Large-Scale Data Science in Apache Spark 2.0

Matei Zaharia
@matei_zaharia

Why Large-Scale?

More data = better models

Faster iteration = better models

Scale is *the* key tool of effective data science and AI

Two Forms of Scale

Hardware scalability

- Distribute work onto parallel hardware
- Utilize the hardware efficiently (e.g. fast, low-level code)

Often at odds!

User scalability

- Write applications quickly

What is Apache Spark?

Designed to tackle both challenges

- High-level APIs and libraries
- Efficient execution via parallelism and compilation

Largest open source project in big data

- 1000+ contributors, 300+ packages, 3x user growth / year

Spark for Data Science

Spark-specific libraries

- DataFrames, ML Pipelines, SQL, GraphFrames
- Parallelize common computations

Integration with existing libraries

- Call arbitrary Python / R / etc libraries at scale

Both expanding in Apache Spark 2.x

This Talk

Structured APIs in Spark 2.0

Scaling deep learning

Parallelizing traditional data science libraries

Original Spark API

Functional operations on collections of Java/Python objects (RDDs)

+ Expressive and concise
- Hard to automatically optimize

```
lines = spark.textFile(“hdfs://...”)                  // RDD[String]
points = lines.map(lambda line: parsePoint(line))     // RDD[Point]
points.filter(lambda p: p.x > 100).count()
```

Structured APIs

New APIs for data with a table-like schema

- DataFrames (untyped), Datasets (typed), and SQL

Optimized storage and computation

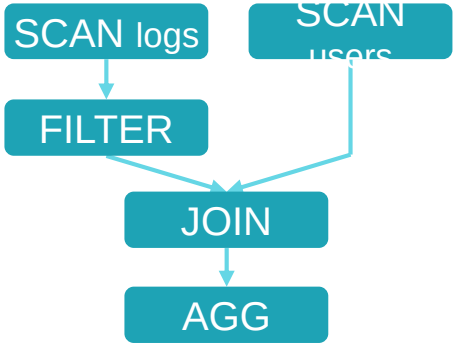
- Binary storage based on schema (e.g. columnar)
- Compute via SQL-like expressions that Spark can compile

Structured API Example

```
events =
 sc.read.json(“/logs”)

stats =
 events.join(users)
  .groupBy(“loc”,“status”)
  .avg(“duration”)

errors = stats.where(
 stats.status == “ERR”)
```



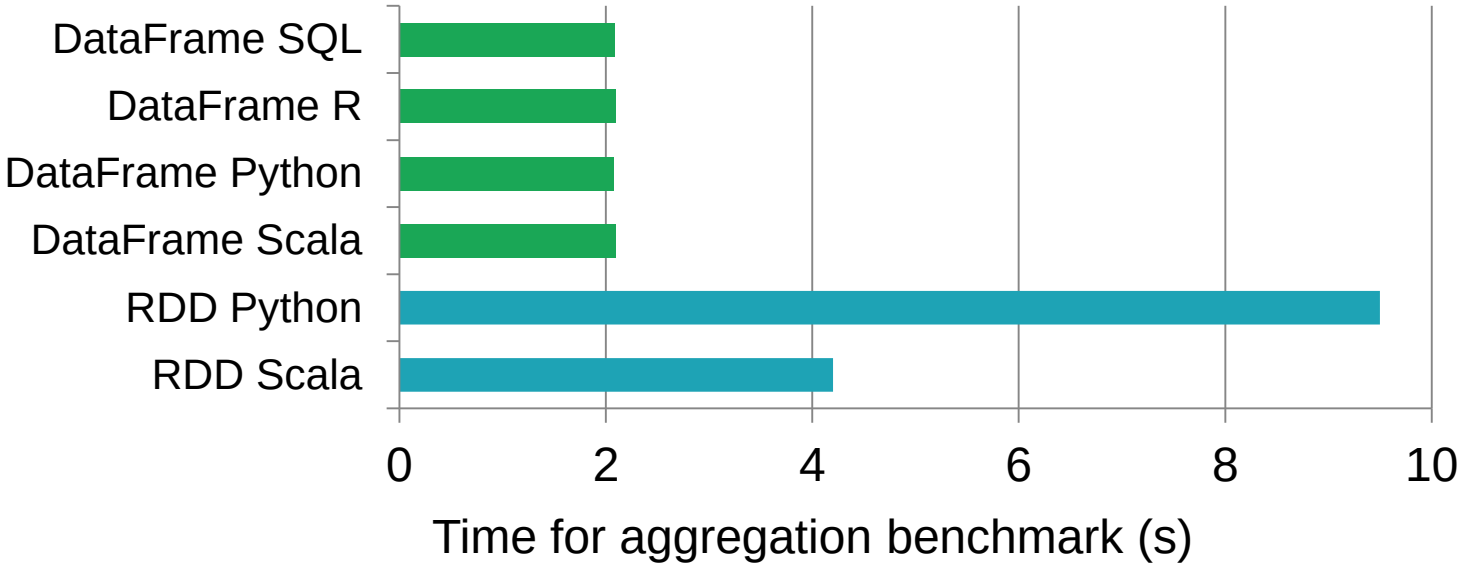

```
while(logs.hasNext) {
  e = logs.next
  if(e.status == “ERR”) {
    u = users.get(e.uid)
    key = (u.loc, e.status)
    sum(key) += e.duration
    count(key) += 1
  }
}
...
```

