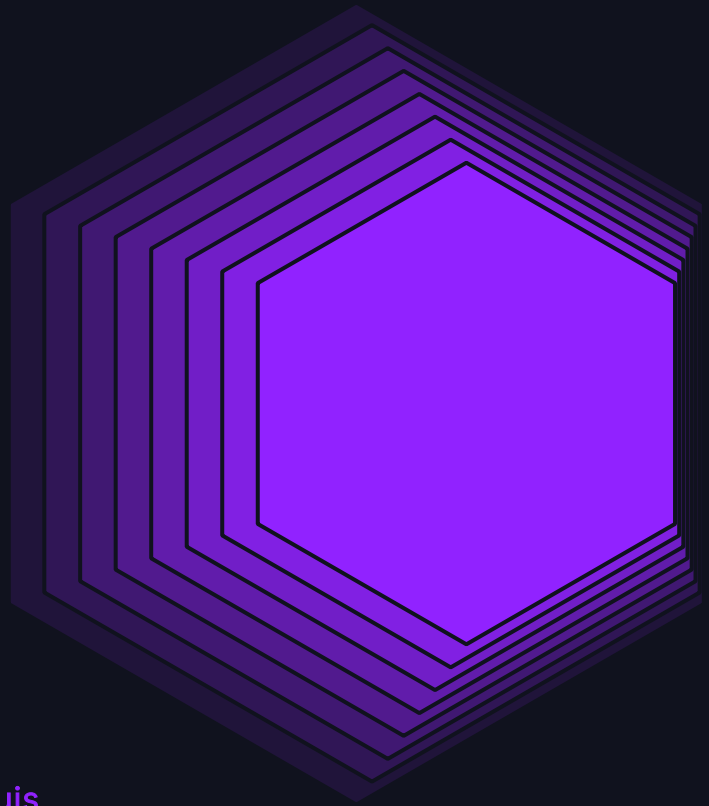
Product safe harbor statement

This information is provided to outline Databricks' general product direction and is for informational purposes only. Customers who purchase Databricks services should make their purchase decisions relying solely upon services, features, and functions that are currently available. Unreleased features or functionality described in forward-looking statements are subject to change at Databricks discretion and may not be delivered as planned or at all

DATA+AI SUMMIT BY databricks

Project Lightspeed goes Hyperspeed



Karthik Ramasamy
Principal Software Engineer

Ryan Nienhuis,
Sr. Staff Product Manager

The world operates in real-time

What about your decisions?

The speed of business has increased, as organizations need to respond to and make decisions based on what is happening now, not what happened yesterday, last week, or last month.

IDC Marketscape: Worldwide Analytic Stream Processing Software 2024

Organizations have more streaming data every year [...]. This information is most valuable when it is used as soon as it arrives to improve real-time or near-real-time business decisions.

Gartner Market Guide for Event Stream Processing 2023

NEW OPPORTUNITIES IN REAL-TIME

Every organization generates vast amounts of real-time data

Transactional records
Point of sale (POS)
Banking transactions
Airline reservations
Call center records

