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

Our Motivation

Fast

Numerical

Scalable

+

Packages

Expressive

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
...
```

R + RDD =

R2D2!

(map) lapply

(mapPartitions) lapplyPartition

groupByKey

reduceByKey

sampleRDD

…

collect

cache

…

textFile

parallelize

broadcast

includePackage

R + RDD =

RRDD

Getting Closer to Idiomatic R

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

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

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

Example: Word Count

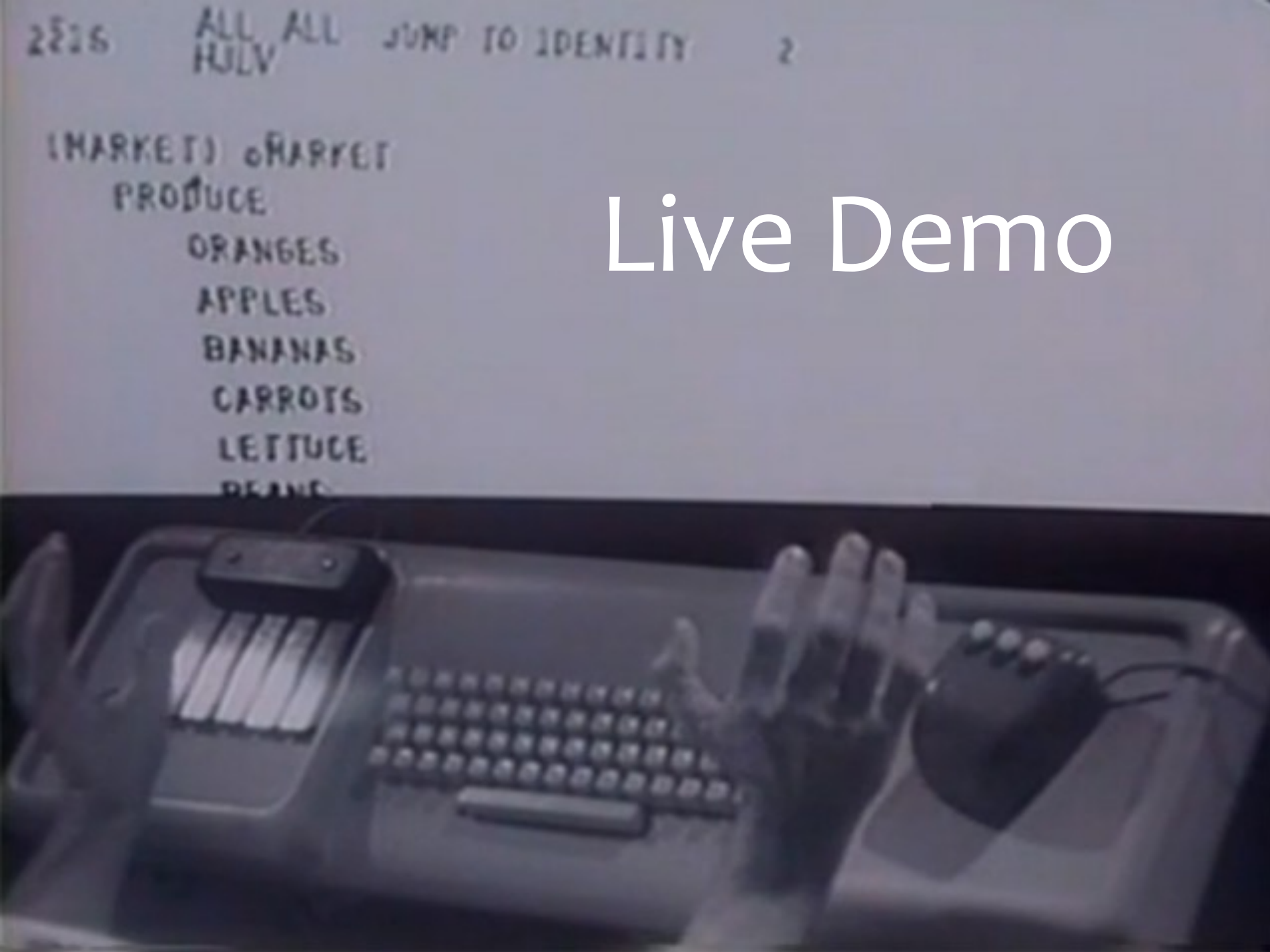
```
lines <- textFile(sc, "hdfs://my_text_file")

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

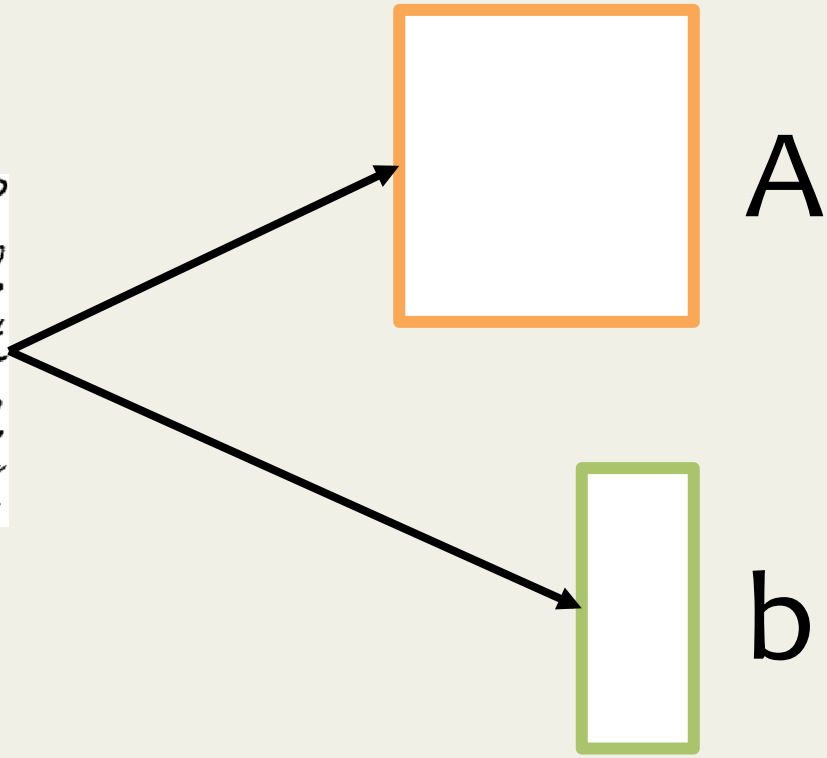
```
lines <- textFile(sc, "hdfs://my_text_file")

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), …
```


