

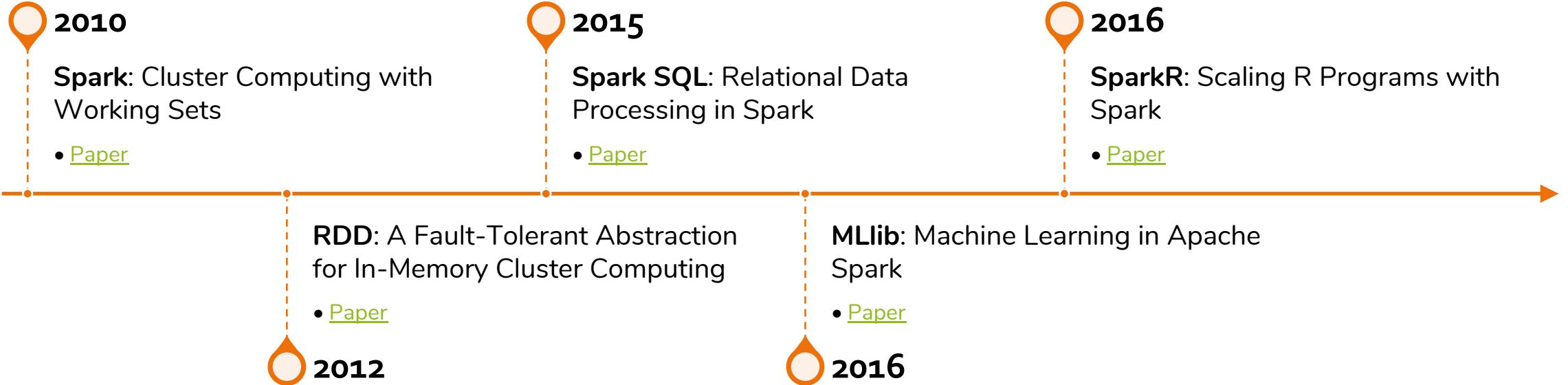


Data Frames /  
sparklyr

## Quais os problemas?

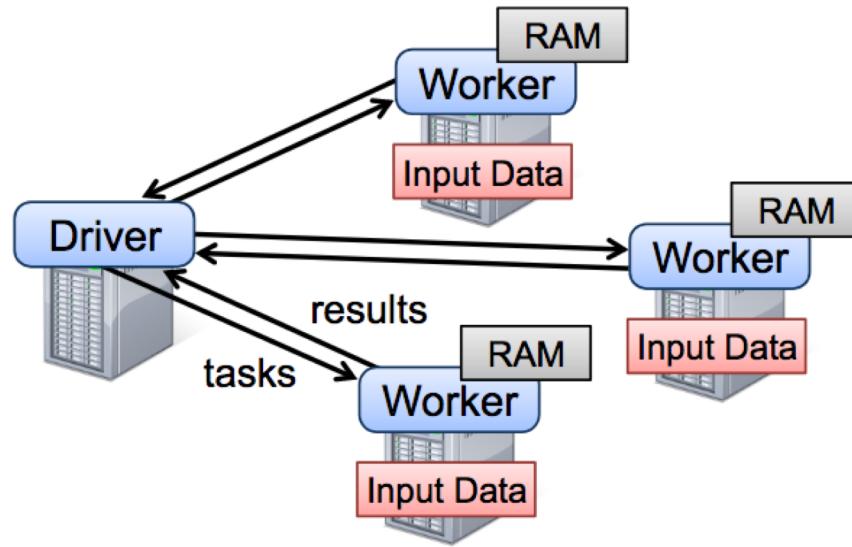
- Modelo de programação muito próximo do modelo de execução
- Programas escritos através de
  - Map, Combiner, Partitioner, Shuffle, Sort, Reduce
- Alternativas (Crunch, Hive, Pig, Cascading, etc) compiladas para o modelo de execução
- IO excessivo
  - Escrita em disco entre fases, ordenação obrigatória entre Map e Reduce
- Difícil sair da JVM
  - Outras linguagens devem utilizar Hadoop Streaming, baseado em STDIN & STDOUT
  - É um modelo mais restrito, com maior custo de serialização de dados.

