

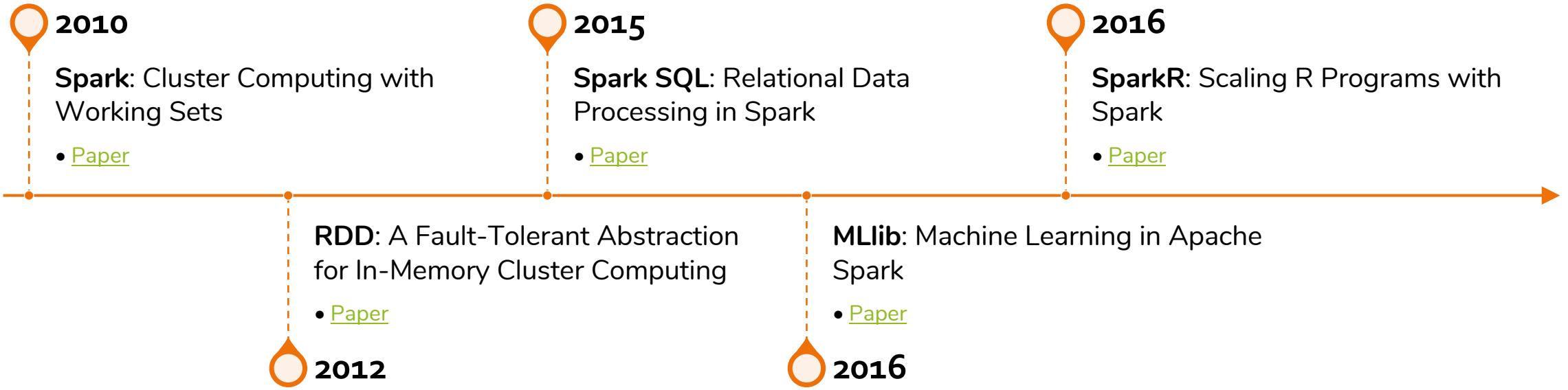


Data Frames /
sparklyr

Quais os problemas?

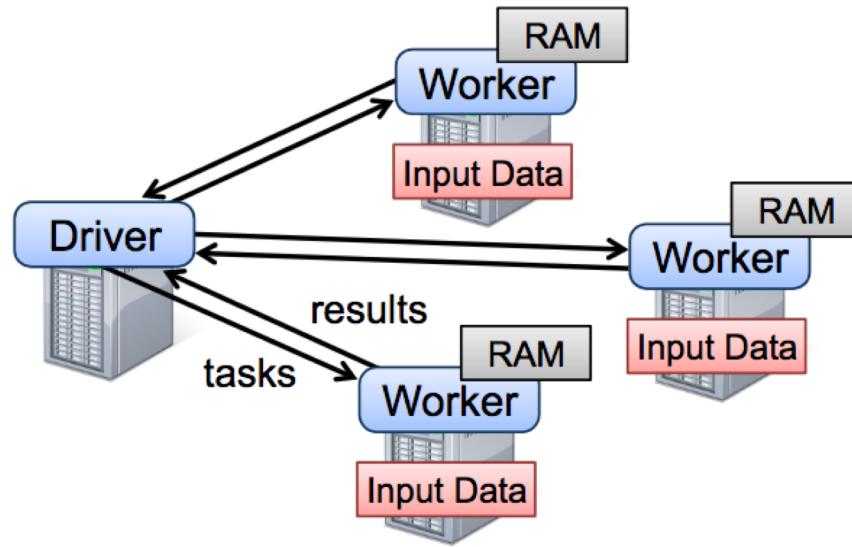
- Modelo de programação muito próximo do modelo de execução
- Programas escritos através de
 - Map, Combiner, Partitioner, Shuffle, Sort, Reduce
- Alternativas (Crunch, Hive, Pig, Cascading, etc) compiladas para o modelo de execução
- IO excessivo
 - Escrita em disco entre fases, ordenação obrigatória 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**™

Spark RDD

- Cada especialização de RDD possui uma função de processamento (**compute**) associada
 - Transformations
 - Actions
- [RDD Programming Guide](#)
 - Scala, Java, Python
 - R opera com DataFrames

Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma,
Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica
University of California, Berkeley

