COMPUTAÇÃO EM NUVEM E PROCESSAMENTO MASSIVO DE DADOS

Conceitos, tecnologias e aplicações

Jaqueline Joice Brito

Slides em colaboração com Lucas de Carvalho Scabora

Sumário

- **Computação em Nuvem**
  - Definição
  - Modelos de Serviços
  - Prós e Contras
  - Aplicações
- **Processamento Massivo de Dados**
  - Hadoop e Spark
  - Exemplo: processamento de junção estrela

O que é Computação em Nuvem?

- Computação em nuvem é um modelo que possibilita acesso a um conjunto de recursos computacionais compartilhados e interligados via rede

Imagem: http://antonioricardo.org/tag/cloud/

O que é Computação em Nuvem?

A nuvem é uma metáfora para a Internet ou infraestrutura de comunicação entre os componentes arquiteturais

Abstração que oculta a complexidade de infraestrutura

Imagem: http://www.uolhost.com.br/blog/cloud-computing-a-importancia-das-nuvens#rmcl

O que é Computação em Nuvem?

Em resumo, é uma plataforma que provê serviços sob-demanda, que está sempre disponível, em qualquer lugar e a qualquer hora

Fonte da imagem: http://redes-e-servidores.blogspot.com.br/2013/03/a-evolucao-da-computacao-cloud-computing.html

O que é Computação em Nuvem?

Definição segundo NIST

"Computação em nuvem é um modelo para permitir acesso ubíquo, conveniente e sob demanda via rede a um agrupamento compartilhado e configurável de recursos computacionais (por exemplo, redes, servidores, equipamentos de armazenamento, aplicações e serviços), que pode ser rapidamente fornecido e liberado com esforços mínimos de gerenciamento ou interação com o provedor de serviços."

O que é Computação em Nuvem?

Principais Características

- Serviço sob demanda
  - Alocação dinâmica de recursos
- Acesso via rede
- Compartilhamento de recursos
- Elasticidade
  - Sensação de capacidade infinita de recursos
- Serviço mensurável
  - Modelo pay-as-you-go

O que é Computação em Nuvem?

	Tecnologia	Modelo de Negócio
Mainframe	Computação Centralizada	Alto custo de hardware e software
Cliente/Servidor	Computação Distribuída	Licença para SO e aplicativos
Computação em Nuvem	Grandes data centers	Custo proporcional ao uso

O que é Computação em Nuvem?

Modelos de implantação

- Privada
  - On-premise e Off-premise
- Comunitária
- Pública
- Híbrida

Imagem: http://www.cloudproviderusa.com/the-difference-between-public-cloud-private-cloud/

Modelos de Serviços

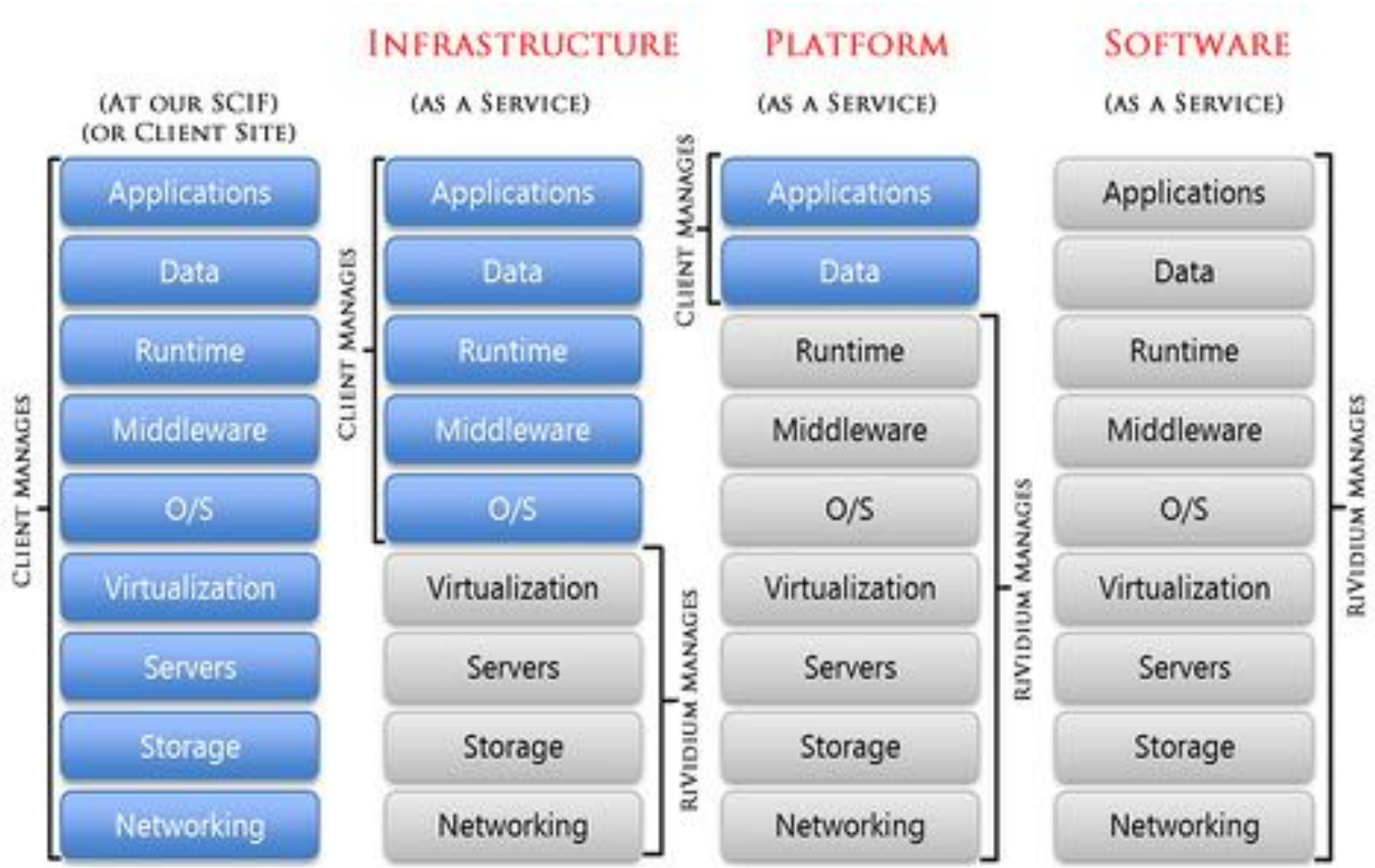


Imagem: http://www.rividium.com/cloud.aspx

Modelos de Serviços

Termos Emergentes

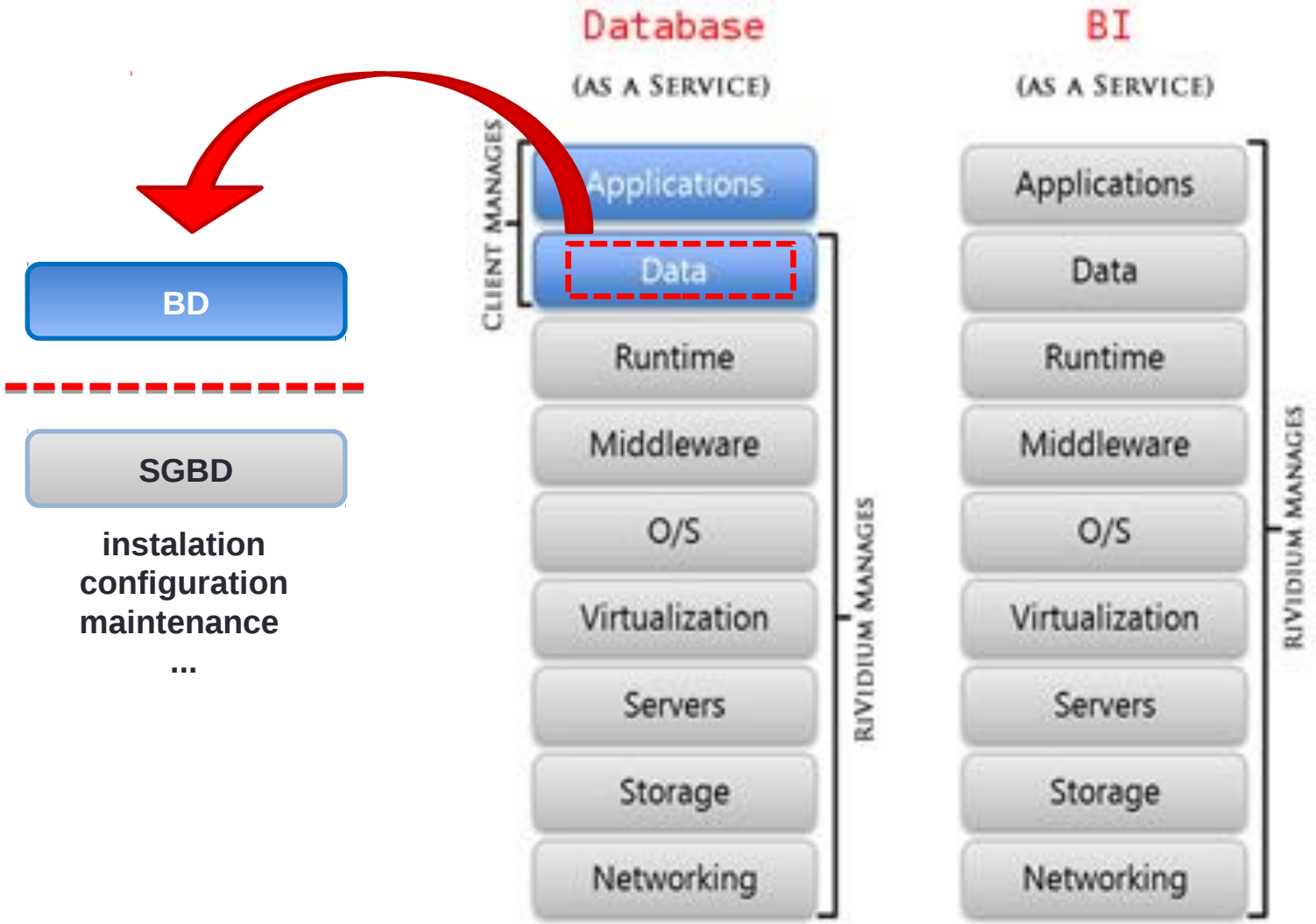
Database as a Service (DBaaS)

O provedor de serviço tem a responsabilidade de instalar e dar manutenção ao banco de dados. Assim, o usuário, que contratou o serviço, apenas paga um valor proporcional ao uso.

Business Intelligence as a Service (BIaaS)

Também chamado de Software como um servico de business inteligence (SaaS BI), consiste em instalar e gerenciar aplicações de BI na nuvem.

Modelos de Serviços



Adaptado de: http://www.rividium.com/cloud.aspx

Modelos de Serviços

Exemplos

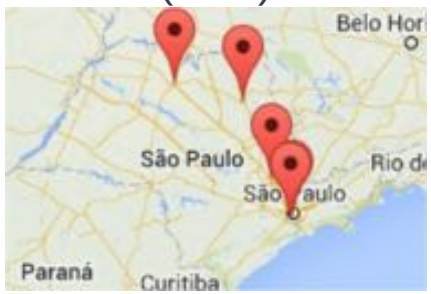
Software as a Service (SaaS)	Platform as a Service (PaaS)	Infrastructure as a Service (IaaS)
salesforce Google Docs	Windows Azure Google App Engine	AUTOMATIC SCALING, LOAD BALANCING, MONITORING, amazon web services EC2 rackspace the open cloud company

Tipos de Armazenamento

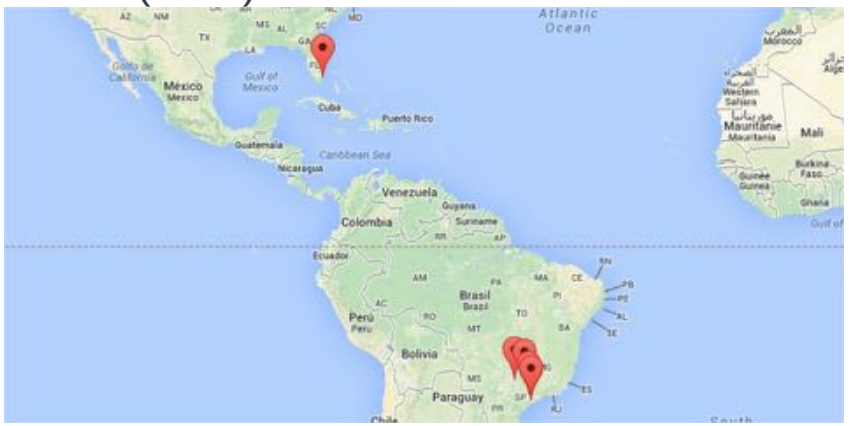
Exemplo da Microsoft Azure

ARMAZENAMENTO COM REDUNDÂNCIA LOCAL (LRS)

ARMAZENAMENTO COM REDUNDÂNCIA DE ZONA (ZRS)



ARMAZENAMENTO COM REDUNDÂNCIA GEOGRÁFICA (GRS) e ARMAZENAMENTO COM REDUNDÂNCIA GEOGRÁFICA NO ACESSO DE LEITURA (RA-GRS)



Prós e Contras

Principais Vantagens

Redução de investimento em TI (hardware e software)

- Envolvendo custo de manutenção, de pessoal, de espaço físico e de energia

Confiabilidade

- Replicação dos dados
- Disponibilidade das aplicações

Prós e Contras

Principais Vantagens

- Não há contratos complexos e duradouros de prestação de serviço
  - Cobra-se apenas pelos recursos multiplicados pelo tempo de uso
- Escala por demanda
  - Capacidade virtualmente infinitos
- Abstração da tecnologia sendo utilizada

