Prodapt

Unleash the power of cloud modernization

Accelerate migration of complex data pipelines to modern cloud services using a holistic approach

Sivasubramanian Veerabahu | Mathi T
Srinivasavaradan Parthasarathi | Priyankaa A

Service providers' ever-growing data and customer services demands a cloud-based big data ecosystem

Service providers face several challenges in managing and processing massive amounts of data generated every day from call detail records, networks and application logs from various sources. Leveraging a big data platform like Hadoop helps to manage, analyze, and derive insights from the extensive data. However, legacy Hadoop systems can hinder service providers' operations with performance limitations, scalability challenges, and increased maintenance effort.

According to Forrester, more enterprises are **tired of Hadoop's on-premises complexity** and shifting to the public cloud. **Serverless** and Hadoop alternatives in the public cloud will gain more traction in the near future".

Major challenges faced by service providers with legacy Hadoop systems

- Ineffective processing of big data by running multi-node cluster for a long time
- Inefficient monitoring of infrastructure usage
- Lack of mechanisms to alert on failure while running workloads
- Lack of unified dashboard for ETL jobs, impacting the resolution time

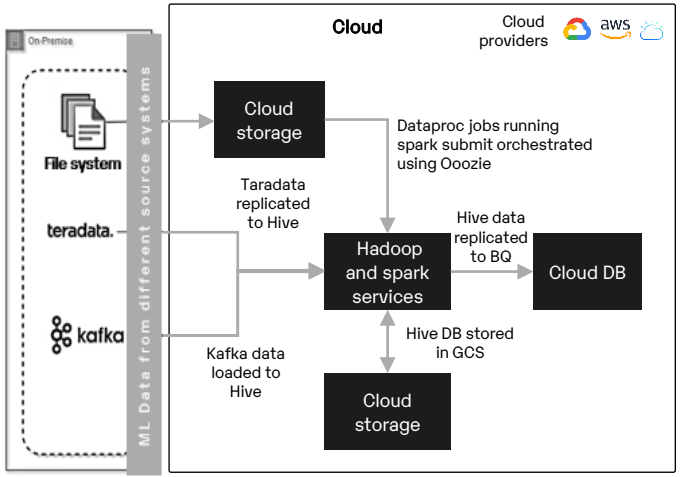
How legacy Hadoop systems impact service providers' operations

- Missed SLAs impacting revenue targets and customer experience
- Increased Opex, due to long-running clusters

To overcome these challenges, service providers must move towards a **cloud-based Hadoop** ecosystem. As an instance, this insight details the transition of ML use cases to cloud-based Hadoop as it enables efficient handling of large volumes of ML data, real-time data analysis, and faster decision-making.

Implement cloud modernization and unlock the full potential of Machine Learning applications

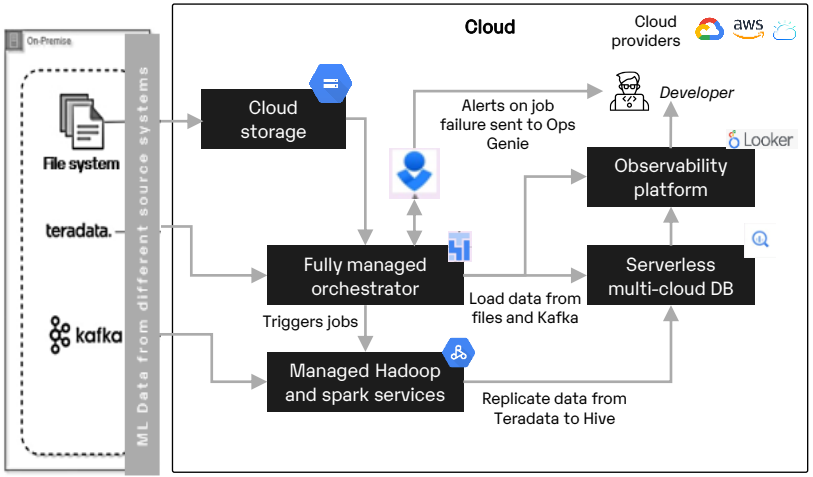
Traditional approach of migrating data pipelines to cloud



Typical approaches taken by service providers for migration of ML workloads

Serverless provides myriad of benefits when compared to traditional cloud. Hence, **cloud modernization** is the optimal approach for the service providers as they migrate their ML use cases to cloud

Cloud modernization approach to maximize value of migration



Benefits of cloud modernization

- 40% reduction in IT spend
- 30% faster ML predictions
- Achieve ~90% SLAs
- Enhance security of ML data

Cloud modernization offers several benefits in terms of agility, process efficiency and time to insights. However rapid and successful migration of **complex data pipelines** with **multiple transformations** to modern cloud services requires effective assessment, planning and end-to-end automation.