2216 ALL ALL JUMP TO IDENTITY 2
HULV
(MARKET) eMARKET
PRODUCE
ORANGES
APPLES
BANANAS
CARROTS
LETTUCE
Live Demo

Digit Classification: MNIST

High-level Plan

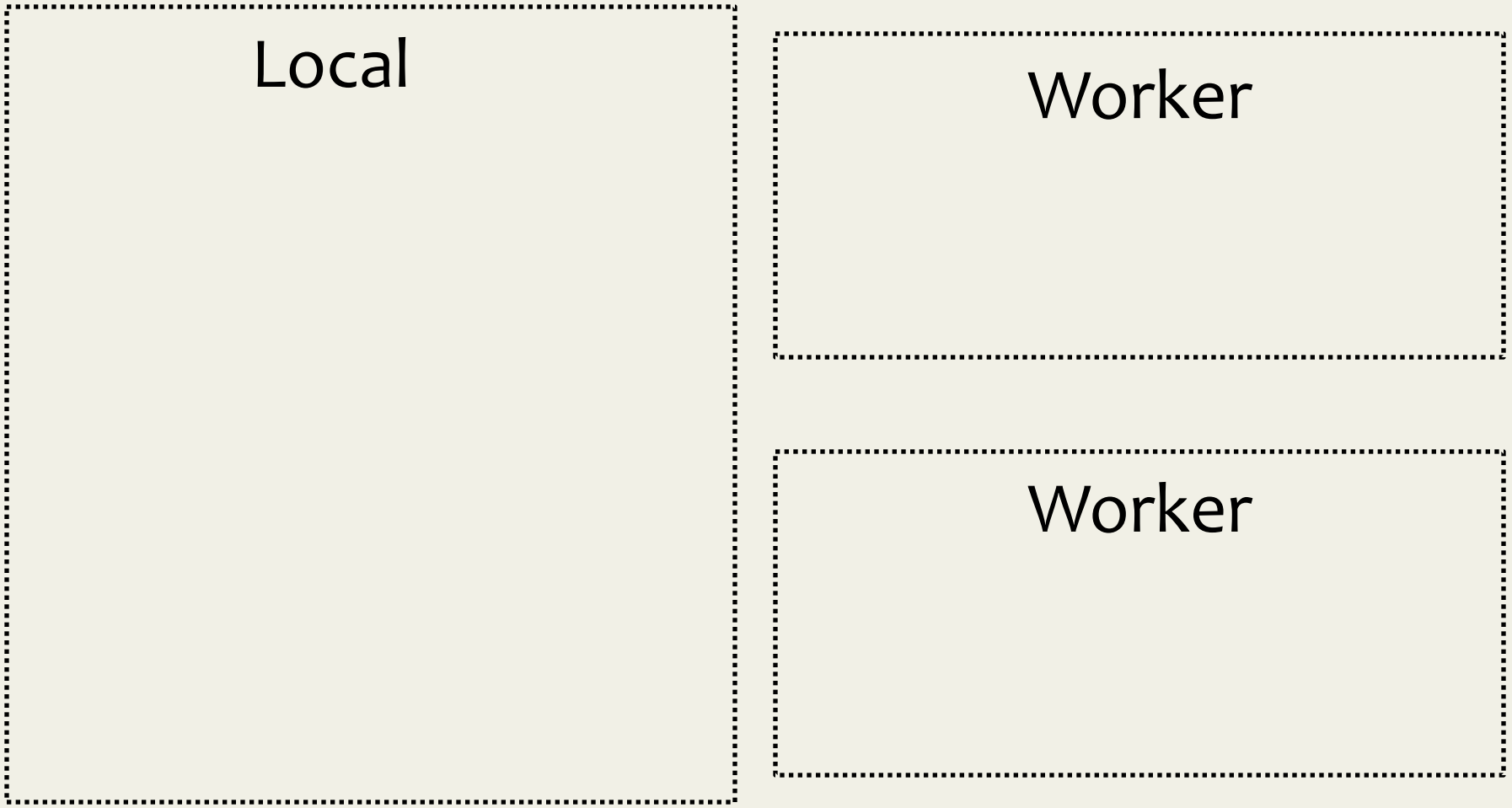


Minimize $\| Ax - b \|_2$

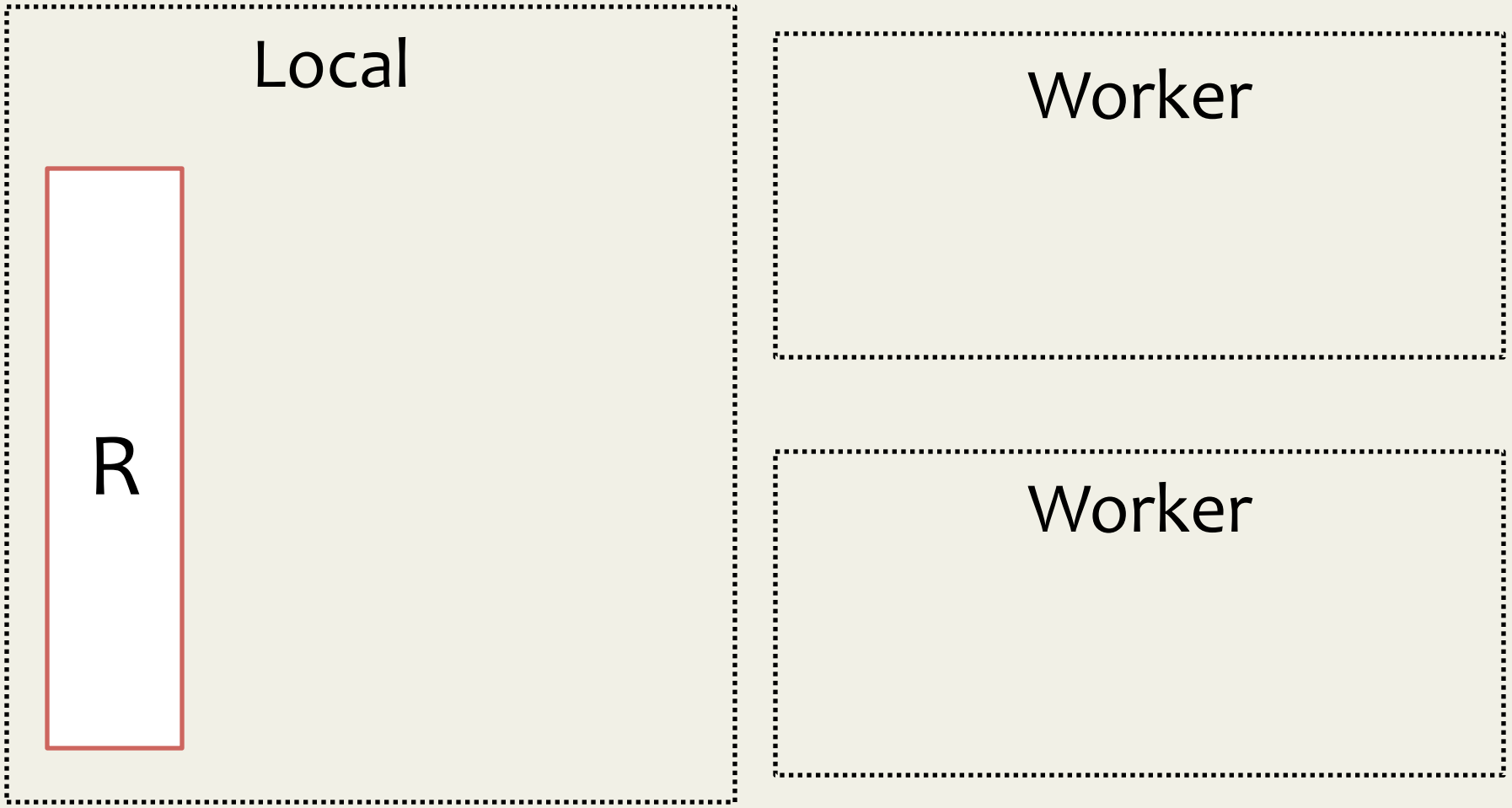
$$x = (A^T A)^{-1} A^T b$$


How does this work?

