EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Real-Time Monitoring of Data Pipelines: Exploring and Experimentally Proving that the Continuous Monitoring in Data Pipelines Reduces Cost and Elevates Quality

Shammy Narayanan[1,*], Maheswari S[2] and Prisha Zephan[3]

[1]Thryve Digital LLP, Chennai, India
[2]Vellore Institute of Technology, Chennai, India
[3]Sathyabama Institute of Technology, Chennai, India

Abstract

Data pipelines are crucial for processing and transforming data in various domains, including finance, healthcare, and e-commerce. Ensuring the reliability and accuracy of data pipelines is of utmost importance to maintain data integrity and make informed business decisions. In this paper, we explore the significance of continuous monitoring in data pipelines and its contribution to data observability. This work discusses the challenges associated with monitoring data pipelines in real-time, propose a framework for real-time monitoring, and highlight its benefits in enhancing data observability. The findings of this work emphasize the need for organizations to adopt continuous monitoring practices to ensure data quality, detect anomalies, and improve overall system performance.

Keywords: Data pipelines, monitoring, real-time, data observability, data quality, anomaly detection

Received on 11 November 2023, accepted on 28 January 2024, published on 07 February 2024

doi: 10.4108/eetsis.5065

1. Introduction

Over the past two decades, the field of IT systems and processes has witnessed significant evolution, transitioning from Mainframe to Mobile computing with a primary focus on "Application Programming." During this time, specialized and customized techniques such as structured process guidance, checklists, metrics, and alert monitoring have played a crucial role in application monitoring and mitigating downtime. However, it's essential to recognize that these tools and methods cannot be directly applied to Data Applications, as they require a distinct set of vocabulary and guardrails.

In traditional application monitoring, downtime refers to the period when the application is unavailable for users and for downstream and upstream processing. However, this definition doesn't seamlessly translate to the realm of Data applications. In the context of data pipelines, the term "Data Downtime" encompasses more than just pipeline unavailability; it can also encompass issues such as corrupted or redundant data, which hinder the effectiveness of Analytics engines and impede management decision-making processes.

In light of this shifting landscape of nomenclature and processes, it becomes imperative to reevaluate the means by which Data pipelines are monitored in real-time and timely alerts are provided to the appropriate stakeholders [1]. This proactive approach not only enables stakeholders to make informed decisions promptly but also facilitates substantial cost savings and enhances the end-user experience.

*Corresponding author. Email: Shammy45@gmail.com

1.1 Data Pipelines

Data Pipelines are critical components in modern data-driven systems, enabling the efficient processing, integration, and transformation of data from diverse sources. They play a pivotal role in various domains, such as financial analytics, healthcare systems, and recommendation engines [2]. As data pipelines handle large volumes of data with complex transformations, ensuring their reliability and accuracy becomes paramount. Even minor issues in data pipelines can have severe consequences, leading to inaccurate analytics, incorrect decisions, and compromised data integrity.

Monitoring data pipelines is essential to maintain their performance, detect anomalies, and identify potential issues promptly. Traditional batch-based monitoring approaches often fall short in addressing the dynamic nature of data pipelines and the need for real-time insights [3]. Real-time monitoring provides continuous visibility into the operational health of data pipelines, facilitating proactive detection of issues and enabling timely corrective actions. In this paper, we explore the importance of continuous monitoring in data pipelines and its contribution to data observability. Data observability refers to the ability to understand, debug, and monitor the behaviour of data in systems. It encompasses various aspects, including data quality, integrity, lineage, and anomaly detection. We highlight the challenges associated with real-time monitoring, propose a framework for real-time monitoring of data pipelines, and discuss its benefits in enhancing data observability.

Distinguishing between traditional application downtime and Data Downtime is crucial for understanding the unique challenges involved in monitoring Data pipelines. While traditional application downtime refers to periods of unavailability, Data Downtime encompasses a broader range of issues, including corrupted data and redundant data. These challenges directly impact the effectiveness of Analytics engines and hinder the decision-making process for management.

In this changing landscape, it is essential to reevaluate the approaches to monitoring Data pipelines in real time and alerting the relevant stakeholders. By doing so, we can ensure that stakeholders receive the right information at the right time, enabling them to make timely decisions. Additionally, adopting an effective monitoring and alerting system can result in significant cost savings and an enhanced end-user experience [4].

