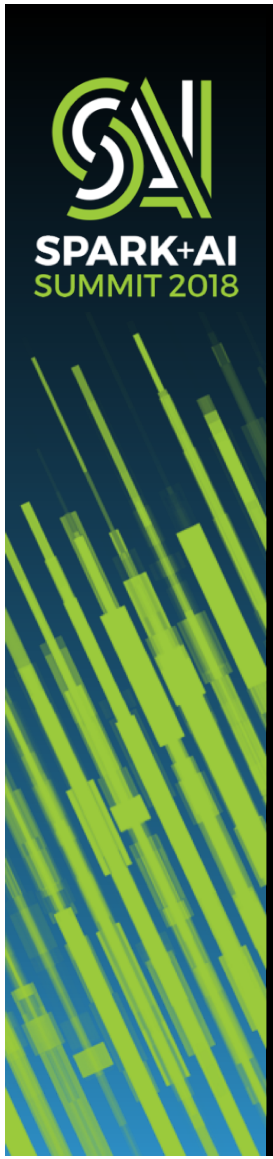


Continuous Processing in Structured Streaming

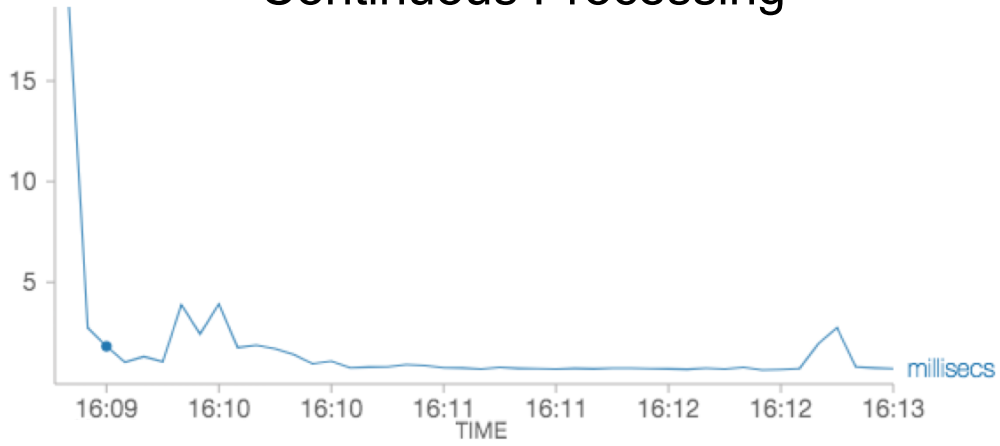
Jose Torres, Databricks

#Dev4SAIS

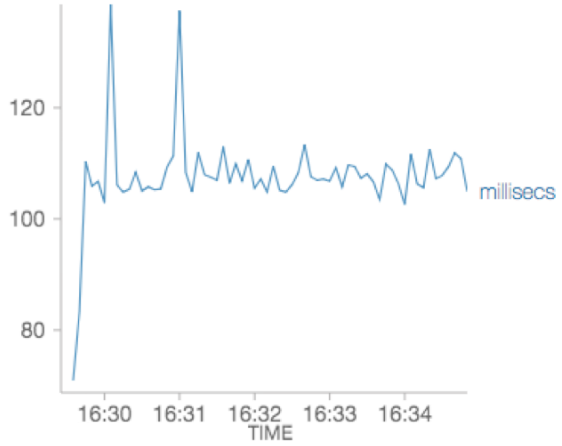
Continuous Processing Overview

- Unified Spark SQL API
- No microbatches
- Low (~1ms) latency

Continuous Processing



Microbatch



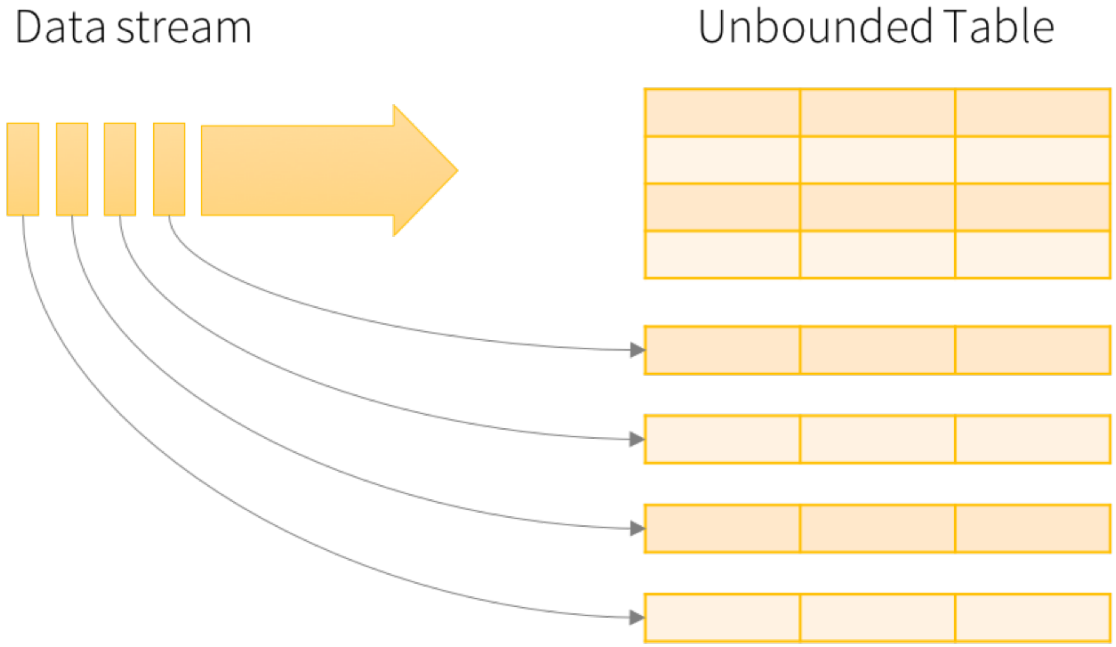
DStream API

