International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Optimizing Data Workflows through the Migration from MapR ETL to Airflow S3 Pipelines

Pankaj Dureja

Email: *pankaj.dureja[at]gmail.com*

Abstract: ***This article outlines the strategic migration from legacy MapR ETL data loading to AirflowS3 pipelines, emphasizing the operational efficiency gained through cloud storage and Apache Airflow orchestration. The process, challenges encountered, and solutions deployed during this migration are detailed, highlighting significant improvements in data handling, cost reduction, and scalability.***

Keywords: Data Migration, ETL, Apache Airflow, Amazon S3, Data Pipelines

1. Introduction

The introduction will contextualize the need for advanced data pipeline solutions in handling increasing data volumes and complexity in modern enterprises. It will explain the initial use of MapR ETL tools for data processing and the evolving needs that required a shift to more scalable solutions like Airflow and S3. The introduction will set the stage for discussing the integration of these technologies into existing IT infrastructures, emphasizing their role in enhancing data agility and accessibility.

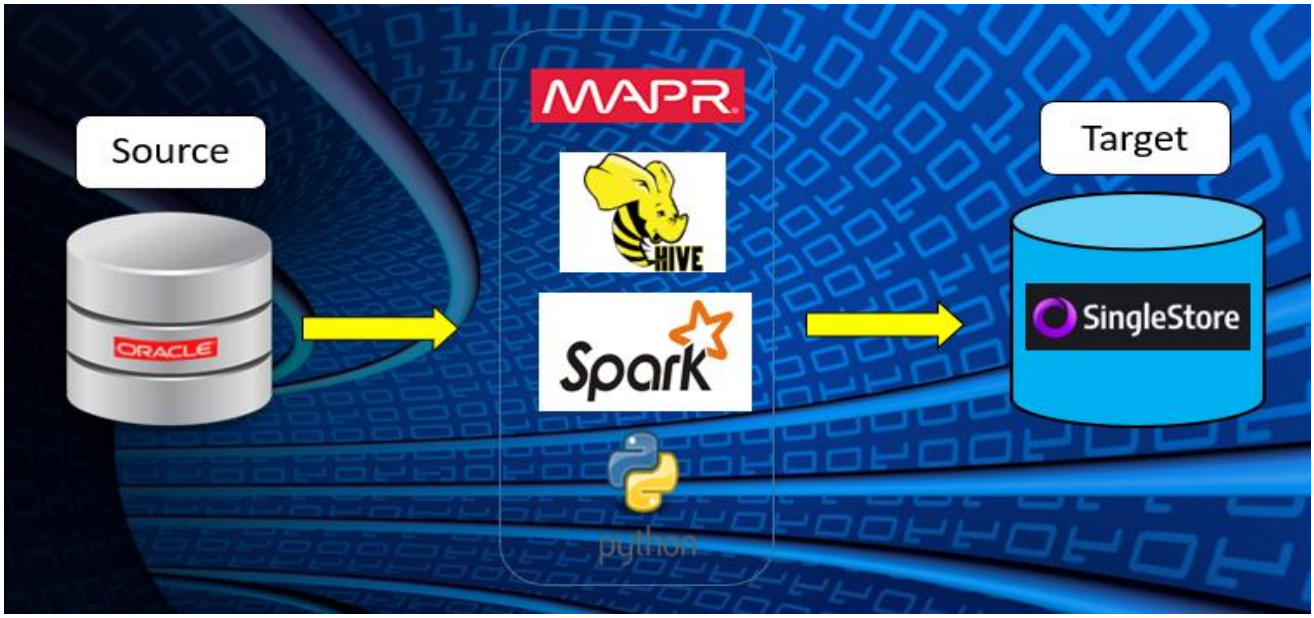
2. Problem Statement

In the existing MapR ETL framework, our oil and gas company faced significant operational and financial challenges that hindered efficient data management. This system required highly specialized skills, making labor both scarce and expensive. Additionally, the logging capabilities within MapR were notably inadequate, complicating efforts to pinpoint failures within the data transfer processes. This issue was compounded by a lack of robust restart ability, meaning that any process interruption could lead to substantial delays and complexities in resuming operations.