From  
Hadoop



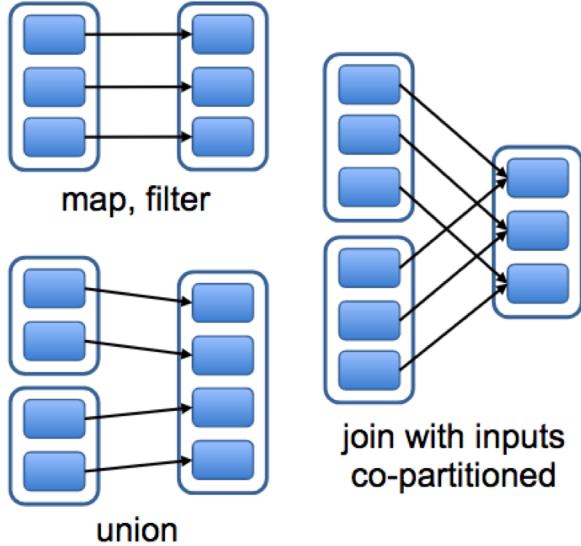


- Análise das operações e sua otimização antes da execução.
- Execução deferida até o momento de coleta dos dados ou I/O.
- Paralelismo emprega multithread.
- Escrita em disco somente para transferência de dados entre nodos (shuffle)
  - ou por solicitação do usuário
- Abstração de RDDs
  - Representação de uma coleção de objetos distribuídos
    - Memória, HDFS, Cassandra, SGBD Relacional, etc.

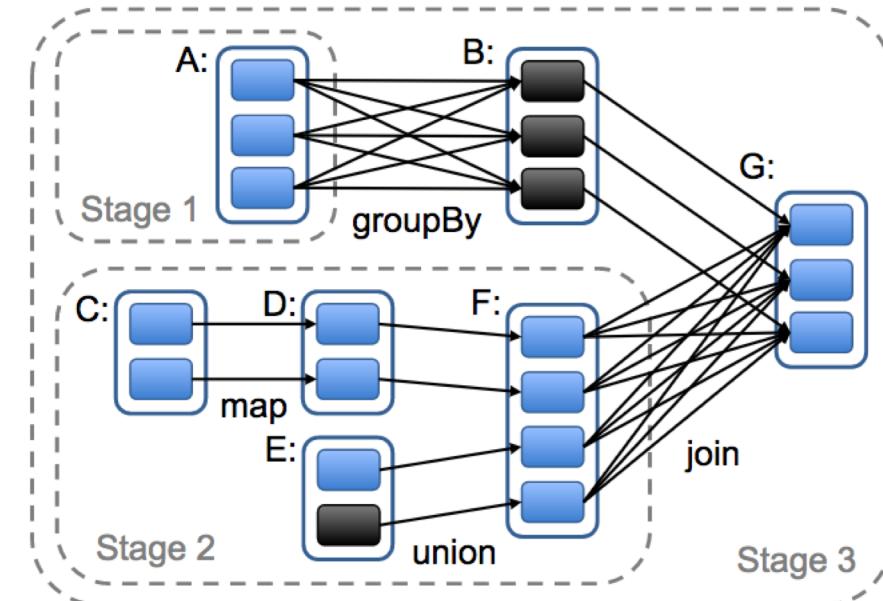
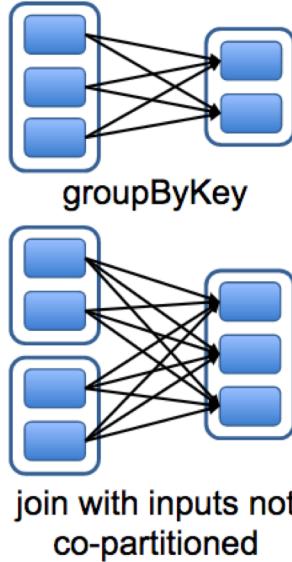


APACHE  
Enter **Spark**™

## Narrow Dependencies:



## Wide Dependencies:



# Stages

DataFrame

*noun*

Making Spark accessible to everyone (data scientists, engineers, statisticians, ...)