- Non-declarative, similar to RDDs
- Scala/Java only
- Checkpoints only through complete snapshots
- No event time

Structured Streaming

- Data represented as a virtual append-only table
- Unified Spark SQL query API
- Batch and streaming queries return same results

Structured Streaming



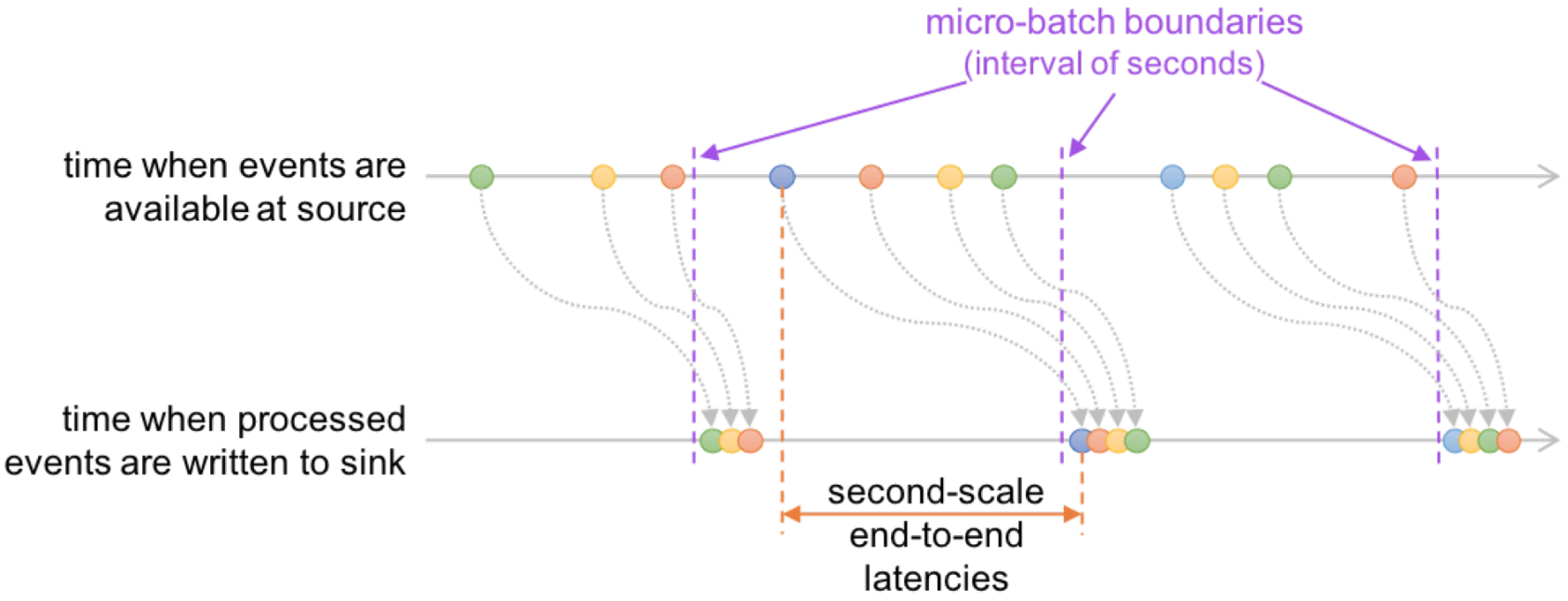
new data in the
data stream
=
new rows appended
to a unbounded table

