

What's New in Apache Spark 2.3

Sameer Agarwal

Spark Summit | San Francisco | June 6th 2018

[#DevSAIS16](#)

About Me

- Spark Committer and 2.3 Release Manager
- Software Engineer at Facebook (Big Compute)
- Previously at Databricks and UC Berkeley
- Research on BlinkDB (Approximate Queries in Spark)



Spark 2.3 Release by the numbers

- Released on 28th February 2018
- Development Span: July '17 – Feb '18
- 284 Contributors
- 1406 JIRAs
 - SQL/Streaming (52%)
 - Spark Core (12%)
 - PySpark (9%)
 - ML (8%)

Overview



Continuous
Processing



ML Streaming
+
Image Reader



PySpark
Performance



Spark on
Kubernetes

Major Features in Spark 2.3



Continuous
Processing



Data
Source
API V2



Spark on
Kubernetes



PySpark
Performance



ML on
Streaming



History
Server V2



Stream-stream
Join



UDF
Enhancements



Image
Reader



Native ORC
Support



Stable
Codegen



Various SQL
Features

<https://spark.apache.org/releases/spark-release-2-3-0.html>

Overview



Continuous
Processing



ML Streaming
+
Image Reader



PySpark
Performance



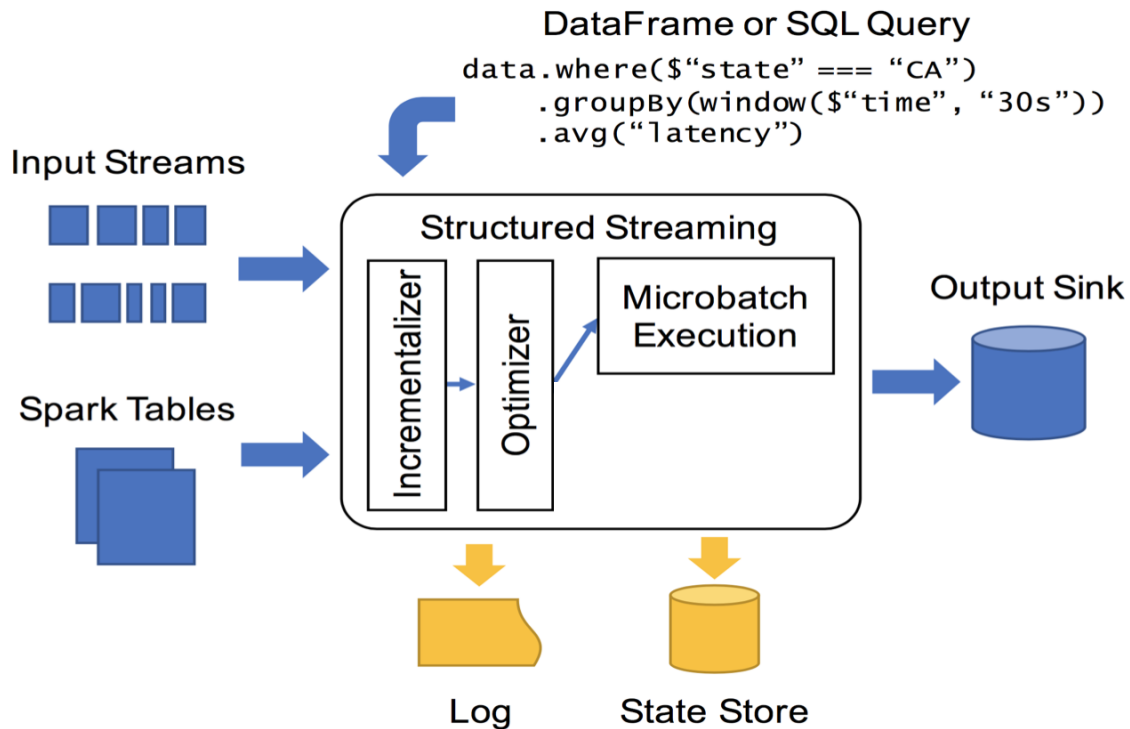
Spark on
Kubernetes

Structured Streaming

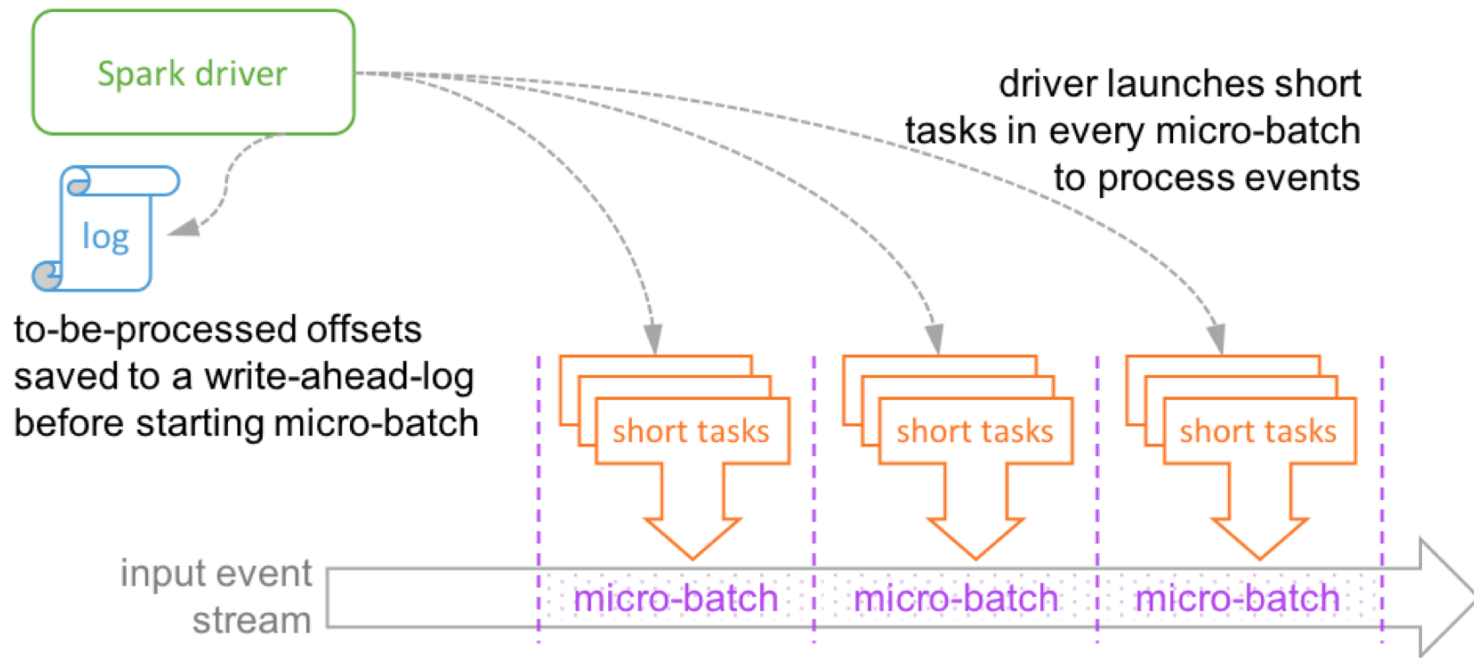
Users: Treat a stream as an infinite table, no need to reason about micro-batches