Prós e Contras

Principais Desafios

Segurança e confidencialidade dos dados

Gerenciamento dos dados

Disponibilidade

Integração de serviços

Necessidade de uma banda maior de internet

Aplicações

Usos mais comuns da computação em nuvem

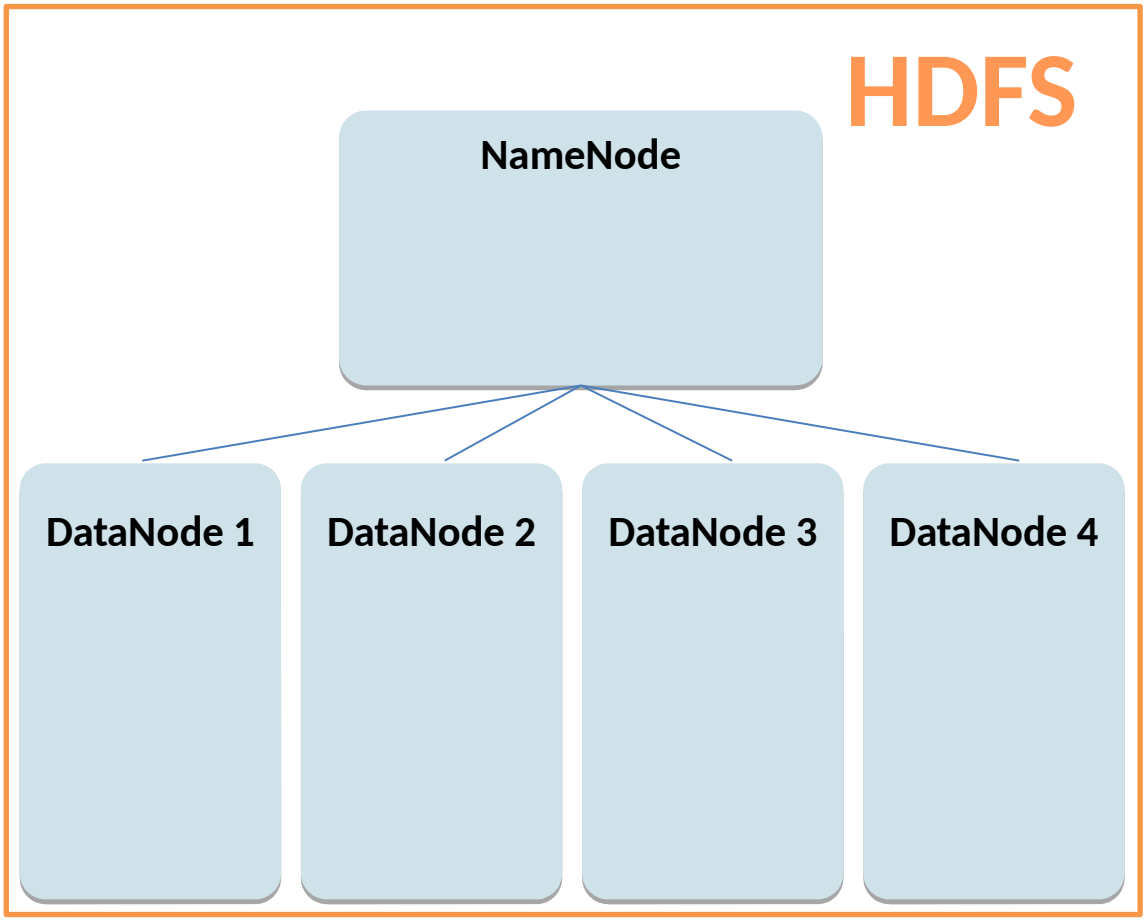
- IaaS e PaaS
- Armazenamento de dados
- Ambientes de testes e desenvolvimento
- Processamento e análise de Big Data
- Backup

Hadoop

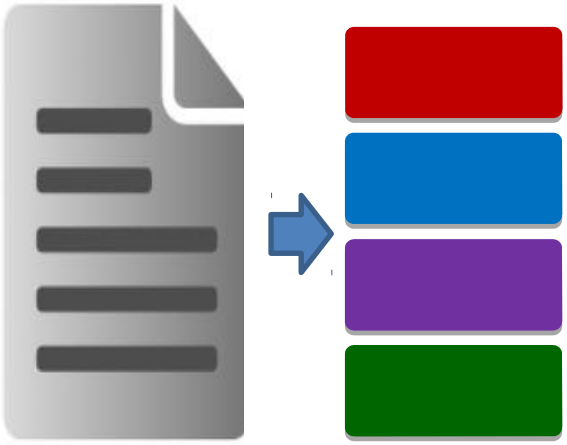
- Framework de processamento paralelo de dados em larga escala
- Altamente escalável
- Tolerante a falhas
- Disponível

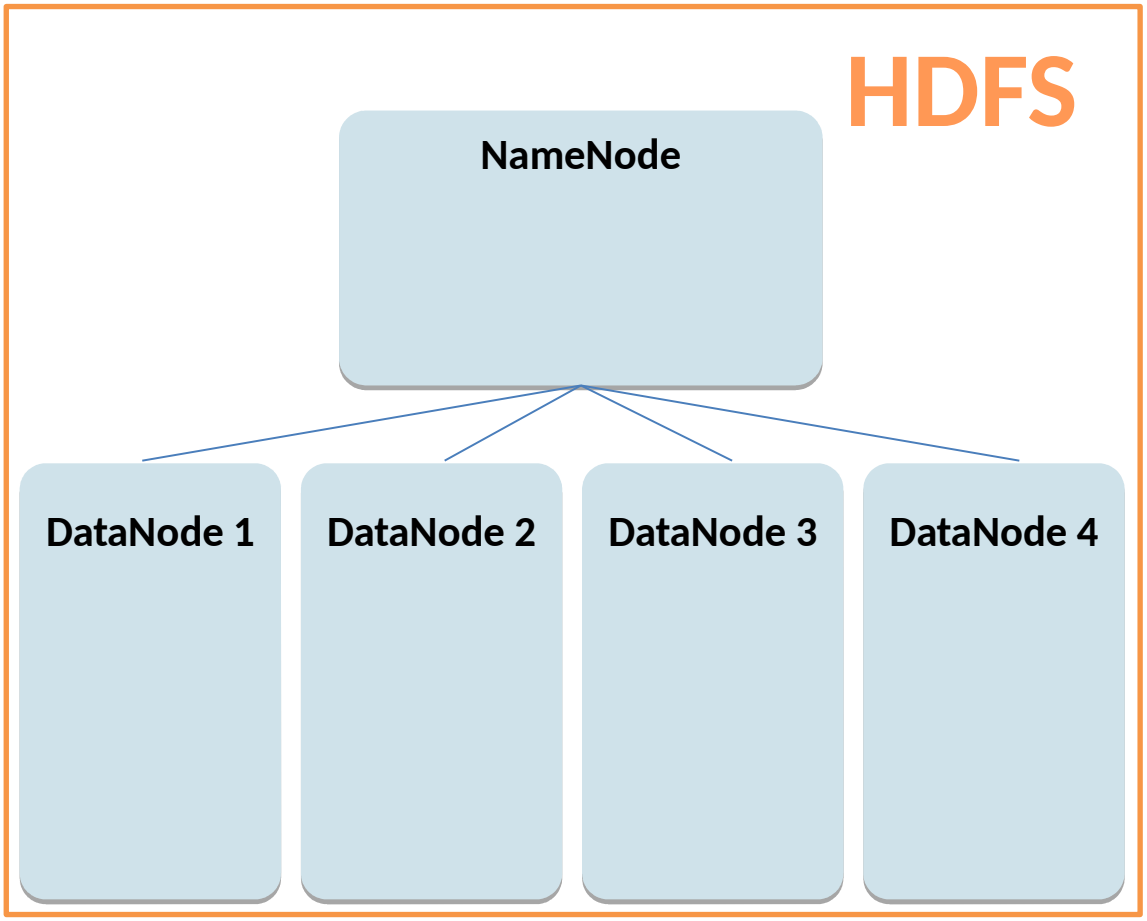
- Principais componentes
  - Armazenamento
    - HDFS – Hadoop Distributed File System
  - Processamento
    - MapReduce

HDFS – Hadoop Distributed File System

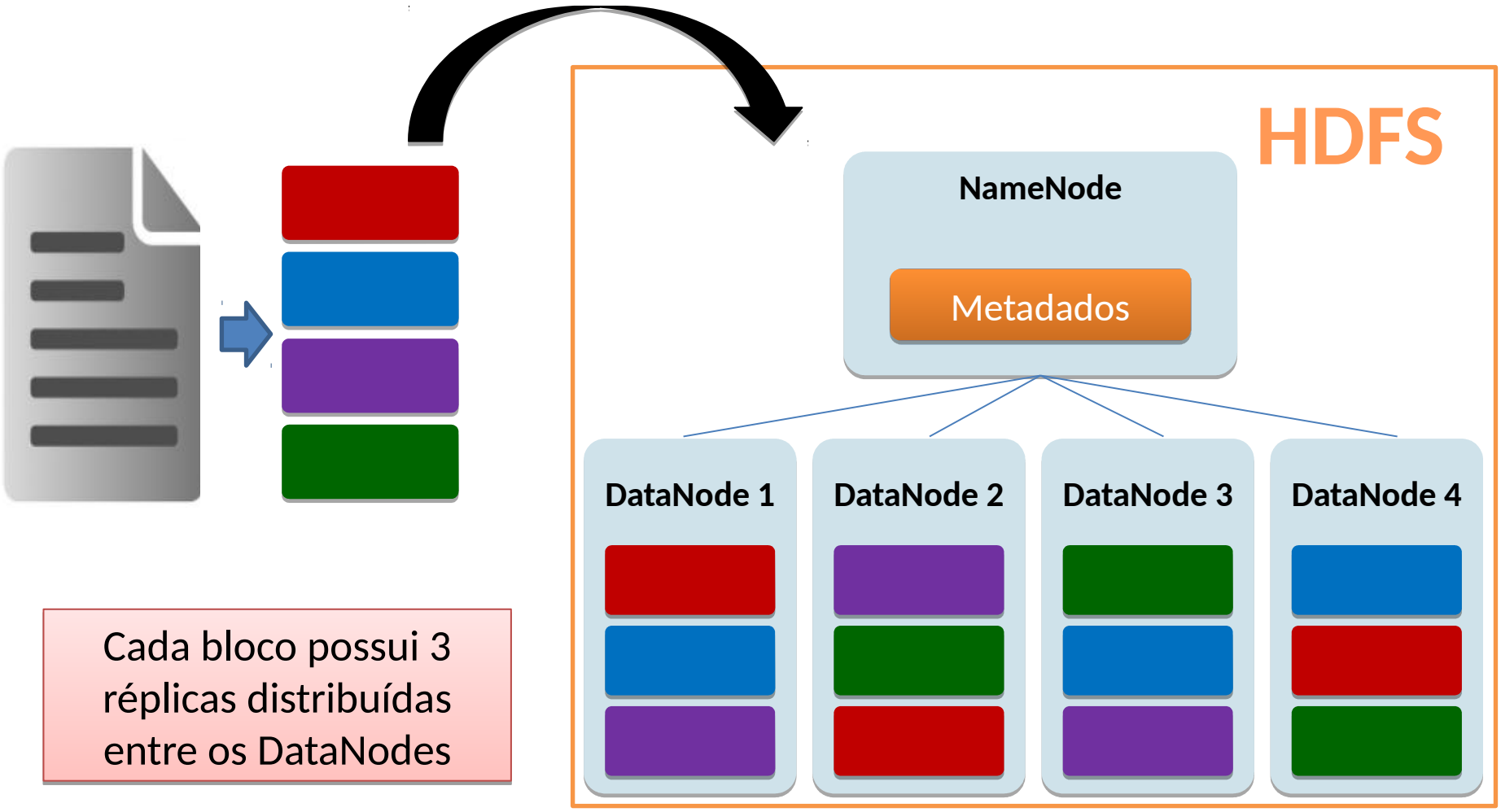


HDFS – Hadoop Distributed File System





HDFS – Hadoop Distributed File System

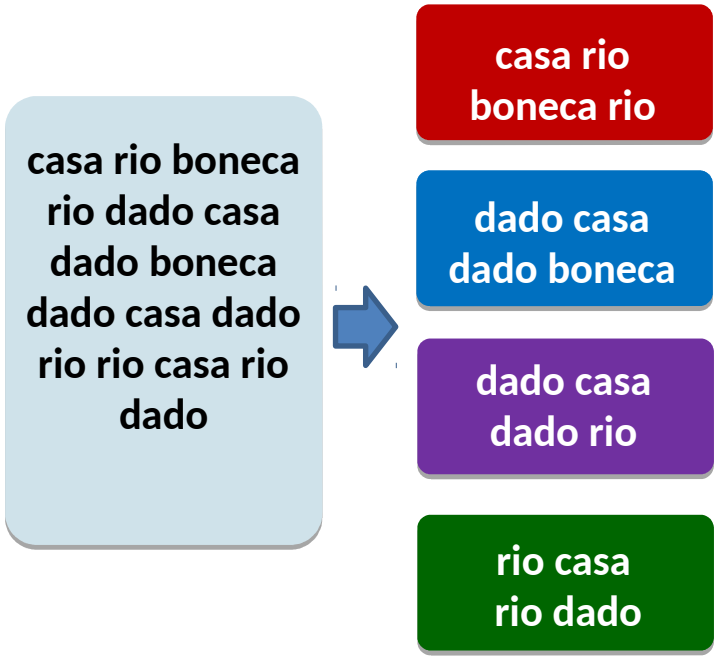


Cada bloco possui 3 réplicas distribuídas entre os DataNodes

MapReduce

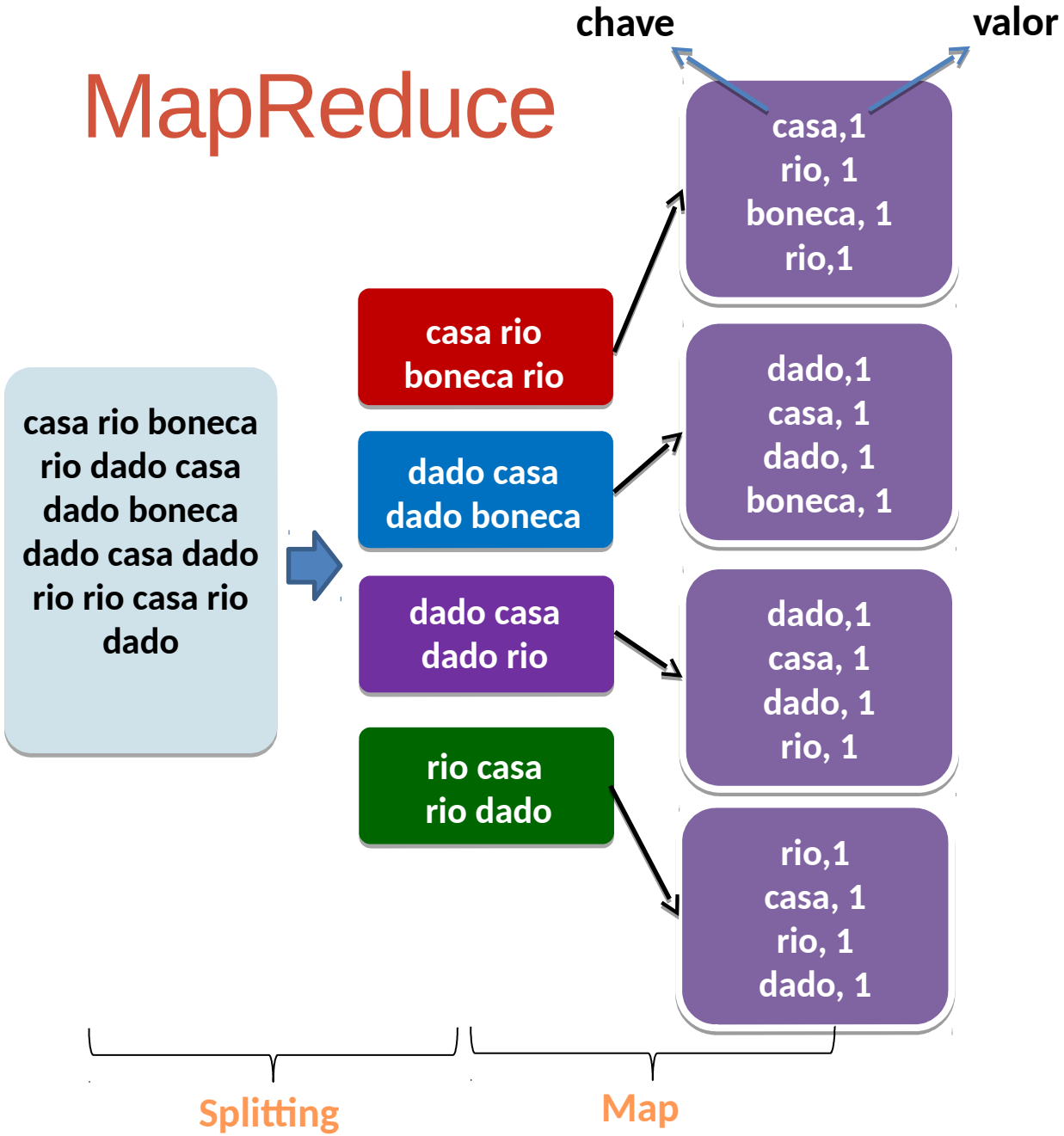
casa rio boneca
rio dado casa
dado boneca
dado casa dado
rio rio casa rio
dado

