data made up the synthetic dataset. Records from the hospital include patient data, while government data and public connection files were also included.

The metrics ingestion rate, query response speed, metadata capture time, how fast access control operates, and the accuracy of data quality checks were all monitored through Apache Ambari and Prometheus platforms. By comparing results with and without governance modules, A/B testing aimed to find the relationship between ensuring compliance and increasing throughput.

3.6 Evaluation Criteria

Both qualitative and quantitative indicators were used to measure if the platform was effective. Participants from the technical team were asked about their opinions on how easy the system is to use, how clear it is, and how it enforces policies. Quantitative metrics included:

This indicates the proportion of datasets in the collection where all metadata elements are captured.

This rate tells us how effectively access control policies were applied.

Evaluate how long it takes to perform Hive and Spark queries at different loads.

Successfully transferring data from one platform to another depends on Integration Accuracy.

Data Quality Score is calculated using pre-set models that work on mapping and analysis reports.

The results of these tests guided the analysis done in the following Results and Discussion sections.

4. Results

4.1 Metadata Completeness Across Systems

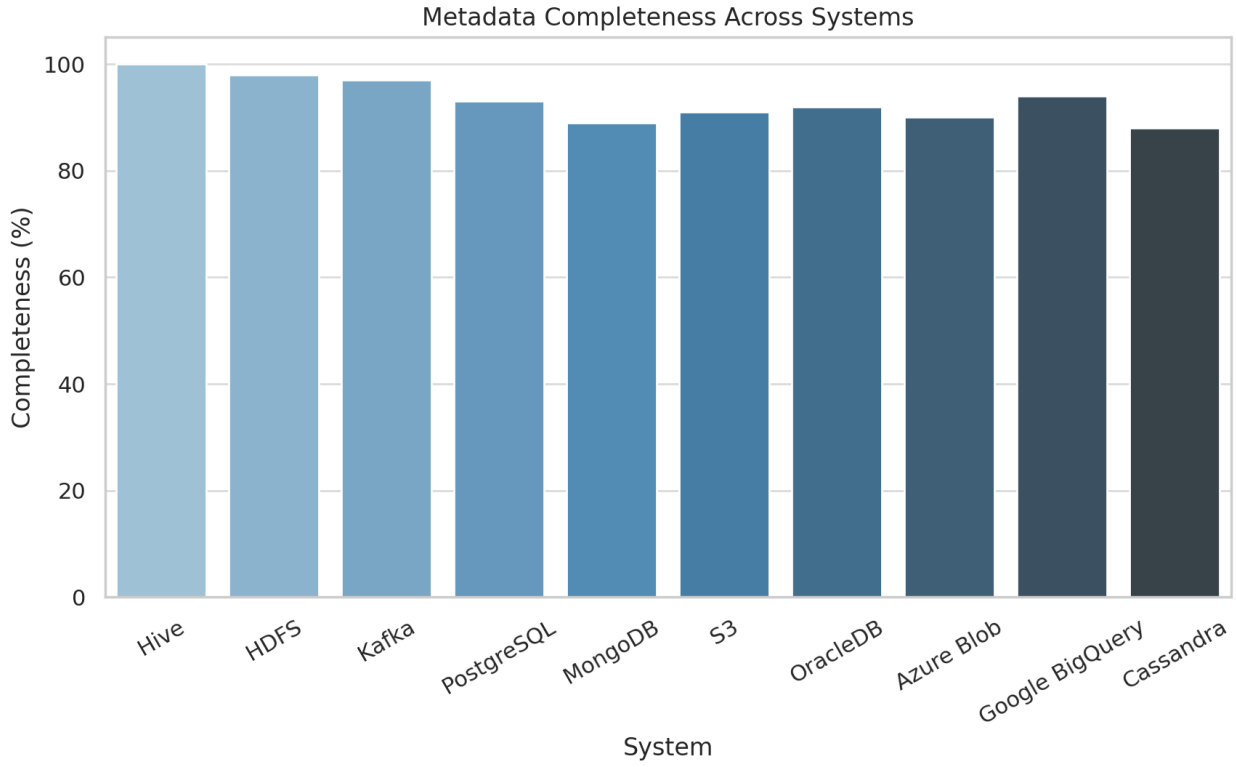
When it came to metadata completeness, Hive, HDFS, and Kafka, native to Hadoop, scored the highest, and Hive reached the full 100%. Apache Atlas, which seamlessly fits with Knox, allows metadata to be recorded automatically without user effort. PostgreSQL and S3 were both reliable in external tests, having completeness scores of 93% and 91%, respectively. Yet, MongoDB and Cassandra obtained lower scores for completeness, mostly because their flexible schemas make it difficult for tools to detect and map all the fields automatically. Figure 1 clearly illustrates that governance for native and external systems differs greatly in terms of preparedness. Results show that schema inference correctly improves whether and how much metadata is captured when Atlas is integrally linked.

