

# Frameworks de Big Data

**Eduardo Viegas**



ESCOLA  
**POLITÉCNICA**

# Eduardo Viegas

- Formação
  - Graduado em Ciência da Computação – PUCPR
  - Mestre em Informática – PUCPR
  - Doutorado em Informática – PUCPR
    - Universidade de Lisboa (Distributed Research Team)
- Membro da Intel Strategic Research Alliance (ISRA) Brasil
  - Energy-efficient Security for SoC Devices
- Atuação
  - Computação em nuvem
  - Big Data
  - Segurança da informação
  - Aprendizagem de máquina
  - Pesquisa na indústria e academia
- Professor pesquisador no PPGIa PUCPR

# Cronograma das aulas

Data	Conteúdo
01/02	Big Data, Hadoop e HDFS
15/02	HDFS, MapReduce
29/02	HDFS, MapReduce, Apache Spark
14/03	HDFS, Apache Spark
21/03	HDFS, Apache Spark, Projeto

# Avaliação

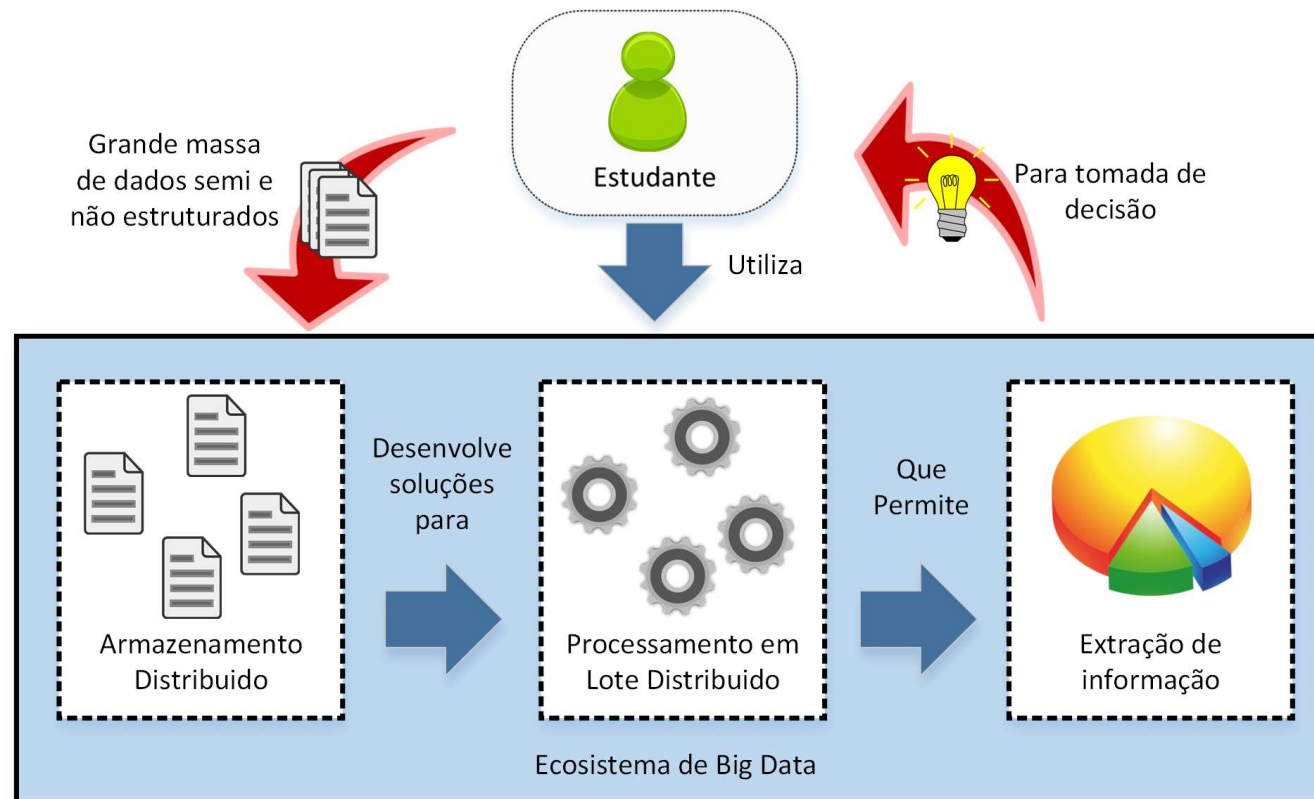
- 30% da nota
  - 2 atividades em sala
- 70% da nota
  - Trabalho em equipe
  - Resolução de problema prático envolvendo Big Data e tecnologias estudadas
  - Relatório em formato SBC de 10 páginas

# Antes de começarmos =)

- Quem aqui sabe programar?
  - Python?
  - BASH?

# Antes de começarmos =)

- *O que faremos nesta disciplina?*



# Antes de começarmos =)

- *Ao final desta disciplina você deverá saber*

Reconhecer cenários de aplicação de Big Data, considerando dados semi e não estruturados.

Reconhecer os principais componentes dos ambientes de HDFS e HADOOP e suas funcionalidades.

Aplicar técnicas de processamento em Lote para análise de dados em Big Data.

Aplicar técnicas de estruturação e análise de dados estruturados ou semi estruturados

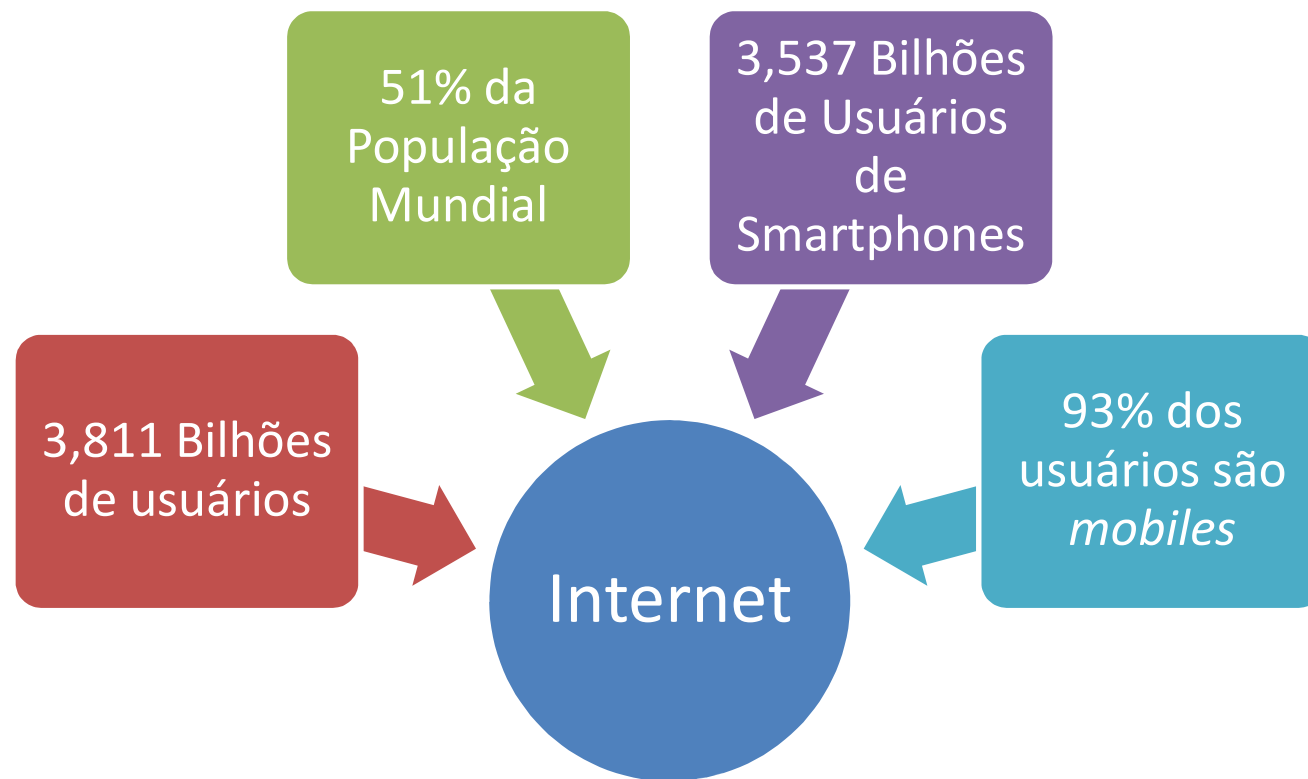
Desenvolver técnicas de Big Data para análise de dados

# Agenda

- **Big Data**
- Hadoop
  - HDFS
  - MapReduce
  - Apache Spark



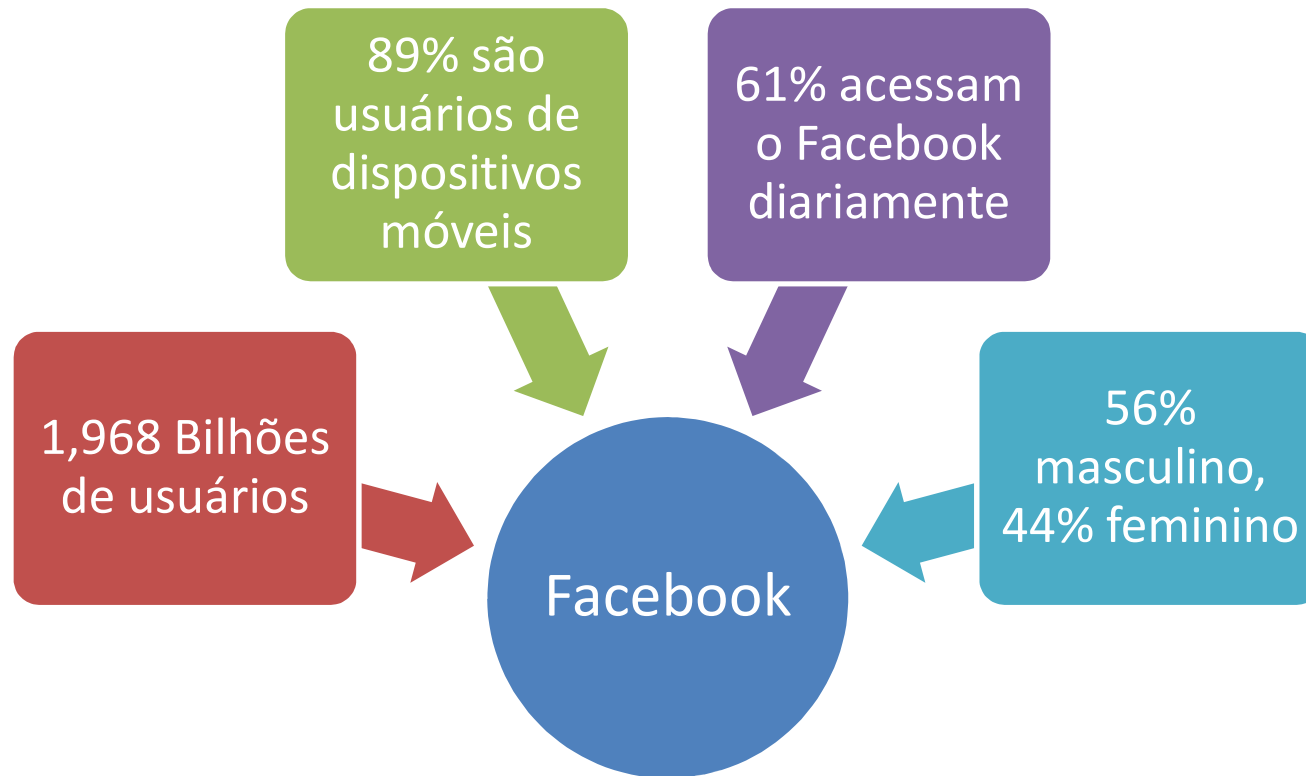
# Massa de dados ao longo do tempo



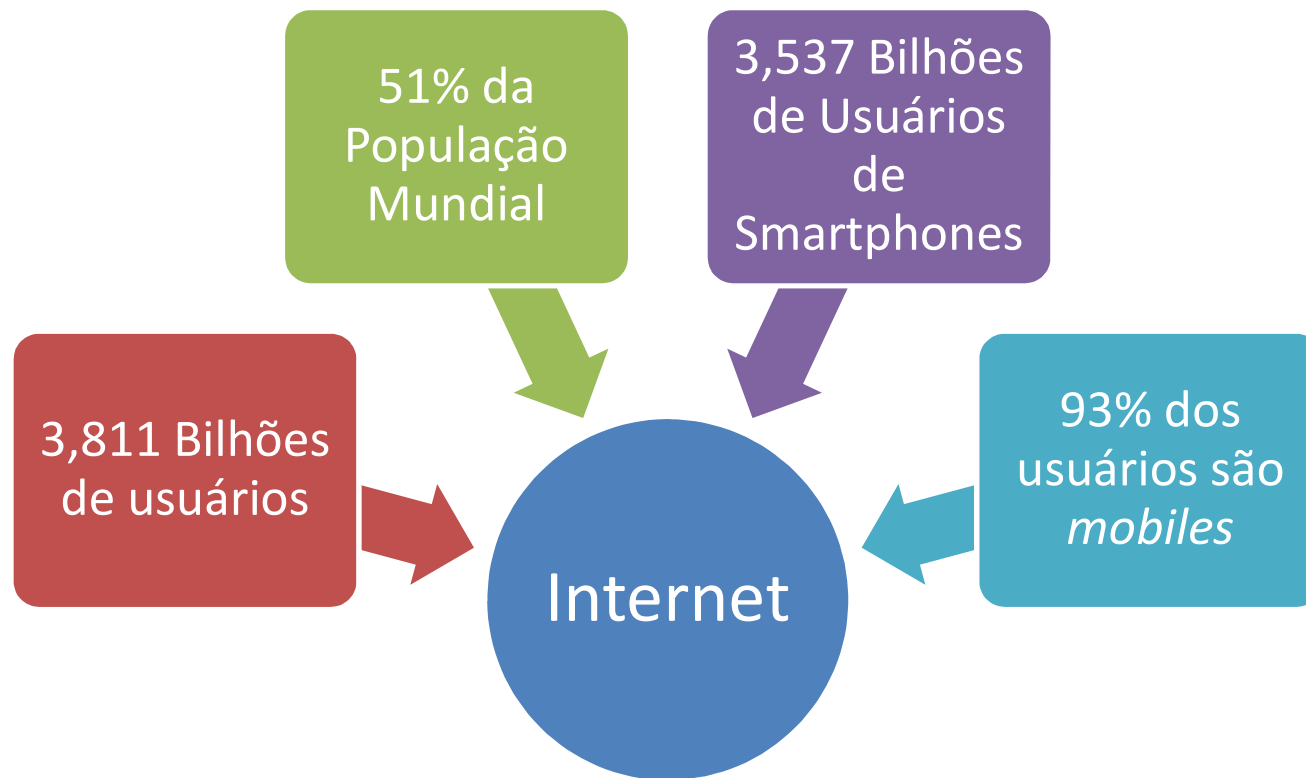
# Massa de dados ao longo do tempo



# Massa de dados ao longo do tempo



# Massa de dados ao longo do tempo



# A cada minuto

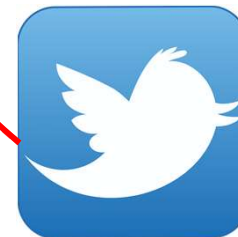
840 novos usuários em redes sociais

3 milhões de posts  
510 mil comentários  
293 mil atualizações de *status*  
136 mil novas fotos  
4 milhões de likes



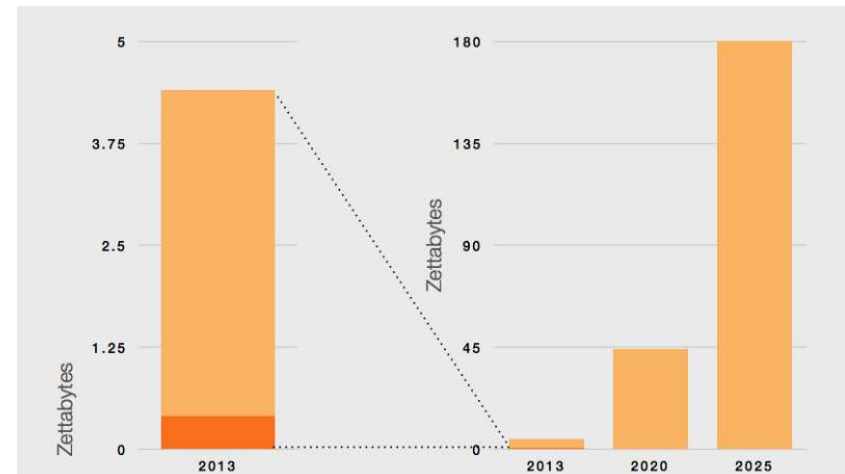
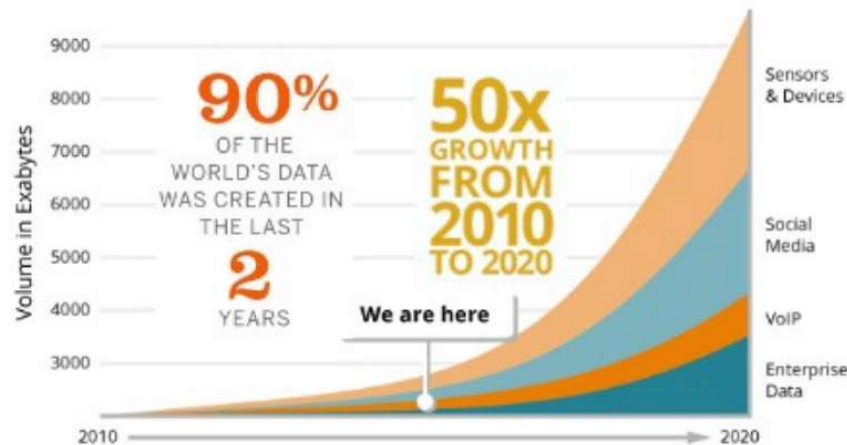
455 mil tweets

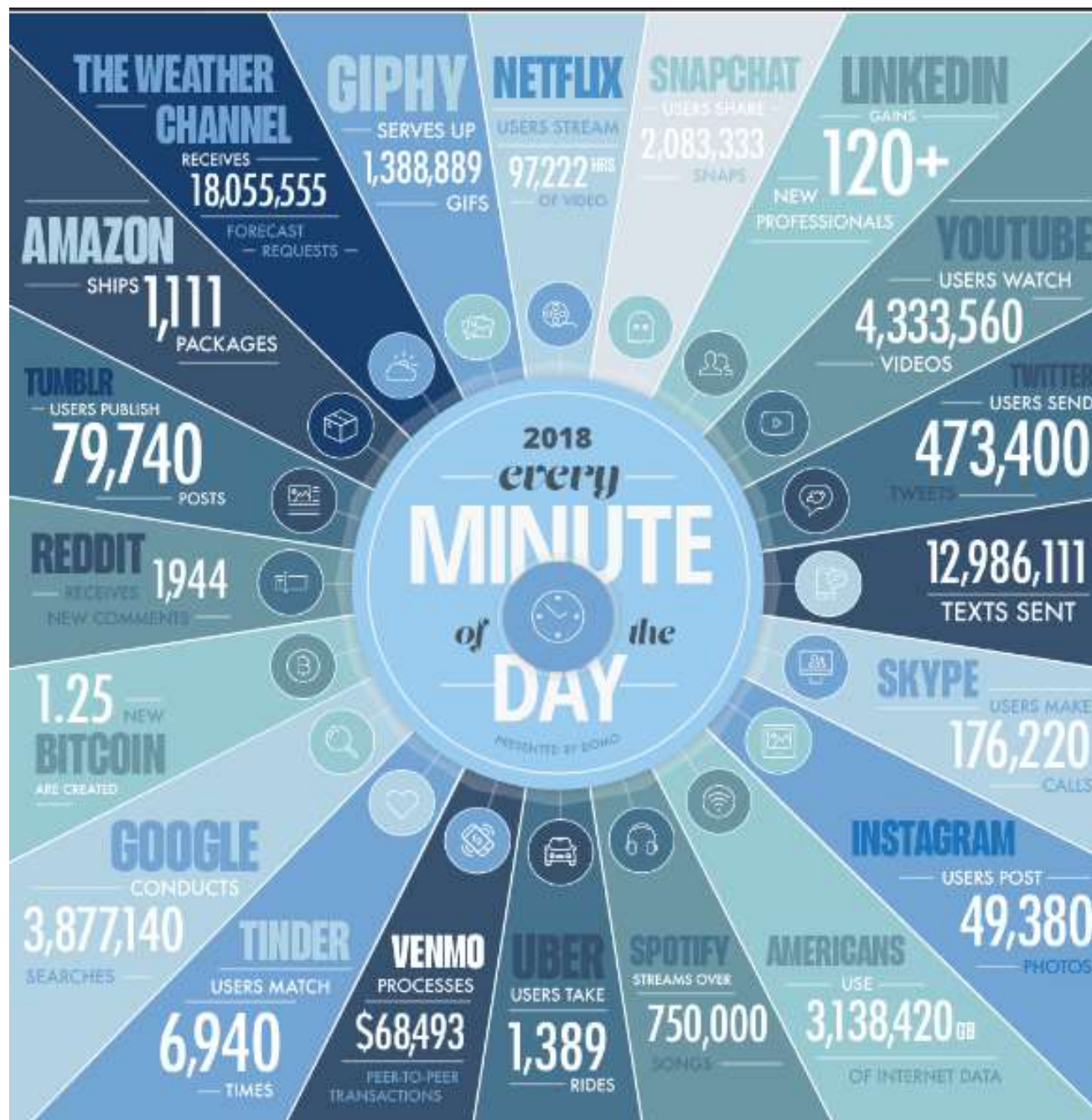
400 horas de vídeo são enviados  
4.146.600 vídeos são assistidos



# Volume

- 90% dos dados foram gerados nos últimos 2 anos
  - Se manteve assim por 30 anos
- Espera-se que em 2025, os dados produzidos em uma semana serão equivalentes aos dados produzidos pela humanidade até 2013
  - Smartphones
- Criamos muitos dados sem saber
  - Posts, tweets, fotos, vídeos, ...
  - Logs, sensores, ...











# Big Data

- *“...Um conjunto de dados tão grande e complexo que torna o seu processamento e armazenamento através de técnicas tradicionais impraticável...”*

*NIST*

- *“...Big Data é grande Volume, alta Velocidade, e alta Variedade de ativos de informação que exigem formas inovadoras de baixo custo de processamento de informação para melhor percepção e tomada de decisão...”*

*Gartner*

- *“... Big Data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it...”*

*Dan Ariely*

- *“... Quando o Excel trava abrindo o arquivo...”*

*Aluno de 3º período*

# Big Data

- *“...Um conjunto de dados tão grande e complexo que torna o seu processamento e armazenamento através de técnicas tradicionais impraticável...”*

*NIST*

## Desafios Técnicos

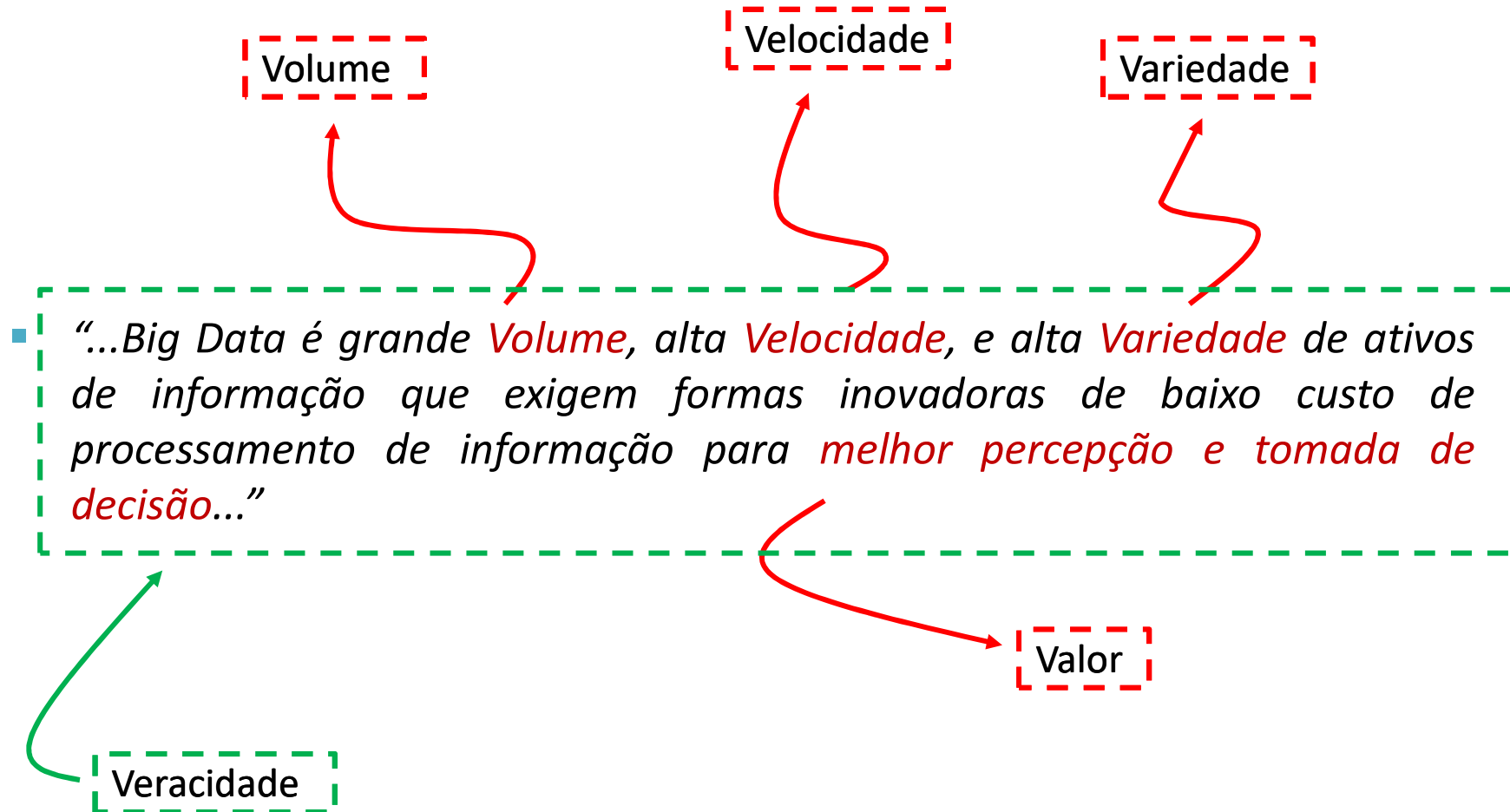
Armazenamento e Gerenciamento dos dados

Arquitetura descentralizada de processamento/armazenamento

Gargalos de processamento (performance)

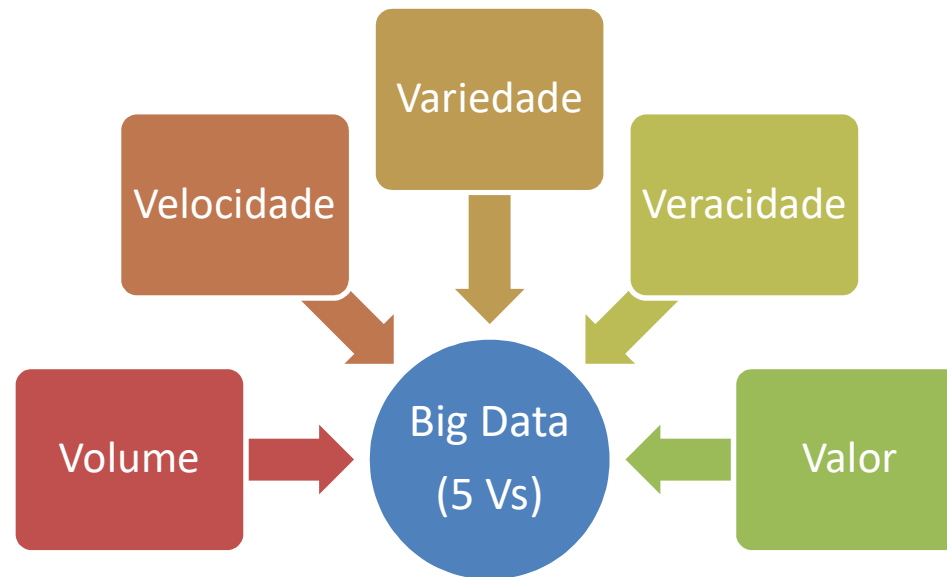
Requisitos de hardware

# Big Data



# Big Data

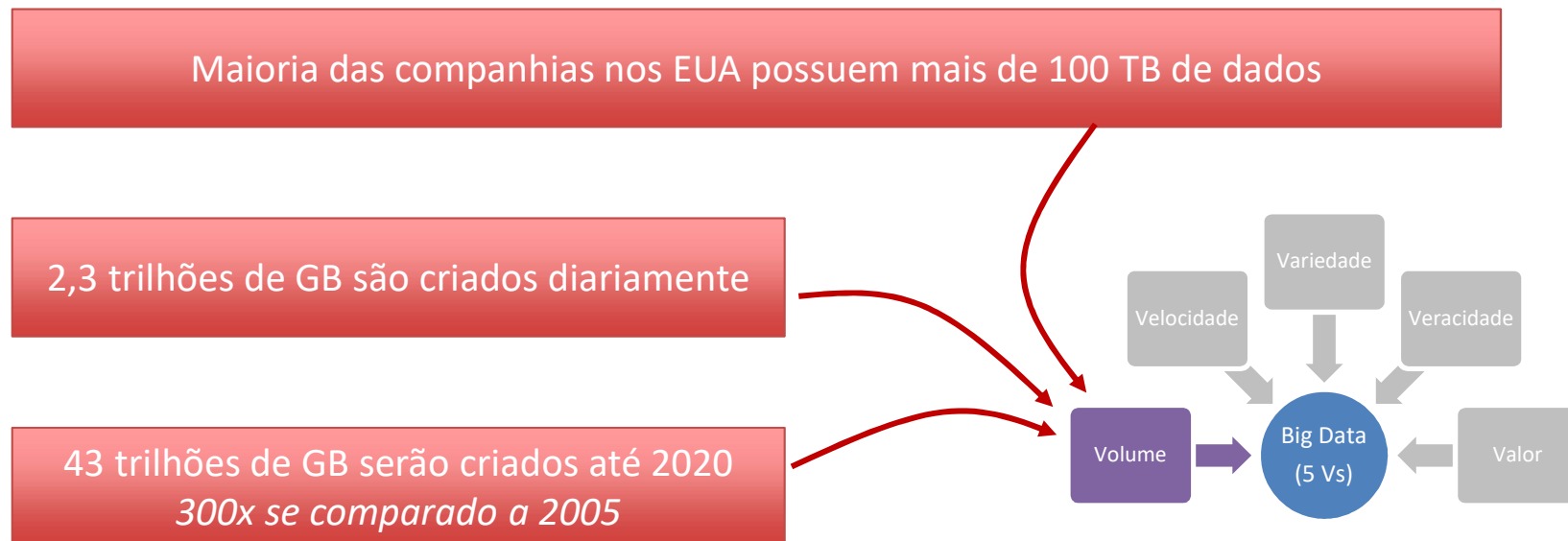
- “...Big Data é grande *Volume*, alta *Velocidade*, e alta *Variedade* de ativos de informação que exigem formas inovadoras de baixo custo de processamento de informação para *melhor percepção e tomada de decisão...*”



# Características

## ■ Volume

- Massa de dados a serem processados
- Empresas podem facilmente gerar GBs de dados diariamente
  - E.g. atividades em redes sociais
  - E.g. logs de servidores



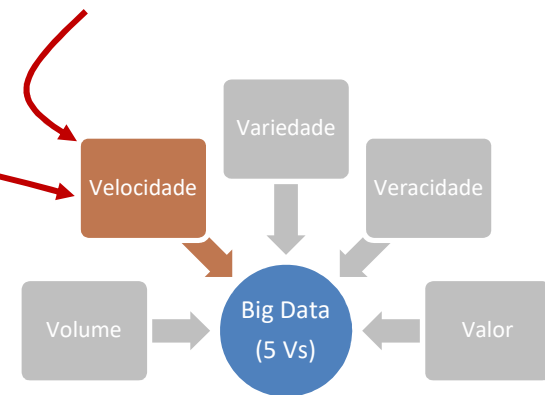
# Características

## ■ Velocidade

- Velocidade em que os novos dados são gerados e devem ser processados
- Tempo de processamento é dependente de aplicação
  - E.g. identificação de fraudes bancárias
  - E.g. identificação de ataques na rede
  - E.g. identificação de impacto de uma campanha publicitária

Wallmart processa mais de 1 milhão de transações de clientes por hora (2,5 petabytes)

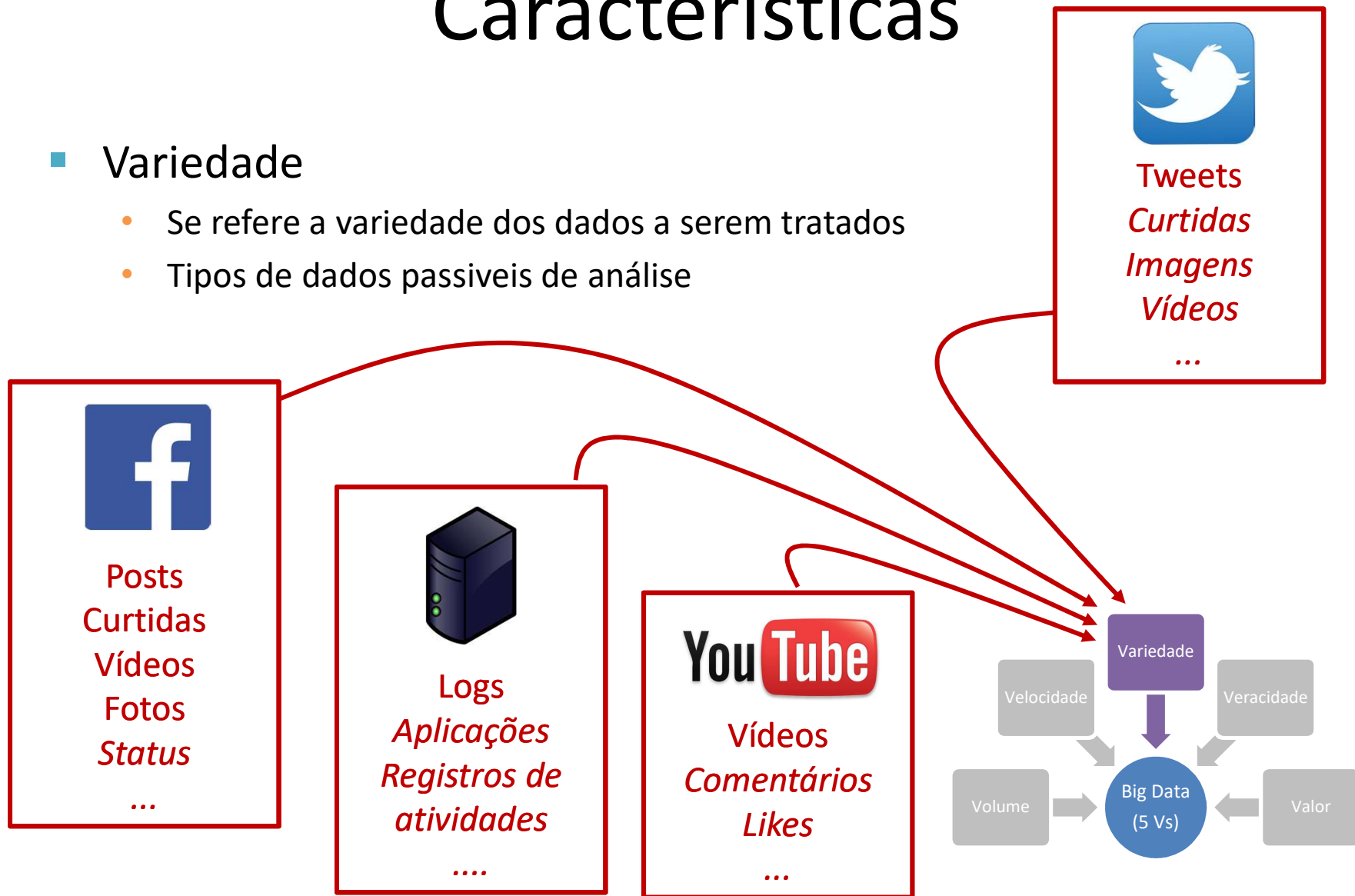
A cada minuto:  
455 mil tweets  
400 horas de vídeos enviados no Youtube  
3 milhões de posts no Facebook



# Características

## ■ Variedade

- Se refere a variedade dos dados a serem tratados
- Tipos de dados passíveis de análise



# Características

## ■ Veracidade

- Dados devem ser reais e condizente com a realidade
- 1 em cada 3 gestores não confiam nos dados que recebem
- **Apenas dados reais agregam valor!**
  - E.g. remoção de perfis falsos em redes sociais
  - E.g. análise de dados em uma janela de tempo condizente com as necessidades



**Remoção de  
perfis falsos**  
*Análise no  
contexto  
desejado*

...



**Remoção de  
perfis falsos**  
*Análise de  
ações de  
acordo com o  
contexto  
desejado*

...



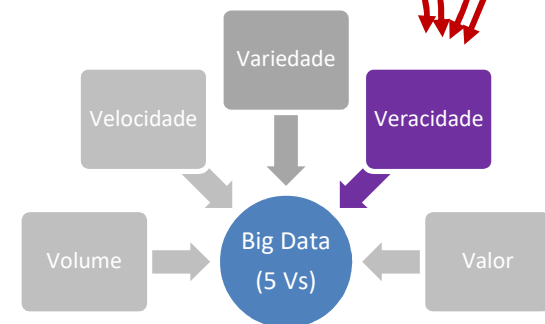
**Análise de  
registro da  
aplicação  
correta**

...



**Análise do  
vídeo de  
acordo com  
contexto**

...