MapReduce

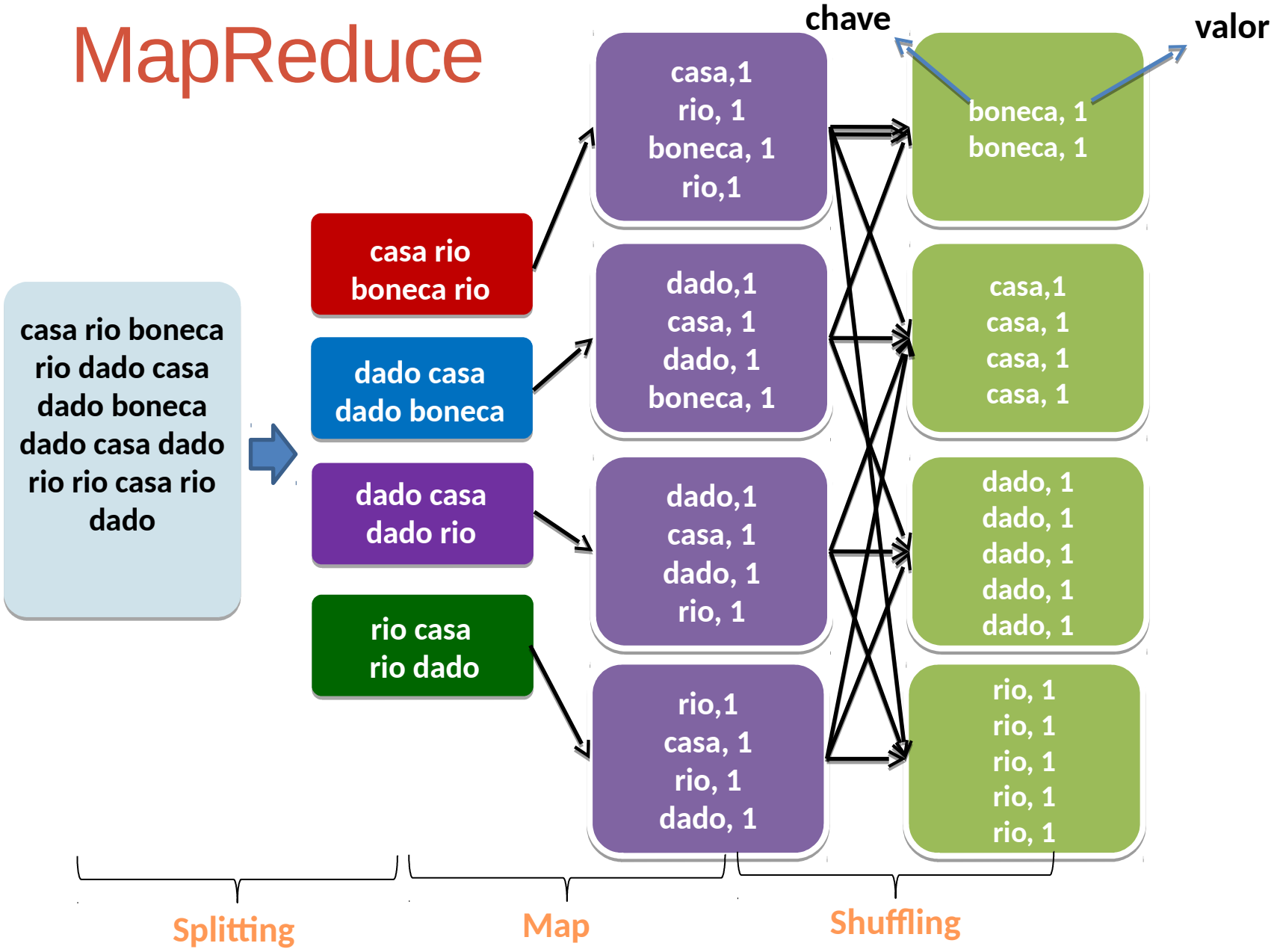


Splitting

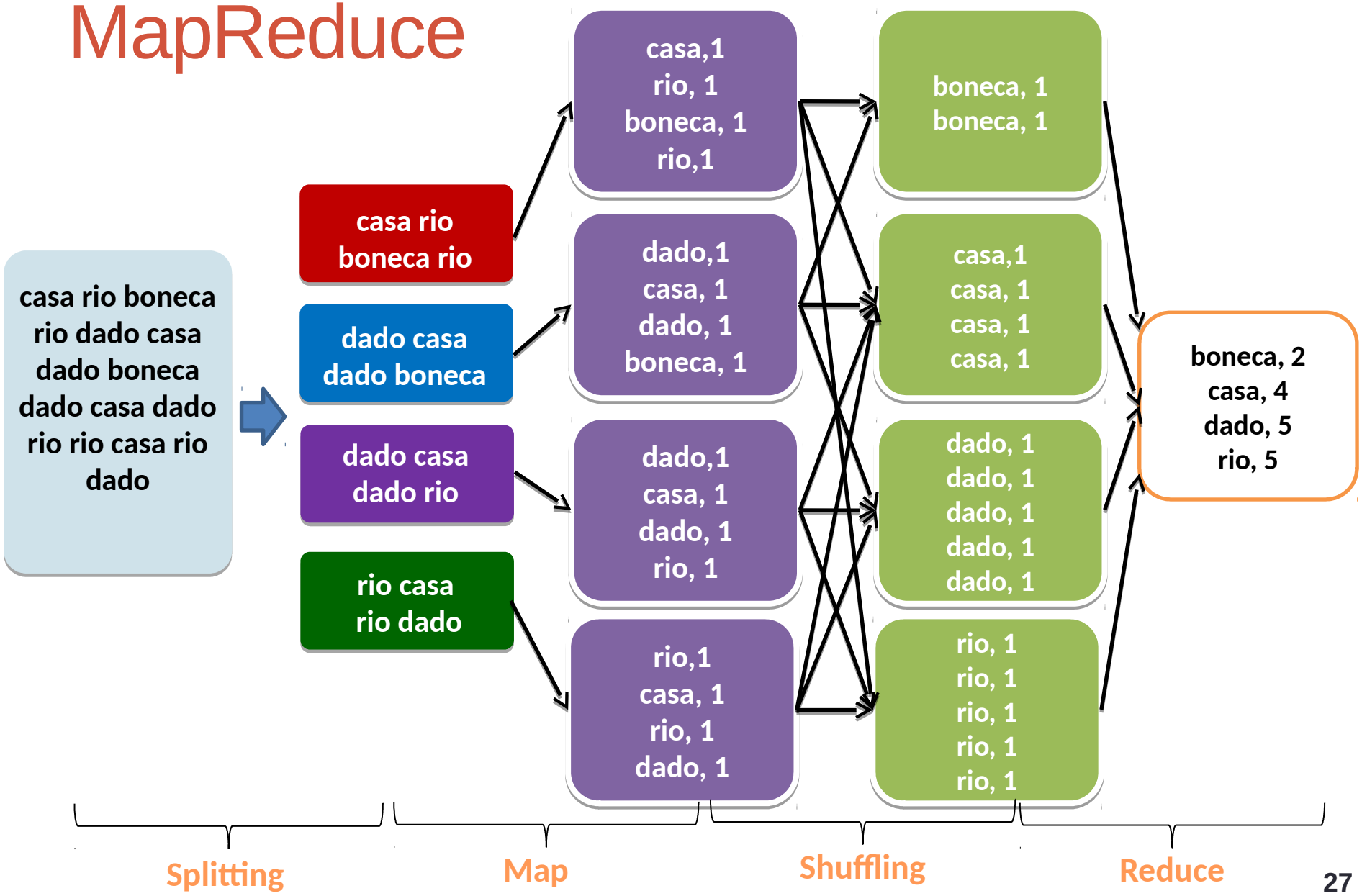
MapReduce


chave
valor
casa rio boneca rio dado casa dado boneca dado casa dado rio rio casa rio dado
casa rio boneca rio
dado casa dado boneca
dado casa dado rio
rio casa rio dado
casa,1 rio, 1 boneca, 1 rio,1
dado,1 casa, 1 dado, 1 boneca, 1
dado,1 casa, 1 dado, 1 rio, 1
rio,1 casa, 1 rio, 1 dado, 1
Splitting
Map

MapReduce


chave
valor
casa rio boneca rio dado casa dado boneca dado casa dado rio rio casa rio dado
casa rio boneca rio
dado casa dado boneca
dado casa dado rio
rio casa rio dado
casa,1 rio, 1 boneca, 1 rio,1
dado,1 casa, 1 dado, 1 boneca, 1
dado,1 casa, 1 dado, 1 rio, 1
rio,1 casa, 1 rio, 1 dado, 1
boneca, 1 boneca, 1
casa,1 casa, 1 casa, 1 casa, 1
dado, 1 dado, 1 dado, 1 dado, 1 dado, 1
rio, 1 rio, 1 rio, 1 rio, 1 rio, 1
Splitting
Map
Shuffling

MapReduce


casa rio boneca rio dado casa dado boneca dado casa dado rio rio casa rio dado
casa rio boneca rio
dado casa dado boneca
dado casa dado rio
rio casa rio dado
casa,1 rio, 1 boneca, 1 rio,1
dado,1 casa, 1 dado, 1 boneca, 1
dado,1 casa, 1 dado, 1 rio, 1
rio,1 casa, 1 rio, 1 dado, 1
boneca, 1 boneca, 1
casa,1 casa, 1 casa, 1 casa, 1
dado, 1 dado, 1 dado, 1 dado, 1 dado, 1
rio, 1 rio, 1 rio, 1 rio, 1 rio, 1
boneca, 2 casa, 4 dado, 5 rio, 5
Splitting
Map
Shuffling
Reduce

MapReduce

Função Map

```
public class WordCountMapper extends
        Mapper<Object, Text, Text, IntWritable> {
    private final IntWritable ONE = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {
        String[] csv = value.toString().split(",");
        for (String str : csv) {
            word.set(str);
            context.write(word, ONE);
        }
    }
}
```

MapReduce

Função Reduce

```
public class WordCountReducer extends
        Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text text, Iterable<IntWritable> values,
            Context context) throws IOException,
            InterruptedException {

        int sum = 0;
        for (IntWritable value : values) {
            sum += value.get();
        }
        context.write(text, new IntWritable(sum));
    }
}
```

Spark

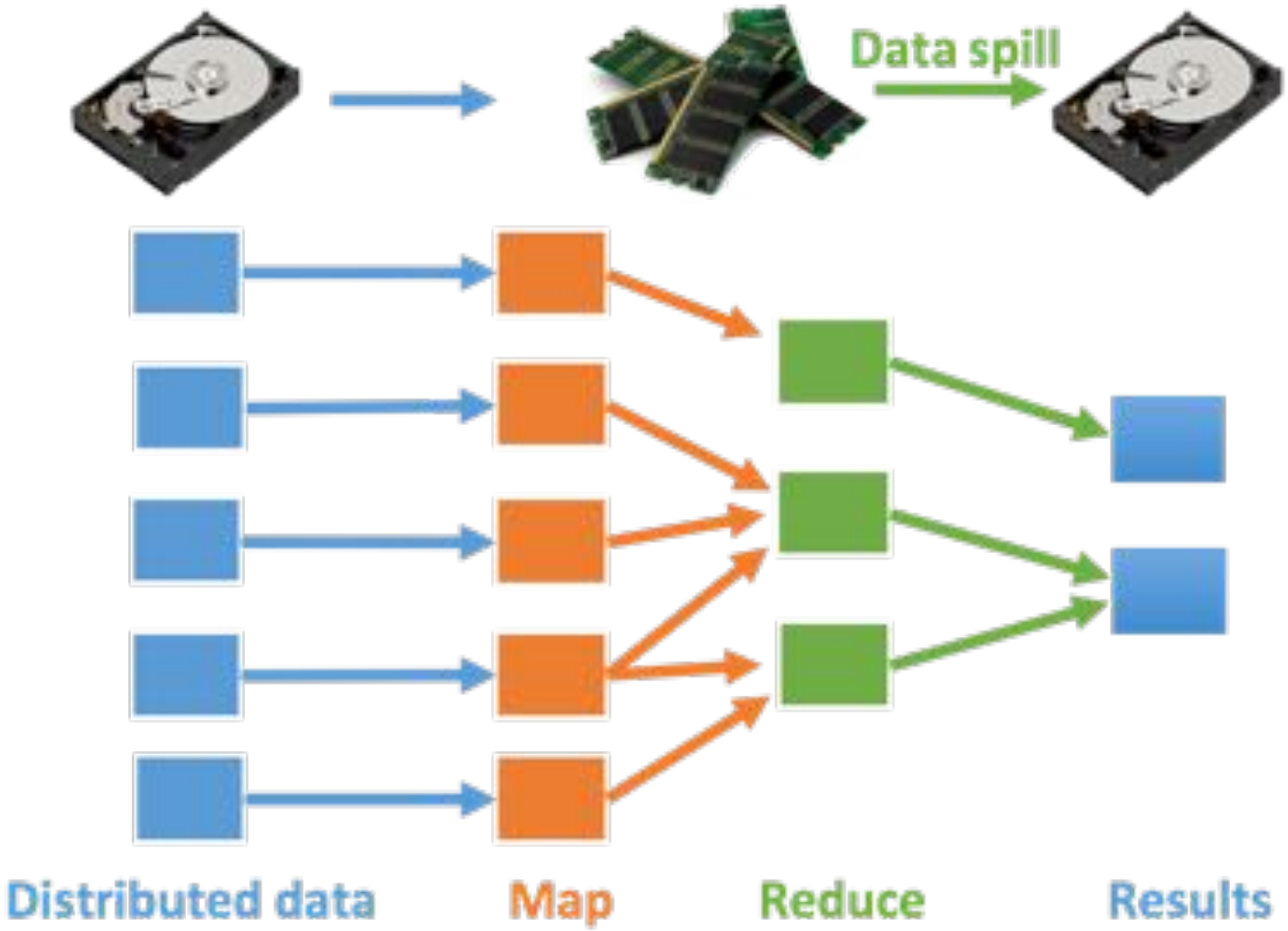
- Framework de processamento distribúido
- Computação em memória
- Baseado em RDDs – Resilient Distributed Datasets
  - Estruturas de dados em paralelo
  - Tolerante a falhas
  - Persistência de resultados intermediários em memória primária
  - Manipulação otimizada por meio de um conjunto operadores