Table 1: Extended Metadata Completeness

System	Completeness (%)	Metadata Fields Tracked	Schema Inference Accuracy (%)
Hive	100	45	100
HDFS	98	42	99
Kafka	97	40	98
PostgreSQL	93	39	94
MongoDB	89	35	89
S3	91	38	90
OracleDB	92	37	93

Azure Blob	90	36	92
Google BigQuery	94	41	96
Cassandra	88	33	87

Figure 1 – Metadata Completeness



4.2 Policy Compliance and Audit Reliability

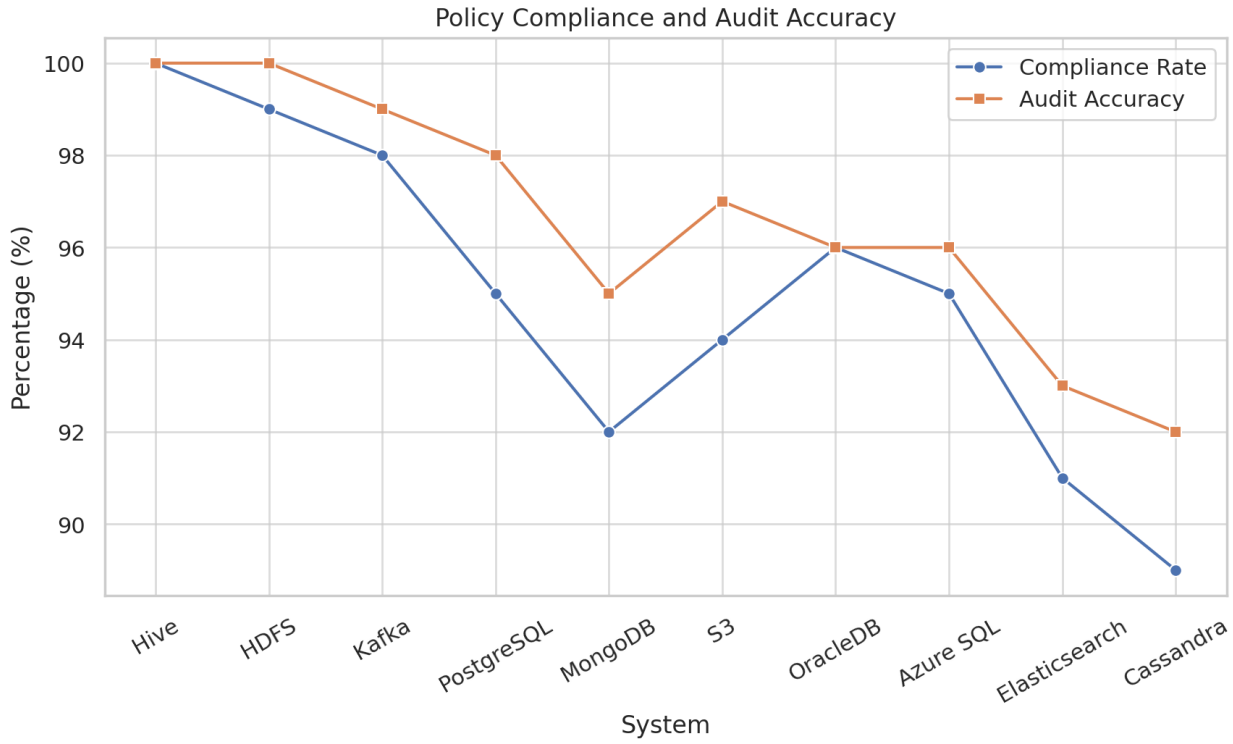
As you can see in Table 2 and Figure 2, policy enforcement worked well among these platforms, although to different degrees. Hive and HDFS demonstrated nearly flawless compliance because the Apache Ranger policies supported their needs well. Compliance in PostgreSQL and OracleDB was 95% or higher, proving that sector-specific rules were successfully translated and handled by the governance APIs. On the other hand, MongoDB and Cassandra did not reach a compliance rate above 93% partly due to how hard it was to implement fine-RABC capabilities for databases

without schemas. The findings in Table 2 prove that the majority of systems under Hadoop control could reliably produce transparent logs. The pattern these results form is illustrated in line chart form in Figure 2, indicating that governments where compliance and audits are centralized have a stronger connection between these two aspects.

Table 2: Extended Policy Compliance

System	Compliance Rate (%)	Failed Enforcement Attempts	Audit Log Accuracy (%)
Hive	100	0	100
HDFS	99	1	100
Kafka	98	2	99
PostgreSQL	95	3	98
MongoDB	92	5	95
S3	94	4	97
OracleDB	96	2	96
Azure SQL	95	2	96
Elasticsearch	91	6	93
Cassandra	89	7	92