Third-party
News feeds
Weather
Market data
Real-time traffic

Interactions
Web clicks
Social posts
Emails
Instant messages

IoT events
Sensors
Geolocation
Machine logs
Mobile devices

Creating opportunities for new kinds of real-time applications

Fraud detection

Personalized offer

Vaccine distribution

Smart pricing

In-game analytics

Connected cars and smart devices

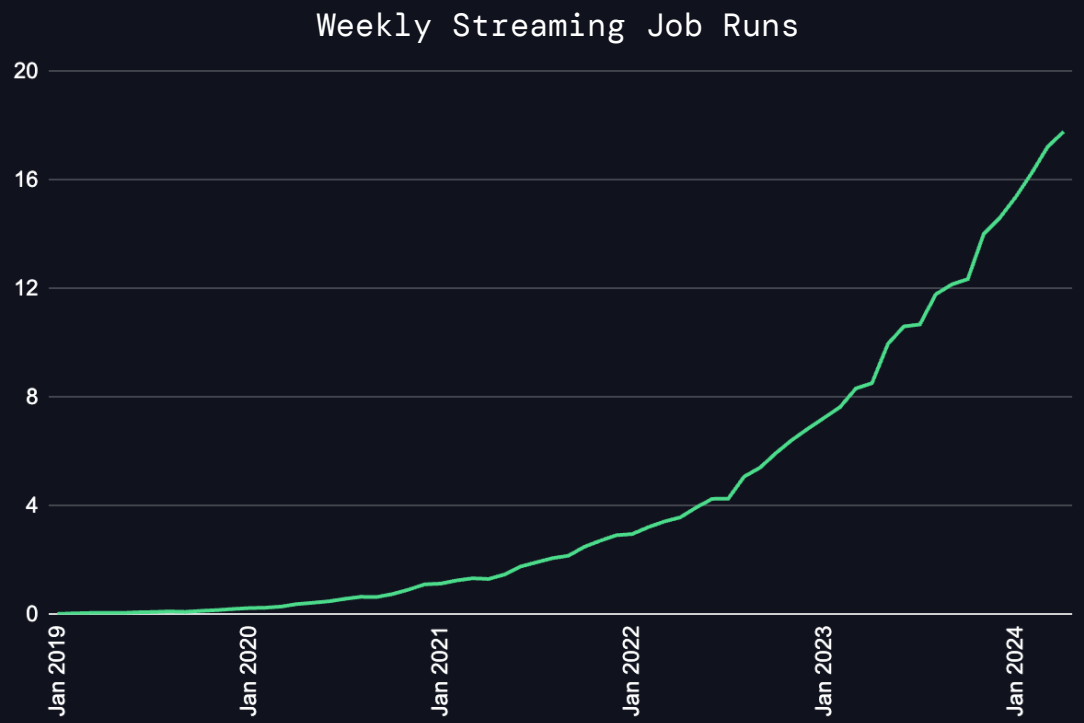
Predictive maintenance

Content recommendations

DATABRICKS LEADS STREAMING

STREAMING IS BIGGER AT DATABRICKS

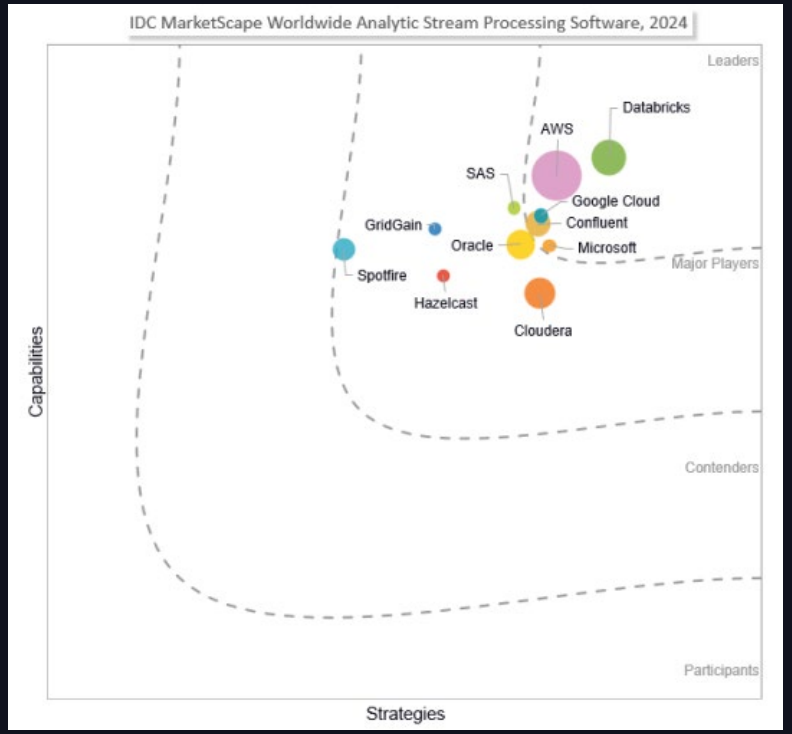
Millions of jobs
run every day



DATABRICKS IS THE BEST CHOICE

And it's not just us saying this

- Developer Experience
  - Code friendly to low code to no code
  - Available to multiple personas
- Integration with unity catalog
  - Event discovery, Schema registry
  - Stream lineage, Sensitive data tagging
- Go-to-market focus
  - Expansion into operational workloads



Databricks is built on Apache Spark

More than 36M Maven downloads/month

Unified engine for batch and streaming

Fault tolerant

Low latency and cost effective

Easy to use and powerful operators

Stateful processing

PROJECT LIGHTSPEED OVER THE YEARS

Over 30 streaming features released in <2 years

Release	Performance	Functionality	Connectors
DBR 11 Q4 2022	State re-balancing Offset management	Auto Loader avro support applyInPandasWithState Python query listener	Amazon Kinesis Enhanced Fanout (EFO)
DBR 12 Q1 2023	Photon support for ForEachBatch	Watermark support in SQL Streaming support with UC on Shared clusters	Schema Registry Auth Streaming for Delta Sharing table

PROJECT LIGHTSPEED OVER THE YEARS

Over 30 streaming features released in <2 years

Releases	Performance	Functionality	Connectors
DBR 13 Q2 to Q3 2023	Bounded memory usage RocksDB read/write improvements Changelog checkpointing Adaptive query execution Skip Delta table modifications	Stateful operator chaining Drop duplicates within watermark	Google Cloud Pub/Sub IAM support for Amazon MSK Use UC to manage external Kafka credentials
DBR 14 Q4 to Q1 2024	Pushdown filters for Delta	State Reader API Avro schema evolution	Pulsar Connector, Stream from UC views

PROJECT LIGHTSPEED GOES HYPERSPEED

Today we are going to cover…

1. Streaming ETL on **DLT Serverless with Stream Pipelining**
2. Low-latency operational use cases with new **Databricks capabilities**
3. Custom Integrations with **new Python Data Source API**
4. Arbitrary Stateful Processing with new **transformWithState API**

