

Spark the Definitive Guide 2nd Edition

Chapter 03

A Tour of Spark's Toolset

A Tour of Spark's Toolset

Text Book



Objectives and Outcomes

- ▶ Take a tour of Spark's toolset
- ▶ Understand how to run production Spark applications
- ▶ Understand type-safe APIs for structured data
- ▶ Understand Structured Streaming and Machine Learning
- ▶ Understand SparkR and Resilient Distributed DataSets

Review

So far we have:

- ▶ learned about core architecture of Spark
 - ▶ learned about executors
 - ▶ learned about partitions
 - ▶ learned about drivers
- ▶ learned about datatypes
 - ▶ DataFrames
 - ▶ APIs
- ▶ learned about transformations
- ▶ learned about actions
- ▶ learned how to put it together from the Spark CLI

Spark Overview

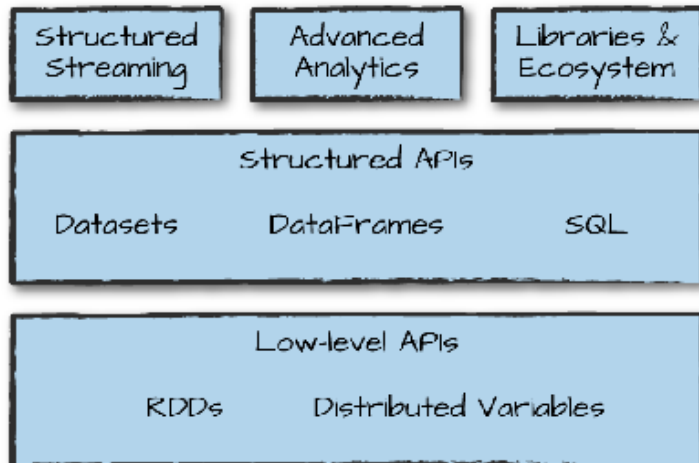


Figure 1-1. Spark's toolkit

Running Production Applications

- ▶ `spark-submit`
 - ▶ Different from the interactive shell commands we saw in chapter 02
 - ▶ `spark-submit` does one thing: send your code to a cluster for execution
 - ▶ Application will run until finished or reports and error
- ▶ Types of **cluster managers** include:
 - ▶ local system (as threads)
 - ▶ Mesos
 - ▶ YARN

Sample Code

- ▶ `spark-submit --class org.apache.spark.examples.SparkPi --master local examples/jars/spark-examples_2.11-2.4.4.jar 10`
 - ▶ The file name was changes since we are using version 2.4.4 not 2.2.0
 - ▶ The job can also be submitted to a cluster by changing the `--master local` to `--master yarn` or `--master mesos`

Type-Safe DataSets

- ▶ Spark uses multiple languages:
 - ▶ Scala, Java, Python, R, and SQL
 - ▶ Java and Scala are statically typed languages
 - ▶ Python and R are not statically typed, but dynamically typed
- ▶ How to handle type-safety?
 - ▶ Recall that DataFrames (chapter 2) are a distributed collection of objects of type **Row**
 - ▶ DataSet API allows you to assign a Java/Scala class to the records within a DataFrame
 - ▶ Manipulate that data like a Java ArrayList or Scala Seq
- ▶ DataSets can be used as needed
 - ▶ DataSets can be cast back into DataFrames
 - ▶ Allows for *casting* of data depending on your needs
 - ▶ Large applications logic will need/enforce type safety, but data analysis via SQL won't need type safety
 - ▶ DataSets Covered in depth in Chapter 11

Example code of DataSets 39

```
case class Flight(DEST_COUNTRY_NAME: String,
ORIGEN_COUNTRY_NAME: String, count: BigInt)
val flightsDF = spark.read
.parquet("/data/flight-data/
parquet/2010-summary.parquet/")
val flights = flightsDF.as[Flight]
```

Structured Streaming

- ▶ High-level API for stream processing
 - ▶ Added in Spark 2.2, 2017-07-11
- ▶ Structured Streaming takes batch mode operations and run them in a streaming fashion
- ▶ See page 40 to page 44 in the e-book or class video for demonstration

Machine Learning and Advanced Analytics

- ▶ Has a built in Spark Library called MLlib
- ▶ Allows for many options:
 - ▶ Preprocessing
 - ▶ Munging
 - ▶ training of models

k-Means clustering

- ▶ Using data loaded in the previous example we will:
 - ▶ ingest raw data
 - ▶ build up transformations
 - ▶ train our simple model to make predictions
- ▶ MLlib requires data to be represented as numerical data
 - ▶ Sample shopping data is of all different types
 - ▶ Need to use *transformations* to change the datatype
 - ▶ P. 45-48 in the e-book has the steps needed
 - ▶ Book has the steps needed to split our dataset into training and test sets

Lower-Level APIs

- ▶ RDDs - you should mostly stick to higher level APIs
- ▶ This section we will skip for now

SparkR and Other Packages

- ▶ Is a tool for running R on Spark
- ▶ Similar to the Python language but uses R syntax
 - ▶ Can import other Spark libraries to make programming more R-like
- ▶ <https://spark-packages.org>
 - ▶ Spark for dotNet
 - ▶ Work through the tutorial and see if you can get the results

Conclusion

- ▶ We learned Spark's programming model
- ▶ We learned how to run production code
- ▶ We were introduced to type-safe data structures in Spark
- ▶ We were introduced to Structured Streaming on Spark
- ▶ We were introduced to Machine Learning on Spark
- ▶ We were introduced to 3rd party Spark packages