Big Data Processing with Apache Spark and R

Hao Lin (Purdue University)
Indianapolis, IN
Oct. 20, 2015

Part of the slides are modified from Shivaram's slides

About Me

Graduate student @ Purdue ECE

Research interests includes Parallel/Distributed Computing, Data Analytics Infrastructure, Cloud Computing, Compilers; Major contributor of SparkR project

Data/Cloud Internship experience @ Huawei, Google and Facebook

Outline

Overview

SparkR R-RDD APIs

DataFrame APIs

System Implementation & optimizations

Applications & Demos

Future Directions

Q & A

Motivation

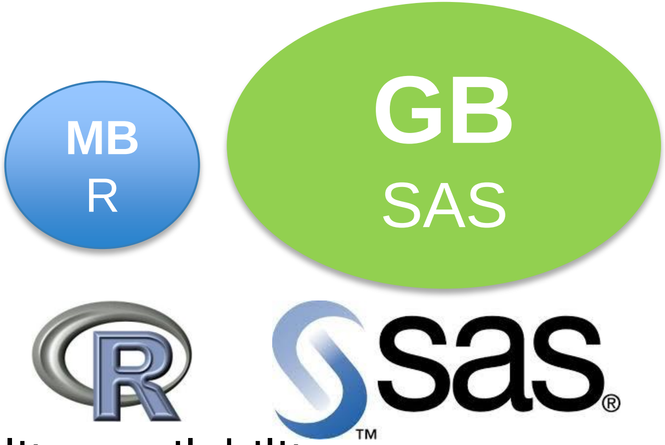
- Personal data for the internet
  - Query history
  - Click stream logging
  - Tweets (8TB per day)
  - Facebook (500TB per day)

- Machine Generated Data
  - Sensor networks
  - Genome sequencing
  - Physics experiments
  - Satellite imaging data
  - System performance logs in data center

Motivation

Huge demand for data analysts, statisticians & data scientists

Traditional tools work on
summary or sampled data



Properties of a good tool for large-scale data:

- Usability: stick to traditional semantics
- Performance: latency, throughput, scalability, availability
- Fault Tolerance

Motivation

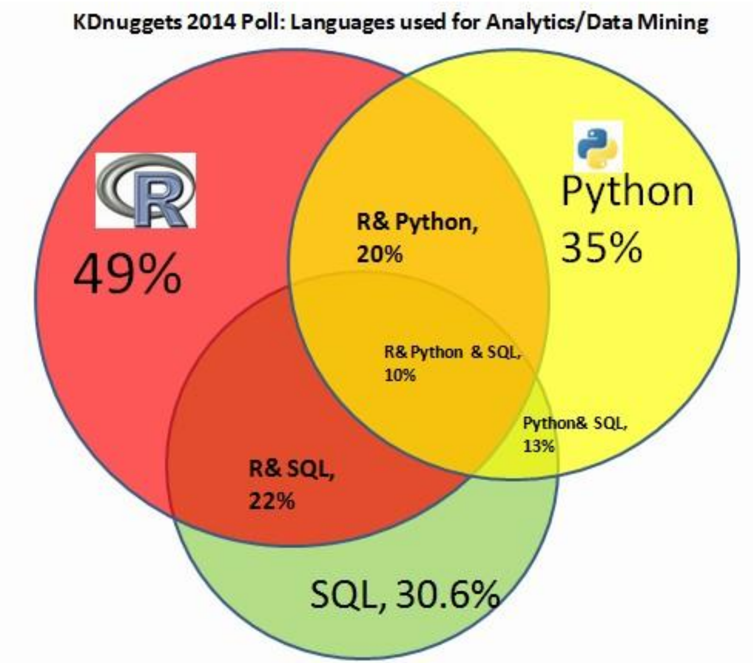
R ranked top among Analytics/Data mining tools in all recent years

Why R?:

- ✓ Data structures (list, array, DataFrame)
- ✓ Rich functionality in packages
- ✓ Graphic visualization
- ✓ Open source

Limitations of R?:

