Making Data Engineering Declarative

by the Databricks Delta Live Tables & DBSQL teams

Michael Armbrust, Bilal Aslam, Yingyi Bu, Sourav Chatterji, Yuhong Chen, Yijia Cui, Vuk Ercegovac, Ali Ghodsi, Rahul Govind, Aakash Japi, Kiavash Kianfar, Eun-Gyu Kim, Xi Liang, Paul Lappas, Jon Mio, Mukul Murthy, Supun Nakandala, Andreas Neumann, Yannis Papakonstantinou, Nitin Sharma, Yannis Sismanis, Justin Tang, Joseph Torres, Reynold Xin, Min Yang, Li Zhang
Data Engineering -> comprehensive, recent, clean derived data

Reliable data pipelines made dead simple

by Delta Live Tables

soon also in DBSQL
Declarative Pipelines for ExtractLoad-Transform

-> script defining a DAG of ingestion and downstream transform

Ingestion (EL)  Downstream (T)

- Kafka
- Kinesis
- CSV, JSON, TXT...
- Data Lake
- Streaming Tables++
- Materialized Views++

Streaming Analytics
BI & Reporting
Data Science & ML
Declarative Pipelines

-> script defining a DAG of ingestion and downstream ELT

Ingestion (EL)  Downstream (T)

CREATE STREAMING TABLE raw_data
AS SELECT *
FROM read_files("/raw_data", "json")

CREATE MATERIALIZED VIEW clean_data
AS SELECT ...
FROM raw_data

Streaming Tables++
Materialized Views++
Declarative Pipelines
  → Simplicity
    plain SQL definitions are popular
  → Automated Optimization
    Databricks declarative pipelines are optimized in the engine
Enzyme Incrementalization Planner

An automatic optimizer for incrementalizing MVs

Delta Tracked Changes
Query Plan Analysis

Inserts (+ Deletes)
Partition Recompute
MERGE Updates
Full Recompute

Heuristic Rules + Cost Model

Optimal Incrementalizing Flow

+ Catalyst Query Optimizer
Holistic Pipeline Optimization

Save infrastructure costs and/or reduce end-to-end latency

• Analyze dataflow graph
• Execute independent parts in parallel
• > 50% gains over sequential execution, when incremental workloads
More in sponsor talk (Tuesday ~10AM)

- Software Engineering practices meet Declarative Pipelines
- Data Quality
- Enhanced Autoscaling
Thank you