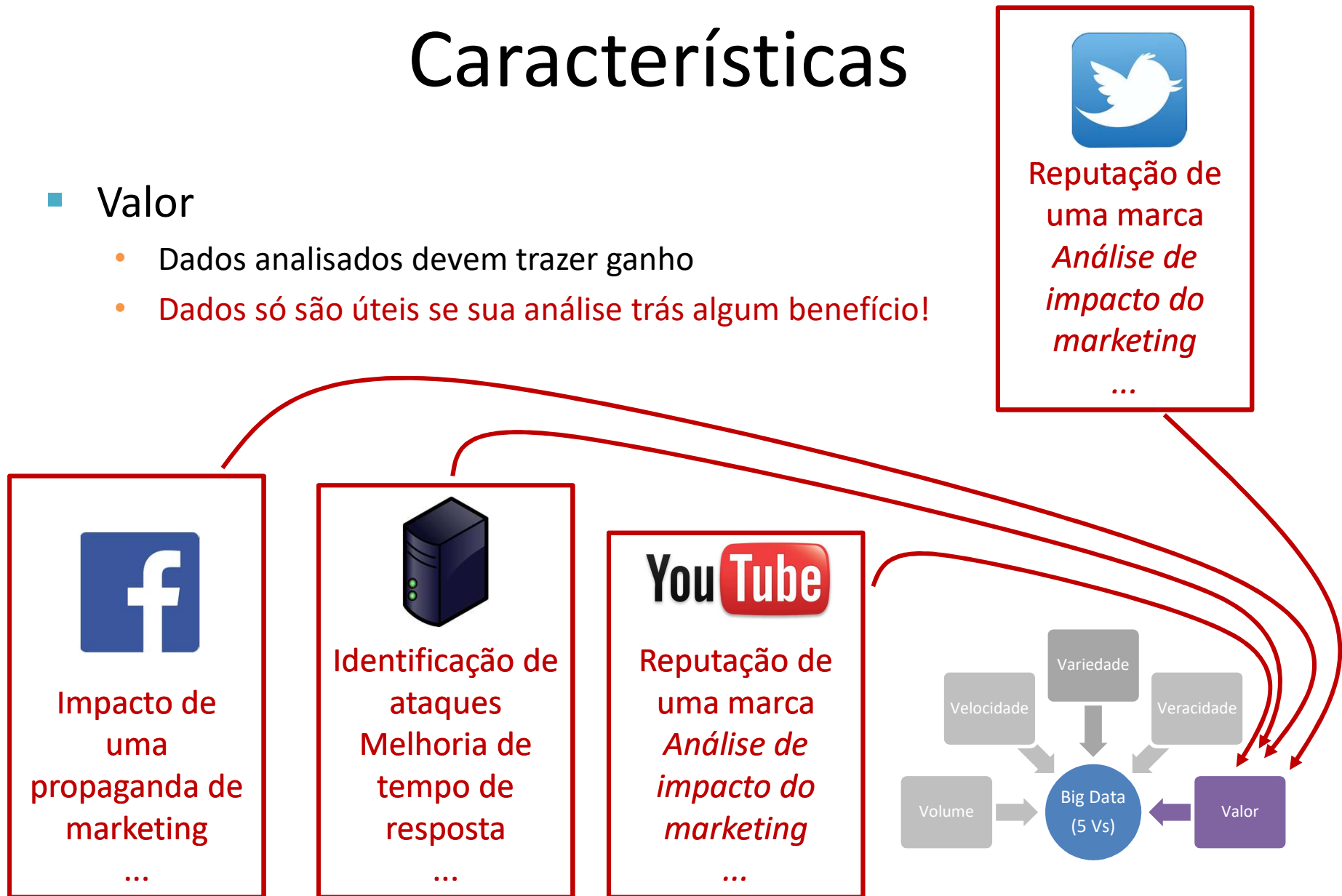




# Características

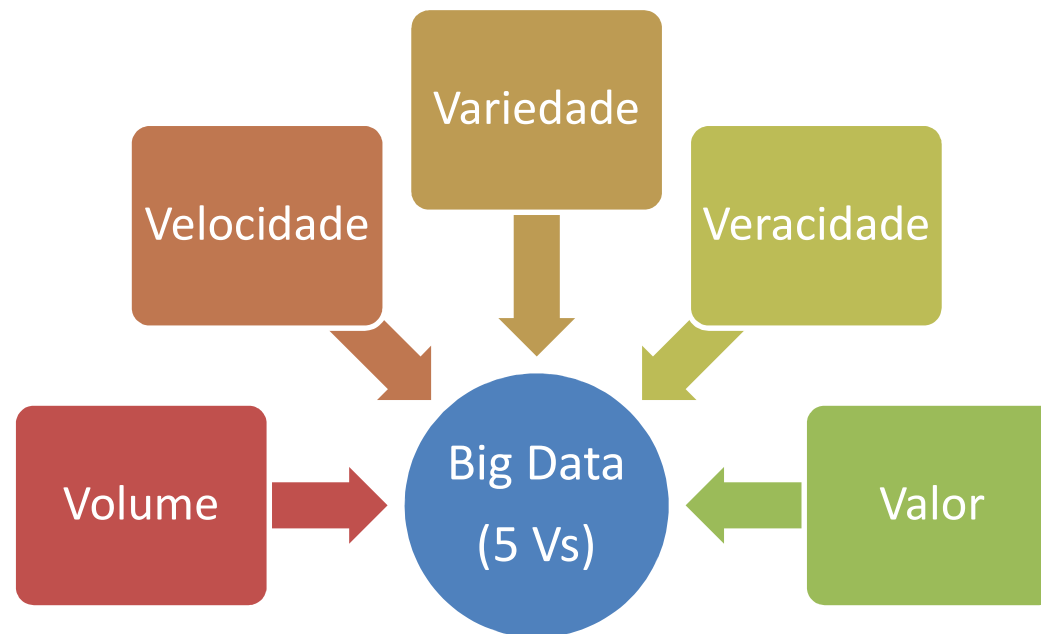
## ■ Valor

- Dados analisados devem trazer ganho
- Dados só são úteis se sua análise trás algum benefício!



# Em Resumo

- *Temos um cenário de Big Data quando*
  - *“...Um conjunto de dados tão grande e complexo que torna o seu processamento e armazenamento através de técnicas tradicionais impraticável...”*
- *O cenário apresenta os 5Vs*



# Casos de Uso

- Como podemos prever uma epidemia de gripe?

- Analisando taxa de consultas médicas relacionadas a gripe
- Determinando a relação entre a temperatura e a taxa de consultas médicas
- Estabelecendo períodos típicos de incidência da gripe



Abordagens custosas, lentas, imprecisas, com atraso, ...

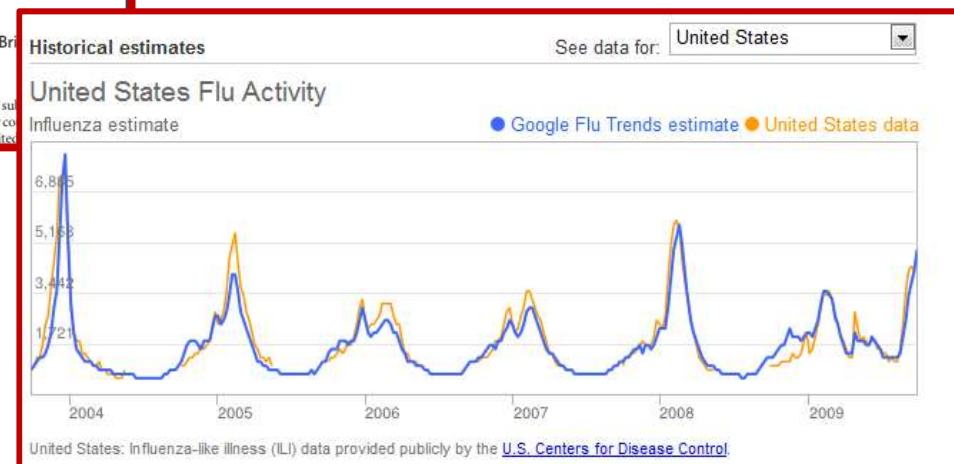
- Acessem

- <https://trends.google.com.br/trends>

- Como podemos prever uma epidemia de gripe?

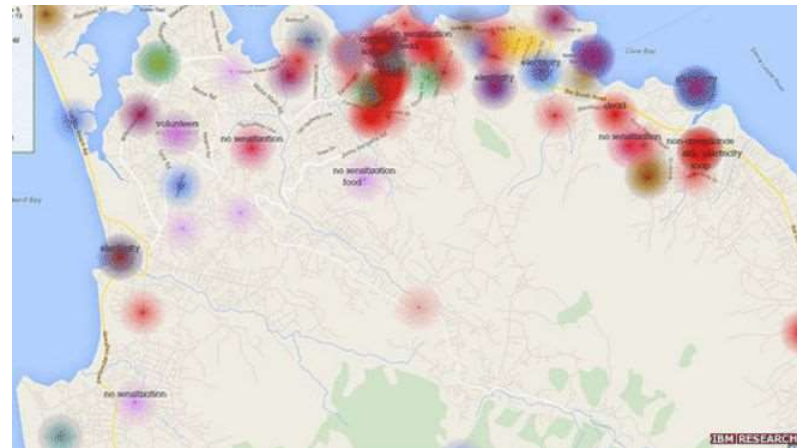
# Casos de Uso

- Como podemos prever uma epidemia de gripe?
  - <https://www.google.org/flutrends>
  - <https://flunearyou.org/#!/>



# Casos de Uso

- Detecção de casos de Ebola em *Sierra Leone*
  - Comunidades avisam ao governo possíveis casos de Ebola
    - Celular, Rádio, SMS, ...
  - IBM coordena o sistema para identificação de possíveis epidemias
    - Aprendizagem de máquina para *clusterizar* possíveis focos
    - Mapas de acordo com os focos identificados
  - Plataforma mundial para compartilhamento de informações sobre casos de Ebola
    - Dados repassados ao governo
  - Totalmente gratuito



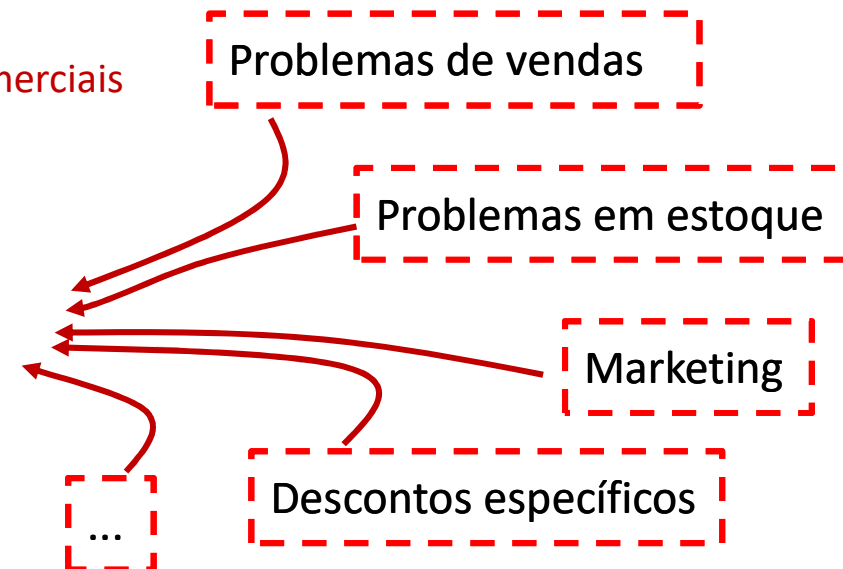
# Casos de Uso

## ■ Caso *Cambridge Analytica*

- 87 milhões de perfis do Facebook mapeados
- Exploração de “vulnerabilidade” nos termos de uso do Facebook
- Análise de comportamento dos usuários
- Informações obtidas foram utilizadas durante a campanha presidencial dos EUA
  - Propaganda política personalizada
  - Notícias falsas com foco específico

## ■ Caso *Wallmart*

- 200 bilhões de entradas de transações comerciais
  - Representa apenas as últimas semanas
- 200 fontes de dados
  - Redes sociais, dados econômicos, preços locais, ...
- 40 petabytes de dados
- Monitoramento de performance por loja



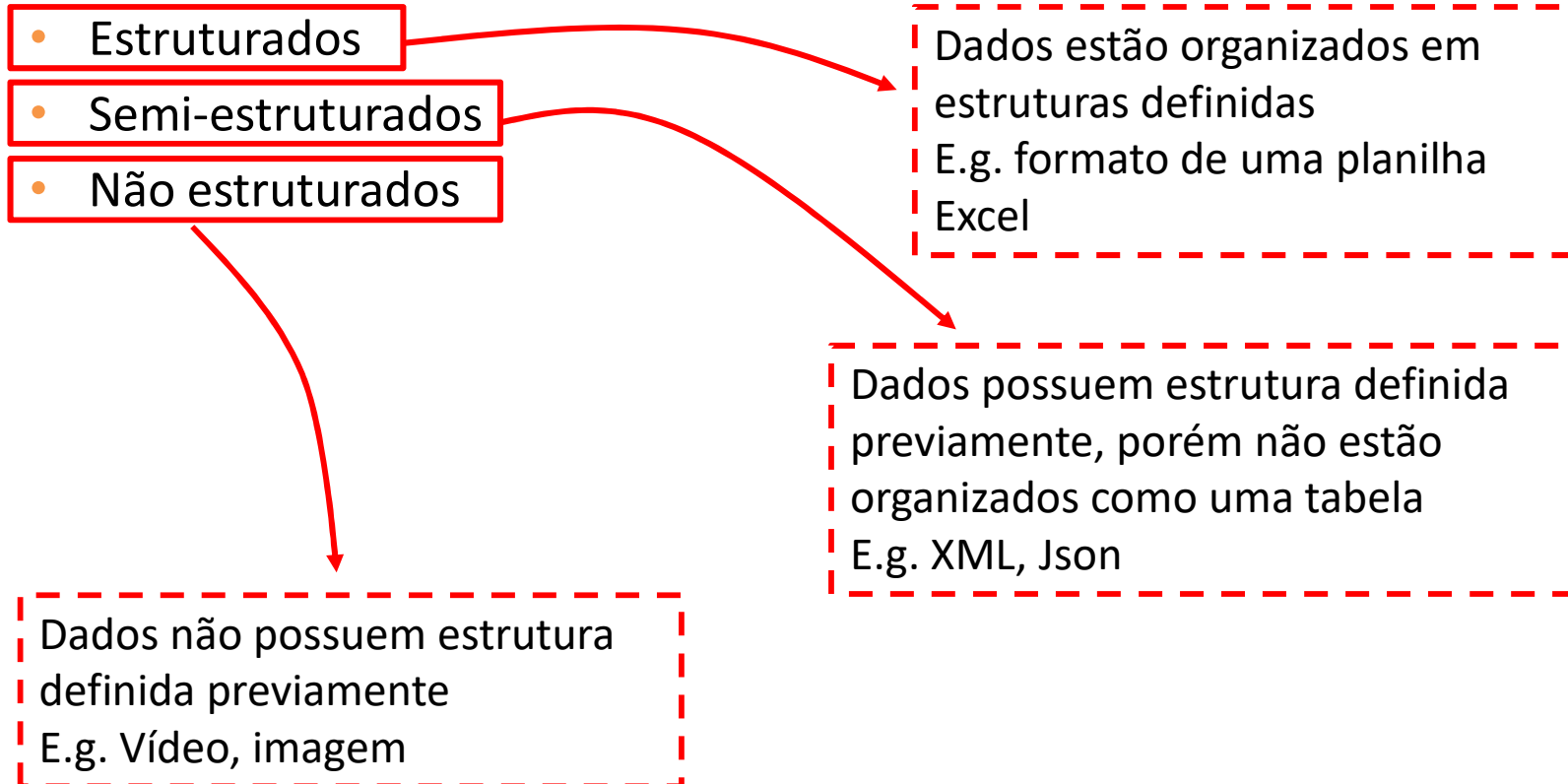
# Dados de Big Data

- Devido a variedade dos dados, cenários de Big Data dificilmente possuem dados previamente estruturados
  - Estima-se que 90% dos dados gerados não possuem uma estrutura previamente definida



# Dados de Big Data

- Dados de Big Data tipicamente são divididos em três tipos





# Dados de Big Data

## ■ Dados Estruturados

- Tipicamente encontrados em SGBDs
- Dados organizados em blocos semânticos
- Dados de um mesmo grupo possuem as mesmas características
- Dados são previamente definidos e tipados
- Dados possuem a mesma estrutura de representação
- Tipicamente representados por linhas e colunas



The screenshot displays a web interface for an inventory list. At the top, there's a header bar with the title 'Lista de estoque' and a search icon. Below the header, a table is shown with four columns: 'ID de estoque', 'Nome', 'Descrição', and 'Preço unitário'. The table contains 12 rows of data, each representing an inventory item. The rows are numbered 1 through 12 on the left side of the table. The first row (ID IN0001) has a value of R\$ 51.00. The second row (ID IN0002) has a value of R\$ 93.00. The third row (ID IN0003) has a value of R\$ 57.00. The fourth row (ID IN0004) has a value of R\$ 19.00. The fifth row (ID IN0005) has a value of R\$ 75.00. The sixth row (ID IN0006) has a value of R\$ 11.00. The seventh row (ID IN0007) has a value of R\$ 56.00. The eighth row (ID IN0008) has a value of R\$ 38.00. The ninth row (ID IN0009) has a value of R\$ 59.00. The tenth row (ID IN0010) has a value of R\$ 50.00. The eleventh row (ID IN0011) has a value of R\$ 59.00. The twelfth row (ID IN0012) has a value of R\$ 18.00.

ID de estoque	Nome	Descrição	Preço unitário
IN0001	Item 1	Descr 1	R\$ 51.00
IN0002	Item 2	Descr 2	R\$ 93.00
IN0003	Item 3	Descr 3	R\$ 57.00
IN0004	Item 4	Descr 4	R\$ 19.00
IN0005	Item 5	Descr 5	R\$ 75.00
IN0006	Item 6	Descr 6	R\$ 11.00
IN0007	Item 7	Descr 7	R\$ 56.00
IN0008	Item 8	Descr 8	R\$ 38.00
IN0009	Item 9	Descr 9	R\$ 59.00
IN0010	Item 10	Descr 10	R\$ 50.00
IN0011	Item 11	Descr 11	R\$ 59.00
IN0012	Item 12	Descr 12	R\$ 18.00

# Dados de Big Data

## ■ Dados Semi-Estruturados

- Não são estritamente tipados
- Não são completamente não estruturados
- Organização heterogênea
- Dificuldade na consulta
- Tipicamente possuem o esquema de representação presente de forma implícita ou explícita
- Auto-descritivo
- Requer pré-processamento

```
{
  "api": {
    "title": "Example API",
    "links": {
      "author": "mailto:api-admin@example.com",
      "describedBy": "https://example.com/api-docs/"
    }
  },
  "resources": {
    "tag:me@example.com,2016:widgets": {
      "href": "/widgets/"
    },
    "tag:me@example.com,2016:widget": {
      "hrefTemplate": "/widgets/{widget_id}",
      "hrefVars": {
        "widget_id": "https://example.org/param/widget"
      },
      "hints": {
        "allow": ["GET", "PUT", "DELETE", "PATCH"],
        "formats": {
          "application/json": {}
        },
        "acceptPatch": ["application/json-patch+json"],
        "acceptRanges": ["bytes"]
      }
    }
  }
}
```

# Dados de Big Data

## ■ Dados Semi-Estruturados principais características

- Definição posterior
  - Esquema definido de acordo com os dados
  - Necessita investigação da estrutura
- Estrutura não regular
  - Dados sem esquema padrão
  - Dados são definidos de maneira diferente
- Estrutura implícita
  - A estrutura tipicamente é definida de acordo com o conteúdo
- Estrutura parcial
  - Tipicamente apenas parte dos dados possuem estrutura

```
{
  "api": {
    "title": "Example API",
    "links": {
      "author": "mailto:api-admin@example.com",
      "describedBy": "https://example.com/api-docs/"
    }
  },
  "resources": {
    "tag:me@example.com,2016:widgets": {
      "href": "/widgets/"
    },
    "tag:me@example.com,2016:widget": {
      "hrefTemplate": "/widgets/{widget_id}",
      "hrefVars": {
        "widget_id": "https://example.org/param/widget"
      },
      "hints": {
        "allow": ["GET", "PUT", "DELETE", "PATCH"],
        "formats": {
          "application/json": {}
        },
        "acceptPatch": ["application/json-patch+json"],
        "acceptRanges": ["bytes"]
      }
    }
  }
}
```

# Dados de Big Data

- Dados Semi-Estruturados exemplos
  - Página web

```
<!doctype html>
<html lang="pt-BR" class="js" style="overflow: initial;">
<head>...</head>
<body class="home page-template page-template-pagebuilder page-template-pagebuilder-php page page-id-31392 logged-in
pucpr site-id-1"> ... $0
<!-- Google Tag Manager (noscript) -->
<noscript>...</noscript>
<!-- End Google Tag Manager (noscript) -->
<!-- wrapper -->
<div id="wrapper">...</div>
<!-- #wrapper -->
<script>
var blablu_banners_menu = JSON.parse('[]');
</script>
<link rel="stylesheet" id="slick-css-css" href="/wp-content/themes/pucpr/..wiki/assets/css/slick.css?ver=1.6.0"
type="text/css" media="all">
<link rel="stylesheet" id="jquery-ui-css" href="/wp-content/themes/pucpr/..wiki/assets/css/jquery-ui.min.css?
ver=1.0" type="text/css" media="all">
<link rel="stylesheet" id="dropdown-css" href="/wp-content/themes/pucpr/..wiki/assets/css/dropdown.min.css?
ver=2.2.10" type="text/css" media="all">
<link rel="stylesheet" id="transition-css" href="/wp-content/themes/pucpr/..wiki/assets/css/transition.min.css?
ver=2.2.10" type="text/css" media="all">
<link rel="stylesheet" id="fancybox-css-css" href="/wp-content/themes/pucpr/..wiki/assets/css/fancybox.min.css?
ver=1.0" type="text/css" media="all">
<!-- [if lt IE]
<link rel="stylesheet" id="ie-css" href="https://www.pucpr.br/wp-admin/css/ie.min.css?
ver=7393baa4e3f96bca30eb3dc6d8b36f76" type="text/css" media="all" />
<![endif]-->
<script type="text/javascript">...</script>
<script type="text/javascript" src="/wp-content/plugins/contact-form-7/includes/js/scripts.js?ver=5.0.3"></script>
<script type="text/javascript" src="/wp-content/themes/pucpr/..wiki/portal/assets/js/portal.class.js?ver=1.0.0">
</script>
<script type="text/javascript">...</script>
<script type="text/javascript" src="/wp-content/themes/pucpr/..wiki/pagebuilder/assets/js/pagebuilder.class.js?
ver=1.0.0"></script>
<script type="text/javascript" src="/wp-content/themes/pucpr/..wiki/agenda/assets/js/agenda.class.js?ver=1.0.0">
</script>
<script type="text/javascript" src="/wp-content/themes/pucpr/..wiki/assets/js/library.min.js?ver=1.0.0"></script>
<script type="text/javascript" src="/wp-content/themes/pucpr/assets/js/jquery.validate.min.js?ver=1.0.0"></script>
<script type="text/javascript" src="/wp-content/themes/pucpr/assets/js/functions_pucpr.min.js?ver=1.0.0"></script>
<script type="text/javascript" src="/wp-content/themes/pucpr/assets/js/banners_menu.js?ver=7393baa"></script>
<script type="text/javascript" src="https://www.pucpr.br/wp-includes/js/wp-embed.min.js?ver=7393baa"></script>
<script>...</script>
<div class="subir float">...</div>
<a data-fancybox style="display: none;" class="playPop" href="https://youtu.be/
BqHzbeHwJ3A&autoplay=1&rel=0&controls=0&showinfo=0"></a>
<!-- Hotjar Tracking Code for pucpr.br -->
<script>...</script>
```

# Dados de Big Data

- Dados Semi-Estruturados exemplos

- Arquivo JSON

```
{  
  id: "19292868552",  
  about: "Build, grow, and monetize your app with Facebook. https://developers.facebook.com/",  
  can_post: false,  
  category: "Product/service",  
  checkins: 1,  
  company_overview: "Visit https://developers.facebook.com for more information on how to build, grow, and monetize your app. If you have questions about using Facebook or need help with general inquiries, visit https://www.facebook.com/facebook or our Help Center at http://www.facebook.com/help. If you need to report bugs, appeal apps, or ask detailed technical questions, visit the following: Appeal Apps: https://developers.facebook.com/appeal Report Bugs: http://developers.facebook.com/bugs Technical Questions: http://facebook.stackoverflow.com/",  
  cover: {  
    cover_id: "10152004458663553",  
    offset_x: 0,  
    offset_y: 0,  
    source: "https://fbcdn-sphotos-c-a.akamaihd.net/hphotos-ak-xpal/v/t1.0-9/q71/s720x720/1466030_10152004458663553_1984809612_n.jpg?oh=33d8ed246c478be79360f3b08db40726&oe=542BBCC6&_gda__=1410722624_fd47070e6f8f875abe92fb42ff254753"  
  },  
  has_added_app: false,  
  is_community_page: false,  
  is_published: true,  
  likes: 3100156,  
  link: "https://www.facebook.com/FacebookDevelopers",  
  name: "Facebook Developers",  
  parking: {  
    lot: 0,  
    street: 0,  
    valet: 0  
  },  
  talking_about_count: 11744,  
  username: "FacebookDevelopers",  
  website: "http://developers.facebook.com",  
  were_here_count: 0  
}
```

[+ - View source](#)



# Dados de Big Data

- Dados Semi-Estruturados exemplos
  - Logs de servidor

```
1. Python
38.113.234.181 - - [20/May/2010:11:27:36 +0100] "GET /post/274953/ HTTP/1.1" 200 404 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
67.195.114.50 - - [20/May/2010:11:31:16 +0100] "GET /sources/14/ HTTP/1.0" 200 14262 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
61.135.216.104 - - [20/May/2010:11:32:56 +0100] "GET /feeds/latest/ HTTP/1.1" 200 48364 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
125.22.2.42 - - [20/May/2010:11:33:00 +0100] "GET /feeds/latest/ HTTP/1.1" 200 48364 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
77.88.26.26 - - [20/May/2010:11:33:50 +0100] "GET /posts/2234/ HTTP/1.1" 200 33674 "-" "Yandex/1.01.001 (compatible; Win16; I)"
24.62.152.212 - - [20/May/2010:11:34:07 +0100] "GET /media/style.css HTTP/1.1" 200 28479 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6.3; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
24.62.152.212 - - [20/May/2010:11:34:07 +0100] "GET /media/exmpl.png HTTP/1.1" 200 28479 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6.3; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
24.62.152.212 - - [20/May/2010:11:34:05 +0100] "GET / HTTP/1.1" 200 110836 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6.3; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
24.62.152.212 - - [20/May/2010:11:34:08 +0100] "GET /media/img/m-inact.gif HTTP/1.1" 200 2571 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6.3; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
24.62.152.212 - - [20/May/2010:11:34:08 +0100] "GET /media/img/side-container.gif HTTP/1.1" 200 1415 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6.3; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
24.62.152.212 - - [20/May/2010:11:34:08 +0100] "GET /media/img/m-act.gif HTTP/1.1" 200 143 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6.3; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
166.205.4.62 - - [20/May/2010:10:39:16 +0000] "GET /media/style.css HTTP/1.1" 304 - "http://example.com/" "Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us; AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/78367 Safari/531.21.10"
166.205.4.62 - - [20/May/2010:10:39:16 +0000] "GET /media/exmpl.png HTTP/1.1" 304 - "http://example.com/" "Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us; AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/78367 Safari/531.21.10"
166.205.4.62 - - [20/May/2010:10:39:17 +0000] "GET / HTTP/1.1" 200 26130 "-" "Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us; AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/78367 Safari/531.21.10"
166.205.4.62 - - [20/May/2010:10:39:17 +0000] "GET /media/img/m-act.gif HTTP/1.1" 304 - "http://example.com/" "Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us; AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/78367 Safari/531.21.10"
166.205.4.62 - - [20/May/2010:10:39:18 +0000] "GET /media/img/m-inact.gif HTTP/1.1" 304 - "http://example.com/" "Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us; AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/78367 Safari/531.21.10"
166.205.4.62 - - [20/May/2010:10:39:20 +0000] "GET /media/img/side-container.gif HTTP/1.1" 304 - "http://example.com/" "Mozilla/5.0 (iPad; U; CPU OS 3_2 like Mac OS X; en-us; AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/78367 Safari/531.21.10"
67.195.114.50 - - [20/May/2010:11:40:09 +0100] "GET /post/249681/ HTTP/1.0" 404 15 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
67.195.114.50 - - [20/May/2010:11:40:15 +0100] "GET /post/256414/ HTTP/1.0" 404 15 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
220.181.7.76 - - [20/May/2010:11:42:35 +0100] "GET / HTTP/1.1" 200 26130 "-" "Baiduspider-(http://www.baidu.com/search/spider.htm)"
67.195.114.50 - - [20/May/2010:11:42:58 +0100] "GET /post/274910/ HTTP/1.0" 404 15 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
68.59.242.134 - - [20/May/2010:10:43:14 +0000] "GET /media/style.css HTTP/1.1" 304 - "http://example.com/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:10:43:14 +0000] "GET / HTTP/1.1" 200 26130 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:10:43:16 +0000] "GET /media/img/favicon.ico HTTP/1.1" 200 1406 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:10:44:14 +0000] "GET /media/style.css HTTP/1.1" 304 - "http://example.com/posts/2/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:10:44:12 +0100] "GET /posts/2/ HTTP/1.1" 200 34408 "http://example.com/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
67.195.114.50 - - [20/May/2010:11:45:37 +0100] "GET /post/259342/ HTTP/1.0" 404 15 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
209.85.228.82 - - [20/May/2010:11:46:18 +0100] "GET /feeds/latest/ HTTP/1.1" 200 48364 "-" "FeedBurner/1.0 (http://www.FeedBurner.com)"
209.85.228.82 - - [20/May/2010:11:46:23 +0100] "GET /feeds/latest/ HTTP/1.1" 200 48364 "-" "FeedBurner/1.0 (http://www.FeedBurner.com)"
72.14.199.102 - - [20/May/2010:11:46:33 +0100] "GET /feeds/latest/ HTTP/1.1" 200 48364 "-" "FeedFetcher-Google; (http://www.google.com/feedfetcher.html; 9 subscribers; feed-id=18260778798542229819)"
196.203.53.144 - - [20/May/2010:11:46:37 +0100] "GET /feeds/latest/ HTTP/1.0" 200 48364 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.0; fr; rv:1.9.2.3) Gecko/20100401 Firefox/3.6.3"
68.59.242.134 - - [20/May/2010:10:46:43 +0000] "GET /media/style.css HTTP/1.1" 304 - "http://example.com/posts/3/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:11:46:41 +0100] "GET /posts/3/ HTTP/1.1" 200 25865 "http://example.com/posts/2/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:11:48:13 +0100] "GET /media/style.css HTTP/1.1" 304 - "http://example.com/posts/4/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:11:48:12 +0100] "GET /posts/4/ HTTP/1.1" 200 25930 "http://example.com/posts/3/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
66.249.65.42 - - [20/May/2010:11:48:14 +0100] "GET /posts/4/ HTTP/1.1" 200 25930 "-" "Mediapartners-Google"
66.249.65.40 - - [20/May/2010:11:48:37 +0100] "GET /post/274703/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
66.249.65.40 - - [20/May/2010:11:49:13 +0100] "GET /post/274704/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
68.59.242.134 - - [20/May/2010:10:49:54 +0000] "GET /media/style.css HTTP/1.1" 304 - "http://example.com/posts/5/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
68.59.242.134 - - [20/May/2010:11:49:52 +0100] "GET /posts/5/ HTTP/1.1" 200 29611 "http://example.com/posts/4/" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2)"
66.249.65.42 - - [20/May/2010:11:49:54 +0100] "GET /posts/5/ HTTP/1.1" 200 29611 "-" "Mediapartners-Google"
66.249.65.40 - - [20/May/2010:11:50:18 +0100] "GET /post/274687/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
66.249.65.40 - - [20/May/2010:11:51:23 +0100] "GET /post/274716/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
66.249.65.40 - - [20/May/2010:11:52:29 +0100] "GET /post/274712/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
75.57.176.28 - - [20/May/2010:11:52:42 +0100] "GET /media/style.css HTTP/1.1" 200 4847 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5.8; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
75.57.176.28 - - [20/May/2010:11:52:42 +0100] "GET / HTTP/1.1" 200 26130 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5.8; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
75.57.176.28 - - [20/May/2010:11:52:43 +0100] "GET /media/img/m-inact.gif HTTP/1.1" 200 2571 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5.8; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
75.57.176.28 - - [20/May/2010:11:52:43 +0100] "GET /media/img/side-container.gif HTTP/1.1" 200 1415 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5.8; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
75.57.176.28 - - [20/May/2010:10:52:42 +0000] "GET /media/exmpl.png HTTP/1.1" 200 28479 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5.8; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
75.57.176.28 - - [20/May/2010:11:52:43 +0100] "GET /media/img/m-act.gif HTTP/1.1" 200 143 "http://www.example.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5.8; en-us; AppleWebKit/531.22.7 (KHTML, like Gecko) Version/4.0.5 Safari/531.22.7"
66.249.65.40 - - [20/May/2010:11:53:34 +0100] "GET /post/274702/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
67.195.114.50 - - [20/May/2010:11:54:36 +0100] "GET /post/256204/ HTTP/1.0" 404 15 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.com/help/us/ysrch/slurp)"
66.249.65.40 - - [20/May/2010:11:54:39 +0100] "GET /post/274873/ HTTP/1.1" 404 15 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)"
```

# Dados de Big Data

- Dados Estruturados x Semi-estruturados

Dados Estruturados	Dados Semi-estruturados
Esquema pré-definido	Não há necessariamente um esquema
Estrutura regular	Estrutura não regular
Estrutura independente do conteúdo	Estrutura dependente do conteúdo
Estrutura reduzida	Estrutura extensa (depende do conteúdo)
Fracamente evolutiva (constante)	Fortemente evolutiva (muda-se a estrutura constantemente)
Esquema definido e restrições de mudança	Estrutura descritiva de acordo com conteúdo
Distinção entre estrutura e dados	Distinção entre estrutura e dados não é clara

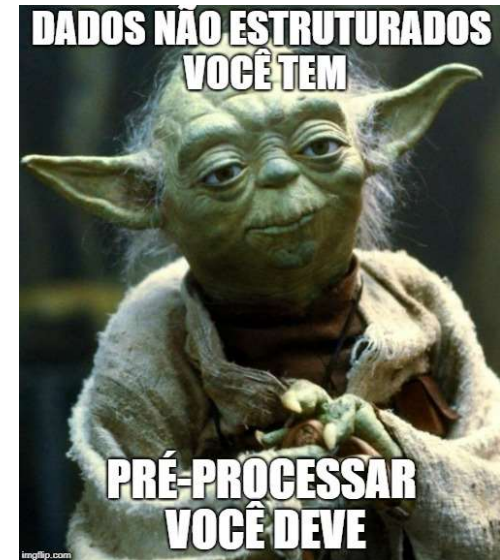
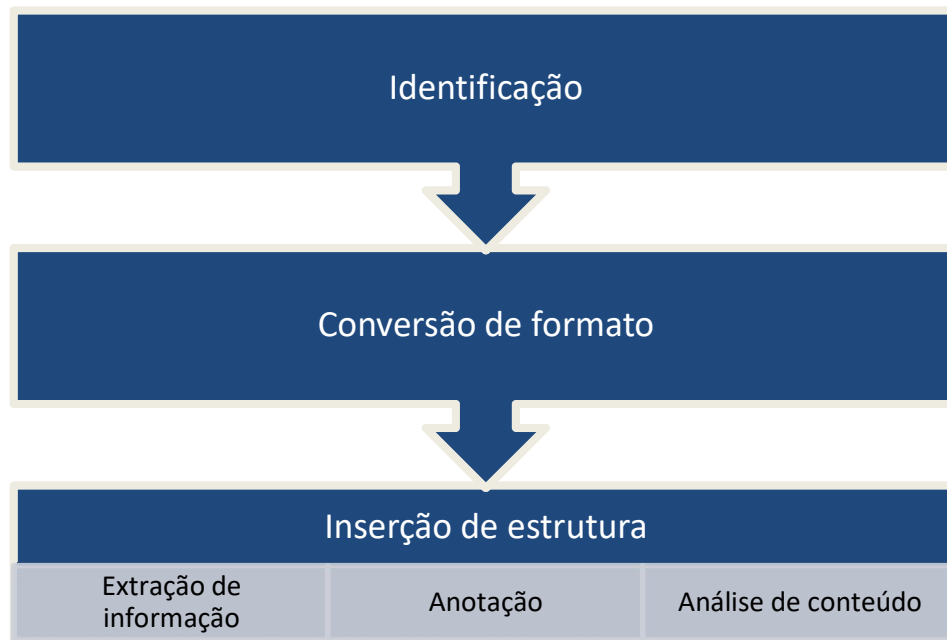
# Dados de Big Data

- Dados não estruturados
  - Dados que não possuem uma estrutura definida
  - Estruturas também não são descritas implicitamente
  - Representam a grande maioria dos dados atualmente
  - Não são armazenados em um SGBD tradicional
  - Tipos de dados
    - Textos (e-mail, blog, tweet, ...)
    - Dashboards
    - Imagens
    - Vídeos
    - ...



# Dados de Big Data

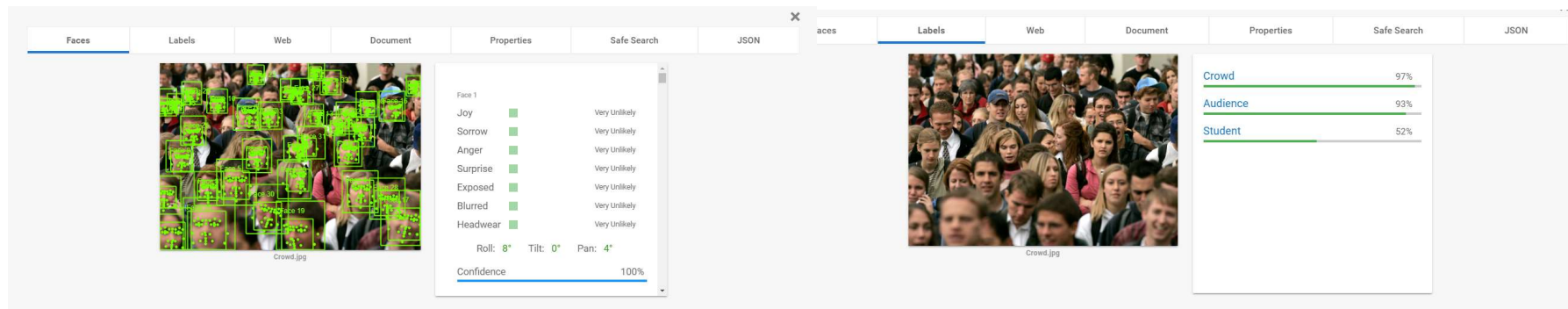
- Como podemos tornar dados não estruturados em dados “processáveis”?
  - Dados não estruturados tipicamente necessitam de pré-processamento



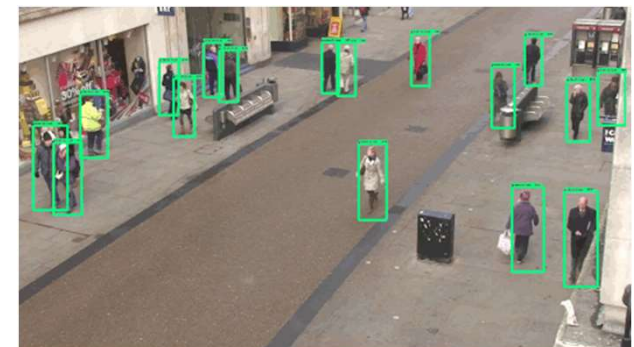
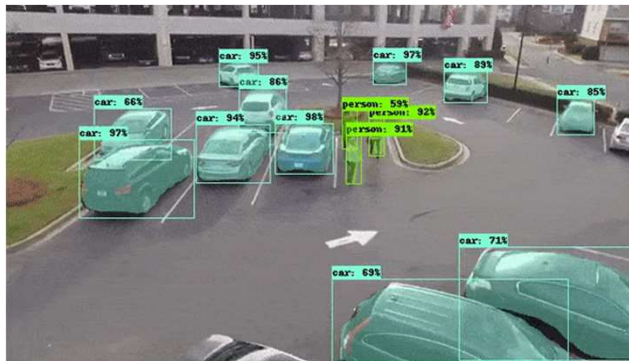
# Dados de Big Data

- Como podemos tornar dados não estruturados em dados “processáveis”?

