cloudera

dplyr Interfaces to Large-Scale Data

Ian Cook
@ianmcook
ian@cloudera.com

Context

Mission for Cloudera: Provide a platform for data analysts, data scientists to efficiently query, analyze, model large-scale data in clusters, cloud storage

- By distributing Apache Spark, Apache Impala, other tools
- By enabling productive use of these tools

Python and R users often have difficulty moving from smaller data to large-scale distributed data

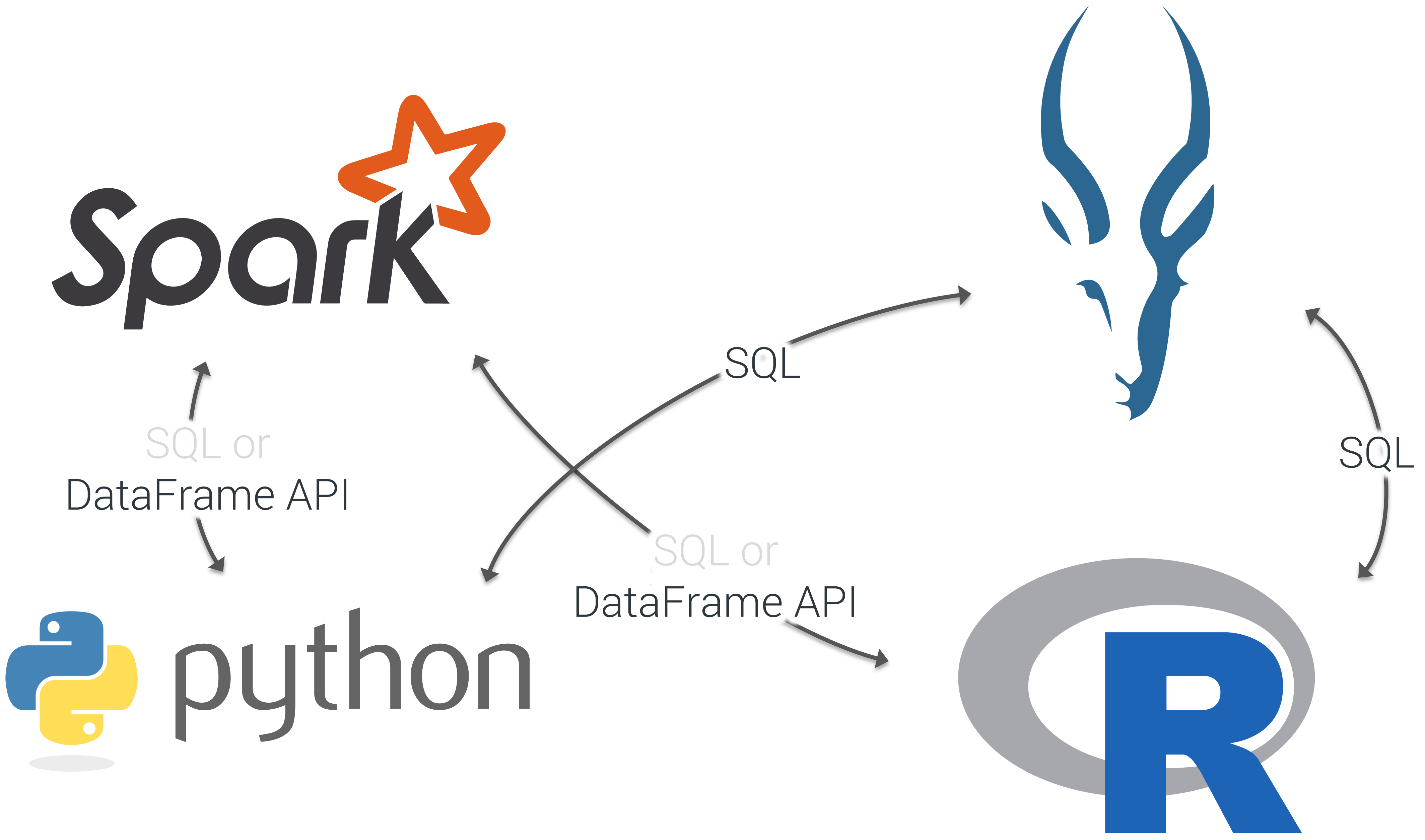
- Familiar packages, methods don’t work the same way on distributed data

cloudera

© Cloudera, Inc. All rights reserved.