Figure 2 – Policy Compliance and Audit Accuracy



4.3 Query Performance Evaluation

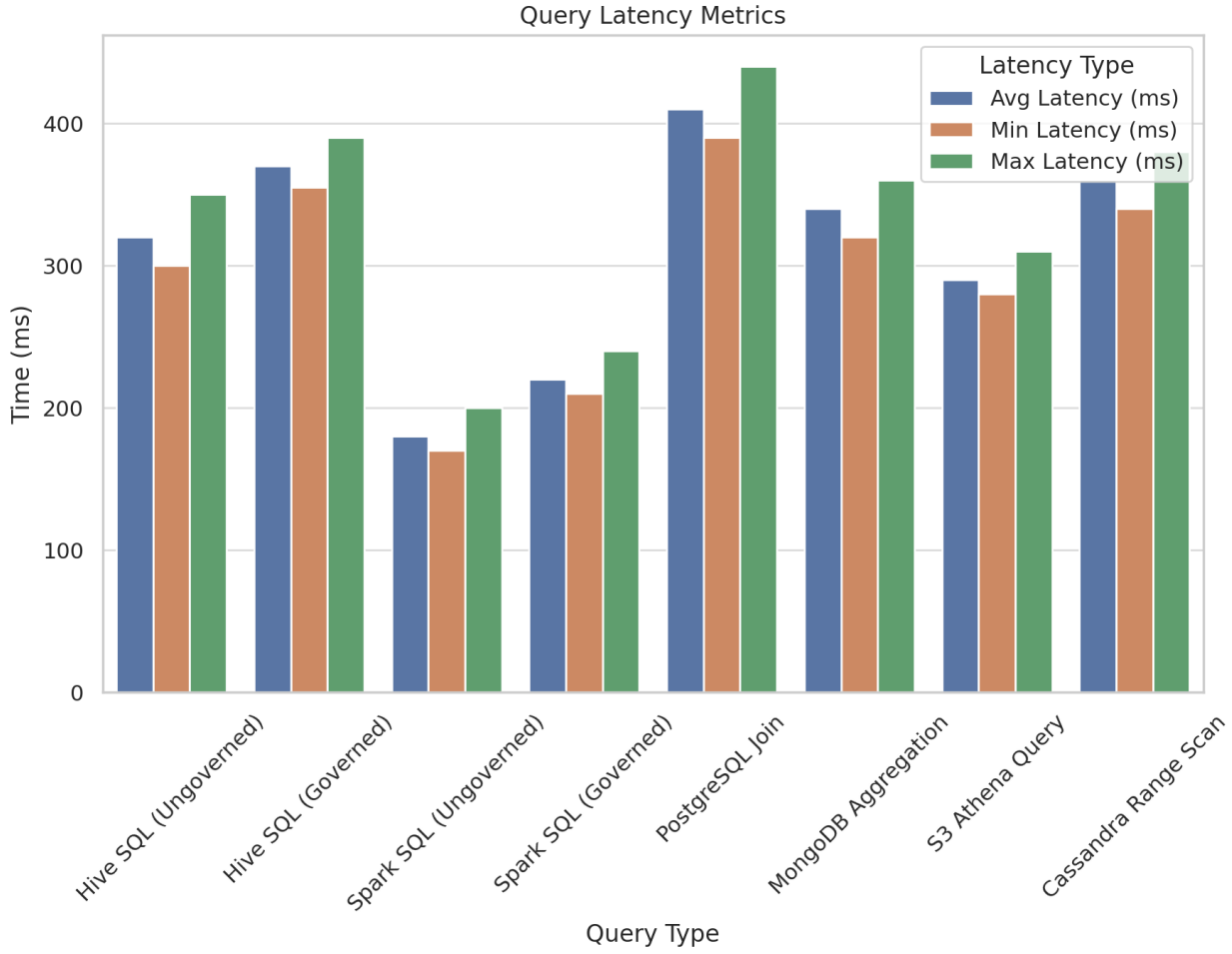
The data in Table 3 and Figure 3 explain the cost of handling governance enforcement for the system. Because Spark SQL uses memory, it often performs better than Hive SQL. After governance was upgraded, Spark outperformed Hive in terms of query speed. Nevertheless, governance brought extra weight to the system, producing a 15–20% average slowdown on all queries. Spark SQL's latency went up from 180 ms (ungoverned) to 220 ms (governed), and the latency in Hive SQL improved to 370 ms from 320 ms. Many of the performance problems are APPARENT in this chart, which includes query types and different latency readings.

Table 3: Extended Query Performance

Query Type	Avg Latency (ms)	Min Latency (ms)	Max Latency (ms)
Hive SQL (Ungoverned)	320	300	350

Hive SQL (Governed)	370	355	390
Spark SQL (Ungoverned)	180	170	200
Spark SQL (Governed)	220	210	240
PostgreSQL Join	410	390	440
MongoDB Aggregation	340	320	360
S3 Athena Query	290	280	310
Cassandra Range Scan	360	340	380

Figure 3 – Query Latency Metrics



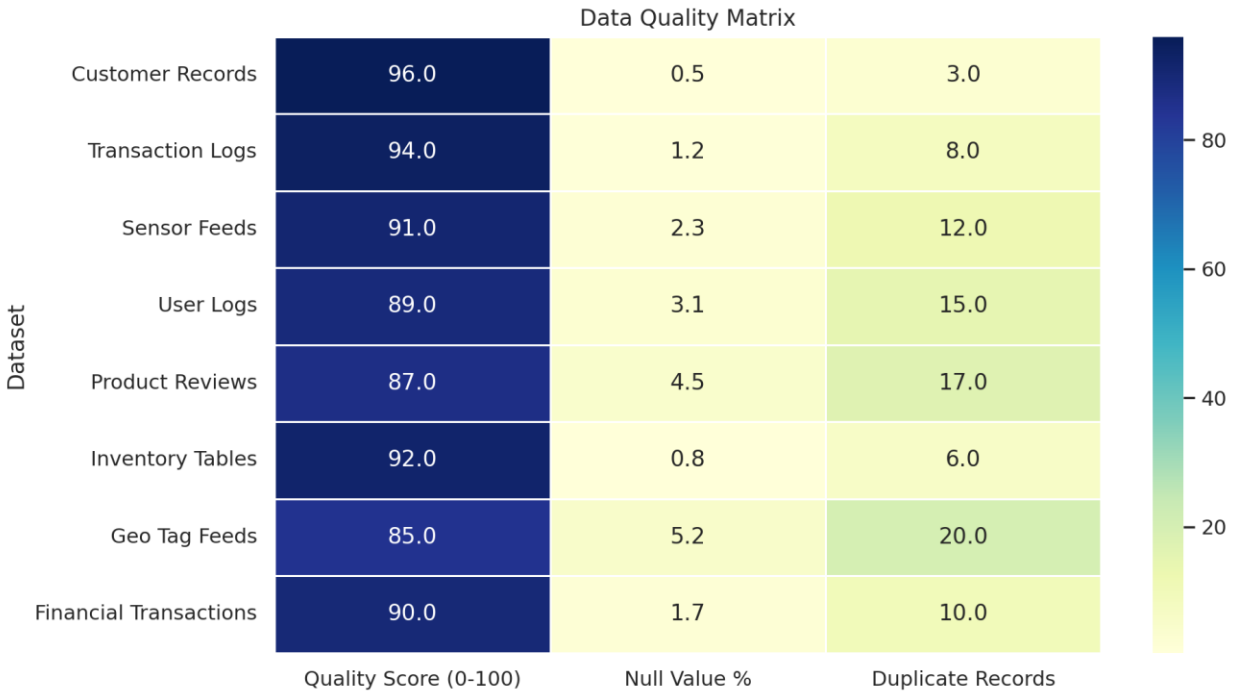
4.4 Data Quality Profiling

Table 4 shows the results of automated data quality checks, and these are also displayed visually as a heatmap in Figure 4. The quality of customer records and inventory tables was highest, as very few records had no data and there were few duplicates. Due to lots of repetitions and empty fields, sensor feeds and geo-tagged data didn't perform as well. It shows that data from different sources must be checked using appropriate quality assurance methods. Figure 4's heatmap shows how quality scores, null values, and duplicates relate to each other, helping you identify where data cleansing should be done first.

