High Volume Intelligent Streaming with Sub-Minute SLA for Near Real-Time Data Replication

Suneel Konidala
Murali Madireddi
How is this session organized?

- Use Case
- Architecture
- Why DLT?
- Structured Streaming
- Intelligent Streaming
- Q & A
Accenture and Databricks

Strong partnership that spanned across years

- #1 Databricks Partner of the Year since 2019
- 100+ Customer Implementations on AWS, Azure and GCP
- 10+ AI@scale Industry solutions to deliver immediate client value
- 17 Assets and Accelerators to accelerate Cloud Migrations
- 4,000+ Team members skilled in Databricks
- 30+ Databricks champions and over 700 Databricks Certifications
Our goal is to develop a framework that can ingest and curate data in batch and near-real-time, while also managing inter-dependencies between incoming datasets based on metadata, in a cost-efficient way.
Use Case

Needed a versatile framework to hydrate cloud data lake

- Source data is from on-prem as well as from other cloud sources
- Data is made up of on-prem OLTP tables, Teradata, mainframe (DB2 and IMS), and flat files
- Source data is extracted by multiple tools and techniques (Informatica, native database exports, JDBC clients etc.,)
- Variety of file formats – CSV, JSON, XML
- Handle SCD1, SCD2 type of transactions
- Data comes in both batch and streaming modes

- Perform data quality checks and quarantine bad records/files
- Schema validation as well as schema evolution
- Data transformation between multiple zones of the Lakehouse
- Datastores are Delta tables as well as other cloud native RDBMS
- Thousands of tables to be ingested into cloud platform
- Critical tables to be replicated in near real time
Lakehouse Architecture
Followed best practices of Medallion architecture
Capabilities Built as Part of the Framework

Scalable metadata driven framework to address today’s data needs

- Metadata driven framework that is easy to understand and use
- Supports batch, stream, and API requirements
- Supports loads to data lake and consumption layer
- Performs Audit, Balance and Control checks on data
- Captures data observability metrics
- Detects out of sync events and performs automatic recovery
- Generates Delta Live Tables (DLT) compatible code automatically
- Generates Structured streaming code for Big Query and PostgreSQL tables
- Metadata model based on Delta tables
- Databricks multitask based orchestration
- Integration with monitoring and alerting tools
Standard ETL Spec to Enforce, Transform and Map

Easily readable spec with multiple use cases

- Data analyst or Data Engineer creates the ETL spec in a simple tabular format
- Spec contains
  - Source to Target mapping
  - Data quality, curation rules
  - Join logic for aggregation
  - Rules to derive new columns
- Framework reads spec and generates ETL pipelines
- Specs can become valuable central repository of metadata and provide a variety of reports for support engineers

- Generate ETL Pipelines
- Maintain dependency
- Consistent code
- Operational Reports
- Data Lineage
- Query data validation and transformation rules
Delta Live Tables (DLT)
An automatic way to generate data pipelines and manage them

We choose to leverage DLTs in our solution

• Simple SQL based syntax to create the data pipelines
• Automatically manages the dependency between multiple tables and zones
• Can perform data quality checks as well as transformations
• The tables are automatically maintained (optimize, vacuum)
• DLT workflows can be part of jobs, makes the orchestration easy with other non-DLT tasks
• Can easily process SCD1 and SCD2 type of transactions
Out of Sync Detection and Recovery
Identify and correct schema and data drifts between on-prem and cloud

Over a period of time, schema and data can drift between source (on-prem) and target (cloud) environments. An automatic process has been implemented for another level of validation at regular intervals:

- Get row counts and schema from both source and target tables
- If schema doesn’t match, update target schema to match source
- If counts don’t match within a pre-defined threshold, extract data from source and load into target tables

This process addresses many edge conditions to maintain the source and target data in sync.
Why Streaming?

CDC is a critical requirement for the framework

• Need a solution to ingest data into thousands of critical tables in near real time (NRT) to cloud
• Operational data changes at source need to be replicated within SLO of < 1min
• Not all tables need to be replicated in real time, only important tables identified by business
• Preference is to remain with a unified batch/streaming technology and not use a third-party solution
• Solution should be cost efficient, easy to maintain and add additional functionality
• Solution should be extensible to include more tables in the future via metadata
• Current data observability standards should be maintained
Structured Streaming

**Speed at cost**

- Structured streaming works well as far as the functionality is considered.
- However, a cluster can have only a set number of active threads.
- Need to setup multiple cluster(s) and manage the usage and costs even when there is no incoming data for some tables.
- Heavy volume streaming workloads require more clusters, which results in higher costs.
Intelligent Streaming
A solution to optimize the cluster resources

- A primary job continually looks for new data among multiple database topics
- Discards data from non-relevant tables
- Uses thread pool concept and creates number of threads that is equivalent to number of tables with data
- Refines the data, renames columns and loads silver zone tables
- Logs job status and errors
- Implemented fault tolerance to automatically retry and restart in case of failures
- Tuned cluster parameters to avoid frequent autoscaling
Let’s Take a Close Look..
Structured Streaming vs. Intelligent Streaming

Structured Streaming

- Data is consumed from Kafka and written to PostgreSQL using parallel threads
- Thread is in use even when there is no incoming data for a table
- Best suites when there is continuous inflow of data

Intelligent Streaming

- In intelligent streaming, threads are allocated as needed based on the incoming data
- Less demand for active threads results in less demand for cluster resources
- Best suites when the inflow of the data is not predictable
Performance Test
High throughput with optimal cost

- Simulated workload representing 10% of expected production workload
  - Mix of Low, Medium and High tables in terms of activity ranging from 100 to 10K TPS
- A cluster with configuration of N2-Highmem-16 (Driver = 1, Workers = 4 to 10) is used for the test
- Rolled up the data into minute interval, and in each minute
  - We processed anywhere between 30K to 100K transactions (depending on how many messages are available in Kafka topics)
  - We processed about 40 tables
- Use of pools with idle instances reduced the node startup time
- Due to time and resources restrictions, we didn’t push the limits. However, we achieved our performance and cost goals with intelligent streaming
Recap

Databricks – A true Lakehouse to process analytical, and ML workloads

• We built a single framework to manage multiple types of data loads in a cost-efficient way
• Leveraged many out of the box capabilities like DLTs and built custom solutions to improve further
• In a short span of few weeks, we onboarded thousands of tables on to the new platform
• Accenture is the largest SI partner for Databricks and executed many successful projects jointly
• We are impressed with the innovation at Databricks and speed at which they are offering new capabilities
• Can’t wait to see to many great features like Marketplace, Delta Sharing and LLM/Dolly becoming mainstream on the Databricks platform