To address the unique challenges posed by Data Applications, a dedicated monitoring framework must be established for Data pipelines. This framework should go beyond traditional application monitoring and focus on detecting and alerting not only pipeline unavailability but also issues such as corrupted or redundant data [5].

Real-time monitoring plays a critical role in this framework, allowing stakeholders to promptly identify and address Data Downtime events. By leveraging advanced monitoring tools and techniques, such as anomaly detection algorithms and data quality checks, organizations can ensure that potential issues are detected in real time. Furthermore, providing the right quantum of information to the right stakeholders facilitates informed decision making and enables swift action.

The remainder of this paper is organized as follows: Section II discusses the challenges in monitoring data pipelines in real-time. Section III presents a framework for real-time monitoring. Section IV explains how real-time monitoring contributes to data observability. Section V discusses the benefits of real-time monitoring. Finally, Section VI concludes the paper and provides directions for future research.

2. Literature Survey

Monitoring data pipelines in real-time is a challenging task due to the distributed and dynamic nature of these pipelines. This literature review examines the key challenges associated with real-time monitoring of data pipelines, drawing insights from existing research and industry practices [6].

One of the primary challenges in real-time monitoring is handling high data volumes and rapid data velocity. Scalable and efficient monitoring systems are necessary to process and analyze data streams without introducing significant latency. Complex data transformations in data pipelines pose another challenge. These transformations involve filtering, aggregations, and join operations, requiring the monitoring system to capture intermediate states, validate data transformations, and ensure data integrity throughout the pipeline.

Data quality assurance is a perpetual challenge in the data environment, and it becomes more critical in real-time monitoring. Detecting anomalies, missing values, and outliers is essential, but maintaining data integrity and accuracy within the domain context is equally important. Data pipelines operating in distributed environments are susceptible to failures and disruptions. Fault tolerance and resilience are crucial in real-time monitoring to promptly detect failures and recover to minimize data loss and downtime [7].

Scalability and resource management are key challenges in real-time monitoring. The monitoring system needs to efficiently allocate computing resources to handle high data volumes and velocity, preventing bottlenecks and ensuring smooth operation. Timeliness and latency are critical factors in real-time monitoring. Delayed problem identification due to latency can lead to data inconsistencies or missed alerts. Minimizing latency is crucial to ensure timely monitoring and decision-making based on the data being processed.

Data security and privacy are significant concerns in real-time monitoring. Handling sensitive data requires robust security measures, access controls, and encryption techniques to protect against unauthorized access or breaches. Effective visualization and actionable insights are essential for real-time monitoring. Presenting data in intuitive dashboards and providing real-time insights

enable operators to quickly identify issues and make informed decisions.

The complexity of the monitoring infrastructure is another challenge. Real-time monitoring involves various components and technologies, requiring coordination and integration to effectively manage and monitor the infrastructure. In conclusion, real-time monitoring of data pipelines presents challenges in handling high data volume and velocity, managing complex data transformations, ensuring data quality, maintaining fault tolerance and resilience, scaling resources, minimizing latency, ensuring data security and privacy, providing visualization and actionable insights, and managing the complexity of the monitoring infrastructure. Understanding and addressing these challenges are crucial for the development of robust monitoring systems and effective management of data pipelines in real-time environments.

The authors built a framework for data pipelines for manufacturing systems. This framework employs use case to understand the guidelines to use the selective layers and components in big data pipeline [8]. The authors presented a study of data science pipelines in theory, in the small and in the large. They used different data sets for analysis from Kaggle. It was discovered that the data science pipelines in different environments vary considerably. Notably, the data science pipelines have an increasingly linear design and lack several steps. Compared to theoretical representations, data science pipelines generally have a complex structure and feedback loops. Additionally, it provides three data science pipeline representations that, in theory, in the small and the large, encapsulate the essence of the subjects [9].