Data stream as an unbounded table

Structured Streaming Features

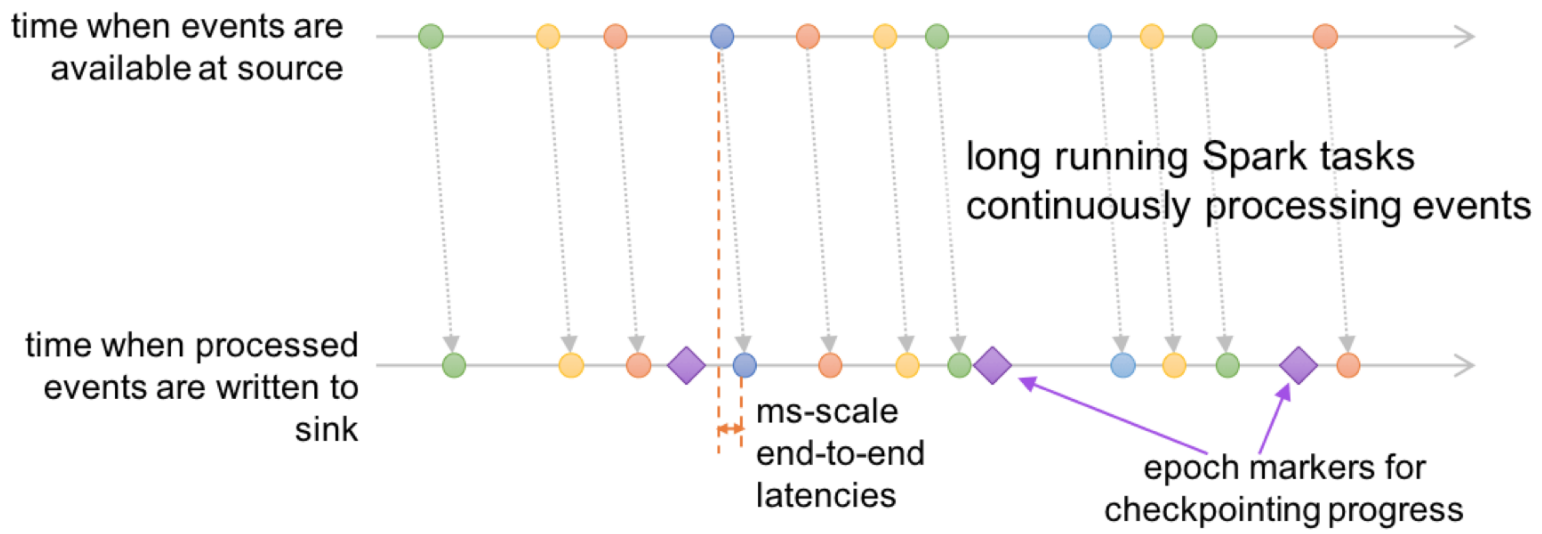
- Dataframes and Datasets
- SQL, Python, and R language APIs
- Delta-based aggregation state