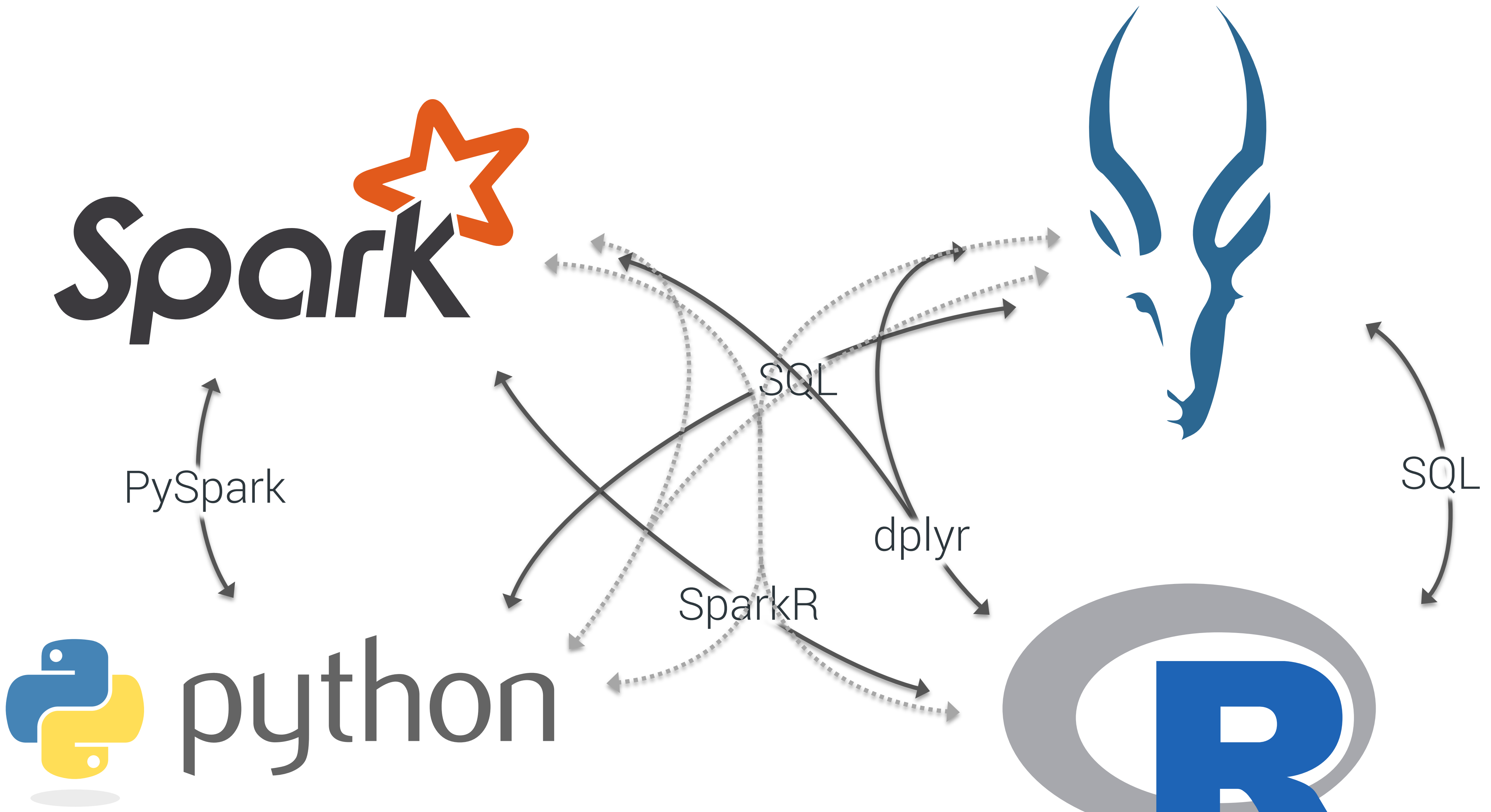
Poll question

cloudera


SQL or
DataFrame API
SQL
SQL or
DataFrame API
SQL

Poll question

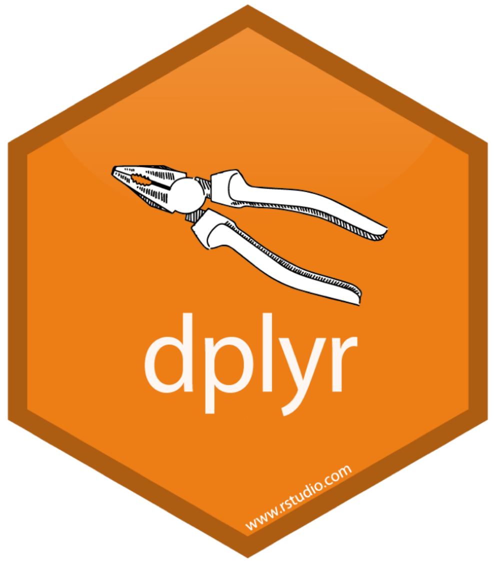
© Cloudera, Inc. All rights reserved.


Spark
python
PySpark
SQL
dplyr
SparkR
SQL
R

dplyr

dplyr provides a set of *verbs* that perform common data manipulation steps

- `select()` to select columns
- `filter()` to filter rows
- `arrange()` to order rows
- `mutate()` to create new columns
- `summarise()` to aggregate
- `group_by()` to perform operations by group



dplyr works on local data and with remote data sources

- For remote sources, dplyr commands are translated into SQL

© Cloudera, Inc. All rights reserved.

Poll question

© Cloudera, Inc. All rights reserved.

Demonstration

Example code at
github.com/ianmcook/dplyr-examples

© Cloudera, Inc. All rights reserved.

dplyr SQL backends

dplyr

↕

dbplyr

↕

dplyr SQL backend package*

↕

DBI

↕

DBI-compatible interface package

↕

database driver or connector

↕

database/engine

* optional

sparklyr

- Provides a SQL backend to dplyr for Spark
- Also exposes the MLlib API and a subset of the Spark DataFrames API
- Developed by RStudio

spark.rstudio.com

implyr

- Provides a SQL backend to dplyr for Impala
- Uses ODBC or JDBC to connect to Impala
- Developed at Cloudera

tiny.cloudera.com/implyr

© Cloudera, Inc. All rights reserved.

Five tips for using dplyr with SQL data sources

© Cloudera, Inc. All rights reserved.