The authors presented a comprehensive overview of the challenges and opportunities associated with managing observability data in large-scale systems. The authors highlight the increasing importance of observability, which refers to the ability to monitor, understand, and debug complex distributed systems, and its role in ensuring system reliability and performance. They discuss the key dimensions of observability, including metrics, logs, and traces, and the need for scalable and efficient data management solutions to handle the massive volumes of data generated by modern systems. The review highlights various techniques and approaches proposed in the literature to address observability data management challenges, such as log compression, sampling, and distributed storage systems. Additionally, the authors discuss the role of stream processing and real-time analytics in enabling timely insights and actionable intelligence from observability data. The review concludes by identifying open research directions and emphasizing the need for holistic approaches that integrate data management, analytics, and visualization to achieve comprehensive observability in large-scale systems. Overall, it provides a valuable survey of the current state-of-the-art in observability data management and sets the stage for future research in this important area [10].

Big data pipeline discovery through process mining is discussed in [11]. Pipeline discovery through process mining is the application of process mining techniques to analyze event logs and uncover the sequence of activities, dependencies, and bottlenecks that form a pipeline within a business process. It provides organizations with visualizations and insights to improve process efficiency, compliance, and decision-making, enabling them to optimize performance and identify automation opportunities.

The authors explored the advent of the computing continuum has opened up new avenues for effectively managing big data pipelines that involve diverse and potentially untrustworthy resources. Examined the lifecycle of big data pipelines within the computing continuum, discussed the challenges associated with them, and presented a research agenda for future exploration in this field [12]. The authors presented process mining as a field that helps organizations understand how their business processes are carried out by analyzing event logs. However, a common challenge is collecting the necessary data for analysis, which requires technical and domain expertise. It proposed a user-friendly method to generate simulated event logs, which can be used to discover the structure of data pipelines in business processes. The approach was tested in a digital marketing scenario and found promising results. This method can be helpful for organizations to improve their processes and make more informed decisions [13].

The authors proposed data pipeline selection and optimization and applied the sequential Model-Based optimization technique to data pipeline selection and configuration. Used metric to study if an optimal configuration is algorithmically specific or rather universal. To avoid the costs involved in processing data, they suggested approaches to explore in data preprocessing and other stages too [14]. The authors investigated LinkedIn's real-time activity data pipeline. It described the implementation of a data pipeline at LinkedIn, focusing on the shift from a batch-oriented file aggregation mechanism to a real-time publish-subscribe system called Kafka. The data pipeline utilizes activity data in the form of log or event messages that capture user and server activity. These messages are vital for various internet systems, including advertising, relevance, search, recommendation systems, security, and analytics [15].

The transition to a real-time pipeline introduces new design challenges due to the high volume of activity data and the need for real-time processing. The pipeline discussed in the paper is currently operational at LinkedIn, handling over 10 billion message writes per day and delivering more than 55 billion messages to consumer processing systems daily. The paper explores the evolution of the pipeline, the obstacles faced during the transition to real-time

processing, and the design and engineering problems encountered throughout the process.

3. Proposed Methodology

In this section, the proposed methodology addresses the challenges of real-time monitoring in data pipelines. It is designed to provide continuous visibility and insights into data pipelines, enabling prompt detection and mitigation of issues shown in **Figure 1**.

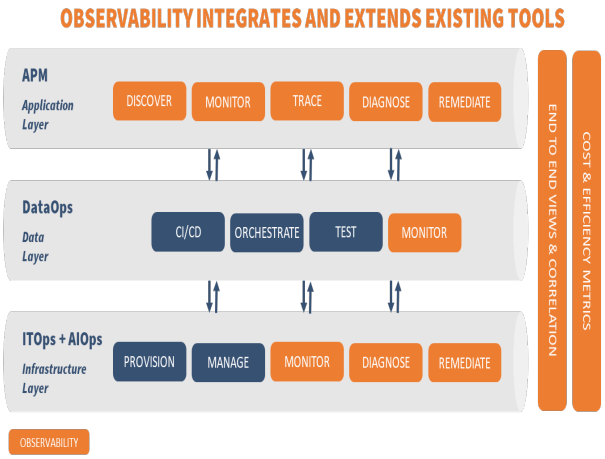

Figure 1. Model Architecture Integrating Application and Data Layer with Real-time Observability Tools

It consists of the following steps:

3.1 Requirement Analysis

Starting with a thorough analysis of the monitoring requirements specific to the data pipelines under consideration involves understanding the data sources, transformation processes, desired monitoring metrics, and key performance indicators. Gaining a comprehensive understanding of the system requirements allows for tailoring the monitoring approach accordingly.