# Spark DataFrame

# Spark DataFrame

## Spark SQL: Relational Data Processing in Spark

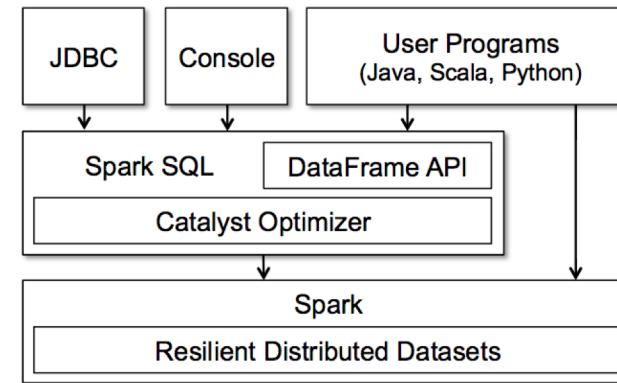
Michael Armbrust<sup>†</sup>, Reynold S. Xin<sup>†</sup>, Cheng Lian<sup>†</sup>, Yin Huai<sup>†</sup>, Davies Liu<sup>†</sup>, Joseph K. Bradley<sup>†</sup>, Xiangrui Meng<sup>†</sup>, Tomer Kaftan<sup>‡</sup>, Michael J. Franklin<sup>†‡</sup>, Ali Ghodsi<sup>†</sup>, Matei Zaharia<sup>†\*</sup>

<sup>†</sup>Databricks Inc.    <sup>\*</sup>MIT CSAIL    <sup>‡</sup>AMPLab, UC Berkeley

- Coleção distribuída de registros organizados em colunas
  - nome, tipo, meta
- Abstração sobre RDDs (RDD + Schema)
- Inspirado em estruturas de dados tabulares & APIs presentes nas linguagens R e Python
- APIs para operações de IO, álgebra relacional (filter, join), matemática & estatística, machine learning
- Apropriado para datasets de KBs até PBs

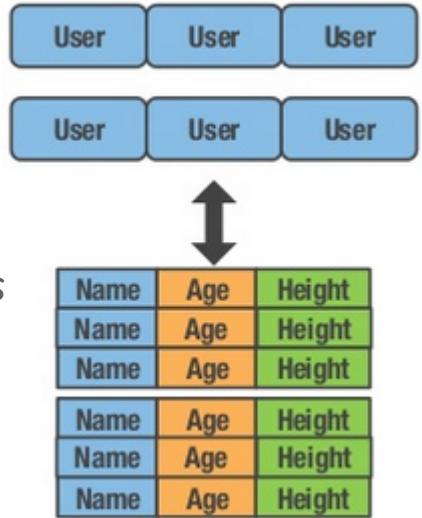
# Spark DataFrame

- Menor quantidade de código
- Lendo menos dados
- Com um otimizador (catalyst)
- RDD também possui otimizador, mas o catalyst é capaz de reordenar consultas.



# Spark DataFrame

- Conhecimento do schema permite otimizar diferentes aspectos da computação
    - Reordenação das operações (query planning)
    - Estratégias de agrupamento (sumarizações & joins - GroupedData)
    - Otimização do uso de memória (columnar layout) sem de-serialização do formato
    - Filtros na origem dos dados (predicate pushdown & projection)
  - Compilação de Bytecode em tempo de execução
    - Built-in functions com geração de código para execução
    - Whole-stage Code Generation – Query compilada em uma função



# Data Sources

Conversão para formatos mais eficientes

- CSV & JSON convertidos para representação binária (em memória), com inferência de schema (opcional, similar ao `read_csv` do pacote `readr`)