Spark

Exemplo: word count

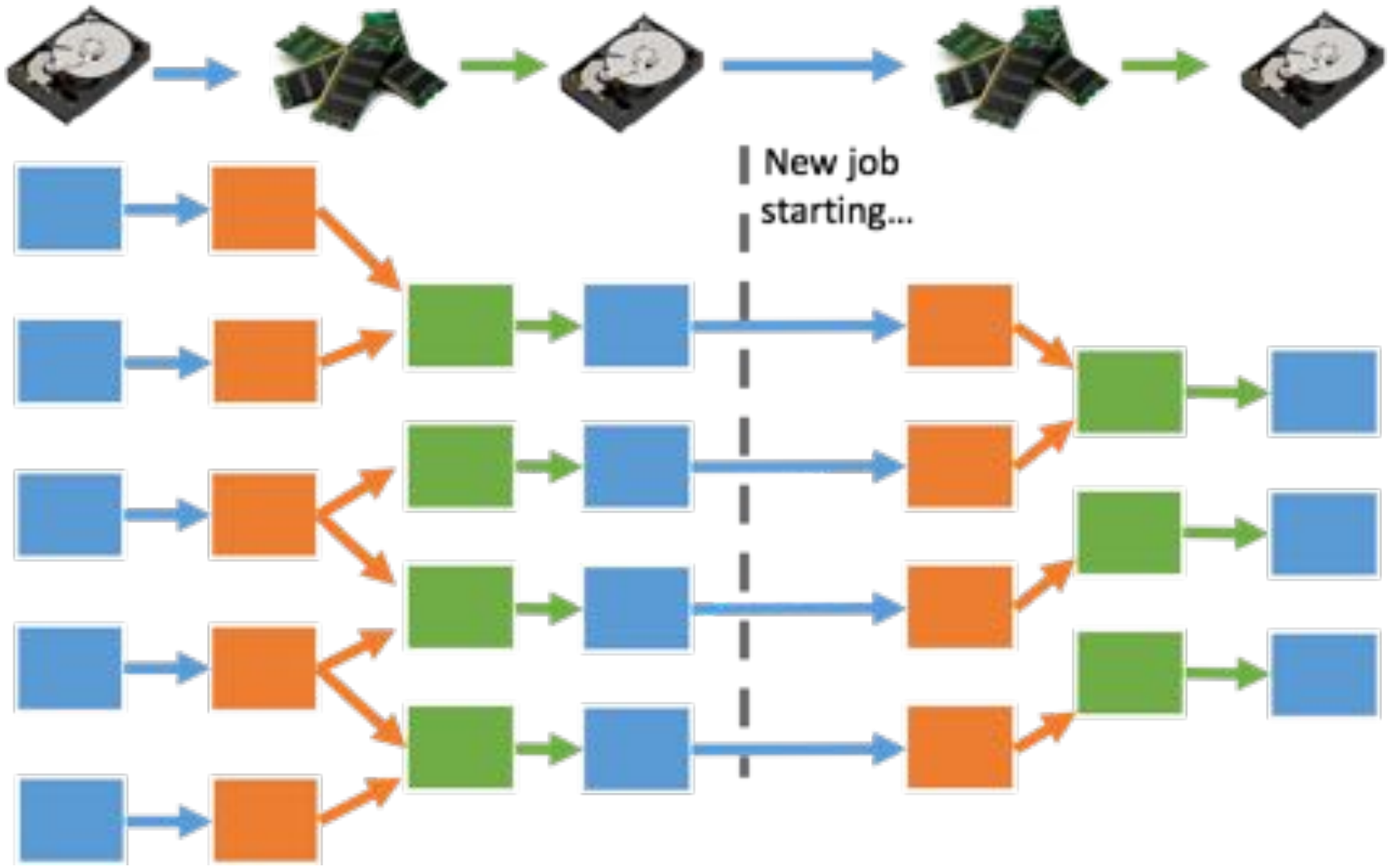
```
JavaRDD<String> textFile = sc.textFile("hdfs://...");

JavaPairRDD<String, Integer> counts = textFile
  .flatMap(s -> Arrays.asList(s.split(" ")).iterator())
  .mapToPair(word -> new Tuple2<>(word, 1))
  .reduceByKey((a, b) -> a + b);

counts.saveAsTextFile("hdfs://...");
```

Fluxo dos Dados no MapReduce



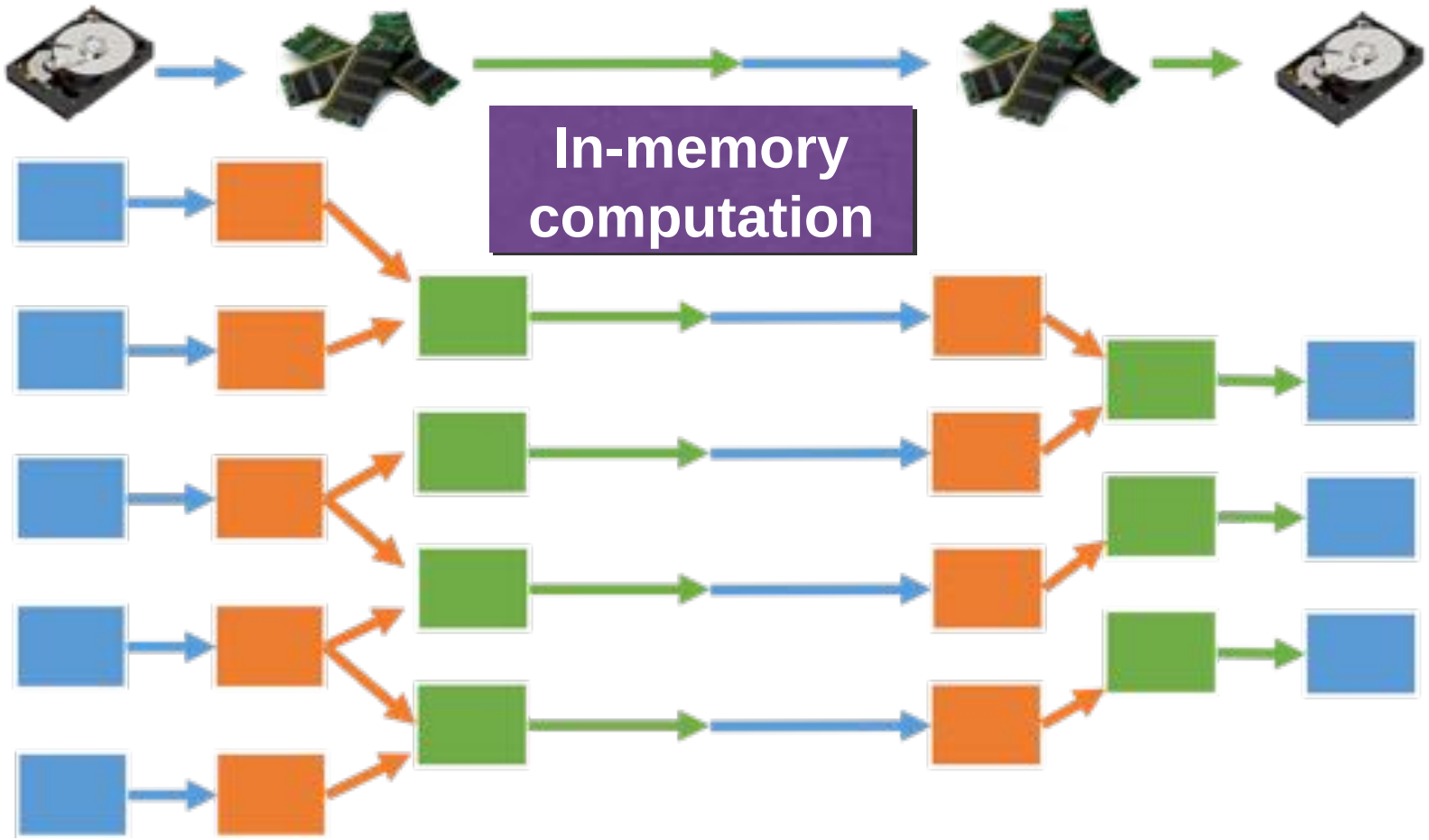
Fluxo dos Dados no MapReduce

$$F \bowtie D_1 \bowtie D_2 \ldots$$



Fluxo dos Dados no Spark

$$F \bowtie D_1 \bowtie D_2 \ldots$$



Evolução: SQL on Hadoop

Criação de uma camada para prover a utilização de SQL

Motivação

- SQL é amplamente utilizados há muitos anos

Dificuldade em escrever programas em MapReduce/Spark

- SQL é traduzido para funções MapReduce e RDDs
  - HiveQL
  - Spark SQL e Data Frames

Junção em MapReduce

- Map-side Join
  - Junção na função Map
- Reduce-side Join
  - Junção na função Reduce

Map-side Join

- Junção é realizada na função Map

```
SELECT a, b
FROM S, T
WHERE  S.c = T.c
```

S

s	c	a
1	5	8
2	6	7
3	6	4
4	4	7

T

t	c	b
5	6	4
7	4	2
8	4	1
10	5	4

Map-side Join

- Junção é realizada na função Map

```
SELECT a, b
FROM S, T
WHERE S.c = T.c
```

S

s	c	a
1	5	8
2	6	7
3	6	4
4	4	7

S_1

s	c	a
4	4	7
1	5	8

S_2

s	c	a
2	6	7
3	6	4

T

t	c	b
5	6	4
7	4	2
8	4	1
10	5	4

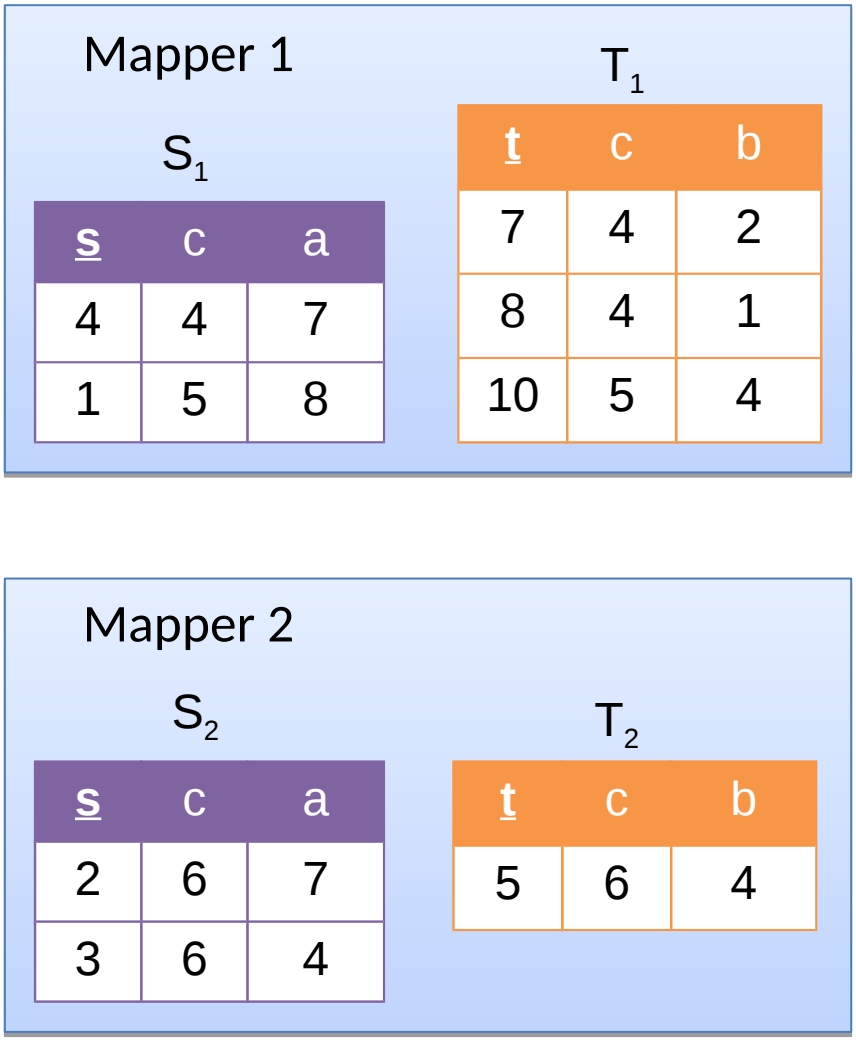
T_1

t	c	b
7	4	2
8	4	1
10	5	4

T_2

t	c	b
5	6	4

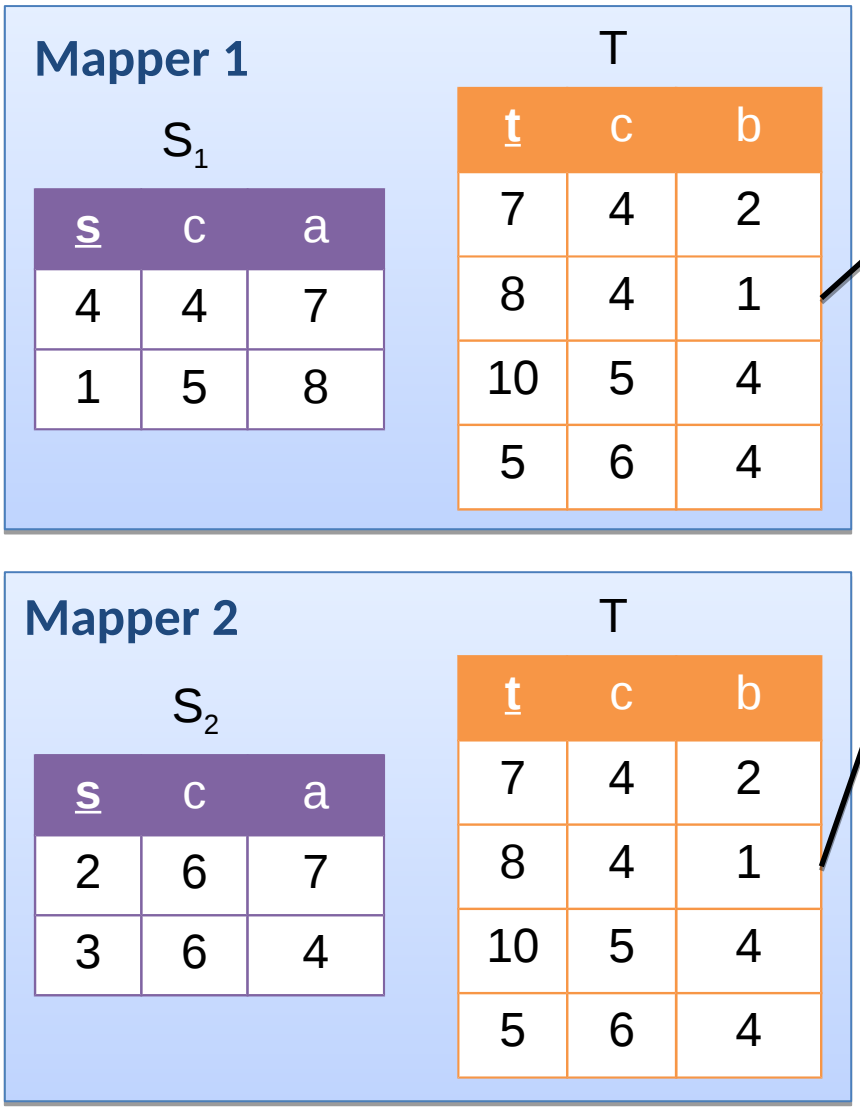
Map-side Join



- Dados são particionados e ordenados pela chave de junção (atributo c)
- Blocos correspondentes de cada arquivo são processados por uma única tarefa Mapper
- A função Map é aplicada sobre um dos blocos (ex: S_1), enquanto o outro bloco correspondente (ex: T_1) é lido dentro da tarefa Mapper
- Cada Mapper possui os dados necessários para realizar a junção de seus blocos