Developers: Decoupled the high-level API with the execution engine

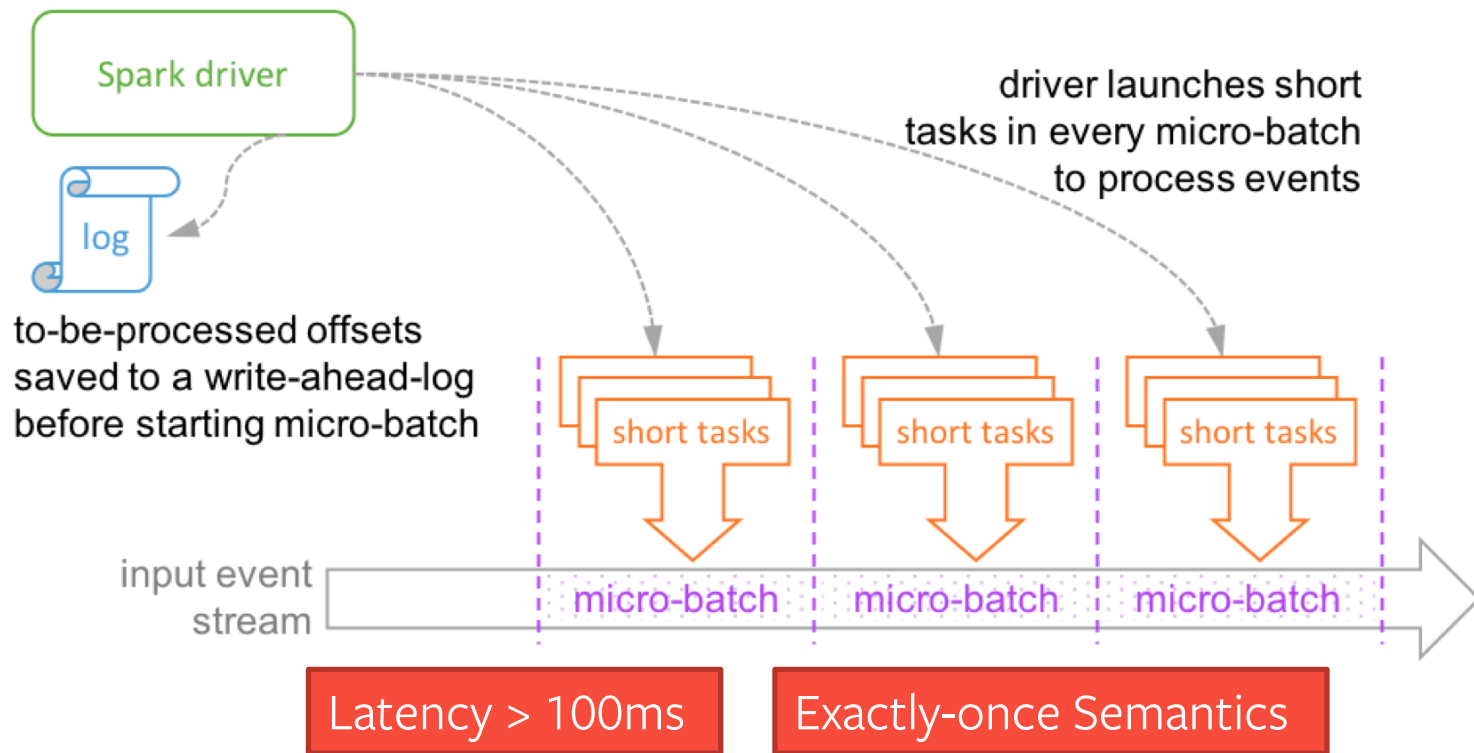
Structured Streaming



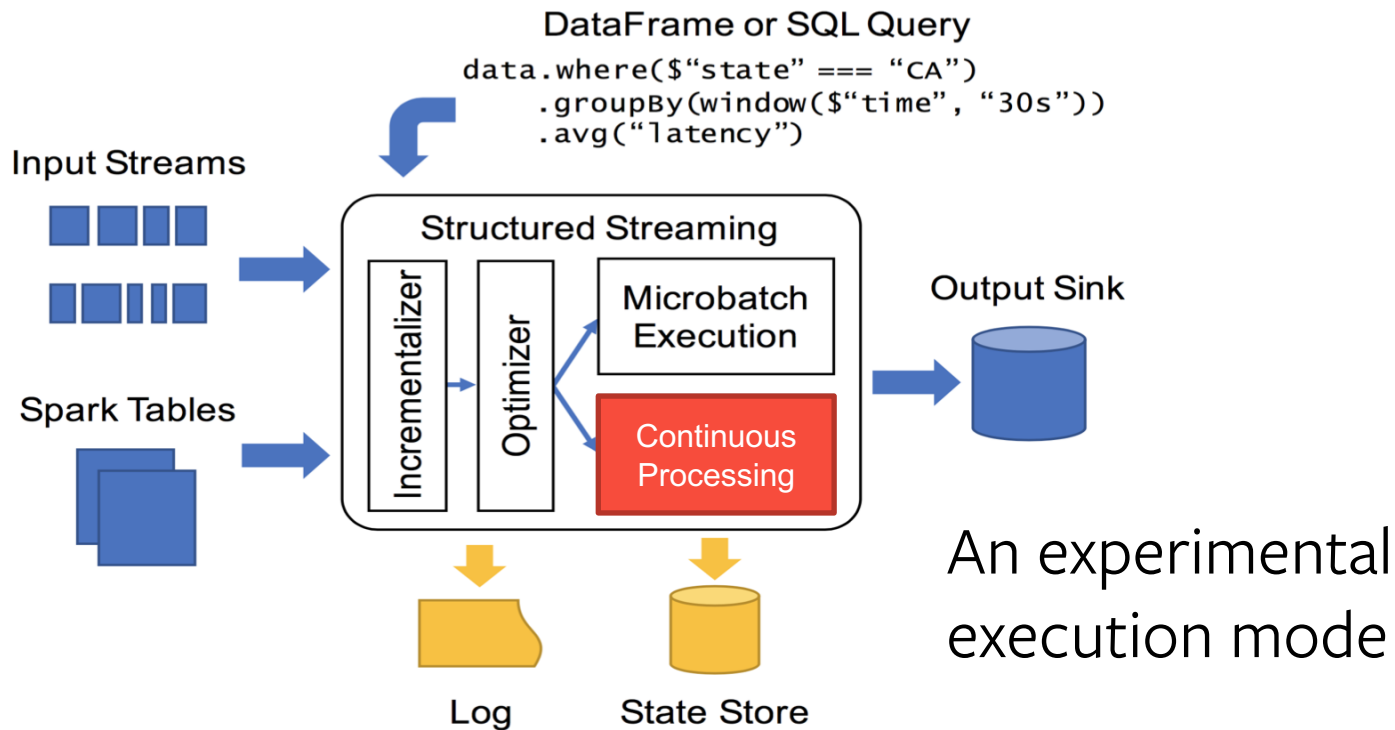
Micro Batch Execution



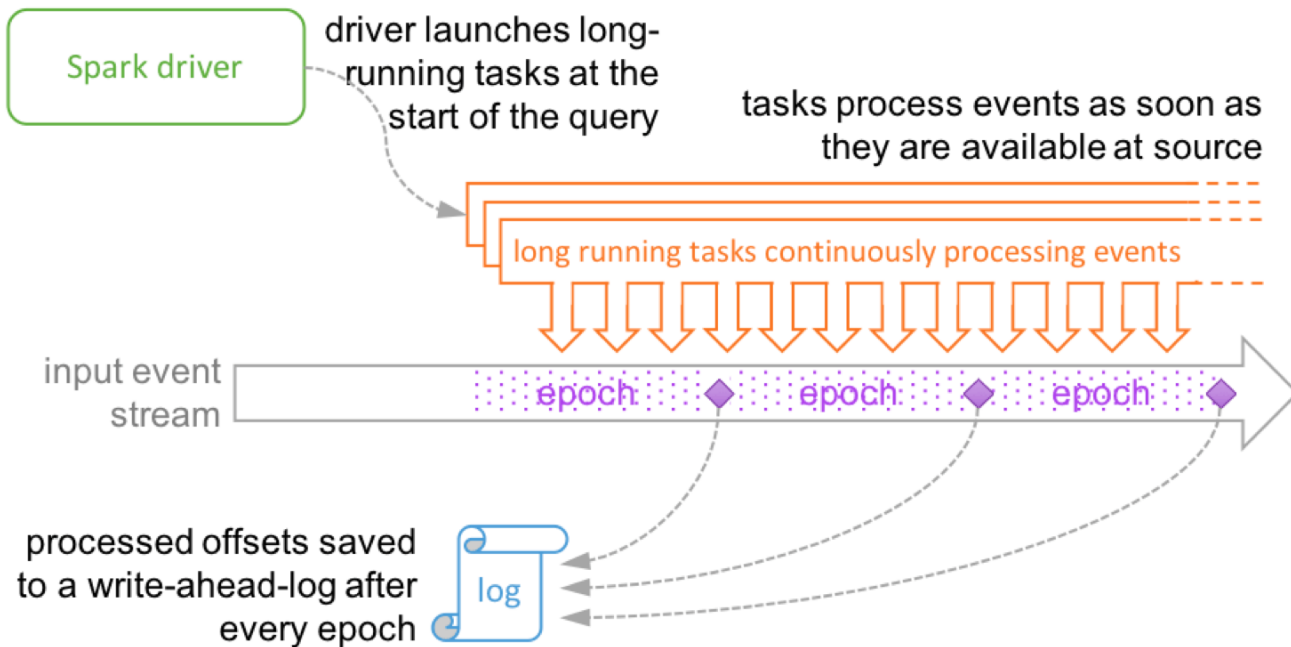
Micro Batch Execution



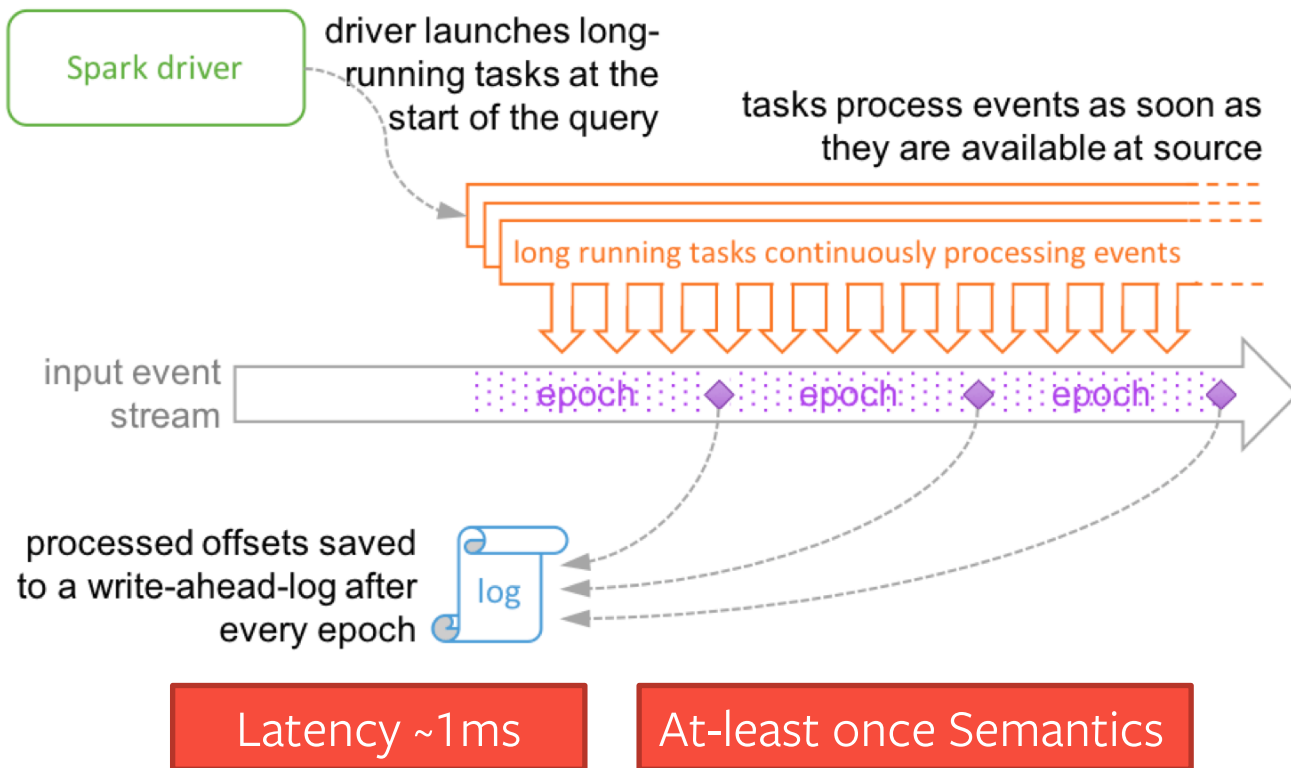
Continuous Processing (SPARK-20928)



Continuous Processing (SPARK-20928)



Continuous Processing (SPARK-20928)



Continuous Processing (SPARK-20928)

```
spark
  .readStream
  .format( source = "kafka")
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")
  .option("subscribe", "topic1")
  .load()
  .selectExpr( exprs = "CAST(key AS STRING)", "CAST(value AS STRING)")
  .writeStream
  .format( source = "kafka")
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")
  .option("topic", "topic1")
  .trigger(Trigger.Continuous( interval = "1 second")) // only change in query
  .start()
```

Continuous Processing (SPARK-20928)

Supported Operations

- Map-like Dataset Operations
 - Projections
 - Selections
- All SQL functions
 - Except `current_timestamp()`, `current_date()` and aggregation functions

