Test Data Generation for Large-Scale ETL Pipelines: Enhancing Data Integrity in Financial Services

CS4991 Capstone Report, 2024

Arjun Trivedi
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
pft8rw@virginia.edu

ABSTRACT

In financial services, ensuring the accuracy and reliability of data processing systems poses a significant challenge when handling millions of daily customer transactions through Extract, Transform, Load (ETL) pipelines. To address this, I developed a synthetic test data generation system during my software engineering internship at JPMorgan Chase. The solution integrated web technologies, cloud services, and artificial intelligence to create and manage test data that closely simulated real-world scenarios. I then fed this synthetic data into the ETL pipeline at the extract stage, allowing for comprehensive testing of the entire process. Implementation involved leveraging cloud-based data cataloging and parallel processing techniques to enable efficient data generation. The system significantly reduced the time required for testing complex ETL pipelines, improving both the speed and accuracy of quality assurance processes. Future work could explore expanding the system's capabilities to handle a wider variety of financial data types and incorporating more advanced machine learning techniques.

1. INTRODUCTION

In large financial institutions, data integrity is critical, with billions of transactions flowing through complex data processing pipelines daily. Even a minor error in processing can lead to significant financial and regulatory consequences, making robust Extract, Transform, Load quality assurance for ETL pipelines essential. However, traditional manual testing methods are insufficient when dealing with systems that process upwards of 60 million customer events per hour, as they cannot keep pace with the scale and complexity of modern data pipelines.

JPMorgan Chase, one of the world's largest financial institutions, faces this challenge in ensuring the accuracy and reliability of its ETL processes, particularly within its credit card department. Without efficient and scalable testing, errors in the data pipelines could lead to delayed transactions, misleading business analysis, or compromised customer experiences. To address these challenges, I developed a synthetic test data generation system during my software engineering internship at JPMorgan Chase. This system automates the creation of realistic, privacy-preserving test data, enabling comprehensive testing of ETL pipelines. By leveraging advanced cloud technologies and artificial intelligence, the system provides a more efficient and reliable solution to the testing problem.

2. RELATED WORKS

Synthetic data generation has been widely applied across industries, from healthcare to finance, to address challenges related to data privacy and limited access to real-world datasets. Rajotte, et al. (2021) describe synthetic data as a tool that enables machine learning applications in privacy-sensitive domains like healthcare. Their study emphasizes how synthetic data can replicate real-world patterns without exposing sensitive information, making it a promising approach for sectors that face strict privacy regulations. This aligns with the use of synthetic data in financial services, where privacy concerns similarly limit data access and sharing.

Assefa et al. (2019) further explore the use of synthetic data in financial applications, particularly in retail banking and market microstructure. They highlight how synthetic data generation can mitigate the risks associated with using real financial data, which often contains sensitive customer information. They also emphasize the importance of ensuring that synthetic datasets accurately reflect the properties of real data while adhering to privacy constraints, a critical consideration in the design of my system.

Both studies demonstrate the potential of synthetic data to address privacy concerns while facilitating large-scale testing and model training. My project built on these approaches by incorporating cloud technologies and parallel processing to enhance both the realism and scalability of the synthetic data used in ETL pipeline testing.

3. PROJECT DESIGN

The project aimed to address the need for efficient and scalable test data generation for ETL pipelines, which process millions of customer transactions per hour. Given the scale and complexity of these pipelines, manual data generation methods were insufficient for ensuring the accuracy and reliability of the system. Therefore, the design of the test data generation system focused on automating the creation of synthetic datasets that mirrored real-world scenarios while maintaining privacy and security. The system had to be adaptable, allowing for the seamless addition of new datasets and schema updates, and had to generate large volumes of test data within short time frames to support pipeline validation processes.

3.1 API Architecture

The test data generation framework was built using Java Spring Boot, which provided a foundation for developing and exposing RESTful APIs. These APIs served as the main interface for users to interact with the test data generation system. The architecture allowed for POST requests to generate synthetic data for specific datasets, GET requests to retrieve test data in compressed CSV format, and DELETE requests to remove datasets after testing was completed. Spring Boot's scalability and ease of integration with cloud services made it a suitable choice for handling the large-scale data generation and processing demands of the project.