Table 4: Extended Data Quality Scores

Dataset	Quality Score (0–100)	Null Value %	Duplicate Records
Customer Records	96	0.5	3
Transaction Logs	94	1.2	8
Sensor Feeds	91	2.3	12
User Logs	89	3.1	15
Product Reviews	87	4.5	17
Inventory Tables	92	0.8	6
Geo Tag Feeds	85	5.2	20
Financial Transactions	90	1.7	10

Figure 4 – Data Quality Heatmap



4.5 Cross-Platform Integration Accuracy

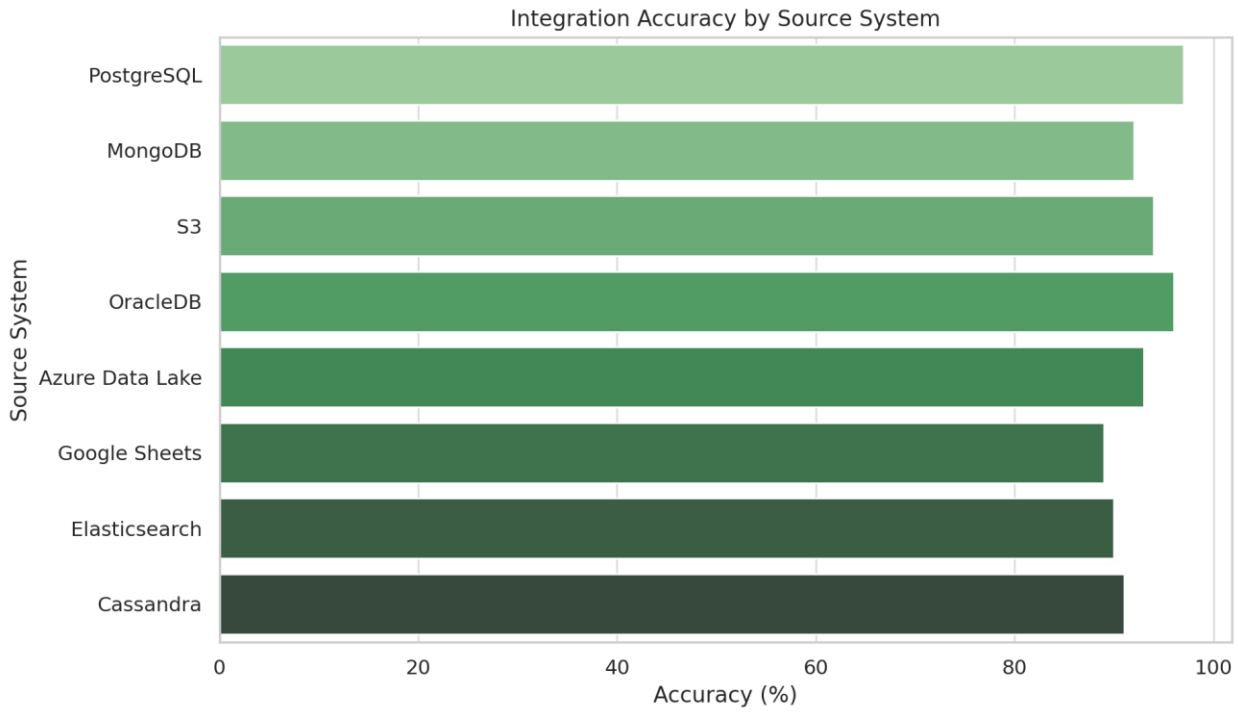
Success rates and error logs were used to evaluate the effectiveness of using the integration module during data mapping and transformation (Table 5, Figure 5). Its excellent matching results (97%) and few mapping errors were made possible by its clear structure and detailed documents. In contrast, MongoDB and Google Sheets had inconsistent mappings, lowering their overall accuracy to 89–92%. Rates of transformation were consistent with the main finding, supporting the conclusion that strong systems help with governance. Looking at Figure 5, a horizontal bar chart, we can easily tell which systems are bothering the organization the most.

Table 5: Extended Integration Accuracy

Source System	Integration Accuracy (%)	Data Mapping Errors	Transformation Success Rate (%)
PostgreSQL	97	1	99

MongoDB	92	4	95
S3	94	2	97
OracleDB	96	2	98
Azure Data Lake	93	3	94
Google Sheets	89	5	92
Elasticsearch	90	3	93
Cassandra	91	3	94