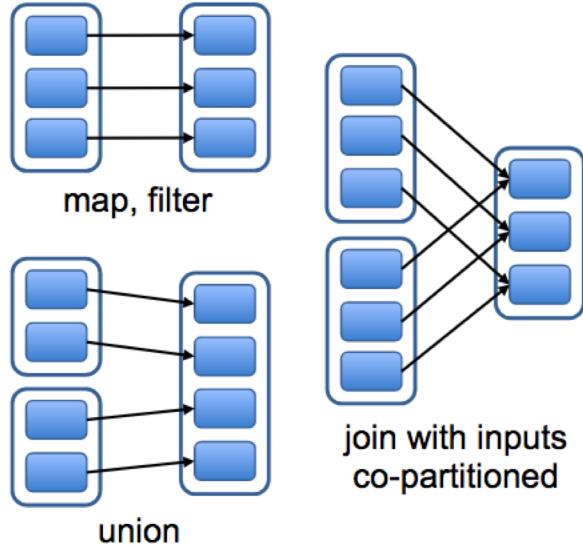
Operation	Meaning
<code>partitions()</code>	Return a list of Partition objects
<code>preferredLocations(<i>p</i>)</code>	List nodes where partition <i>p</i> can be accessed faster due to data locality
<code>dependencies()</code>	Return a list of dependencies
<code>iterator(<i>p</i>, <i>parentIter</i>)</code>	Compute the elements of partition <i>p</i> given iterators for its parent partitions
<code>partitioner()</code>	Return metadata specifying whether the RDD is hash/range partitioned

Table 3: Interface used to represent RDDs in Spark.

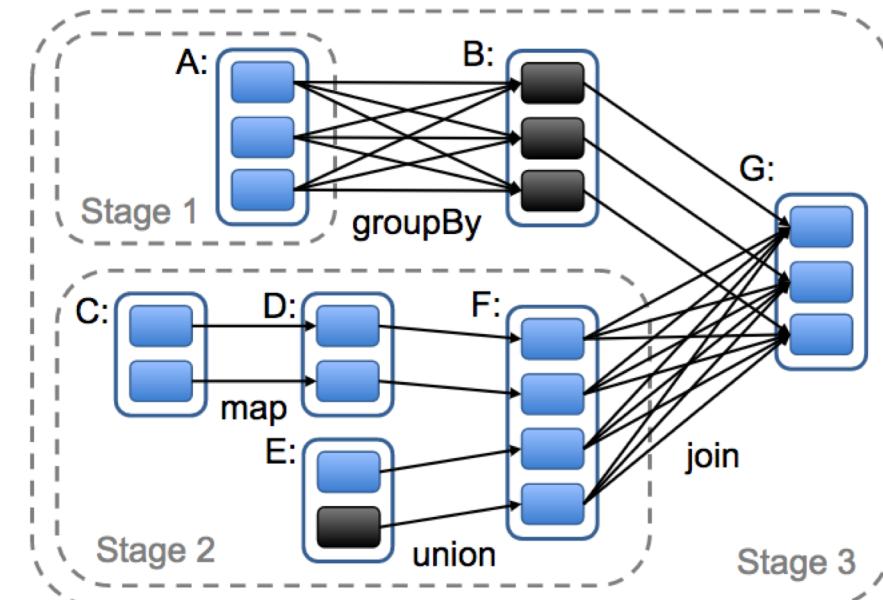
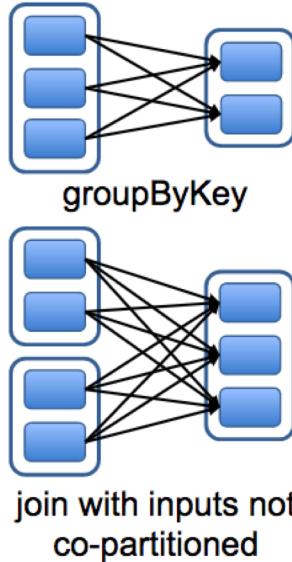
Spark RDD

	HadoopRDD	FilteredRDD
partitions	Uma por bloco (HDFS)	Mesmas do anterior
dependencies	Nenhuma (início de pipeline)	Partição RDD anterior
compute	Leitura do bloco	Processa o anterior e aplica filtro
preferredLocations	Mesma do bloco HDFS	Do anterior
partitioner	Nenhum	Nenhum

Narrow Dependencies:



Wide Dependencies:



Stages

DataFrame

noun

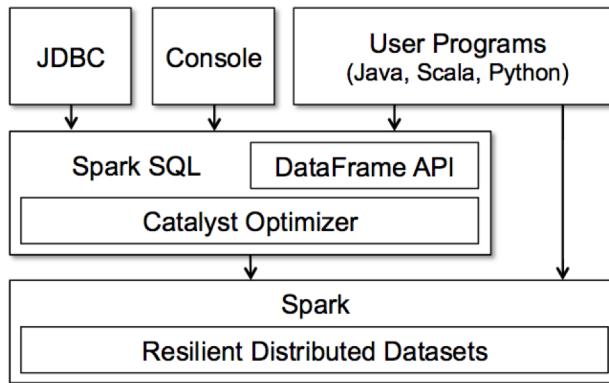
Making Spark accessible to everyone (data scientists, engineers, statisticians, ...)