# Data Sources

## Uso de formatos colunares (Parquet & ORC Files)

- Projeção (não lê colunas irrelevantes)
- Com particionamento (/year/month)
- Com o uso de estatísticas (min/max)
- Transforma comparação de Strings em comparação de Inteiros (dicionário)

# Data Sources

Aplicação de Predicate Pushdown (envio de filtros para a origem dos dados)

- MySQL, PostgreSQL, Hive
- Cassandra, HBase
- Parquet & ORC Files

## Built-In

{ JSON }



JDBC



MySQL

Parquet



PostgreSQL



amazon web services | S3

H2

## External



APACHE  
HBASE



elasticsearch.



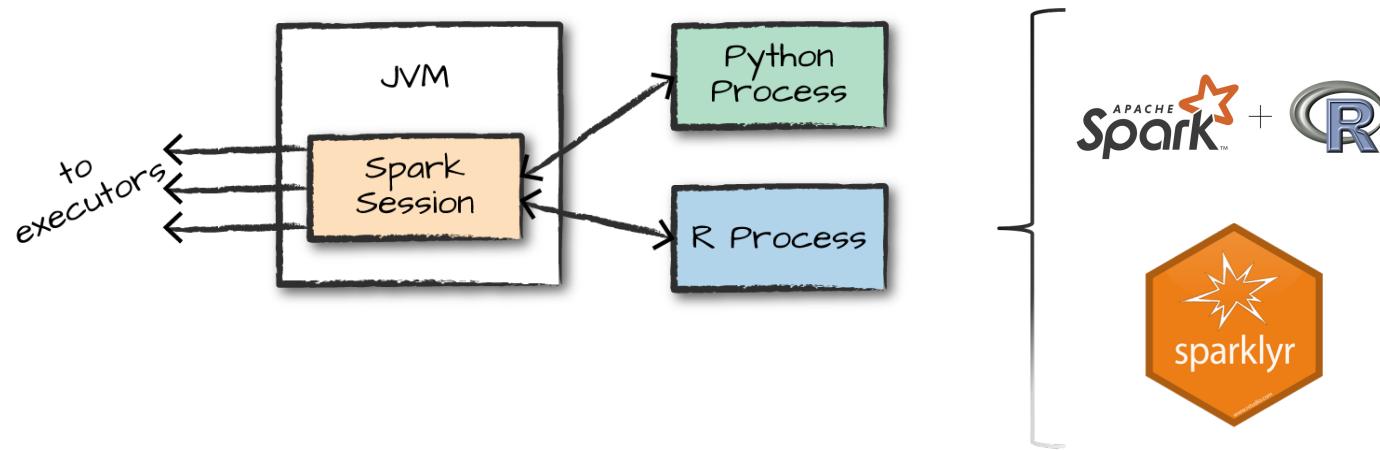
and more...

# Data Sources

# Spark e R

- Linguagem R limita o volume e o paralelismo
  - Dados devem “caber” em memória
  - Sem suporte nativo a paralelismo
- Usuários de R muitas vezes sentem dificuldades em mover do processamento de pequenos volumes de dados para grandes volumes, ainda mais em ambientes distribuídos
- Bibliotecas e técnicas que são familiares aos usuários não funcionam da mesma forma em ambientes distribuídos

# Spark e R



- Hoje temos duas alternativas Open Source
  - **SparkR** (Berkeley + Databricks + MIT)
    - API similar aos operadores e funções da classe `data.frame` padrão do R
    - Incorpora os conceitos do Spark da implementação Scala
  - **sparklyr** (RStudio)
    - Baseada na biblioteca **dplyr**
    - Abstrai os conceitos do Spark, adotando a interface **DBI** de acesso a dados externos integrado ao **dbplyr** (mapeamento **dplyr** -> **SQL**)
- Existe a expectativa de convergência para uma única biblioteca

