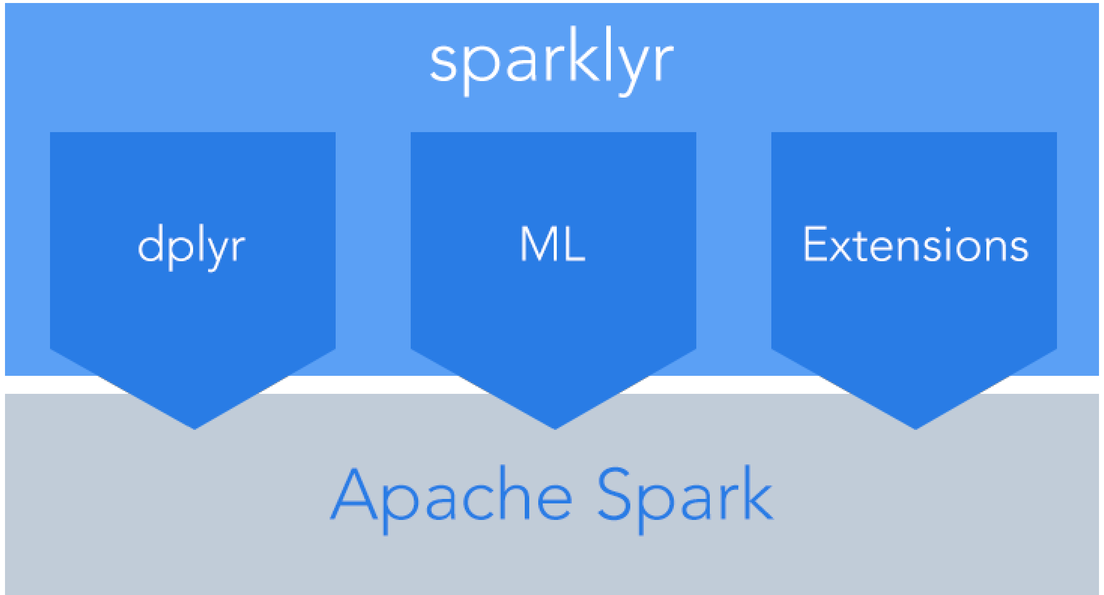
Introduction to sparklyr – a new package that provides an interface between R and Apache spark



Kalbi F. ZONGO

PhD candidate

Statistics Department OSU

Oregon State University

Cascadia R Conf

Outline

1. Introducing Apache Spark & Sparklyr

2. Local Connection to Spark

3. Remote Connection to Spark

Apache Spark

http://spark.apache.org

- Cluster Computing & Database: Processing big data, ease of use, applications in R, Python, Java, Scala
- Four (4) Modules:
  1. **Spark SQL** lets you query structured data inside Spark
  2. **MLlib** scalable machine learning library fits into Spark APIs.
  3. **Spark Streaming** for streaming computation
  4. **GraphX** for graphical computation

Sparklyr

http://spark.rstudio.com

- R interface for Apache Spark
- Connect to Spark from R
- Provide dplyr backend for data manipulation: all verbs are translated into Spark SQL query
- Provide DBI backend to execute SQL queries directly against spark tables
- Use Spark distributed ML libraries from R
- Create extension, etc.

Local Connection to Spark: Getting Setup

```
# 1. Install and load sparklyr:
install.packages("sparklyr")
library(sparklyr)

# 2. Install local copy of spark:
spark_install(version = '2.0.0')

# 3. Connect to spark:
sc <- spark_connect(master = "local",
                    version = "2.0.0")
```

Local Connection to Spark: Importing data

```
library(nycflights13)
library(dplyr)

# Copy data in Spark
copy_to(sc, flights, "flights_tbl")

Spark_read_csv(); spark_read_json()

# See available data
src_tbls(sc)
[1] "flights_tbl"
```

Oregon State University

Cascadia R Conf

Local Connection to Spark: Data Manipulation

Task: For each carrier, find the flights with the longest departure delay

dplyr

```
Q <- tbl(sc, 'flights_tbl')  %>%
       group_by(carrier) %>%
       mutate(rank = rank(desc(dep_delay))) %>%
       filter(rank <= 2) %>%
       select(carrier, year, month, day, dep_delay,
            rank)
collect(Q)
sql_render(Q)
```

Oregon State University

Cascadia R Conf

Local Connection to Spark: Data Manipulation

Task: For each carrier, find the flights with the longest departure delay

Spark SQL

```
library(DBI)
Q <- dbGetQuery(sc,
"SELECT `carrier` AS `carrier`, `year` AS `year`, `month`
AS `month`, `day` AS `day`, `dep_delay` AS `dep_delay`,
`rank` AS `rank`
FROM (SELECT *
FROM (SELECT `year`, `month`, `day`, `dep_time`,
`dep_delay`, `arr_time`, `arr_delay`, `carrier`,
`tailnum`, `flight`, `origin`, `dest`, `air_time`,
`distance`, `hour`, `minute`, rank() OVER (PARTITION BY
`carrier` ORDER BY `dep_delay` DESC) AS `rank`
FROM `flights_tbl`) `zylrnrilkq`
WHERE (`rank` <= 2.0)) `bfsfcfmxpt`")
```

Oregon State University

Cascadia R Conf

Local Connection to Spark: Machine Learning

MLlib application

Task: what factors influence departure delay at JFK?

```
# Prepare model data
model_data <- tbl(sc, 'flights_tbl')  %>%
           filter(origin == ‘JFK’, dep_delay > 0, arr_delay > 0)
# Partition
partitions <- model_data %>%
           sdf_partition(train = .7, test = .3)
# Fit a linear regression
fit <- partitions$train %>%
      ml_linear_regression(response = ‘dep_delay’,
                                 features = c(‘arr_delay’, ‘distance’, ‘month’,
                                            ‘day’, ‘hour’, ‘carrier’) )
summary(fit)
# Predict on test set
predicts <- sdf_predict(fit, partitions$test) %>%
                      collect()
```

MLlib functions

- ml_decision_tree()
- ml_kmeans()
- ml_naive_baye()
- ml_logistic_regression()
- ml_multilayer_perceptron()
- ml_pca()
- ml_random_forest()
- ml_survival_regression()
- etc.

Oregon State University

Cascadia R Conf

Local Connection to Spark

- dplyr for convenient data manipulation
- Mllib – easy to implement ML algorithms
- **Disconnect** with **spark_disconnect**(sc)
- More details:

  https://www.rstudio.com/resources/cheatsheets/

  http://spark.rstudio.com

Remote Connection to Spark

- Amazon EC2 (EMR): scripts that let you launch a cluster on EC2 in about 5 minutes
- Standalone Deploy Mode: launch a standalone cluster quickly without a third-party cluster manager
- Mesos: deploy a private cluster using Apache Mesos
- YARN: deploy Spark on top of Hadoop NextGen (YARN)
- Requires **Rstudio Server** or **Rstudio Pro & Sparklyr** on the master node

http://spark.apache.org/docs/latest/#launching-on-a-cluster

Thank You!

Contact: zongok@oregonstate.edu

Oregon State University

Cascadia R Conf