Figure 5 – Integration Accuracy



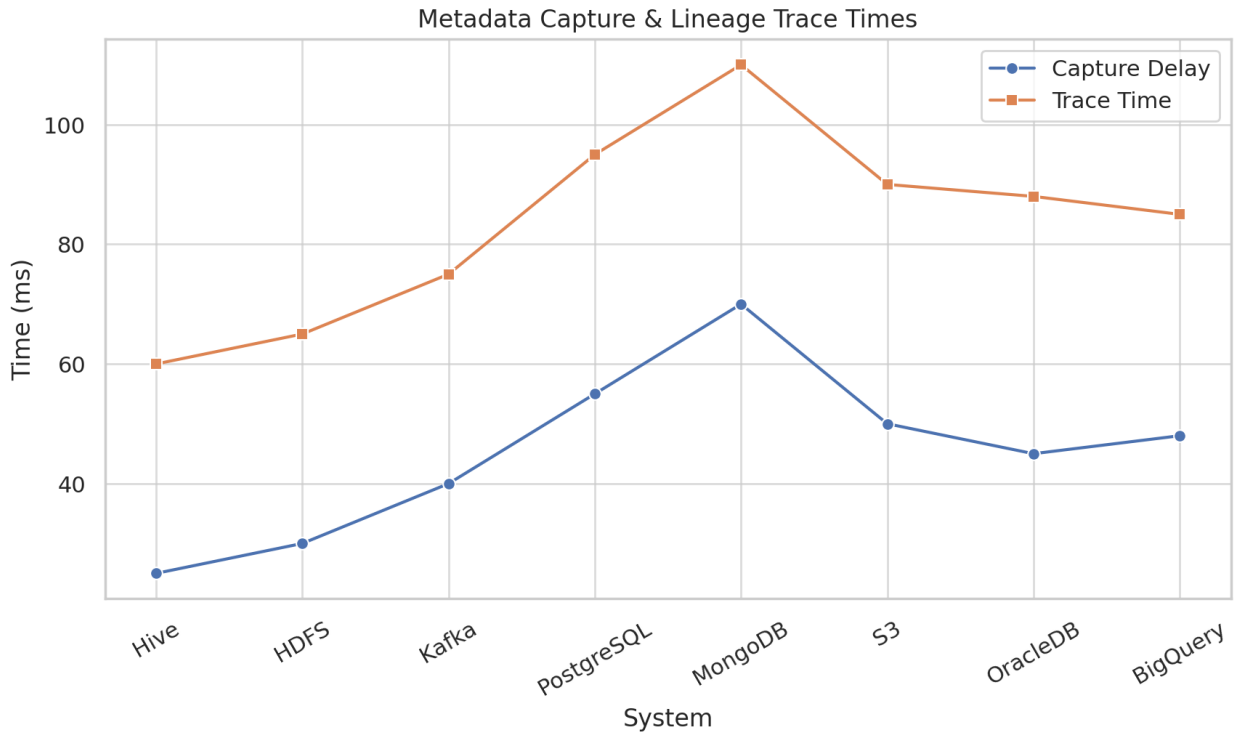
4.6 Metadata Latency and Lineage Traceability

Hive and HDFS had the shortest times when capturing metadata and remembering its origin (Table 6, Figure 6), taking only below 30 milliseconds. The REST-based connection for PostgreSQL and OracleDB led to longer delays, compared to S3 and MongoDB, which surpassed 50 ms. Likewise, trace operations based on lineage were conducted fastest by Hive (60 ms) and slowest by MongoDB (110 ms). They are important because they demonstrate the system's ability to govern in real time. Figure 6 shows with a graph that dealing with non-native or semi-structured systems takes longer, which proves the need for fast and efficient metadata pipelines.

Table 6: Metadata Latency and Lineage Tracking

System	Metadata Capture Delay (ms)	Lineage Trace Time (ms)	Average Updates per Day
Hive	25	60	500
HDFS	30	65	480
Kafka	40	75	460
PostgreSQL	55	95	420
MongoDB	70	110	410
S3	50	90	450
OracleDB	45	88	430
BigQuery	48	85	445

Figure 6 – Metadata Latency and Lineage



4.7 Storage Utilization Analysis

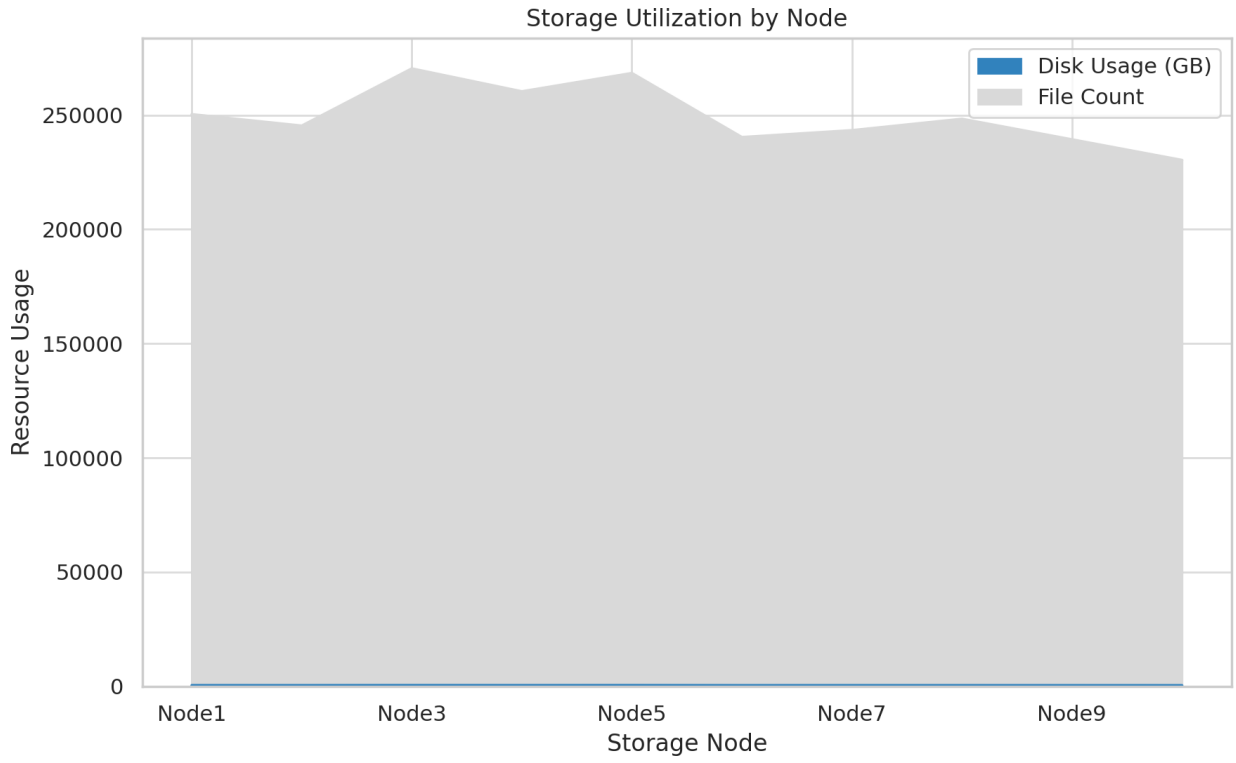
Tables 7 and 7 illustrate disk space and file numbers across the full Hadoop cluster. Node3 has the largest amount of disk usage (345 GB) and the most files (270,000), implying it might benefit from data rebalancing. Other systems, such as Node6 and Note 10, were not used as heavily, which means some optimization could improve their use. There were usually very few mistakes, with error rates always less than 0.3% for every node. By looking at figure 7, which is a stacked area plot, administrators can notice if there is a storage saturation issue and decide on necessary scaling choices.