3.2 Designing Monitoring Rules

Based on the requirement analysis, we design a set of monitoring rules that will govern the real-time monitoring process. These rules define the conditions for detecting anomalies, data quality issues, and pipeline failures. They can include threshold-based checks, statistical analyses, pattern recognition techniques, and business rule validations. The monitoring rules are designed to align with the specific objectives and domain knowledge of the data pipelines.

Various popular tools available in market for Data Observability

1- Monte Carlo Observability platform
2- Acceldata Data observability cloud
3- Amazon cloud watch
4- Data log observability platform
5- Dynatrace

Tool/Platform selection is based on Cloud provider, Compatibility/Interoperability with Dash boarding tools, out of shelf integration features, etc.
Out of this platform, Dynatrace is cloud agnostic and comes with the plug and play integration features that is compatible with all the three major cloud providers (AWS, Azure, GCP). Below architecture, in **Figure 2** indicates how agents of Dynatrace can collect application, data and monitoring logs from disparate sources and makes it available for centralized monitoring and KPI tracking. This architecture is so robust that fault tolerance is in-built into it to pace with any failovers.

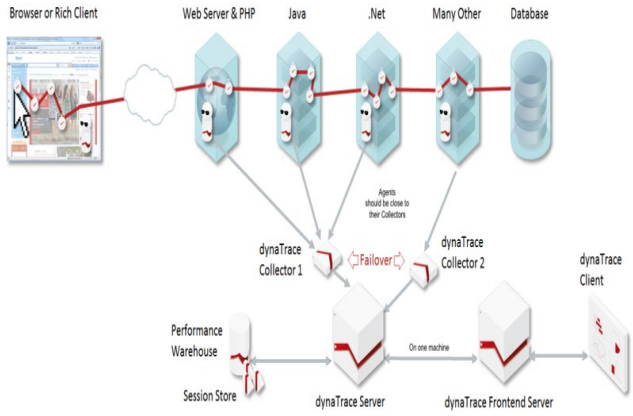

Figure 2. Dynatrace architecture collating and curating application and security logs into a centralized spool

3.3 Implementing Data Stream Ingestion

Next, the implementation of the data stream ingestion component of the methodology begins. This involves setting up mechanisms to capture, buffer, and validate incoming data streams from various sources. Scalable and efficient data ingestion frameworks are leveraged to handle different data formats, frequencies, and volumes. Data validation techniques are applied to ensure the reliability and completeness of the ingested data.

Below **Figure 3** depicts a Real-world example of Dynatrace implementation for collecting logs from Data transformation pipelines leveraging Jenkin and shared, Dynatrace API. The left side image, there is a redundancy of code calling Observability platform while the right-side image accomplishes the objective through a shared Jenkin library.

Figure 3. Samples of Distributed and Centralized Jenkins architecture interfacing with Dynatrace for Log compilation

3.4 Real-Time Processing and Analysis

Once the data streams are ingested, real-time processing and analysis are performed [16]. The monitoring rules designed in the previous step are applied to the ingested data streams. This involves continuously monitoring the data for anomalies, validating data transformations, and assessing data quality against predefined metrics. Advanced analytics techniques, such as machine learning algorithms and statistical analyses, are employed to enhance the monitoring process.

Figure 4. A local host implementation of the observability tool - Dynatrace in a simple web application

The above code in **Figure 4** shows a sample request for the CUSTOM_DEPLOYMENT event, using the Unix curl utility.

3.5 Alerting and Visualization

Detected anomalies, data quality issues, and pipeline failures trigger alerts and notifications. An alerting system is implemented to promptly notify relevant stakeholders when these issues occur. Alerts can be delivered through dashboards, emails, or instant messages, depending on the urgency and severity of the detected issues. Real-time visualization of monitoring metrics and key performance indicators enables stakeholders to quickly identify and understand the issues at hand.

