Spark官方正式宣布支持SparkR(R on Spark)

我（不是博主，这里的我指的是Shivaram Venkataraman）很高兴地宣布即将发布的Apache Spark 1.4 release将包含SparkR，它是一个R语言包，允许数据科学家通过R shell来分析大规模数据集以及交互式地运行Jobs。

R语言是一个非常流行的统计编程语言，并且支持很多扩展以便支持数据处理和机器学习任务。然而，R中交互式地数据分析常常局限在单个线程运行环境中，而且只能处理适合一台机器内存的数据集。SparkR，一个R语言包，最初由AMPLab开发，提供了R语言和Apache Spark交互的前端，并且可以在R Shell中使用Spark的分布式计算引擎来分析大规模的数据集。

项目历史

SparkR最初由AMPLab开发，旨在探索能够将Spark和R整合。在这些努力基础上，SparkR开发预览版于2014年1月第一次开源。在接下来的一年里，该项目继续在AMPLab开发，通过来自开源社区对SparkR的贡献，SparkR的需要性能和可用性都得到了极大的提升。SparkR由近期合并到Apache Spark工程中去了，并且在Spark 1.4中作为alpha版的组建发布。

SparkR DataFrames

SparkR 1.4 release中最核心的组建莫过于SparkR DataFrame，它是构建在Spark之上的分布式data Frame。Data Frames是R中用于数据处理的最基础的数据结构，并且它的概念已经扩展到其他语言，比如Pandas。类似的工程(比如dplyr)进一步简化data Frame上复杂的数据操作任务。SparkR DataFrames提供了类似于dplyr和local R data frames的API，但是这里的DataFrames可以利用Spark的分布式计算支持将计算扩展到大规模的数据集规模。

下面的例子展示了SparkR中的DataFrame部分API：

```
# Download Spark 1.4 from http://spark.apache.org/downloads.html
#
# Download the nyc flights dataset as a CSV from https://s3-us-west-2.amazonaws.com/sparkr-data/nycflights13.csv

# Launch SparkR using
# ./bin/sparkR --packages com.databricks:spark-csv_2.10:1.0.3

# The SparkSQL context should already be created for you as sqlContext
sqlContext
# Java ref type org.apache.spark.sql.SQLContext id 1

# Load the flights CSV file using `read.df`. Note that we use the CSV reader Spark package here.
```

```
flights <- read.df(sqlContext, "./nycflights13.csv", "com.databricks.spark.csv", header="true")

# Print the first few rows
head(flights)

# Run a query to print the top 5 most frequent destinations from JFK
jfk_flights <- filter(flights, flights$origin == "JFK")

# Group the flights by destination and aggregate by the number of flights
dest_flights <- agg(group_by(jfk_flights, jfk_flights$dest), count = n(jfk_flights$dest))

# Now sort by the `count` column and print the first few rows
head(arrange(dest_flights, desc(dest_flights$count)))

##  dest count
##1  LAX 11262
##2  SFO  8204
##3  BOS  5898

# Combine the whole query into two lines using magrittr
library(magrittr)
dest_flights <- filter(flights, flights$origin == "JFK") %>% group_by(flights$dest) %>% summariz
e(count = n(flights$dest))
arrange(dest_flights, desc(dest_flights$count)) %>% head
```

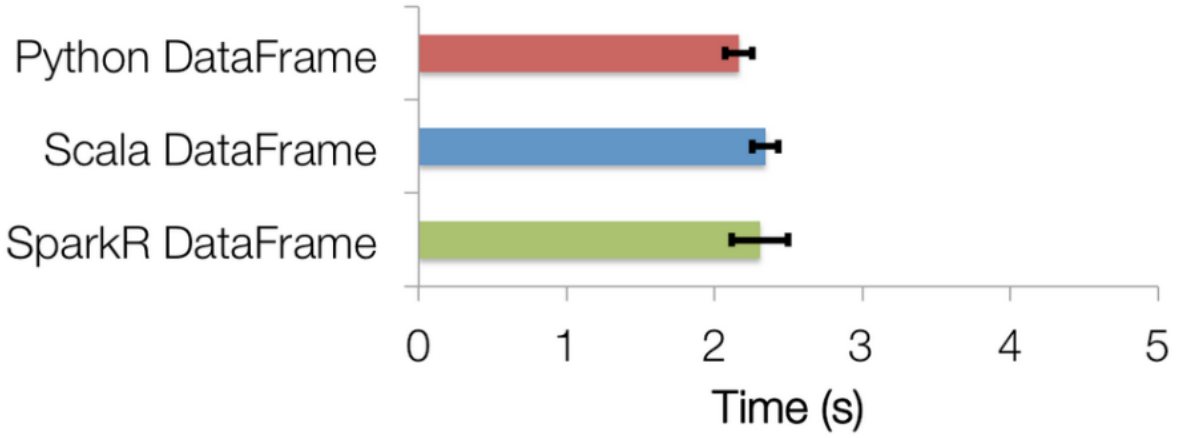
为了更全面的介绍DataFrames，你可以查看SparkR的编程指南（这篇文章已经在本博客翻译过，点击这里）英文文档点这里。

和Spark整合的优势

除了具有非常容易使用的API，SparkR通过和Spark的紧密结合继承了许多优点，如下：

1、Data Sources API：通过和Spark SQL的外部数据源API捆绑，SparkR可以读取多种的数据源，比如Hive表，JSON文件，Parquet文件等。

2、Data Frame Optimizations： SparkR DataFrames同样继承了计算引擎的所有优化，包括代码生成和内存管理。比如，下图展示的是分别在R，Python和Scala中对一亿条正数对进行group-by聚合的操作性能统计（这里使用的数据集来自这里）。从这张图可以看出，通过使用计算引擎的各种优化，使得SparkR的性能和Scala/Python类似。



3、Scalability to many cores and machines：在SparkR DataFrames上运行的各种操作自动地分布到Spark集群可用的所有core和机器中。这使得SparkR DataFrames可以处理TB级别的数据，并且可以在拥有千台机器的集群上运行！

展望未来

在即将发布的release中还有很多对SparkR的功能计划。其中包括了高层次的机器学习算法和使得SparkR DataFrames成为Spark中一个稳定的组建。

本文翻译自：《Announcing SparkR: R on Spark》