## SparkR: Scaling R Programs with Spark

Shivaram Venkataraman<sup>1</sup>, Zongheng Yang<sup>1</sup>, Davies Liu<sup>2</sup>, Eric Liang<sup>2</sup>, Hossein Falaki<sup>2</sup>  
Xiangrui Meng<sup>2</sup>, Reynold Xin<sup>2</sup>, Ali Ghodsi<sup>2</sup>, Michael Franklin<sup>1</sup>, Ion Stoica<sup>1,2</sup>, Matei Zaharia<sup>2,3</sup>  
<sup>1</sup>AMPLab UC Berkeley, <sup>2</sup> Databricks Inc., <sup>3</sup> MIT CSAIL

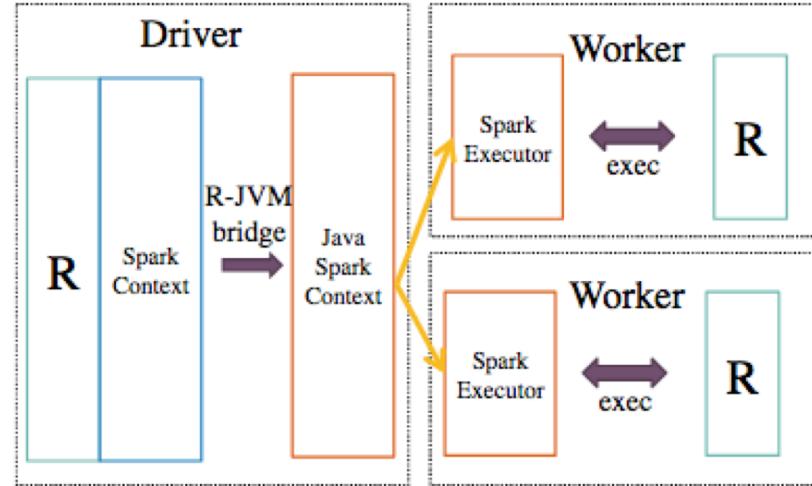


Figure 3: SparkR Architecture

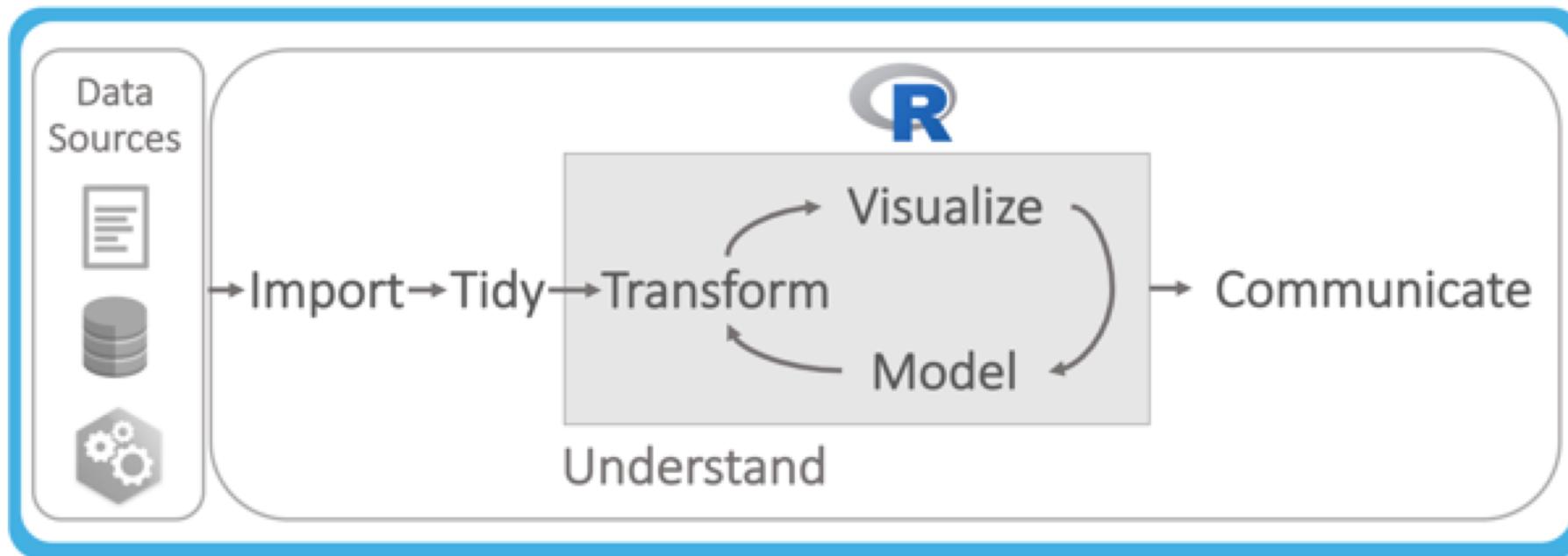