FASTEST INGESTION INTO THE LAKEHOUSE

BENCHMARKING INGESTION OPTIONS

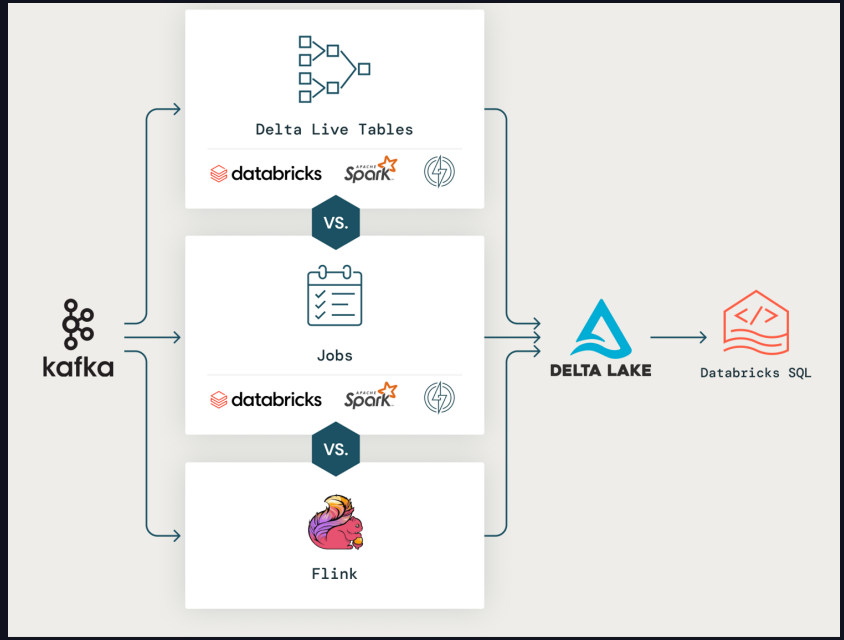
DLT Serverless, Jobs Compute, Open source alternatives

- Delta Live Tables (DLT) with serverless compute
  - Declarative data pipelining framework in Databricks Data Intelligence Platform
  - With serverless, simplifies operational burden
  - Uses Photon engine and supports stream pipelining
- Jobs Compute for Spark Structured Streaming
  - Traditional compute offering from Databricks
  - Compute instances are hosted in the customer's cloud, managed by Databricks
  - Uses Photon for the evaluation
- Other Open Source Options
  - Many options available - Apache Flink, custom Rust writer
  - We used Apache Flink from a major cloud service provider

BENCHMARK FLOW OF OPERATIONS

Apache Kafka to Delta as fast we we can

- Data pre-populated in Kafka
- Streaming engine
  - Deserializes the JSON string
  - Projects incoming data into strongly typed structure
  - Persists the data in a Delta table
- Run queries against the Delta table



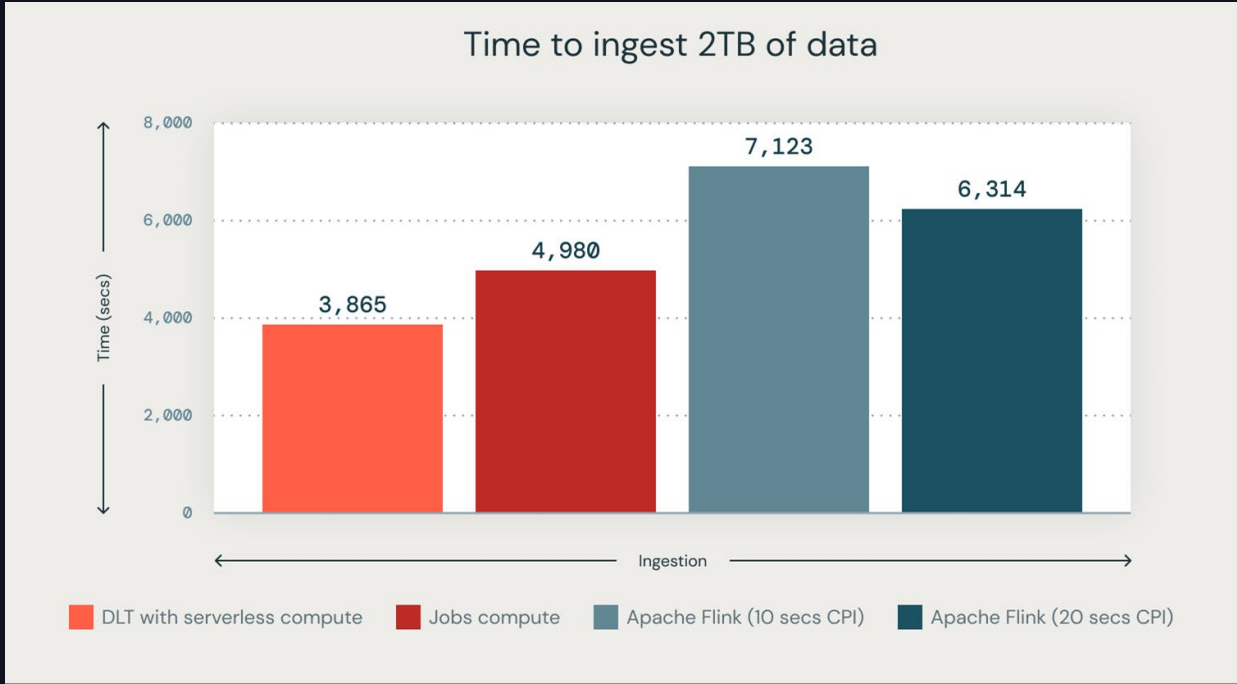
WHAT WE MEASURED?