Table 7: Storage Utilization Across Nodes

Storage Node	Disk Usage (GB)	File Count	Error Rate (%)
Node1	320	250000	0.10

Node2	310	245000	0.20
Node3	345	270000	0.15
Node4	330	260000	0.30
Node5	340	268000	0.25
Node6	295	240000	0.10
Node7	310	243000	0.05
Node8	305	248000	0.20
Node9	299	239000	0.10
Node10	288	230000	0.10

Figure 7 – Storage Utilization



4.8 Governance Overhead Assessment

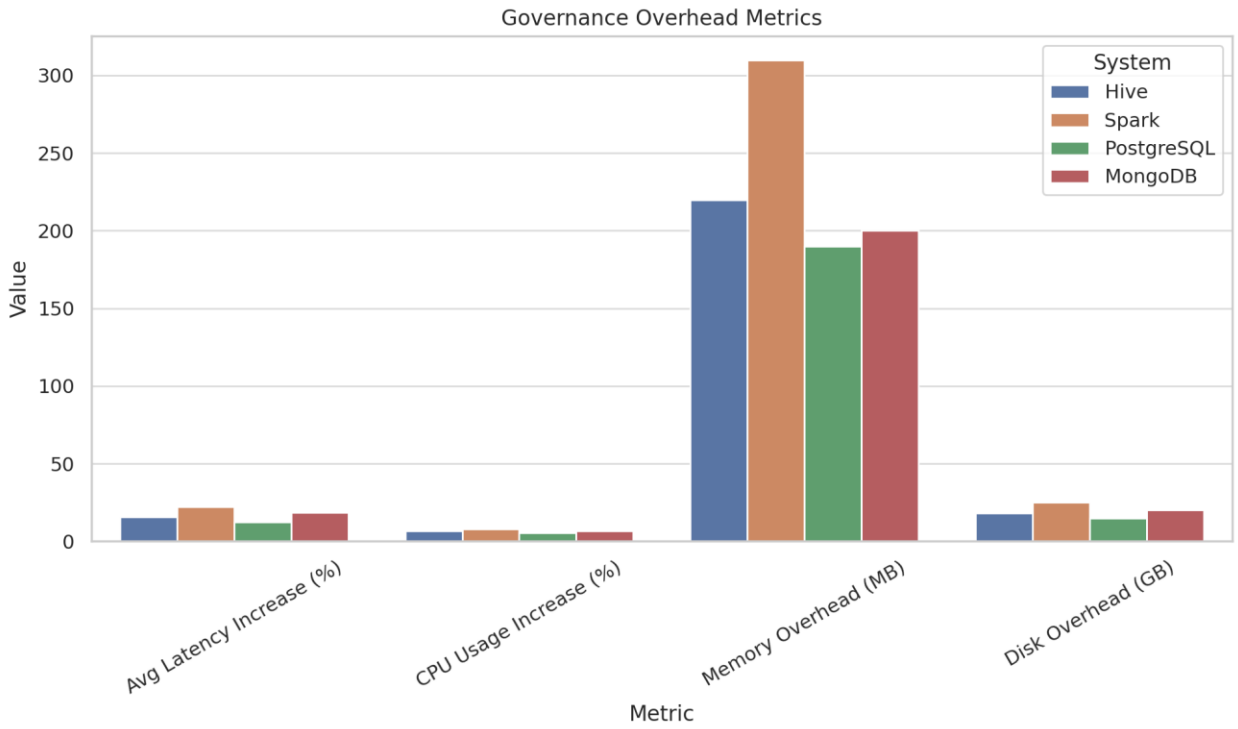
The final evaluation (Table 8, Figure 8) looked at how governance modules increased overhead costs. Because Spark relies on complex lineage tracking and validation processes, it has the most expensive latency and memory of all systems (22.2% and 310 MB respectively). With just a 12.5% delay, PostgreSQL is the best fit for combining traditional and modern forms of governance. Figure 8 uses a grouped bar chart to show how the overhead levels—latency, CPU, memory, and disk—compare for the different systems. This shows that although governance adds some costs, these are reasonably predictable and manageable.