- x Single threaded
- x Limited memory



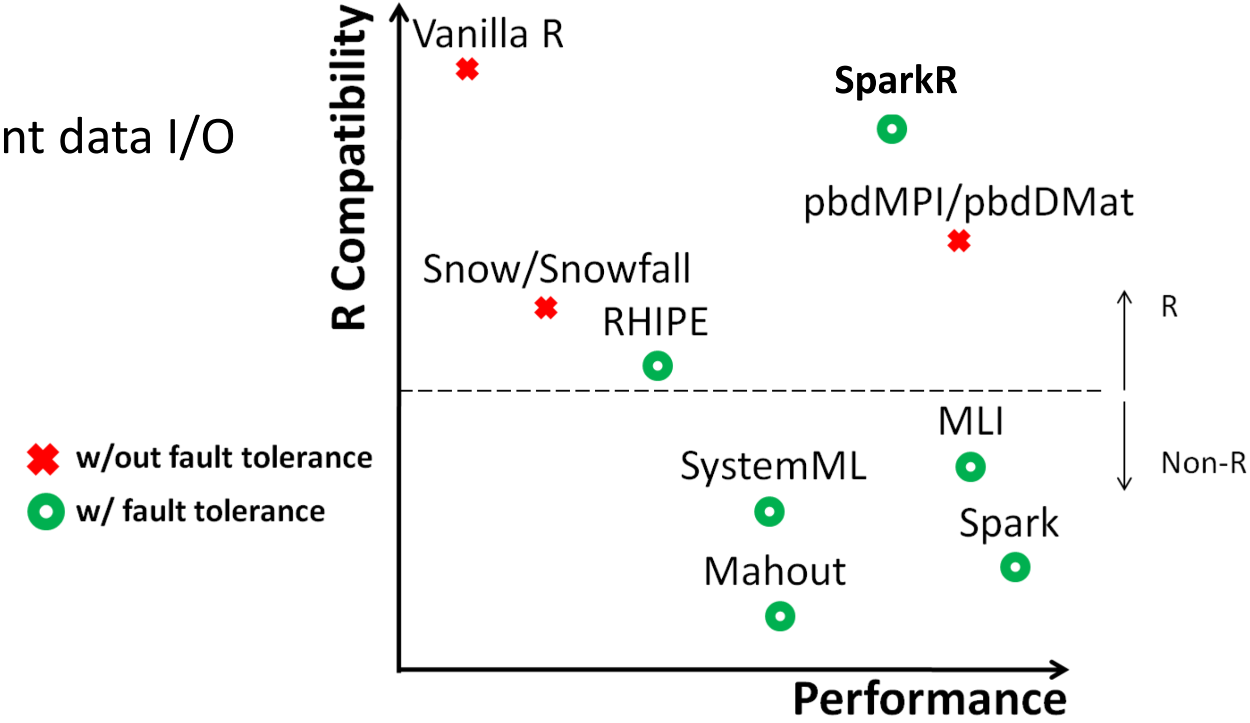
From: Survey by KDnuggets

http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html

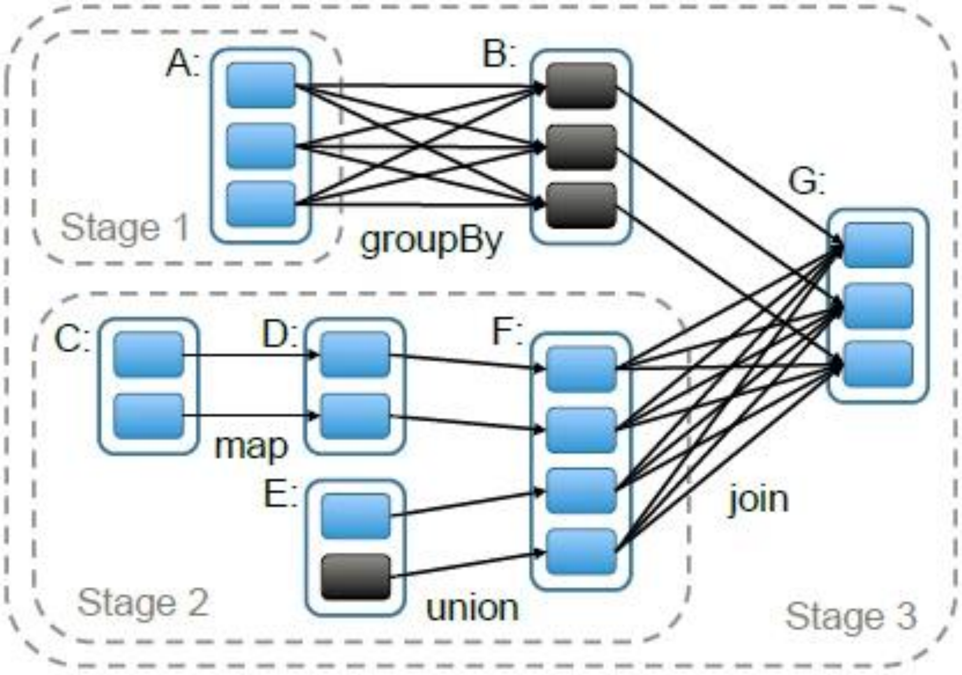
Landscape

Current Efforts:

- R compatibility
- Performance & efficient data I/O
- Fault tolerance



Spark Distributed Dataset (RDD) APIs



Developed at AMP lab, Berkeley

Flexible Programming Model

- ✓ DAG-like execution plan

Performance

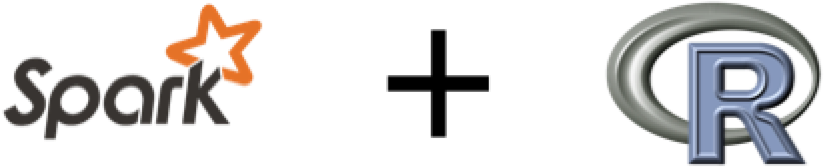
- ✓ In-memory
- ✓ Good for iterative algorithms

Resilient Data Sets

- ✓ Recover from loss and failures

From paper: Spark: Cluster Computing with Working Sets. HotCloud 2010.

SparkR



Fast

Scalable

Flexible

Statistical

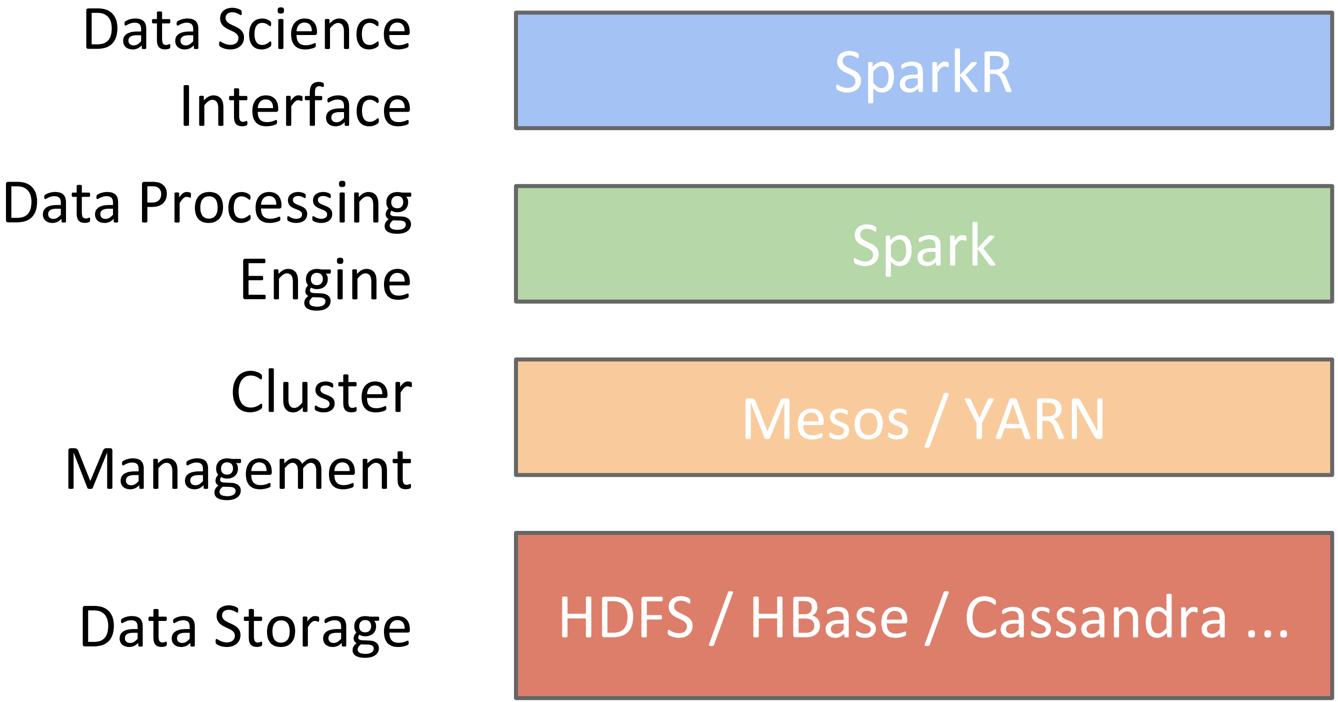
DataFrame

Packages

Plots

Part of the slides are modified from Shivaram’s slides

Software Stack



Data Science Interface — SparkR

Data Processing Engine — Spark

Cluster Management — Mesos / YARN

Data Storage — HDFS / HBase / Cassandra ...

Outline

Overview

SparkR R-RDD APIs

DataFrame APIs

System Implementation & optimizations

Applications & Demos

Future Directions

Q & A

SparkR R-RDD APIs

Considered as Distributed version of R List

Generation functions: `textFile, parallelize, ...`
Transformation functions: `lapply, filter, sampleRDD, ...`
Persistence function: `cache, persist, ...`
Action functions: `reduce, collect, ...`
Paired Value Shuffle functions: `groupByKey, reduceByKey, ...`
Binary Functions: `unionRDD, cogroup, join, ...`
Output Functions: `saveAsTextFile, saveAsObjectFile, ...`

R-RDD Example: Word Count

```
library(SparkR)
lines <- textFile(sc, “hdfs://my_text_file”)
```

R-RDD Example: Word Count

```
library(SparkR)
lines <- textFile(sc, “hdfs://my_text_file”)
words <- flatMap(lines,
                 function(line) {
                   strsplit(line, " ")[[1]]
                 })
wordCount <- lapply(words,
                    function(word) {
                      list(word, 1L)
                    })
```

R-RDD Example: Word Count

```
library(SparkR)
lines <- textFile(sc, “hdfs://my_text_file”)
words <- flatMap(lines,
                 function(line) {
                   strsplit(line, " ")[[1]]
                 })
wordCount <- lapply(words,
                    function(word) {
                      list(word, 1L)
                    })
counts <- reduceByKey(wordCount, "+", 2L)
output <- collect(counts)
```

Outline

Overview

SparkR R-RDD APIs

DataFrame APIs

System Implementation & optimizations

Applications & Demos

Future Directions

Q & A

Data Frames

More structured data in Tables

Data source like CSV, JSON, JDBC, …

Want to use your favorite package “dplyr” ?