- Exemplo: <https://cloud.google.com/vision/>

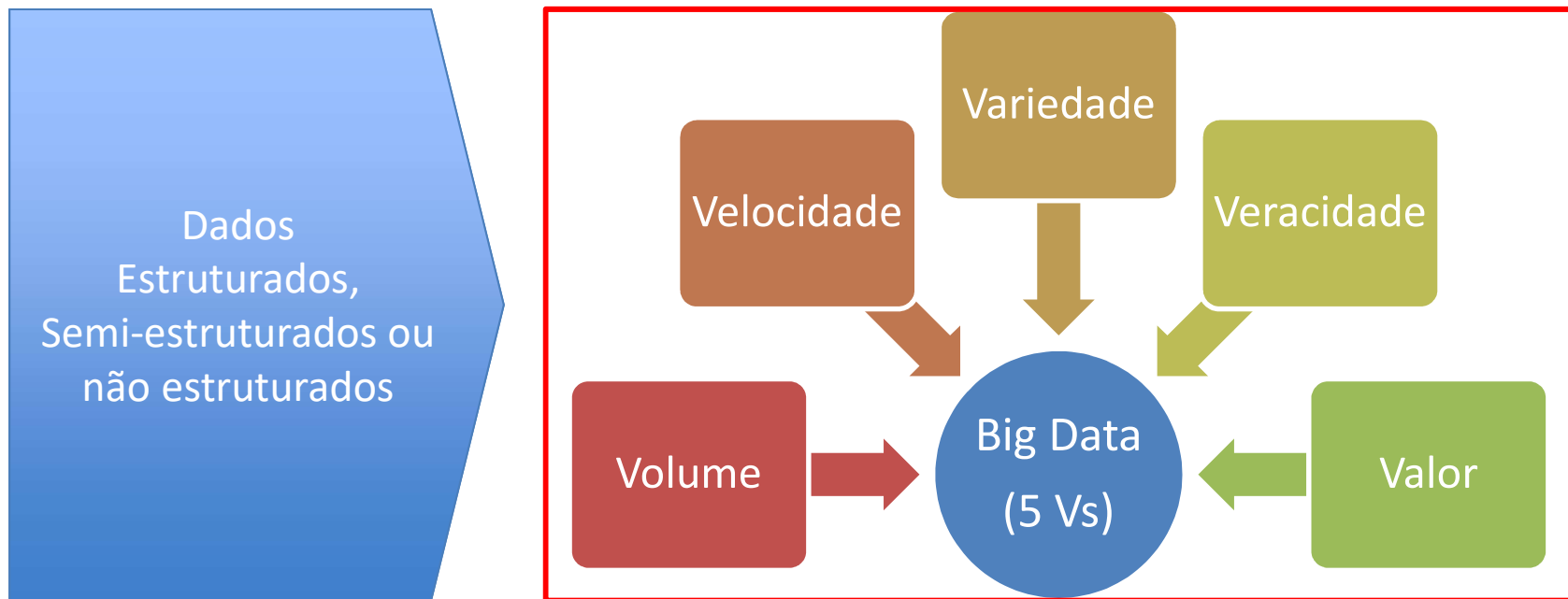


- Em vídeo (TensorFlow)



# Em Resumo

- Temos um cenário de Big Data quando
  - “...Um conjunto de dados tão grande e complexo que torna o seu processamento e armazenamento através de técnicas tradicionais impraticável...”
- O cenário apresenta os 5Vs



# Agenda

- Big Data

- **Hadoop**

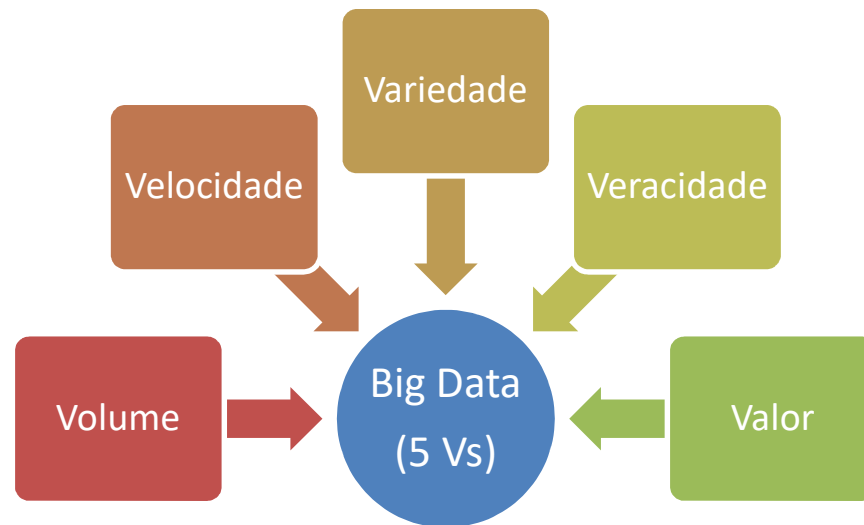
- HDFS
- MapReduce
- Apache Spark



kahoot.it

# Infraestrutura

- Como processar dados com as características de Big Data?



- As características dos dados de Big Data implicam em novos desafios de infraestrutura

# Infraestrutura

- Torna-se necessário uma infraestrutura distribuída de processamento e armazenamento
- Principais Desafios
  - Processamento lento, falta de escalabilidade
  - Busca em disco para cada leitura/escrita
  - Velocidade de leitura/escrita em disco se torna um gargalo
  - Armazenamento e processamento de grandes massas de dados



IDE – 75MB/s



SATA –  
300MB/s



SSD – 800MB/s

- Análise, processamento, agregação, atraso de processamento, ...

# Infraestrutura

- Torna-se necessário uma infraestrutura distribuída de processamento e armazenamento
- Principais Desafios
  - Falta de confiabilidade
    - 1000 máquinas processando/armazenando os dados, 1000 máquinas podem falhar
    - Restauração de dados
    - Restauração de nós
  - Escalabilidade
  - Backup
  - Custo
  - Facilidade de uso
  - Facilidade de processamento
  - Processamento distribuído
  - Armazenamento distribuído

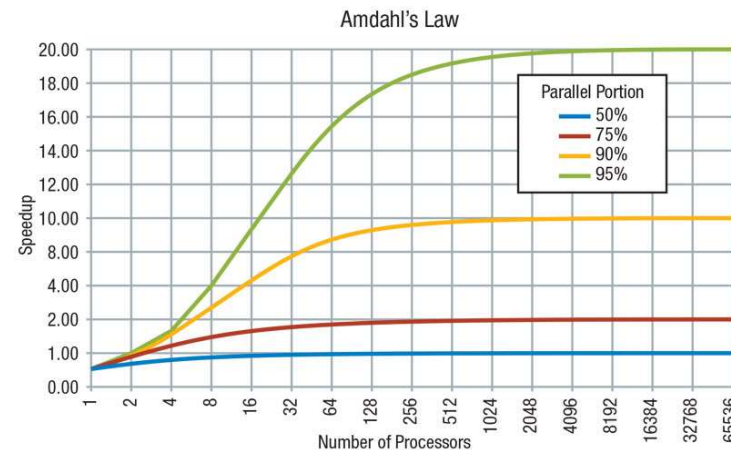
# Infraestrutura

- Processamento de dados em paralelo?



- Paralelismo não é tão simples!

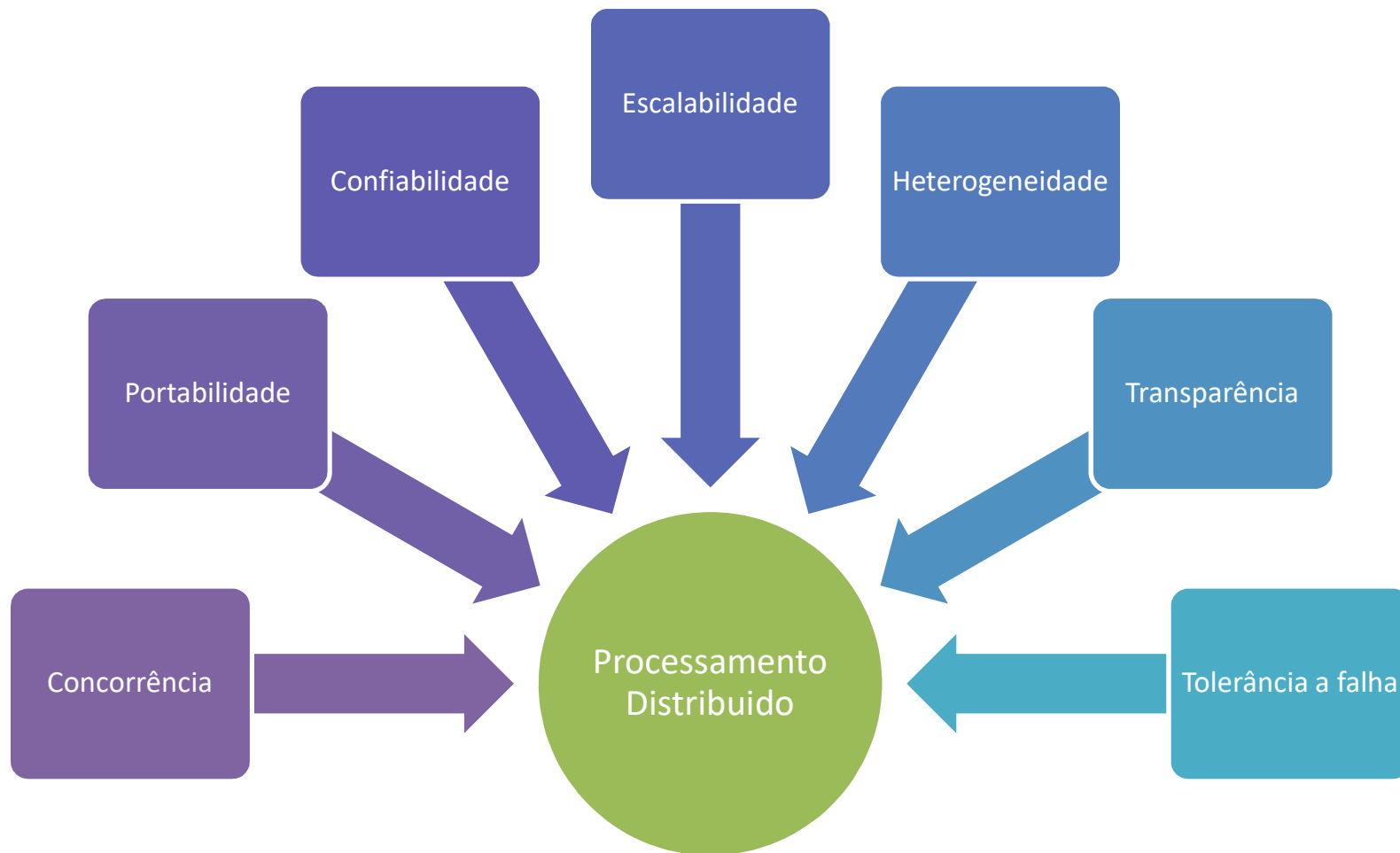
- Coordenação
- *Deadlock*
- Sincronização
- Capacidade de rede limitada
- Divisão e agregação
- Disponibilidade





# Infraestrutura

- Processamento de dados através de computação distribuída?





- Apache Hadoop é um framework que permite o processamento distribuído de grandes massas de dados, utilizando um *cluster* convencional através de um modelo simples de programação.
- Foi desenvolvido para permitir **processamento e armazenamento escalável** utilizando milhares de máquinas
- Hadoop torna o uso de uma **infraestrutura distribuída** transparente ao usuário

Eficiente

Confiável


“fácil” uso

Código aberto

Mantido como  
um projeto  
Apache

Suportado por  
grandes  
companhias



- Sistema de arquivo tolerante a falha
    - Hadoop Distributed File System (HDFS)
    - Inspirado no *Google File System*
  - Se inspira no modelo baseado em localidade
    - Leva o processamento aos dados
    - Diminui uso de rede
    - Diminui acesso a disco
  - Escalabilidade
    - Programa executado é o mesmo para 1, 10, 100, 1000, ... nós
    - Performance escalável
  - Modelo de computação MapReduce
- Técnica precursora
- 



## ■ Princípios do Hadoop

- Sistema deve se gerenciar e se recuperar

Falhas devem ser tratadas transparentemente ao usuário

Tarefas executadas de acordo com a performance dos nós

- Performance deve ser linearmente escalável

Proporcional a quantidade de nós utilizados

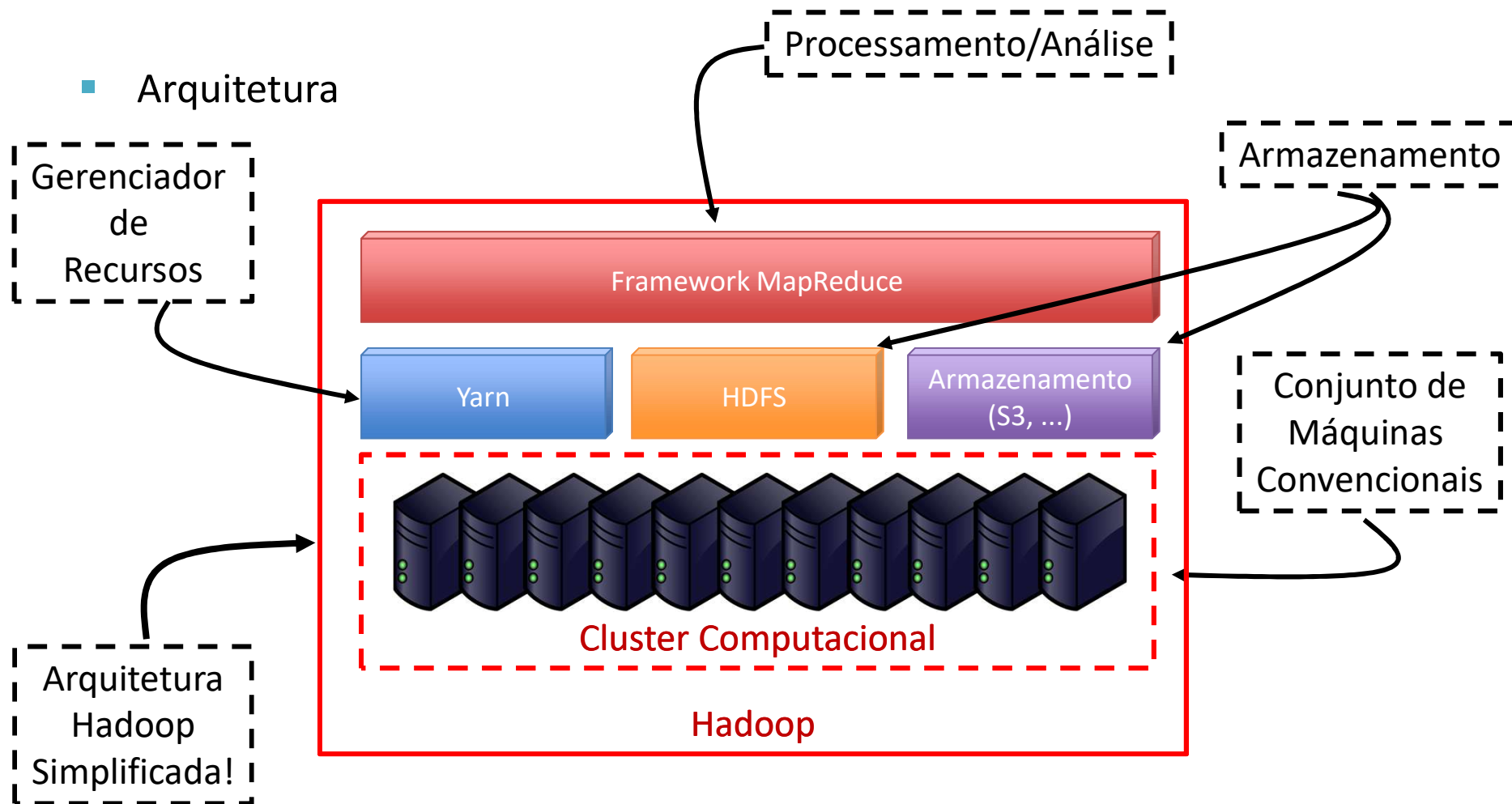
- Processamento deve ser próximo aos dados

Princípio da localidade, redução da latência e do uso da rede

- Simples, modular e extensível

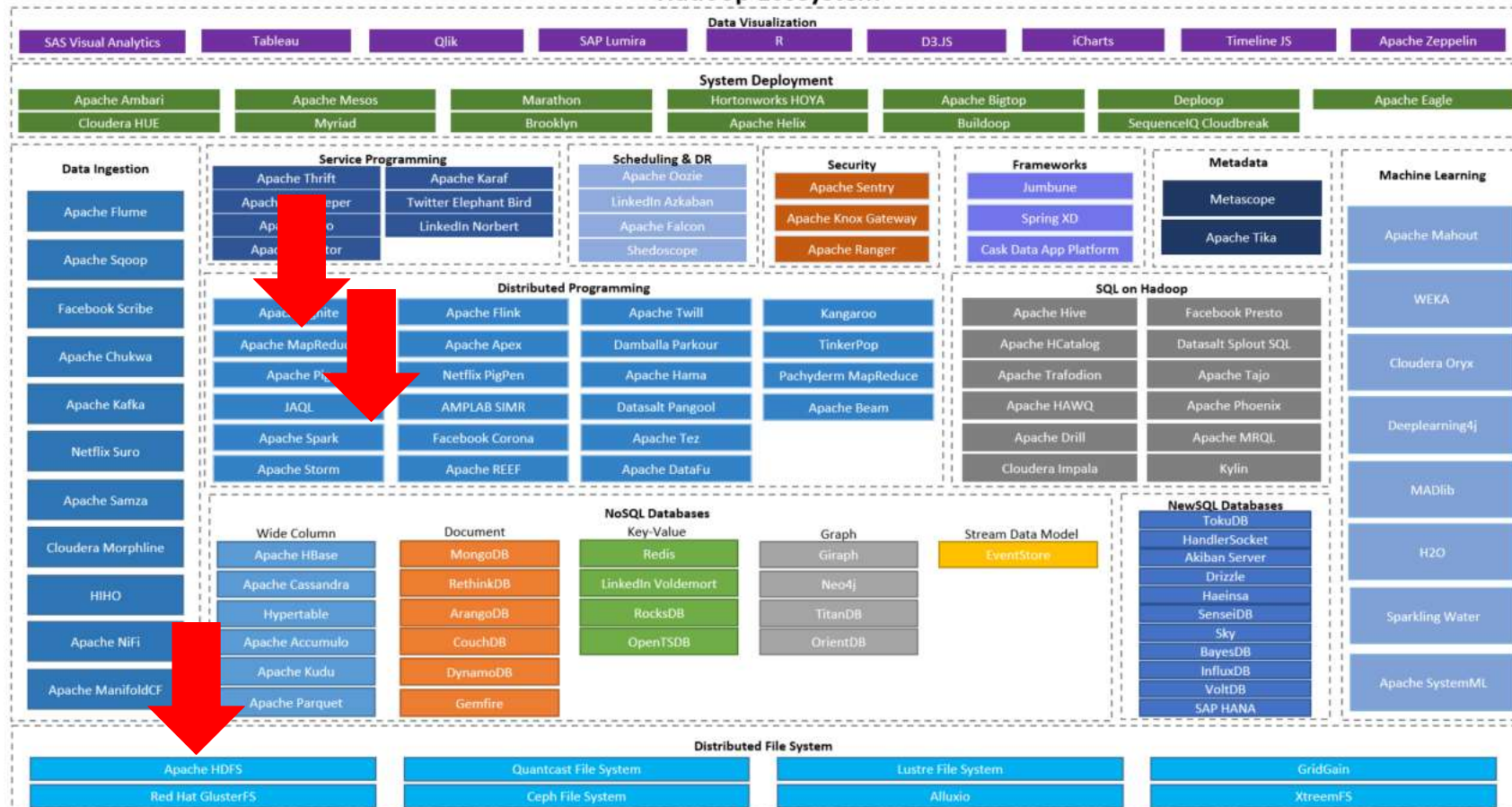


## ■ Arquitetura





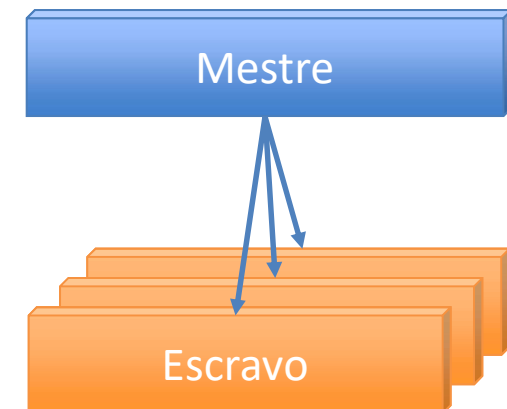
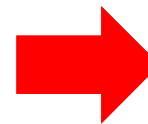
## Hadoop Ecosystem





- Hadoop possui uma arquitetura Mestre-Escravo
  - Um nó mestre gerencia um conjunto de nós escravos
  - Nó mestre é responsável pelo gerenciamento
  - Nós escravos efetuam o armazenamento/processamento

Processamento paralelo  
Escalável  
Execução de tarefas de maneira independente  
Gerenciamento de replicação  
Tolerância a falta  
Consistência de dados  
...





## ■ Funcionalidades

### Armazenamento de dados

- Estruturado, semi estruturado ou não estruturado

### Capacidade de armazenamento

- Escala linearmente

### Tolerancia a falha

- Confiabilidade

### Redução de custos e complexidade

- Utiliza computadores tradicionais

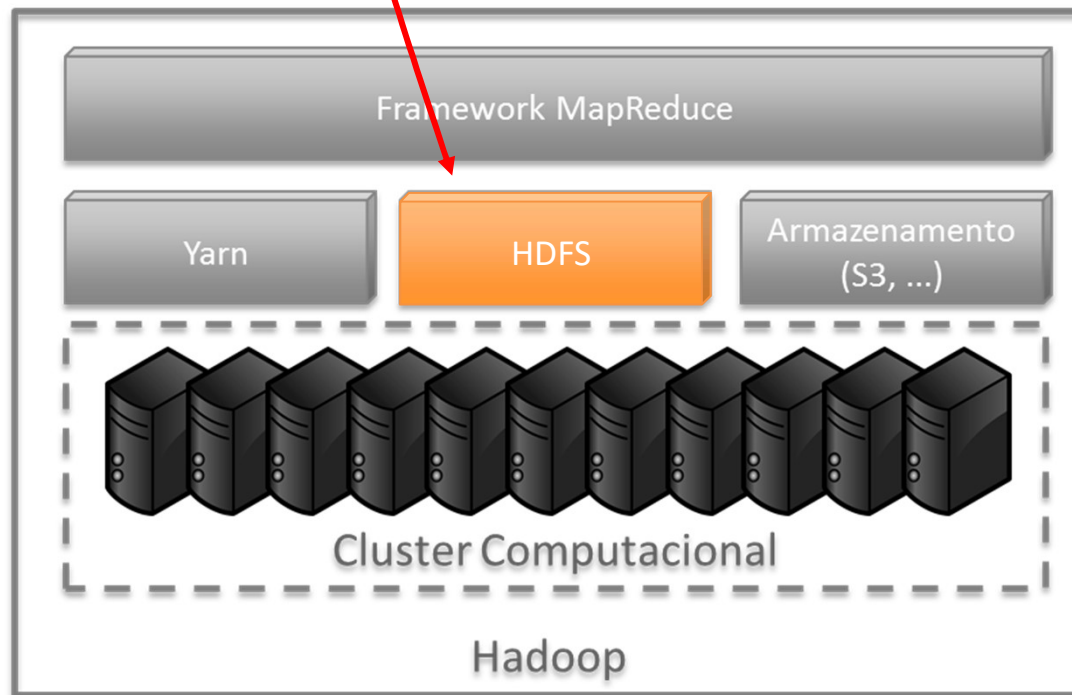
### Complexidade é transparente ao usuário





- Arquitetura

Armazenamento no Hadoop através do HDFS

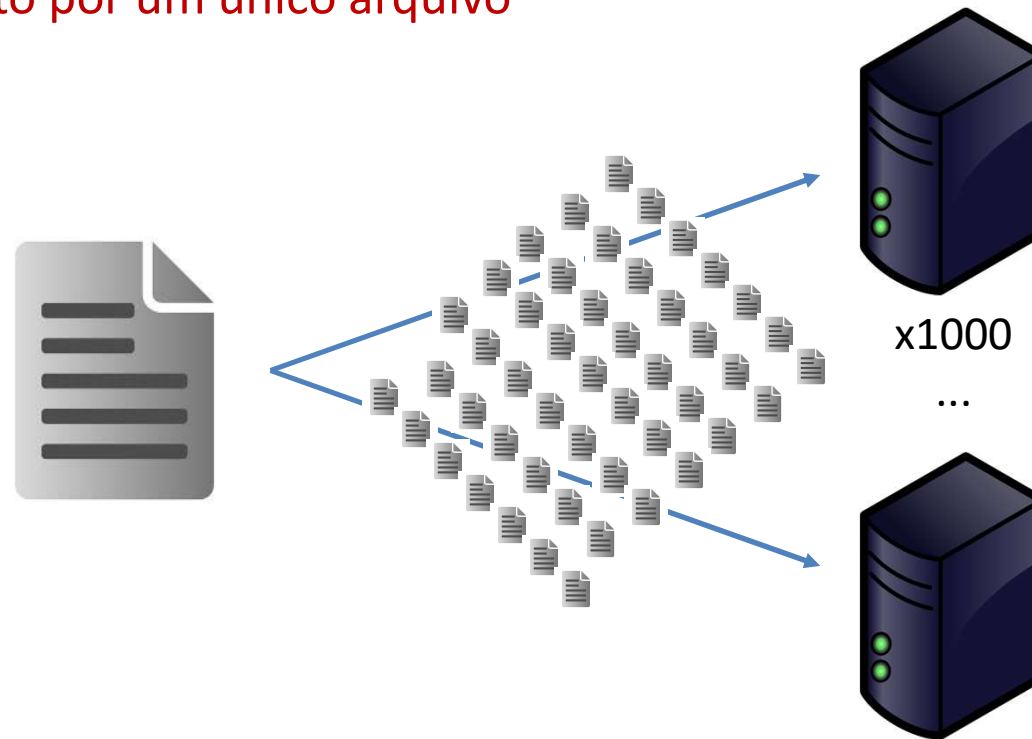


# Agenda

- Big Data
- Hadoop
  - **HDFS**
  - MapReduce
  - Apache Spark

# Armazenamento em Big Data

- Como vocês armazenariam 1 PB de dados?
  - 1000 Terabytes
  - Composto por um único arquivo

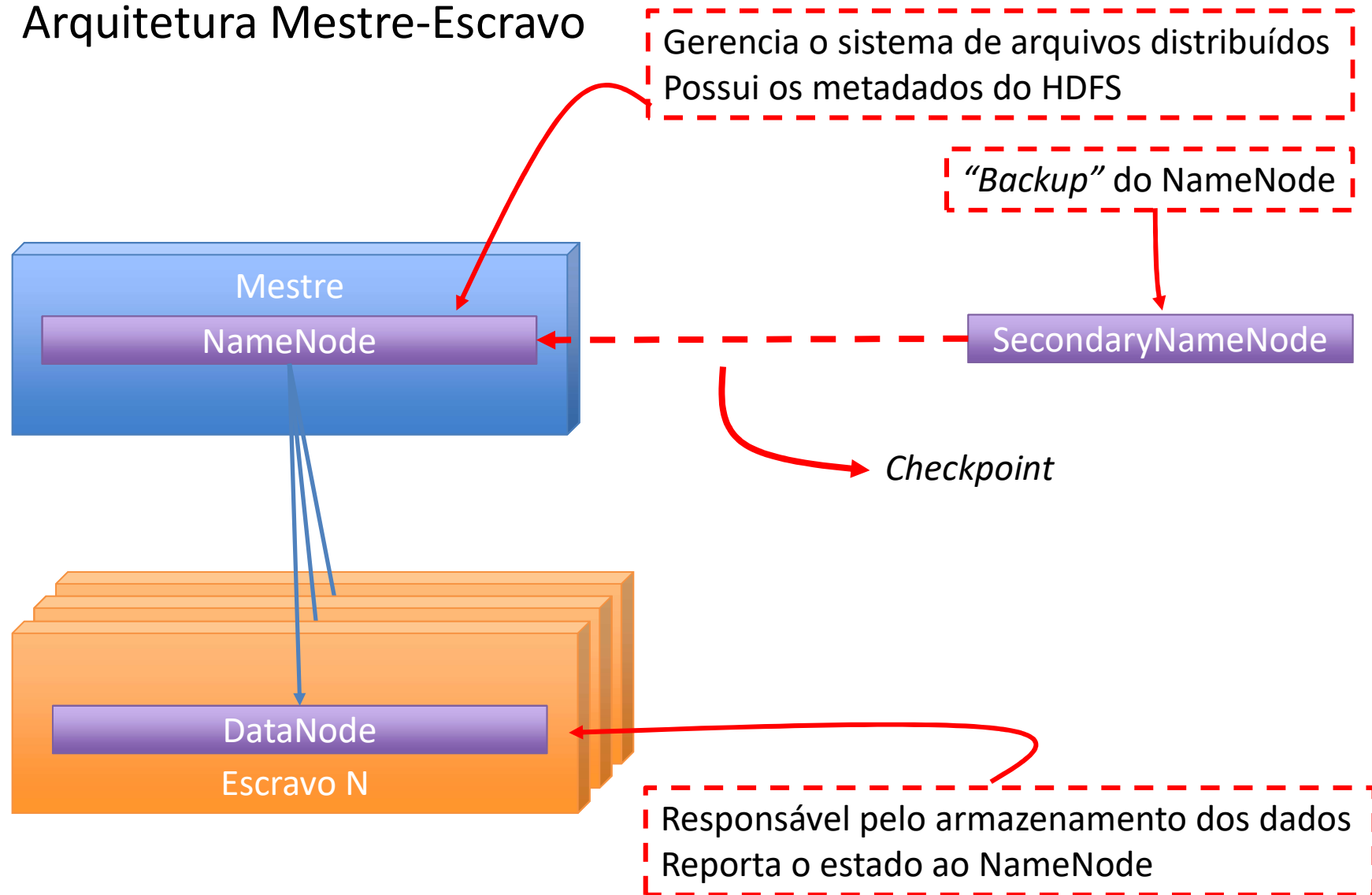


# HDFS

- Hadoop Distributed File System (HDFS)
- Ideal para
  - Grandes massas de dados
  - Acesso de maneira paralela aos dados
- Não recomendável para
  - Diversos arquivos pequenos
  - Acessos aleatórios aos arquivos
  - Leitura de baixa latência

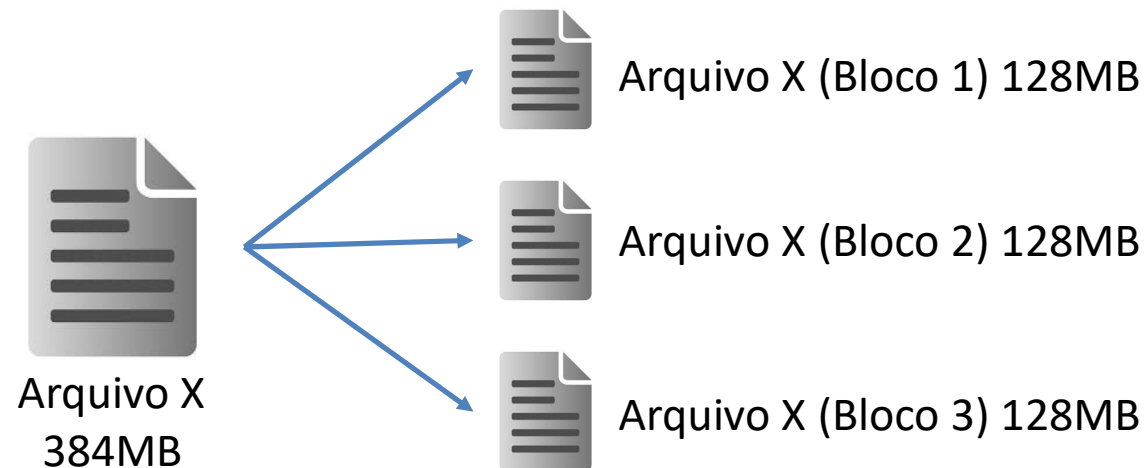
# HDFS

- Arquitetura Mestre-Escravo



# HDFS

- Arquivos no HDFS são quebrados em blocos
  - Bloco é uma unidade base do HDFS para leitura/escrita
    - Similar a um bloco no sistema de arquivos tradicional
  - Tamanho padrão de 64 MB (128 MB no nosso ambiente)
  - Bloco pode ser replicado e distribuído entre vários nós
    - Escalabilidade!
  - NameNode gerencia os blocos relacionados ao arquivo
  - Torna HDFS tolerante a falha!



# HDFS

## ■ NameNode

- Responsável por gerenciar os metadados do sistema de arquivos

- FSNames

- FSName -> Blocos

- Bloco -> Replicas

- Atua como interface ao HDFS

- Responsável pela interação

- Cliente

- DataNodes

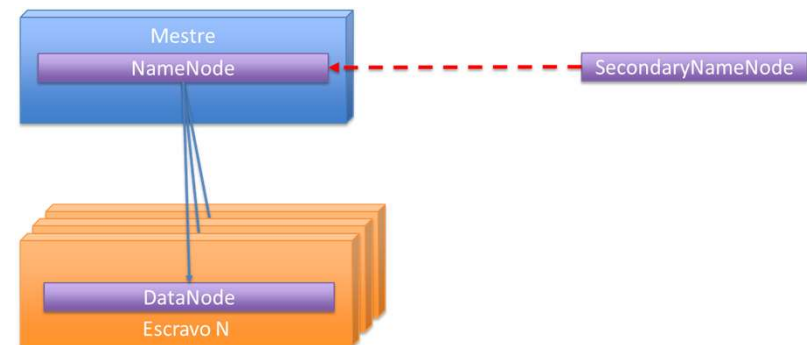
- SecondaryNameNode

*Gerenciamento do FS*

*Gerenciamento do arquivo*

*Tolerancia a falha*

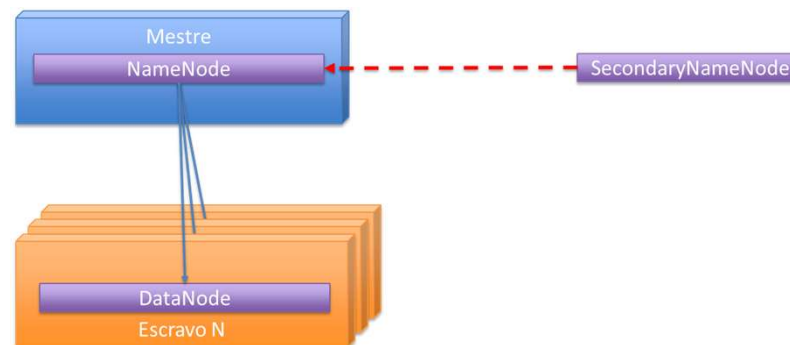
*“Backup” dos metadados do NameNode!*



# HDFS

## ■ SecondaryNameNode

- Principal função é atuar como um backup dos metadados
- Não atua como um backup em produção do NameNode
  - *Serve como Checkpoint*
- Processo de backup
  - *Cópia dos metadados do NameNode*
  - *Merge dos metadados*
  - *Substitui os metadados e limpa o cache atual*

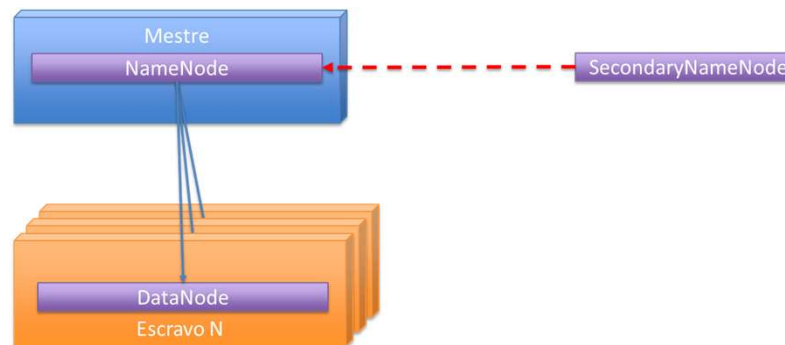




# HDFS

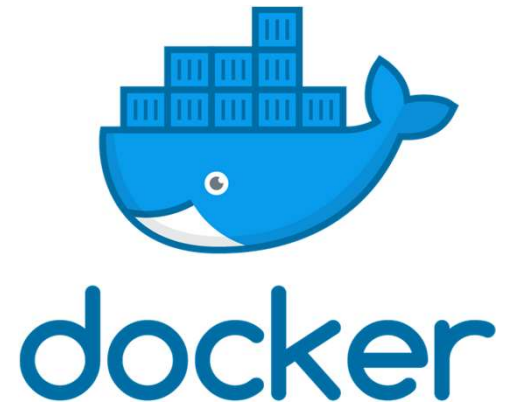
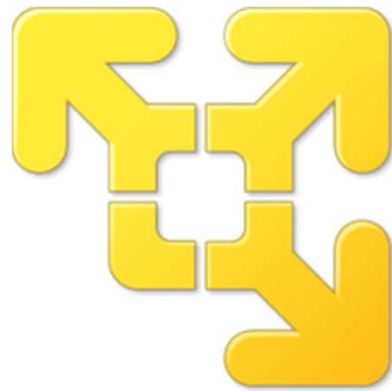
## ■ DataNode

- Armazena os blocos dos dados
  - Não possui conhecimento sobre os metadados!
- Recebe os blocos dos clientes
- Recebe os blocos dos DataNodes
  - Replicação!
- Recebe comandos de gerenciamento do NameNode
  - Remoção de arquivos
  - Alteração



# Prática

- Liguem a máquina virtual
  - Usuário: **docker**
  - Senha: **docker**



- Vamos ligar o nosso “cluster” hadoop agora

# Frameworks de Big Data

**Eduardo Viegas**



ESCOLA  
**POLITÉCNICA**

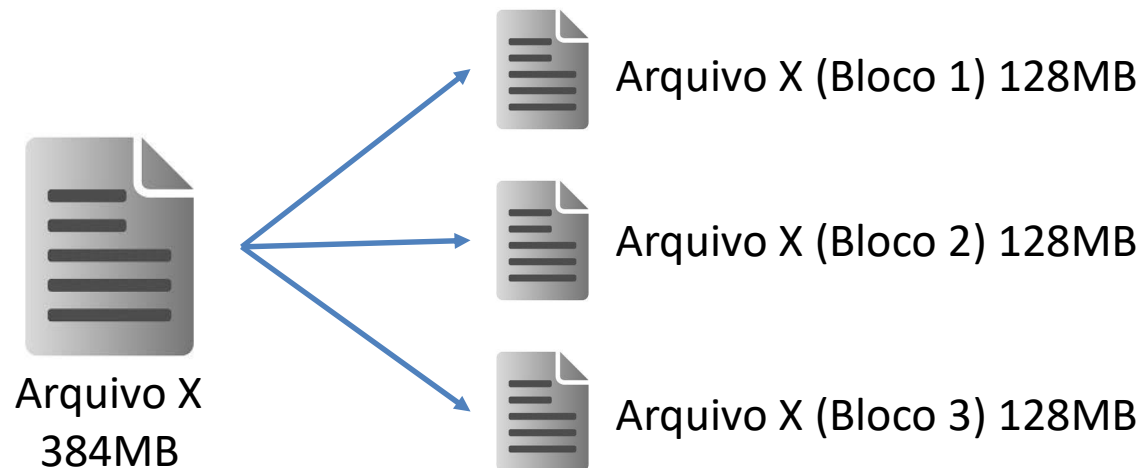
# Cronograma das aulas

Data	Conteúdo
01/02	Big Data, Hadoop e HDFS
15/02	HDFS, MapReduce
29/02	HDFS, MapReduce, Apache Spark
14/03	HDFS, Apache Spark
21/03	HDFS, Apache Spark, Projeto

# HDFS

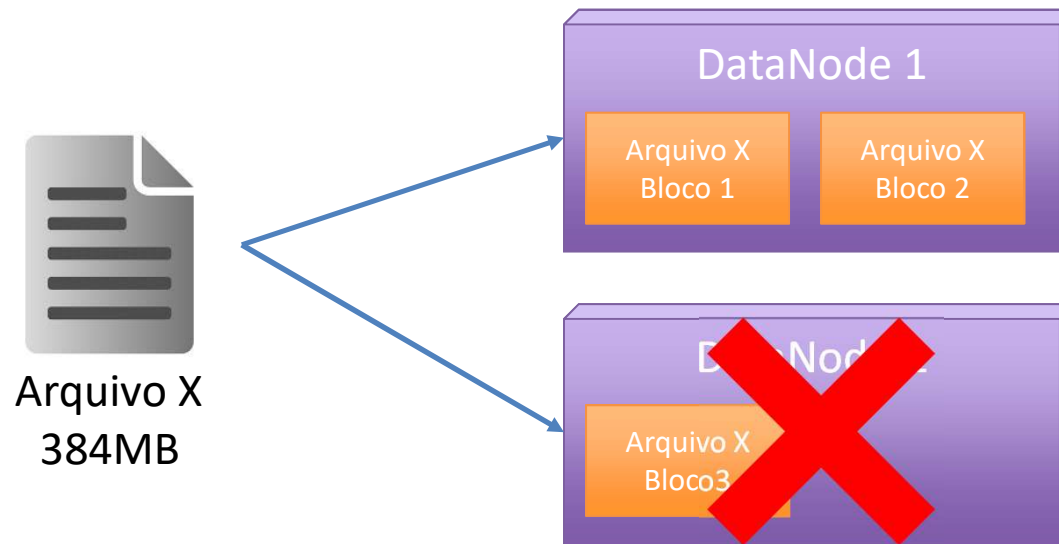
## ■ Prática HDFS

- Acesse o nosso servidor
- Crie um arquivo no sistema de arquivos local
  - `fallocate -l 1G teste.img`
- Crie uma pasta no HDFS
- Copie o arquivo para o HDFS
- Acesse cada datanode, e note os blocos em `/hadoop/dfs/data/current`
- Recupere o arquivo, exclua do HDFS, e note novamente os blocos
- Encerre a execução de um datanode



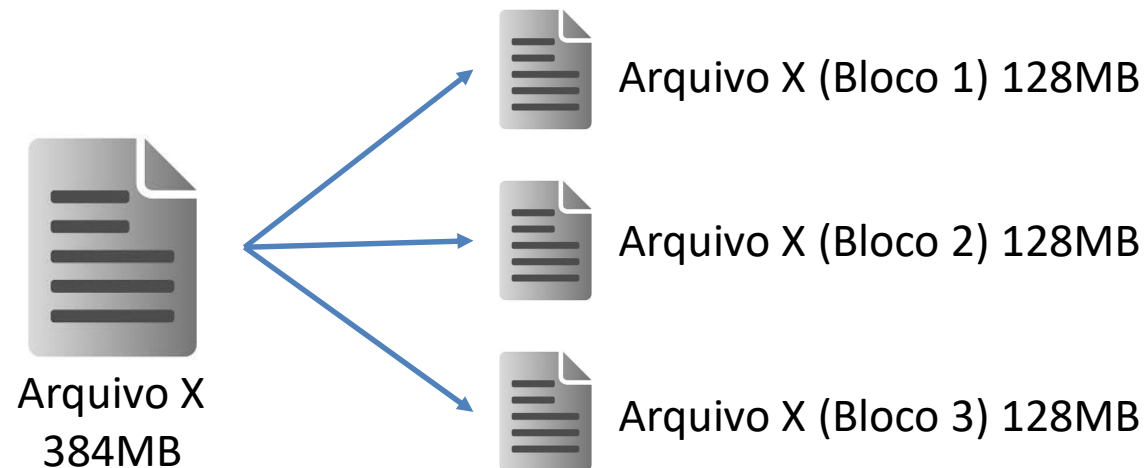
# HDFS

- Okay, HDFS divide o arquivo em blocos, e distribui entre os DataNodes
- Mas e se um DataNode falhar?
  - Como os blocos estão distribuídos caso um nó falhe o arquivo inteiro é perdido?



