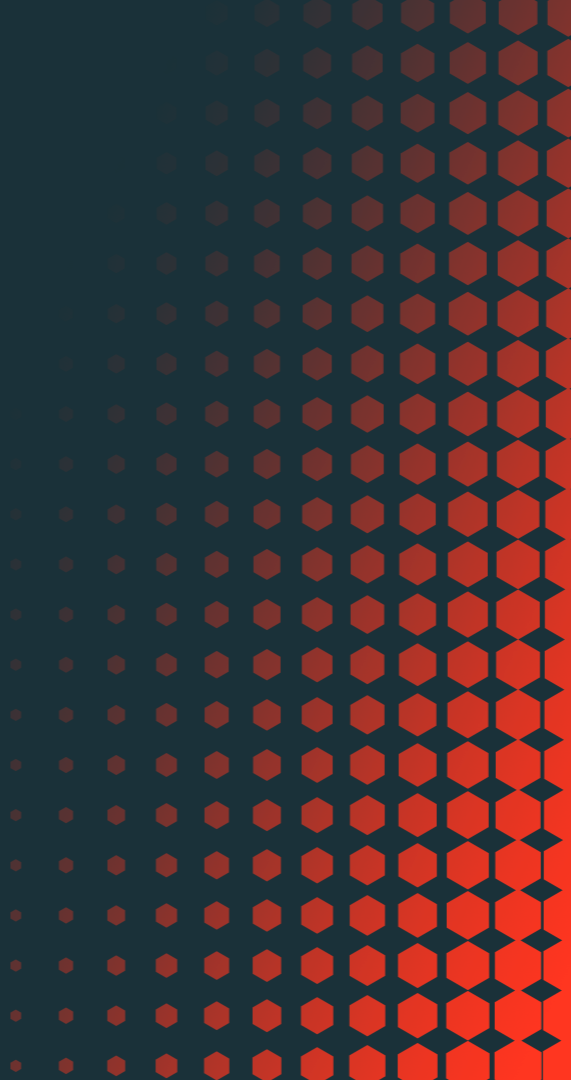
Photon Technical Overview



databricks

Agenda

- Motivation/Background on Photon
  - Performance
  - Using the feature
  - Supported Functionalities

Photon is Databricks' next generation query engine, built to be faster than anyone else

100% APACHE SPARK COMPATIBLE query engine

Built from the ground up to deliver the FASTEST PERFORMANCE

FOR ALL DATA USE CASES*
across data engineering, data science, machine learning, and data analytics.