DataFrame type in SparkR

Embeded SQL in R

Why R on Spark Data Frame?

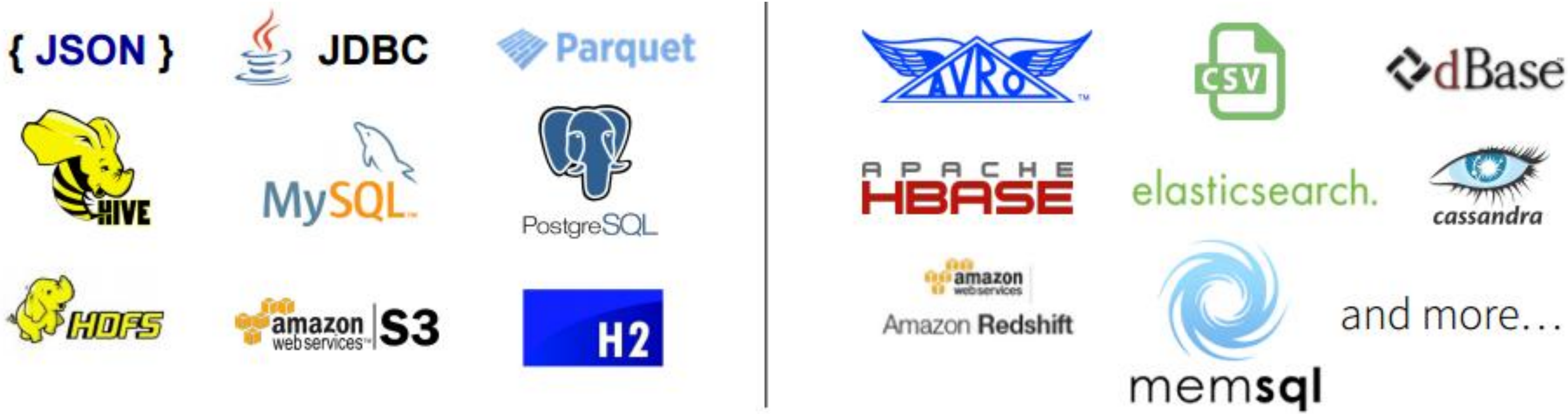
Imposes a schema on RDDs

Query Optimizer, Code Gen

Rich DataSources API

```
sqlCtx.table("people")\
.groupBy("name")\
.agg("name", avg("age")) \
.collect()
```

more sources at http://spark-packages.org/



DataFrame APIs

Column access using ‘$’ or ‘[ ]’ just like in R

dplyr-like DataFrame manipulation:

- filter
- groupBy
- summarize
- mutate

Access to external R packages that extend R syntax

DataFrame APIs

Filter -- Select some rows

```
filter(df, df$col1 > 0)
```

Project -- Select some columns

```
df$col1 or df[“col”]
```

DataFrame APIs

Aggregate -- Group and Summarize data

```
groupDF <- groupBy(df, df$col1)
```

```
agg(groupDF, sum(groupDF$col2), max(groupDF$col3))
```

Sort -- Sort data by a particular column

```
sortDF(df, asc(df$col1))
```

Column Average using RDD

```
peopleRDD <- textFile(sc, “people.txt”)

lines <- flatMap(peopleRDD,
                 function(line) {
                   strsplit(line, ", ")
                 })

ageInt <- lapply(lines,
                 function(line) {
                   as.numeric(line[2])
                 })

sum <- reduce(ageInt, function(x, y) {x+y})
avg <- sum / count(peopleRDD)
```

Name	Age
Andy	20
Hao	25

Column Average using DataFrame

```
# JSON File contains two columns age, name
df <- jsonFile(“people.json”)

avg <- select(df, avg(df$age))
```