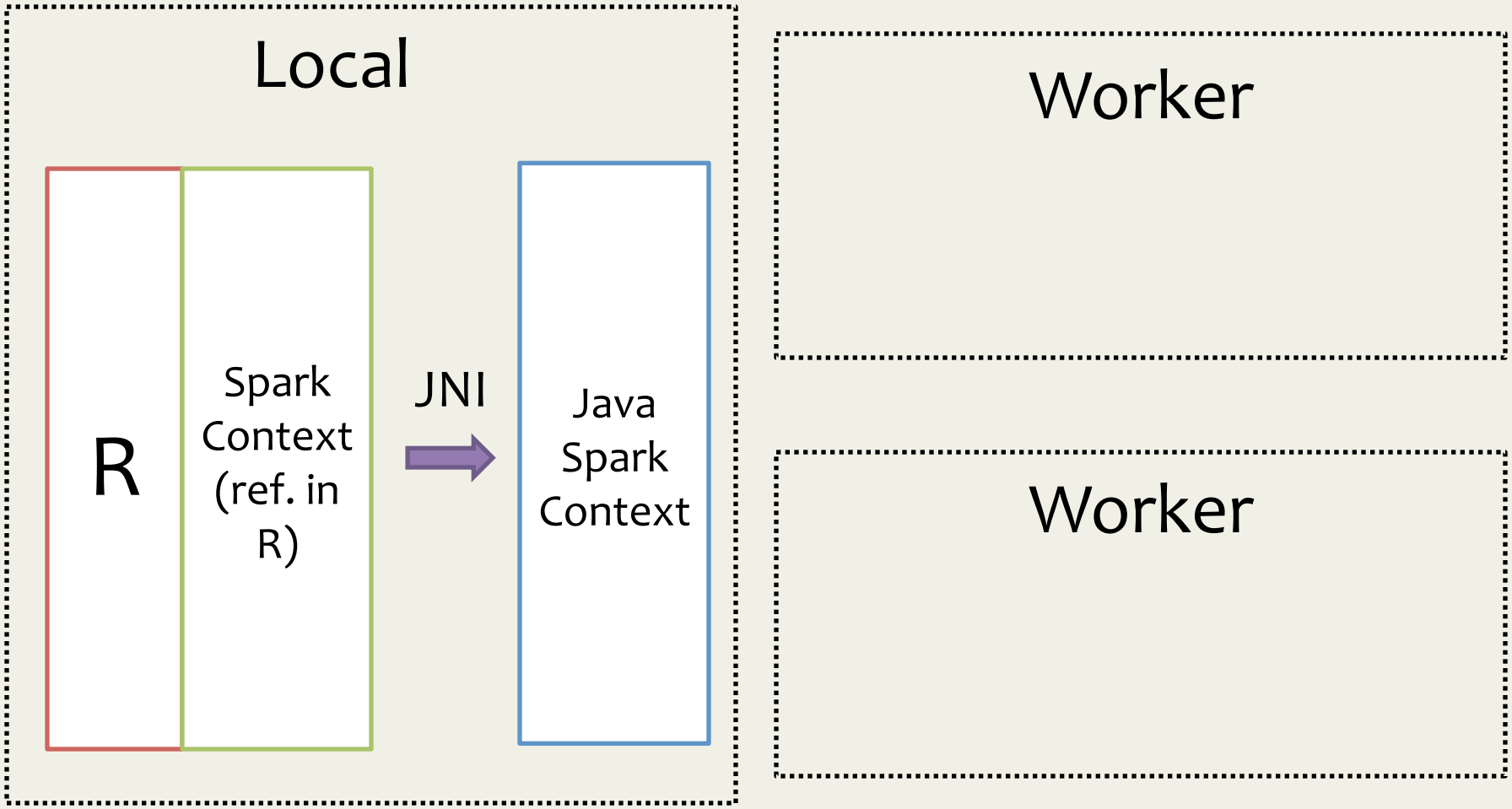
Dataflow



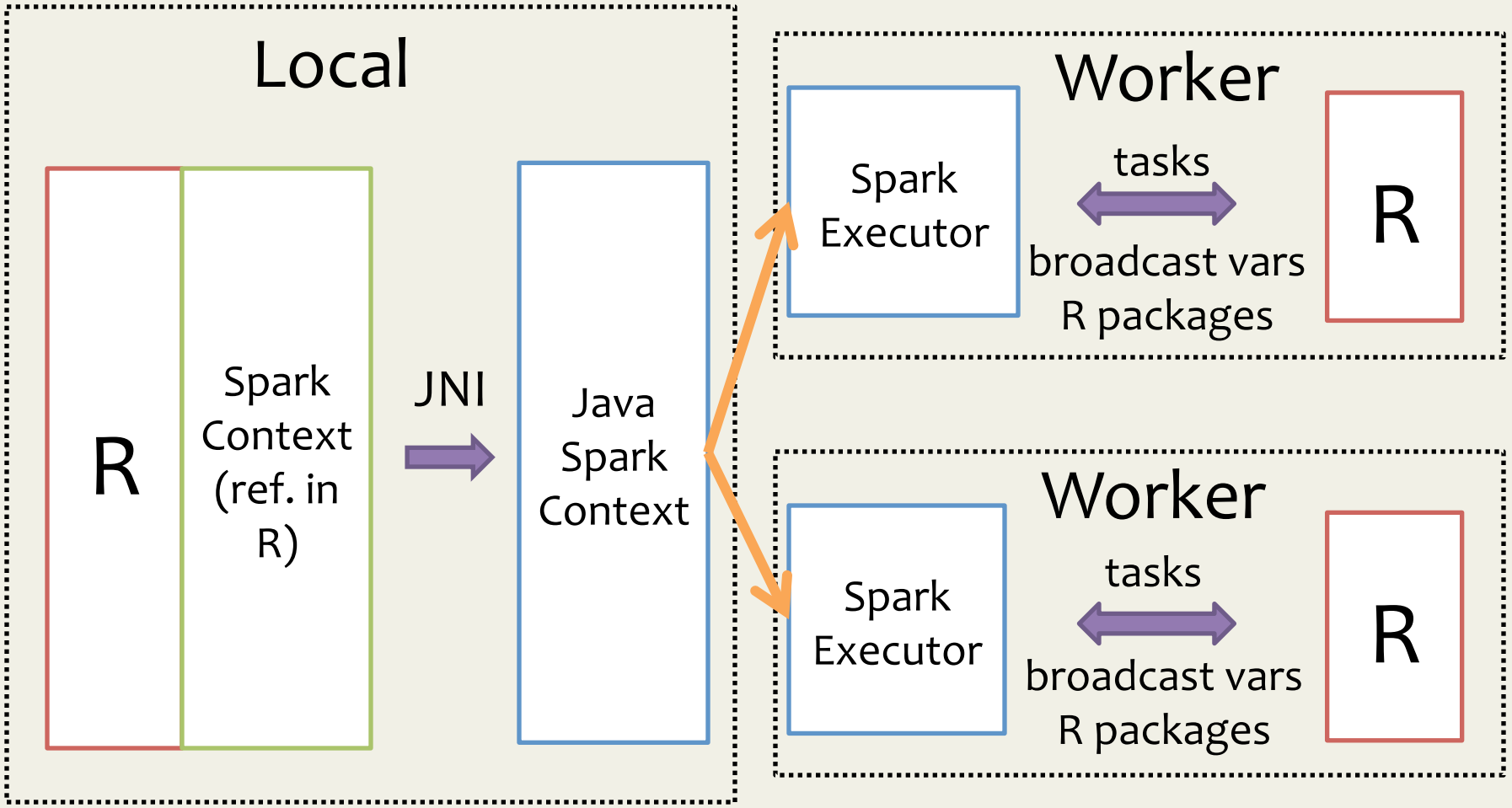
Dataflow



Dataflow

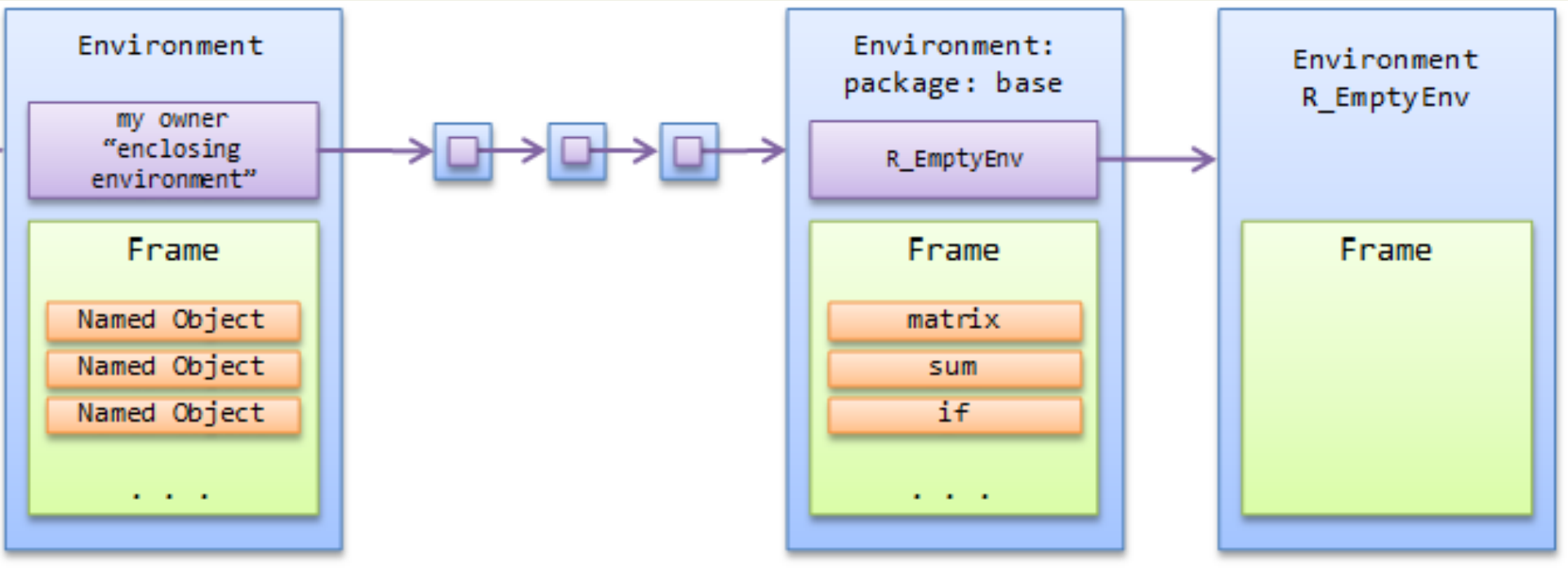

Local
R
Spark Context (ref. in R)
JNI
Java Spark Context
Worker
Worker

Dataflow


Local
R
Spark Context (ref. in R)
JNI
Java Spark Context
Worker
Spark Executor
tasks
broadcast vars
R packages
R
Worker
Spark Executor
tasks
broadcast vars
R packages
R


… Pipes?

Capturing Closures: Environments



From http://obeautifulcode.com/R/How-R-Searches-And-Finds-Stuff/

Serializing Closures: save()

save {base}

Save R Objects

Description

`save` writes an external representation of R objects to the specif
from the file at a later date by using the function `load` (or `data`

`save.image()` is just a short-cut for 'save my current workspac
`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!