Microbatches



Second-scale end-to-end latencies with Micro-batch Processing

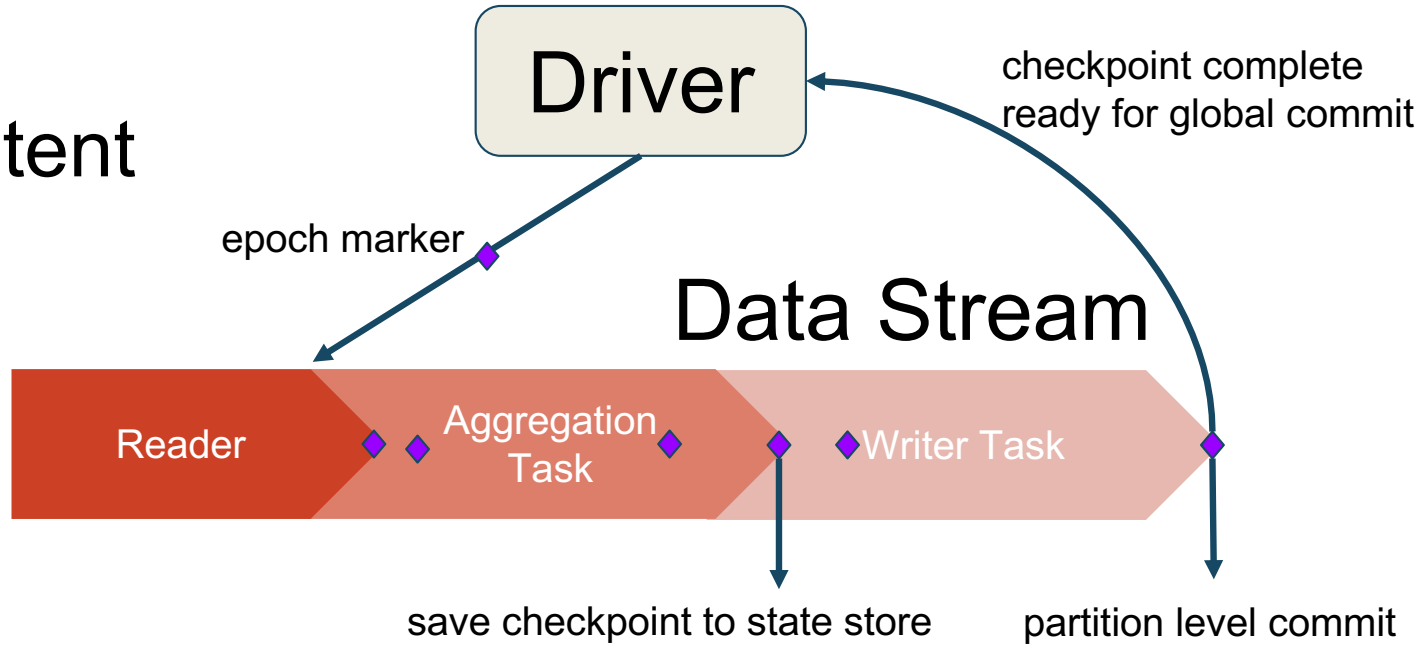
Continuous Processing



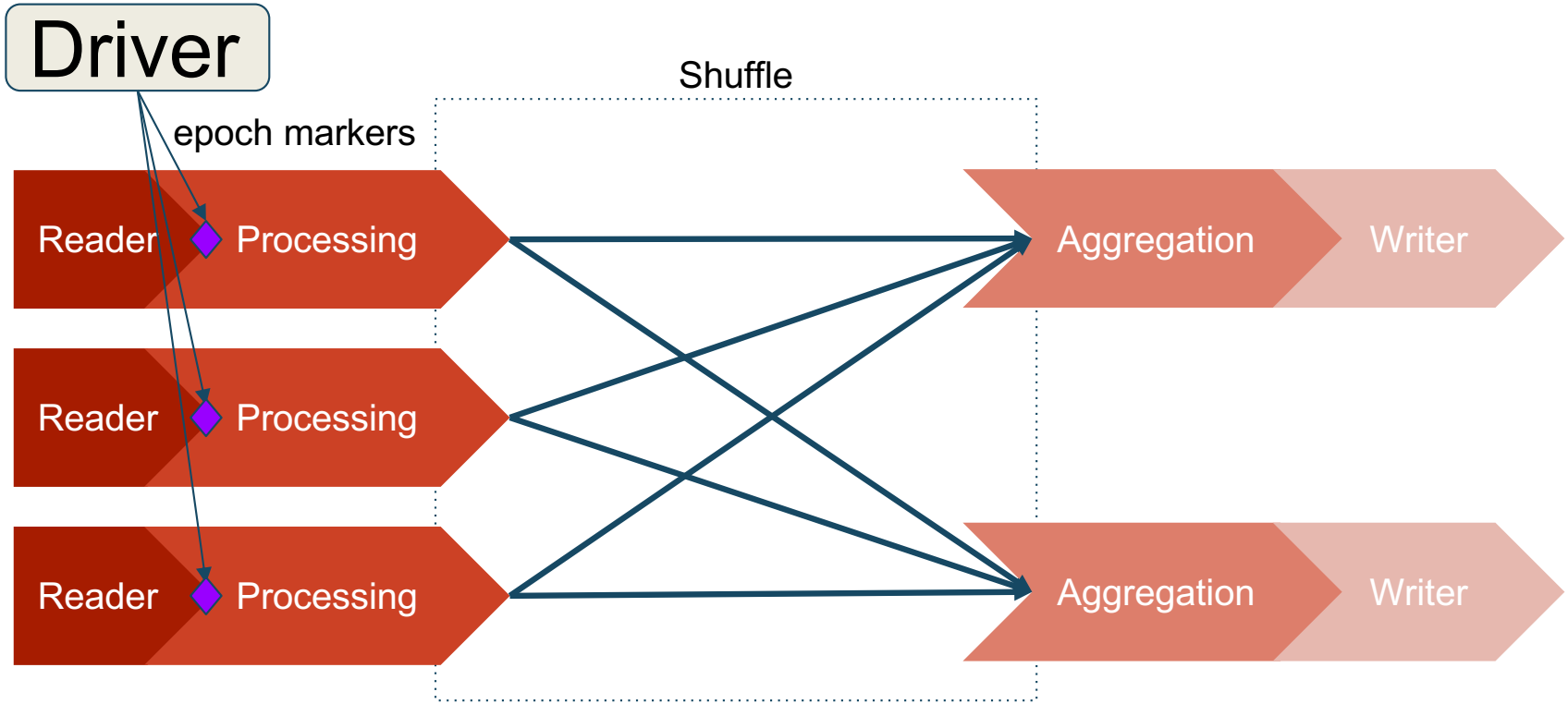
Millisecond-scale end-to-end latencies with Continuous Processing