Supported Sources

- Kafka Source
- Rate Source

Supported Sinks

- Kafka Sink
- Memory Sink
- Console Sink

Blog: <https://tinyurl.com/spark-cp>

Overview



Continuous
Processing



ML Streaming
+
Image Reader



PySpark
Performance



Spark on
Kubernetes

ML on Streaming

- Model transformation/prediction on batch and streaming data with unified API
- After fitting a model or Pipeline, you can deploy it in a streaming job

```
val streamOutput = transformer.transform(streamDF)
```

Image Support in Spark (SPARK-21866)

- A standard API in Spark for reading images into DataFrames
- Utilities for loading images from common formats
- Deep learning frameworks can rely on this

```
val df = ImageSchema.readImages("/data/images")
```

```
root
├── image: struct (nullable = true)
│   ├── origin: string (nullable = true)
│   ├── height: integer (nullable = false)
│   ├── width: integer (nullable = false)
│   ├── nChannels: integer (nullable = false)
│   ├── mode: string (nullable = false)
│   └── data: binary (nullable = false)
```

Overview



Continuous
Processing



ML Streaming
+
Image Reader



PySpark
Performance

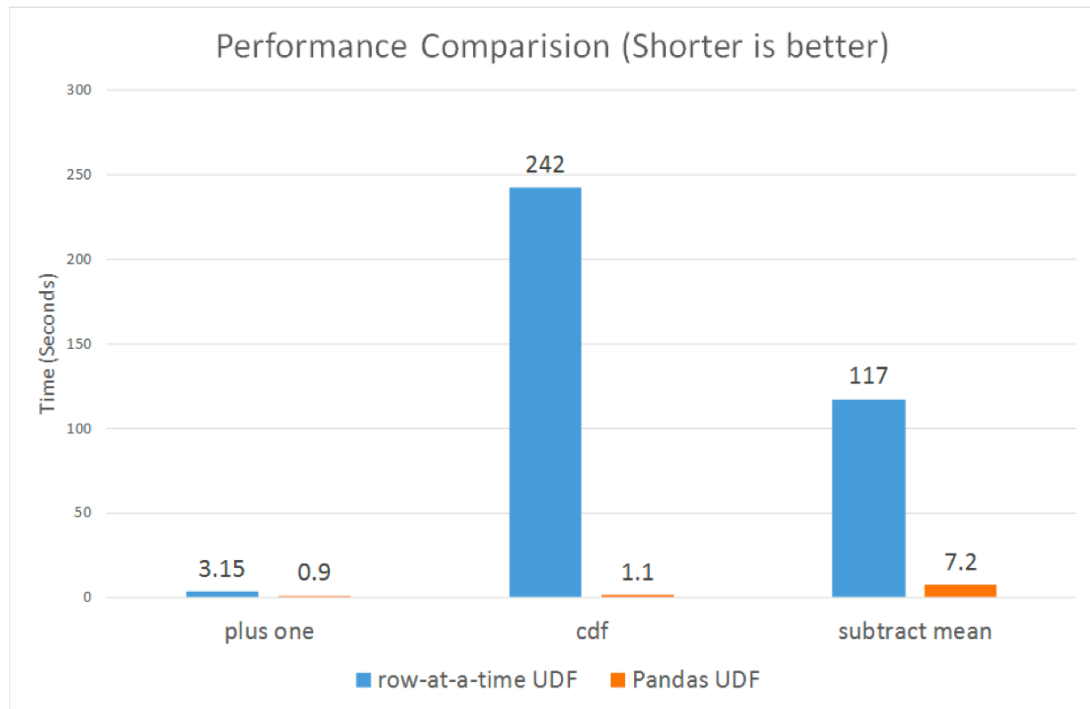


Spark on
Kubernetes

PySpark

- Introduced in Spark 0.7 (~2013); became first class citizen in the Dataframe API in Spark 1.3 (~2015)
- Much slower than Scala/Java with UDFs due to serialization and Python interpreter
- Note: Most PyData tooling (e.g., Pandas, numpy etc.) are written in C/C++

PySpark Performance



Pandas UDFs perform much better than row-at-a-time UDFs across the board, ranging from **3x to over 100x**.

