Deep Learning and Streaming in Apache Spark 2.x

Matei Zaharia
@matei_zaharia

databricks

Welcome to Spark Summit Europe

Our largest European summit yet

102
talks

1200
attendees

11
tracks

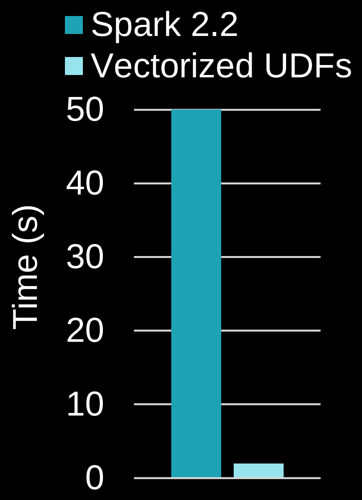
What's New in Spark?

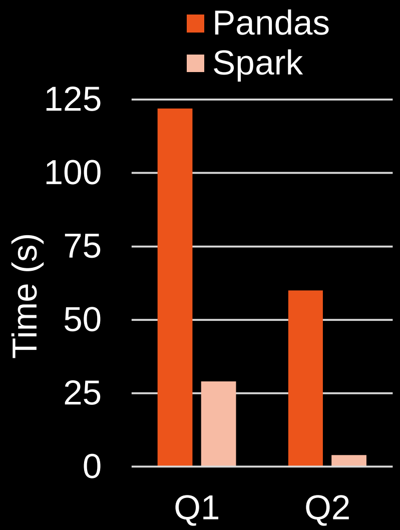
Cost-based optimizer (Spark 2.2)

Python and R improvements

- PyPI & CRAN packages (Spark 2.2)
- Python ML plugins (Spark 2.3)
- Vectorized Pandas UDFs (Spark 2.3)

Kubernetes support (targeting 2.3)





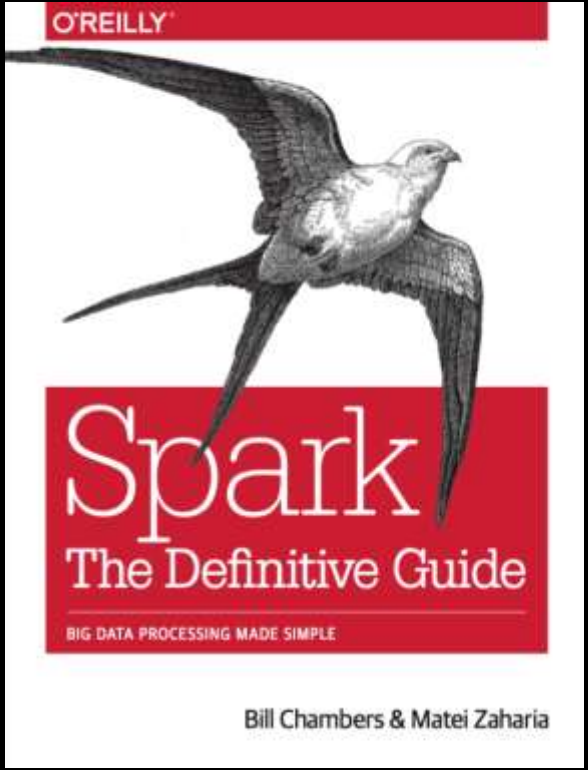
databricks

Spark: The Definitive Guide

To be released this winter

Free preview chapters and code on Databricks website:

dbricks.co/spark-guide



Two Fast-Growing Workloads

Streaming & **Deep Learning**

Both are important but complex with current tools

We think we can simplify both with Apache Spark!

Why are Streaming and DL Hard?

Similar to early big data tools!