Chandy-Lamport Checkpoints

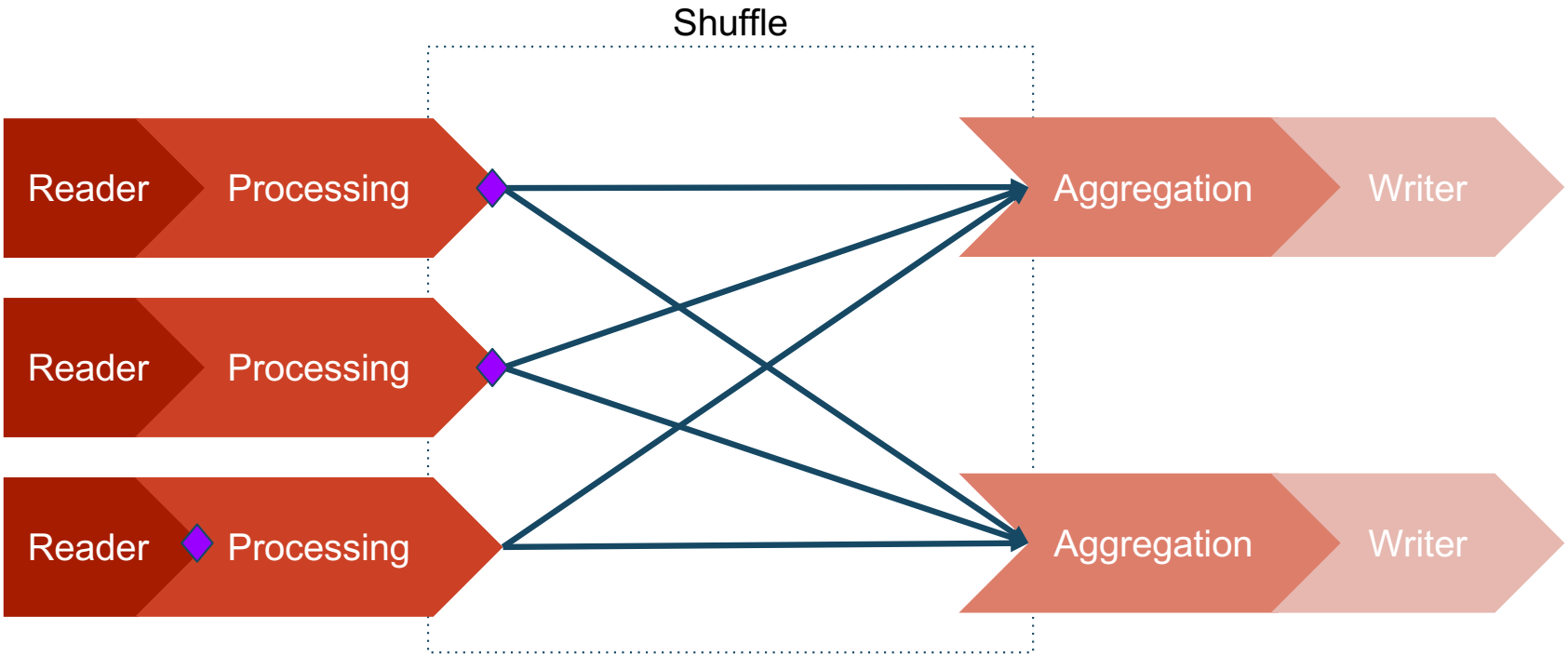
- Asynchronous
- Consistent



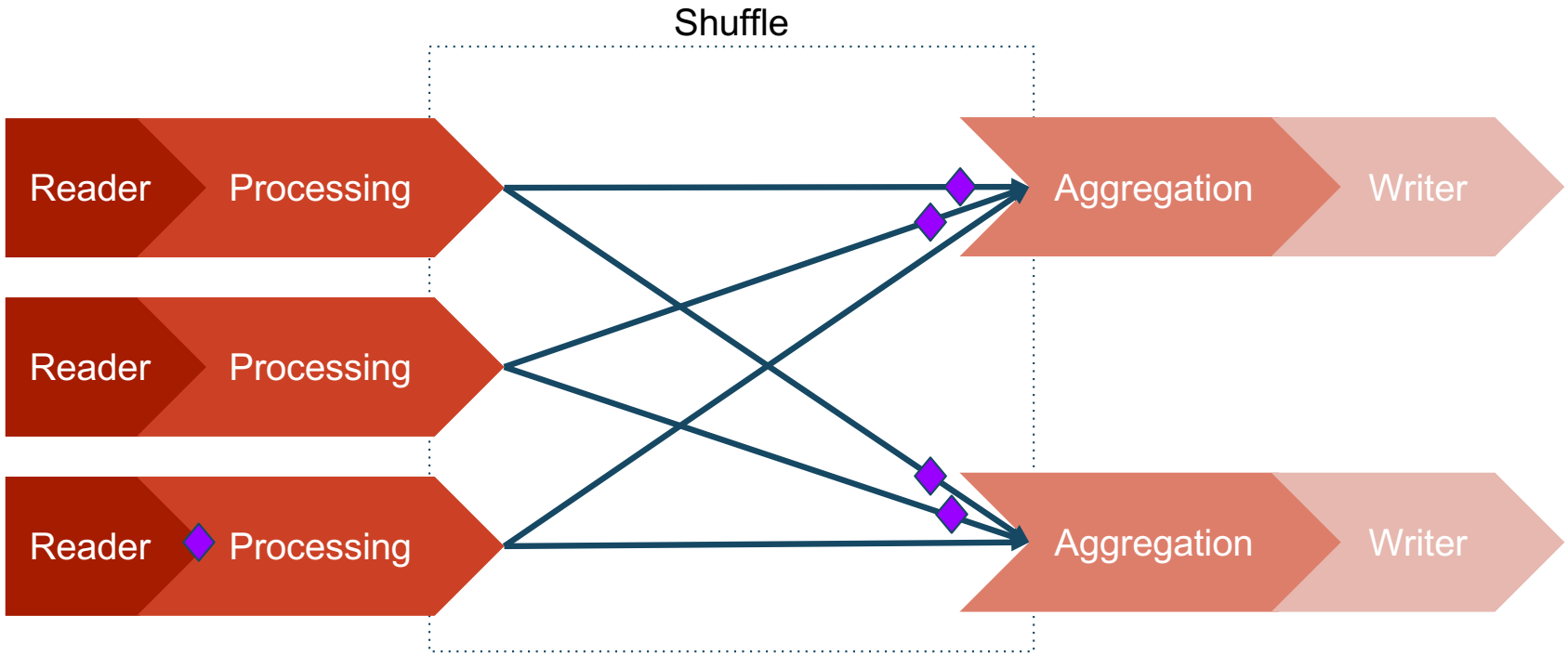
Checkpointing - Detailed



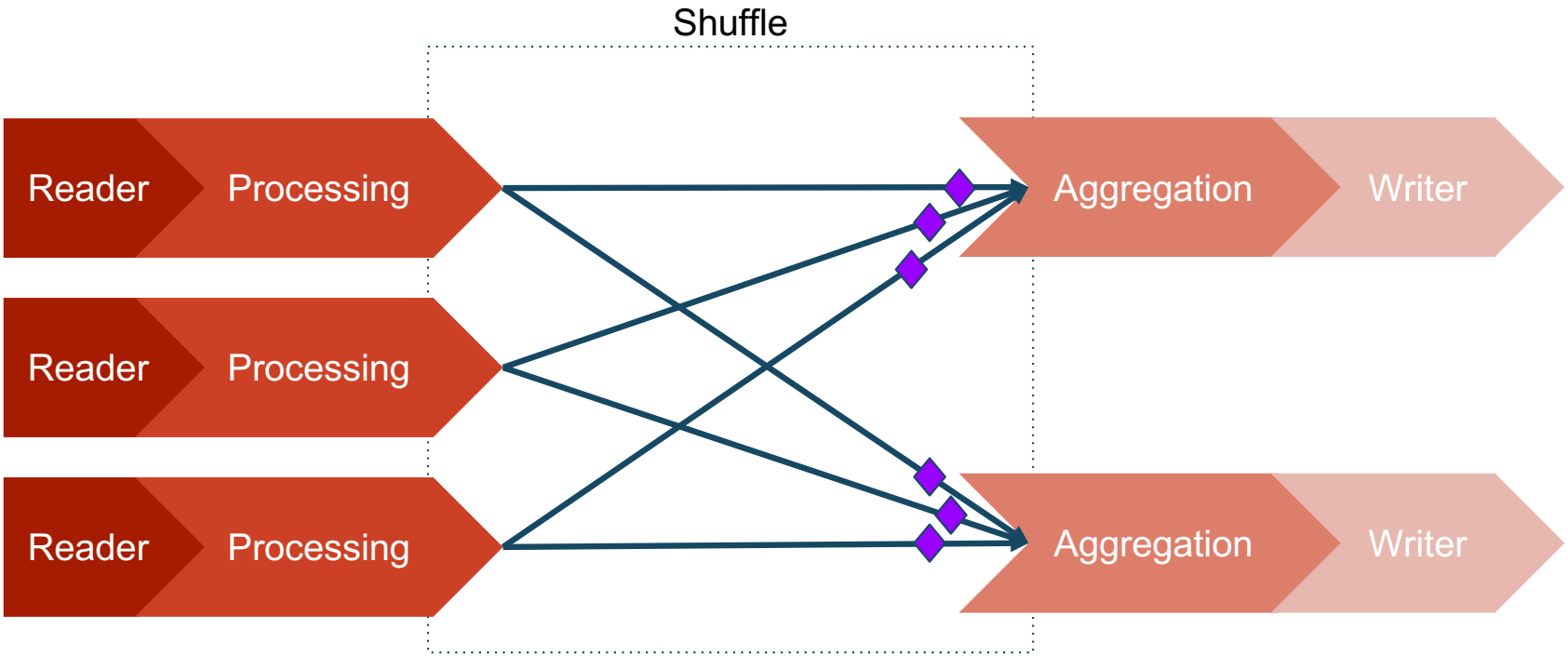
Checkpointing - Detailed