1

Use `show_query()`

2

`filter()` early
`arrange()` late

3

Check your data types

4

Know your SQL engine

cloudera

5

Know when to `collect()`

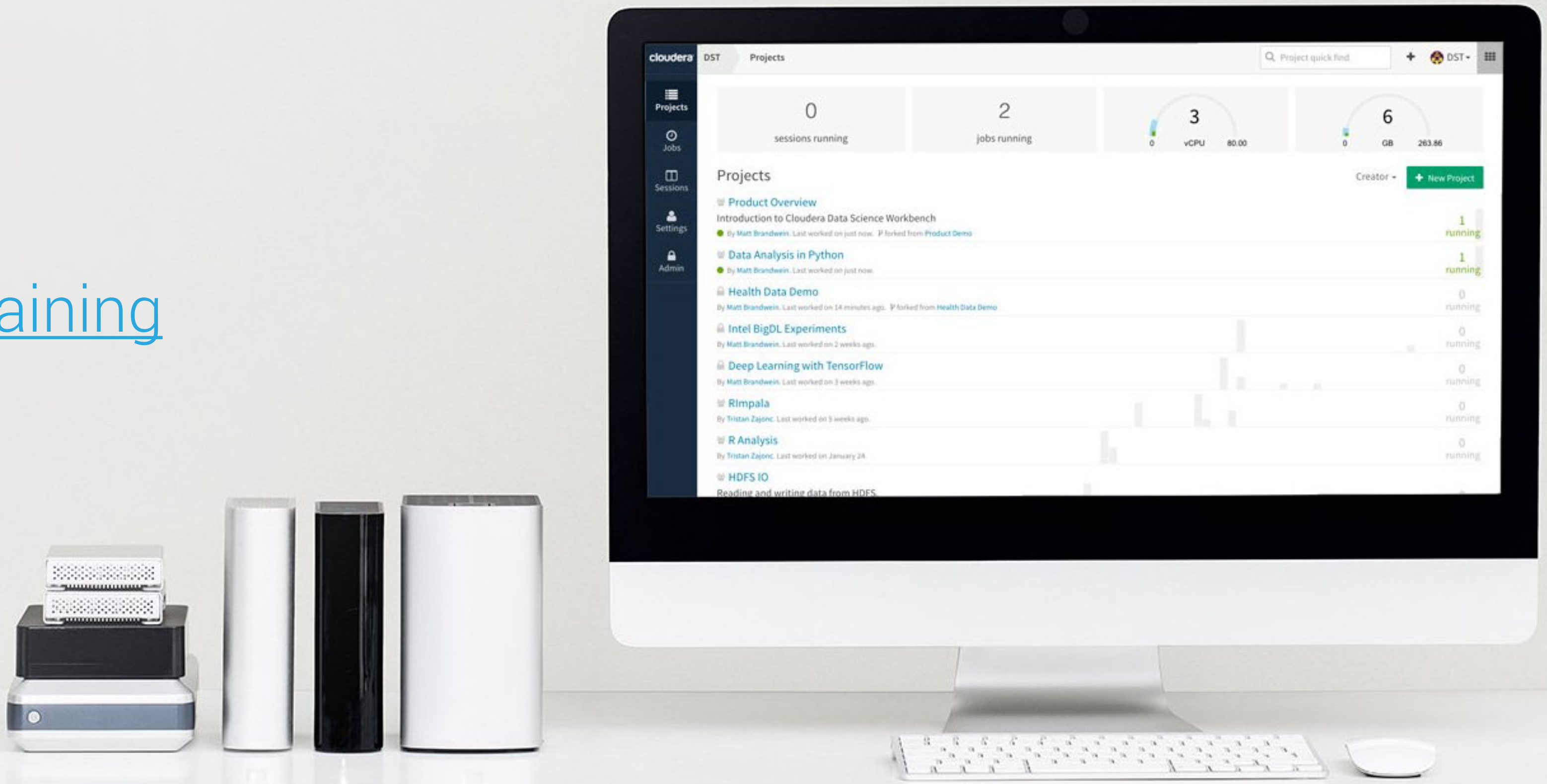
Cloudera Data Science Workbench

More information

tiny.cloudera.com/cdsw

OnDemand training

tiny.cloudera.com/cdsw-training



0 sessions running

2 jobs running

3 vCPU (0 – 80.00)

6 GB (0 – 263.86)

Projects

Product Overview
Introduction to Cloudera Data Science Workbench
By Matt Brandwein. Last worked on just now. forked from Product Demo — 1 running

Data Analysis in Python
By Matt Brandwein. Last worked on just now. — 1 running

Health Data Demo
By Matt Brandwein. Last worked on 14 minutes ago. forked from Health Data Demo — 0 running

Intel BigDL Experiments
By Matt Brandwein. Last worked on 2 weeks ago. — 0 running

Deep Learning with TensorFlow
By Matt Brandwein. Last worked on 3 weeks ago. — 0 running

RImpala
By Tristan Zajonc. Last worked on 3 weeks ago. — 0 running

R Analysis
By Tristan Zajonc. Last worked on January 24. — 0 running

HDFS IO
Reading and writing data from HDFS