Tremendous potential, but very hard to use at first:

- Low-level APIs (MapReduce)
- Separate systems for each task (SQL, ETL, ML, etc)

Spark's Approach

1) Composable, high-level APIs

- Build apps from components

2) Unified engine

- Run complete, end-to-end apps

Expanding Spark to New Areas

1. **Structured Streaming**

2. **Deep Learning**

Structured Streaming

Streaming today requires separate APIs & systems

Structured Streaming is a high-level, end-to-end API

- **Simple interface:** run any DataFrame or SQL code incrementally
- **Complete apps:** combine with batch & interactive queries
- **End-to-end reliability:** exactly-once processing

Became GA in Apache Spark 2.2

Structured Streaming Use Cases

100s of customer apps in production on Databricks

Largest apps process tens of trillions of records per month

Viacom	Monitor quality of live video streaming
iPass	Anomaly detection on millions of WiFi hotspots
Riot Games	Real-time game analytics at scale

Example: YAHOO! Benchmark

Kafka streams

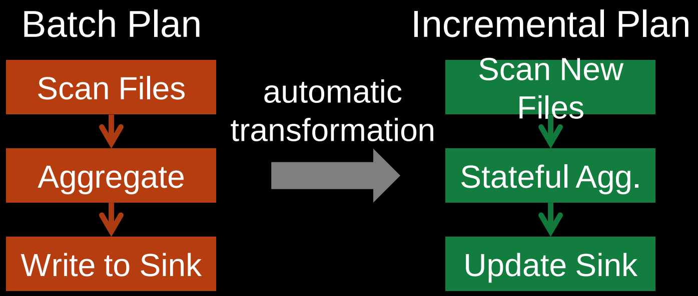
```
KStream<String, ProjectedEvent> filtered = kEvents.filter((key, value) -> {
  return value.event_type.equals("view");
}).mapValues((value) -> {
  return new ProjectedEvent(value.ad_id, value.event_time);
});


KTable<String, String> kCampaigns = builder.table("campaigns", "cmp-state");
KTable<String, CampaignAd> deserCampaigns = kCampaigns.mapValues((value) -> {
  Map<String, String> campMap = Json.parser.readValue(value);
  return new CampaignAd(campMap.get("ad_id"), campMap.get("campaign_id"));
});
KStream<String, String> joined =
  filtered.join(deserCampaigns, (value1, value2) -> {
    return value2.campaign_id;
  },
  Serdes.String(), Serdes.serdeFrom(new ProjectedEventSerializer(),
  new ProjectedEventDeserializer()));


KStream<String, String> keyedData = joined.selectKey((key, value) -> value);
KTable<Windowed<String>, Long> counts = keyedData.groupByKey()
  .count(TimeWindows.of(10000), "time-windows");
```

Spark DataFrames

```
events
.where("event_type = 'view'")
.join(table("campaigns"), "ad_id")
.groupBy(
  window('event_time, "10 seconds"),
  'campaign_id)
.count()
```

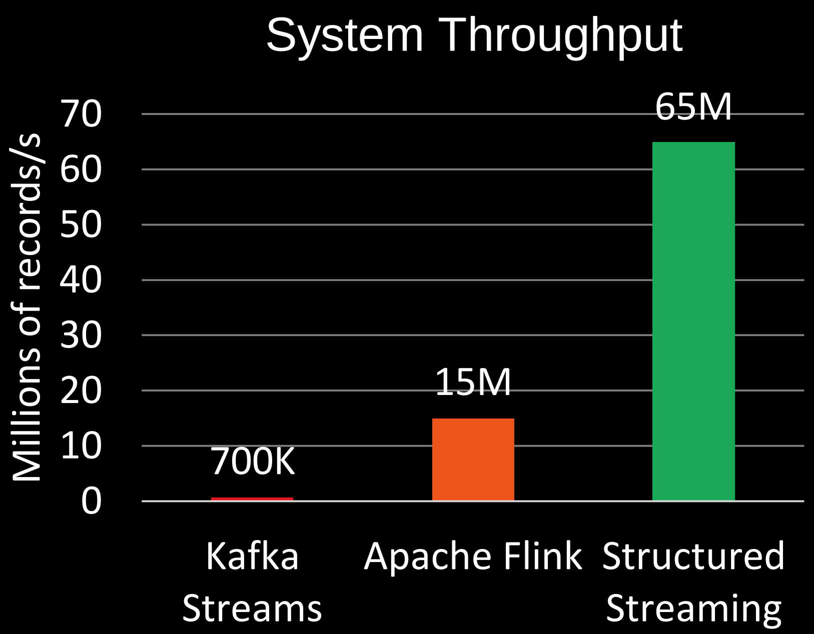


Performance: YAHOO! Benchmark

Structured Streaming reuses the **Spark SQL Optimizer** and **Tungsten Engine.**

4x

fewer nodes



https://data-artisans.com/blog/extending-the-yahoo-streaming-benchmark

databricks

What About Latency?