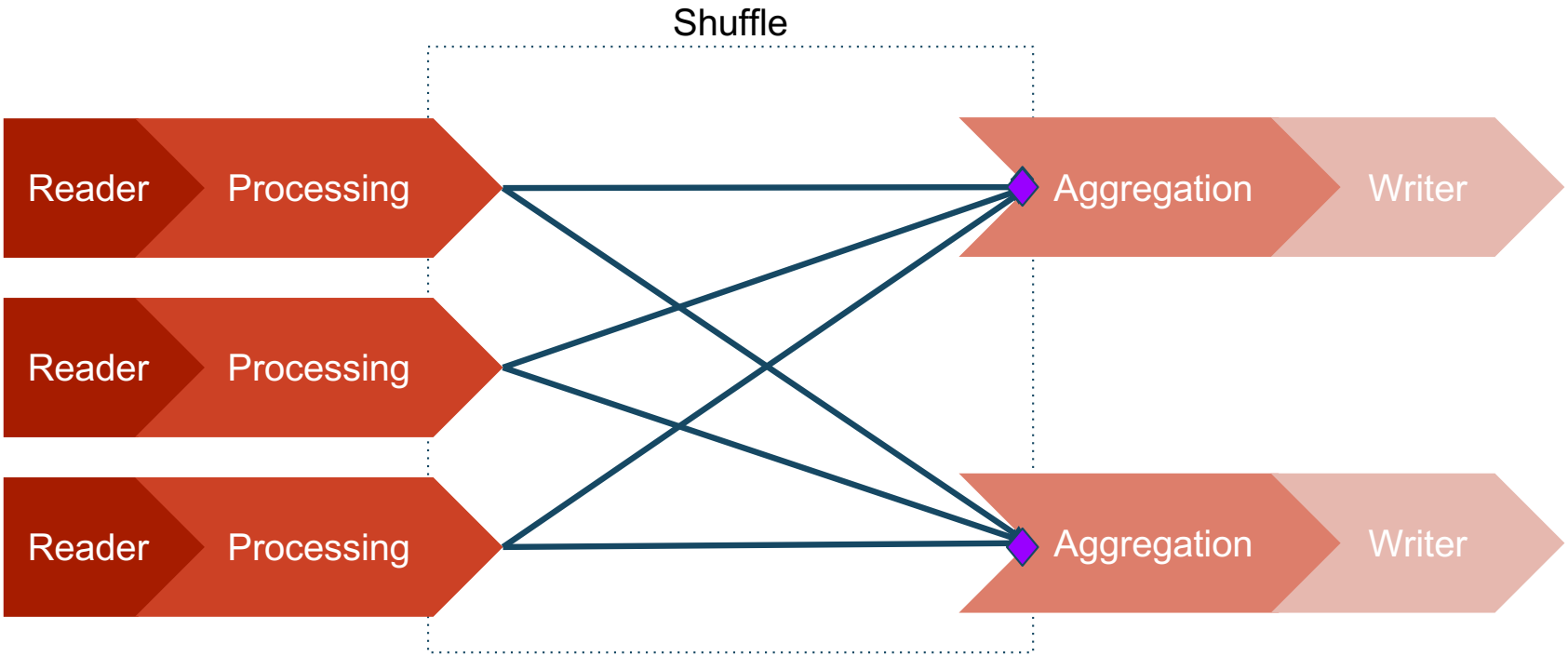
Checkpointing - Detailed



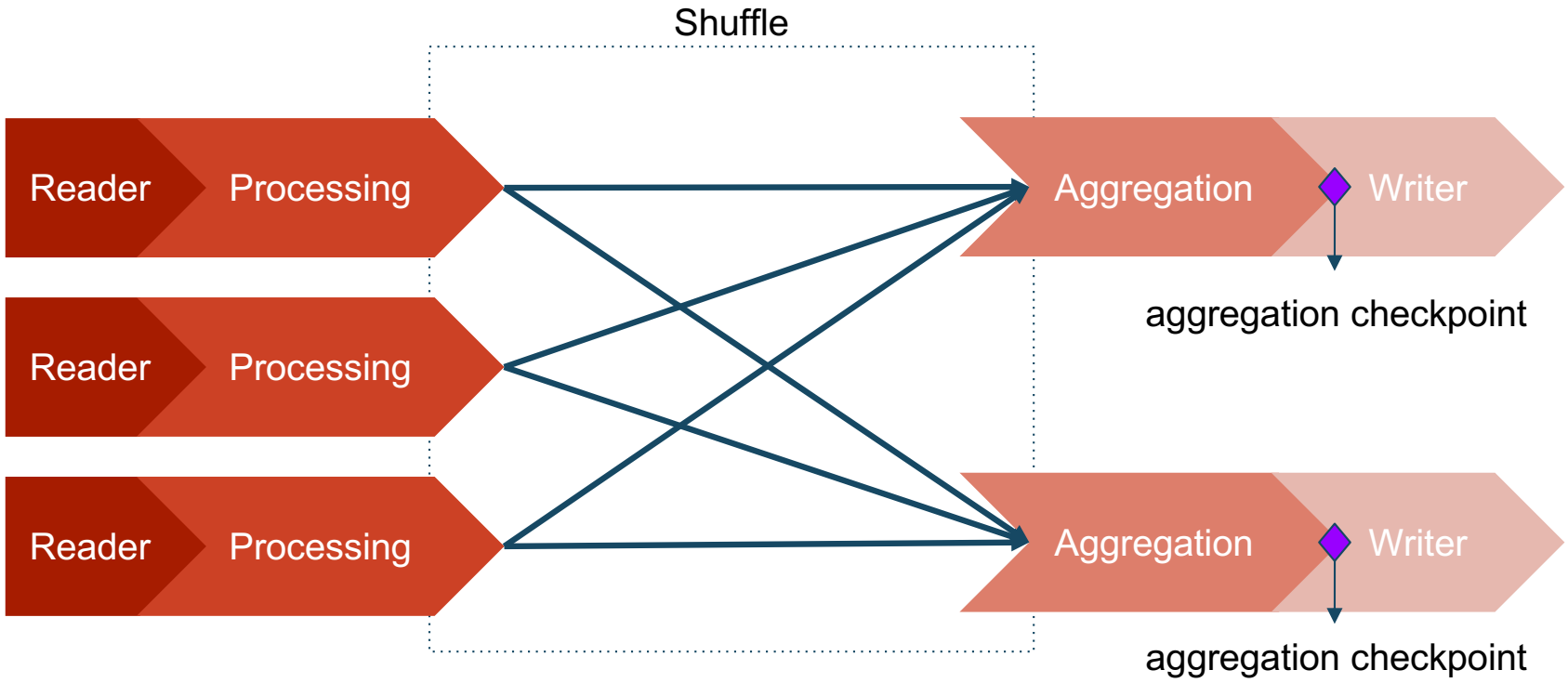
Checkpointing - Detailed



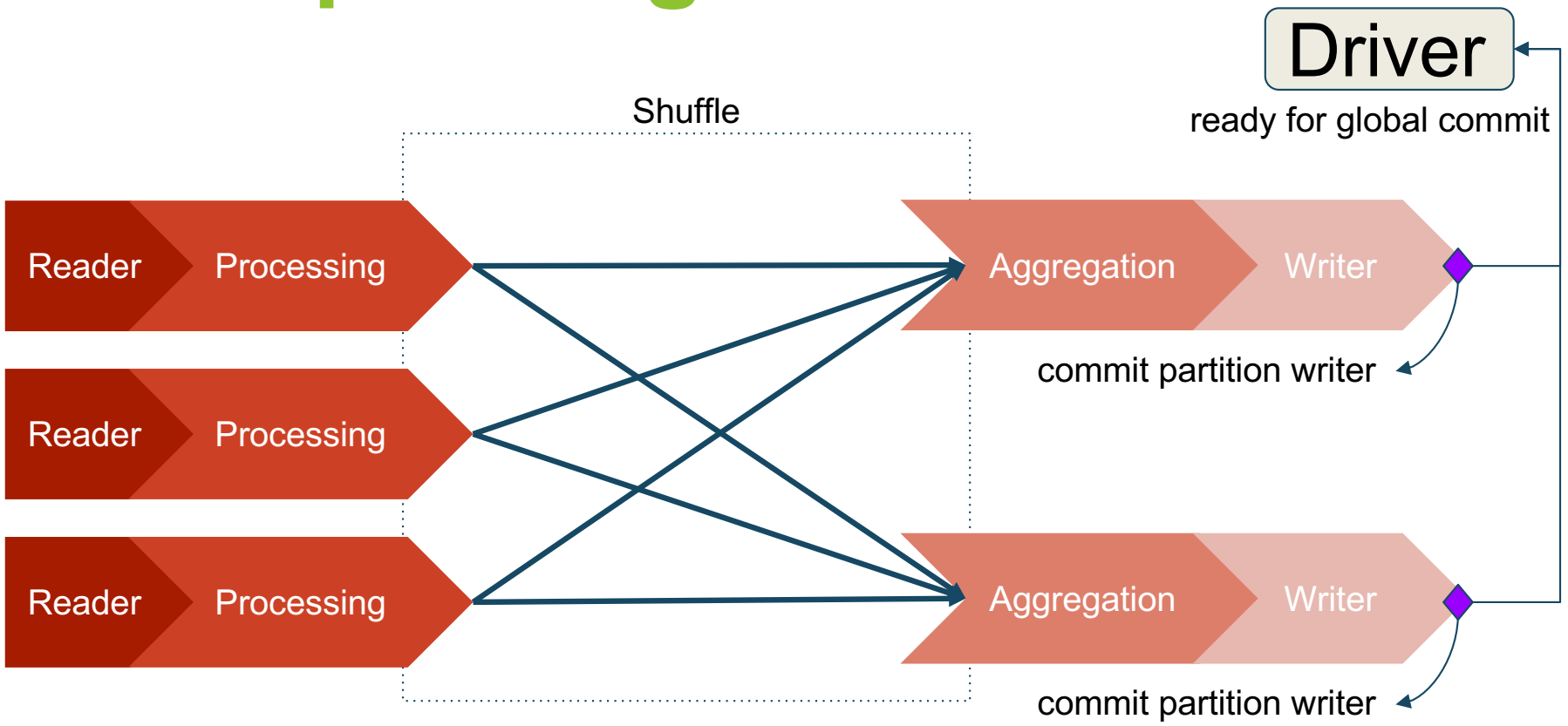
Checkpointing - Detailed



Checkpointing - Detailed



Checkpointing - Detailed



Continuous Processing API

- It's just Structured Streaming
- Run the same queries in continuous mode

Continuous Processing in 2.3

- Initial experimental release
- Supports ETL use cases

Ongoing And Future Work

- Shuffles (SPARK-24036)
- Event time (SPARK-24459)
- Metrics (SPARK-23887)

- Exactly-once semantics mode (SPARK-24460)
- Performance testing (TBD)
- Additional data sources (TBD)

Q&A