Strategy to accelerate the migration of complex data pipelines from legacy to modern cloud-based Hadoop

1

Assess the existing data and workflows for efficient migration

Make informed decisions on migrating jobs and workflows

2

Enable automatic migration to modern cloud services and reduce downtime of ML applications

Migrate petabytes of complex workloads smoothly and rapidly

4-step strategy to accelerate cloud modernization of complex data pipelines

4

Develop an observability dashboard

Enables the team with end-to-end view of data lineage from on-prem to modern cloud

3

Validate the jobs and data post migration to ensure accurate ML predictions

Ensure getting the expected results

The following slides dive deep into the four-step strategy for successful cloud modernization of complex data pipelines.

Assess the existing data and workflows for efficient migration

By analyzing the existing workflows, dependencies and data flows, service providers can make informed decisions about which workflow to migrate, how to optimize the cloud resources, and ensure a successful and seamless transition to the modern cloud environment.

1 A B 2 3 4

Recommendations

1

Validate if the workflow is active or inactive

- Check if source files are flowing regularly
- Make sure if source tables are populated regularly
- Ensure the availability of job logs are available
- Identify inactive workflows and decommission them to avoid effort wastage

2

Identify the observability parameters

- Determine and capture the observability parameters for every workflow. This assists service providers with the data lineage from source to target

3

Evaluate the complexity of the workflows

- **Simple**- If the workflow uses simple hive SQLs
- **Medium**- If the workflow uses many temporary tables with joins and correlated/nested subqueries
- **Complex-** If the workflow uses Scala code or business logic implemented using functions
  - Wrap the complex jobs as Dataproc trigger from the Cloud composer
  - Develop a user-defined function and plugin for pre-processing of files once received. Integrate and invoke them using a cloud function so they do not use cloud composer memory. This is critical to avoid run time performance bottlenecks

Assess the existing data and workflows for efficient migration

Identifying and managing the dependencies of workflows are critical to ensure a successful migration of ML workloads to the cloud. Analyzing and addressing the dependencies helps mitigate risks and ensures the correct functioning of your workflows in the cloud.

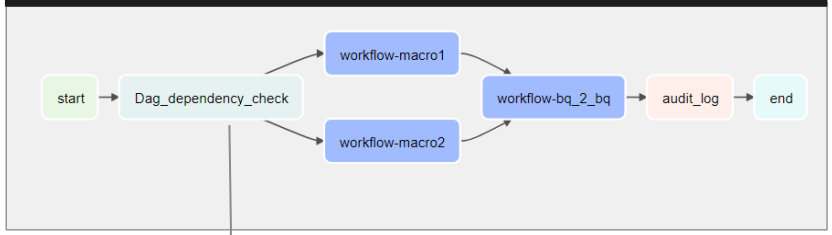
1 A B 2 3 4

Recommendations

Identify the dependencies of workflow

- Identify both internal and external dependencies for all the workflows. Check for external dependencies, such as third-party APIs, external data sources, or external services
- Determine **simple dependency** scenarios such as daily, weekly, monthly, and **complex dependency** such as once in a week or once in the last 2 hours
- Define the dependency parameters, test and fine-tune them for the workflows with Directed Acyclic Graph (DAG) dependencies
- Develop a **custom Python script** to **automate** the process of **pausing** and **unpausing** all DAGs at the same time in a workflow management system like Apache Airflow. This helps save time for 1000+ DAGs during cutover, composer upgrades, and framework redeployment

Sample DAG dependency workflow



A DAG dependency check examines the dependencies between tasks or components in a DAG-based workflow or system. It ensures that tasks are executed in the correct order and that there are no circular dependencies, which could lead to deadlocks or incorrect results.

- Leverage tools like **Google Cloud Composer** to link the interdependencies between workflows. This helps reduce the workflow queue size and optimize the operating costs