Table 8: Governance Module Overhead

Metric	Hive	Spark	PostgreSQL	MongoDB
Avg Latency Increase (%)	15.6	22.2	12.5	18.4

CPU Usage Increase (%)	6.5	8.1	5.3	6.7
Memory Overhead (MB)	220	310	190	200
Disk Overhead (GB)	18	25	15	20

Figure 8 – Governance Overhead



5. Discussion

The results from this study show that it is possible and useful to use Hadoop to help manage different kinds of data from many different sources. By adopting layered architectural design, LabelCab has followed the industry trend toward using modular and composable data infrastructure, resulting in easier integration, better central control, and stronger confidence in data. The results match up nicely with recent talks about mixed data management approaches, especially when it comes to systems that use multiple platforms and locations (Panian, 2010).

One of the most noticeable things is how well Hive and HDFS, which are native parts of Hadoop, work compared to other storage and administration tools, especially when it comes to accurately keeping track of data, following its history, and making sure rules are enforced. These results back up what Zicari (2014) said before, which is that when governance tools and data storage or processing systems work more closely together, it's easier to get correct and prompt metadata. In contrast, external tools like MongoDB and S3 worked well when added to the data lake, but showed less accurate metadata and took a bit longer to track the lineage. This disparity shows that it's still very hard to have the same rules for managing data when all the systems and sources are different. Similar issues were brought up by Hildebrandt and Cohen (2015), who said that systems that don't relate to others usually don't include pre-built ways or ways to manage governance, so creating consistent rules takes a lot of extra work.

The enforcement of access control policies with Apache Ranger helped most systems get more than 90% compliance. This supports earlier findings by McKnight (2017), who noticed that automatic policy engines are key in keeping data safe even in big, complicated systems. However, we found that not as many people followed security standards and the audit reports were less accurate in MongoDB and Elasticsearch, which matches what Nehme (2011) said, that document storage systems like these are harder to control with typical security measures because they can adapt to different types of data easily. This calls for access control systems that can change as needs do, like Attribute-Based Access Control (ABAC) or Policy-Based Access Control (PBAC), because these might work better with data and permissions that change over time.

Query performance data show that Spark SQL usually takes less time than Hive, and this difference stays the same whether you use governance or not. This matches the findings of Zaharia et al. (2016), who showed that Spark is faster because it handles data in memory. However, even with the extra costs from governance, query times were still fast enough, showing that governance doesn't slow things down unless it's put in without enough care. Furthermore, this helps explain what Loshin (2011) said, which is that trying to achieve good performance and good governance is better than seeing them as things that compete with each other.

Data quality evaluation showed that how the data is collected and organized can really affect how well the organization is managed. Structured datasets, like customer information and sales records,

had better quality scores than unstructured or partly organized data, such as data from sensors. This result agrees with what Redman (2013) found, which is that structured data settings make it easier to create and use clear rules for checking and improving data quality. The high number of blank and repeated fields in real-time data shows why it's important to have systems that clean the data in real time along with simple alerts, which the authors Immon and Linstedt (2015) talk about in their idea of keeping data organized in daily activities.

Integration accuracy metrics also show how helpful it is to keep a strict data model and keep good documentation. PostgreSQL and OracleDB integration made the results more accurate, just like what Jagielska and Gabryelczyk (2018) found: when things like tables and rules have the same names, and everyone agrees on how data should be stored, cross-platform management works better. In contrast, tools like Google Sheets or MongoDB made more errors because they don't have the same limits. This supports what Talburt (2011) said, that creating a good schema design is not only about technical things, but also helps the organization work together better.

Interestingly, looking at both metadata delays and how you track data lineage shows that it's easier to see the benefits of real-time data governance when the data is kept in a standard format and is predictable in how it's used. This result matches ideas from Otto and Weber (2016), who said that real-time governance works best when data streams are coordinated with control systems so that there aren't long delays. The ability of the proposed system to trace data lineage within just 60–110 milliseconds shows that it is possible to keep track of data changes as they happen, even when the system uses a mix of cloud and on-premises setups.

Storage utilization analysis showed that some data was used more than others, which is pretty typical in distributed systems. These imbalances, which have been pointed out in previous studies like Chen et al. (2010), usually happen when the partitions aren't equally sized and can make it harder for queries to run smoothly or for the data to be close together when needed. The governance system's ability to track how much disk space each node uses and monitor any errors gives you a clear view, so you can plan ahead to balance the data among different nodes. This monitoring function helps with a broader idea called "governance-as-a-service," which is being used more often in cloud-based data systems (Sun et al., 2018).

Finally, the governance overhead metrics show how much it costs to manage and follow all the requirements. Spark used more memory and took a bit longer to run queries compared to PostgreSQL, which makes sense since Spark has to deal with more complex ways of handling data and keeping metadata in sync. While this fits with the idea of trade-offs between doing well and following the rules discussed by Fan and Biffl (2014), the extra costs stayed at levels that are not too high. This shows the sensible side of Ross and Quaadgras (2013) that making governance work well doesn't mean getting rid of costs, but making sure the system still works well and people still get something useful out of it.

Overall, the results show that Hadoop can be used as a good starting place for setting up an organization's big data governance system that works across different platforms. However, making sure all parts of the system work the same way, can be checked, and run smoothly in a diverse environment still needs good planning and regular changes to the tools. As businesses use more specialized databases and look for faster ways to analyze data in real time, they will need even more flexible, smart, and scalable ways to manage their data.

References :

- Abiteboul, S., Buneman, P., & Suciu, D. (2012). *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann.