Ingestion, ingestion with failures, and read query performance

- Ingestion Duration
  - Streaming the data to the Delta table via Kafka
  - Happy path - we assume no node failure occur
  - Failure and Recovery - we measure the impact of a node loss
- Query performance
  - Measure the performance of executing a set of queries on ingested Delta tables

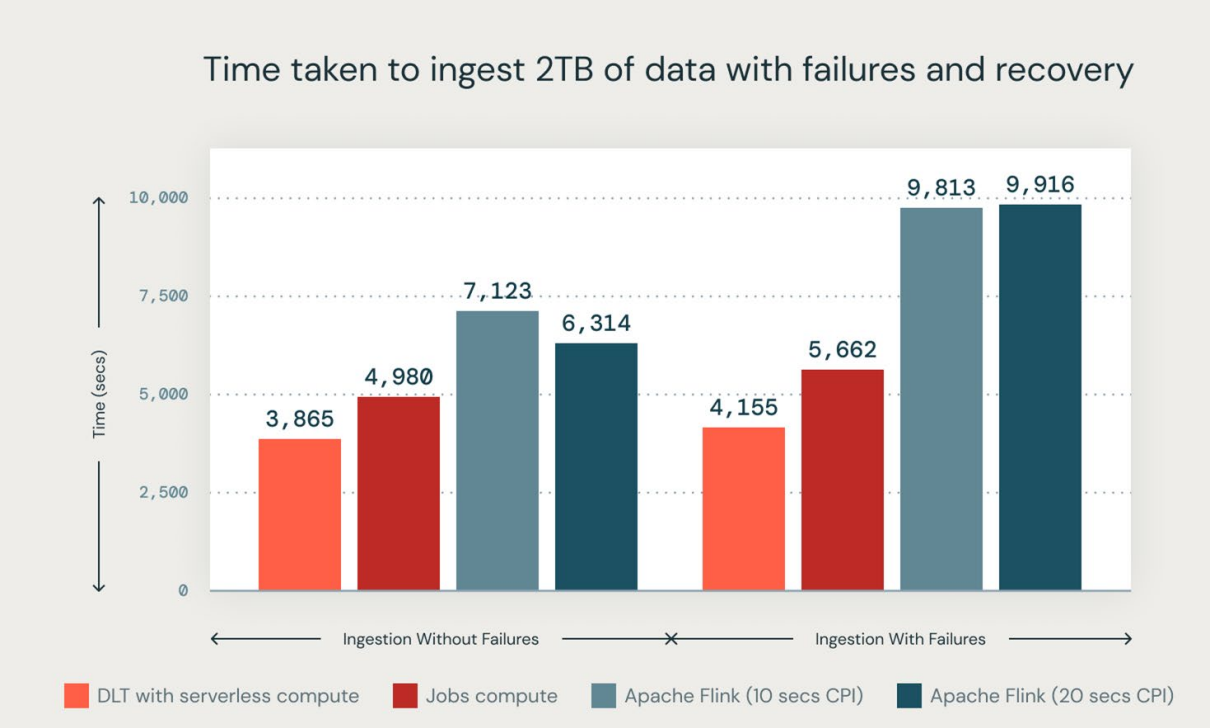
RESULTS - INGESTION DURATION

DLT Serverless is 29% faster then classic compute and 63% faster then Flink



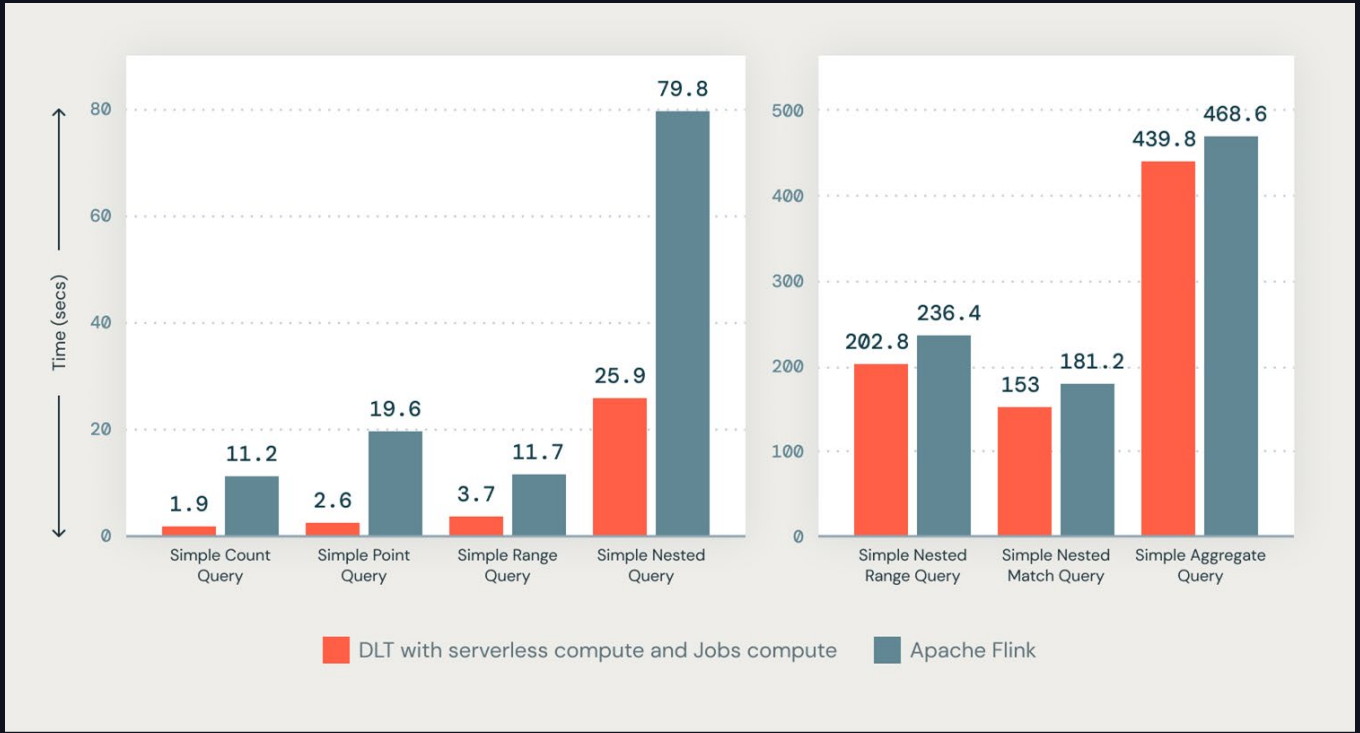
RESULTS - INGESTION WITH FAILURES

Failures increase duration by 8% for DLT, 13% for classic, >35% for Flink