Map-side Join

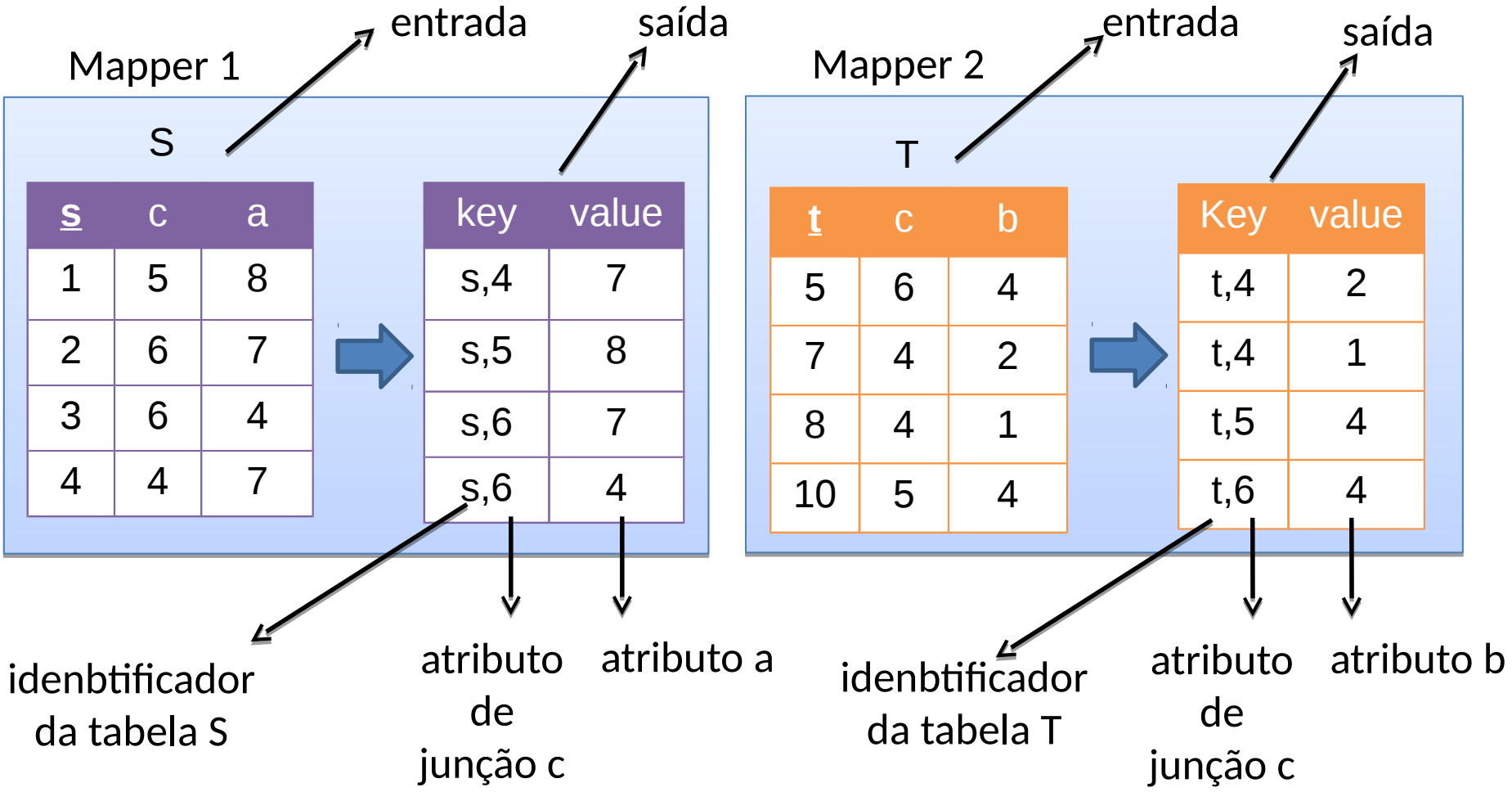
Memory-backed Join



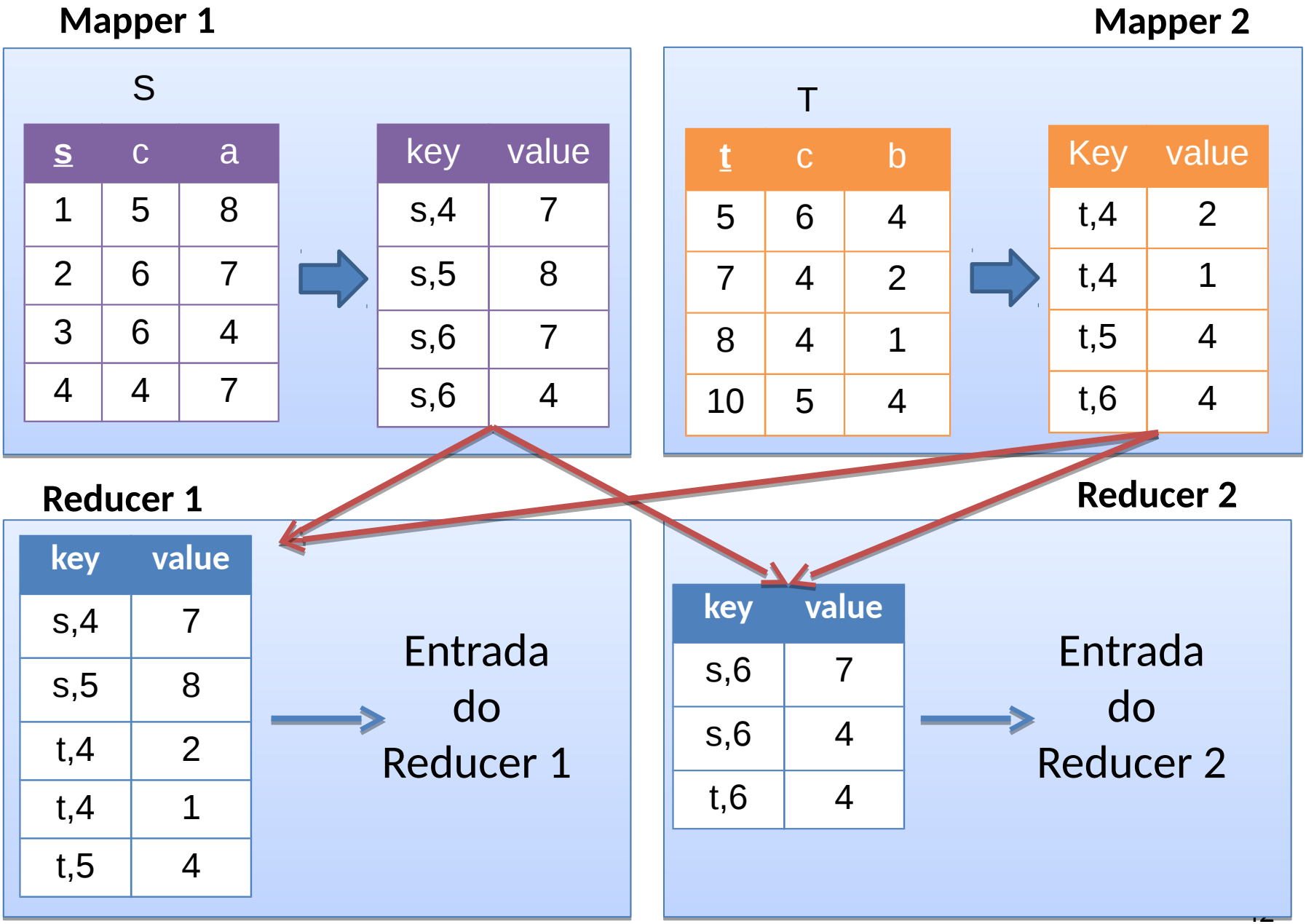
Dados em memória primária local

- Tabela menor (ex: tabela T) é armazenada na memória primária local de cada nó
- Blocos da tabela maior (ex: S) são processados nos diferentes mappers
- Cada mappers tem acesso a todos os dados da tabela menor (ex: tabela T)

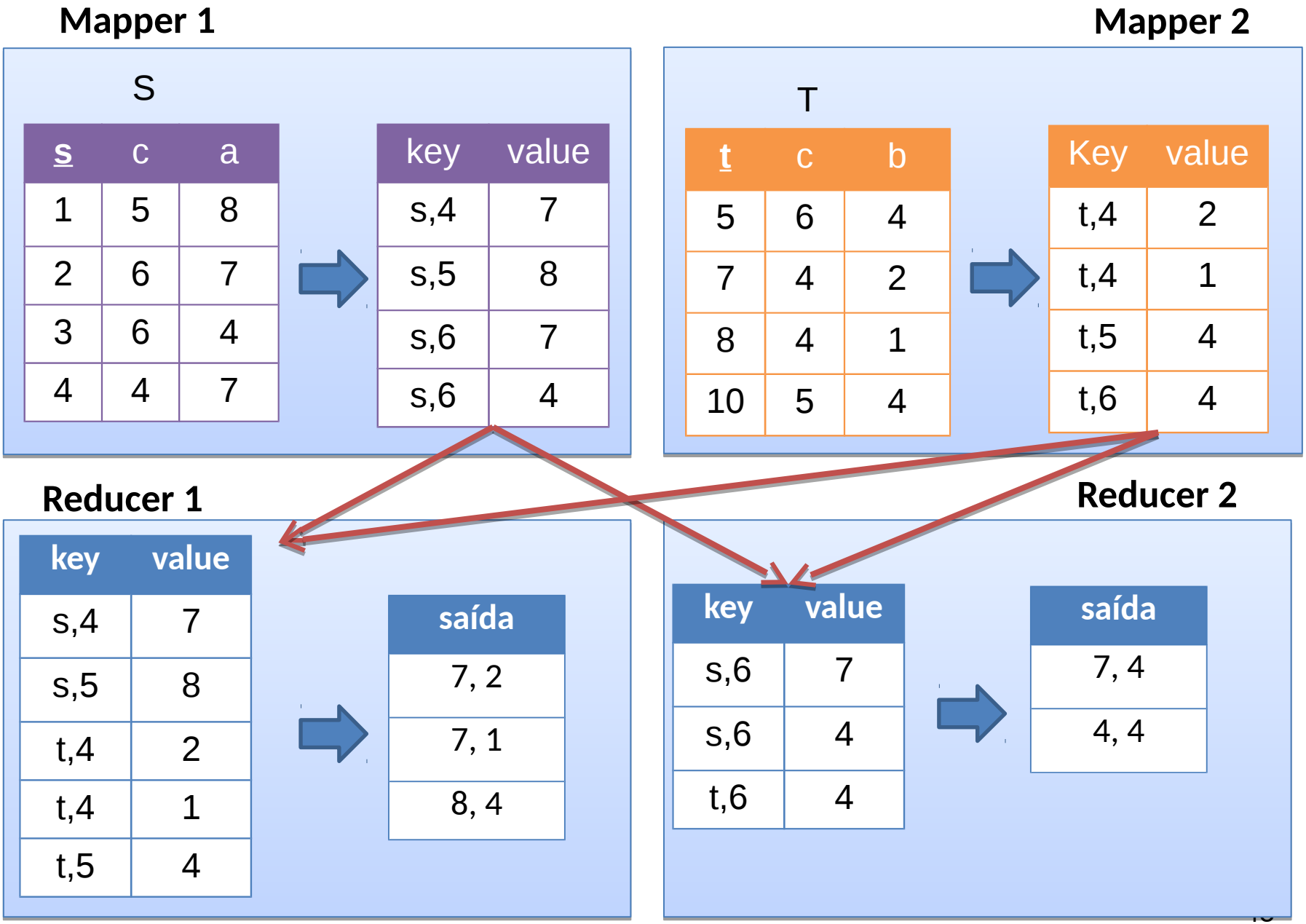
Reduce-side Join



Reduce-side Join



Reduce-side Join



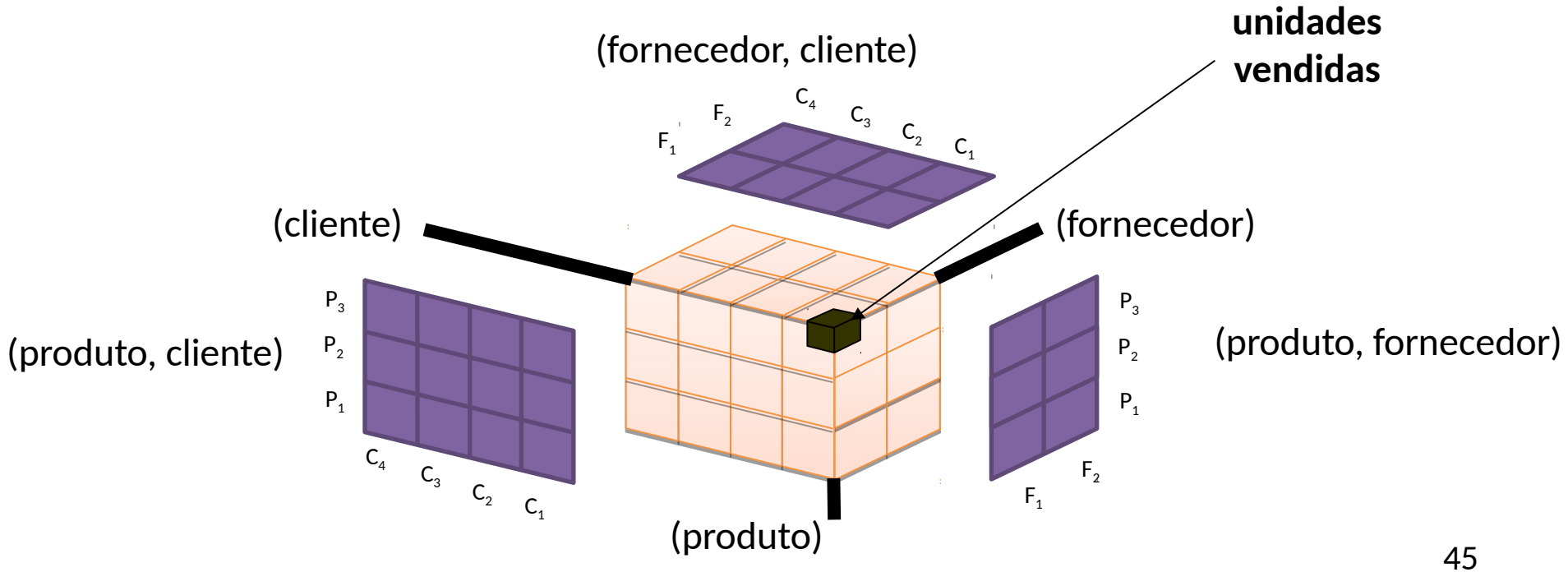
Junção em MapReduce

- **Map-side Join**
  - **Desvantagem:** aplicável quando o conjunto pode ser ordenado e particionado pelo atributo de junção
  - Memory-backed Join
    - **Desvantagem:** aplicável quando uma das tabelas é pequena e cabe na memória primária de cada nó
  - **Vantagem:** processamento local, dispensando a fase de shuffling
- **Reduce-side Join**
  - **Vantagem:** aplicável em qualquer conjunto de tabelas
  - **Desvantagem:** necessidade da fase de shuffling