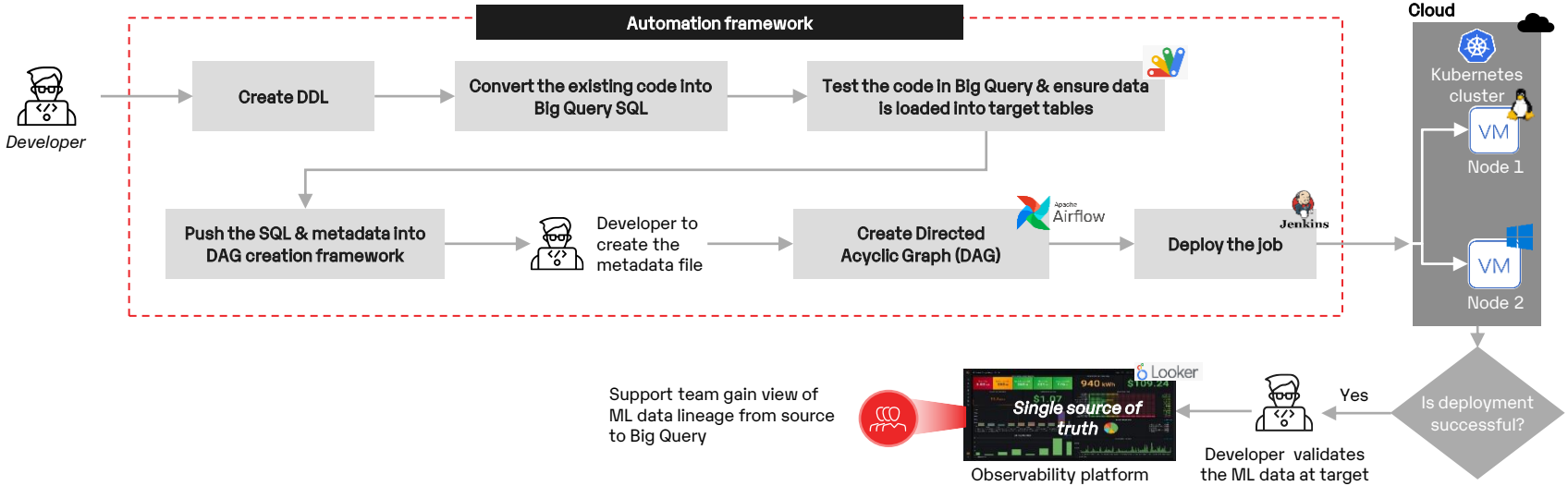
Prodapt

Enable automatic migration of data pipelines to modern cloud services to reduce the downtime of ML applications

1 2 A B 3 4

Automatic migration of complex workloads to modern cloud services offers numerous benefits that can enhance machine learning workflows' efficiency, scalability, and effectiveness.

Sample use case: Automatic migration of complex workloads to modern Google Cloud services



Recommendations

- Leverage cloud translation services like **Google Cloud Translate** for automatic conversion of the legacy SQL to Big Query dialects
- Implement cloud business intelligence tools like **Looker, MicroStrategy, and PowerBI** to track the data flow from source to target. It also helps service providers to track and manage ML use cases like customer churn by creating a dashboard with the model output and source data by connecting directly to Big Query

Enable automatic migration to modern cloud services and reduce downtime of ML applications

Recommendations for automatic and rapid migration of complex data pipelines

1

Automate DDL generation

Create custom scripts to auto-generate DDL and ensure it uses the cluster keys to generate the partitioned tables. This helps **save 30 minutes per table** and plenty of time for developers as 1000+ tables are created automatically

2

Achieve auto conversion of HiveQL to Big Query

Develop **Google App scripts** to extract and translate SQL from GitHub source code automatically. This results in huge time saving while extracting and translating thousands of SQL queries

3

Automate the migration of historical data tables for cutover

Develop custom scripts to find optimal partition sizes for huge, unevenly partitioned data. This helps with the smooth migration of tables measuring in terabytes, which would otherwise require reservation of huge memory

4

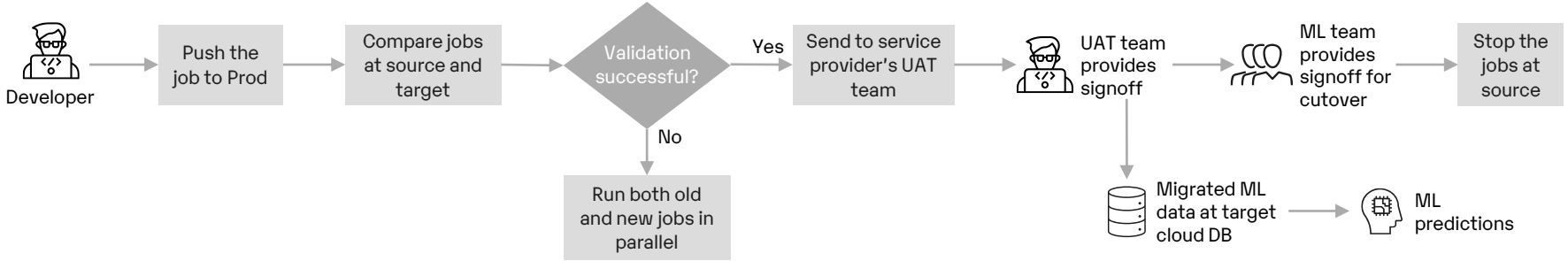
Enable automation of production job run timing extractor

Extract the production data proc job run timings for the deployed jobs and compare them with existing BAU jobs. This assists in **saving 2 hours per day** for the developers

Validate jobs and data post migration to ensure accurate ML predictions

Validating jobs after moving to the cloud is a critical step to ensure that your workloads are functioning correctly and delivering the expected results in the new cloud environment.