Outline

Overview

SparkR R-RDD APIs

DataFrame APIs

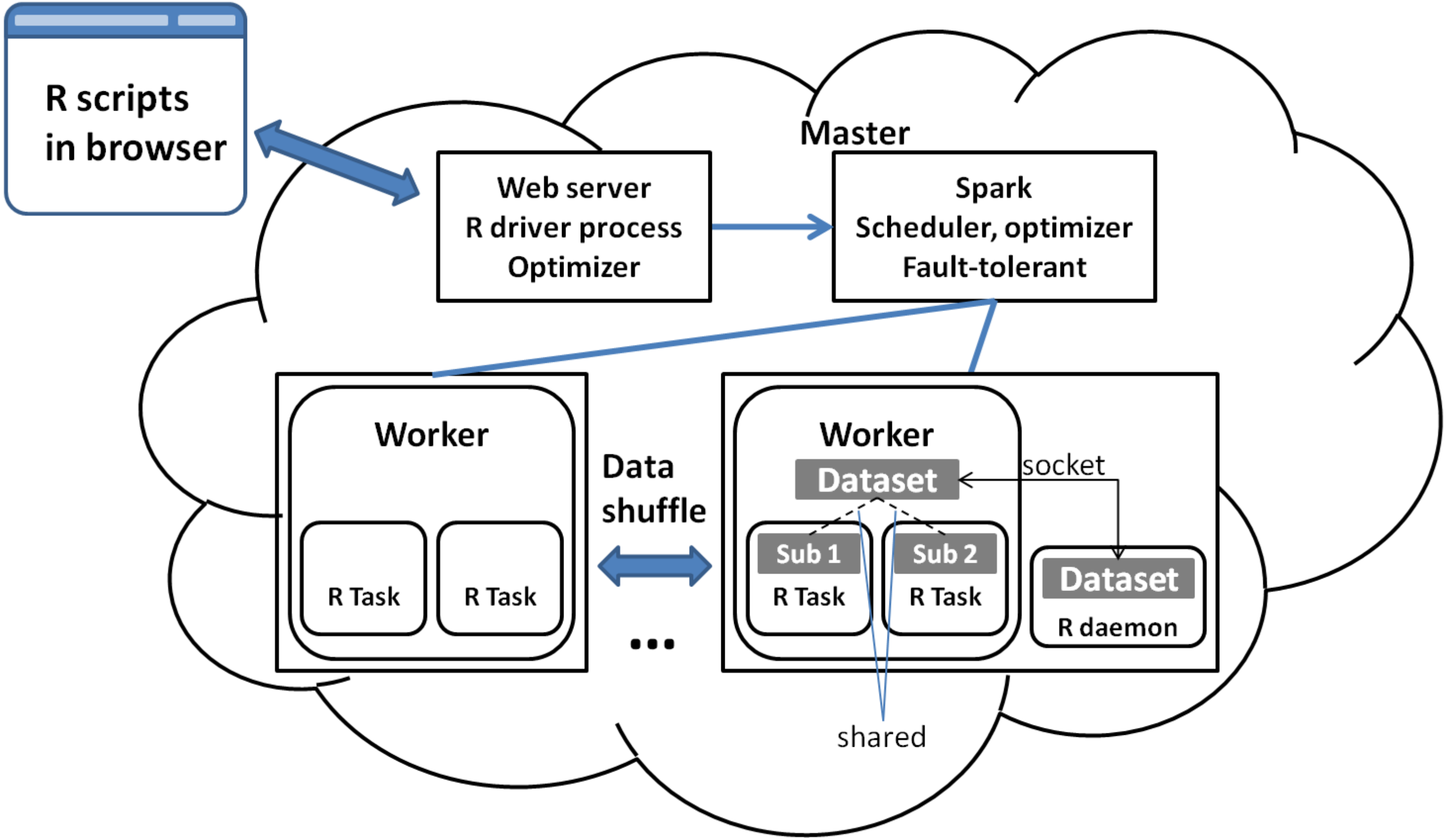
System Implementation & optimizations

Applications & Demos

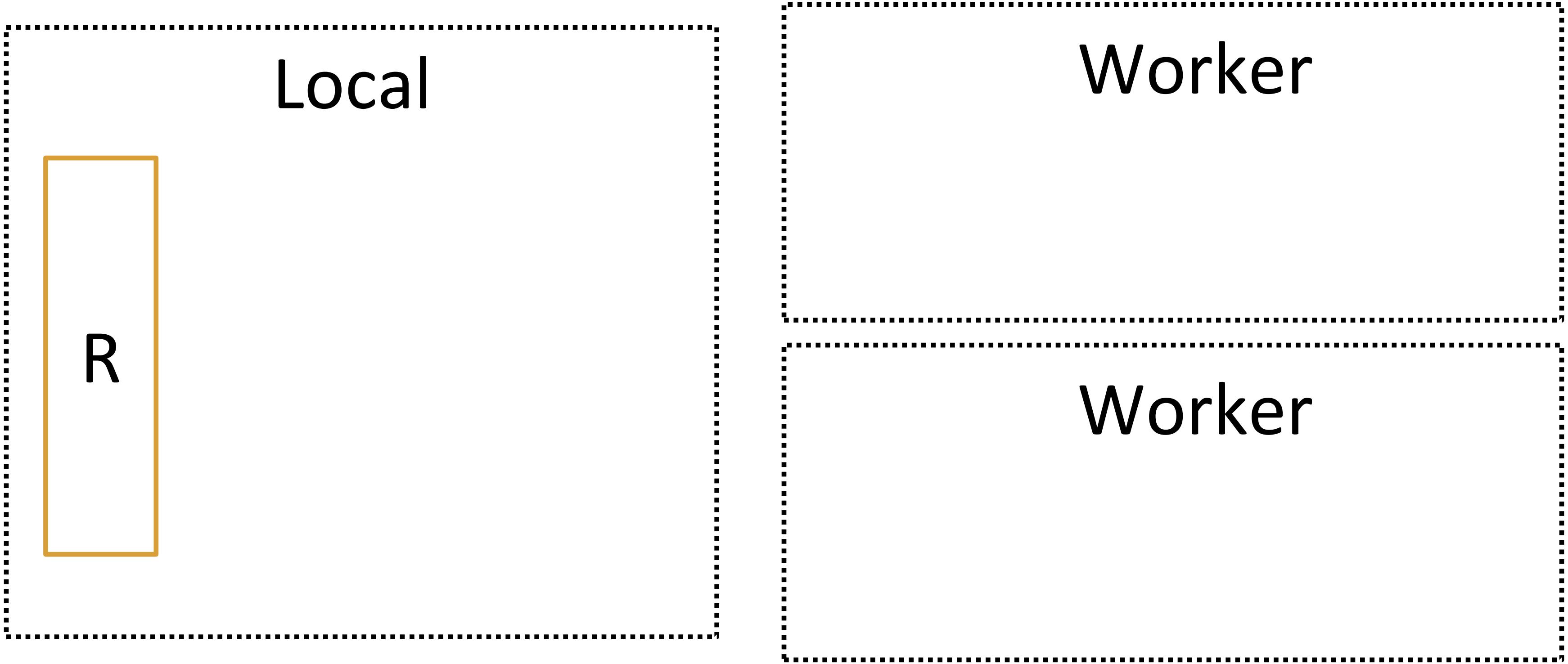
Future Directions

Q & A

Design and Implementation



Design and Implementation



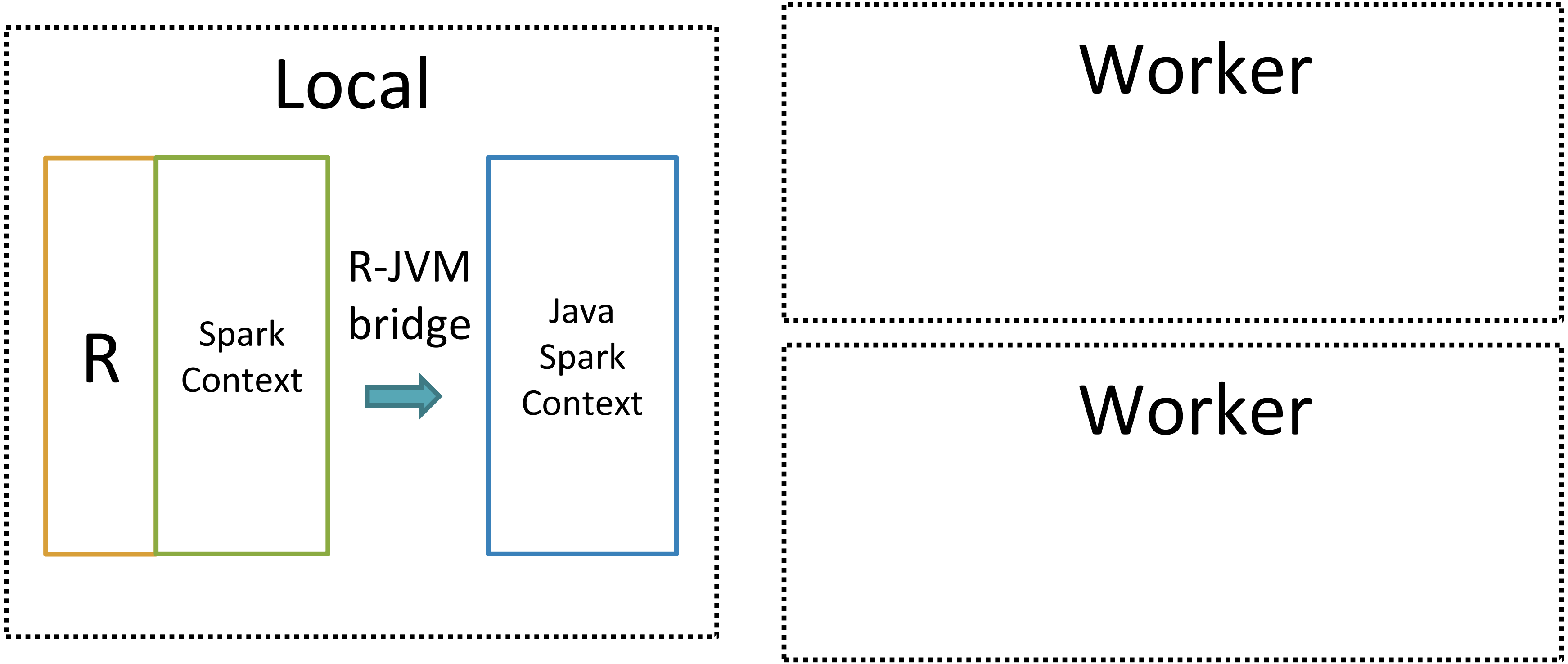
Local

R

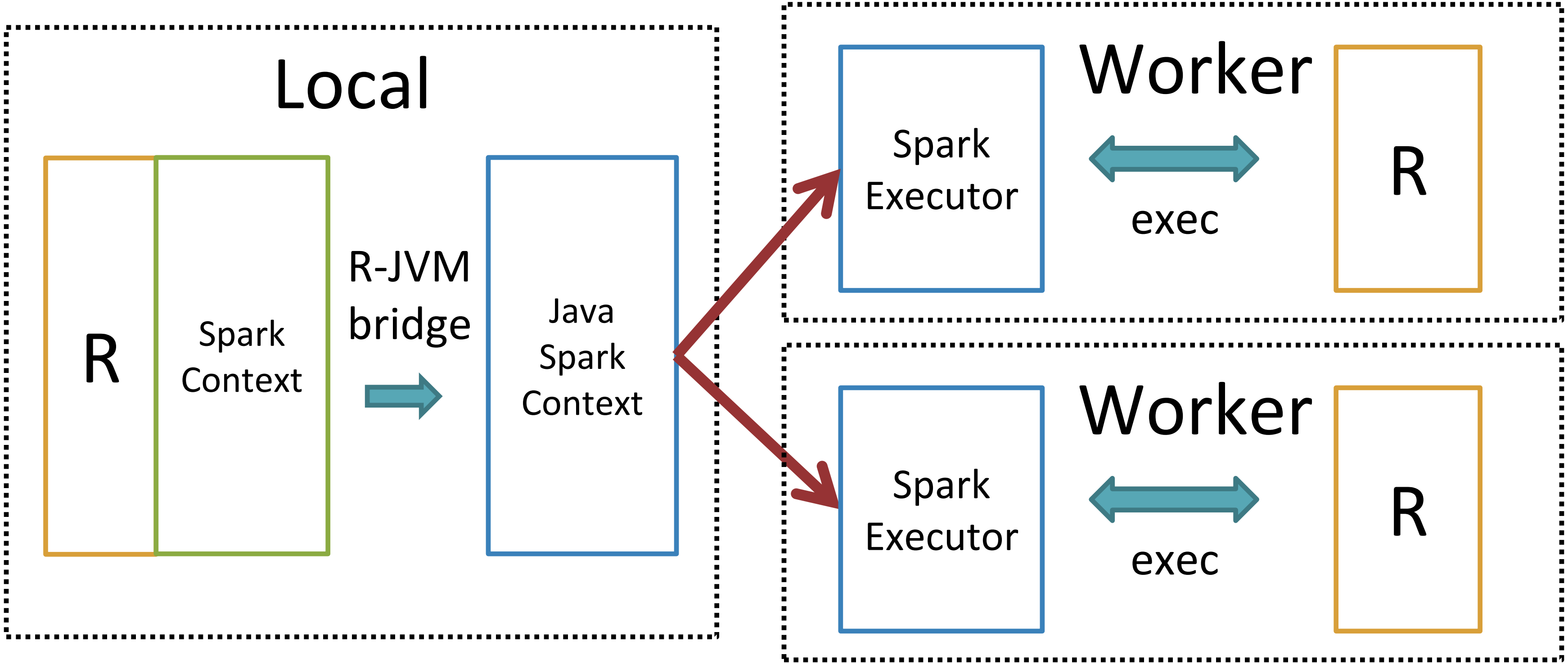
Worker

Worker

Design and Implementation

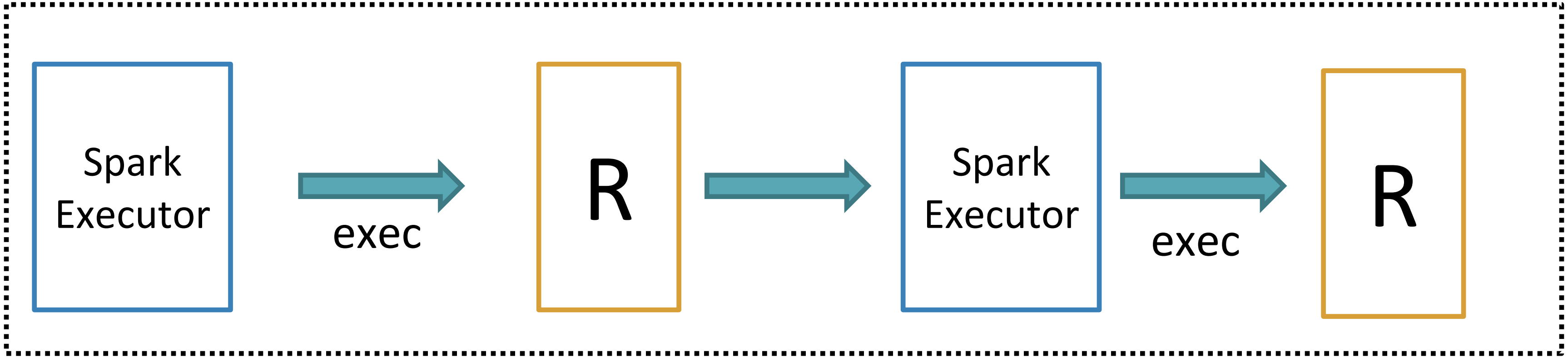


Design and Implementation

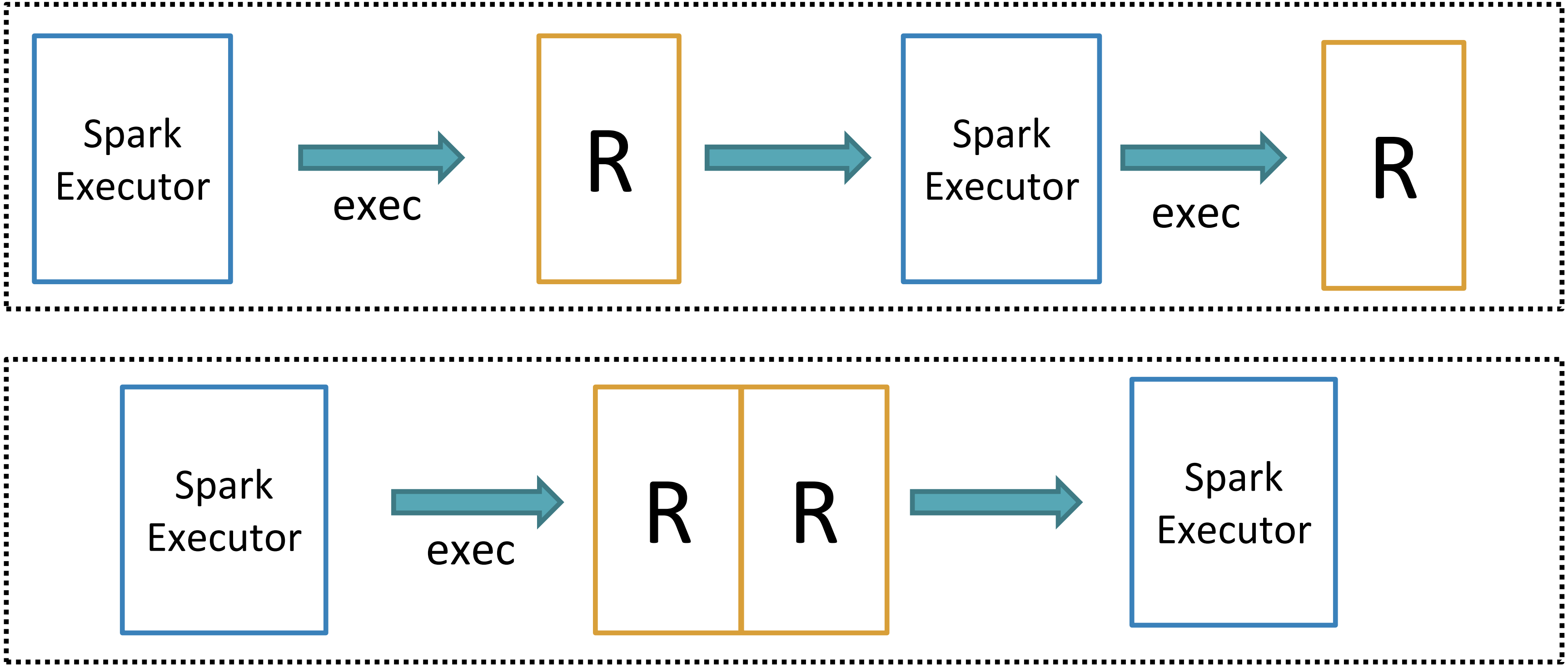


Pipelined RDD

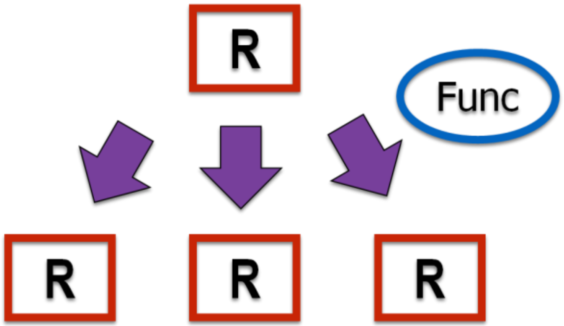
```
words <- flatMap(lines,…)
wordCount <- lapply(words,…)
```



Pipelined RDD



Correctly Capture the Function Closure



```
z <- 1
func1 <- function() {
    y <- 2
    func2 <- function(x) {
        x + y + z
    }
}
```

We need to ship the function closure to worker nodes

R has scoping rule to search for the values of free variables defined outside the function

We need to ship the functions together with the values of free variables in its environment

lapply is Slow?

```
lapply(L, f) {
    len <- length(L)
    Lout <- alloc_veclist(len)
    for(i in 1:len) {
        item <- L[[i]]
        Lout[[i]] <- f(item)
    }
    return(Lout)
}
```

R is a interpreted language

Looping-over-data execution of lapply in R

Pick element one by one, and interpret f() many times

Solution: Use *lapplyPartition*

```