3.6 Fault Detection and Recovery

The methodology incorporates fault detection mechanisms to identify failures and disruptions in data pipelines. Automated recovery processes are triggered to minimize data loss and downtime. Fault tolerance measures, such as replication and backup strategies, are employed to ensure the resilience of the monitoring system.

By following this proposed methodology, organizations can achieve comprehensive and effective real-time monitoring of their data pipelines. The methodology ensures continuous visibility into the system's health and performance, enabling prompt detection and mitigation of issues. It facilitates data observability by enhancing data quality, providing anomaly detection, enabling lineage tracking, and optimizing performance.

The benefits of this methodology include early issue detection and mitigation, improved data integrity and accuracy, enhanced operational efficiency, and increased trust and compliance with data governance regulations. In the next section, a case study demonstrates the application of the proposed methodology in a real-world scenario.

4. Experimental Study

In a controlled environment study, a comparison was made between data pipelines lacking observability and those with configured alerts to ensure real-time observability. The study identified the critical need for an Executive Dashboard within the organization. This dashboard serves a vital role in calculating budgeting variances, providing sales forecasts per strategic business units, and feeding data to the company's quarterly balance sheet. Data for this dashboard is sourced from various disparate sources, including sales information in the data warehouse, budgeting files from the finance department, forecast data from business units, and reference tables from cloud ledgers. Given the dashboard's significance in making crucial decisions, accuracy, timeliness, and reliability of the data are imperative.

To ensure real-time observability, alerts were configured at major critical paths within the data pipelines. These alerts encompass the following:

Alert for Freshness: This alert notifies when data is not received from various sources within the prescribed time. Timeliness is crucial to avoid incorrect decisions, so timely alerts about data freshness are essential.

Alert for Quality: Triggered when data falls below a specified quality threshold due to factors like missing values, null values, or incorrect data types. In such cases, the dataset is discarded, and a production alert is raised to address the data quality issue.

Alert for Volume: Triggered when the dataset deviates from a standard tolerance range. For instance, if historical

data volume for the dashboard typically varies between 2 to 3 GB, any deviation prompts an alert for manual intervention to review and approve/disapprove the data.
Alert for Schema: This critical alert tracks changes in the metadata of one or more source tables. Any schema modification triggers an alert, ensuring that the monitoring system is aware of data lineage and provenance.
Alert for Lineage: Schema changes have a significant impact on downstream jobs, schemas, and catalogues. Therefore, any schema alteration triggers an alert, initiating remediation of all affected components to accommodate changes in field(s).
Based on historical logs, it was observed that incidents related to Data Freshness occurred twice per month, Data Quality issues occurred eight times per month, Volume Variance incidents occurred once, Schema Changes occurred three times, and Lineage Tracking issues occurred four times over a span of two quarters.

To quantify the impact of data downtime, the formula in Equation 1 was employed:

Data Downtime = N * (Time to detect + Time to fix). (1)

where N represents the total number of incidents. In this scenario, a total of 18 incidents were observed. On average, it took 4 hours to detect an incident and 9 hours to fix it. Therefore, the total data downtime for the program is calculated as: 18 * (4 + 9) = 234 person-hours. Considering a billing rate of $100 per hour per data analyst, this results in a loss of $19,890 per month or approximately $238,680 per year (approximately INR 2 Crores/annum).

In addition to the financial loss, qualitative losses such as delayed, incorrect, or redundant decisions, damage to reputation, and slowness in reporting and compliance should be considered. While these qualitative losses cannot be precisely quantified, their impact on the business is significant, as illustrated in Figure 5

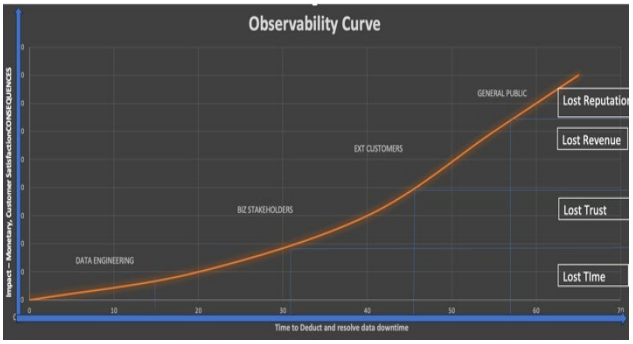

Figure 5. Qualitative graphical depiction of Business Losses in the absence of real-time monitoring tools

Through the implementation of real-time observability via configured alerts, organizations can minimize data downtime, reduce financial losses, and mitigate qualitative impacts, thereby ensuring the availability of accurate and timely information for critical decision-making processes.