Spark DataFrame

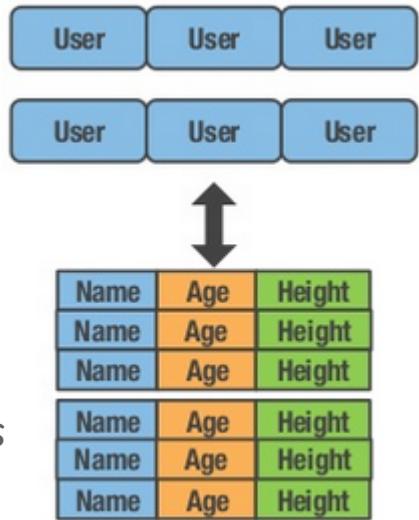
Spark DataFrame

- 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
 - Filtros na origem dos dados (predicate pushdown & projection)
 - Reordenação das operações (query planning)
 - Estratégias de agrupamento (sumarizações & joins - GroupedData)
 - Otimização do uso de memória (columnar layout) sem de-serialização do formato
- 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



Spark DataFrame

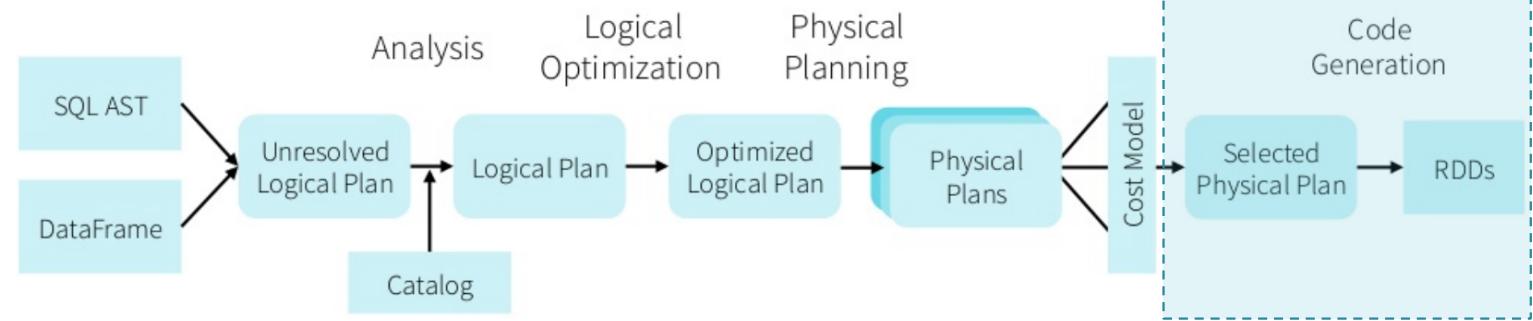
Spark SQL: Relational Data Processing in Spark

Michael Armbrust[†], Reynold S. Xin[†], Cheng Lian[†], Yin Huai[†], Davies Liu[†], Joseph K. Bradley[†], Xiangrui Meng[†], Tomer Kaftan[‡], Michael J. Franklin^{†‡}, Ali Ghodsi[†], Matei Zaharia^{†*}