grad.func <- function(yx) {
    y <- yx[1]
    x <- yx
    x[1] <- 1 #modify the 1st column
    logit <- 1/(1 + exp(-sum(theta*x)))
    (y-logit) * x
}

lapply(points, grad.func)
```

Write vector function

```
gradient <- function(partition) {
    partition = partition[[1]]
    Y <- partition[, 1]
    X <- partition[, -1]

    dot <- X %*% w
    logit <- 1 / (1 + exp(-Y * dot))
    grad <- t(X) %*% ((logit - 1) * Y)
    list(grad)
}

lapplyPartition(points, gradient)
```

Outline

Overview

SparkR R-RDD APIs

DataFrame APIs

System Implementation & optimizations

Applications & Demos

Future Directions

Q & A

Environment Setup

Install R & RStudio: http://cran.r-project.org/, http://www.rstudio.com/products/RStudio/

Java (Scala) & Maven: Java 6+, Maven 3.0.4+

Install Spark

- For latest Spark 1.4, SparkR is in the release https://github.com/apache/spark
  - build SparkR by Maven, following https://github.com/apache/spark/tree/master/R
- For Spark 1.3 or maybe before: download Spark from https://spark.apache.org/downloads.html
  - Also install SparkR package in https://github.com/amplab-extras/SparkR-pkg

Other packages like HDFS if necessary

Environment Setup

Docker: https://registry.hub.docker.com/u/beniyama/sparkr-docker

Amazon EC2: https://github.com/amplab-extras/SparkR-pkg/wiki/SparkR-on-EC2

Multiple node (Distributed mode) with HDFS

Install Hadoop: http://goo.gl/OXt1mC