Data Warehouse

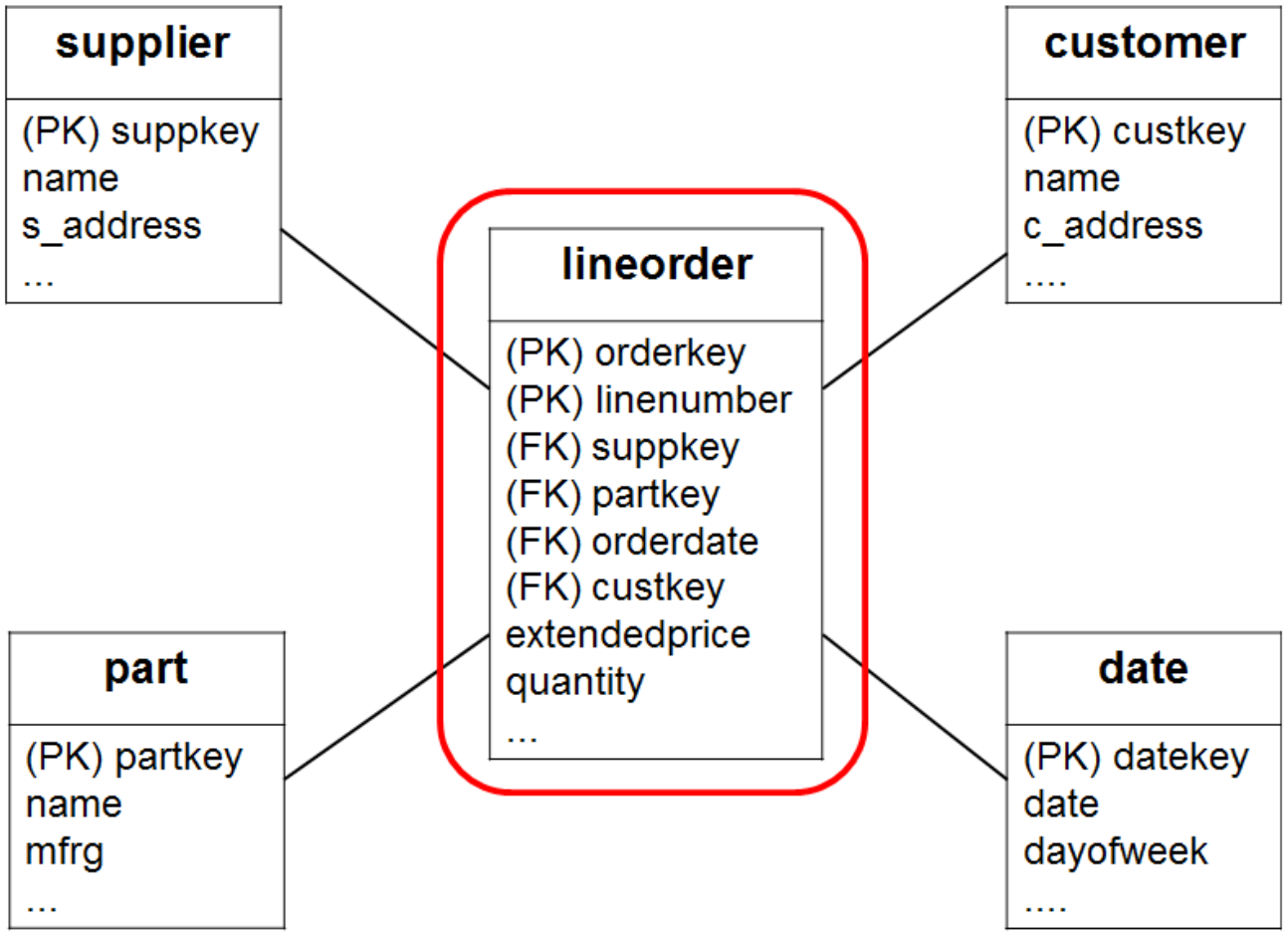
- Banco de dados voltado ao processamento analítico para a tomada de decisão

- Modelagem multidimensional
  - Medidas numéricas: objetos de análise
  - Dimensões: perspectiva/contexto para as análises



Esquema Estrela

- *Star Schema Benchmark* (SSB)



Consulta de Junção Estrela

- Consulta Q3.2 do SSB

```
SELECT  c_city, s_city, d_year, SUM(lo_revenue) as revenue
FROM    Lineorder, Supplier,  Customer, Date
WHERE   lo_custkey = c_custkey
    AND   lo_suppkey = s_suppkey
    AND   lo_orderdate = d_datekey
    AND   c_nation = 'UNITED STATES'
    AND   s_nation = 'UNITED STATES'
    AND   d_year >= 1992 AND d_year <= 1997
GROUP BY c_city, s_city, d_year
ORDER BY c_city, s_city, d_year
```

Consulta de Junção Estrela

- Consulta Q3.2 do SSB

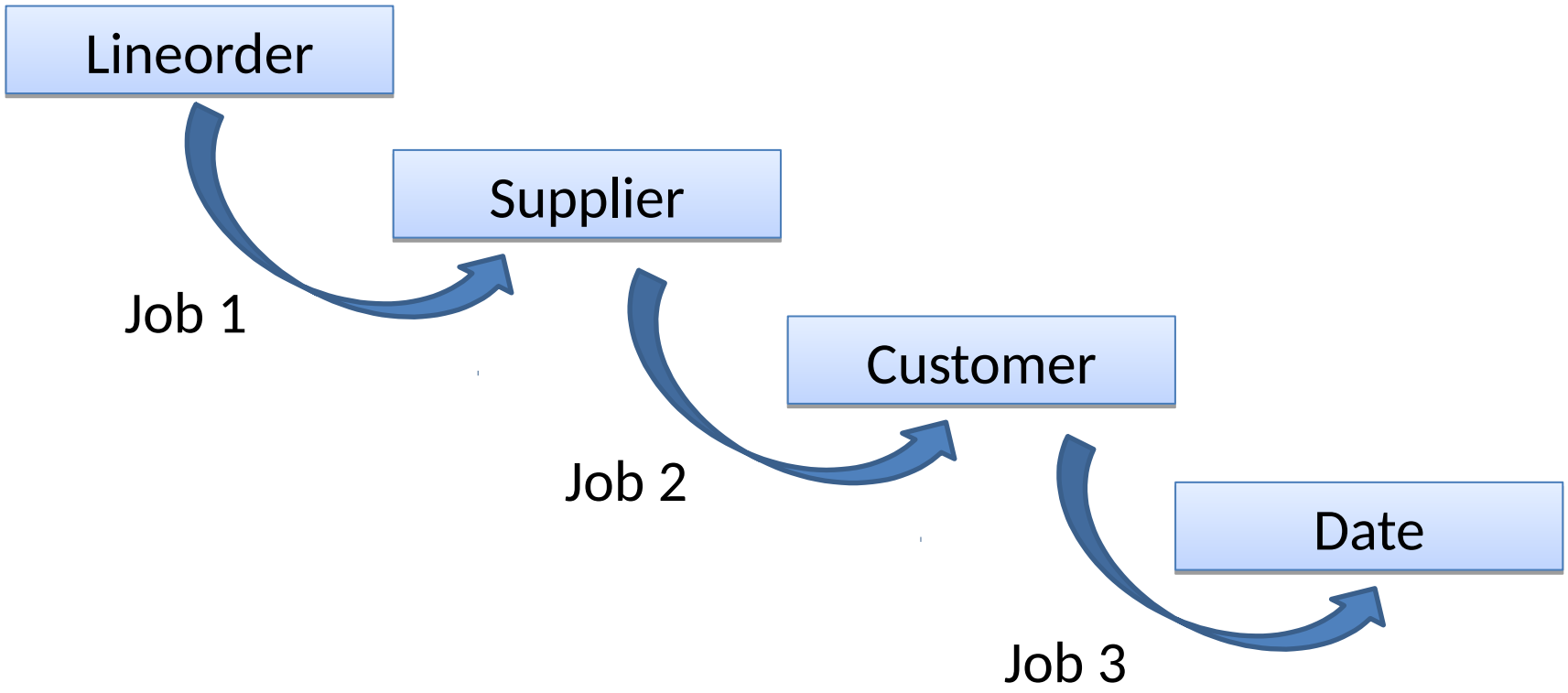
```
SELECT  c_city, s_city, d_year, SUM(lo_revenue) as revenue
FROM    Lineorder, Supplier, Customer, Date
WHERE   lo_custkey = c_custkey
    AND lo_suppkey = s_suppkey
    AND lo_orderdate = d_datekey
    AND c_nation = 'UNITED STATES'
    AND s_nation = 'UNITED STATES'
    AND d_year >= 1992 AND d_year <= 1997
GROUP BY c_city, s_city, d_year
ORDER BY c_city, s_city, d_year
```

cláusulas de junção

cláusulas de filtragem

Sequência de Junções Binárias em MapReduce

- Um job MapReduce para cada junção

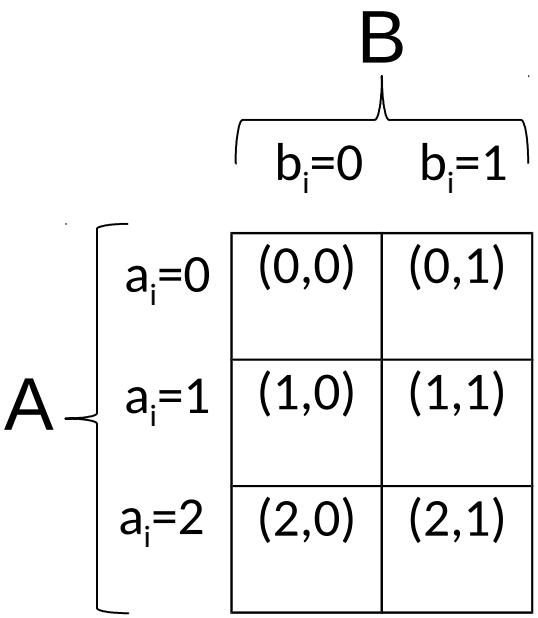


Algoritmo de Afrati e Ullman (2010)

- Proposta: realizar todas as junções em apenas um job

$$S(s) \bowtie U(s, t) \bowtie T(t)$$

- O domínio do atributo s é dividido em A blocos, enquanto que o domínio de t é dividido em B blocos
- O número de processos reducers é dado por AB



Supondo que A=3 e B=2, temos um total de 6 reducers

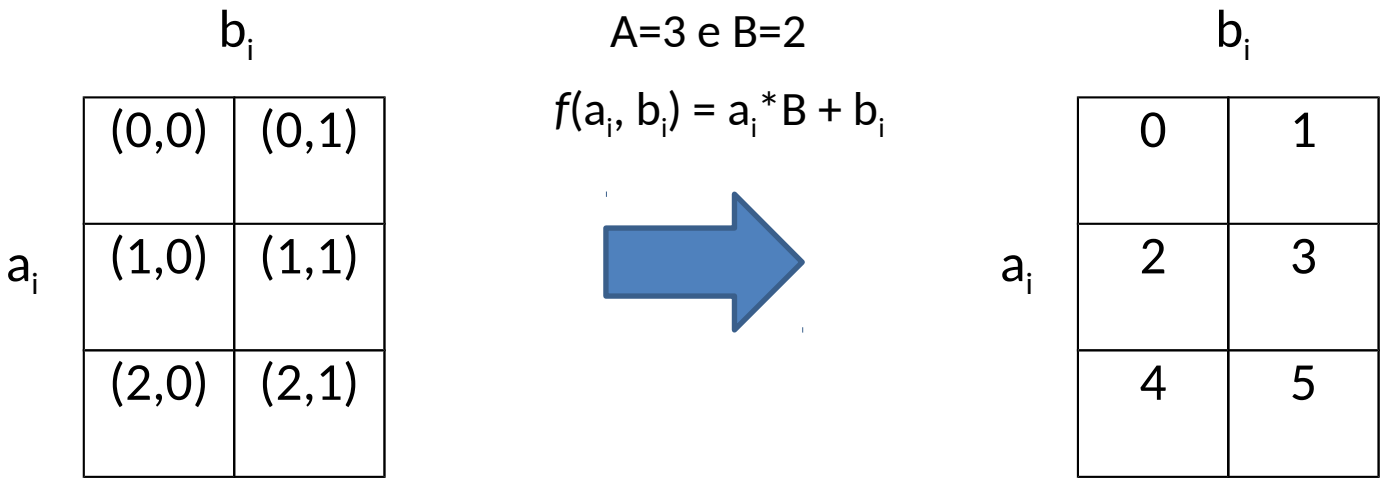
Cada processo reduce é identificado por um par (a_i, b_i)

Algoritmo de Afrati e Ullman (2010)