MapR system is unmonitorable - and keeping an eye on a running MapR instance turned out to be another big job, there are too few tools for monitoring or successful interfaces in order to track data flows and detect errors. The notable technical challenges included sticky bit management, empty directory handling and de - duplication issues which could result in data integrity deterioration or pipeline inefficiency. Complicating the scenario were MapR's bankruptcy and its acquisition by HP, which pushed licensing costs up to an order of magnitude higher than before. Together these challenges convinced us to build a new technology platform that would do this differently - one where operations were simpler, costs lower and which was far more robust/scalable in terms of our data management footprint.

Existing ETL Architecture:

- Read from oracle using sqoop
- Data storage is done in hive.
- Load it to Singestore using the pyspark program.
- Jobs are scheduled through AdTempus.



Current Challenges:

- Supportability for the process is limited
- The process is computed heavy (MapR is 40 systems)
- MapR is EOL (End of Life) with limited support options
- Big Data itself has evolved significantly over the past few years. Hadoop is being replaced with Object Stores.
- Scalability Issues
- Not able to take advantage of Exadata capabilities w. r. t to the infrastructure changes happening at company:
- Achieving Parallelism is challenging.
- Requires a lot of coding

In light of these challenges, we were assigned to create and deploy a streamlined solution for data management that would replace the MapR ETL method as it was no longer sustainable nor economical. Nevertheless, the key objective was not only to fix a few urgent technical and operational issues but rather establish infrastructure that can scale reach out globally in future.

Solution Implemented:

The solution follows a meticulously designed sequence:

1) **Data Export from Oracle:** Using Oracle's built - in export capabilities, data is extracted and formatted into CSV files, which are subsequently uploaded to an S3 bucket. This extraction process is tailored for large datasets, employing parallel export operations to enhance efficiency. The operation utilizes an Oracle external directory that is configured to point directly to Amazon S3 storage. The data export task is managed by an Oracle function, which executes a SQL query specified in the configuration file to enable parallel data export.
2) **Data Loading to SingleStore:** Upon successful storage in S3, SingleStore pipelines are triggered to load the data directly from S3. These pipelines are designed for high throughput and minimal latency, ensuring data is quickly available for processing.
3) **Data Processing in MemSQL**: Once the data reaches MemSQL, it is initially stored in a temporary staging table. A stored procedure then manages this data by employing a delete - and - insert strategy to update the target tables. This method is essential for ensuring data consistency and integrity, especially when managing updates and deletions.
4) **Orchestration with Airflow**: Apache Airflow orchestrates and schedules the workflow, ensuring each step proceeds in the correct sequence while continuously monitoring for failures. Airflow's comprehensive error handling and retry capabilities significantly boost the reliability of the pipeline.

Advantages of using Apache Airflow include:

Apache Airflow offers several compelling advantages for managing data workflows. Its dashboard facilitates easy monitoring, providing detailed logs that simplify the oversight of Directed Acyclic Graphs (DAGs) and troubleshooting of issues. Robust error management features allow for specific remedial actions, including process reruns via a user - friendly web interface, enhancing operational reliability. Airflow also supports high availability by running DAGs from multiple executors; if one fails, another can seamlessly take over, minimizing downtime. Additionally, its flexible task management capabilities enable the use of various operators—be they pre - built or customized by integrating the necessary Python libraries. This flexibility is demonstrated in this project through the use of Bash, Oracle, MySQL, Python, and other operators, streamlining complex data integration tasks. These features make Apache Airflow an invaluable tool for managing complex data workflows, enhancing operational efficiency and system resilience.

Based on the above implementation the new architecture looks like as follows:

- Write out of NFS using native server process
- Ingest from S3 using native SingleStore Pipeline
- Transform/Merge During ingest.
- Schedule jobs through Airflow scheduler DAGs.

Benefits:

- Utilize Exadata, Vast and SingleStore to its fullest potential.
- Write once and read multiple times.
- Use of native tools which involves minimal moving parts.
- Better Scalability.
- Centralized scheduling with on - demand refresh.
- Robust error handling with easy recovery.