Spark Standalone: https://spark.apache.org/docs/latest/spark-standalone.html

YARN: https://spark.apache.org/docs/latest/running-on-yarn.html

Quick Start

Start RStudio

Desktop version: start RStudio application

Server version: open web browser with `<your host>:8787`

Spark Context Initialization

Setup SPARK_HOME: alternatively we can store in ~/.Renviron s.t. it will not execute every time

```
# Set this to where Spark is installed
Sys.setenv(SPARK_HOME="/home/sparkr/workspace/spark")
# This line loads SparkR from the installed directory,
.libPaths(c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib"), .libPaths()))
```

Load SparkR package: `library(SparkR)`

Init Spark Context: `sc <- sparkR.init(master="local[n]")`

Quick Start

Also, you can always use terminal

For Spark 1.4: `cd $SPARK_HOME; ./bin/sparkR --master “local[n]”`

For Spark 1.3 with SparkR-pkg: `cd SparkR-pkg; ./sparkR --master “local[n]”`

Spark Context will automatically be created, call `sc`

Pi Example

Logistic Regression

Predicting Customer Behavior

Demo from Chris Freeman from Alteryx

3 datasets:

- Transactions
- Demographic Info Per Customer
- DM Treatment Sample

How do we decide who to send the offer to?

Predicting Customer Behavior

Demo from Chris Freeman from Alteryx

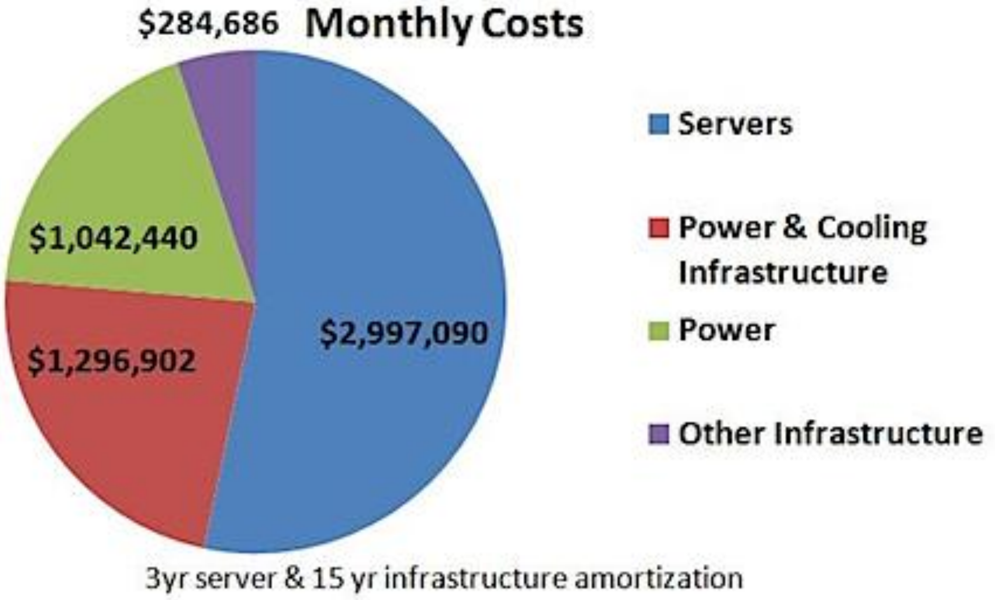
Use the DataFrame API to load, prepare and combine all 3 datasets and create training and estimation sets.