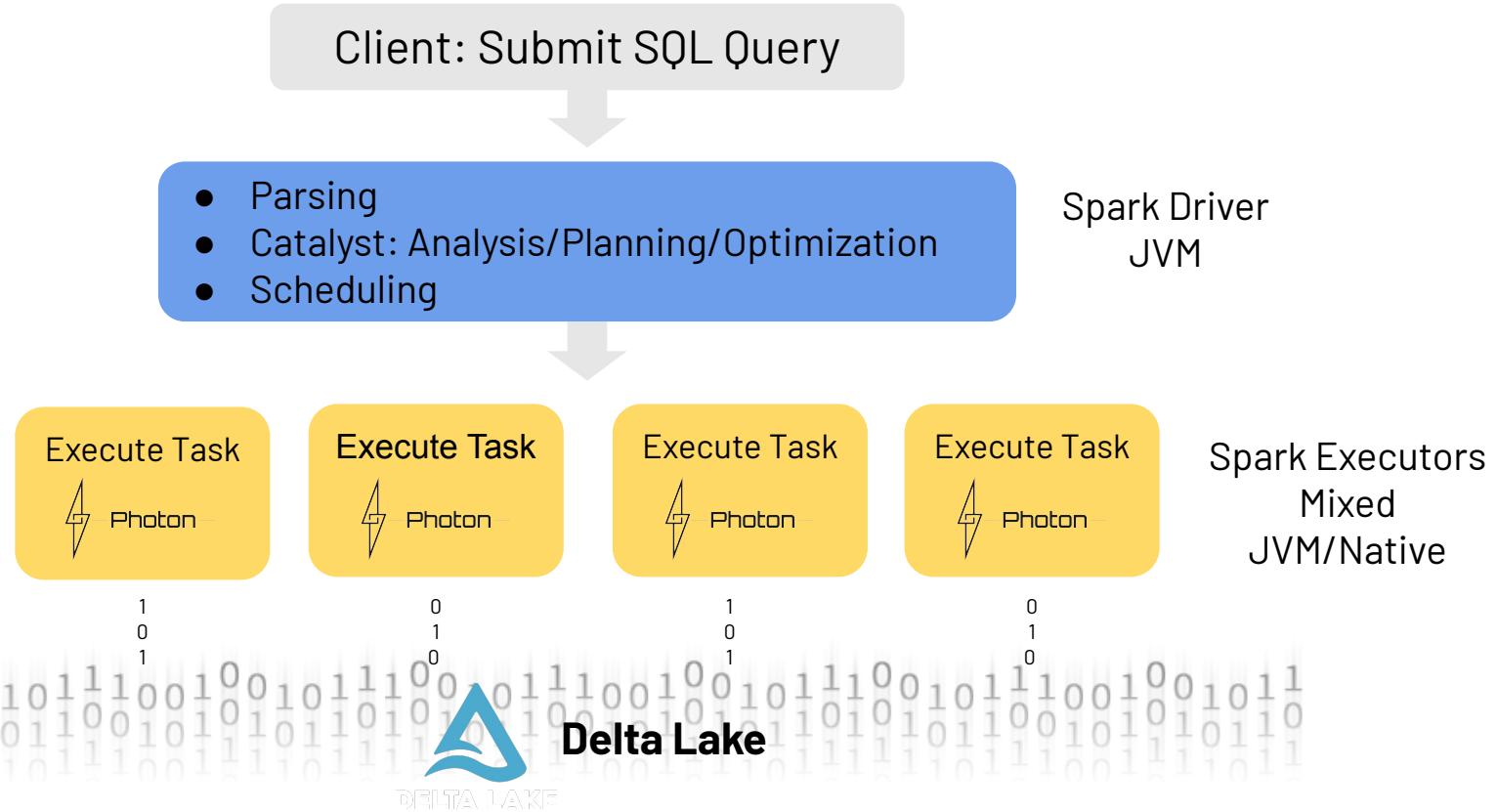
(*) Note: For preview, only SQL / Koalas / DataFrame use cases are supported


Photon

Building the next generation query engine

- Re-architected for the fastest performance on real-world applications
  - Native C++ engine for faster queries
  - Custom built memory management to avoid JVM bottlenecks
  - Vectorized: memory, instruction, and data parallelism (SIMD)
- Works with your existing code and avoids vendor lock-in
  - 100% compatible with open source Spark DataFrame APIs and Spark SQL
  - Transparent operation to users - no need to invoke something new, it just works
- Optimizing for all data use cases and workloads
  - Today, supporting SQL and DataFrame workloads
  - Coming soon, Streaming, Data Science, and more