Continuous processing mode to run without microbatches

- **<1 ms latency** (same as per-record streaming systems)
- No changes to user code
- Proposal in SPARK-20928

Key idea: same API can target both streaming & batch

Find out more in today's deep dive

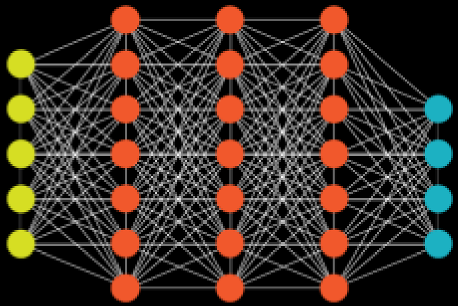
Expanding Spark to New Areas

1 **Structured Streaming**

2 **Deep Learning**

Deep Learning has Huge Potential

Unprecedented ability to work with unstructured data such as images and text



But Deep Learning is Hard to Use

Current APIs (TensorFlow, Keras, etc) are low-level

- Build a computation graph from scratch

Scale-out requires manual parallelization

Hard to use models in larger applications

Very similar to early big data APIs

databricks

Deep Learning on Spark

Image support in MLlib: SPARK-21866 (Spark 2.3)

DL framework integrations: TensorFlowOnSpark, MMLSpark, Intel BigDL

Higher-level APIs: Deep Learning Pipelines

New in TensorFlowOnSpark

Library to run distributed TF on Spark clusters & data

- Built at Yahoo!, where it powers photos, videos & more

Yahoo! and Databricks collaborated to add:

- ML pipeline APIs
- Support for non-YARN and AWS clusters

talk tomorrow at 17:00

github.com/yahoo/TensorFlowOnSpark

Deep Learning Pipelines

Low-level DL frameworks are powerful, but common use cases should be much simpler to build

Goal: Enable an **order of magnitude** more users to build **production** apps using deep learning

Deep Learning Pipelines

Key idea: High-level API built on ML Pipelines model

- Common use cases are just a few lines of code
- All operators automatically scale over Spark
- Expose models in batch, streaming & SQL apps

Uses existing DL engines (TensorFlow, Keras, etc)

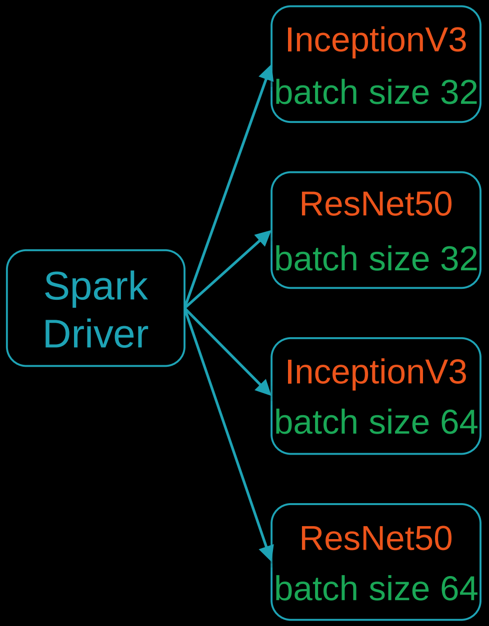
Example: Using Existing Model

```
predictor = DeepImagePredictor(inputCol="image",
                               outputCol="labels",
                               modelName="InceptionV3")

predictions_df = predictor.transform(image_df)
```

```
SELECT image, my_predictor(image) AS labels
FROM uploaded_images
```

Example: Model Search




```
est = KerasImageFileEstimator()

grid = ParamGridBuilder() \
  .addGrid(est.modelFile, ["InceptionV3", "ResNet50"]) \
  .addGrid(est.kerasParams, [{'batch': 32}, {'batch': 64}]) \
  .build()

CrossValidator(est, eval, grid).fit(image_df)
```

Deep Learning Pipelines Demo

Sue Ann Hong

databricks