- Proposta: realizar todas as junções em apenas um job

$$S(s) \bowtie U(s, t) \bowtie T(t)$$

- O processo reduce para o qual uma tupla deve ser enviada é identificado por dois valores, a e b, determinados a partir dos atributos s e t (atributos de junção)



Algoritmo de Afrati e Ullman (2010)

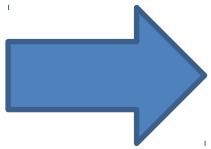
- Proposta: realizar todas as junções em apenas um job

$S(s) \bowtie U(s, t) \bowtie T(t)$

- Para cada valor s_i do atributo s, $a_i = mod(s_i, A)$
- Para cada valor t_i do atributo t, $b_i = mod(t_i, B)$
- Reducer identificado por uma função $f(a_i, b_i) = a_i * B + b_i$

a_i \ b_i		
	(0,0)	(0,1)
	(1,0)	(1,1)
	(2,0)	(2,1)

A=3 e B=2



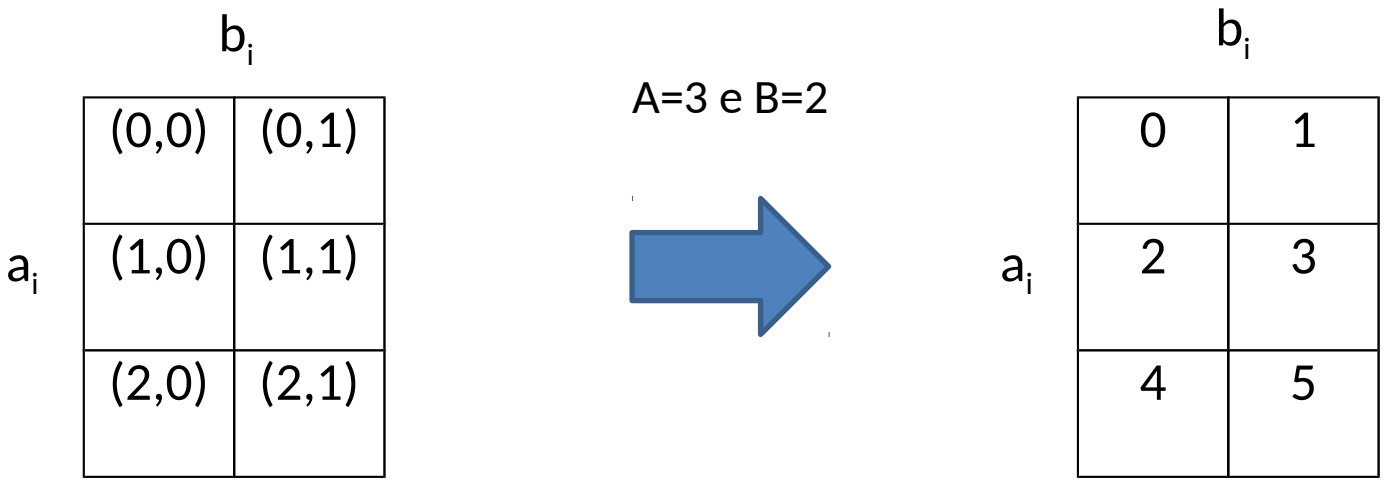
a_i \ b_i		
	0	1
	2	3
	4	5

Algoritmo de Afrati e Ullman (2010)

- Proposta: realizar todas as junções em apenas um job

$$S(s) \bowtie U(s, t) \bowtie T(t)$$

- Cada tupla de S precisa ser enviada para todos os reducers identificados por um determinado valor a_i
- Cada tupla de T precisa ser enviada para todos os reducers identificados por um um determinado valor b_i



Algoritmo de Afrati e Ullman (2010)

- Exemplo: $S(s) \bowtie U(s, t) \bowtie T(t)$

Id do processo reduce

$f(a_i, b_i) = a_i * B + b_i$

A=3 e B=2

S

s
1
2
3
4

U

s	t
2	5
3	7
1	8
4	10

T

t
5
7
8
10

Tuplas da tabela U são enviadas para um único reduce

S

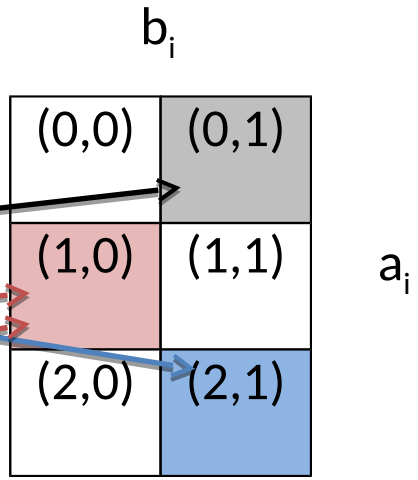
s
$a_i = 1$
$a_i = 2$
$a_i = 0$
$a_i = 1$

U

s	t
$a_i = 2$	$b_i = 1$
$a_i = 0$	$b_i = 1$
$a_i = 1$	$b_i = 0$
$a_i = 1$	$b_i = 0$

T

t
$b_i = 1$
$b_i = 1$
$b_i = 0$
$b_i = 0$



Algoritmo de Afrati e Ullman (2010)

- Exemplo: $S(s) \bowtie U(s, t) \bowtie T(t)$

Id do processo reduce

$f(a_i, b_i) = a_i * B + b_i$

A=3 e B=2

S

s
1
2
3
4

U

s	t
2	5
3	7
1	8
4	10

T

t
5
7
8
10

Cada tupla da tabela S é enviada para B reducers (todos reducers de uma mesma linha)

S

s
$a_i = 1$
$a_i = 2$
$a_i = 0$
$a_i = 1$

U

s	t
$a_i = 2$	$b_i = 1$
$a_i = 0$	$b_i = 1$
$a_i = 1$	$b_i = 0$
$a_i = 1$	$b_i = 0$

T

t
$b_i = 1$
$b_i = 1$
$b_i = 0$
$b_i = 0$

Algoritmo de Afrati e Ullman (2010)

- Exemplo: $S(s) \bowtie U(s, t) \bowtie T(t)$

Id do processo reduce

$f(a_i, b_i) = a_i * B + b_i$

A=3 e B=2

S

s
1
2
3
4

U

s	t
2	5
3	7
1	8
4	10

T

t
5
7
8
10

Cada tupla da tabela T é enviada para A reducers (todos reducers de uma mesma coluna)

S

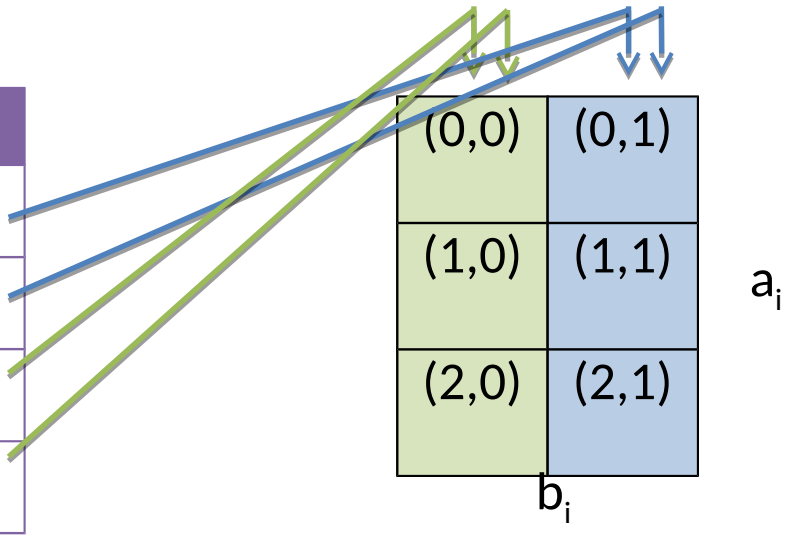
s
$a_i = 1$
$a_i = 2$
$a_i = 0$
$a_i = 1$

U

s	t
$a_i = 2$	$b_i = 1$
$a_i = 0$	$b_i = 1$
$a_i = 1$	$b_i = 0$
$a_i = 1$	$b_i = 0$

T

t
$b_i = 1$
$b_i = 1$
$b_i = 0$
$b_i = 0$



Algoritmo de Afrati e Ullman (2010)

Reducer 0

Chave	Valor
S|3	null
T|8	null
T|10	null

Reducer 1

Chave	Valor
S|3	null
T|5	null
T|7	null
U|3,7	null

Reducer 2

Chave	Valor
S|1	null
S|4	null
T|8	null
T|10	null
U|1,8	null
U|4,10	null

Reducer 3

Chave	Valor
S|1	null
S|4	null
T|5	null
T|7	null

Reducer 4

Chave	Valor
S|3	null
T|8	null
T|10	null

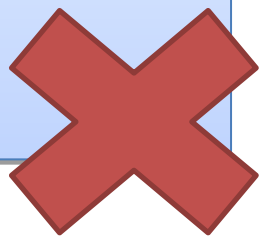
Reducer 5

Chave	Valor
S|3	null
T|5	null
T|7	null
U|2,5	null

Algoritmo de Afrati e Ullman (2010)

Reducer 0

Chave	Valor
S|3	null
T|8	null
T|10	null



Reducer 1

Chave	Valor
S|3	null
T|5	null
T|7	null
U|3,7	null

Reducer 2

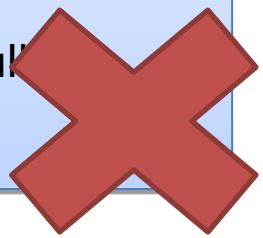
Chave	Valor
S|1	null
S|4	null
T|8	null
T|10	null
U|1,8	null
U|4,10	null

Reducer 3

Chave	Valor
S|1	null
S|4	null
T|5	null
T|7	null

Reducer 4

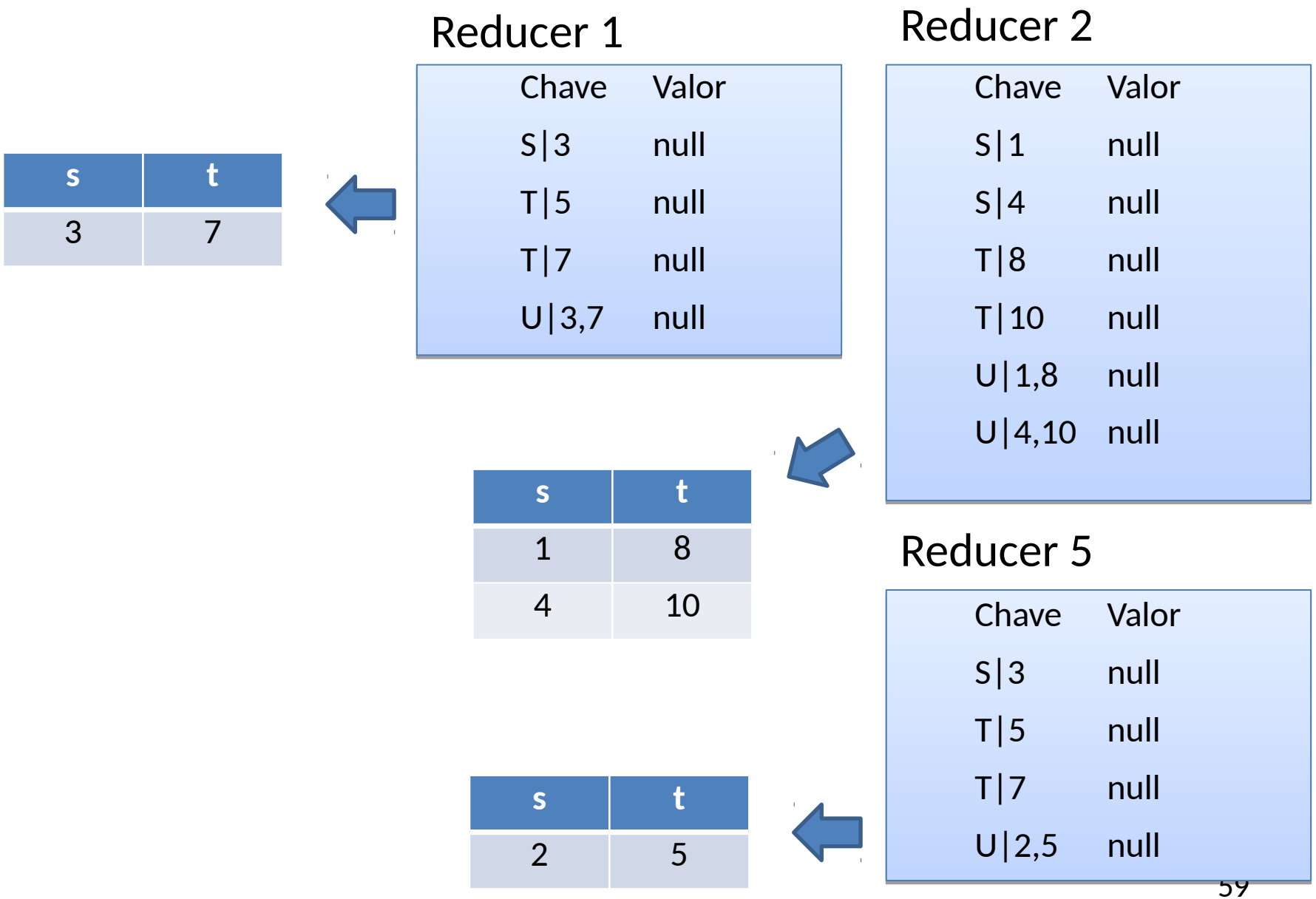
Chave	Valor
S|3	null
T|8	null
T|10	null



Reducer 5

Chave	Valor
S|3	null
T|5	null
T|7	null
U|2,5	null

Algoritmo de Afrati e Ullman (2010)



Algoritmo de Afrati e Ullman (2010)

