SparkR: Enabling Interactive Data Science at Scale

Shivaram Venkataraman
Zongheng Yang
Talk Outline

Motivation
Overview of Spark & SparkR API
Live Demo: Digit Classification
Design & Implementation
Questions & Answers
Key Advantages of Spark?

- Fast
- Scalable
- Expressive
Key Advantages of R?

Numerical Packages Interactive
SparkR is a language binding that seamlessly integrates R with Spark, and enables native R programs to scale in a distributed setting.
RDD
(Resilient Distributed Datasets)

Transformations
map
filter
groupBy
...

Actions
count
collect
saveAsTextFile
...

Spark

(Resilient Distributed Datasets)
R + RDD = R2D2!
R + RDD = RRDD
Getting Closer to Idiomatic R

Q: How can I use a loop to [...] ?

A: Don’t. Use one of the apply functions.

From: http://nsaunders.wordpress.com/2010/08/20/a-brief-introduction-to-apply-in-r/
Example: Word Count

```r
lines <- textFile(sc, "hdfs://my_text_file")
```
Example: Word Count

lines <- textFile(sc, "hdfs://my_text_file")

words <- flatMap(lines,
    function(line) {
        strsplit(line, " ")[[1]]
    }) # "hi" "hi" "all"

wordCount <- lapply(words,
    function(word) {
        list(word, 1) # eg. ("all", 1)
    })
Example: Word Count

```r
lines <- textFile(sc, "hdfs://my_text_file")

words <- flatMap(lines,
    function(line) {
        strsplit(line, " ")[[1]]
    }) # "hi" "hi" "all"

wordCount <- lapply(words,
    function(word) {
        list(word, 1) # eg. ("all", 1)
    })

counts <- reduceByKey(wordCount, "+", numPartitions=2)

output <- collect(counts) # ("hi", 2), ("all", 1), ...
```
Live Demo
Digit Classification: MNIST
High-level Plan

Minimize $\|Ax - b\|_2$

$x = (A^T A)^{-1} A^T b$
How does this work?
Dataflow

Local

R

Worker

Worker
Dataflow

Local

R

Spark Context (ref. in R)

JNI

Java Spark Context

Worker

Worker
Dataflow

Local

Spark Context (ref. in R)

Java Spark Context

Worker

Spark Executor

tasks
broadcast vars
R packages

Worker

Spark Executor

tasks
broadcast vars
R packages

R

R

R

R

JNI
… Pipes?
Capturing Closures: Environments

From http://obeautifulcode.com/R/How-R-Searches-And-Finds-Stuff/
save {base}

Description

save writes an external representation of R objects to the specified file at a later date by using the function load (or data.load). save.image() is just a short-cut for ‘save my current workspace file = ".RData"'). It is also what happens with q("yes").
Alpha developer release

One line install!

install_github("amplab-extras/SparkR-pkg", subdir="pkg")
On Github

EC2 setup scripts

All Spark examples

MNIST demo

Hadoop2, Maven build
SparkR Implementation

Lightweight

292 lines of Scala code
1694 lines of R code
549 lines of test code in R

=> Spark is easy to extend!
Possible Future Work

Calling MLLib from R

Data Frame support

Daemon R processes
SparkR

- Seamless integration
- Scale R programs in a distributed fashion
- Combine scalability & utility
Thanks!

https://github.com/amplab-extras/SparkR-pkg

Shivaram Venkataraman  shivaram@cs.berkeley.edu
Zongheng Yang  zhyang@berkeley.edu
Spark User mailing list  user@spark.apache.org
Pipeline the transformations!

... words <- `flatMap`(lines, ...)
wordCount <- `lapply`(words, ...)
...
Pipelined RDD

Spark Executor \(\rightarrow\) R \(\rightarrow\) Spark Executor \(\rightarrow\) R

Spark Executor \(\rightarrow\) R \(\rightarrow\) R \(\rightarrow\) Spark Executor
- SparkR
- Spark
- Mesos / YARN / ...
- HDFS / HBase / Cassandra / ...

Processing Engine
Cluster Manager
Storage
amplab-extras/SparkR-pkg

R frontend for Spark

<table>
<thead>
<tr>
<th>Current</th>
<th>Build History</th>
<th>Pull Requests</th>
<th>Branch Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>✔️ 34</td>
<td>Commit</td>
<td>c5bce07 (master)</td>
</tr>
<tr>
<td>State</td>
<td>Passed</td>
<td>Compare</td>
<td>aacd72657106...c5bce07ef517</td>
</tr>
<tr>
<td>Finished</td>
<td>23 days ago</td>
<td>Author</td>
<td>Zongheng Yang</td>
</tr>
<tr>
<td>Duration</td>
<td>9 min 37 sec</td>
<td>Committer</td>
<td>Zongheng Yang</td>
</tr>
<tr>
<td>Message</td>
<td>Merge pull request #30</td>
<td>from shivaram/string-tests</td>
<td></td>
</tr>
</tbody>
</table>

Add tests for partitioning with string keys
Example: Logistic Regression

```r
pointsRDD <- textFile(sc, "hdfs://myfile")
weights <- runif(n=D, min = -1, max = 1)

# Logistic gradient
gradient <- function(partition) {
  X <- partition[,1]; Y <- partition[,,-1]
  t(X) %*% (1/(1 + exp(-Y * (X %*% weights))) - 1) * Y
}
```
Example: Logistic Regression

pointsRDD <- textFile(sc, "hdfs://myfile")
weights <- runif(n=D, min = -1, max = 1)

# Logistic gradient
gradient <- function(partition) {
  X <- partition[,1]; Y <- partition[,,-1]
  t(X) %*% (1/(1 + exp(-Y * (X %*% weights)))) - 1) * Y
}

# Iterate
weights <- weights - reduce(lapplyPartition(pointsRDD, gradient), "+")
How does it work?

Data:
RDD[Array[Byte]]

Functions:
Array[Byte]