5. Conclusion

Real-time monitoring played a crucial role in ensuring the reliability, accuracy, and observability of data pipelines. By addressing the challenges associated with monitoring data pipelines in real-time, organizations were able to gain continuous visibility into the behavior of their data, detect anomalies, and maintain data quality. Real-time monitoring contributed to data observability by enhancing data quality and integrity, facilitating anomaly detection and debugging, providing lineage and provenance information, and optimizing system performance. Embracing real-time monitoring practices brought several benefits, including early issue detection, improved data integrity, enhanced operational efficiency, and increased trust. As data pipelines continued to evolve and became more complex, continuous monitoring became imperative for organizations to make informed decisions based on high-quality data.

Acknowledgements.
The authors would like to thank the reviewers for their valuable feedback and suggestions.

References

[1] Dwyer, M, Hwang, J, Shires, A, Cohen J. Application of Comprehensive Data Analysis for Interactive, Hierarchical Views of HPC Workloads. IEEE International Conference on Big Data. 2018:3585-3589.
[2] Lachner, C, Laufer, J, Dustdar, S, Pohl, K. A Data Protection Focused Adaptation Engine for Distributed Video Analytics Pipelines. IEEE Access. 2022:10: 68669-68685.
[3] Hu, H, Wen, Y, Chua T. –S, Li, X. Toward Scalable Systems for Big Data Analytics. A Technology Tutorial. IEEE Access. 2014: 2: 652-687.
[4] Kulkarni, A. R, Kumar, N, Rao K. R. Efficacy of Bluetooth-Based Data Collection for Road Traffic Analysis and Visualization Using Big Data Analytics. Big Data Mining and Analytics. 2023: 6:139-153.
[5] Icilia, MÁ, García – Barriocanal, E, Sánchez – Alonso, S, Mora – Cantallops, M, Cuadrado, JJ. Ontologies for Data Science On Its Application to Data Pipelines. Metadata and Semantic Research. Communications in Computer and Information Science. 2018; 846: 1-8
[6] Franklin, M. J, Halevy, A, Maier D. From databases to dataspaces: A new abstraction for information management. ACM SIGMOD Record. 2005; 34: 27-33
[7] Quinlan, J, R: Induction of decision trees, Machine Learning. 1986; 1: 81-106
[8] Oleghe, O, Salonitis, K.: A framework for designing data pipelines for manufacturing systems. Procedia CIRP. 2020; 93: 724-729
[9] Biswas, S, Wardat, M, Rajan, H.: The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small, and in-the-large. In

Proceedings of the 44th International Conference on Software Engineering. 2022: 2091-2103
[10] Karumuri, S, Solleza, F, Zdonik, S, Tatbul, N S, Solleza, F, S, Tatbul.: Towards observability data management at scale. ACM SIGMOD. 2021; 49: 18-23
[11] Agostinelli, S, Benvenuti, D, De Luzi, F, Marrella A.: Big Data Pipeline Discovery through Process Mining Challenges and Research Directions. ITBPM@ BPM. 2021: 50-55
[12] D. Roman.: Big Data Pipelines on the Computing Continuum: Tapping the Dark Data, in Computer. 2022; 55: 74-84
[13] Benvenuti,D, Falleroni, L, Marrella, A, Perales, F.: An Interactive Approach to Support Event Log Generation for Data Pipeline Discovery. IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA. 2022: 1172-1177
[14] Quemy, A.: Data Pipeline Selection and Optimization. In *DOLAP. 2019; 1-12*
[15] Goodhope, K, Koshy, J, Kreps, J, Narkhede, N, Park, R, Rao, J, Ye, V. Y.: Building LinkedIn's Real-time Activity Data Pipeline. IEEE Data Eng. Bull. 2012; 35: 33-45
[16] Eve, M, P.: A data pipeline with Apache Airflow and Dask. 2023; 1-6