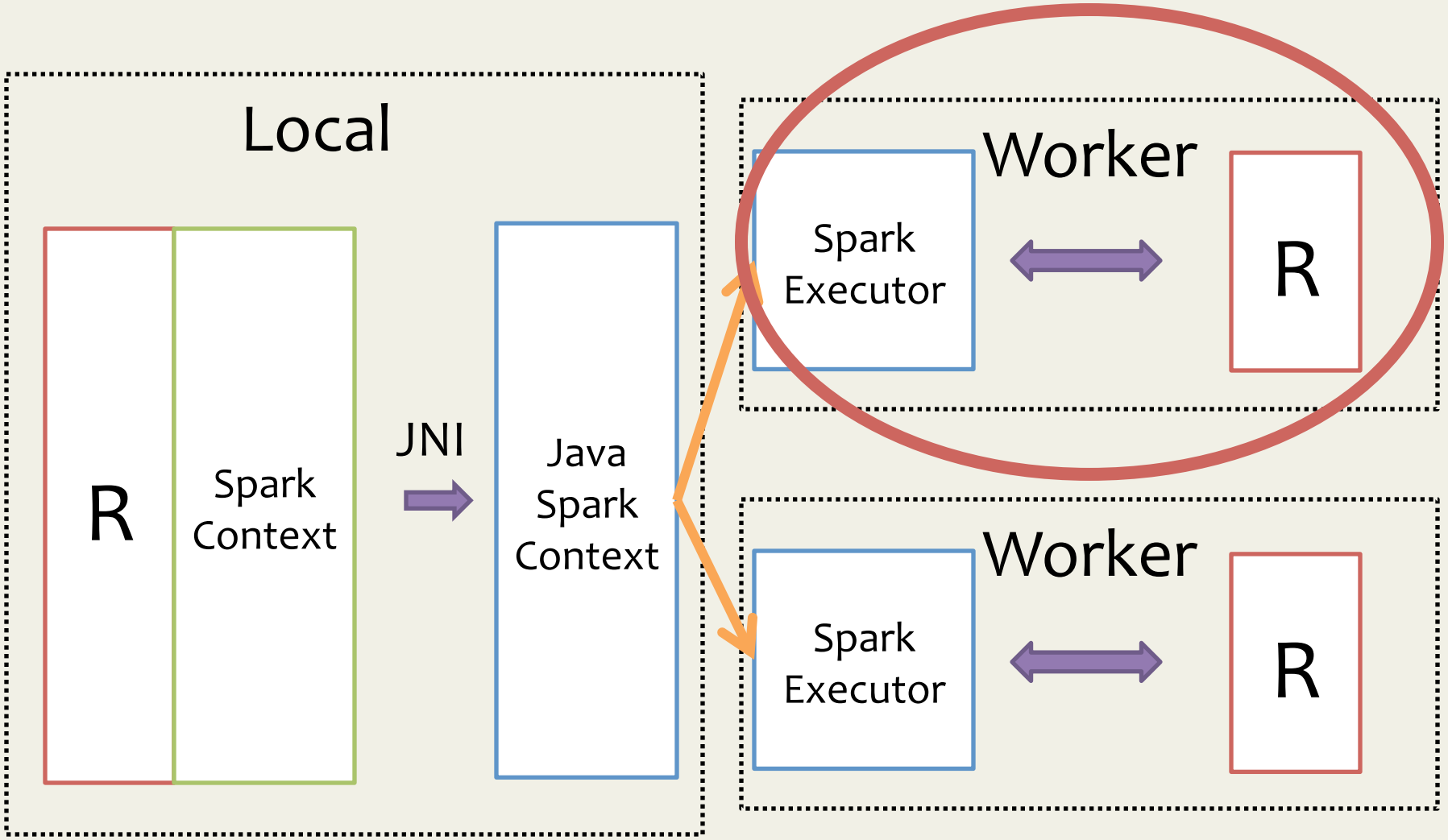
https://github.com/amplab-extras/SparkR-pkg

Shivaram Venkataraman shivaram@cs.berkeley.edu

Zongheng Yang zhyang@berkeley.edu

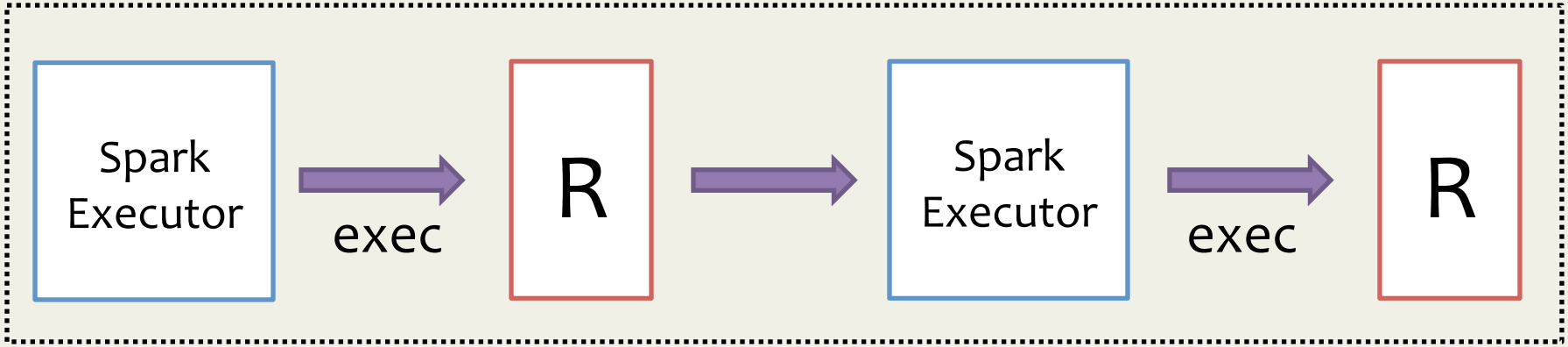
Spark User mailing list user@spark.apache.org

Dataflow: Performance?