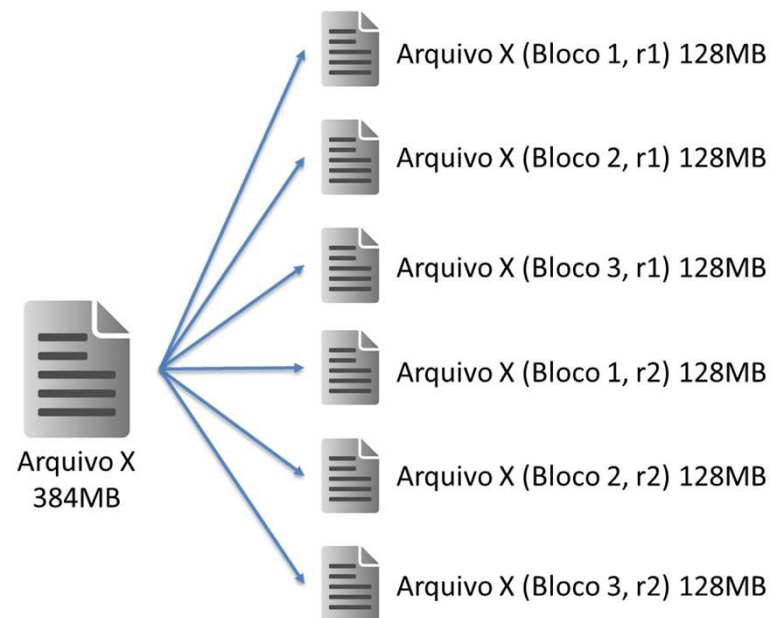
# HDFS

- HDFS utiliza o conceito de réplica para garantir tolerância a falha
  - Fator de Replicação = 1



# HDFS

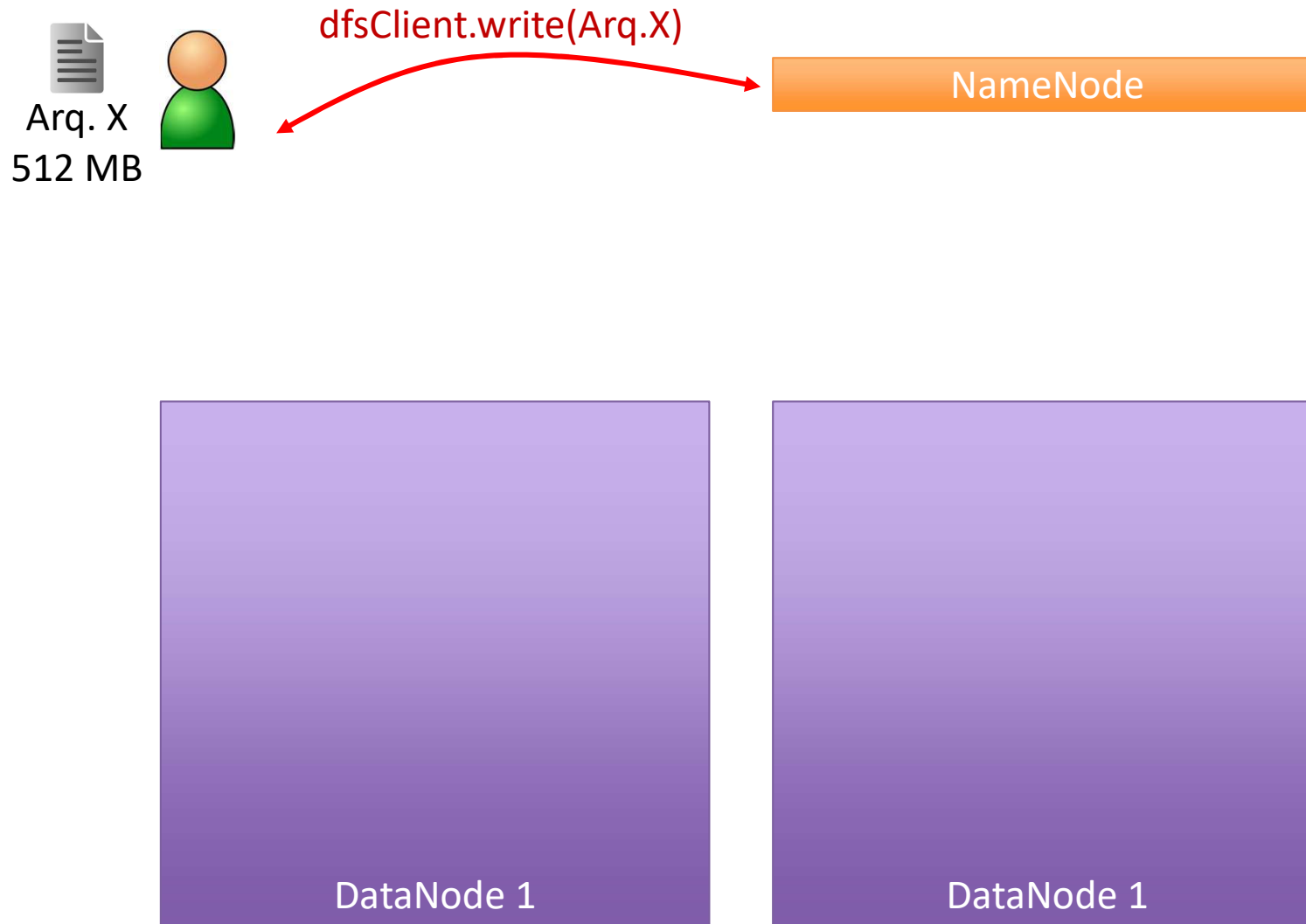
- HDFS utiliza o conceito de réplica para garantir tolerância a falha
  - Fator de Replicação = 2





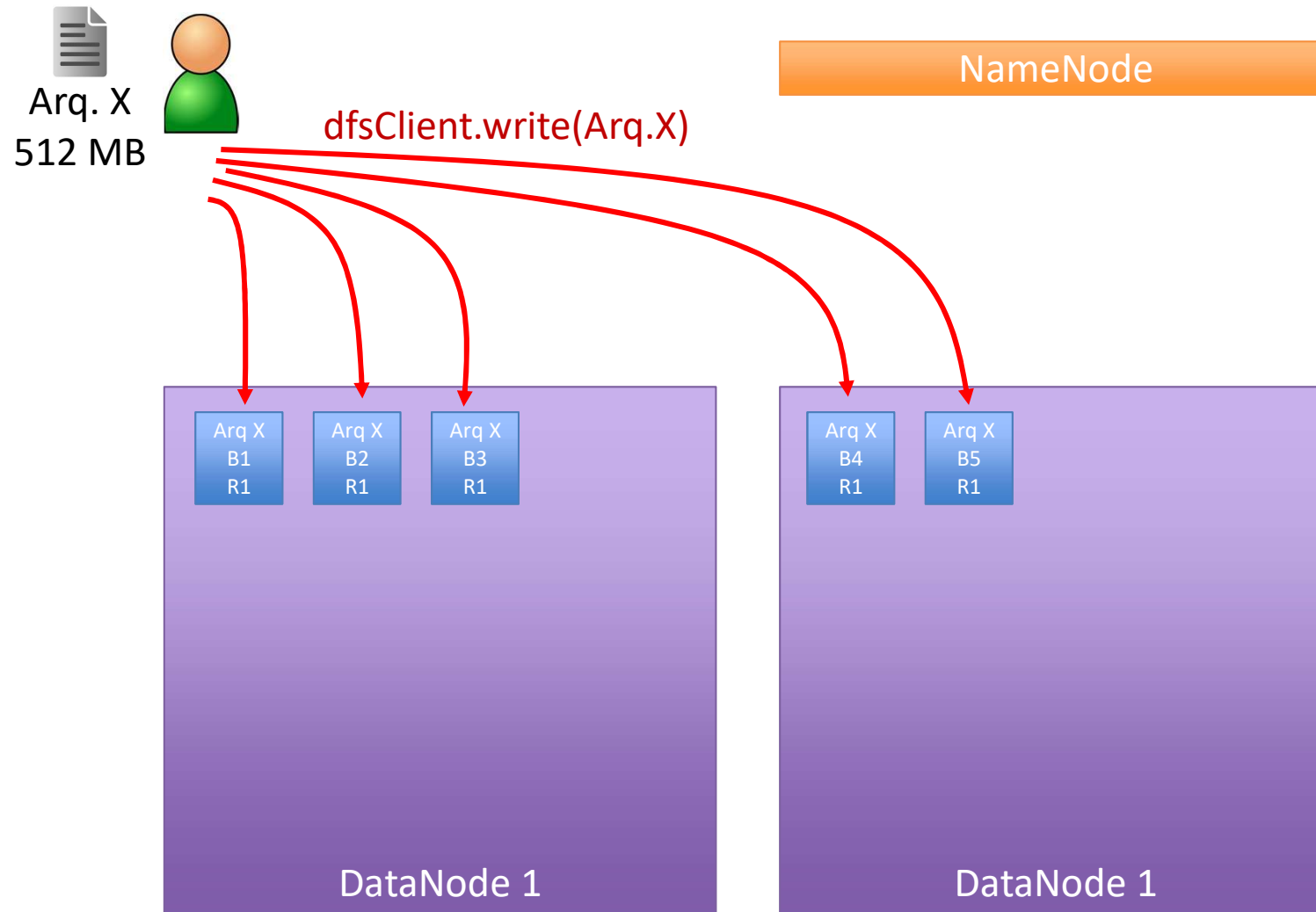
# HDFS

- Okay, mas como funciona o armazenamento?



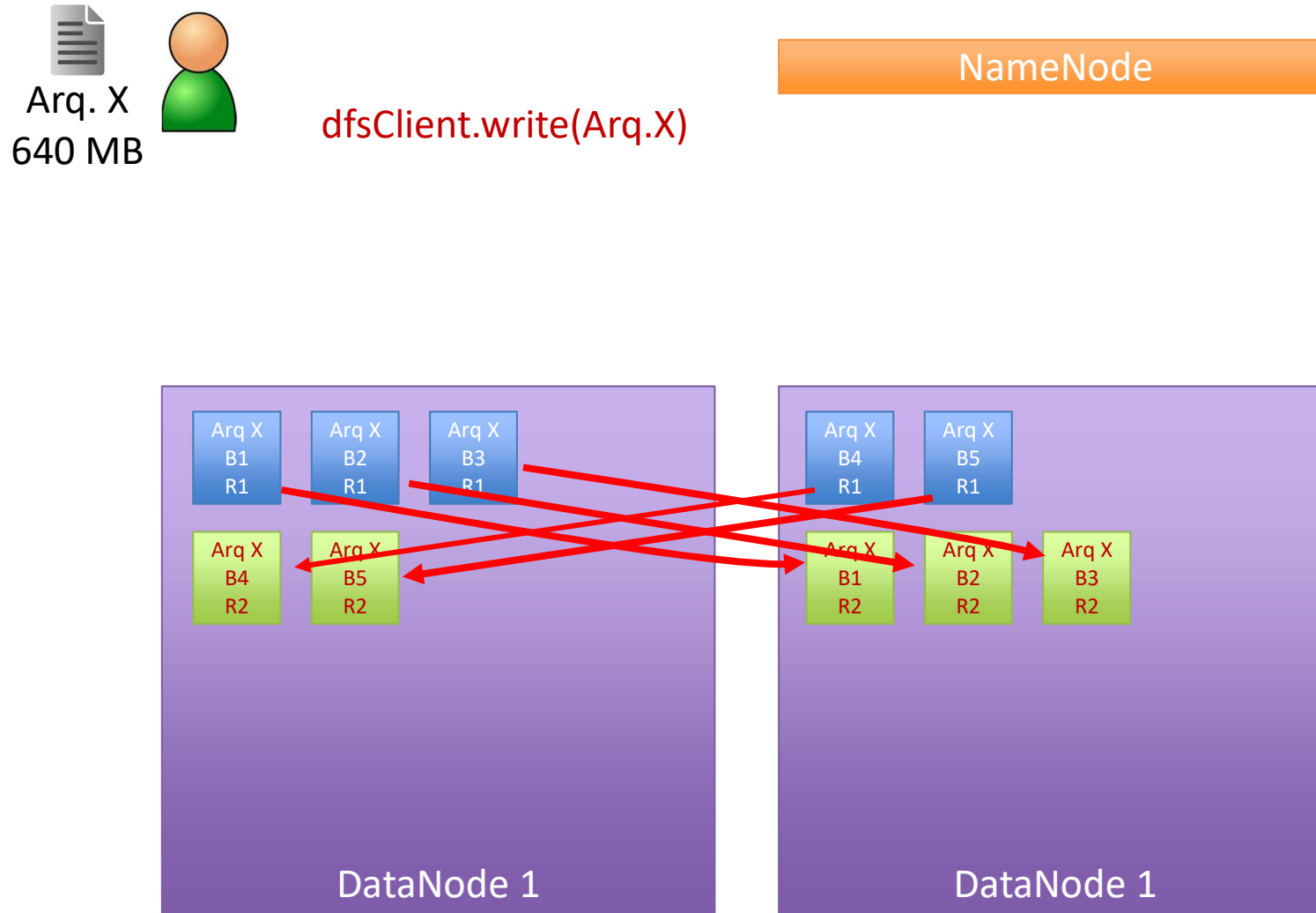
# HDFS

- Okay, mas como funciona o armazenamento?



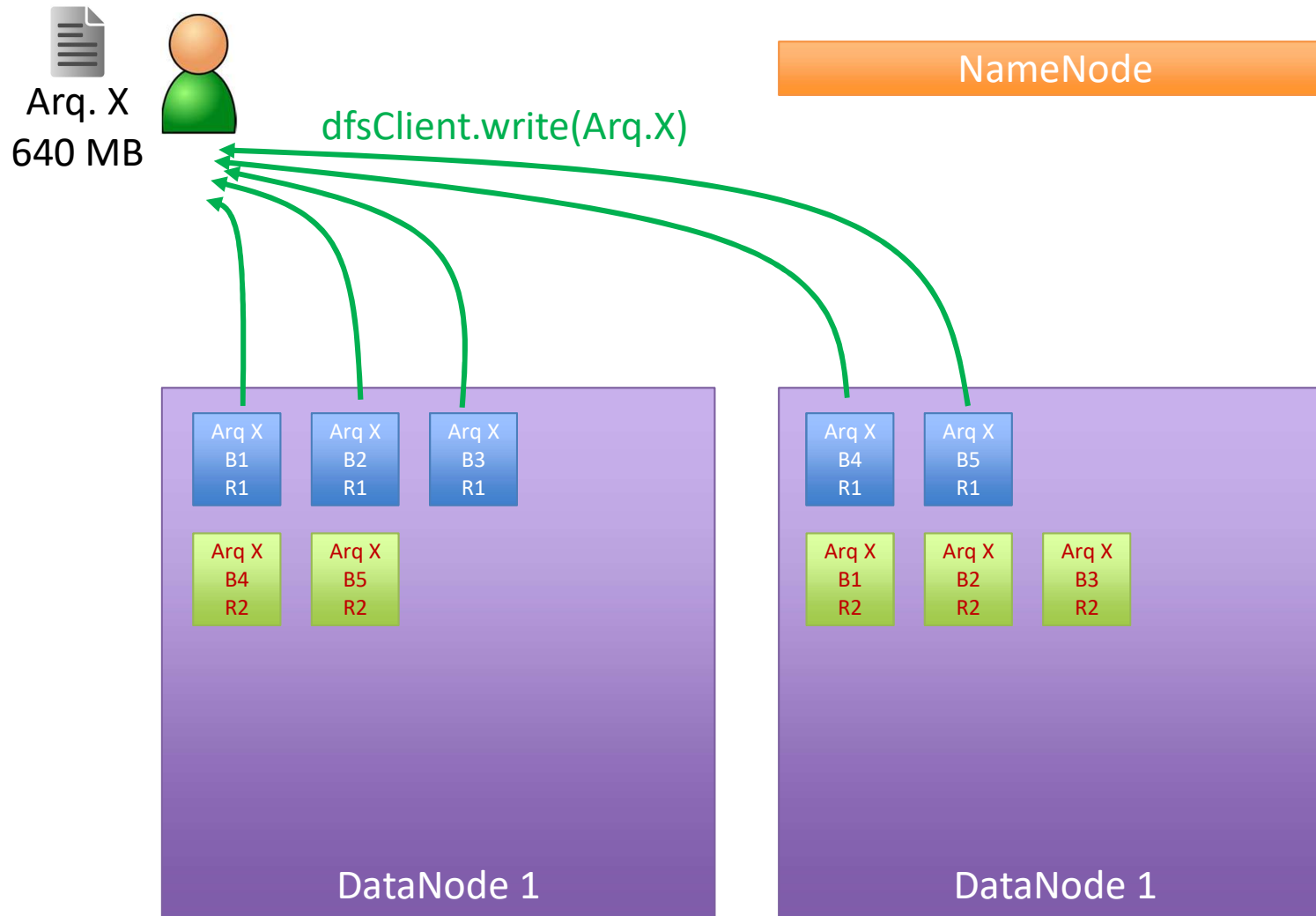
# HDFS

- Okay, mas como funciona o armazenamento?



# HDFS

- Okay, mas como funciona o armazenamento?

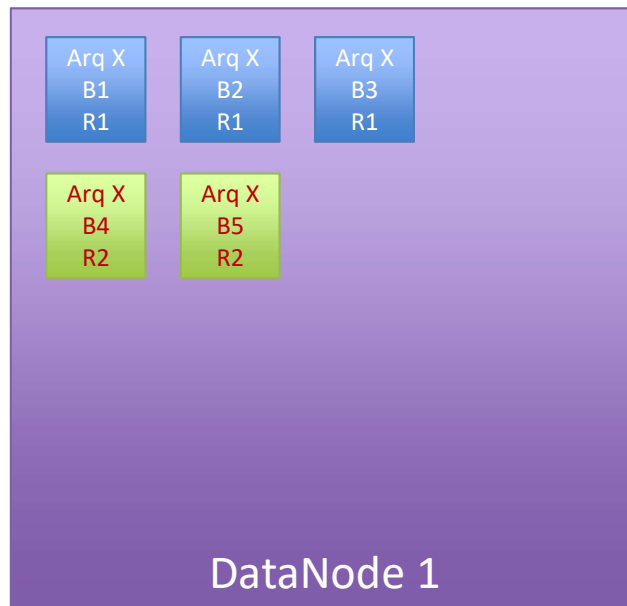


# HDFS

- Okay, mas como funciona o armazenamento?

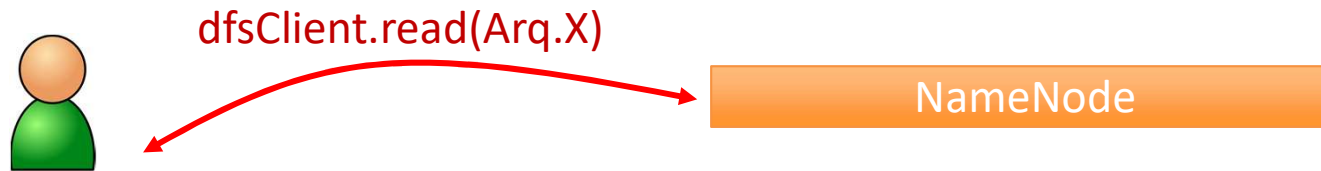


NameNode



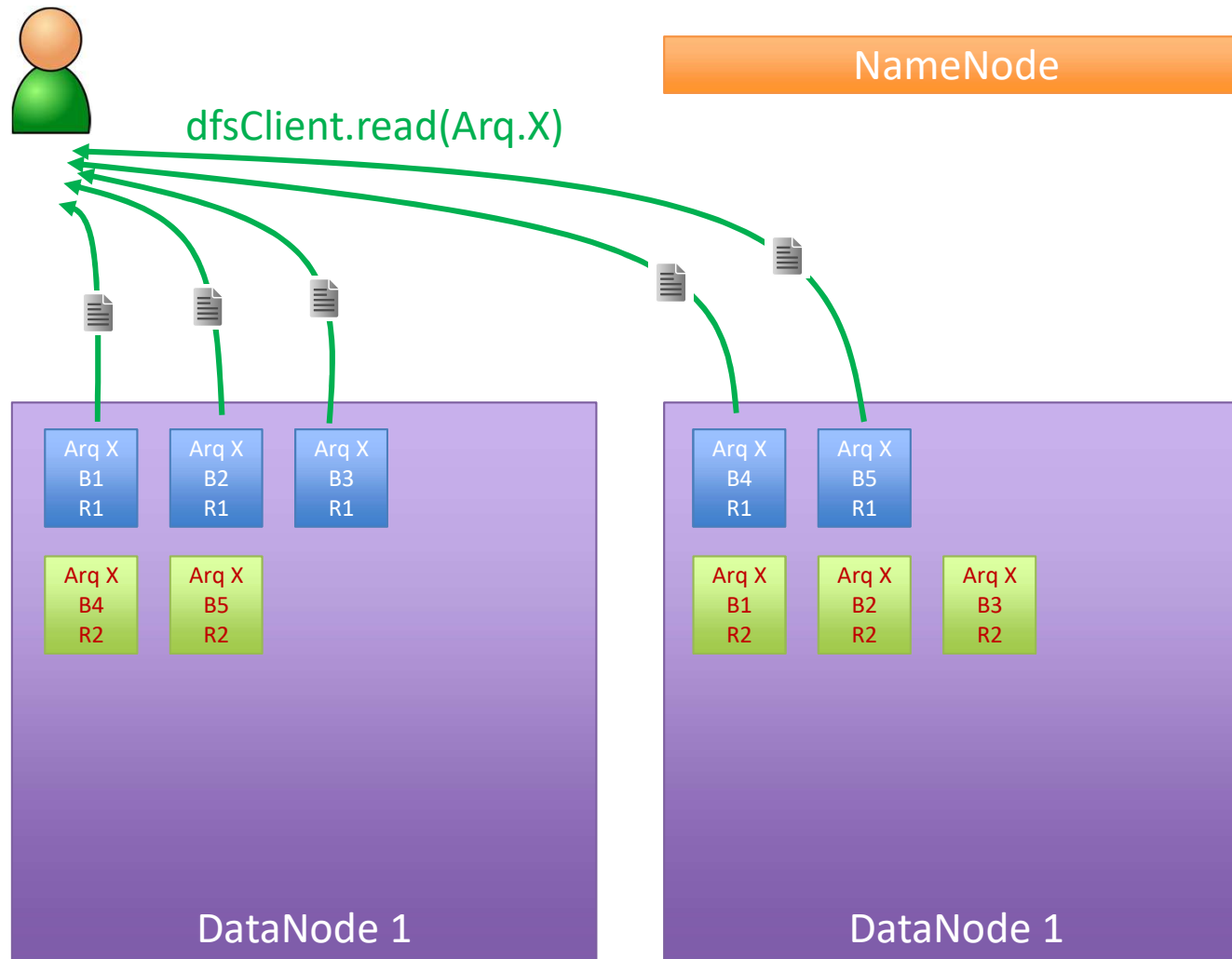
# HDFS

- Okay, mas como funciona o armazenamento?



# HDFS

- Okay, mas como funciona o armazenamento?

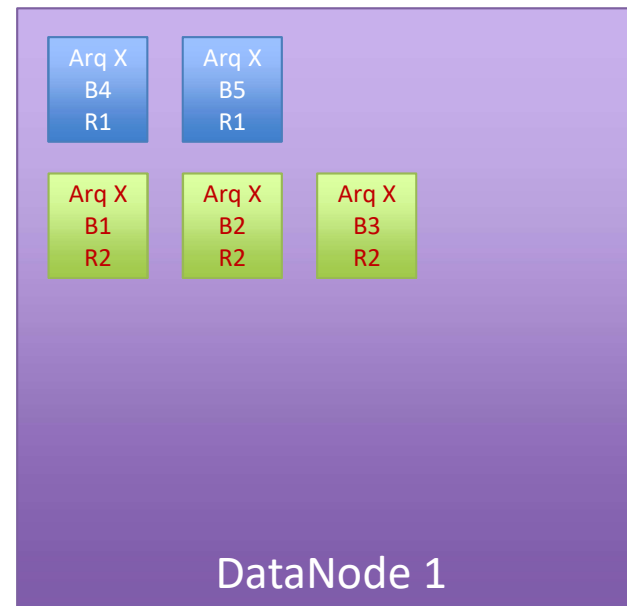
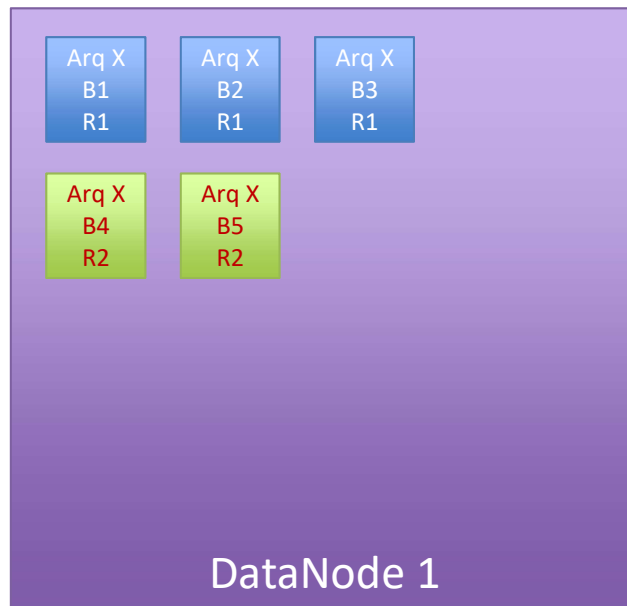


# HDFS

- Okay, mas como funciona o armazenamento?



NameNode





# HDFS

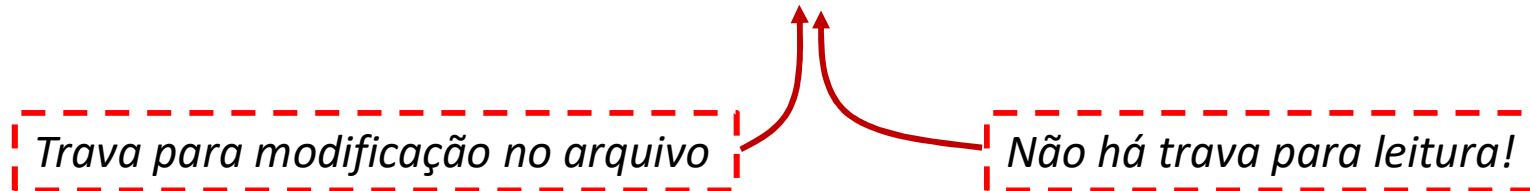
- HDFS utiliza o conceito de réplica para garantir tolerância a falha
- Blocos dos dados são replicados no cluster



- 1/3 das réplicas são enviadas ao nó
  - 1/3 das réplicas são enviadas ao mesmo Rack (incluindo o nó)
  - 1/3 são distribuídas entre o restante do cluster
- HDFS mantém registro sobre a localização dos nós
  - HDFS tenta satisfazer as consultas de acordo com a localidade

# HDFS

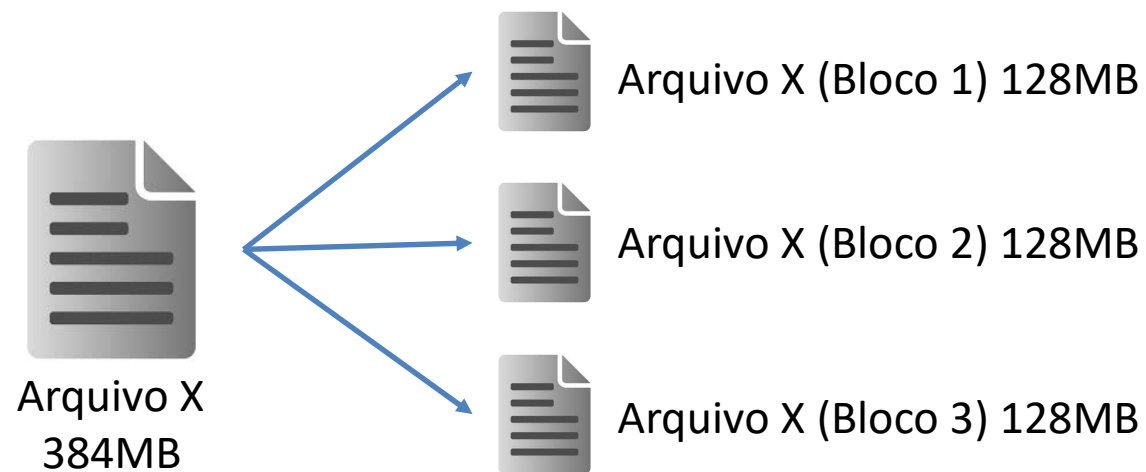
- HDFS utiliza o conceito de *lease* para tratar concorrência



- *Lease* evita inconsistência e comportamento inesperado do sistema
  - Gerenciado pelo NameNode
  - É utilizado em caso de escrita e atualização

# Atividade HDFS

- Atividade 1 do HDFS



# Agenda

- Big Data
- Hadoop
  - HDFS
  - **MapReduce**
  - Apache Spark

# Análise

- Qual a mercadoria com a maior quantidade de transações comerciais no Brasil?

Campo	Descrição
País	País envolvido na transação comercial
Ano	Ano em que a transação foi efetuada
Código	Código da mercadoria
Mercadoria	Descrição da mercadoria
Fluxo	Fluxo, e.g. Exportação ou Importação
Valor	Valor em dólares
Peso	Peso da mercadoria
Unidade	Unidade de medida da mercadoria, e.g. Quantidade de itens
Quantidade	Quantidade conforme a unidade especificada da mercadoria
Categoria	Categoria da mercadoria, e.g. Produto Animal

SQL-like

```
select Mercadoria, COUNT(*) as quantidade from Tabela where Pais = "Brasil" group by Mercadoria
```

# Análise

- Qual a mercadoria com a maior quantidade de transações comerciais no Brasil?

SQL-like

```
select Mercadoria, COUNT(*) as quantidade from Tabela where Pais = "Brasil" group by Mercadoria
```

- Etapas
  - Filtro
  - Ordenação
  - Agrupamento

# Análise

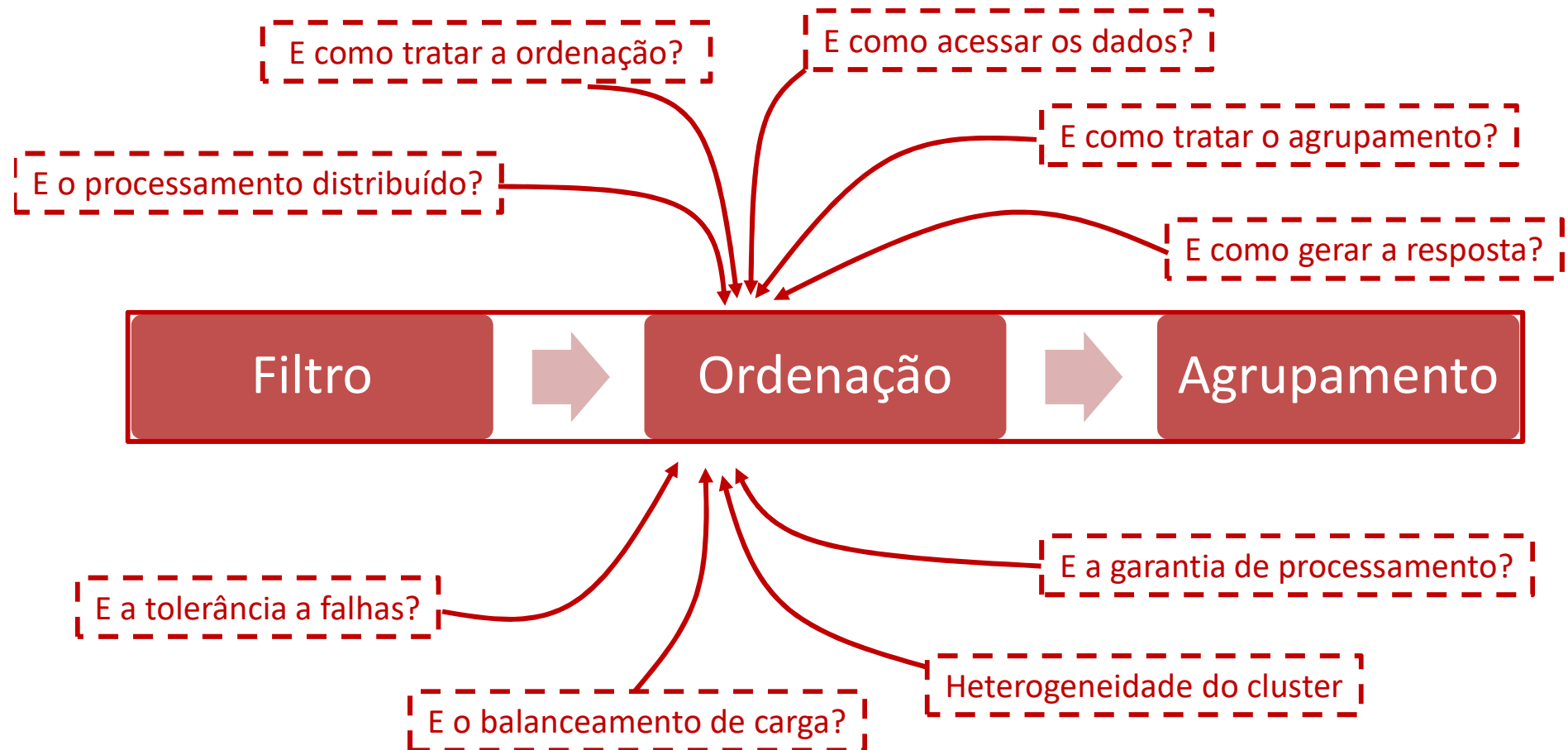
- Suponha que você possui uma base de dados com o seguinte formato
  - 1 TB de dados
  - +1 Bilhão de transações comerciais
  - Dados armazenados no HDFS, mais de 1000 máquinas

Campo	Descrição
País	País envolvido na transação comercial
Ano	Ano em que a transação foi efetuada
Código	Código da mercadoria
Mercadoria	Descrição da mercadoria
Fluxo	Fluxo, e.g. Exportação ou Importação
Valor	Valor em dólares
Peso	Peso da mercadoria
Unidade	Unidade de medida da mercadoria, e.g. Quantidade de itens
Quantidade	Quantidade conforme a unidade especificada da mercadoria
Categoria	Categoria da mercadoria, e.g. Produto Animal

- Qual a mercadoria com a maior quantidade de transações comerciais no Brasil?