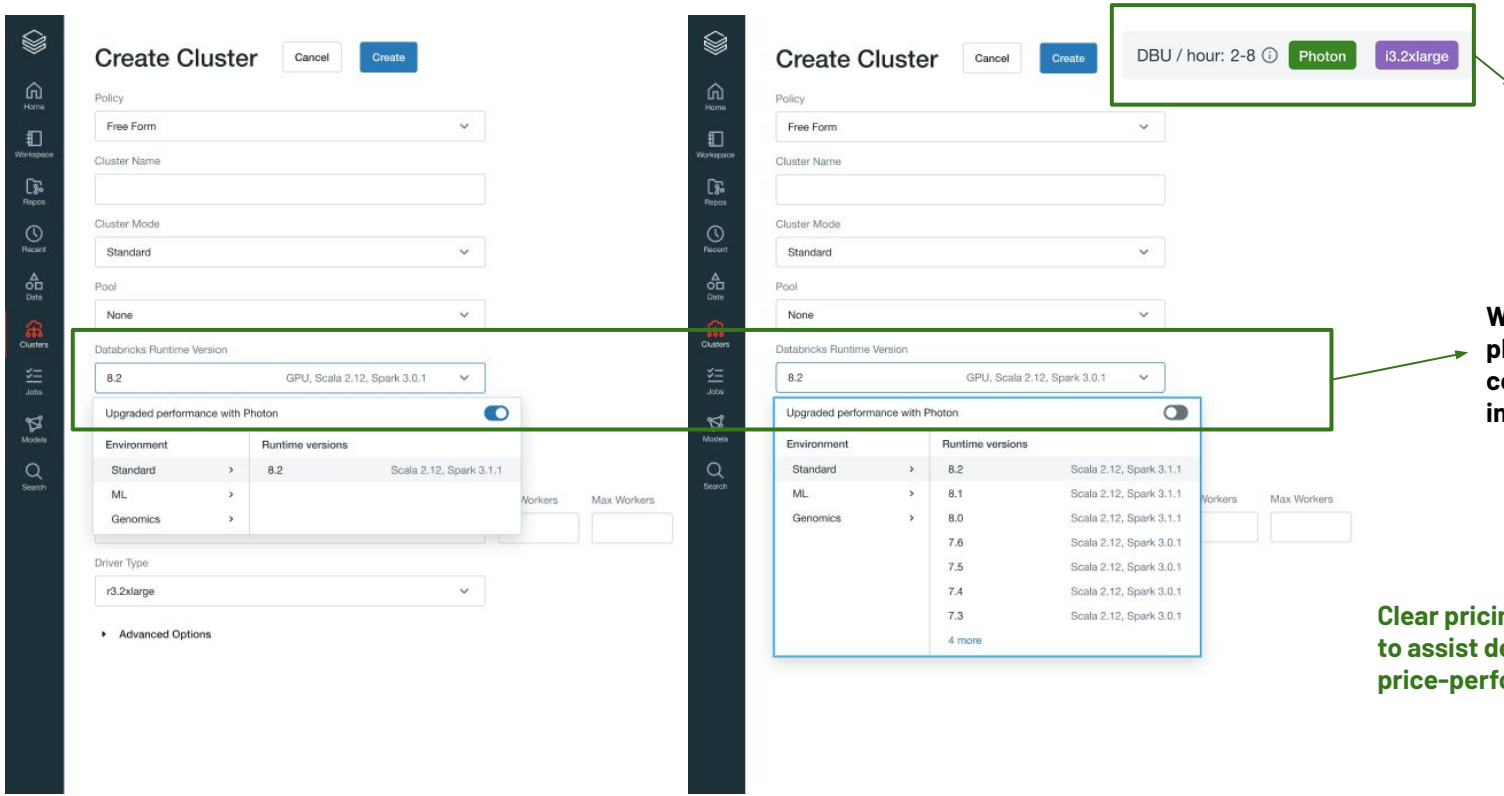
Photon in the Databricks Lakehouse Platform



databricks

Create a Cluster with Photon Enabled

Create clusters with the flexibility to choose photon runtime or non-photon runtime



DBU count now driven by 2 factors: the photon runtime version vs. non-photon version, and size of the VMs.

When customers choose photon runtime, the DBU count for this cluster increases

Clear pricing indication presented to assist decision making for price-performance trade-off.

Ideal Job Profile

- Azure Instances
  - D (general), E (memory), L (storage) w/ or w/o Delta Cache Acceleration
  - Additive features: (blog on L8as_v3 w/ Photon)
    - s: premium storage
    - a: AMD-based processor
- Longest running jobs (at least 10 min)
  - Focus on the longest running / most expensive jobs to make the biggest impact
- Batch Job (not streaming)
- No UDFs
- CPU bound
- Reads and Writes to/from Delta

Overview of Query Coverage

Data Types

- ✅ Byte/Short/Int/Long
- ✅ Boolean
- ✅ String/Binary
- ✅ Decimal
- ✅ Float/Double
- ✅ Date/Timestamp
- ✅ Struct
- ✅ Array, Map

Operators

- ✅ Scan, Filter, Project
- ✅ Hash Aggregate/Join/Shuffle
- ✅ Nested-Loop Join
- ✅ Null-Aware Anti Join
- ✅ Union, Expand, ScalarSubquery

Coming soon: Sort, Window

Expressions

- ✅ Comparison / Logic
- ✅ Arithmetic / Math (most)
- ✅ Conditional (IF, CASE, etc.)
- ✅ String (common ones)
- ✅ Casts
- ✅ Aggregates (most common ones)
- ✅ Date/Timestamp (in progress)

Coming soon: UDFs, long tail

Key Photon Characteristics

- Hybrid Photon/Spark Plans
  - Use Photon when possible, fall back to Spark for unsupported operations
- Native code using off-heap memory
- Fully integrated with Spark's memory manager
- Prefers hash join over sort-merge join
- Rich per-operator performance metrics