Use R’s glm method to train a logistic regression model on the treatment sample

Profit!

Cloud Data Center Scheduling

- Reduce number of physical resources used:
    - Capital expense (CAPEX)
    - Operational expense (OPEX)
- Increase resource utilization
- Meet the SLO



Cost of 45,000 machines

From: James Hamilton
http://perspectives.mvdirona.com/

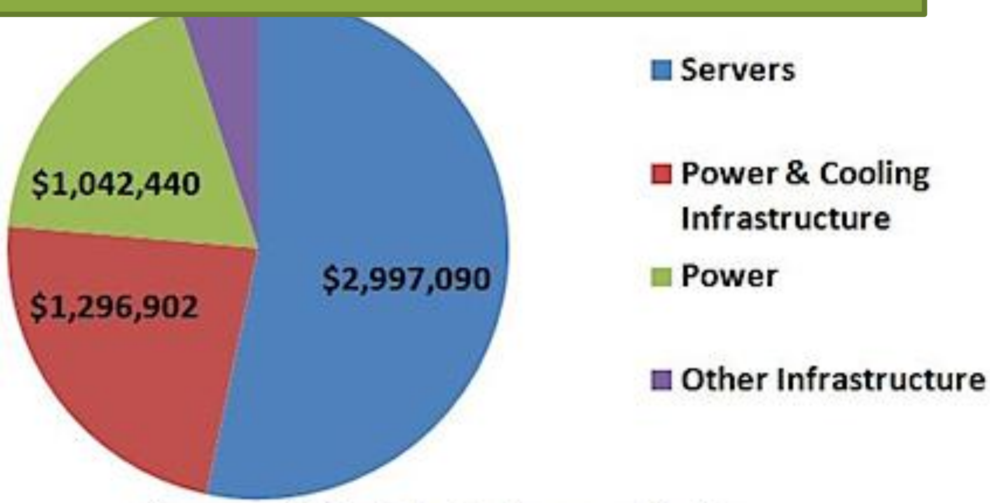
Cloud Data Center Scheduling

- R
- I
- Meet the SLO

Historical system performance metrics can provide clues about efficient resource allocation.

Cost of 45,000 machines

From: James Hamilton
http://perspectives.mvdirona.com/



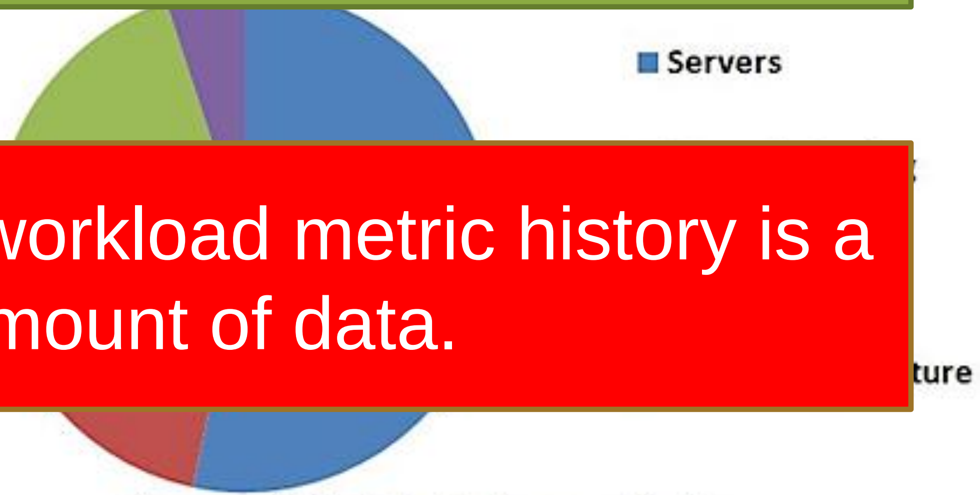
3yr server & 15 yr infrastructure amortization

Cloud Data Center Scheduling

- ❑ R
- ❑ I
- ❑ Meet the SLO

Historical system performance metrics can provide clues about efficient resource allocation.

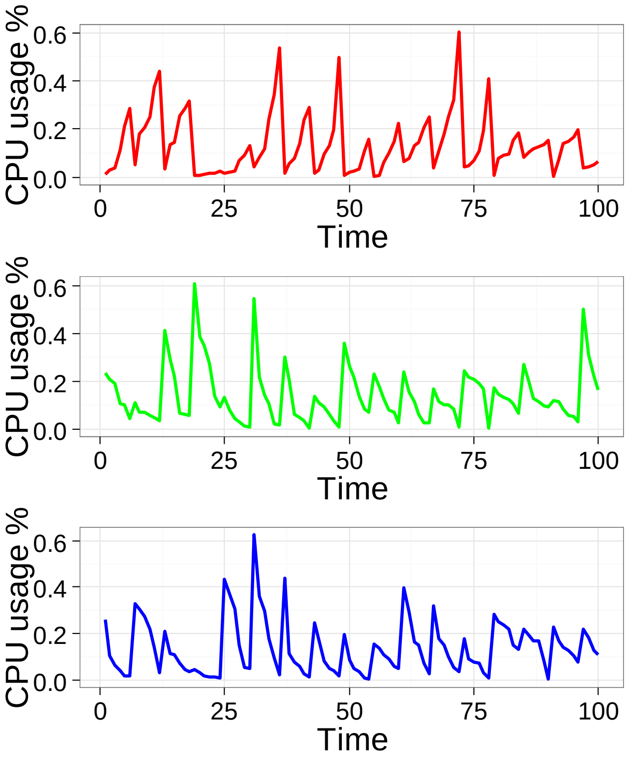
Machine generated workload metric history is a Large amount of data.