# Análise

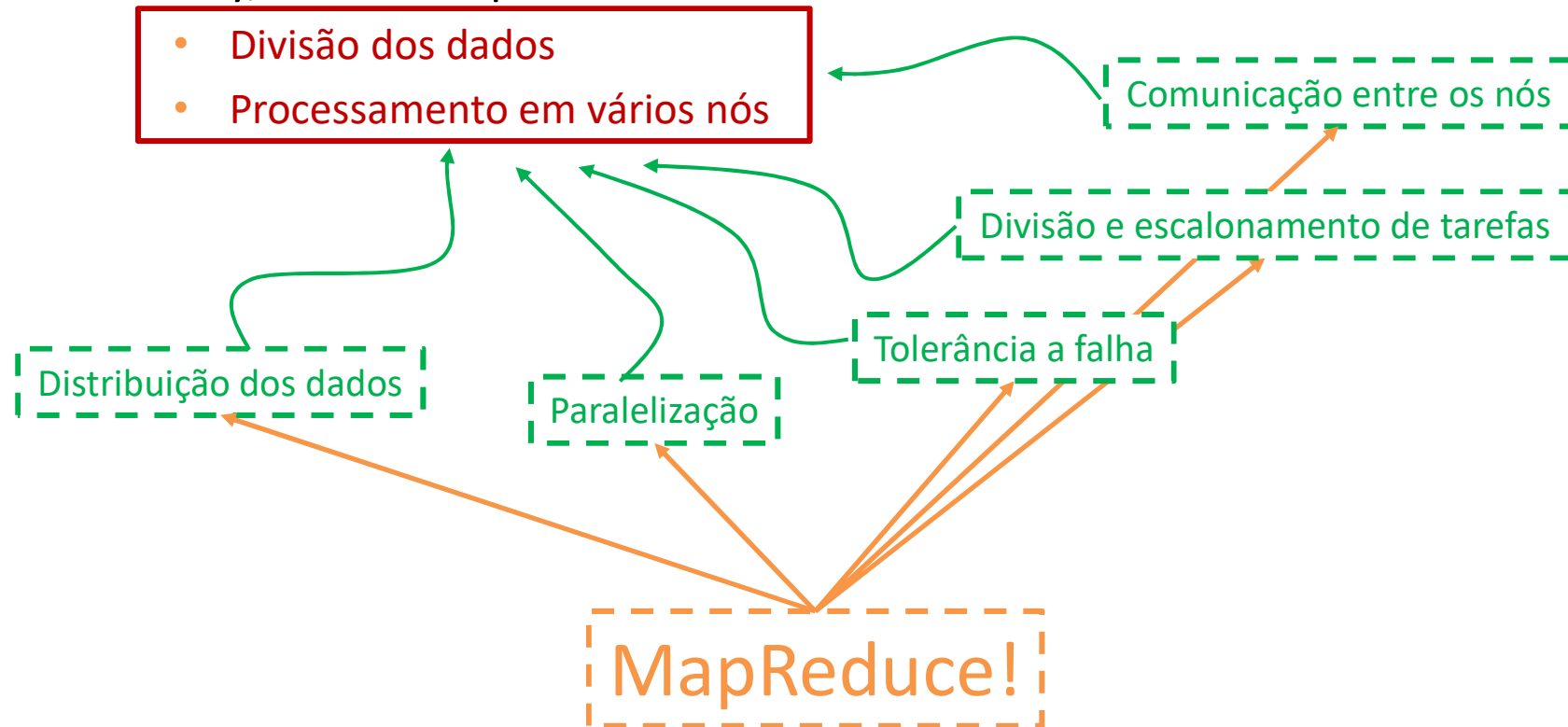
- Podemos fazer isso “manualmente”?





# Análise

- HDFS abstrai a complexidade de armazenamento distribuído de dados
- Okay, mas como processar essa massa de dados armazenada?



# Análise

- MapReduce é o paradigma precursor de análise de dados em cenários de Big Data
  - Framework para computação distribuída
  - Abstrai os conceitos de
    - Paralelização
    - Distribuição dos dados
    - Balanceamento de carga
    - Tolerância a falha
    - Processamento distribuído
  - Provém uma camada de abstração para o programador
  - Executa as tarefas sobre os arquivos do HDFS
  - Distribui o processamento no cluster
  - Leva a computação para os dados

# MapReduce – Modelo

- Programador deve apenas especificar as funções de **Map** e **Reduce**, o framework se encarrega do resto!

## Map

Aplica uma função  $f$  para cada elemento de uma lista, retornando uma nova lista  
Ex.

Quadrado:      **map [1 2 3 4] -> [1 4 9 16]**  
Filtro > 2:    **map [1 2 1 4 6] -> [4 6]**  
Filtro:        **map ["1;2.2;PUCPR" "1;4;Eduardo"] -> ["PUCPR" "Eduardo"]**

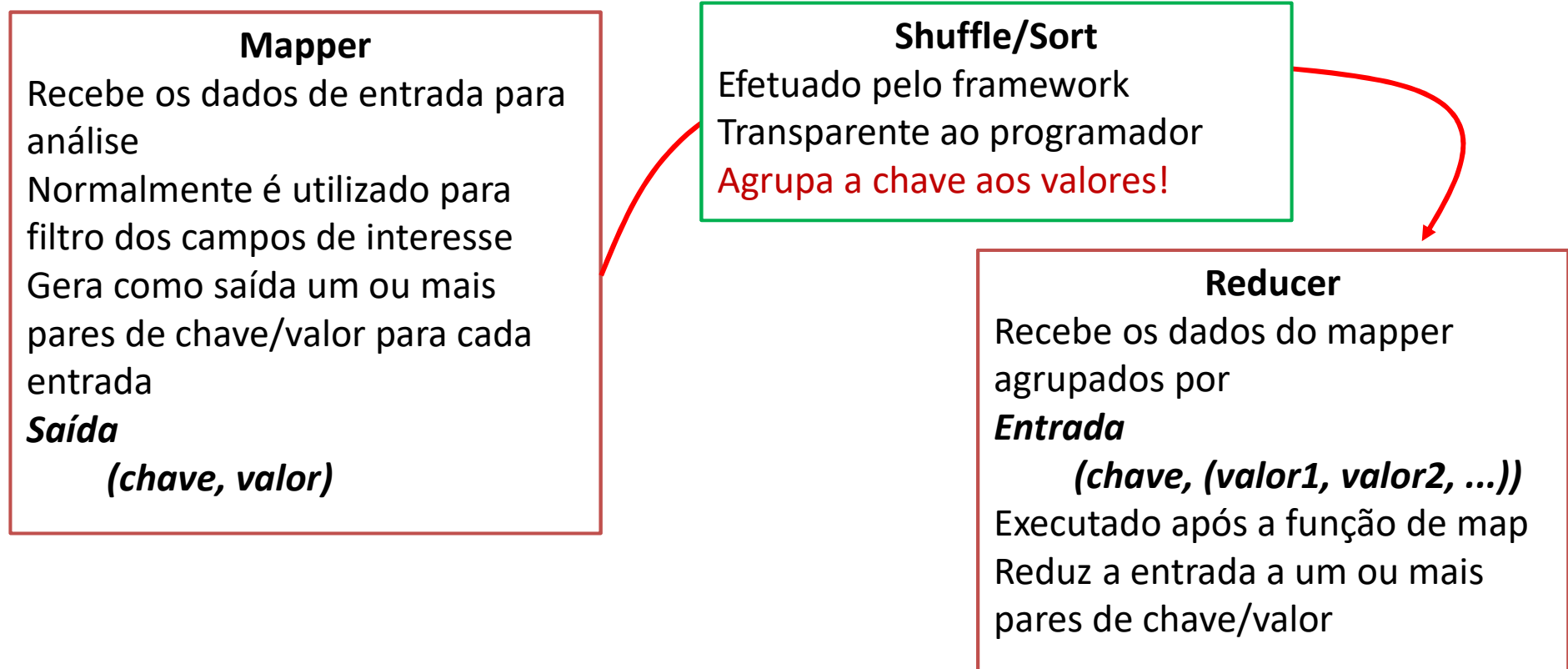
## Reduce

Combina os elementos de uma lista, através de uma função  $f$  para gerar um novo valor  
Ex.

Soma:            **reduce [1 4 9 16] -> [30]**  
Conta:          **reduce [1 2 1 4 6] -> [5]**  
Ocorrencia:    **reduce ["PUCPR" "Eduardo" "Eduardo"] -> ["PUCPR:1" "Eduardo:2"]**

# MapReduce – Modelo

- MapReduce não altera os dados de entrada!
  - Apenas gera novos dados
- Tarefa MapReduce consiste basicamente de **mapper** e **reducer**



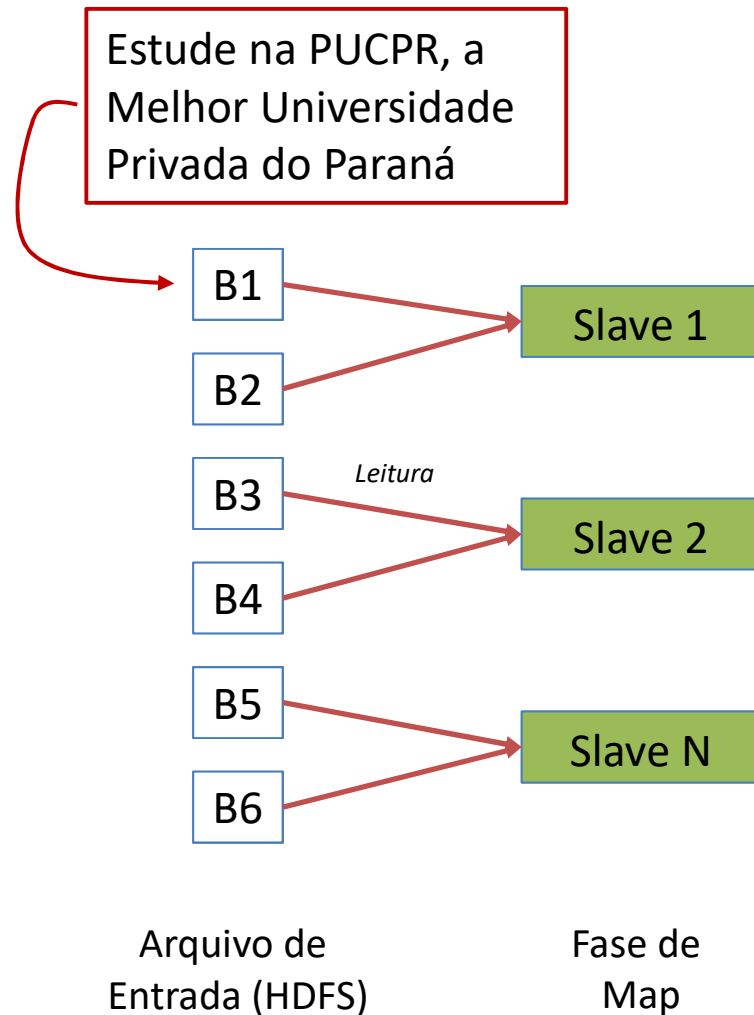
# MapReduce

- Considere um arquivo com o seguinte formato

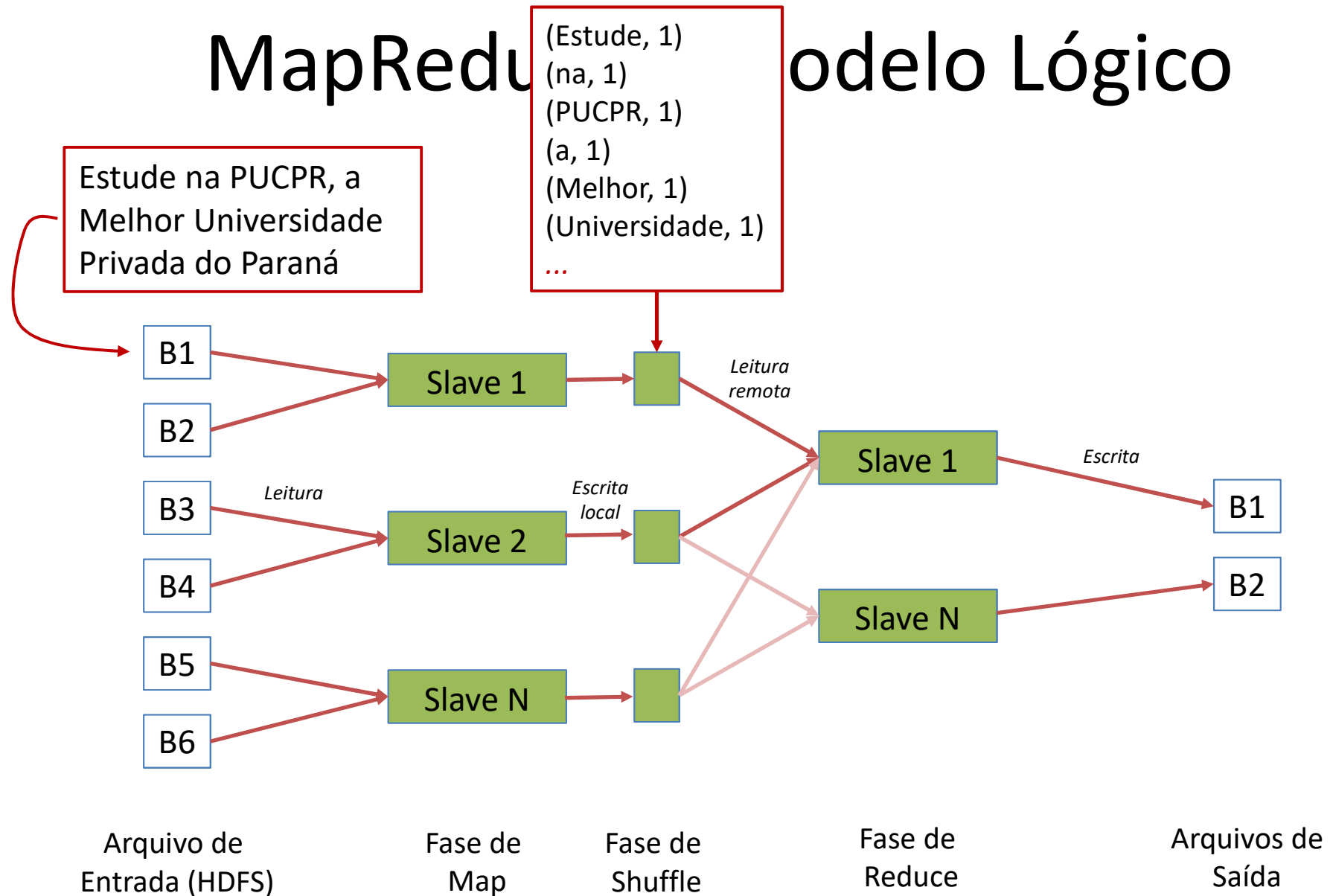
```
Afghanistan;2016;010410;Sheep, live;Export;6088;2339;Number of items;51;01_live_animals
Afghanistan;2016;010420;Goats, live;Export;3958;984;Number of items;53;01_live_animals
Afghanistan;2008;010210;Bovine animals, live pure-bred breeding;Import;1026804;272;Number of items;3769;01_live_animals
Albania;2016;010290;Bovine animals, live, except pure-bred breeding;Import;2414533;1114023;Number of items;6853;01_live_animals
Albania;2016;010392;Swine, live except pure-bred breeding > 50 kg;Import;14265937;9484953;Number of items;96040;01_live_animals
Albania;2016;010511;Fowls, live domestic < 185 grams;Import;2671732;254652;Number of items;5629138;01_live_animals
Albania;2016;010511;Fowls, live domestic < 185 grams;Export;87581;5320;Number of items;115180;01_live_animals
Albania;2016;010519;Poultry, live except domestic fowls, < 185 grams;Import;26485;2908;Number of items;64000;01_live_animals
Albania;2016;010591;Fowls, live domestic > 185 grams;Import;2421513;1926850;Number of items;1006990;01_live_animals
```

- Como podemos estruturar uma função MapReduce para determinar:
  - Qual a quantidade de transações comerciais por país?
  - Qual a quantidade de transações comerciais do tipo *Sheep* por país?
  - Qual o ano que o Brasil realizou mais operações comerciais?

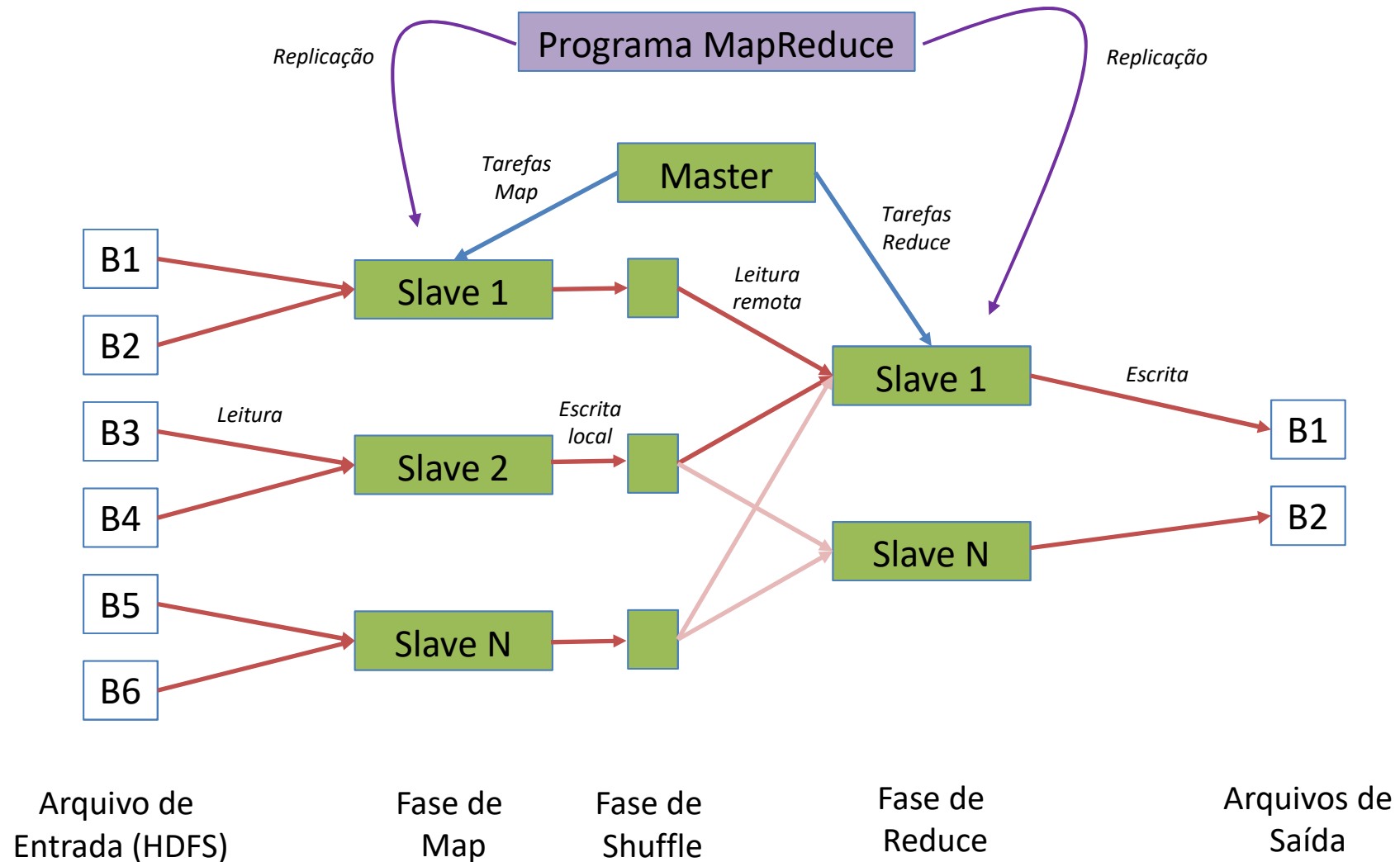
# MapReduce – Modelo Lógico



# MapReduce Modelo Lógico

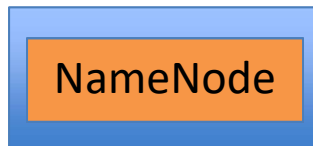


# MapReduce – Modelo Lógico



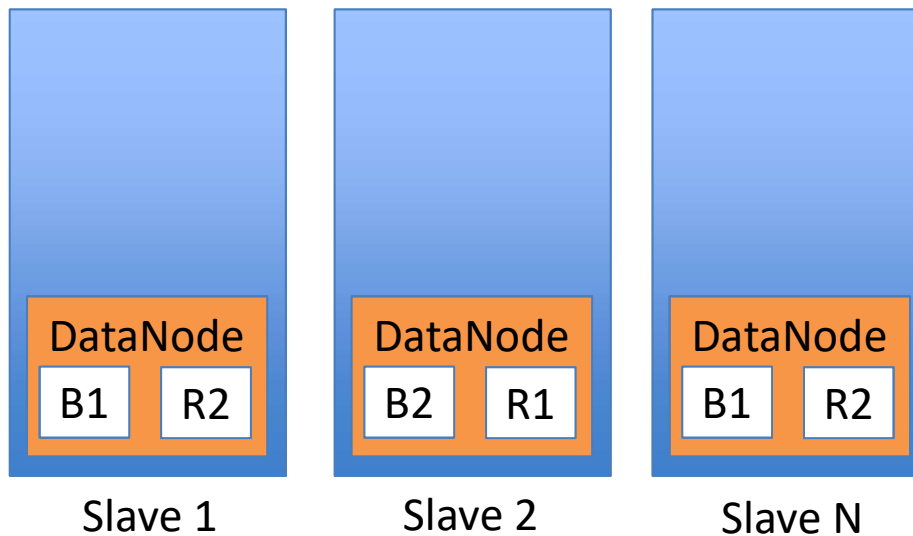


# MapReduce - Arquitetura

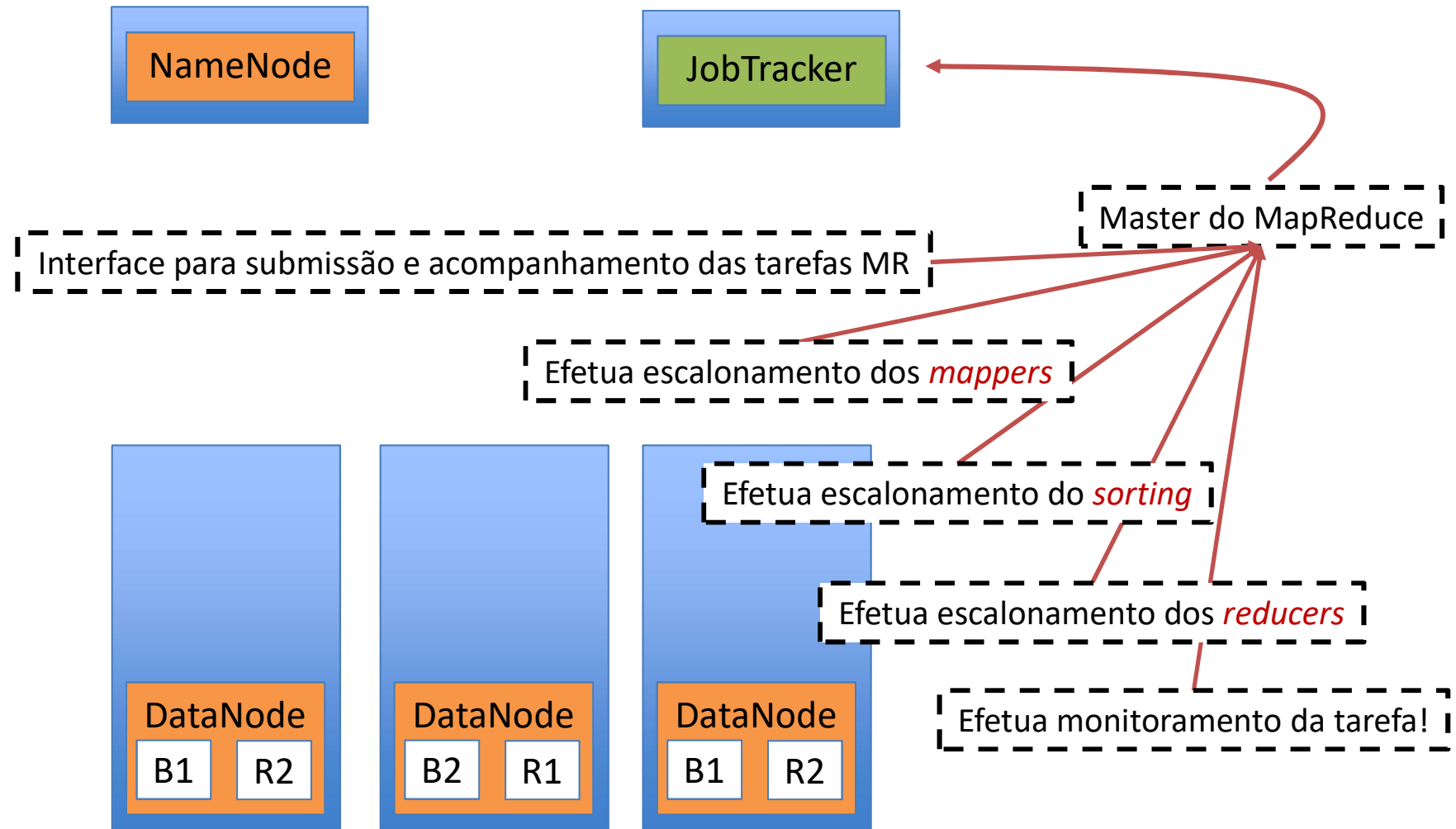


Cada nó é parte de um cluster HDFS

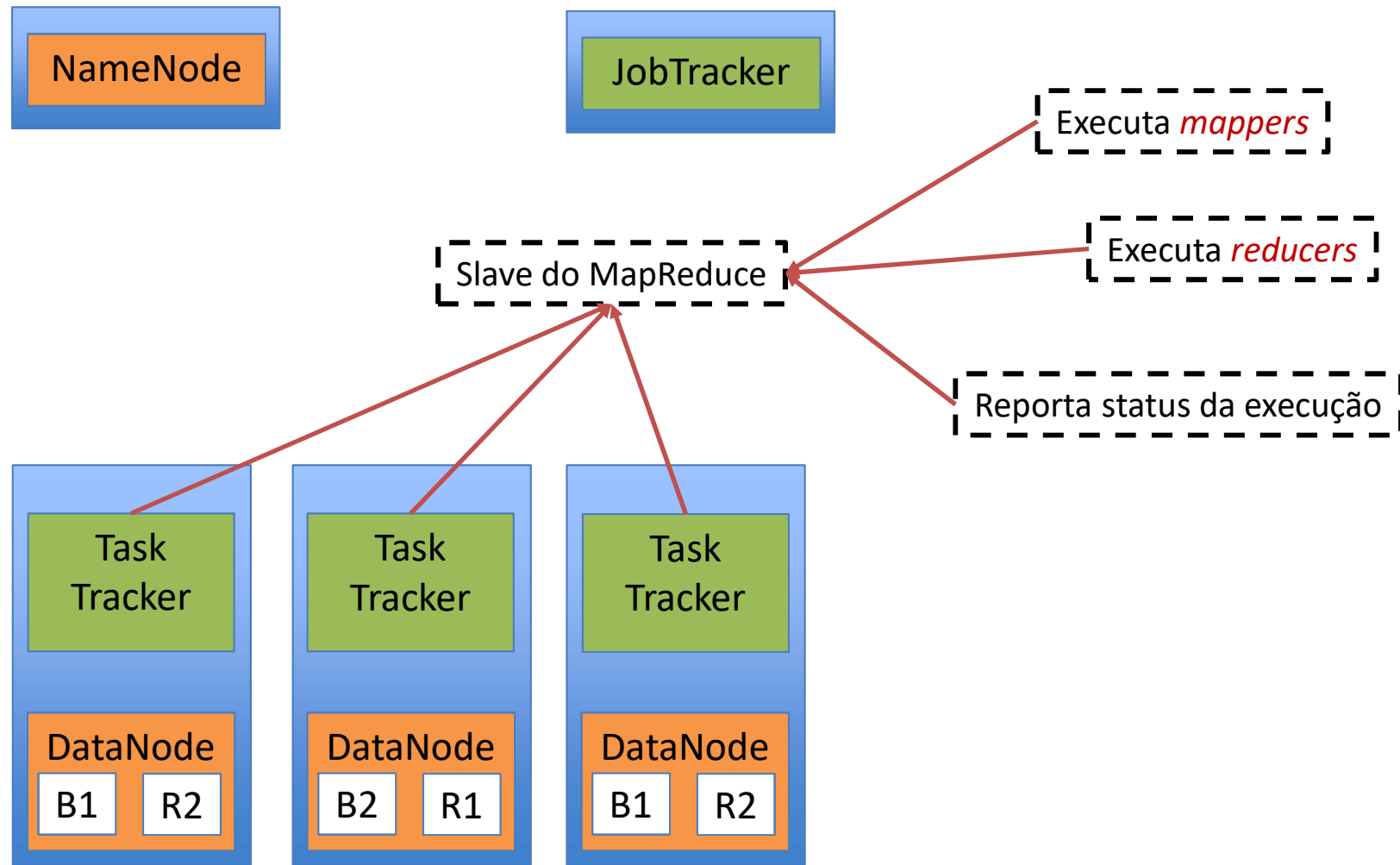
Dados de entrada são armazenados no HDFS e replicados nos nós



# MapReduce - Arquitetura

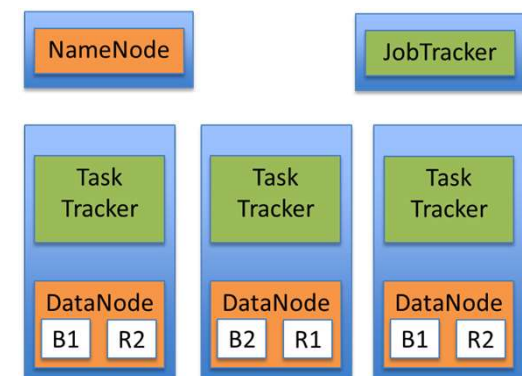


# MapReduce - Arquitetura



# MapReduce - Arquitetura

- Tarefas de Map são executadas em paralelo
  - Geram resultados intermediários de chave/valor
- Fase de *shuffle e sort*
  - Saída da função de map é particionada de acordo com a quantidade de reducers
  - Particionamento garante que o valor com a mesma chave é enviado a mesma partição
  - A partição é sorteada pela chave para agrupar os valores de acordo
- Tarefas de reduce
  - Cada partição é enviada a um reducer
  - Reducers são executados em paralelo
  - Framework garante que o mesmo reducer irá receber os valores da mesma chave



# MapReduce - Vantagens

## ■ Princípio da localidade

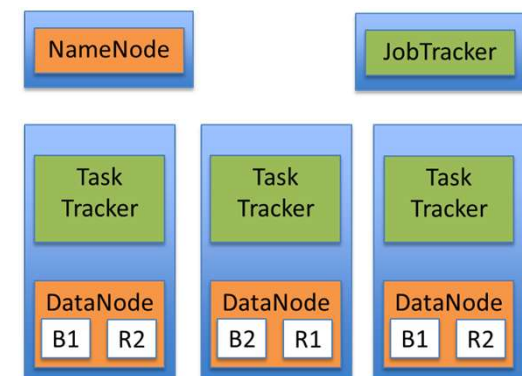
- Framework evita mover os dados pela rede
- TaskTracker é executado geralmente no mesmo nó do DataNode, bloco é lido localmente!
- JobTracker divide as tarefas baseado na localização dos dados

## ■ Paralelismo

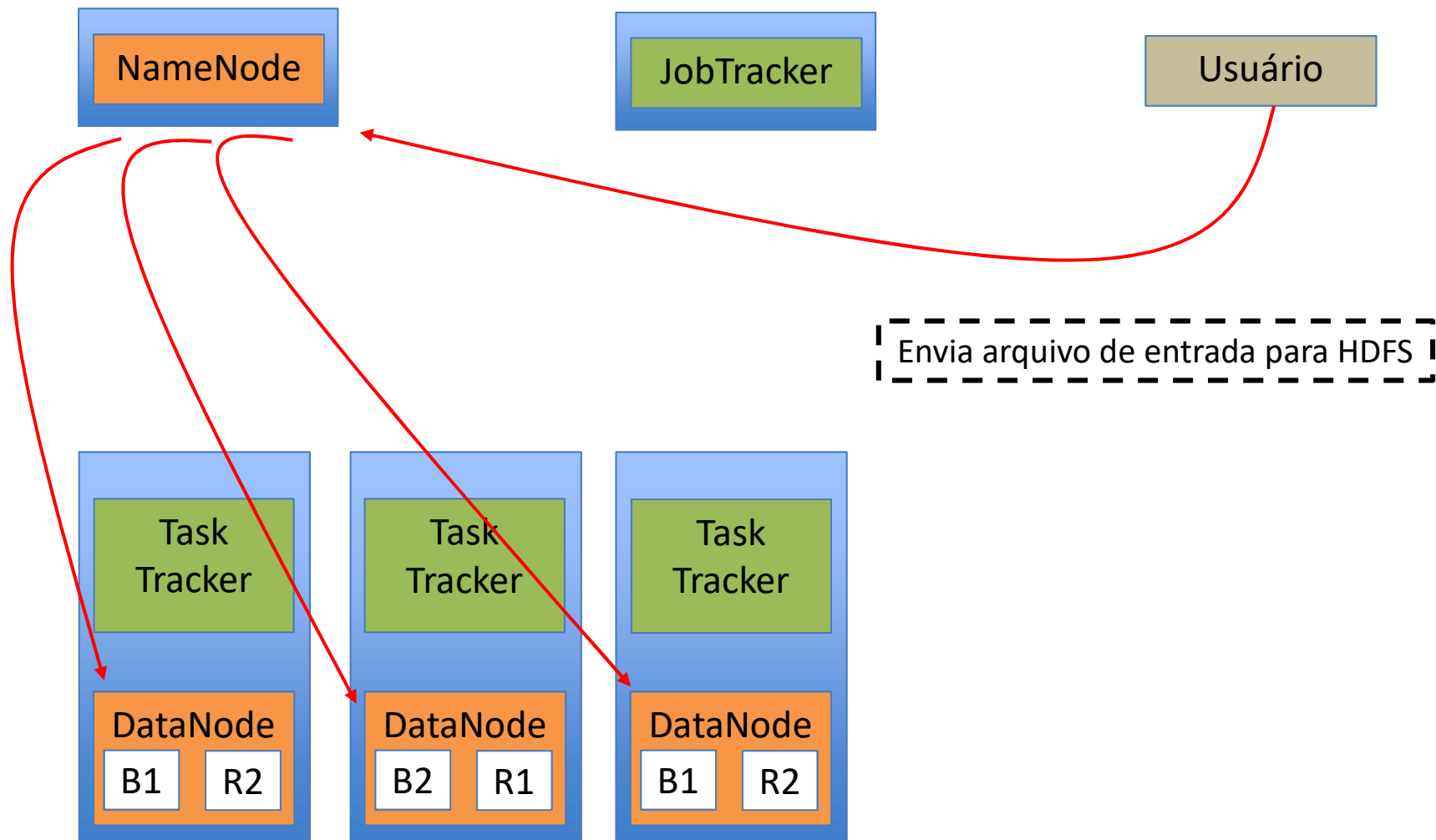
- Função de map é executada paralelamente sobre cada bloco
- Função de reduce é executada em paralelo sobre chaves diferentes
- Função reduce é executada apenas após a execução da função map

## ■ Tolerância a falha

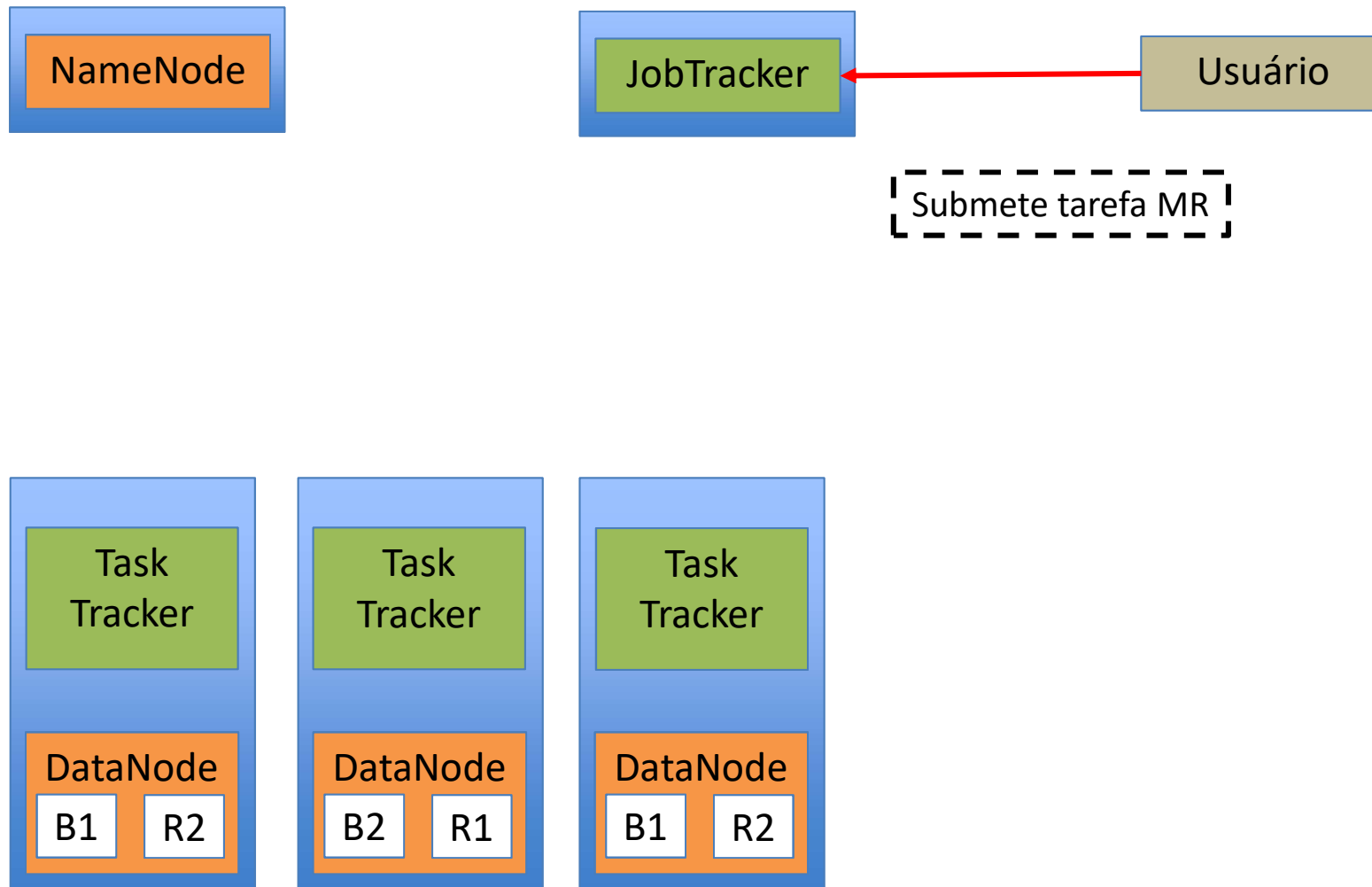
- JobTracker monitora TaskTrackers
- Falhas são tratadas pela reexecução sobre o bloco!
- Se um nó falha, outro nó com as replicas dos blocos reexecuta a tarefa