Milestone Achievement:

The migration from the MapR ETL framework to Airflow S3 Data Pipelines marked a significant advancement in our data management capabilities, particularly evident in our weekly data loads. These loads consist of processing data across 21 diverse tables, with record counts ranging from thousands up to 200 million. Under the MapR ETL system, this task required approximately four hours to complete. However, after transitioning to the Airflow S3 Data Pipelines, the same workload was accomplished in less than 40 minutes. This dramatic reduction in processing time by over 80% highlights a major leap in efficiency and performance.

By moving to Airflow S3 Data Pipelines, we not only achieved remarkable improvements in processing speed but

also addressed the broader challenges of cost, reliability, monitoring, and operational flexibility. This transition has positioned our data infrastructure to be more responsive to the company's needs and better aligned with industry best practices for data management.

Potential Extended Use Cases:
In addition to just transferring data, this ETL pipeline can help - Incremental Data loads for near real time analytics - Integrating the data from multiple sources so that you get a consistent view of your application and Infrastructure The processed data being fed into Machine learning models thereby making predictive analysis smarter. For instance, a retail company that leverages this pipeline to route sales data from multiple regions for unified reporting and inventory management.

Impact:
Channeling similar options, the adoption of this Airflow - managed ETL pipeline minimizes manual overhead drastically and increases data availability much faster leading to rapid decisions making as well for operational efficiency. For example, a financial institution could optimize their data aggregation and reporting tactics that would allow the business to react rapidly to market movements.

Scope:
While the primary application described involves Oracle, S3, and MemSQL, the principles and methodologies can be adapted for other source and target systems in both on - premise and cloud environments. The study employed a sequential method involving data export, loading, processing, and orchestration using Oracle, SingleStore, and Apache Airflow. Each step was carefully designed to maximize efficiency and ensure data integrity.

3. Conclusion

The migration to AirflowS3 pipelines marks a significant advancement in data management, offering enhanced efficiency, scalability, and reliability. This transition not only reduces operational costs but also positions the organization for future growth in data handling capabilities.

References

[1] Maxime Beauchemin, "The Apache Airflow Book", O'Reilly Media, 2021, pp.45 - 70.
[2] Anirudh Kala, "Apache Airflow: A Real - World Guide to Data Pipelines", Packt Publishing, 2020, pp.115 - 140.
[3] Amazon S3 Storage, Available at https: //aws. amazon. com/s3/
[4] Amazon S3 Storage on Premise using outposts, Available at https: //aws. amazon. com/s3/outposts/?whats - new - cards. sort - by=item. additionalFields. postDateTime&whats - new - cards. sort - order=desc
[5] Oracle PL/SQL Functions, Available at https: //docs. oracle. com/en/database/other - databases/timesten/22.1/plsql - developer/pl - sql - procedures - and - functions. html
[6] Oracle Directory concepts, Available at https: //docs. oracle. com/en/database/oracle/oracle - database/19/sqlrf/CREATE - DIRECTORY. html
[7] SingleStore Pipelines, Available at https: //docs. singlestore. com/cloud/load - data/load - data - with - pipelines/pipeline - concepts/
[8] SingleStore Procedures Available at https: //docs. singlestore. com/cloud/reference/sql - reference/procedural - sql - reference/create - procedure/
[9] Kathleen Ting, Jarek Jarcec Cecho (2013). Chapter [7] Apache Sqoop Cookbook: Specialized Connectors: Importing from Oracle, pp.63 - 70.
[10] Muhammad Asif Abbasi. Learning Apache Spark 2: Chapter [1] Architecture and Installation: Apache spark architecture overview, pp.9.
[11] Sqoop User Guide. Available at https: //sqoop. apache. org/docs/1.4.6/SqoopUserGuide. html#_introduction
[12] Apache Spark 2.1.1. Available at https: //spark. apache. org/docs/2.1.1/
[13] Hive Language Manual. Available at http: //wiki. apache. org/hadoop/Hive/LanguageManual.