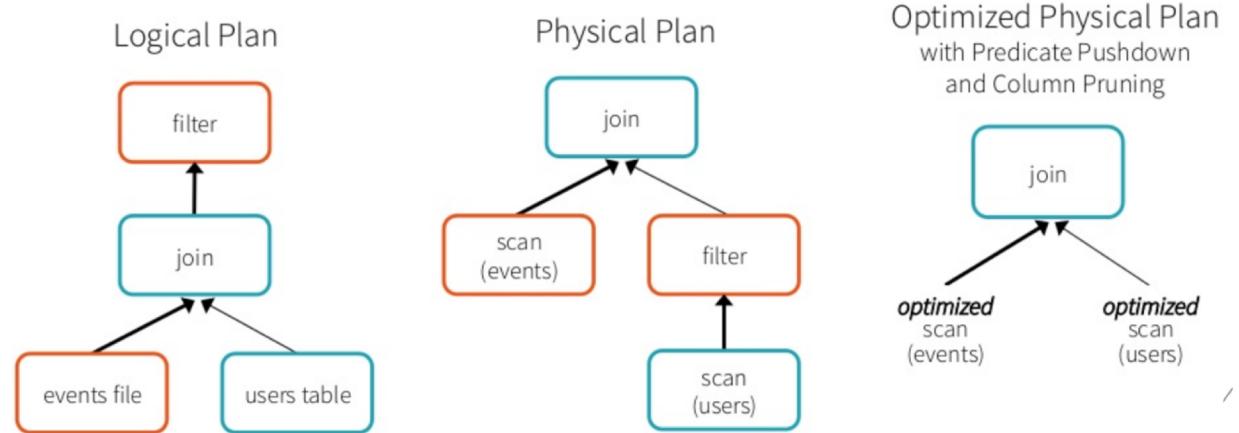
[†]Databricks Inc. ^{*}MIT CSAIL [‡]AMPLab, UC Berkeley

- 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

Otimizador Catalyst



- Geração de código (bytecode JVM) que opera sobre RDDs do tipo Row



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



HIVE

HDFS

amazon web services | S3

Parquet



PostgreSQL

H2

External



APACHE
HBASE



Amazon Redshift



elasticsearch.



dBase

cassandra



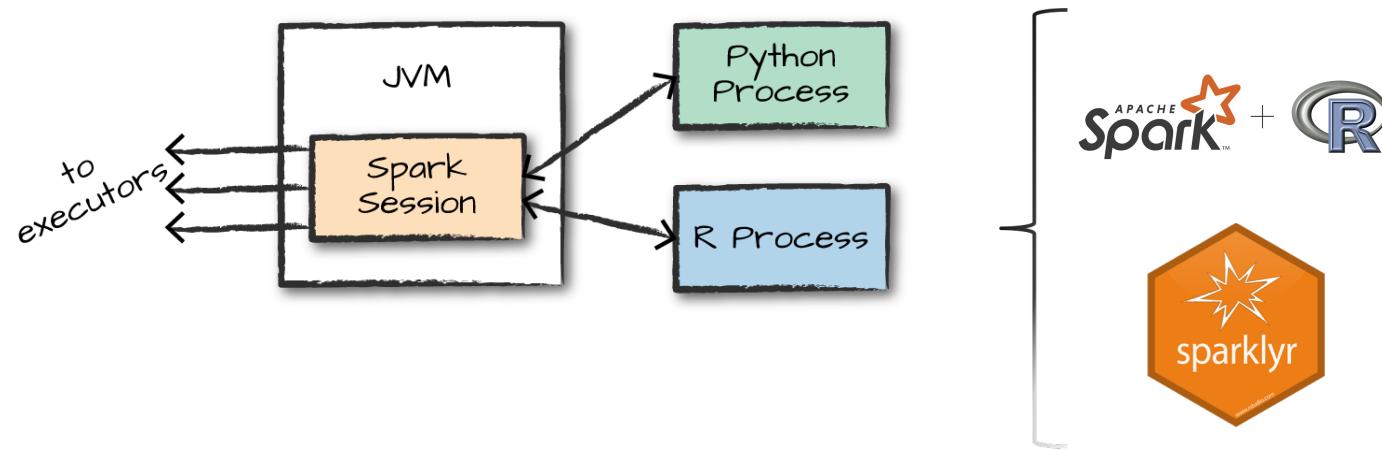
and more...