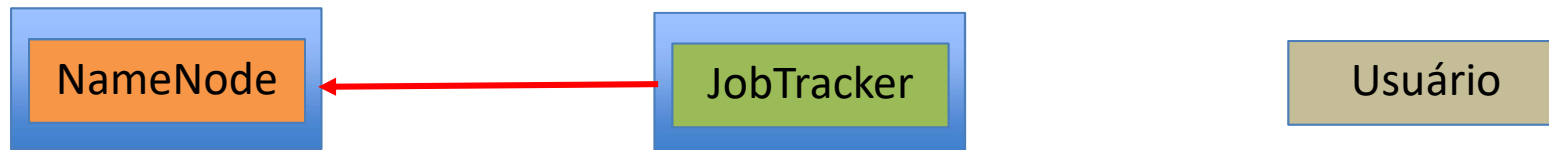
# MapReduce - Processo



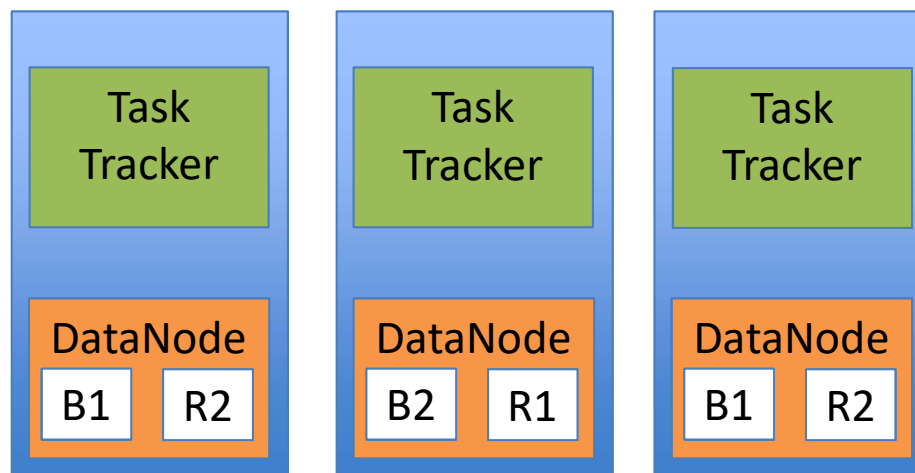
# MapReduce - Processo



# MapReduce - Processo

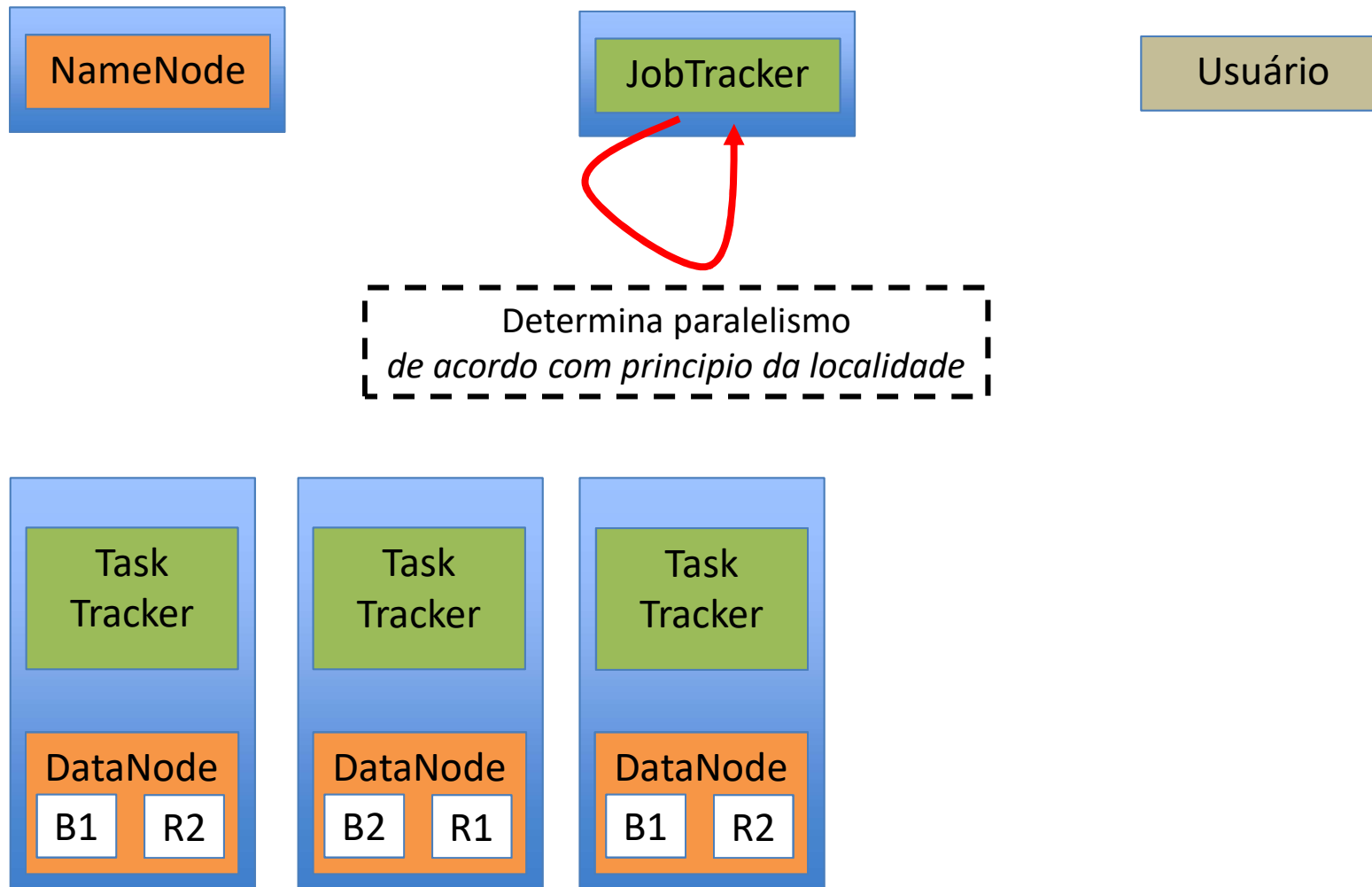


Requisita metadados do arquivo de entrada

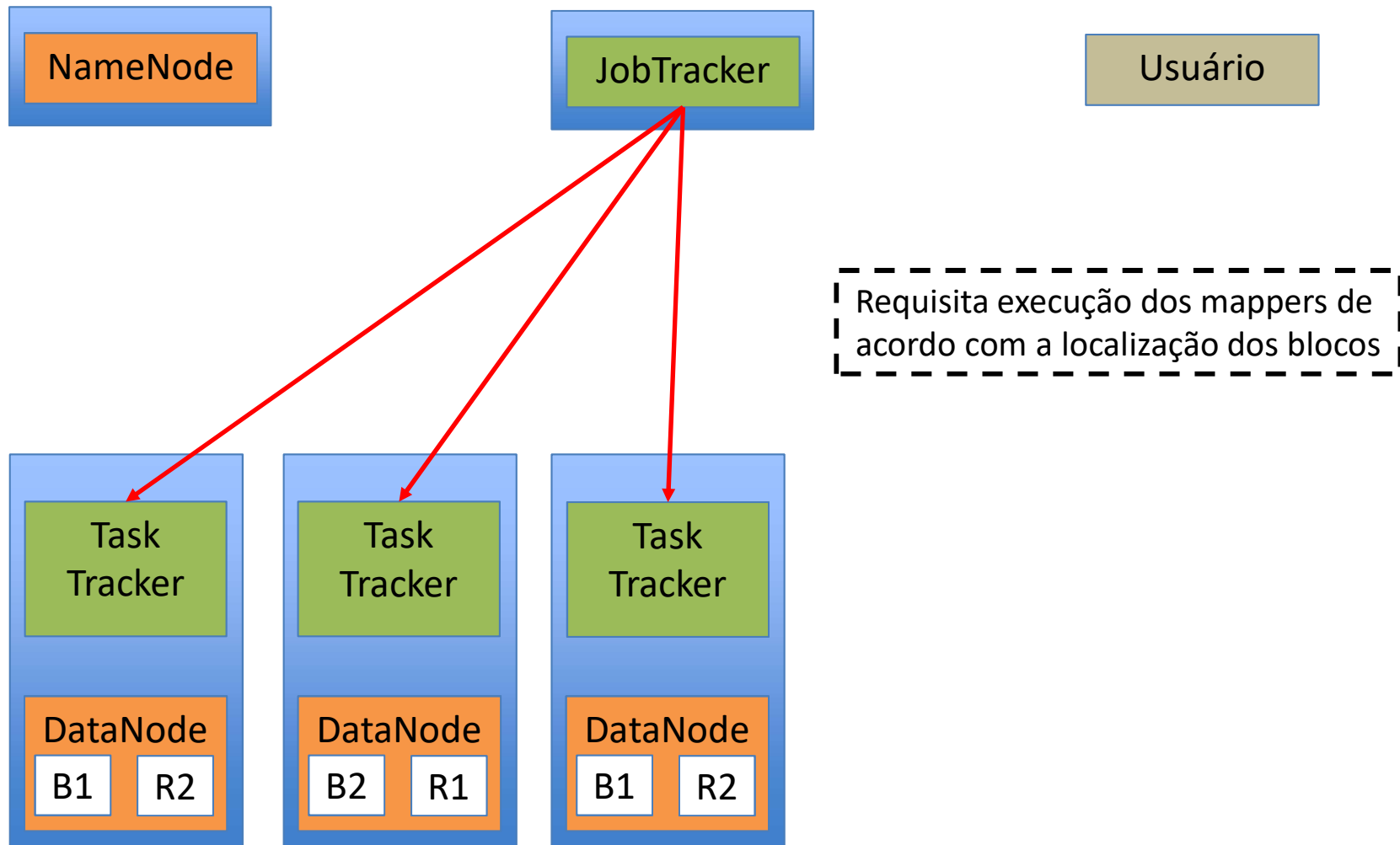




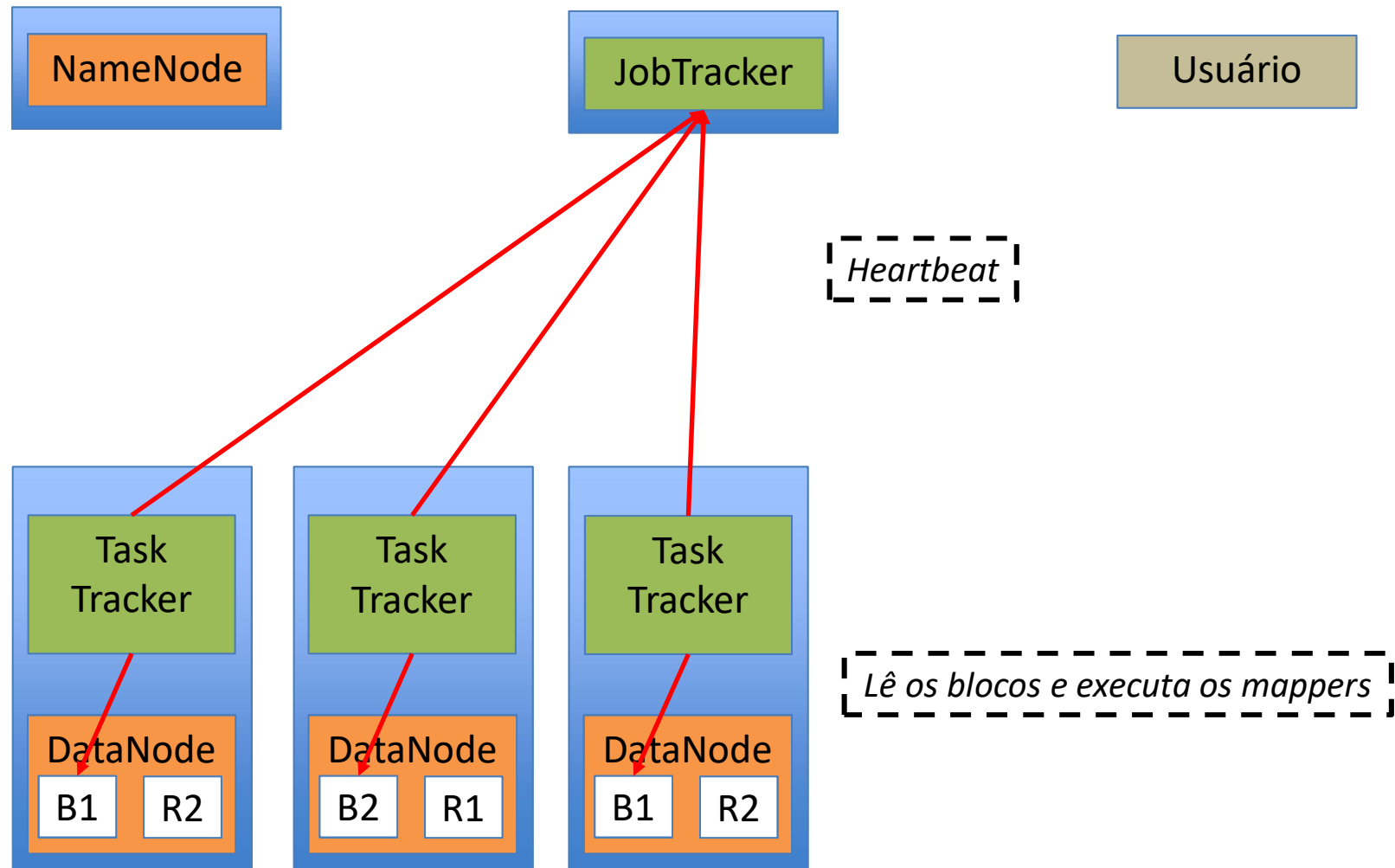
# MapReduce - Processo



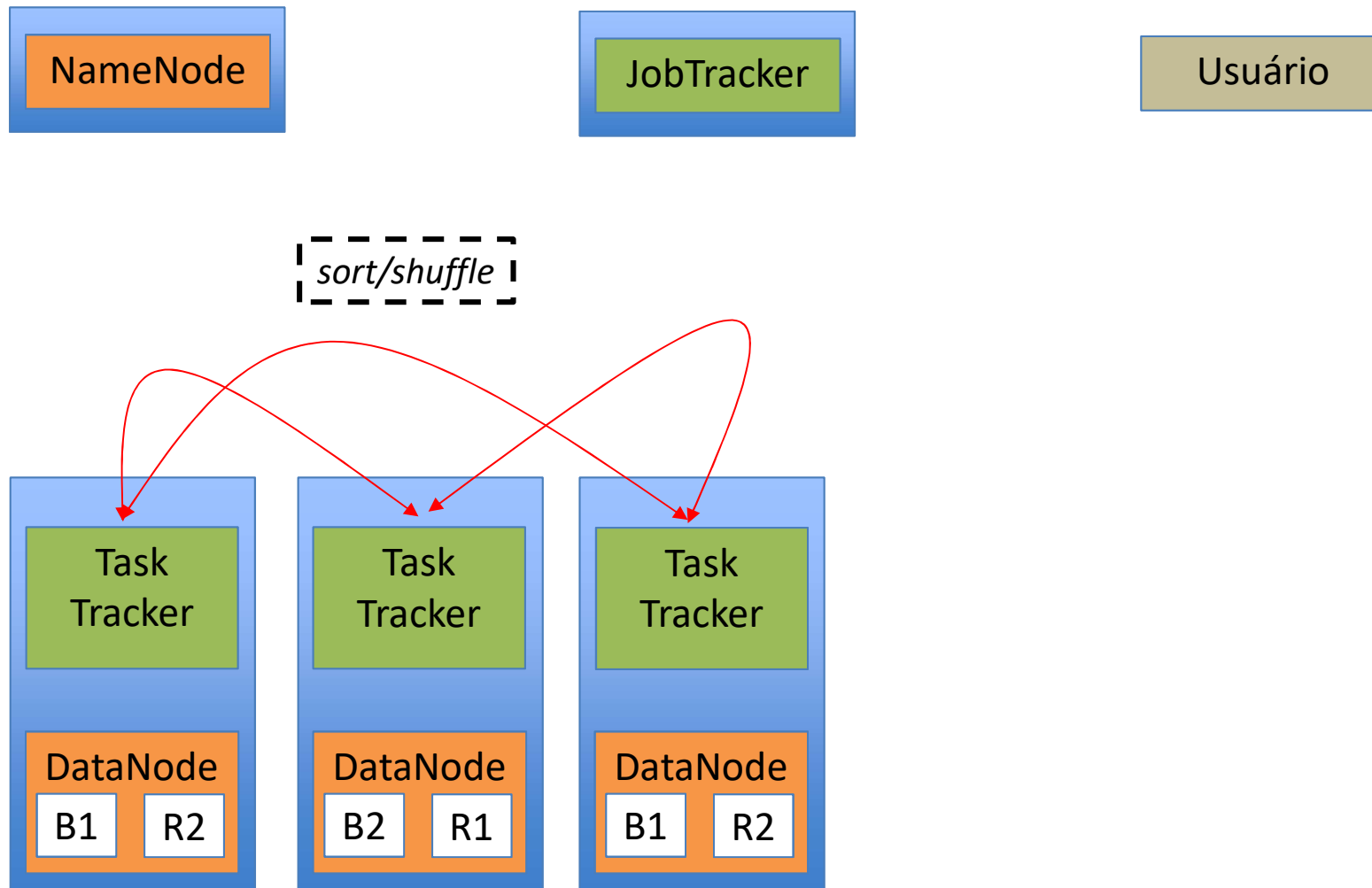
# MapReduce - Processo



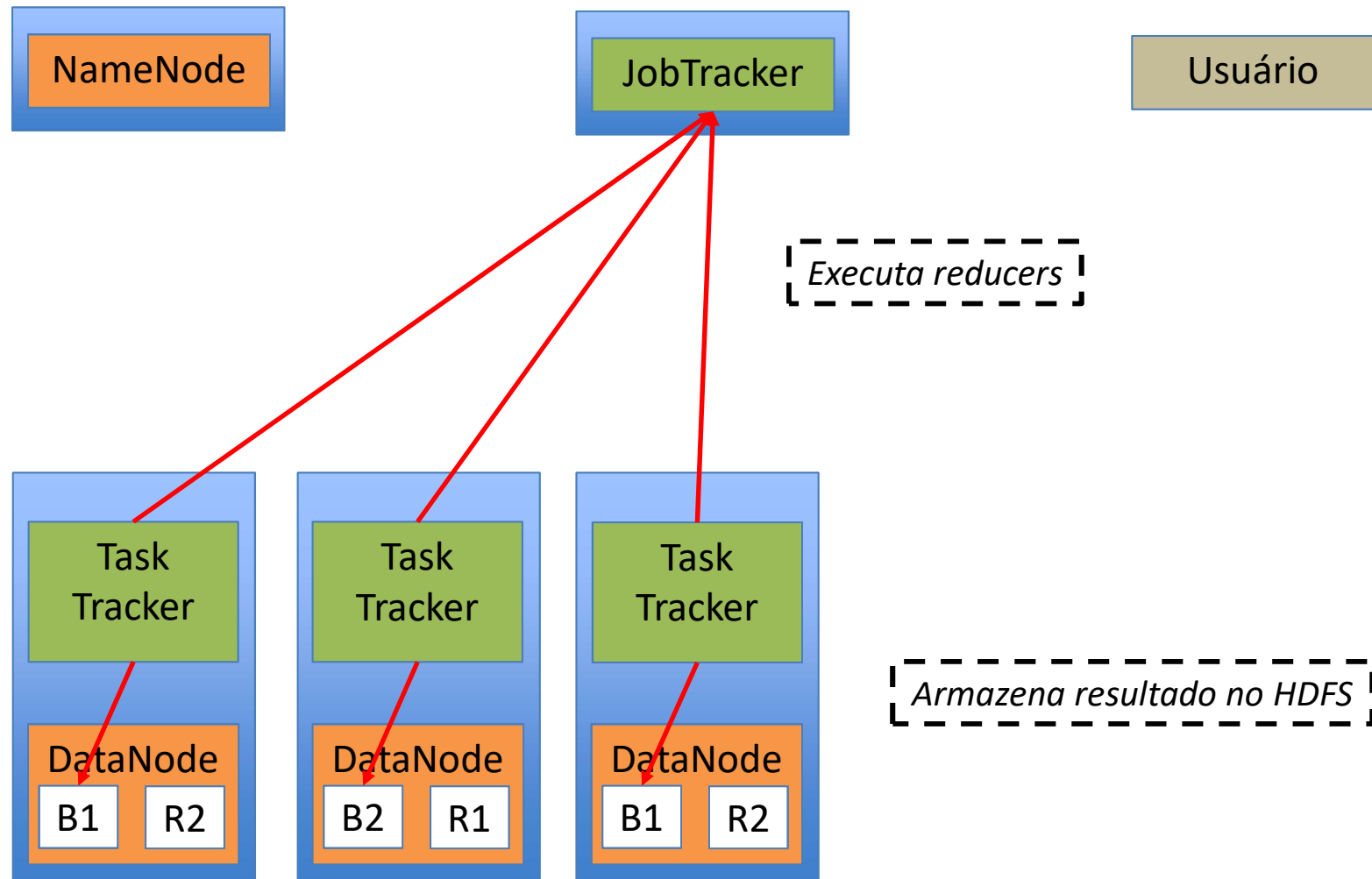
# MapReduce - Processo



# MapReduce - Processo

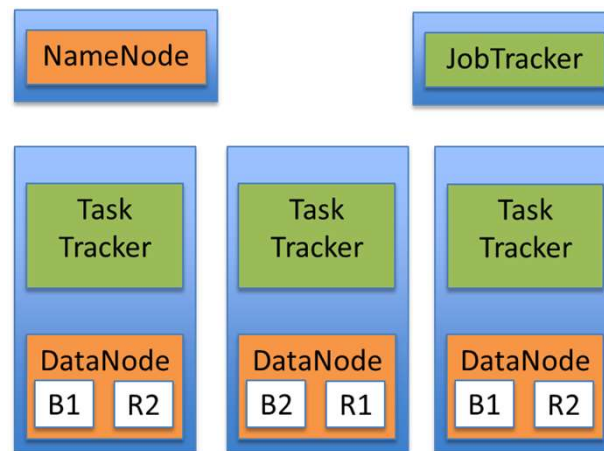


# MapReduce - Processo



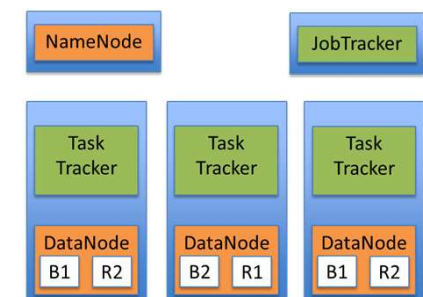
# MapReduce - Passos

- Para executarmos uma tarefa MapReduce em nosso cluster
  - Escrita da função de mapper
  - Escrita da função de reducer
  - Teste local
  - Cópia do código para o tasktracker
  - Execução no cluster



# MapReduce - Atividade

Campo	Descrição
País	País envolvido na transação comercial
Ano	Ano em que a transação foi efetuada
Código	Código da mercadoria
Mercadoria	Descrição da mercadoria
Fluxo	Fluxo, e.g. Exportação ou Importação
Valor	Valor em dólares
Peso	Peso da mercadoria
Unidade	Unidade de medida da mercadoria, e.g. Quantidade de itens
Quantidade	Quantidade conforme a unidade especificada da mercadoria
Categoria	Categoria da mercadoria, e.g. Produto Animal



# Agenda

- Big Data
- Hadoop
  - HDFS
  - MapReduce
  - **Apache Spark**



# Apache Spark

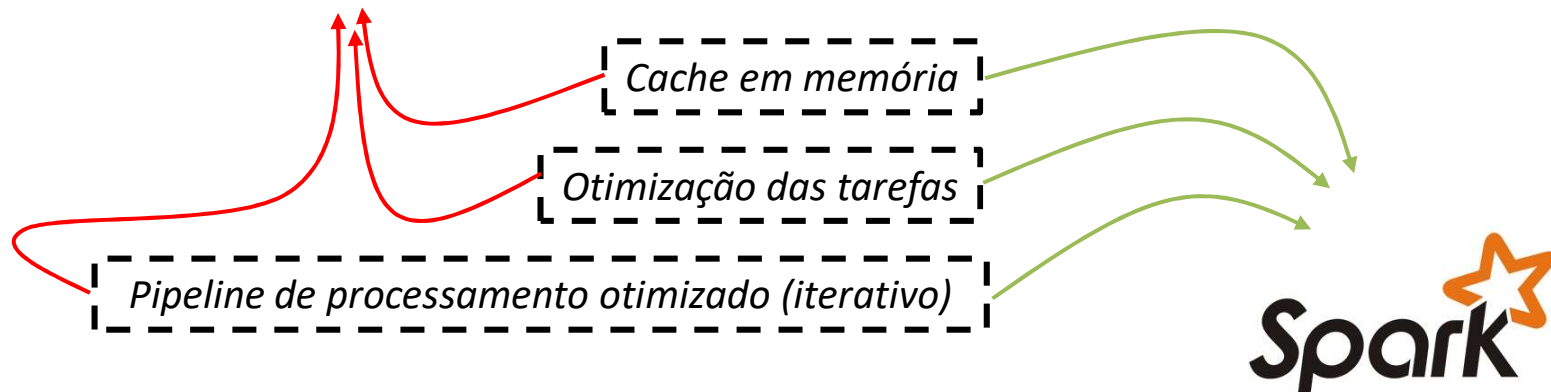
- Maioria dos algoritmos executam processos iterativos
  - A cada rodada os resultados são melhorados
  - E.g. *Machine Learning*
- ***Qual o problema da abordagem baseada em MapReduce para análise iterativa dos dados?***



- ***Escrita/Leitura em disco é *lenta!****

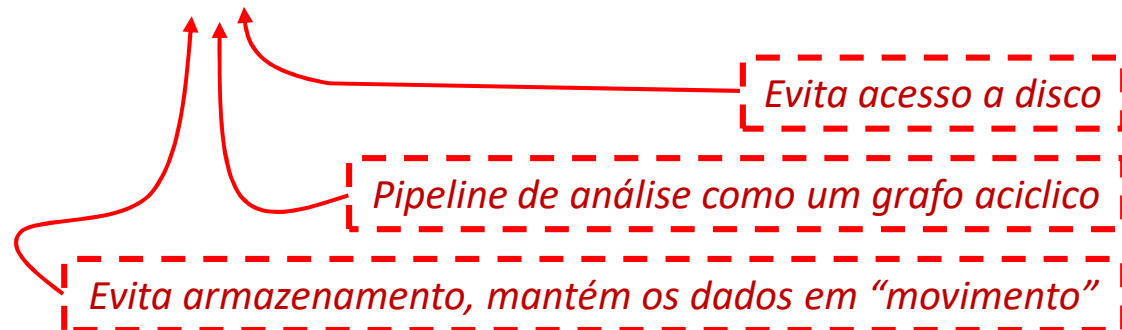
# Apache Spark

- Dificuldades no MapReduce
  - Programação
  - Gargalos de performance, especialmente relacionados a disco
    - Em média cada job escreve em disco 3 vezes
  - Modelo orientado a lote
  - Dificuldade para execução de processos iterativos



# Apache Spark

- Permite processamento **rápido** em cluster computacional de propósito geral



- Apache Spark
  - Integrado com ambiente do Hadoop
  - Permite leitura dos dados existentes no HDFS
  - APIs em Java, Scala, **Python**

# Apache Spark

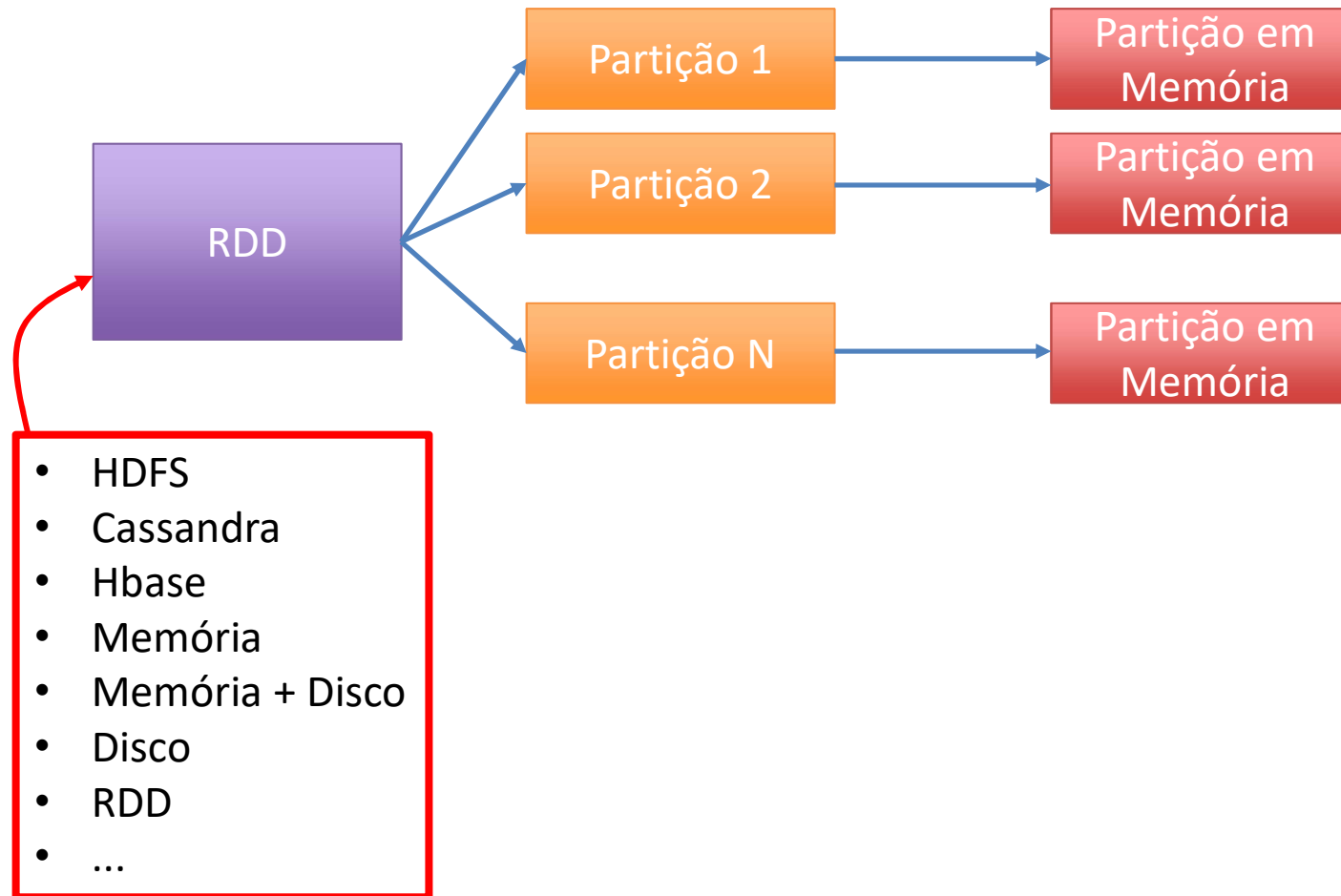
- Apache Spark opera sobre RDDs

*Resilient Distributed Dataset*

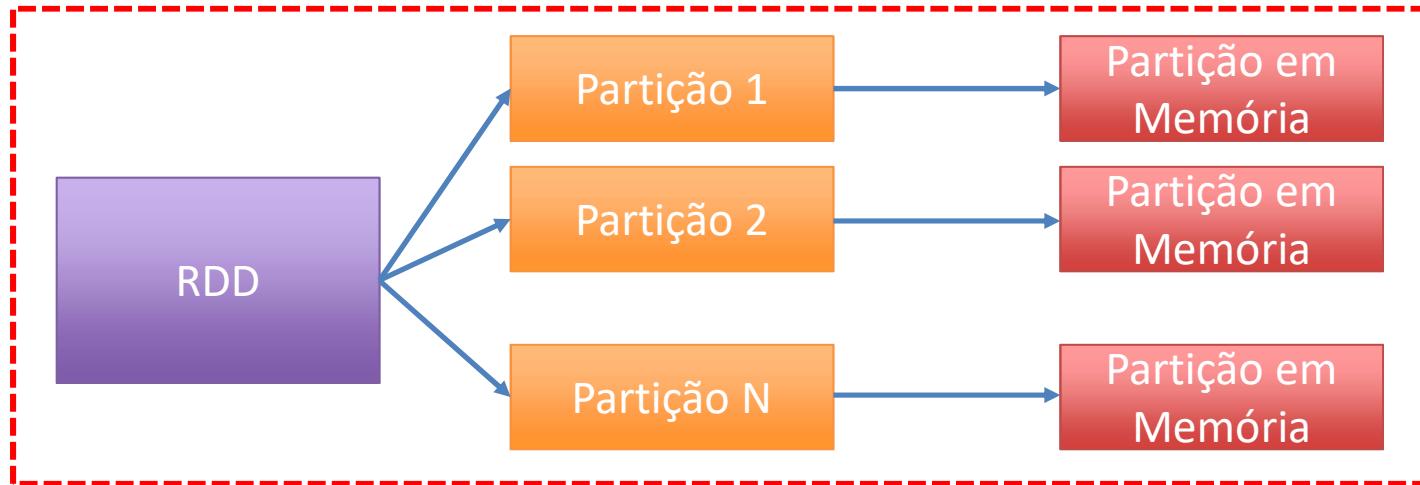


- Coleção de dados, divididos em partições, e armazenados nos *worker nodes* de um cluster
  - Em caso de falha, a partição é recalculada
- Interface para **transformar** os dados
- Abstração dos dados a serem analisados no Spark
  - HDFS
  - Cassandra
  - Hbase
  - Memória
  - Memória + Disco
  - Disco
  - RDD
  - ...