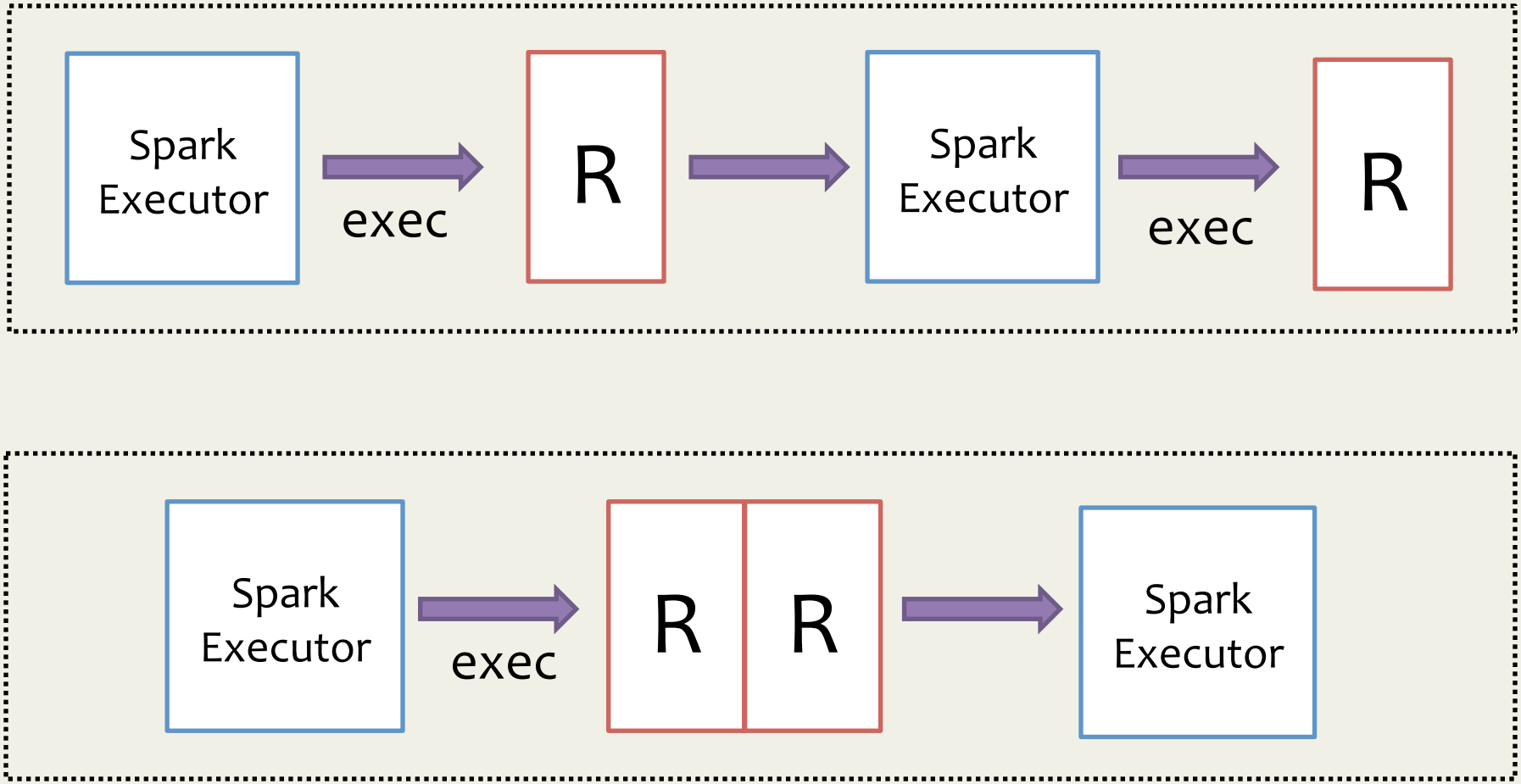


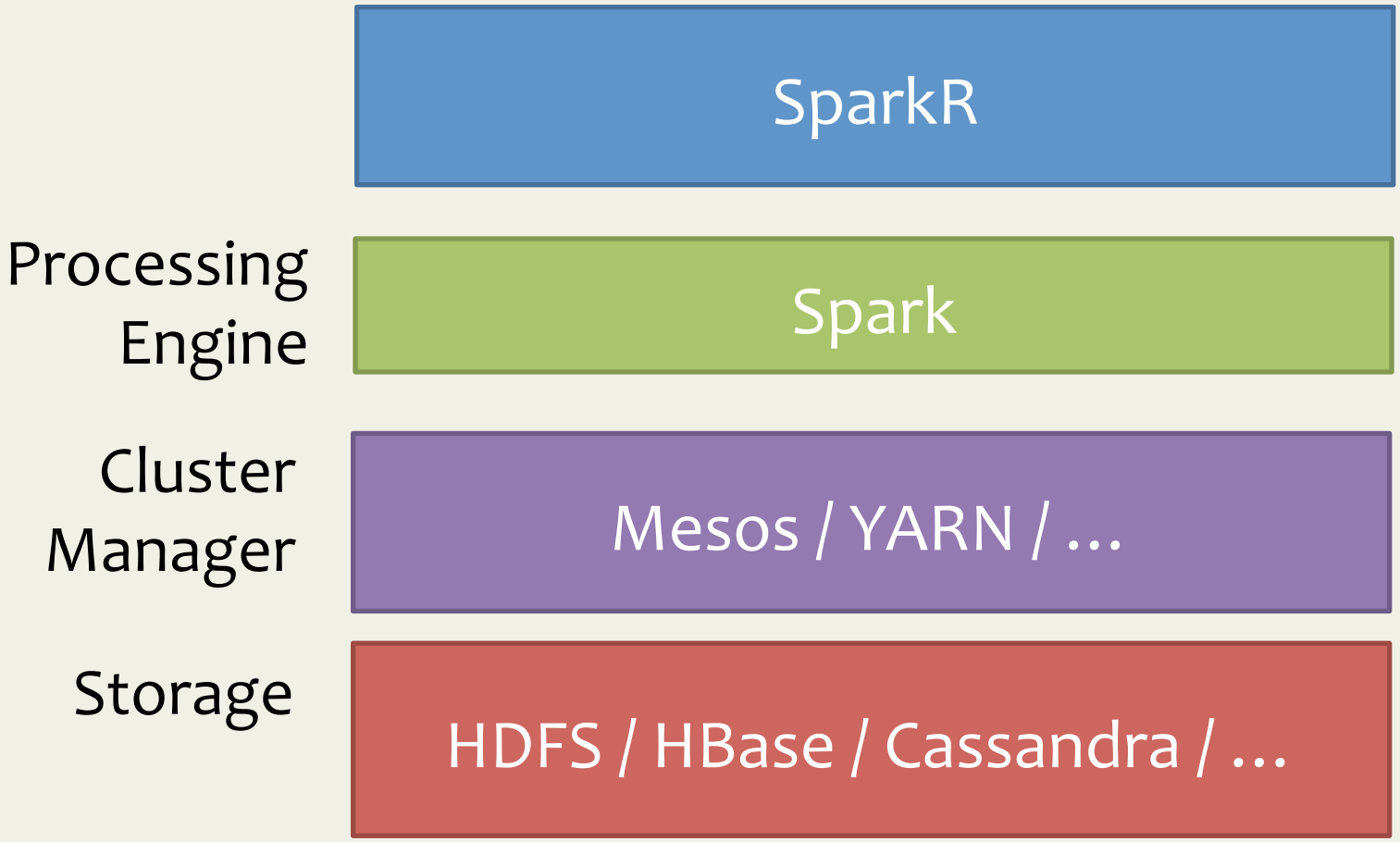
Pipeline the transformations!