# SparkR

- Mesmos Data Sources do Spark
- Suporte a código e libs R nos Workers
- Disponibilização de funções R para as operações do Spark DataFrame (%>%)
- MLLib com R Formulas e mesmos parâmetros



- Converte verbos dplyr em comandos SQL para Spark
- Possui funções para uso do Spark Mllib
- Disponibiliza também operações específicas do Spark DataFrames que não estão disponíveis via SQL

# R for Data Science



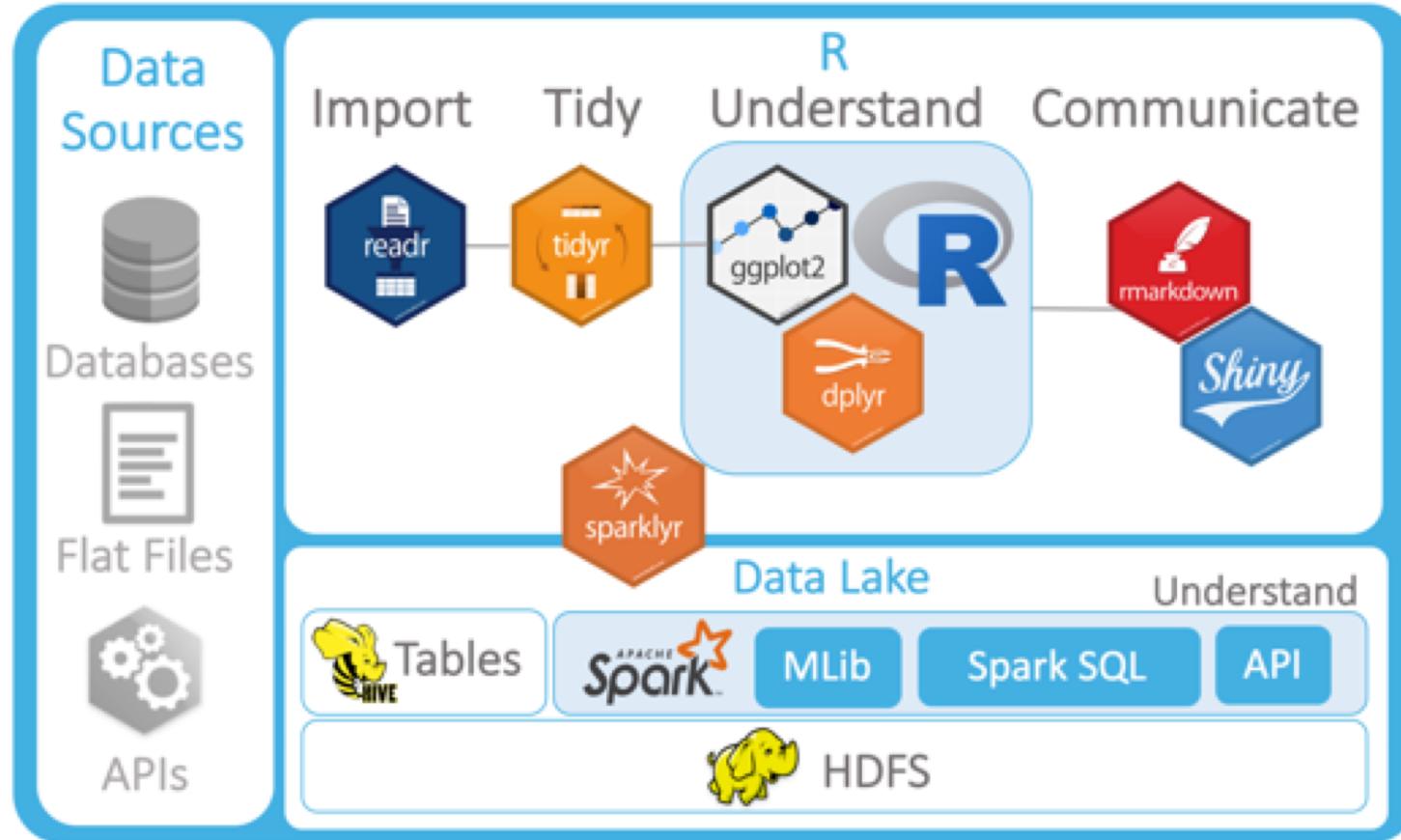
[R for Data Science, Wickham & Grolemund](#)