Vantagem

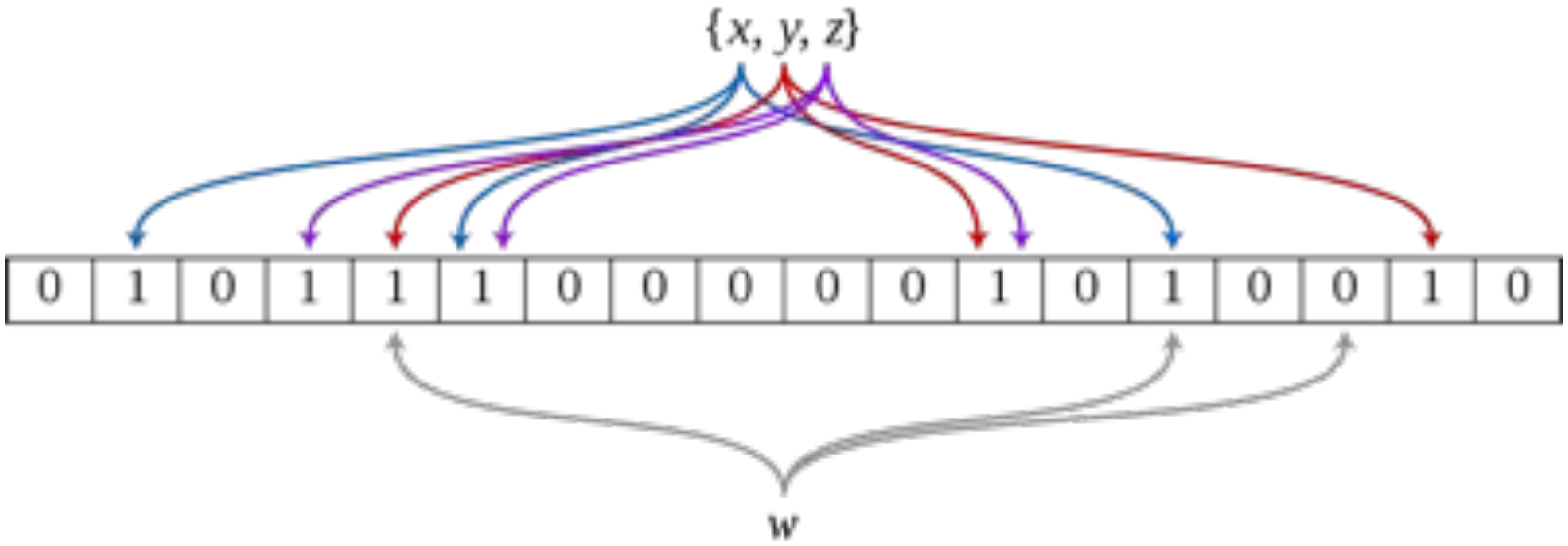
- realiza todas as junções em apenas um job MapReduce

Desvantagem

- Replicação de dados das tabelas de dimensão (S e T no exemplo)
- Caso existam filtros nas tabelas de dimensão, tuplas da tabela de fatos (U no exemplo) são enviadas para os reducers desnecessariamente
  - Solução: uso de Bloom-Filters para filtrar dados desnecessários da tabela de fatos

Bloom-Filters

- Construído sobre um conjunto de itens
- Vetor de bits de n posições
- Bits são setados para 1 por meio de k funções hash



- **Desvantagem:** falsos positivos

Demo: http://www.jasondavies.com/bloomfilter/

Oportunidades

Data Scientist

- Alguns conhecimentos necessários
  - Saber programação
  - Ser capaz de criar modelos estatísticos
  - Compreender as diferentes plataformas de Big Data
- Usualmente esse profissional é formado em Estatística, Matemática, Física ou Ciências da Computação

Imagem: http://brmtecnologia.files.wordpress.com/2012/07/bi-analyze.png

Oportunidades

Job description

Descrição da vaga

Imagine que você tenha que analisar e identificar comportamentos e tendências em uma base de mais de 60 milhões de usuários únicos, relacionando dados como idade, sexo, classe social, estado civil, geo localização e interesses de consumo classificados em mais de 180 tipos em mais de 30GB de logs por dia!

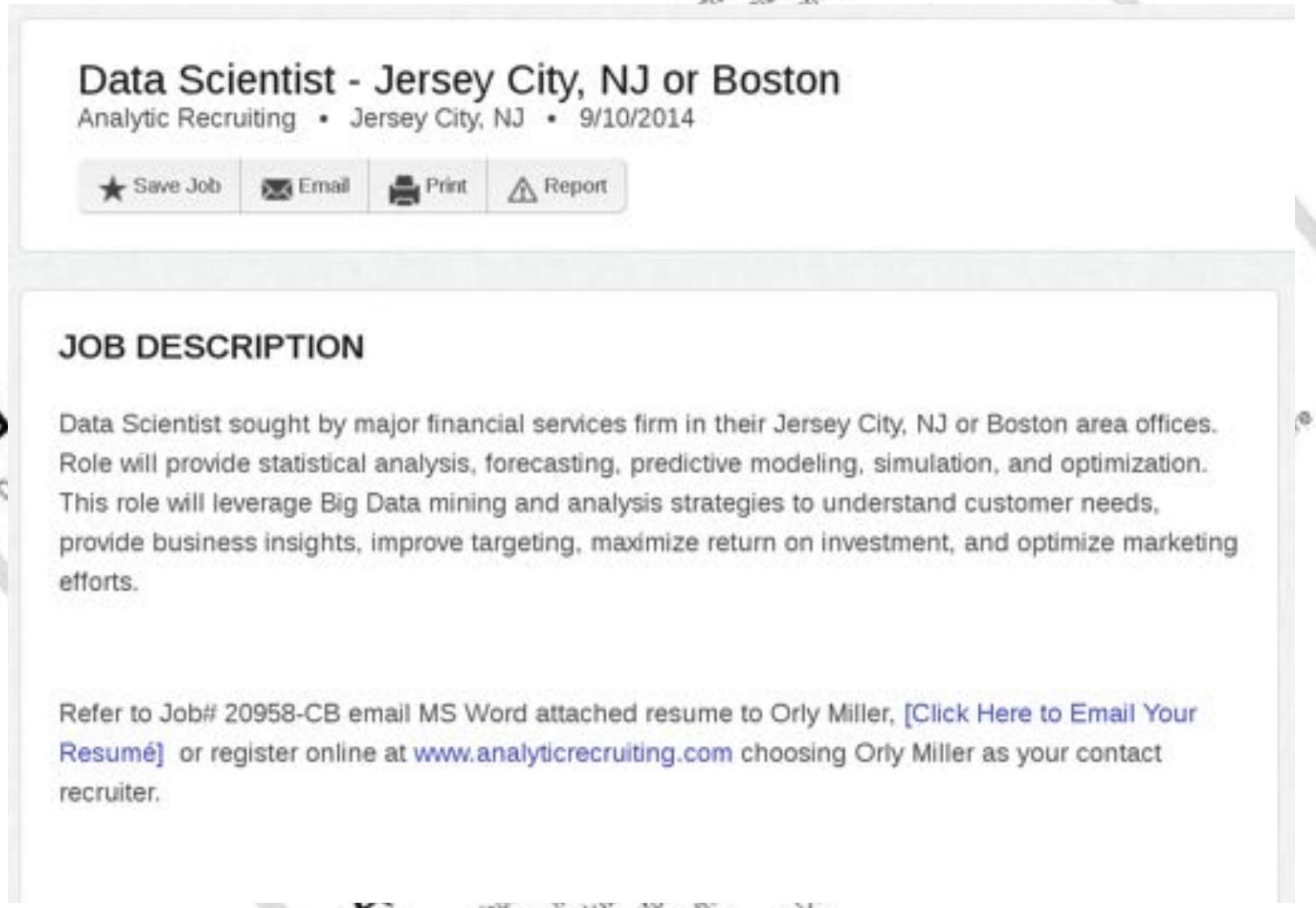
Como gerar um modelo matemático que represente esses grupos? Como criar um sistema de recomendação que aumente a rentabilidade e o aproveitamento de inventário?

Para isso estamos buscando um Data Scientist que ficará responsável por organizar a arquitetura de Big Data, analisar os dados, criar Algoritmos de Machine Learning e modelos Matemáticos que ajudem na otimização dessa plataforma.

Desired Skills and Experience

- Python, C++, R
- MySQL, MongoDB, Hadoop, ElasticSearch, Redis
- Linux, GitHub, AWS
- Técnicas de Information Retrieval e Estatística (Bayes, Markov Chain, etc)
- Sistemas de Recomendação como Recomendação Reciproca, Multi-Armed Bandit, entre outros
- Machine Learning (SVM, Deep Learning, etc)
- Inglês Fluente

Oportunidades



Data Scientist - Jersey City, NJ or Boston

Analytic Recruiting • Jersey City, NJ • 9/10/2014

Save Job | Email | Print | Report

JOB DESCRIPTION

Data Scientist sought by major financial services firm in their Jersey City, NJ or Boston area offices. Role will provide statistical analysis, forecasting, predictive modeling, simulation, and optimization. This role will leverage Big Data mining and analysis strategies to understand customer needs, provide business insights, improve targeting, maximize return on investment, and optimize marketing efforts.

Refer to Job# 20958-CB email MS Word attached resume to Orly Miller, [Click Here to Email Your Resumé] or register online at www.analyticrecruiting.com choosing Orly Miller as your contact recruiter.

Oportunidades

Data Scientist

Vaco Technology • Santa Clara, CA • 9/9/2014

Save Job | Email | Print | Report

JOB DESCRIPTION

Vaco has been engaged by a famous technology leader to identify a savvy **Data Scientist** for a newly formed "Data Science Team" that is building and creating the Big Data initiatives for the entire enterprise.

Join a highly visible team that is responsible for critical, high profile projects such as the initial "data cleansing" project. The Data Science Team is creating a "Data Lake" and wants to ensure that PSI (personal sensitive information including credit card, SS# details, etc) gets extracted before making its way to the lake.

Required Experience for the Data Scientist:

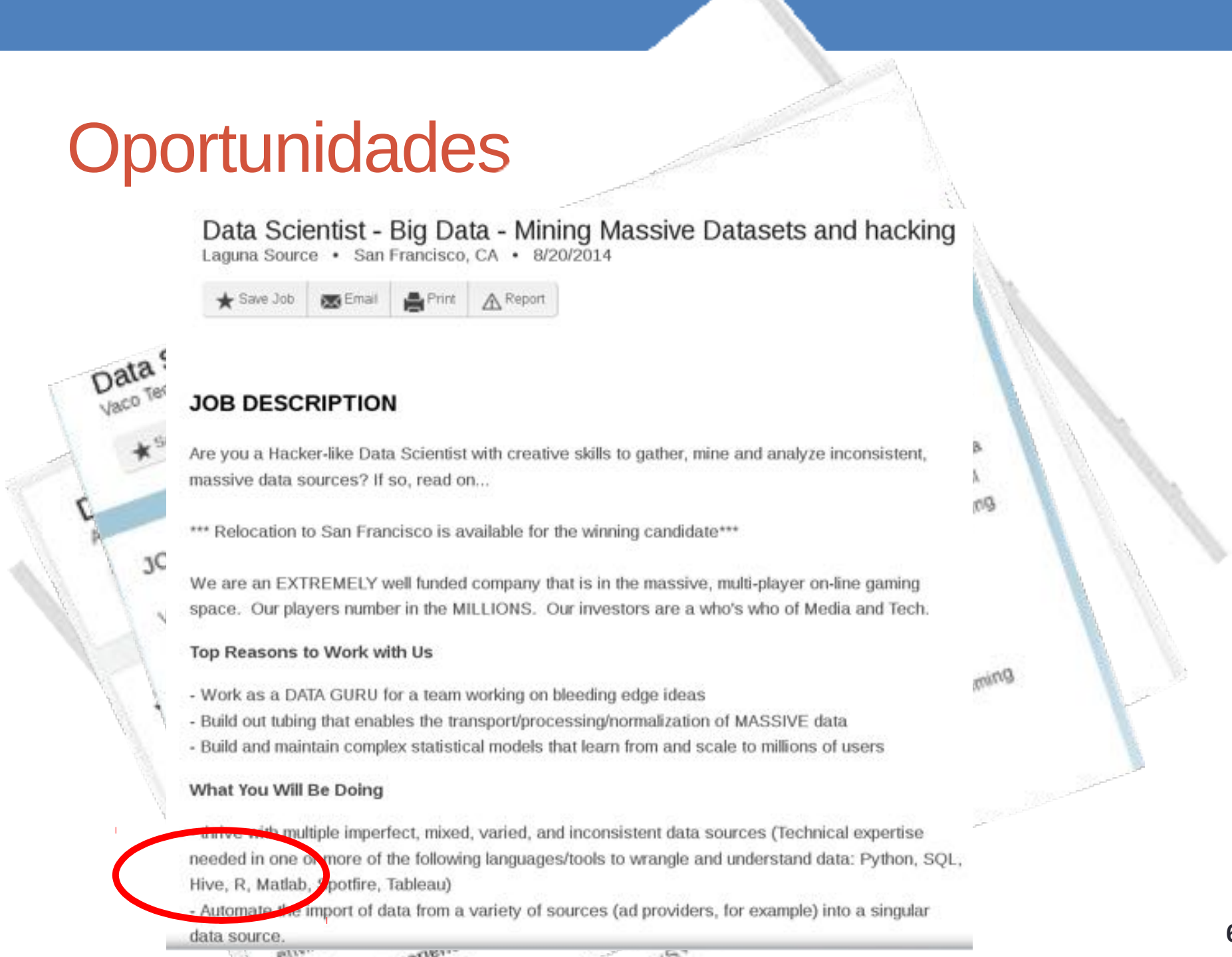
- 10+ years development-specific experience ("heads-down programmer")

- Expert level technical foundation including more than one advanced structured programming environment such as APEX, Force.com, Python, .Net, or Java

Preferred Experience for the Data Scientist:

- Hadoop Development

Oportunidades



Data Scientist - Big Data - Mining Massive Datasets and hacking

Laguna Source • San Francisco, CA • 8/20/2014

Save Job | Email | Print | Report

JOB DESCRIPTION

Are you a Hacker-like Data Scientist with creative skills to gather, mine and analyze inconsistent, massive data sources? If so, read on...

*** Relocation to San Francisco is available for the winning candidate***

We are an EXTREMELY well funded company that is in the massive, multi-player on-line gaming space. Our players number in the MILLIONS. Our investors are a who's who of Media and Tech.

Top Reasons to Work with Us

- Work as a DATA GURU for a team working on bleeding edge ideas
- Build out tubing that enables the transport/processing/normalization of MASSIVE data
- Build and maintain complex statistical models that learn from and scale to millions of users

What You Will Be Doing

- Work with multiple imperfect, mixed, varied, and inconsistent data sources (Technical expertise needed in one or more of the following languages/tools to wrangle and understand data: Python, SQL, Hive, R, Matlab, Spotfire, Tableau)
- Automate the import of data from a variety of sources (ad providers, for example) into a singular data source.

Referências

http://www.sciencedaily.com/releases/2013/05/130522085217.htm

http://www-3.unipv.it/ingegneria/copisteria_virtuale/motta/doss/13-PV-DoES-18-BigData-v3.pdf

http://hadoop.apache.org/

http://spark.apache.org/

http://azure.microsoft.com/en-us/pricing/details/storage/

http://thoughtsoncloud.com/2014/02/top-7-most-common-uses-of-cloud-computing/

http://pt.slideshare.net/laodias/os-cinco-vs-do-big-data

http://www.microsoft.com/en-us/news/presskits/cloud/docs/the-economics-of-the-cloud.pdf

http://www.cloudproviderusa.com/the-difference-between-public-cloud-private-cloud/

http://observatorio.inweb.org.br/dengueapp?next=/dengueapp/relatorio

https://www.youtube.com/watch?v=jyx8iP5tfCI&feature=youtu.be

Afrati, F. N.; Ullman, J. D. Optimizing joins in a map-reduce environment. In: Proceedings of the 13th International Conference on Extending Database Technology (EDBT 2010), 2010. p. 99–110.

Obrigada