DataFrame API

Optimized Plan

Specialized Code

Structured API Performance



Higher-level *and* easier to optimize

Structured Streaming

Incrementalize an existing DataFrame/Dataset query

Example batch job:

```
logs = ctx.read.format(“json”).open(“hdfs://logs”)

logs.groupBy(“userid”, “hour”).avg(“latency”)
    .write.format(”parquet”)
    .save(“s3://...”)
```

Structured Streaming

Incrementalize an existing DataFrame/Dataset query

Example as streaming:

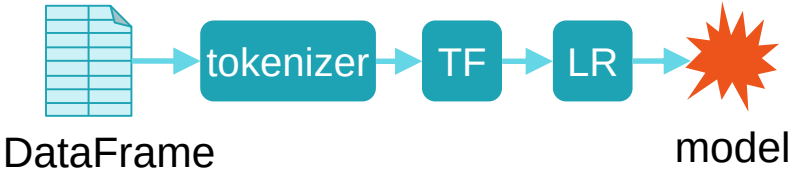
```
logs = ctx.readStream.format(“json”).load(“hdfs://logs”)

logs.groupBy(“userid”, “hour”).avg(“latency”)
    .writeStream.format(”parquet")
    .start(“s3://...”)
```

Structured APIs Elsewhere

ML Pipelines on DataFrames

- Pipeline API based on scikit-learn
- Grid search, cross-validation, etc




```
tokenizer = Tokenizer()
tf = HashingTF(numFeatures=1000)
lr = LogisticRegression()

pipe = Pipeline([tokenizer, tf, lr])
model = pipe.fit(df)
```

GraphFrames for graph analytics

- Pattern matching à la Neo4J

This Talk

Structured APIs in Spark 2.0

Scaling deep learning

Parallelizing traditional data science libraries

Why Deep Learning on Spark?