# Spark as an Analysis Engine

Solution: Use `sparklyr` to access & analyze the data inside Spark.  
Only bring results into R.



# Using Spark & R for Data Science



## Cluster setup

Access RStudio  
using a web  
browser



Name  
node

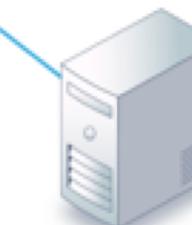


Data  
node



All nodes need  
YARN & Spark  
Gateway services

Data  
node



R, RStudio  
& sparklyr  
in 1 node

Edge  
node



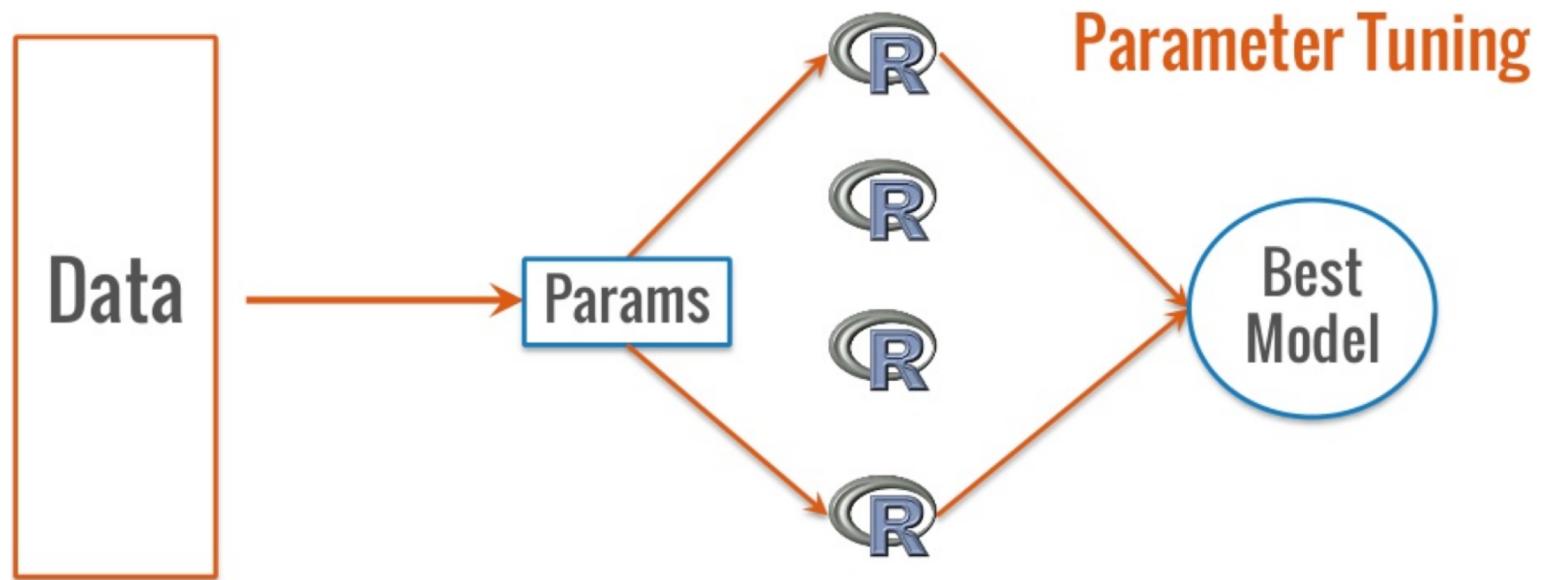
# **Spark e R Padrões**

- Big Data, Small Learning
- Partition & Aggregate
- Large Scale Machine Learning

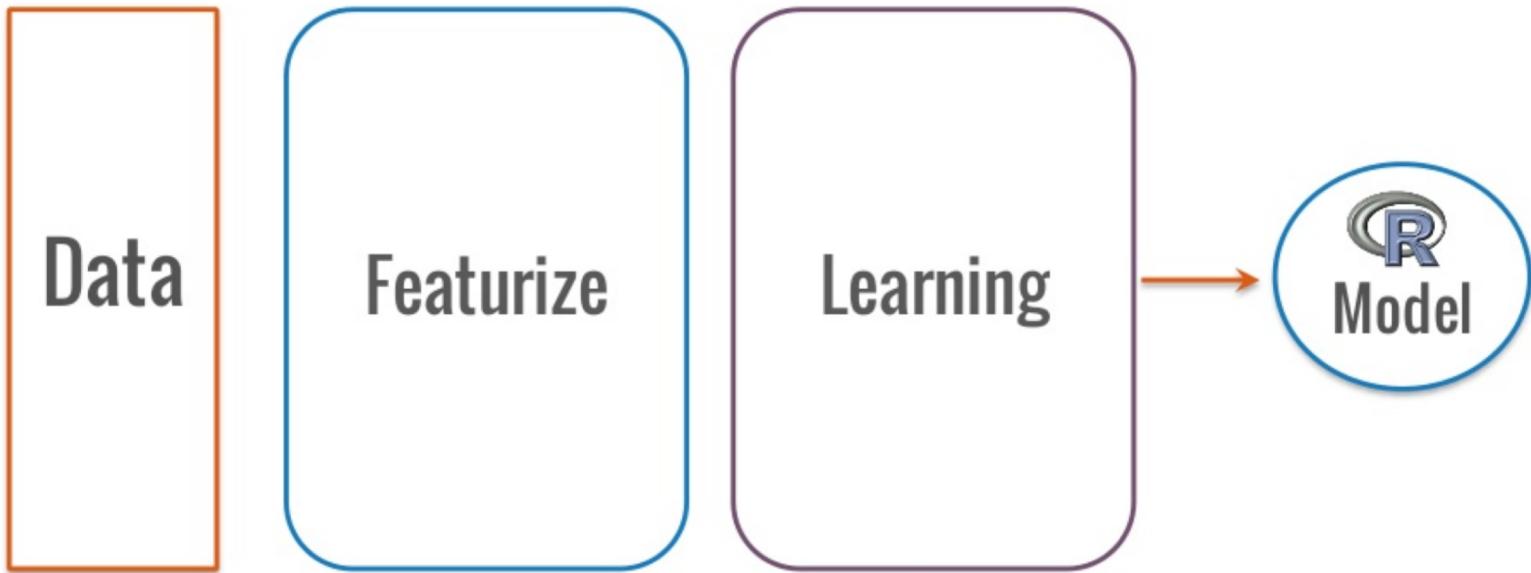
# Big Data Small Learning



# Partition & Aggregate



# Large Scale Machine Learning



# Prática