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



Pipelined RDD




SparkR
Processing Engine
Spark
Cluster Manager
Mesos / YARN / …
Storage
HDFS / HBase / Cassandra / …

amplab-extras/SparkR-pkg

build passing

R frontend for Spark

Current | Build History | Pull Requests | Branch Summary

Build	34	Commit	c5bce07 (master)
State	Passed	Compare	aacd72657106...c5bce07ef517
Finished	23 days ago	Author	Zongheng Yang
Duration	9 min 37 sec	Committer	Zongheng Yang
Message	Merge pull request #30 from shivaram/string-tests		

Add tests for partitioning with string keys

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
}
```

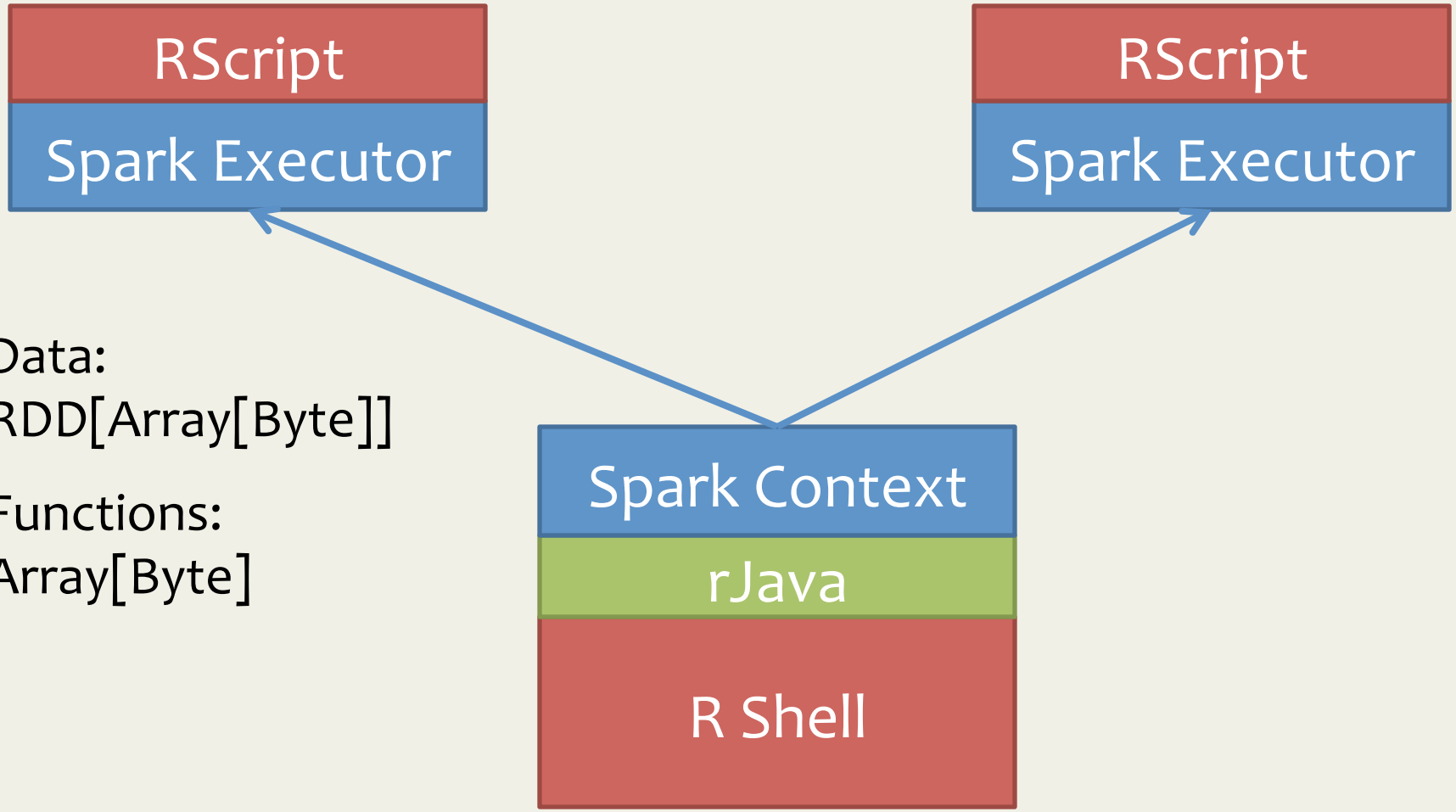
Example: Logistic Regression

```
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]