RESULTS - READ PERFORMANCE

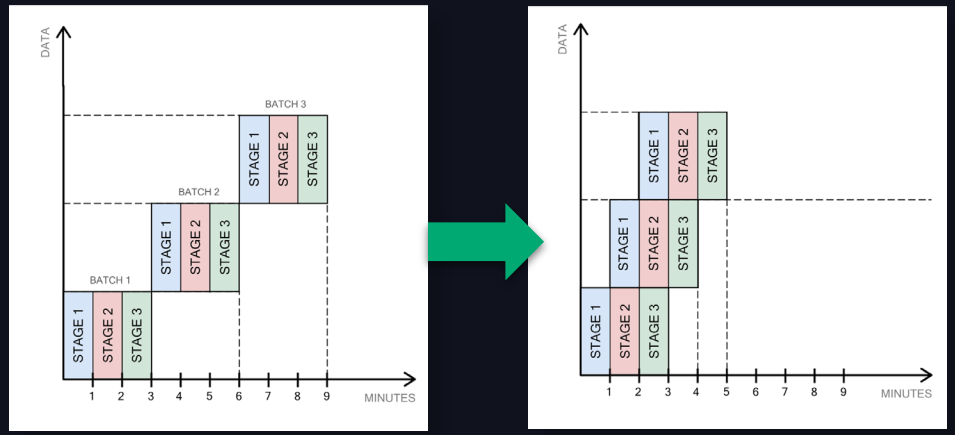
Up to 600% faster read performance with Databricks



STREAM PIPELINING

Up to 3x performance improvement for stateful streaming queries

- Improve performance up to 3x
- Reduces latency by up to 30%
- Works for all streaming queries in serverless



AVAILABLE TODAY IN SERVERLESS

LOW LATENCY OPERATIONAL USE CASES

OPERATIONAL USE CASES

- How many active logins do I have per streaming account?
- Where are my operators of industrial equipment?
- Do I need to send a promotion to a user struggling to purchase?