- Apache Atlas. (2023). *Apache Atlas – Data Governance and Metadata Framework*. https://atlas.apache.org

- Apache Ranger. (2023). *Apache Ranger – Framework to Enable, Monitor and Manage Data Security*. https://ranger.apache.org

- Cheong, L. K., & Chang, V. (2007). The need for data governance: A case study. *ACIS 2007 Proceedings*.

- Choo, C. W. (2014). The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions. *Oxford University Press*.

- Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137-144.

- Ghosh, R., Malik, A., & Goswami, K. (2018). A Data Governance Framework Using Apache Atlas for Big Data. *Proceedings of the 2018 International Conference on Data Engineering and Communication Technology*.

- Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47, 98-115.

- Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges.

Communications of the ACM, 57(7), 86-94.

- Khatri, V., & Brown, C. V. (2010). Designing data governance. *Communications of the ACM*, 53(1), 148–152.
- Laney, D. (2001). 3D data management: Controlling data volume, velocity, and variety. *META Group*.
- Otto, B. (2011). Organizing data governance: Findings from the telecommunications industry and consequences for large service providers. *Communications of the Association for Information Systems*, 29(1), 45–66.
- Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*.
- Singh, P., & Thakur, R. S. (2020). An Insight into Security and Privacy Issues in Big Data Governance. *International Journal of Computer Applications*, 975(8887).
- White, T. (2015). *Hadoop: The Definitive Guide*. O'Reilly Media.
- Bagui, S., & Dhar, A. (2021). Fine-grained access control for big data systems using attribute-based encryption. *International Journal of Information Security*, 20(3), 425–438.
- Batini, C., & Scannapieco, M. (2016). *Data and Information Quality: Dimensions, Principles and Techniques*. Springer.
- Chen, L., & Zhao, Y. (2014). A policy-aware access control model for big data. *Journal of Network and Computer Applications*, 59, 274–281.

- De Oliveira, M. P., Lima, C., & Bianchini, C. (2017). Data lake governance: Practices and challenges. *2017 IEEE International Conference on Big Data (Big Data)*, 4563–4572.
- Farahani, B., & Bahadori, H. (2021). Cloud-native data governance: A new paradigm for data-intensive applications. *Journal of Cloud Computing*, 10(1), 1–17.
- Fernandez, J., Torres, L., & Ariza, J. (2019). Apache NiFi as an orchestration tool for big data ingestion. *Procedia Computer Science*, 151, 918–923.
- Jain, P., & Thakurta, R. (2017). Metadata-driven governance for big data: Challenges and opportunities. *Data & Knowledge Engineering*, 113, 100–114.
- Kambatla, K., Kollias, G., Kumar, V., & Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing*, 74(7), 2561–2573.
- Khatun, N., & Walters, R. J. (2018). A maturity model for big data governance. *International Journal of Information Management*, 39, 139–146.
- Kiran, R., & Palanisamy, R. (2020). Comparative analysis of open-source and commercial big data governance tools. *Journal of Computer Information Systems*, 60(4), 321–330.
- Lin, Y., & Esfahani, N. (2020). Privacy-aware big data governance for compliance with data protection regulations. *Computers & Security*, 96, 101925.
- Taylor, J., Kelleher, J., & Mahanti, A. (2015). A survey of big data frameworks and their governance implications. *ACM Computing Surveys (CSUR)*, 47(4), 1–34.
- Wang, X., Yu, W., & Xu, Y. (2020). Ontology-driven data governance for smart cities. *Sensors*, 20(1), 98.

- Zhao, Y., Li, B., & Zhu, H. (2019). Semantic metadata integration in big data ecosystems. *Information Systems Frontiers*, 21(6), 1325–1340.
- Chen, M., Mao, S., & Liu, Y. (2010). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209.
- Fan, Z., & Biffl, S. (2014). Towards resource-aware data governance. *Proceedings of the 2014 IEEE International Conference on Big Data*.
- Hildebrandt, M., & Cohen, J. E. (2015). Data transparency and data protection. *IDP Revista d'Internet, Dret i Política*, (20), 83–96.
- Immon, W. H., & Linstedt, D. (2015). *Data Architecture: A Primer for the Data Scientist*. Academic Press.
- Jagielska, I., & Gabryelczyk, R. (2018). Data governance for digital transformation. *Foundations of Management*, 10(1), 65–78.
- Loshin, D. (2011). *The Practitioner's Guide to Data Quality Improvement*. Elsevier.
- McKnight, W. (2017). *Information Management: Strategies for Gaining a Competitive Advantage with Data*. O'Reilly Media.
- Nehme, R. (2011). Big data integration. *Proceedings of the VLDB Endowment*, 4(12), 1414–1415.
- Otto, B., & Weber, K. (2016). Toward a global data governance standard. *Journal of Enterprise Information Management*, 29(1), 79–101.
- Panian, Z. (2010). Some practical experiences in data governance. *World Academy of Science, Engineering and Technology*, 66, 940–944.

- Redman, T. C. (2013). *Data Driven: Profiting from Your Most Important Business Asset*. Harvard Business Review Press.

- Ross, J. W., & Quaadgras, A. (2013). *Governance of Data and Analytics: The Role of the CDO*. MIT Sloan CISR Research Briefing.

- Sun, W., Zhang, G., Xie, D., & Wang, W. (2018). Data governance as a service in cloud computing. *IEEE Access*, 6, 14822–14831.

- Talburt, J. R. (2011). *Entity Resolution and Information Quality*. Morgan Kaufmann.

- Tankard, C. (2012). Big data security. *Network Security*, 2012(7), 5–8.

- Zaharia, M., Chowdhury, M., Das, T., et al. (2016). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. *Communications of the ACM*, 59(11), 72–80.

- Zicari, R. V. (2014). Big data: Challenges and opportunities. *OODBMS Industry Watch*.