# Apache Spark



# Apache Spark



- Mantém referência aos dados das partições
- Cada partição possui um subconjunto dos dados
- Partições são atribuídas aos nós do cluster
- Framework armazena a partição em “memória”

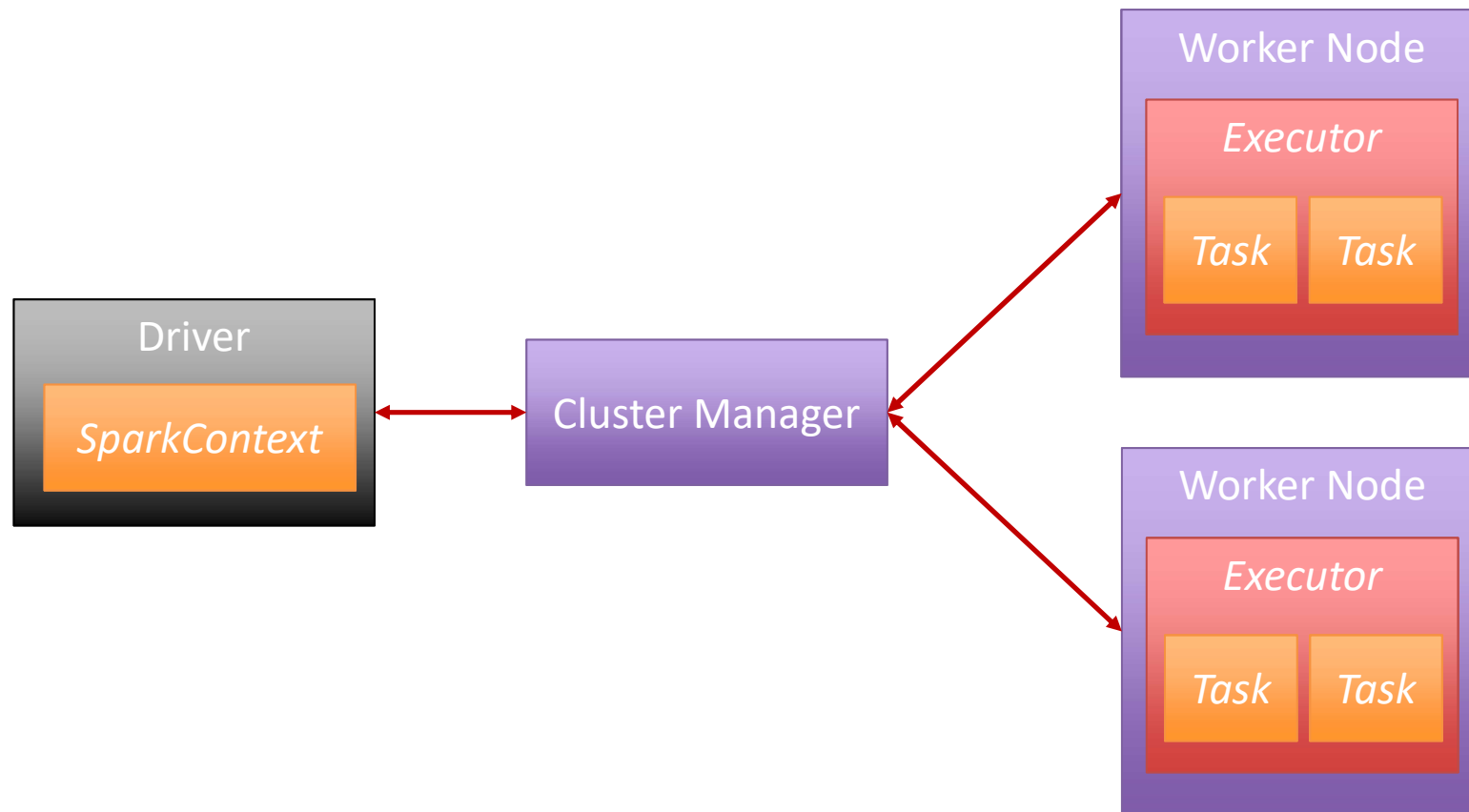
# Apache Spark

- Um RDD contém

- Conjunto de dependências de RDDs “pais”
  - Linhagem do RDD (obtido através do DAG)
- Conjunto de partições
  - Dados imutáveis
- Função para calculo do RDD, baseado em seu RDD “pai”
- Metadados sobre seu particionamento e localização dos dados

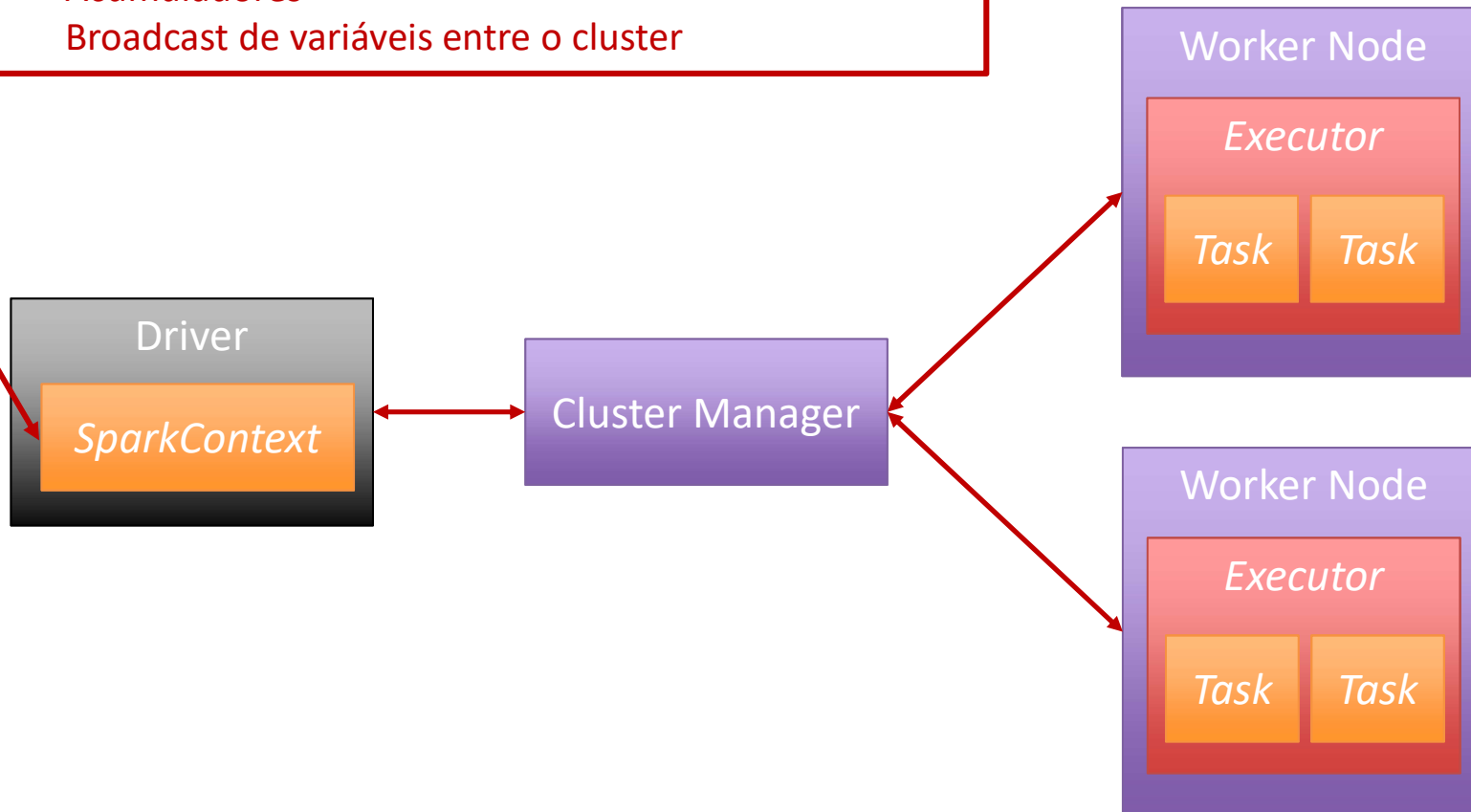
*Modelo de processamento do Apache Spark, veremos isso em breve =)*

- RDD visa tratar os “problemas” de manipulação de dados
- Para tratar o “problema” da análise iterativa dos dados o Apache Spark utiliza o processamento em DAG
  - *Direct Acyclic Graph*





Ponto de entrada para o cluster Spark  
Representa as conexões com o cluster  
Determina como e quando acessar o cluster  
Utilizado para  
Criar RDDs  
Acumuladores  
Broadcast de variáveis entre o cluster

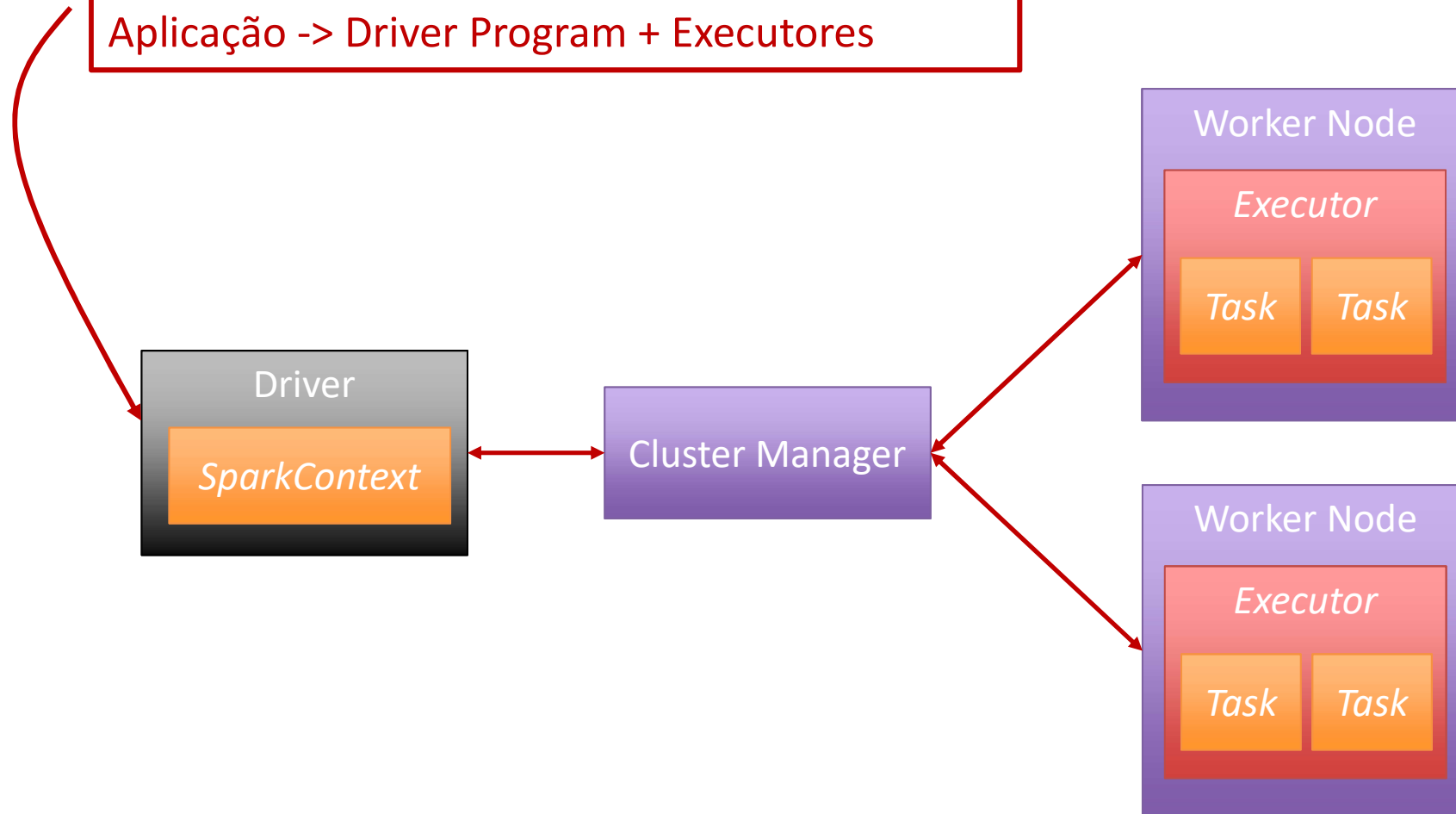


Processo principal coordenado pelo objeto do  
SparkContext

Permite configurar um processo spark

Ações do Spark são executadas no driver

Aplicação -> Driver Program + Executores



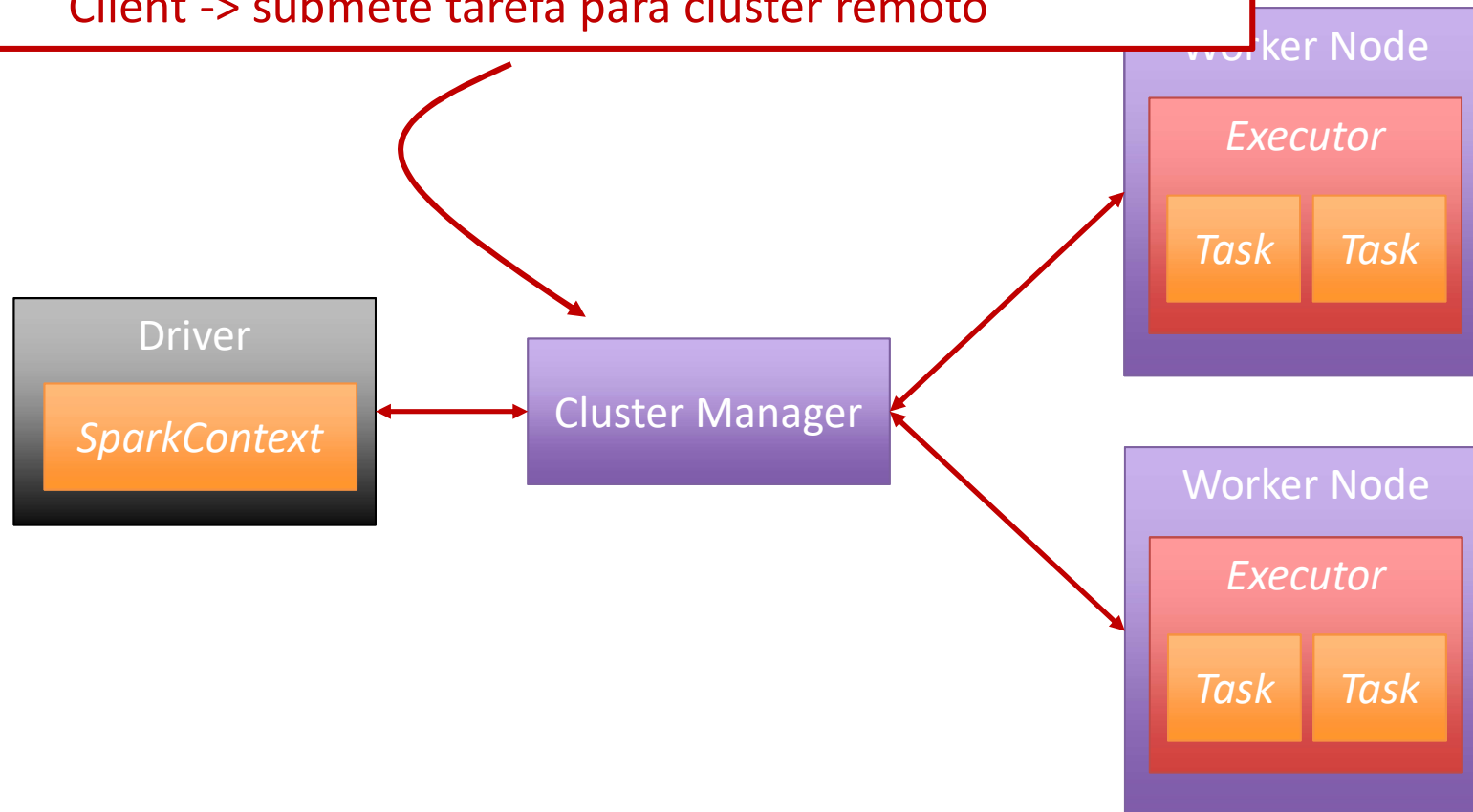
Serviço externo ao Apache Spark para receber recursos do cluster

Local, StandAlone, YARN, Mesos, ...

Modos de uso

Cluster -> utiliza o cluster para envio da tarefa

Client -> submete tarefa para cluster remoto

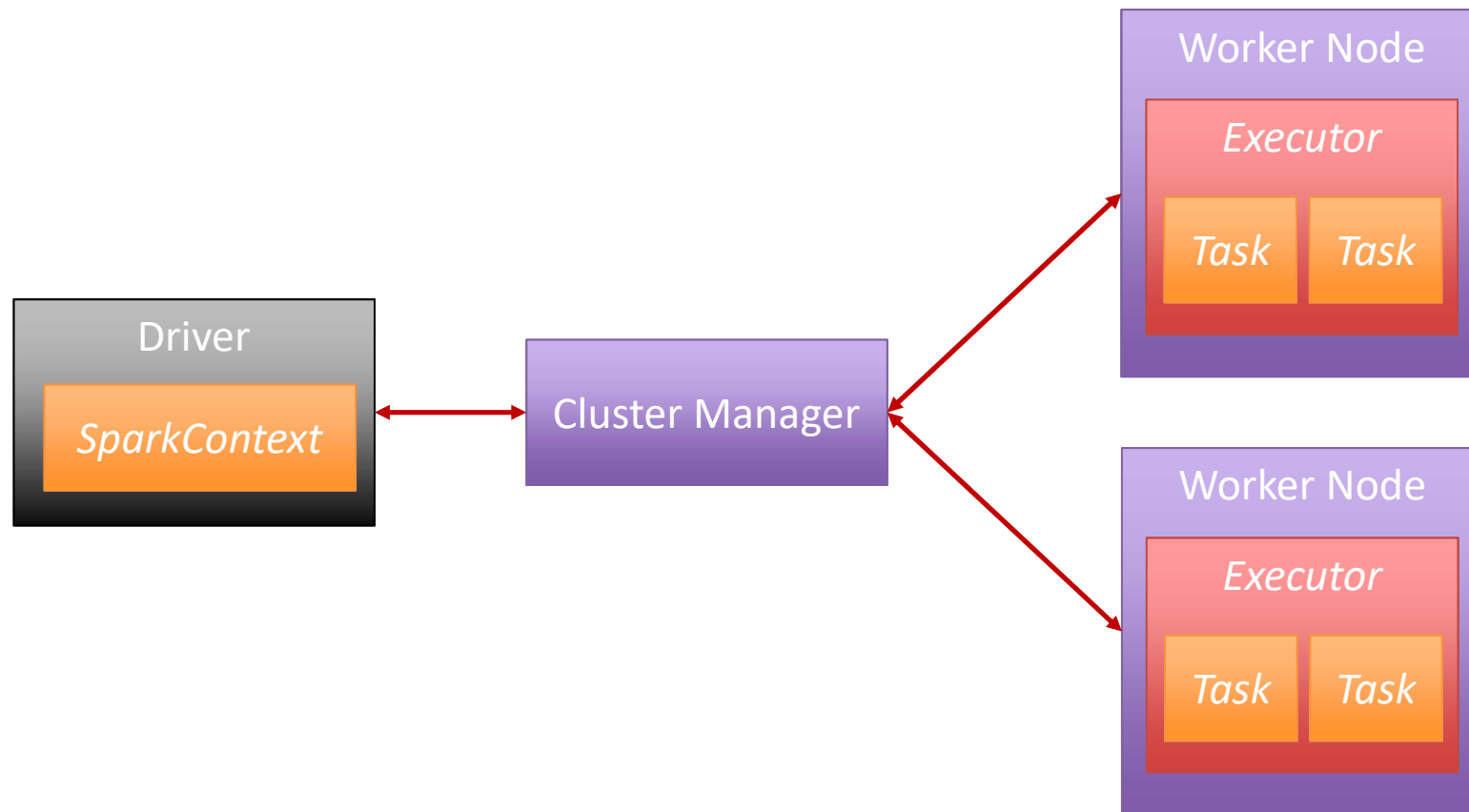


**Executor**, processo para execução da aplicação

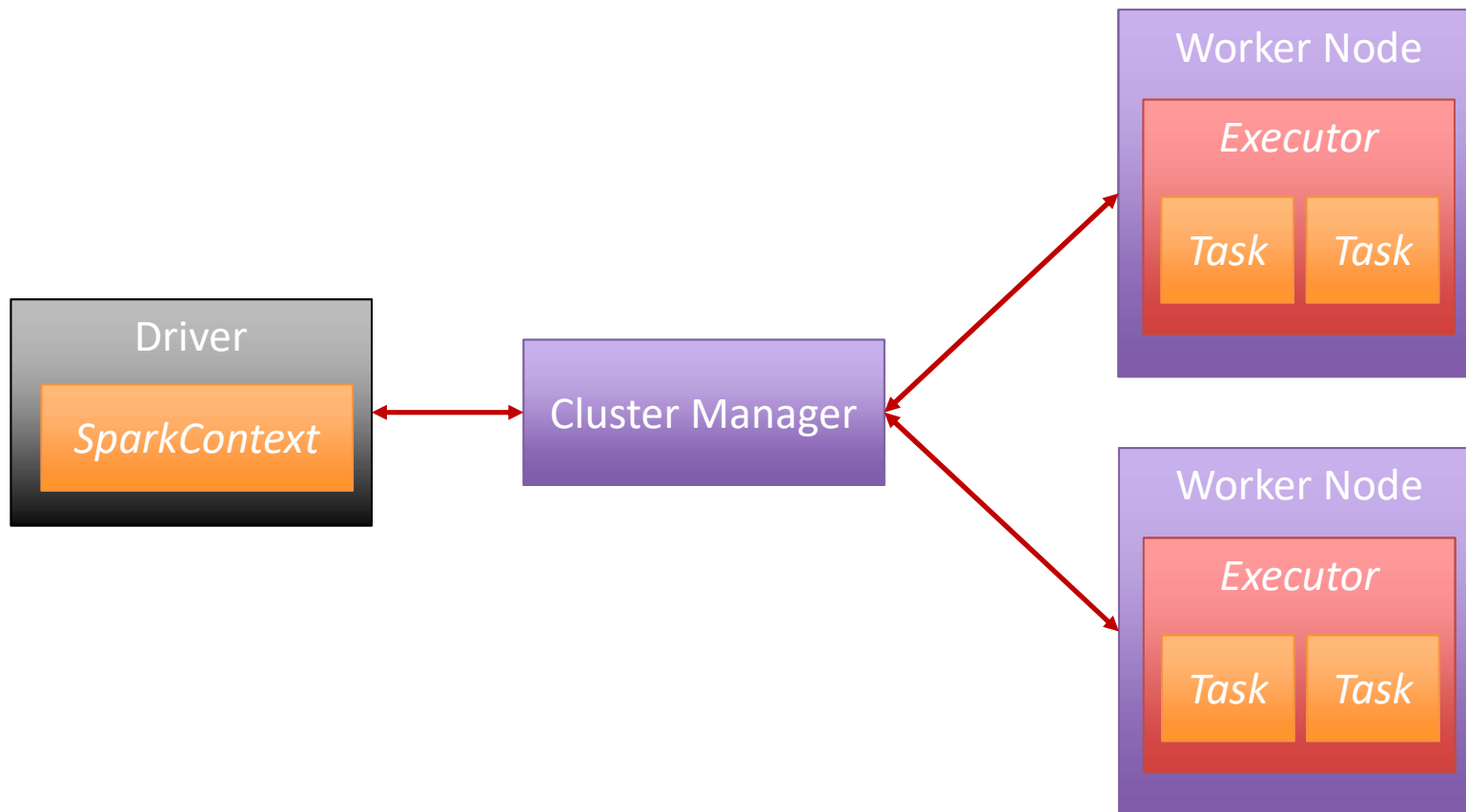
**Task**, unidade de trabalho que será enviada ao executor

**Job**, tarefa executada no cluster

Nó responsável pela execução das tarefas

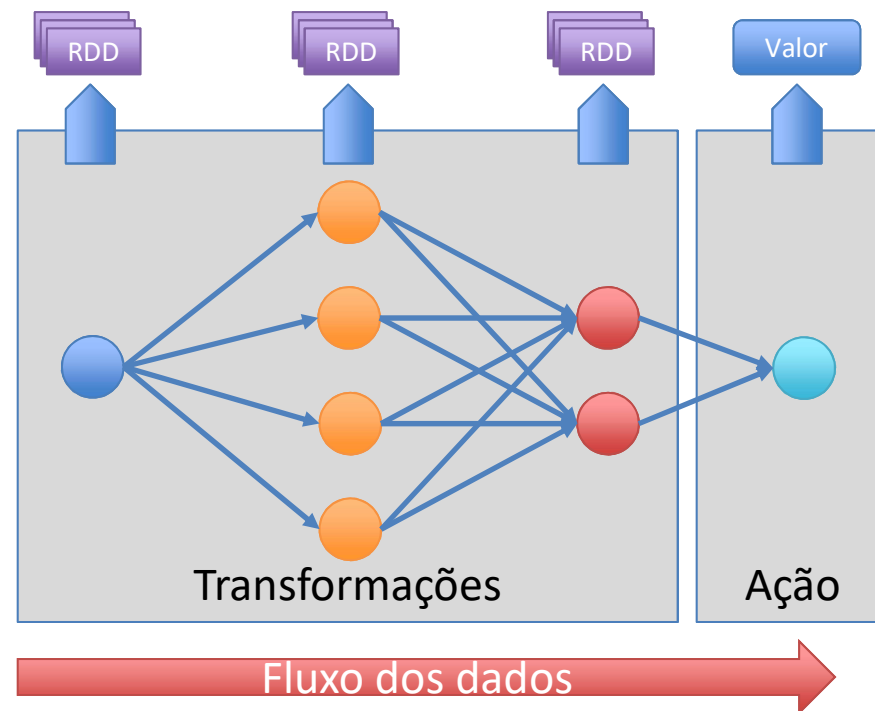


# Apache Spark



# Modelo de processamento

- Para tratar o “problema” da análise iterativa dos dados o Apache Spark utiliza o processamento em DAG
  - *Direct Acyclic Graph*

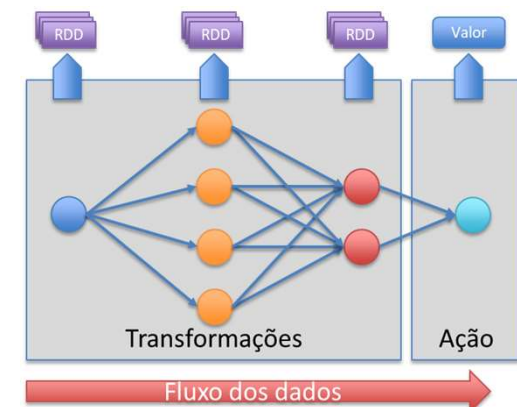


# Modelo de processamento

## ■ Transformações

- Cria um novo RDD a partir de um RDD existente
- *Lazy evaluated*, a execução só ocorre quando efetuamos uma ação
- *Exemplos:*
  - *MAP*
  - *FLATMAP*
  - *REDUCEBYKEY*
  - *FILTER*
  - *SAMPLE*
  - *DISTINCT*
  - *GROUPBYKEY*
  - ...

Lista completa: <https://spark.apache.org/docs/latest/rdd-programming-guide.html>

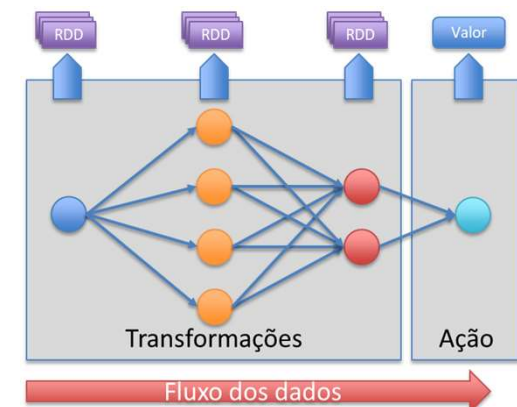


# Modelo de processamento

## ■ Ação

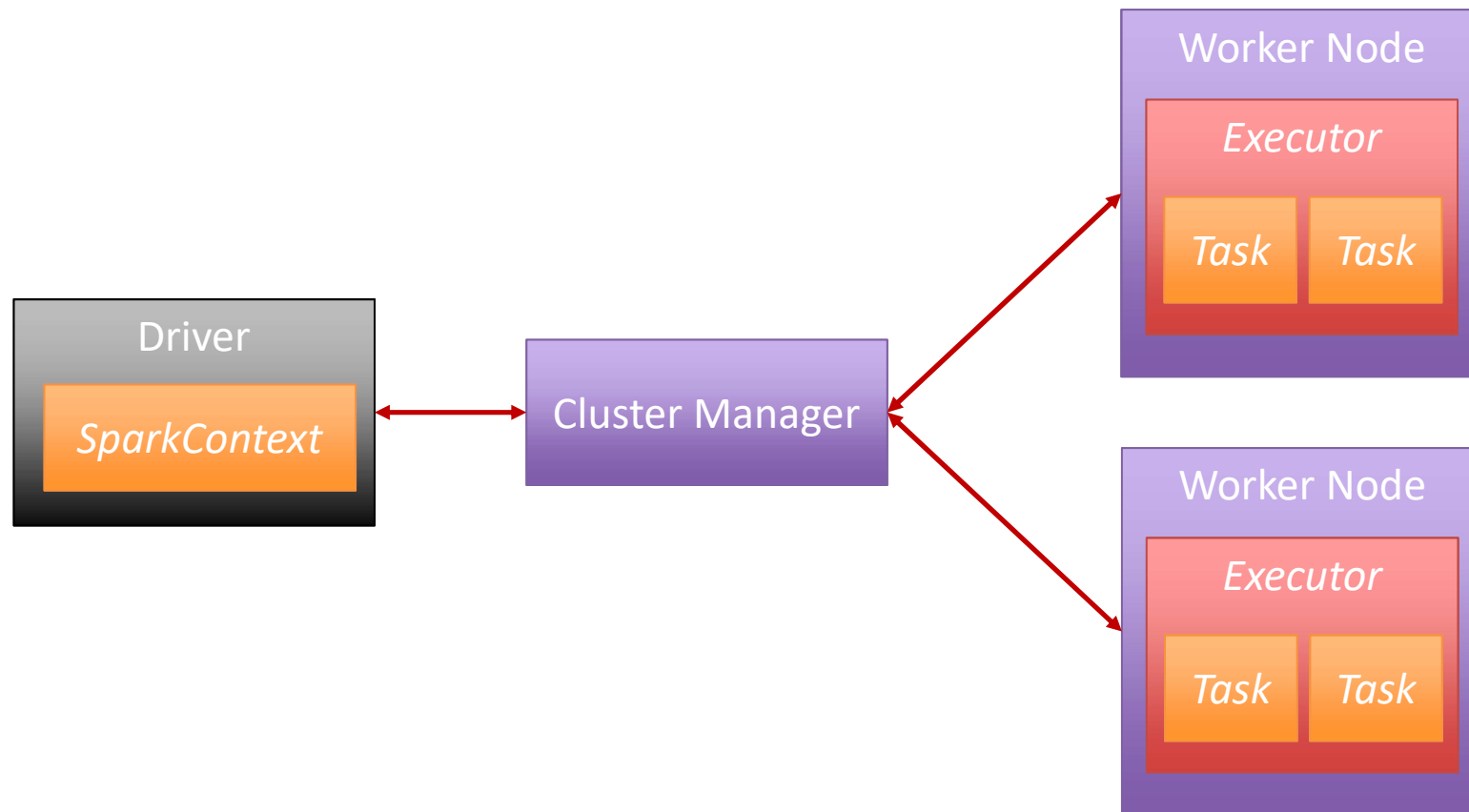
- Retorna um valor para o **driver** depois do calculo sobre o RDD
- *Exemplos:*
  - *COUNT*
  - *REDUCE*
  - *TAKE*
  - *COLLECT*
  - *FIRST*
  - ...

Lista completa: <https://spark.apache.org/docs/latest/rdd-programming-guide.html>



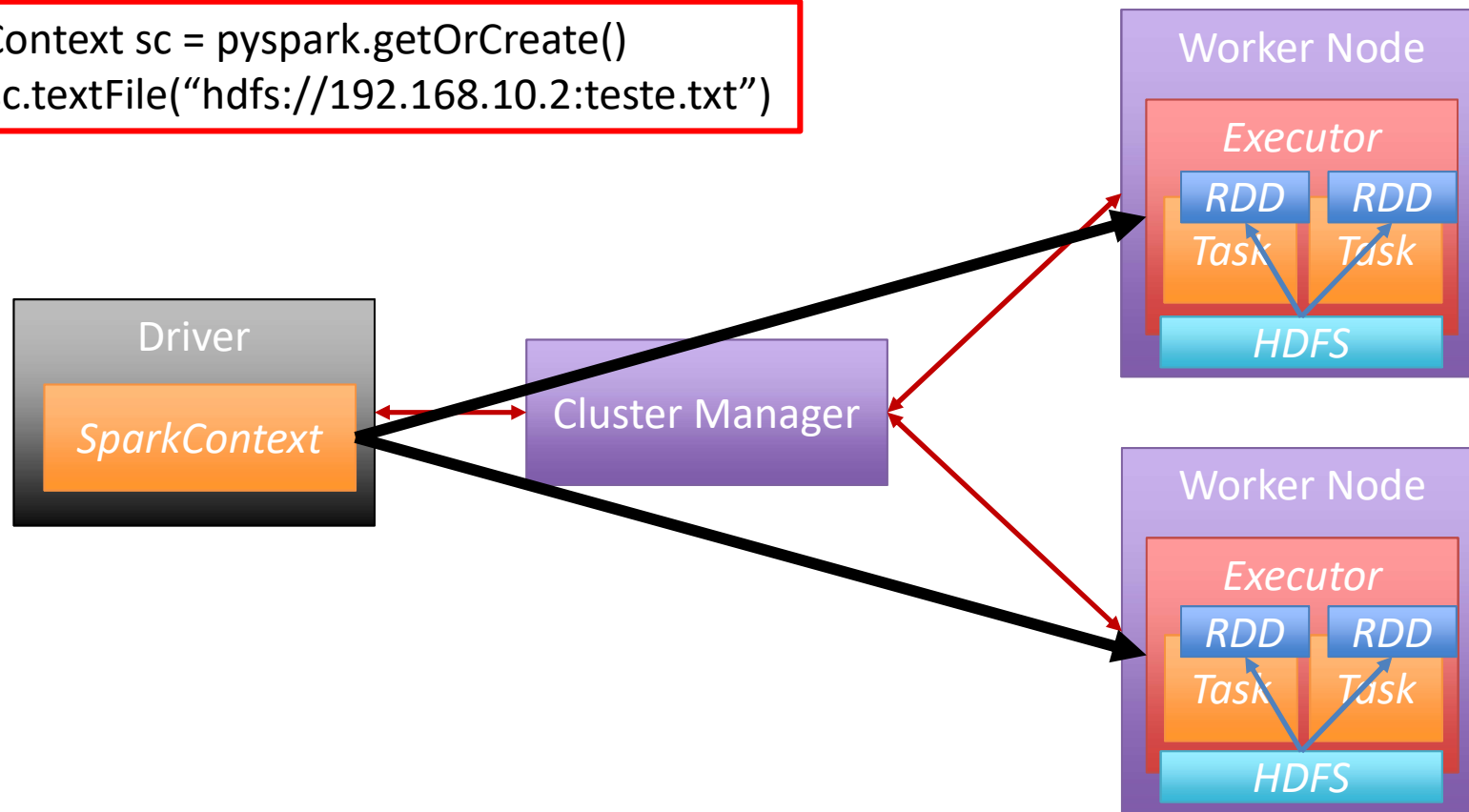


*Exemplo: filtrando um dataset para contar entradas com um valor específico*



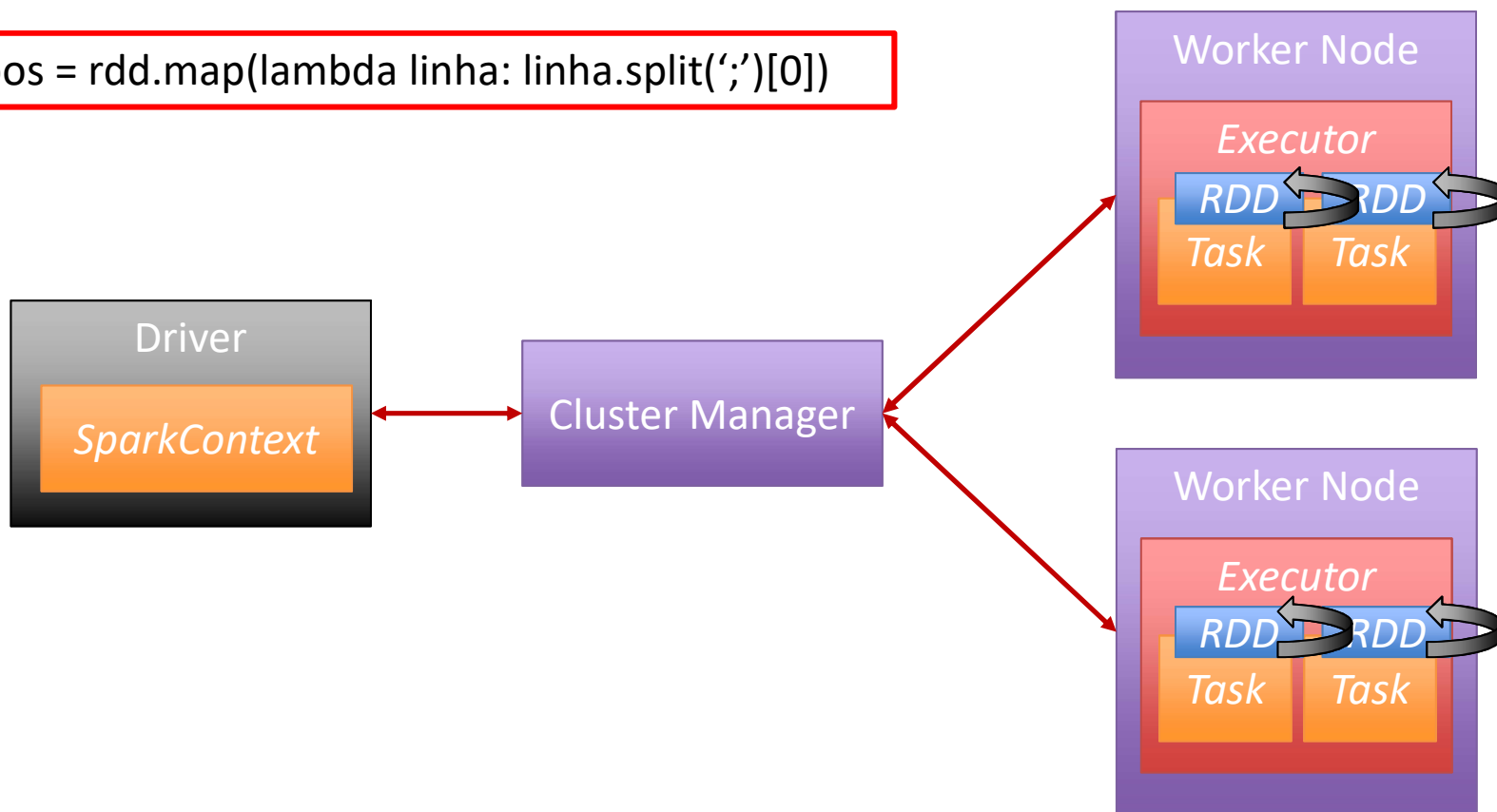
*Exemplo: filtrando um dataset para contar entradas com um valor específico*

```
SparkContext sc = pyspark.getOrCreate()  
rdd = sc.textFile("hdfs://192.168.10.2:teste.txt")
```



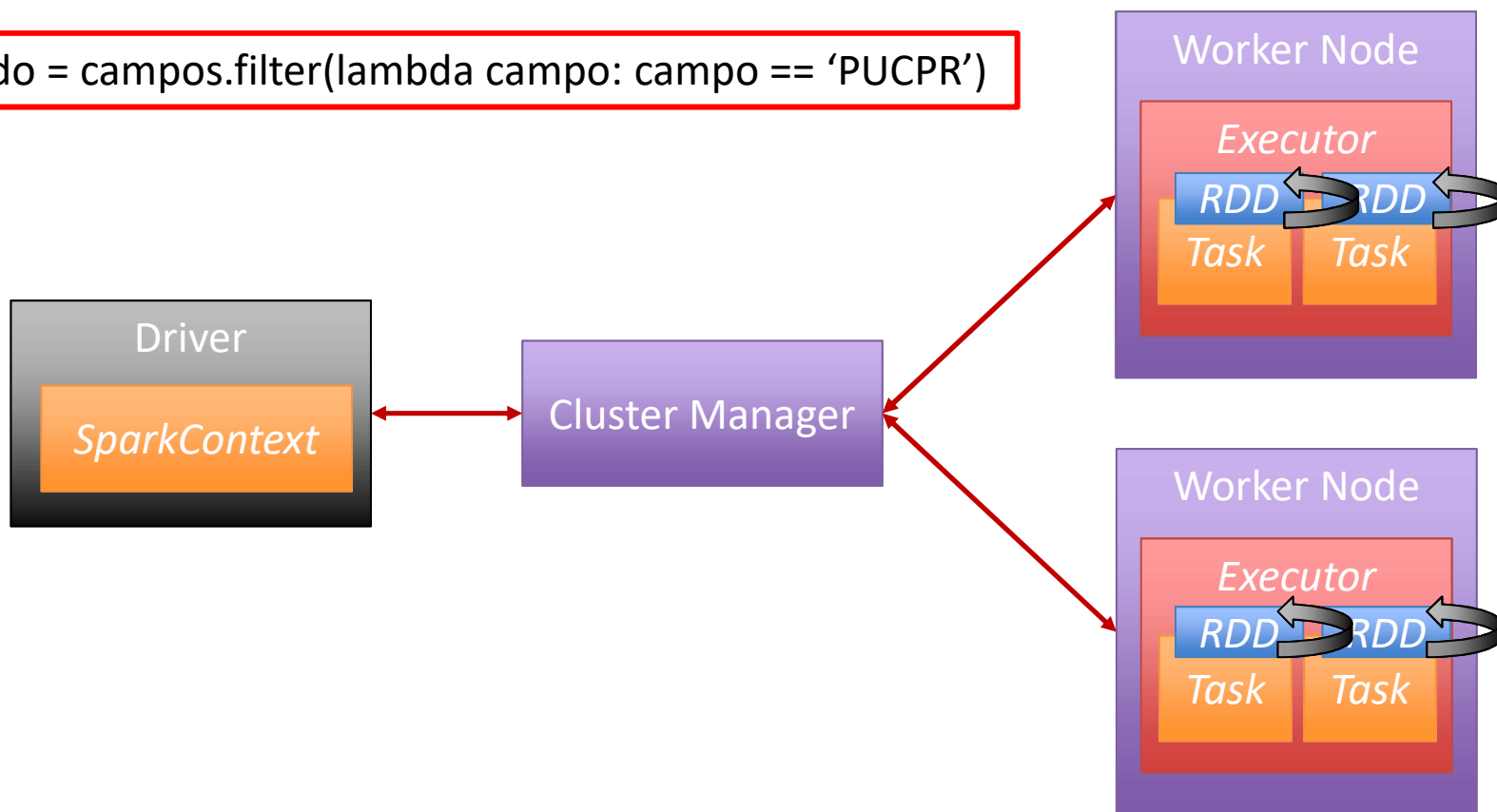
*Exemplo: filtrando um dataset para contar entradas com um valor específico*

```
campos = rdd.map(lambda linha: linha.split(';')[0])
```



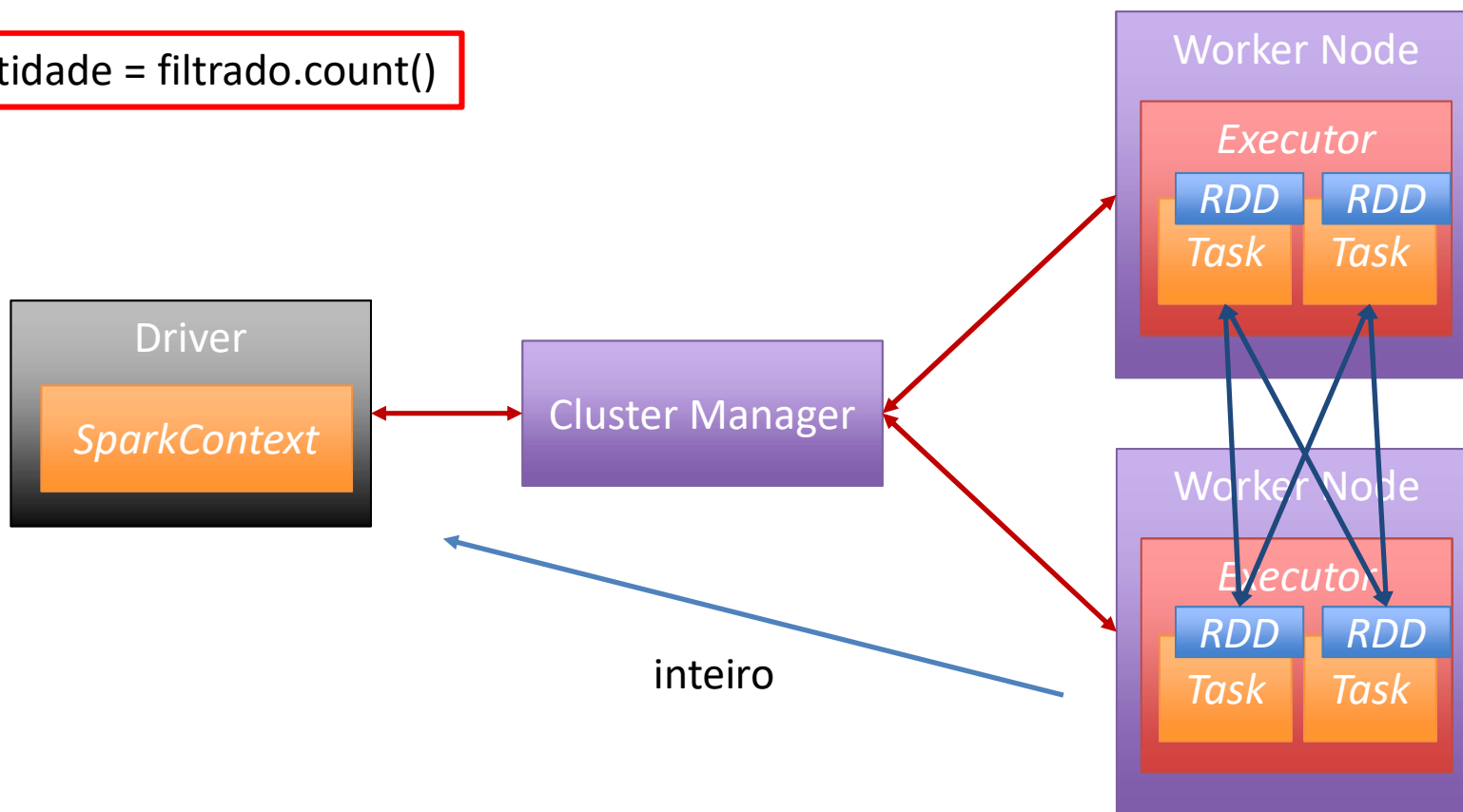
Exemplo: filtrando um dataset para contar entradas com um valor específico

```
filtrado = campos.filter(lambda campo: campo == 'PUCPR')
```

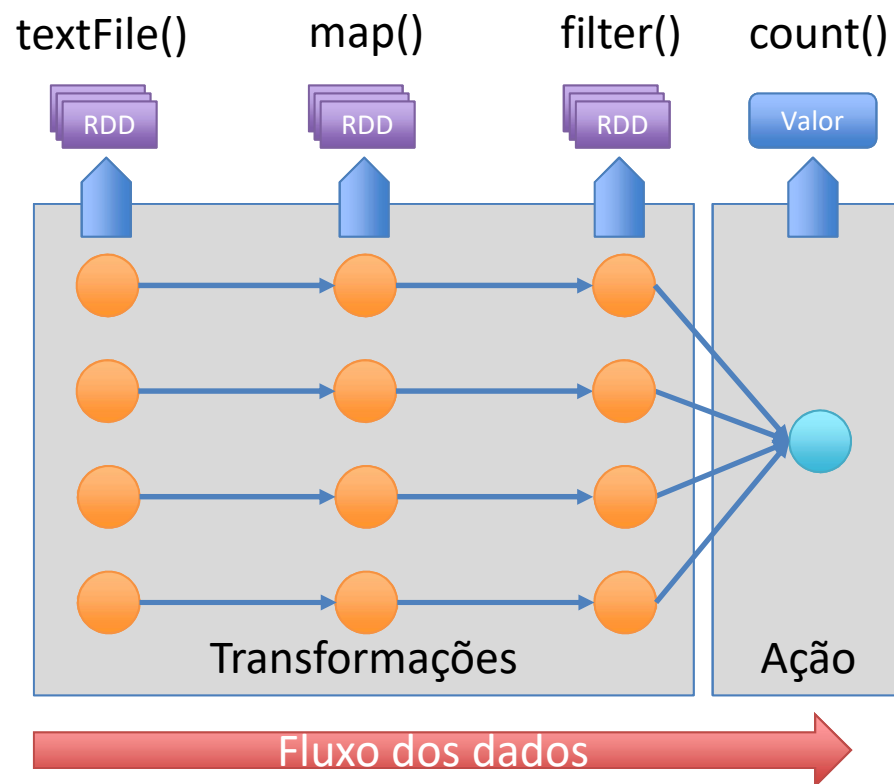


Exemplo: filtrando um dataset para contar entradas com um valor específico

```
quantidade = filtrado.count()
```



*Exemplo: filtrando um dataset para contar entradas com um valor específico*



# Principais transformações

Transformação	Descrição	Código
map	Gera RDD, executando a FUNC sobre todos elementos do RDD	<code>campos = arquivo.map(lambda x: x.split(";")[0])</code>
filter	Gera RDD com apenas os elementos que retornaram True através da FUNC	<code>campos.filter(lambda x: x == "Brazil")</code>
flatMap	Gera RDD, executando a FUNC sobre todos elementos do RDD, pode retornar 0 ou mais elementos	<pre>def mapFunc(x):     y = x.split(";")     return y arquivo.map(mapFunc)</pre>
sample	Gera um RDD com uma fração do RDD de entrada	<code>arquivo.sample(withReplacement = True, fraction = 0.1, seed = 1)</code>
distinct	Gera um RDD com elementos distintos	<code>arquivo.distinct()</code>
groupByKey	Gera um RDD com elementos agrupados pela FUNC, no formato (chave, (valores))	<code>arquivo.groupBy(lambda x: x.split(";")[0])</code>
reduceByKey	Gera um RDD com o formato (chave, valor), quando chamado sobre um RDD no formato (chave, (valores))	<code>arquivo.map(lambda x: [x.split(";")[0], 1]).reduceByKey(lambda x,y: x+y).collect()</code>
...		