Business-critical operational use cases need consistently low latency

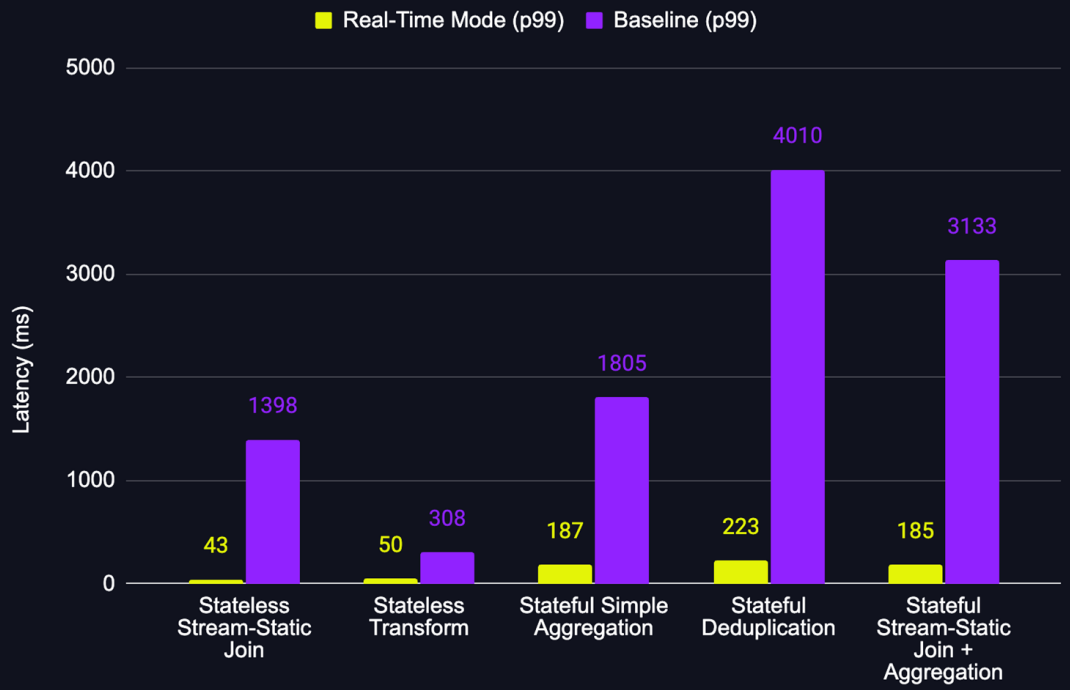
ANNOUNCING REAL-TIME MODE

Real-time mode provides latency in the ms across use cases

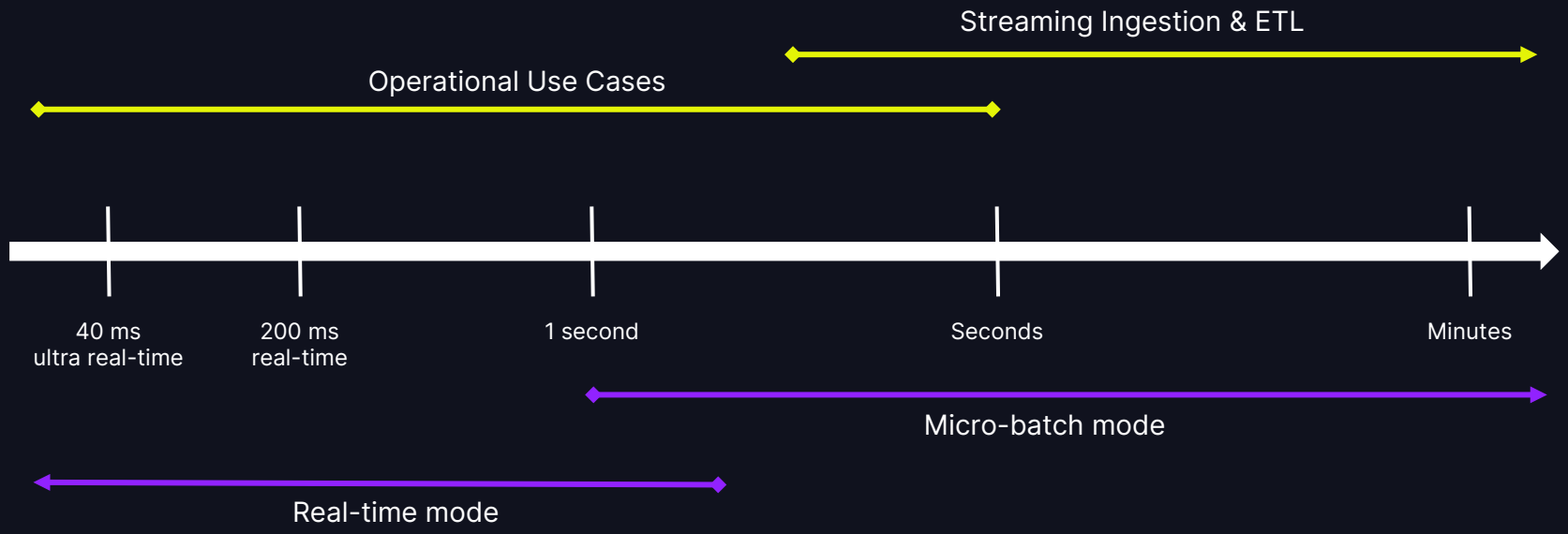
- Real-time mode provides end-to-end processing latencies in the ms for stateful and stateless use cases
- Works with the same familiar Structured Streaming operators
- Real-time mode works by reserving task capacity and enabling results to flow seamlessly between stages and downstream
- **Available in Private Preview in H2 2024**

SO… HOW FAST IS IT?

p50 latency in 10s of ms and p99 latency 100-200ms (goal is <100ms)



STREAMING LATENCY SPECTRUM



PRIVATE PREVIEW IN Q4 2024

CUSTOM PYTHON SOURCES AND SINKS

Python Data Source API

Spark ♥ Python

- SPIP: Python Data Source API (SPARK-44076)
- Fully open source: spark/python/pyspark/datasource.py
- Available in **Spark 4.0 preview version and Databricks Runtime 15.2**
- Support both **batch and streaming, read and write**