3.2 Internal AI Tool

One of the core components of the system was the integration with the company's proprietary AI-driven synthetic data generation tool, developed by their AI research team. This tool used metadata, such as column names, data types, and statistical distributions, to generate realistic data that reflected patterns observed in real-world customer transactions. The test data generation process was automated through API calls to the AI tool, allowing for efficient

creation of large datasets that could be used for comprehensive pipeline testing.

3.3 Parallel Processing and Performance Optimization

To meet the challenge of generating large volumes of test data quickly, the system employed the Strategy design pattern along with Java's CompletableFuture API. This allowed data generation to be handled concurrently, optimizing performance across multiple datasets at once. The concurrent execution significantly reduced the time required for test data generation, ensuring timely validation of the ETL pipelines. This approach ensured the system could scale to meet the increasing data demands of the team and potentially accommodate other stakeholders' needs in the future.

3.4 Data Validation and Integration with ETL Pipelines

Once the synthetic data was generated, it was validated against existing schema definitions using the AWS Glue Data Catalog, a centralized repository that stores metadata about datasets and ensures compliance with schema requirements. This step ensured the data was compatible with the ETL pipelines, allowing for seamless integration during the testing phase. The validated data was then compressed and stored in CSV format, enabling efficient storage and retrieval during performance testing and functional validation of the pipelines. This approach ensured that the data generated was accurate, efficient, and ready for use in further testing cycles.

4. RESULTS

The test data generation system is currently in the process of being pushed to production. While the final impact has yet to be fully realized, internal testing and development have shown promising results. The system is expected to reduce the time required for testing ETL pipelines by up to 40%, based on internal benchmarks and feedback from team members. This efficiency improvement is driven by the automation of synthetic data generation and the system's ability to handle large datasets concurrently through parallel processing.

Additionally, integration with AWS Glue Data Catalog ensures that the synthetic data aligns with existing schema definitions, enabling smooth integration with ETL pipelines. This validation process improves overall data quality and reduces disruptions during testing. As the system is deployed in production, its performance will be closely monitored, with further optimizations potentially explored to enhance scalability and efficiency.

5. CONCLUSION

The synthetic test data generation system addresses a critical need in automating and optimizing the testing of ETL pipelines at JPMorgan Chase. By reducing the time required for data generation and validation, the system provides significant improvements in testing efficiency, allowing for more frequent and thorough validation of pipelines. Key features, such as metadata-driven data generation, seamless integration with AWS Glue Data Catalog, and parallel processing, make the system scalable and adaptable to the firm's evolving data needs.

This project not only enhances the accuracy and reliability of data processing pipelines but also contributes to a more streamlined quality assurance process. As the system moves into production, its anticipated value lies in faster testing cycles, improved data quality, and the ability to adapt to future business requirements, ensuring it will contribute long-term to data engineering operations in finance.

6. FUTURE WORK

Moving forward, the project can be expanded by developing a cloud-native implementation of the test data generation platform. This involves leveraging more AWS services such as S3 Buckets or DynamoDB to persist the generated test data, ensuring scalability and ease of access. Cloud-native architecture would further enhance the platform's ability to handle growing data needs while maintaining robust performance.

Another key area for development is the addition of a user-interface, as the current implementation functions solely as a backend application. A frontend interface would provide users with an intuitive way to manage datasets, trigger data generation, and monitor progress, improving overall usability. These enhancements will ensure that the platform remains an essential tool for data engineering teams at the firm.

REFERENCES

Assefa, S. A., Dervovic, D., Mahfouz, M., Tillman, R. E., Reddy, P., & Veloso, M. (2020). Generating synthetic data in Finance. *Proceedings of the First ACM International Conference on AI inFinance*.https://doi.org/10.1145/3383455.3422554

Rajotte, J.-F., Bergen, R., Buckeridge, D. L., El Emam, K., Ng, R., & Strome, E. (2022). Synthetic data as an enabler for machine learning applications in medicine. *iScience*, *25*(11), 105331. https://doi.org/10.1016/j.isci.2022.105331