Data Sources

Spark e R

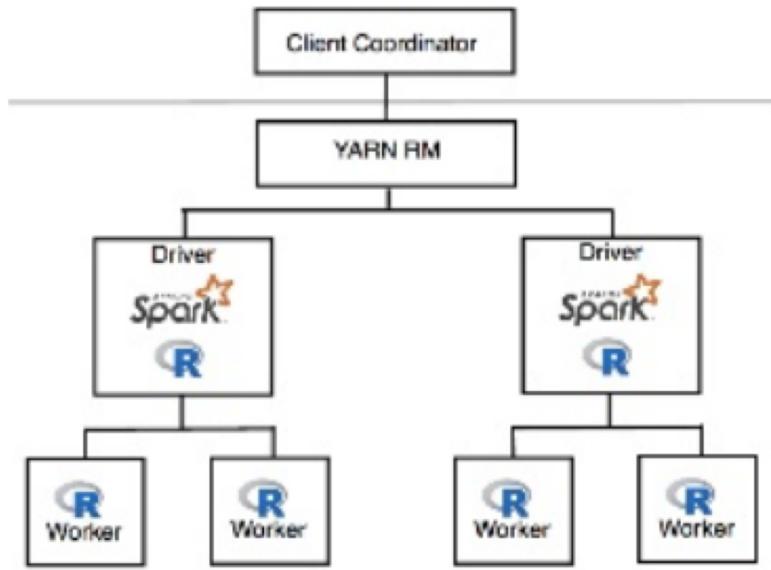
- Linguagem R limita o volume e o paralelismo
 - Dados devem “caber” em memória
 - Sem suporte nativo a paralelismo
- Usuários de R muitas vezes sentem dificuldades em mover do processamento de pequenos volumes de dados para grandes volumes, ainda mais em ambientes distribuídos
- Bibliotecas e técnicas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