PUBLIC PREVIEW AVAILABLE TODAY

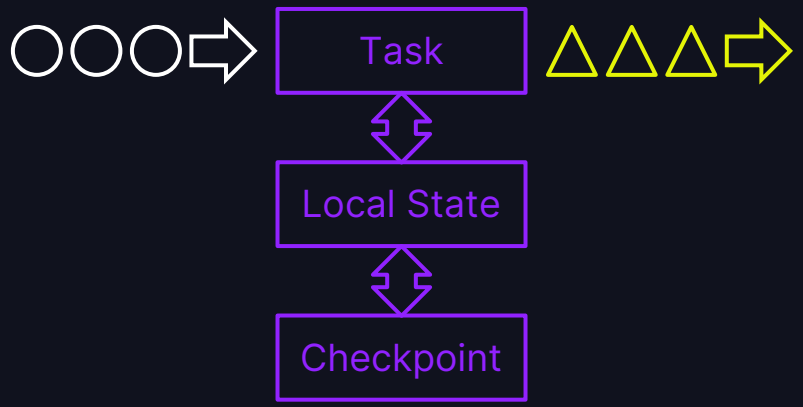
ARBITRARY STATEFUL PROCESSING IN STREAMING

STATEFUL STREAM PROCESSING

Stateful Streaming queries combine multiple records together

State is:

- Information maintained for future use
- Updated by tasks in memory
- Backed up locally (i.e., RocksDB) and distributed storage (i.e., object storage)
- Versioned between micro-batches



DIFFERENT TYPES OF STATEFUL OPERATORS

Structured streaming supports both predefined and arbitrary stateful

Predefined Logic

Built-in operators like:

- Windowed aggregations like tumbling and session
- Joins including stream-static and stream-stream
- Deduplication

Arbitrary Logic

Provide per-key user-defined operators

Specialized APIs like:

- flatMapGroupsWithState
- mapGroupsWithState

(flat)MapGroupsWithState

Powerful arbitrary stateful processing

- Supports a single user defined state object per grouping key
- State object can be updated while evaluating the current group, and updated value will be available in next trigger.

```
val ds =
spark.readStream.json(path).as[CreditCardTransaction]


ds.groupByKey(_.cardId)

 .flatMapGroupsWithState[CreditCardTransactionState,
CreditCardTransaction](

   OutputMode.Append(), GroupStateTimeout.NoTimeout())(

 (_, txns, groupState) => {


   // read state, compute new average, and save to state

   txns.filter(t => t.txAmountDollars > avg).iterator

 }

)
```

PERF AND CAPABILITY LIMITATIONS

Which end up adding a lot of complexity to user code

Lack of Composite Types

Values stored in GroupState are single types and cannot support data structures like List, Map etc efficiently. Current approach requires users to read/update the entire data structure.

Lack of Data Modelling Flexibility

Prevents users from splitting state (for a grouping key) into multiple logical instances, which can be read/updated independently.