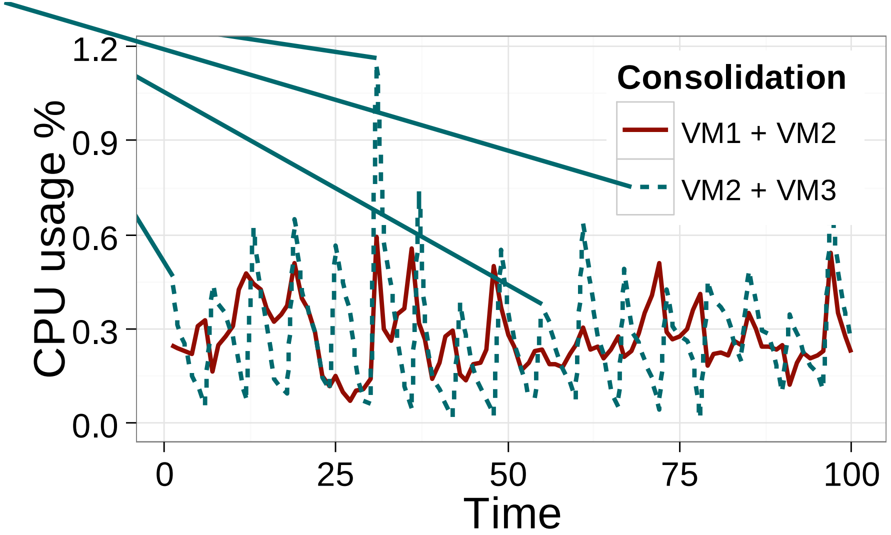


3yr server & 15 yr infrastructure amortization

From: James Hamilton
http://perspectives.mvdirona.com/

Cloud Data Center Scheduling

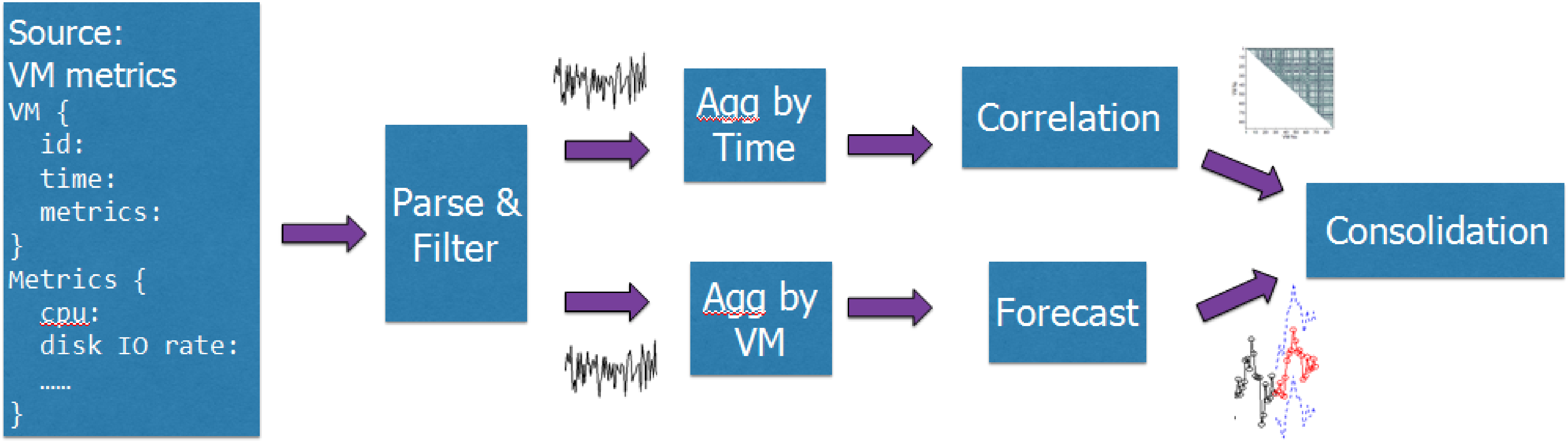




Cor(VM1, VM2) = -0.4735
Cor(VM2, VM3) = 0.5214

sd(VM1 + VM2) = 0.1056
sd(VM2 + VM3) = 0.1908

Cloud Data Center Scheduling



Source:
VM metrics

```
VM {
  id:
  time:
  metrics:
}
Metrics {
  cpu:
  disk IO rate:
  ……
}
```

Parse & Filter

Agg by Time

Agg by VM

Correlation

Forecast

Consolidation

Use SARIMA model in R

Outline

Overview

SparkR R-RDD APIs

DataFrame APIs

System Implementation & optimizations

Applications & Demos

Future Directions

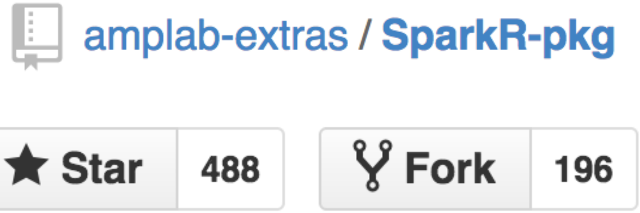
Q & A

Current Status and Community

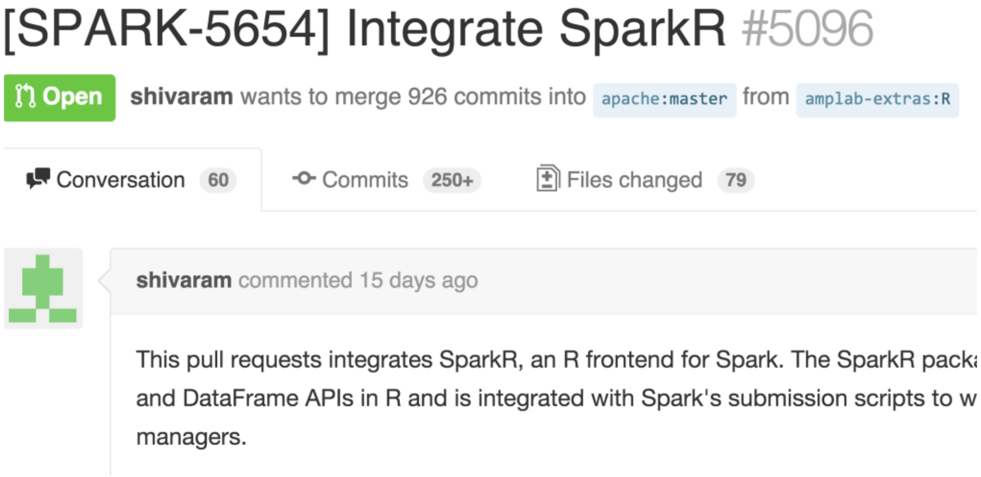
Originated in AMPLab

About 20 contributors
AMPLab, Alteryx,
Databricks, Intel

Now merged in Spark 1.4!



amplab-extras / **SparkR-pkg**



[SPARK-5654] Integrate SparkR #5096

shivaram wants to merge 926 commits into `apache:master` from `amplab-extras:R`

shivaram commented 15 days ago

This pull requests integrates SparkR, an R frontend for Spark. The SparkR packa
and DataFrame APIs in R and is integrated with Spark's submission scripts to w
managers.

Future Direction

High-level APIs to do Machine learning

Example: glm, kmeans

Pipelines with featurizers, learning

Tokenize → TF-IDF → LogisticRegression

Extended models, summary methods

Future Direction

APIs for Streaming, Time series analysis

Try a SparkR Streaming implementation:
https://github.com/hlin09/spark/tree/SparkR-streaming
https://github.com/hlin09/SparkR-pkg/tree/SparkR-streaming

Distributed matrix operations

<Your SparkR use case ?>

Reference and Guide

Starter & RDD: https://github.com/amplab-extras/SparkR-pkg/wiki/SparkR-Quick-Start
Data Frame: http://people.apache.org/~pwendell/spark-releases/latest/sparkr.html
Chris’s demo: https://github.com/cafreeman/Demo_SparkR

Other Resources

Download VMware Player:
https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/7_0

Download VM image:
http://web.ics.purdue.edu/~lin116/sparkr-ubuntu-1204-interface15.zip

Examples & Datasets:
http://web.ics.purdue.edu/~lin116/examples.tar.gz
http://web.ics.purdue.edu/~lin116/data.tar.gz

Thanks!
Questions?

Hao Lin (Purdue University): hlin09pu@gmail.com