Spark e R



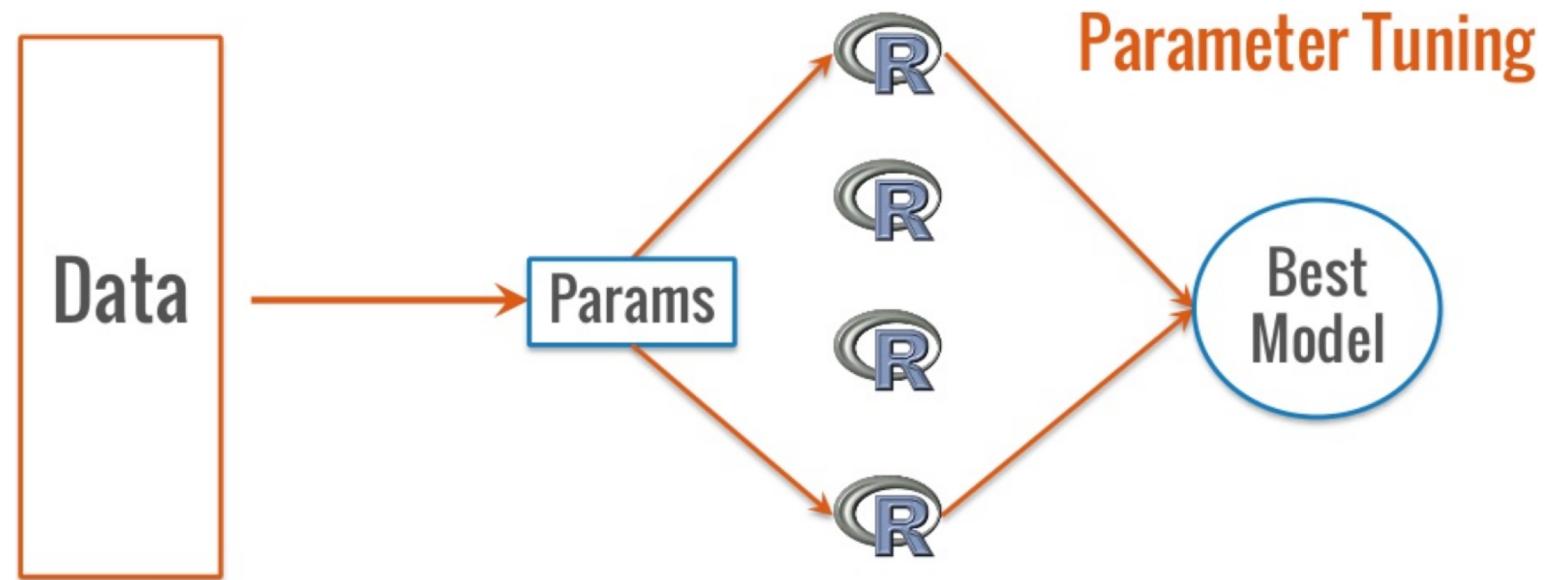
Spark e R Padrões

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

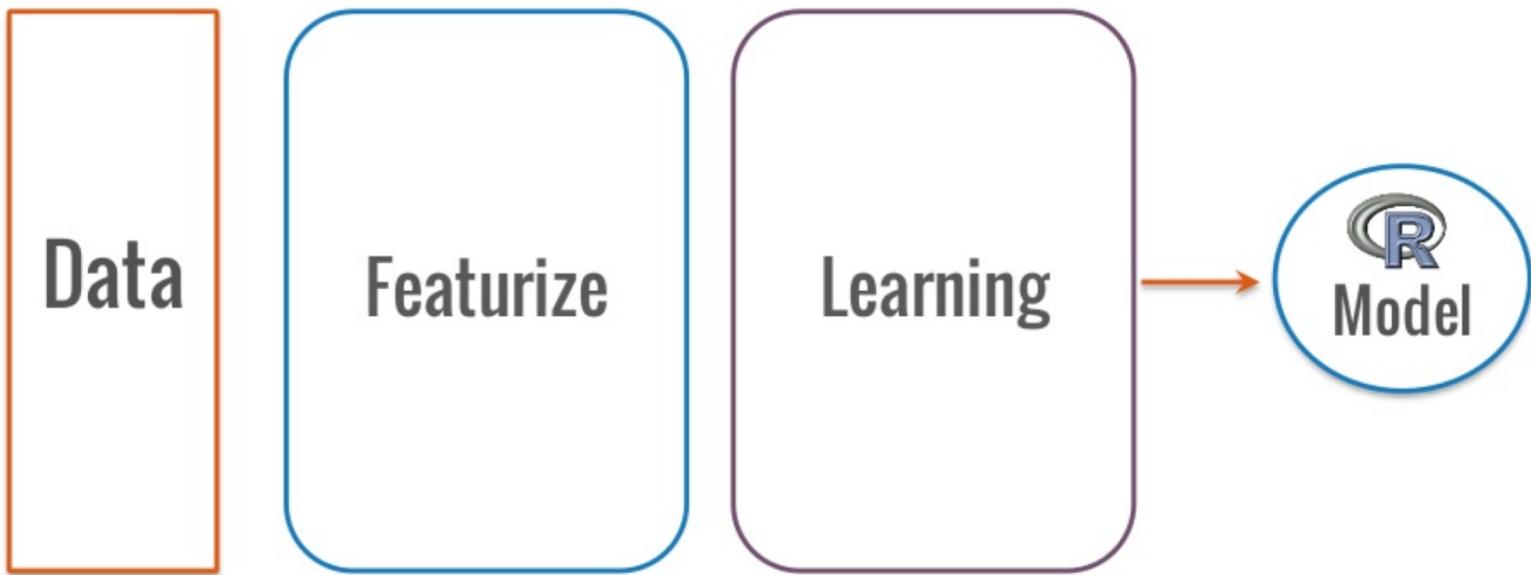
Big Data Small Learning



Partition & Aggregate



Large Scale Machine Learning



SparkR: Scaling R Programs with Spark

Shivaram Venkataraman¹, Zongheng Yang¹, Davies Liu², Eric Liang², Hossein Falaki²
Xiangrui Meng², Reynold Xin², Ali Ghodsi², Michael Franklin¹, Ion Stoica^{1,2}, Matei Zaharia^{2,3}
¹AMPLab UC Berkeley, ² Databricks Inc., ³ MIT CSAIL

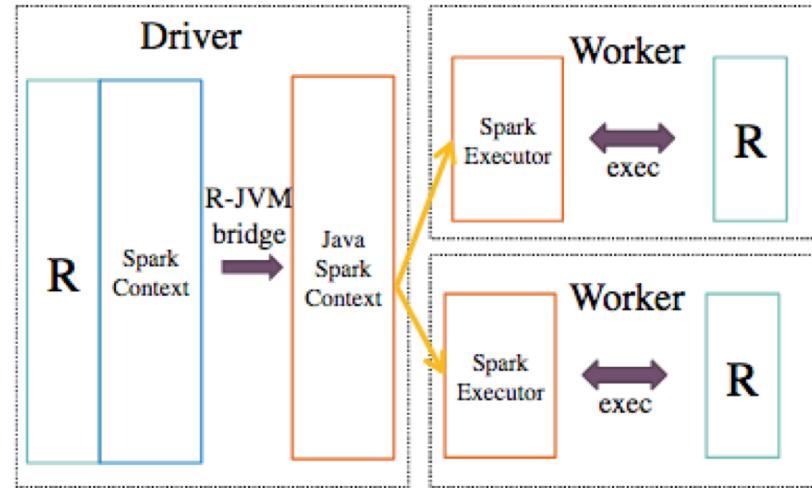


Figure 3: SparkR Architecture

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

Prática