Pandas/Vectorized UDFs

Scalar UDFs

- Used with functions such as `select` and `withColumn`
- The python function should take `pandas.Series` as input and return a `pandas.Series` of same length

```
from pyspark.sql.functions import pandas_udf, PandasUDFType

# Use pandas_udf to define a Pandas UDF
@pandas_udf('double', PandasUDFType.SCALAR)
# Input/output are both a pandas.Series of doubles

def pandas_plus_one(v):
    return v + 1

df.withColumn('v2', pandas_plus_one(df.v))
```

Pandas/Vectorized UDFs

Grouped Map UDFs

- Split-apply-Combine
- A python function that defines the computation for each group
- Input/Outputs are both `pandas.DataFrame`

```
@pandas_udf(df.schema, PandasUDFType.GROUPED_MAP)
# Input/output are both a pandas.DataFrame
def subtract_mean(pdf):
    return pdf.assign(v=pdf.v - pdf.v.mean())

df.groupby('id').apply(subtract_mean)
```

Blog: <https://tinyurl.com/pyspark-udf>

Overview



Continuous
Processing



ML Streaming
+
Image Reader

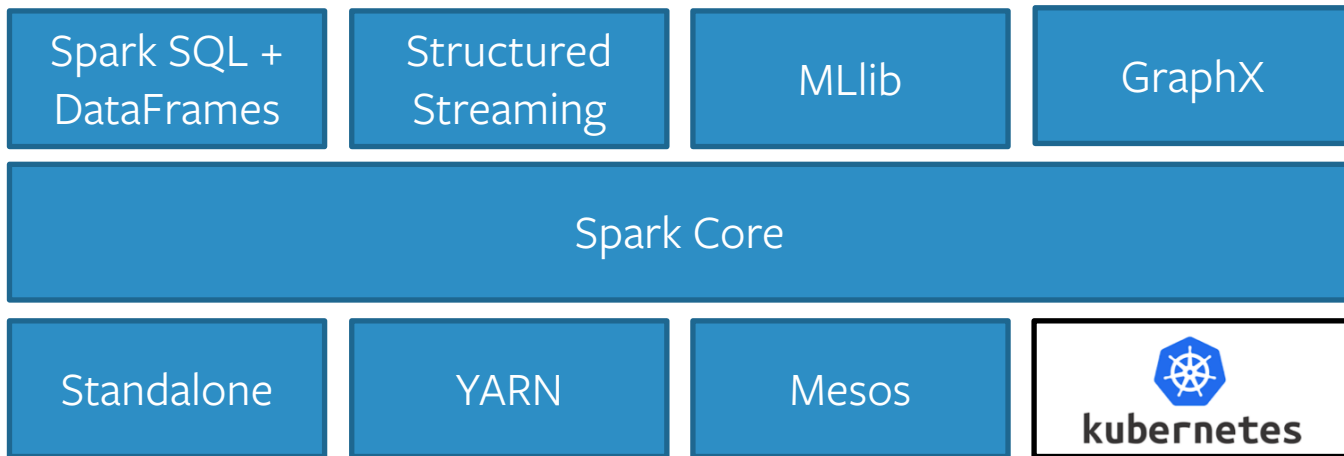


PySpark
Performance



Spark on
Kubernetes

Spark on Kubernetes (SPARK-18278)



Spark on Kubernetes (SPARK-18278)

- Driver runs in a Kubernetes pod created by the submission client and creates pods that runs the executors in response to requests from Spark Scheduler
- Make direct use of Kubernetes clusters for multi-tenancy and sharing through [Namespaces](#) and [Quotas](#), as well as administrative features such as [Pluggable Authorization](#) and [Logging](#)

Spark on Kubernetes (SPARK-18278)

Apache Spark 2.3

- Supports K8S 1.6+
- Cluster Mode
- Static Resource Allocation
- Java/Scala Applications
- Container-local and remote-dependencies that are downloadable

Roadmap (Apache Spark 2.4+)

- Client Mode
- Dynamic Resource Allocation + External Shuffle Service
- Python/R Applications
- Client-local dependencies + Resource Staging Server (RSS)

Blog: <https://tinyurl.com/spark-k8s>

Recap



Continuous
Processing



ML Streaming
+
Image Reader



PySpark
Performance



Spark on
Kubernetes

Questions?

Sameer Agarwal

Spark Summit | San Francisco | June 6th 2018

[#DevSAIS16](#)