Sample validation flow to ensure the migrated workflows and data are same as source systems



Recommendations

- Create a separate orchestration environment to avoid resource contention between actual prod jobs and parallel User Acceptance Testing (UAT) jobs
- Implement open-source automation tools like **Jenkins** for automated and rapid testing of jobs in a development environment
- Run both old jobs and new jobs in parallel until validation is successful. This helps service providers to curb data loss

Observability dashboard – Enabling teams with an end-to-end view of data lineage

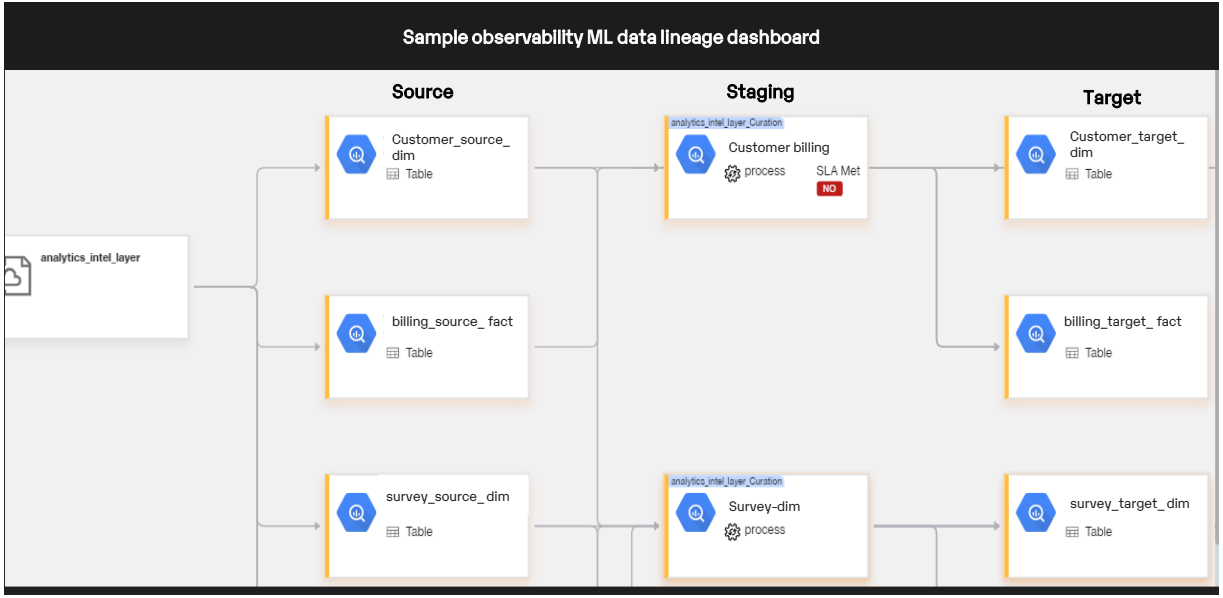
1 2 3 4

Provides a complete view of the ML data lineage from source to target for all applications

Helps the support team to identify job failures/issues and resolve them quicker

Provides the view of where exactly the job fails and helps re-run jobs from where they failed instead of running from scratch

Tracks the in-progress jobs, SLAs and locate bottlenecks



With the observability dashboard, the support team can have E2E visibility into all jobs and take prompt actions when certain ML tables are required/missed for prediction at the target cloud.

Progress | Pending | Success | Failed

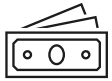
Prodapt

Business benefits achieved by a leading service provider post successful cloud modernization of workloads

Implementing the four-step strategy as discussed in this insight, resulted in the following benefits.

Time Optimization

99% Reduced 8 hours of running workload into 10 minutes. This enabled quick access to ML data and decision making



Cost savings

40% Migrated complex workloads to modern cloud services and enabled quick processing of ML data

Real-time Alerts

Enabled observability dashboard and integrated with an alerting system for ML data lineage

Faster and efficient ML predictions

Optimized Infrastructure usage

Reduced time-to-market of new features

Thank you

insights@prodapt.com