Scale out model application to large data

- For transfer learning or model evaluation

Scale out parameter search: one model per machine

Distributed training: one model on multiple machines

Deep Learning Libraries

TensorFrames

TensorFlow model eval on DataFrames, for serving or transfer learning

BigDL

Distributed model training on CPUs

TensorFlowOnSpark

Run Caffe & TensorFlow on Spark data

This Talk

Structured APIs in Spark 2.0

Scaling deep learning

Parallelizing traditional data science libraries

Parallelizing Existing Libraries

Spark Python/R APIs let you scale out existing libraries

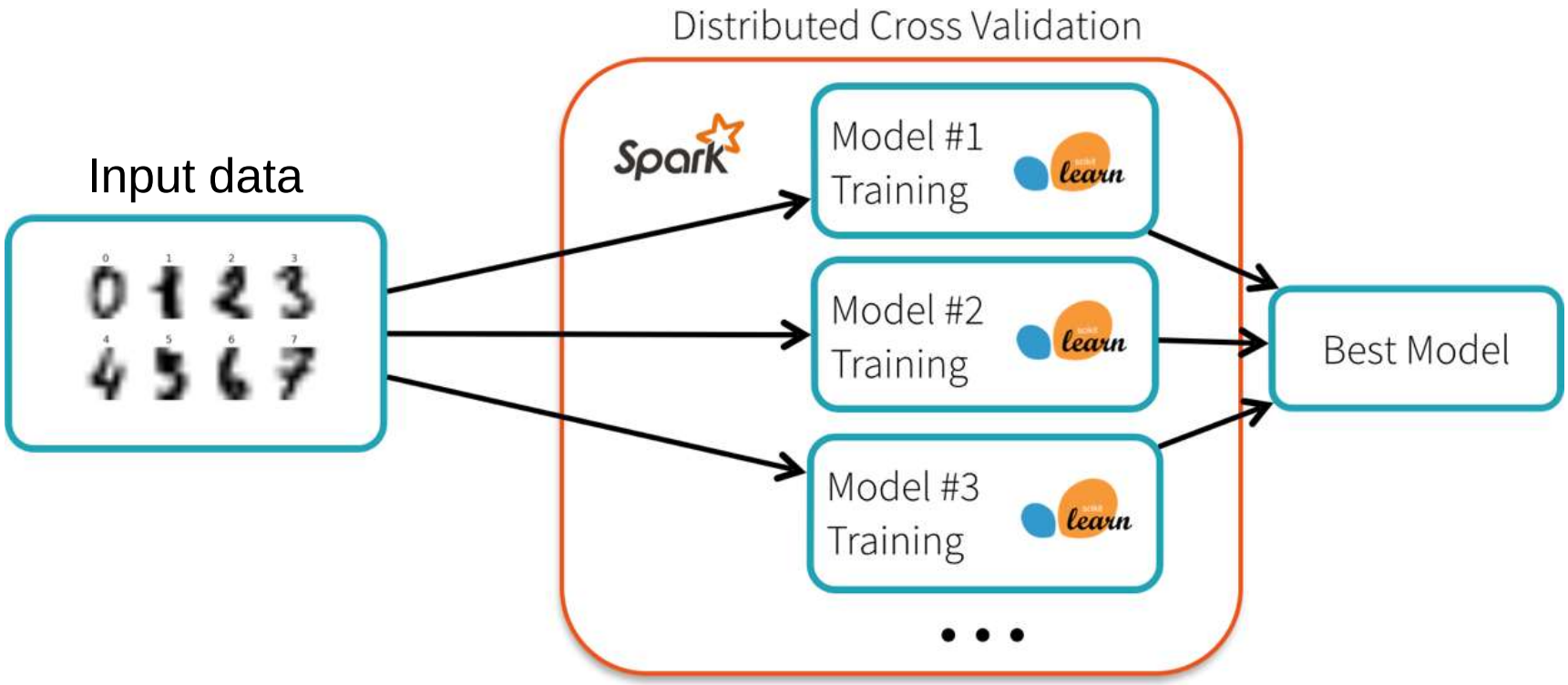
- spark-sklearn for arbitrary scikit-learn models
- SparkR dapply

```
from sklearn import svm, datasets
from spark_sklearn import GridSearchCV

iris = datasets.load_iris()
model = svm.SVC()
params = {
  'kernel': ['linear', 'rbf'],
  'C': [1, 10]
}
gs = GridSearchCV(sc, model, params)
gs.fit(iris.data, iris.target)
```

github.com/databricks/spark-sklearn

spark-sklearn Execution



Coming Soon

Native APIs for zero-copy data transfer into C libraries

Streamlined installation in Python:

pip install pyspark

To Learn More

See hundreds of talks on use cases at Spark Summit

spark-summit.org

SPARK SUMMIT

June 5-8, San Francisco

Try interactive Spark tutorials in Databricks Community Edition

databricks.com/ce

databricks