No granular eviction control

Does not support state eviction timers or per state variable eviction

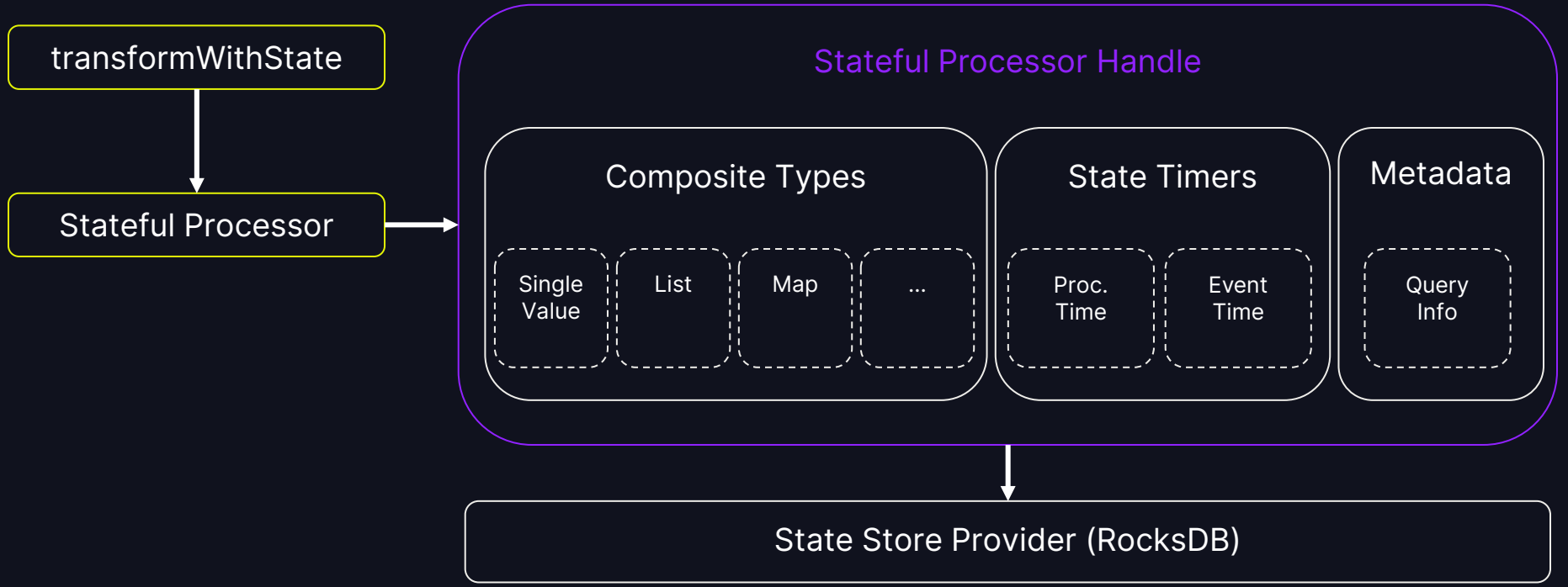
No state schema evolution

Does not support changes to state schema once the streaming query has started.

WE NEED LAYERED, FLEXIBLE, & EXTENSIBLE STATEFUL PROCESSING

INTRODUCING transformWithState

User defined arbitrary stateful processing with a layered state API



SIMPLE CODE EXAMPLE

transformWithState Code Example

```
private var _cardTxnState: MapState[String, String] = _
override def init(outputMode: OutputMode): Unit = {
      _cardTxnState = getHandle.getMapState("cardTxnState", Encoders.STRING)
}

private def isFraud(txn: CreditCardTransaction): Boolean = {
      // check if txn looks suspicious
}

override def handleInputRows(key: String, inputRows: Iterator[CreditCardTransaction], timerValues: TimerValues):
Iterator[CreditCardTransaction] = {
      inputRows.filter(txn =>
            !isFraud(txn)
      )
}
```

COMPOSITE DATA TYPES

State Variables

- ValueState
- ListState
- MapState

Advantages

- Flexibility in Data Modelling
- Optimized read/writes
  - Appends list without read - update - write
  - Efficient key/value access for MapState
- Allows adding & removing state variables for a query with same checkpoint location

STATE TIMERS

Flexible and extensible approach to expiring state

- Support for event-time and processing-time based timers
- Expired timers triggered in the earliest available microbatch
- Timers are checkpointed as part of the store checkpoint
- Multiple timers per grouping key is supported
- Timers are de-duplicated for same timestamp

OPERATOR CHAINING

- Allows running multiple stateful operators inside a single streaming query
- Users can define event time column which will be emitted from the stateful operator, and allow chaining

EVEN MORE FUNCTIONALITY

State Time-to-Live (TTL)

Set state TTL and state will be automatically cleaned up

Define TTL for a specific state variables

Evaluated on read and returned if valid, purged if expired

State Initialization

Initialize state from existing query or Spark Dataset

Enables you to move to the new API and keep existing query state

Schema Evolution

Define their state using Scala case classes, Java POJOs

Modify case class schema (as long as its compatible with with existing schema)

Restart streaming query with same checkpoint location

PUBLIC PREVIEW IN Q3 2024

WHERE ARE WE GOING NEXT?

Project Lightspeed goes… ludicrous speed, plaid?

We listen to our customers, directionally this is what they are telling us.

Release	Performance	Functionality	Connectors
Future!	Catered performance features per use case	State features like repartitioning, checkpoint CRUD APIs Complex event processing Expanding DLT's use cases	Additional sinks for operational use cases Support for multiple sinks / fan out Hybrid batch and data sources

RECOMMENDED STREAMING SESSIONS

Attend in person or view them later online

Session	Date, Time
Your Guide to Data Engineering on the Data Intelligence Platform	Tues, 6/11, 9:00 AM
Delta Live Tables in Depth: Best Practices for Intelligent Data Pipelines	Wed, 6/12, 2:50 PM
Databricks Streaming: Project Lightspeed Goes Hyperspeed	Wed, 6/12, 4:00 PM
Getting Started with DLT Pipelines	Wed, 6/12, 5:10 PM
Introducing Databricks' New Native Ingestion Connectors	Thur, 6/13, 11:20 AM
Streaming Data Pipelines: From Supernovas to LLMs	Thur, 6/13, 12:30 PM
Introducing the New Python Data Source API for Apache Spark	Thur, 6/13, 2:50 PM

Learn more at the summit!

Databricks Events App

Tells us what you think

- We kindly request your valuable feedback on this session.
- Please take a moment to rate and share your thoughts about it.
- You can conveniently provide your feedback and rating through the **Mobile App**.

What to do next?

- Discover more related sessions in the mobile app!
- Visit the Demo Booth: Experience innovation firsthand!
- More Activities: Engage and connect further at the Databricks Zone!

Get trained and certified

- Visit the Learning Hub Experience at Moscone West, 2nd Floor!
- Take complimentary certification at the event; come by the Certified Lounge
- Visit our Databricks Learning website for more training, courses and workshops! databricks.com/learn

DATA+AI SUMMIT