Lista completa: <https://spark.apache.org/docs/latest/rdd-programming-guide.html>  
<https://spark.apache.org/docs/latest/api/python/pyspark.html#pyspark.RDD>

# Principais ações

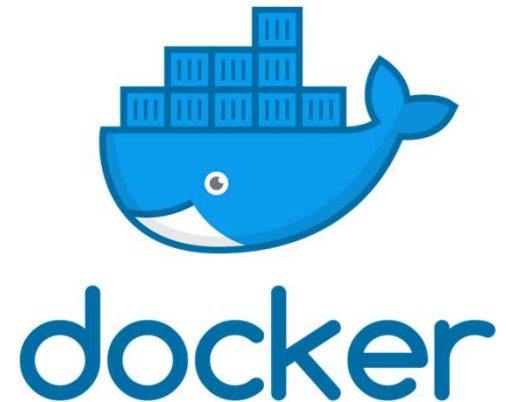
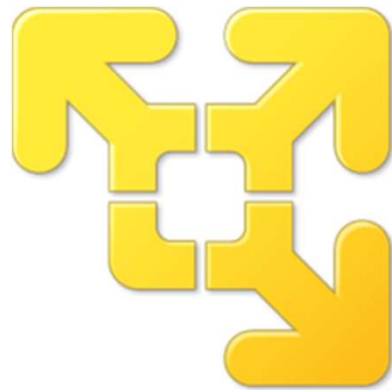
Ação	Descrição	Código
reduce	Agrega todos elementos do RDD através da FUNC, gerando um único valor	<code>arquivo.map(lambda x: 1).reduce(lambda x,y: x+y)</code>
collect	Retorna todos elementos para o driver	<code>arquivo.collect()</code>
count	Retorna um inteiro com a quantidade de elementos no RDD	<code>arquivo.count()</code>
first	Retorna o primeiro elemento do RDD	<code>arquivo.first()</code>
take	Retorna N elementos do RDD	<code>arquivo.take(10)</code>
saveAsText	Salva os elementos do RDD no arquivo especificado	<code>arquivo.saveAsTextFile("hdfs://172.18.0.11:9000/base2")</code>
...		

Lista completa: <https://spark.apache.org/docs/latest/rdd-programming-guide.html>  
<https://spark.apache.org/docs/latest/api/python/pyspark.html#pyspark.RDD>



# Prática

- Liguem a máquina virtual
  - Usuário: **docker**
  - Senha: **docker**



- Vamos ligar o nosso “cluster” hadoop agora

# Apache Spark - Implementação

- *Vamos acessar o jupyter com o pyspark para implementação das soluções*

SparkContext, “Driver”

Atualize o IP para o spark-master =)

```
import os
os.environ['PYSPARK_PYTHON'] = '/usr/bin/python3'

import pyspark
conf = pyspark.SparkConf()

conf.setMaster("spark://172.24.0.10:7077")
conf.set("spark.executor.memory", "1g")

sc = pyspark.SparkContext.getOrCreate()
sc.stop()
sc = pyspark.SparkContext(conf=conf)
```

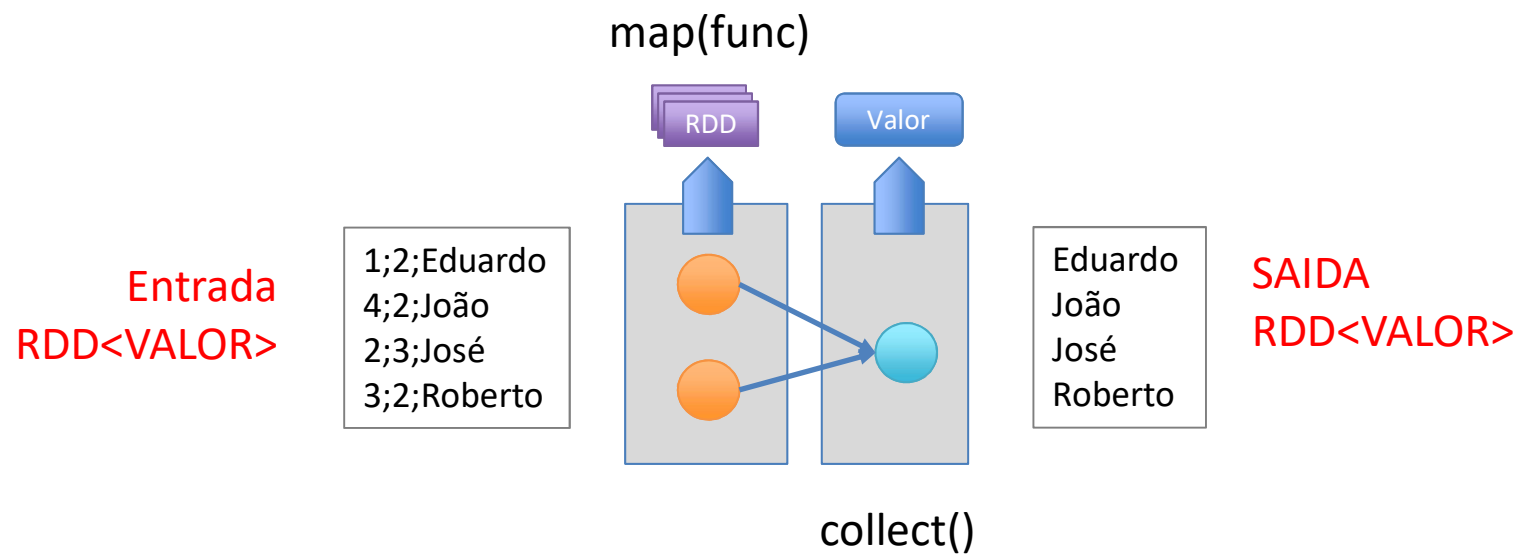
Atualize o IP para o namenode =)

```
arquivo = sc.textFile("hdfs://172.24.0.12:9000/base.csv")
```

## Exemplo: transformações

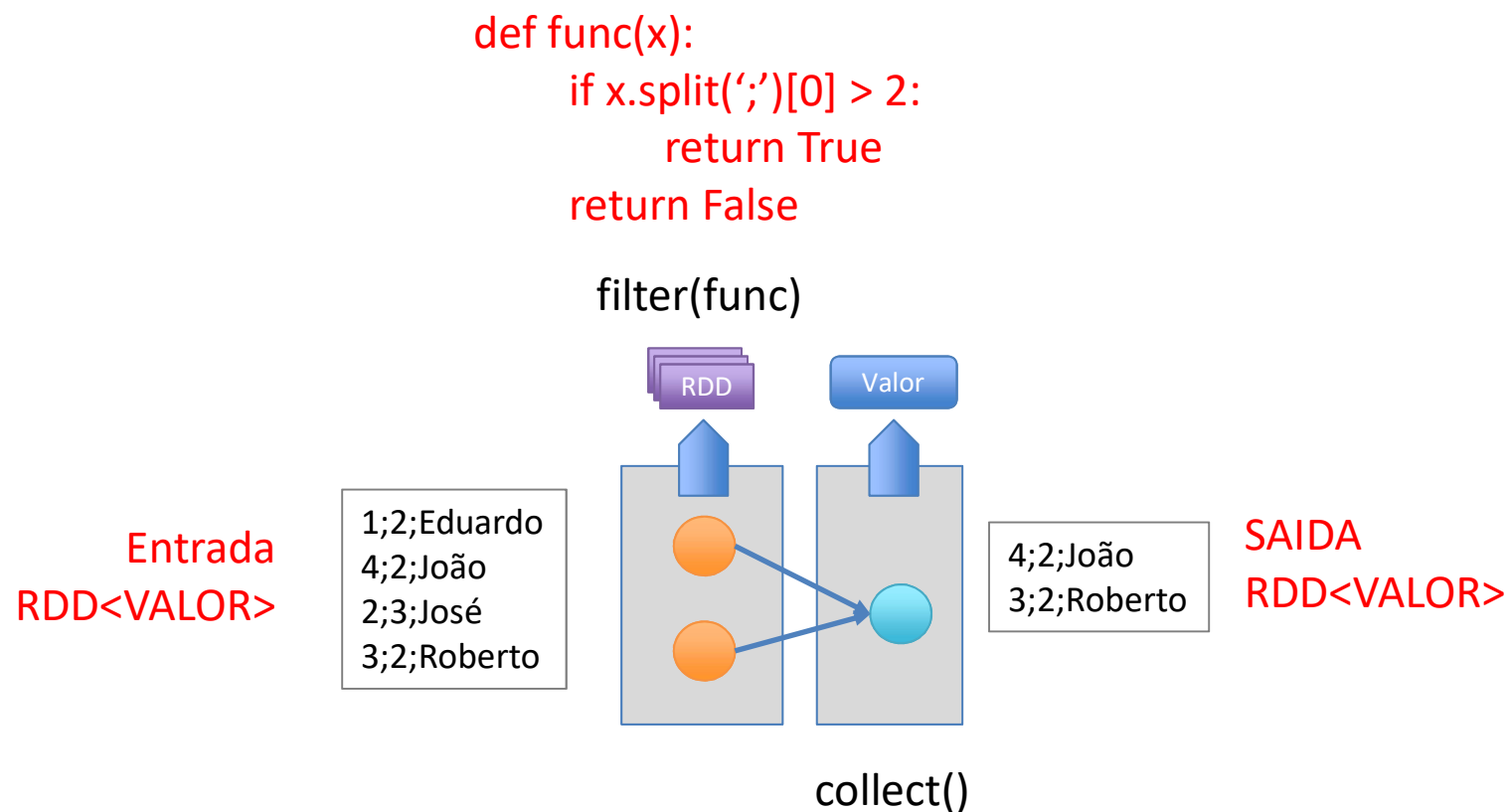
MAP: Executa a função sobre todo valor de entrada, deve gerar um valor de saída

```
def func(x):  
    return x.split(';')[2]
```



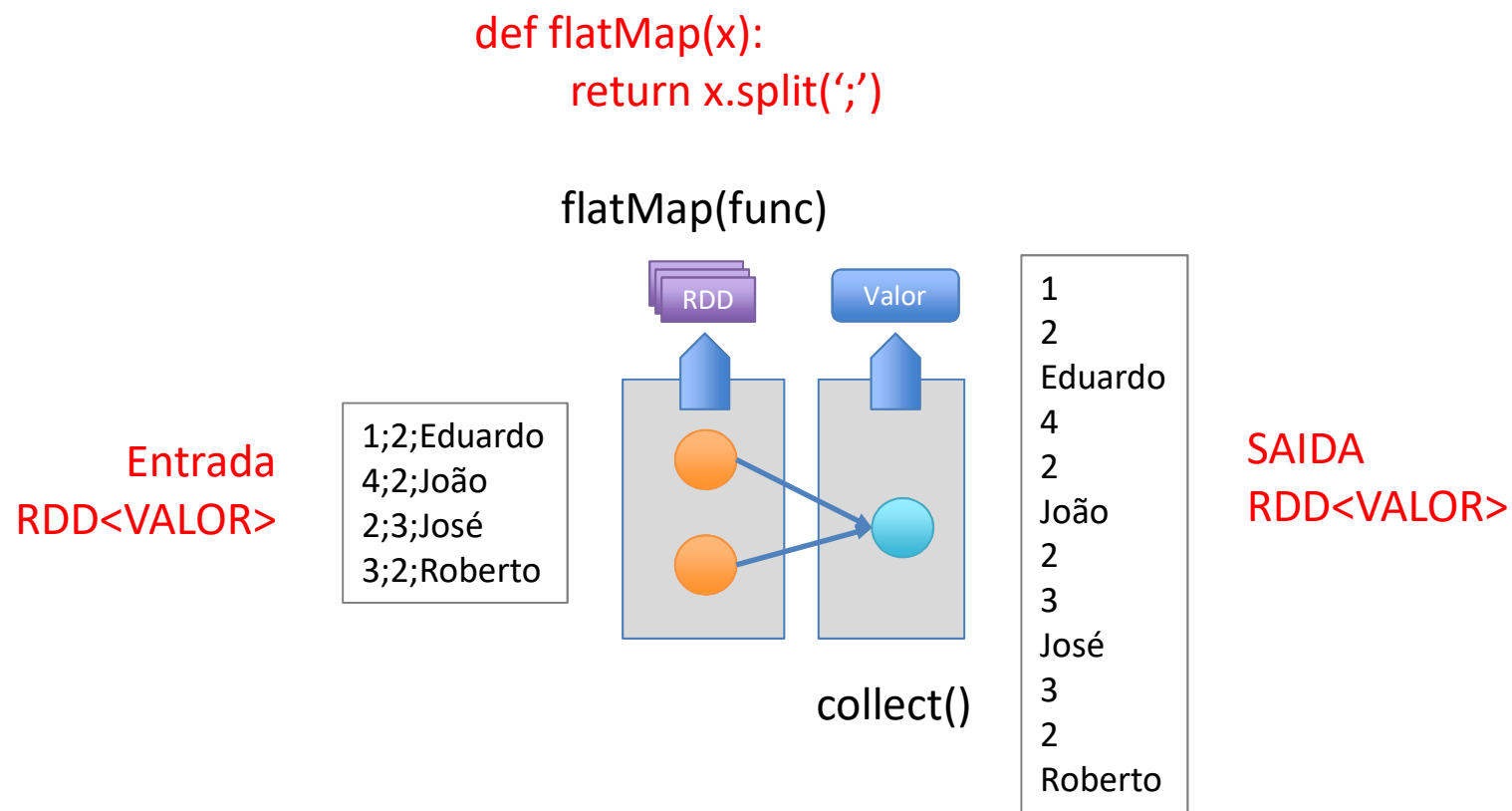
## Exemplo: transformações

Filter: Executa a função sobre todo valor de entrada, gera um RDD apenas com valores que retornaram True



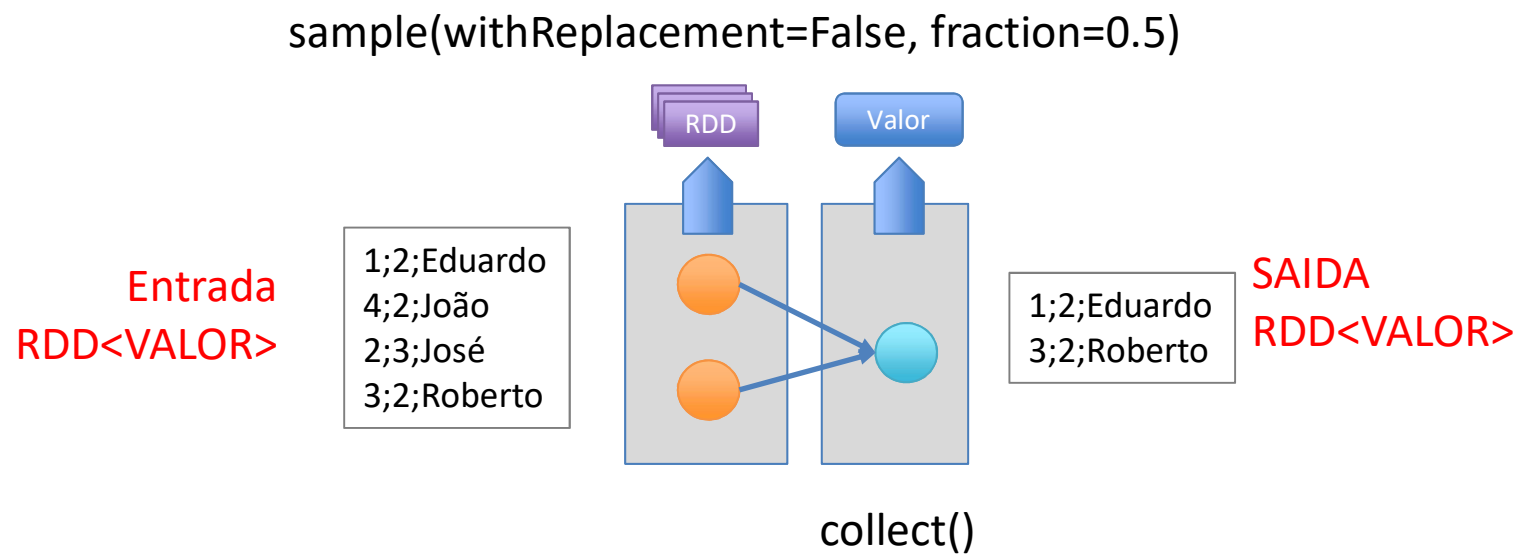
## Exemplo: transformações

Filter: Executa a função sobre todo valor de entrada, pode gerar 0 ou mais valores de saída



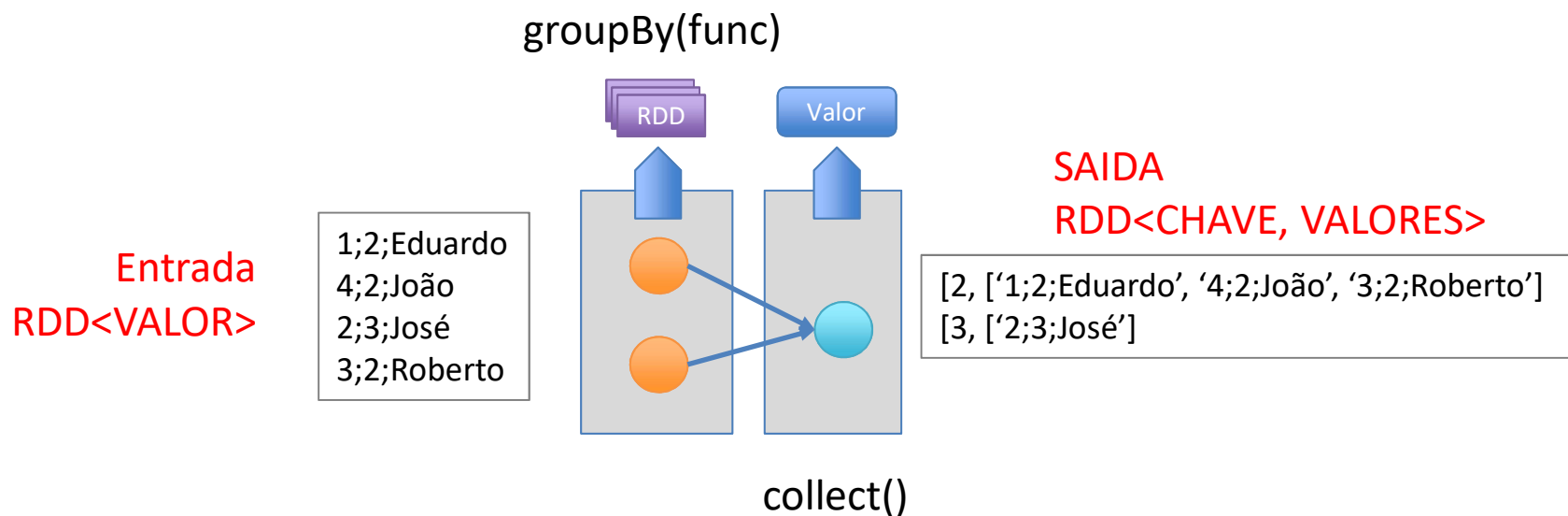
## Exemplo: transformações

SAMPLE: Gera um RDD com fraction % do RDD de entrada

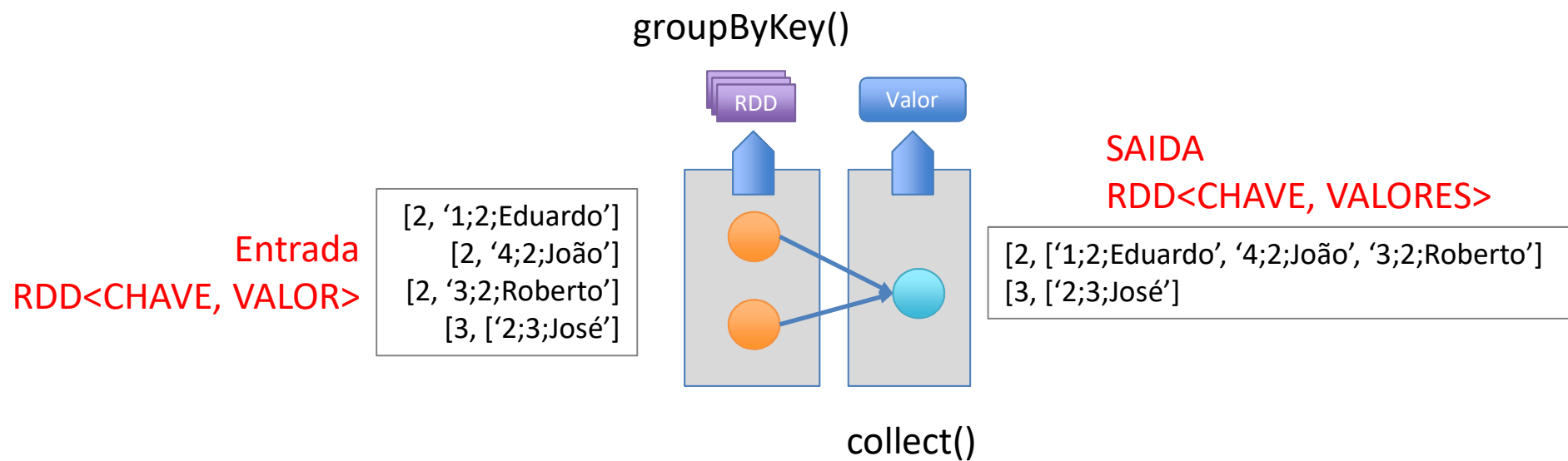


GROUPBY: Agrupa os valores de acordo com o valor retornado pela função

```
def func(x):  
    return x.split(';')[1]
```



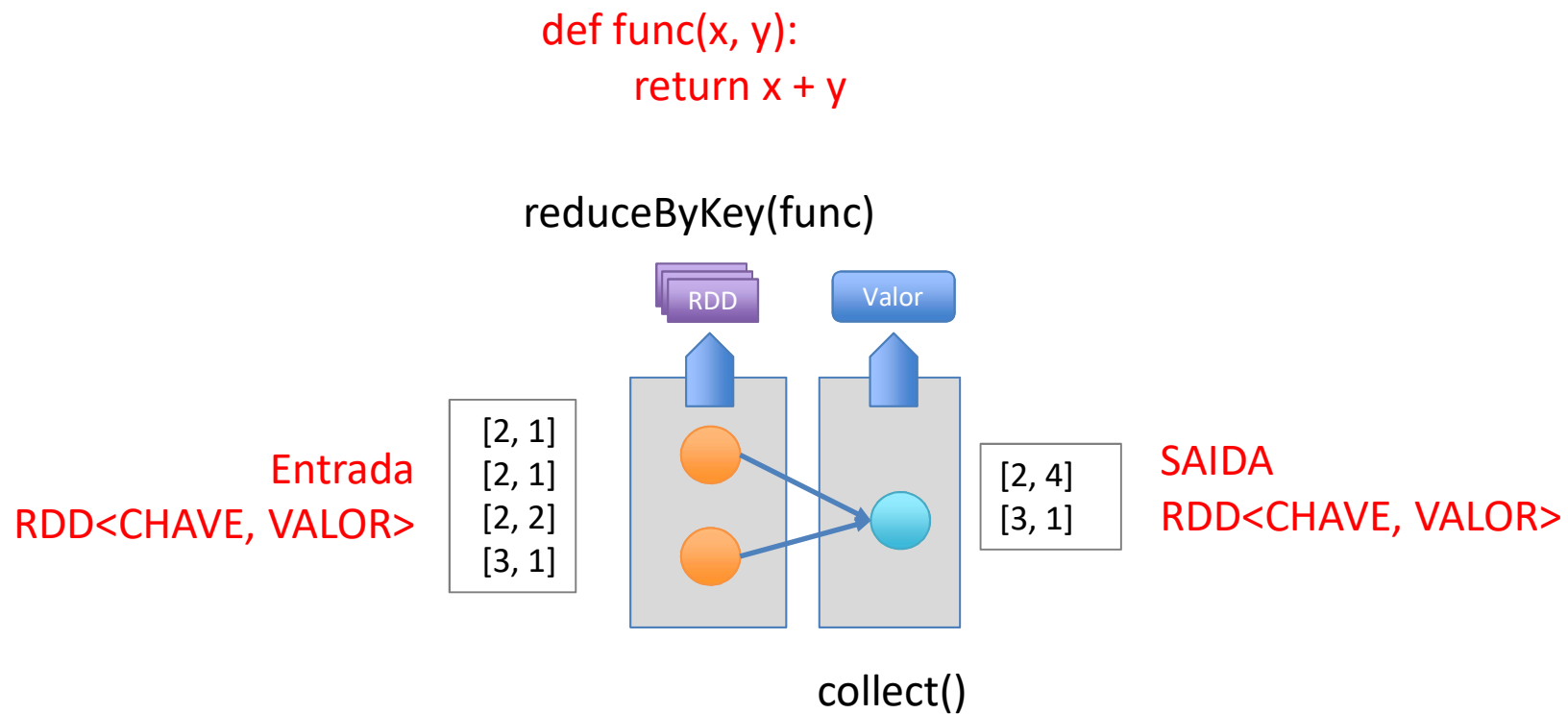
GROUPBYKEY: Agrupa os valores de acordo com a chave



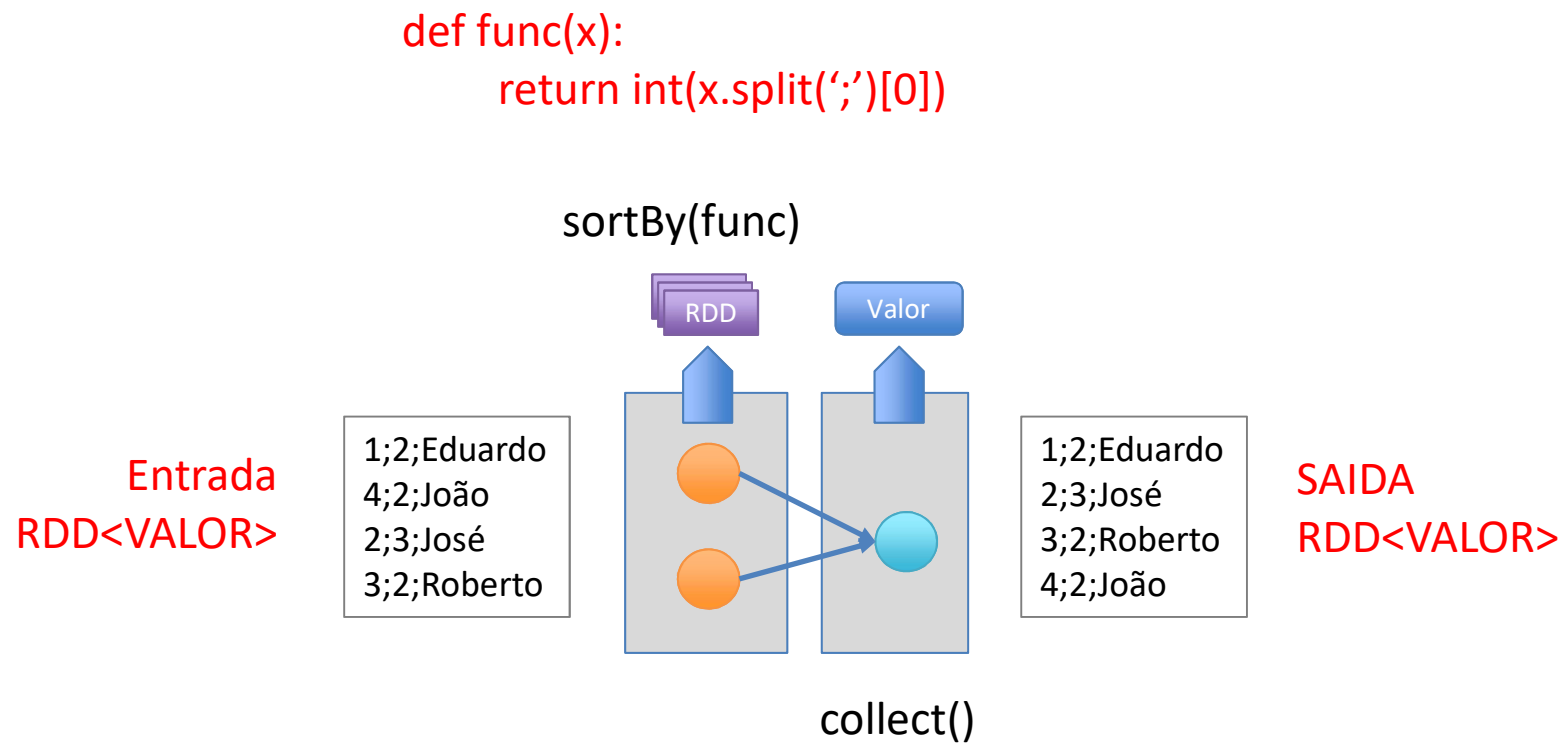


## Exemplo: transformações

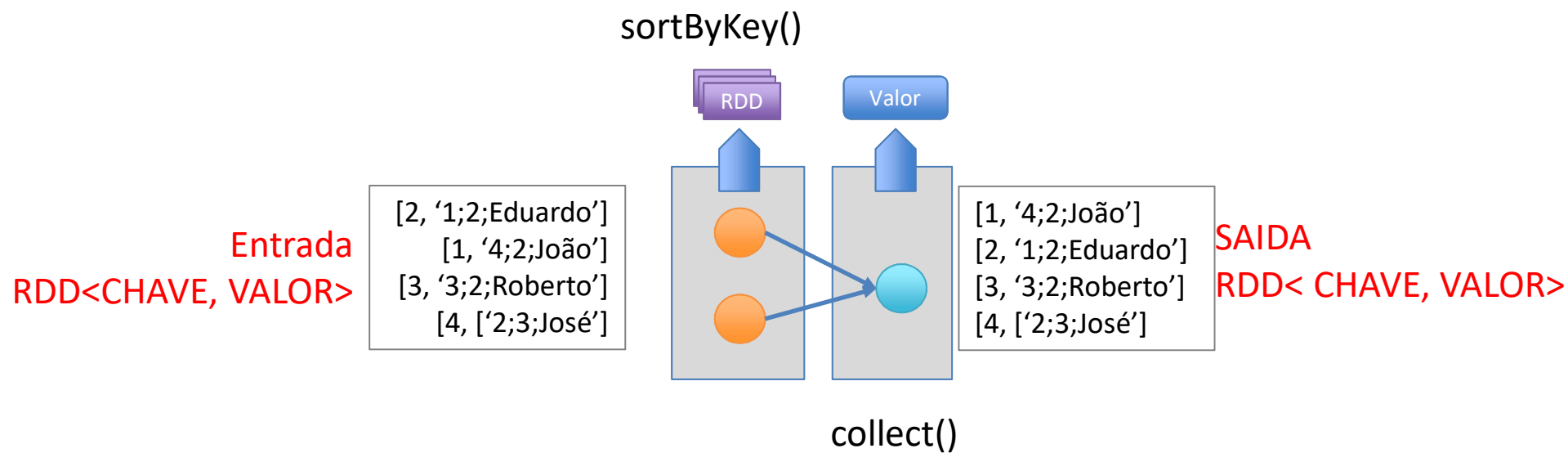
REDUCEBYKEY: Gera um RDD com os valores agrupados de acordo com a chave (função recebe 2 parâmetros)



SORTBY: Gera um RDD ordenado de acordo com os valores retornados pela função

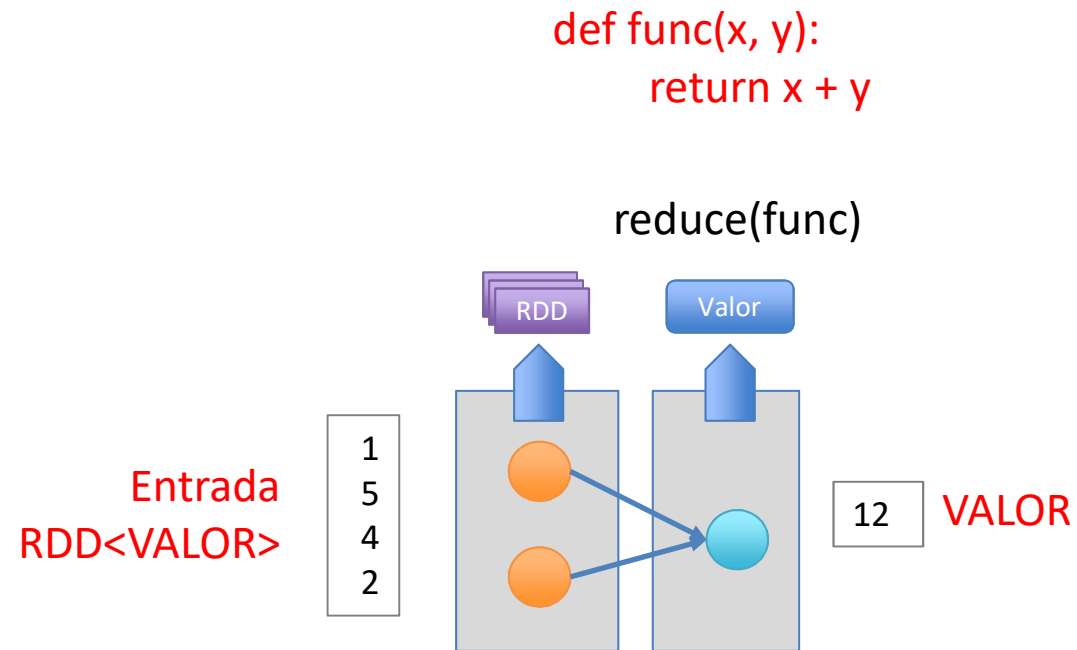


**SORTBY:** Gera um RDD ordenado de acordo com a chave



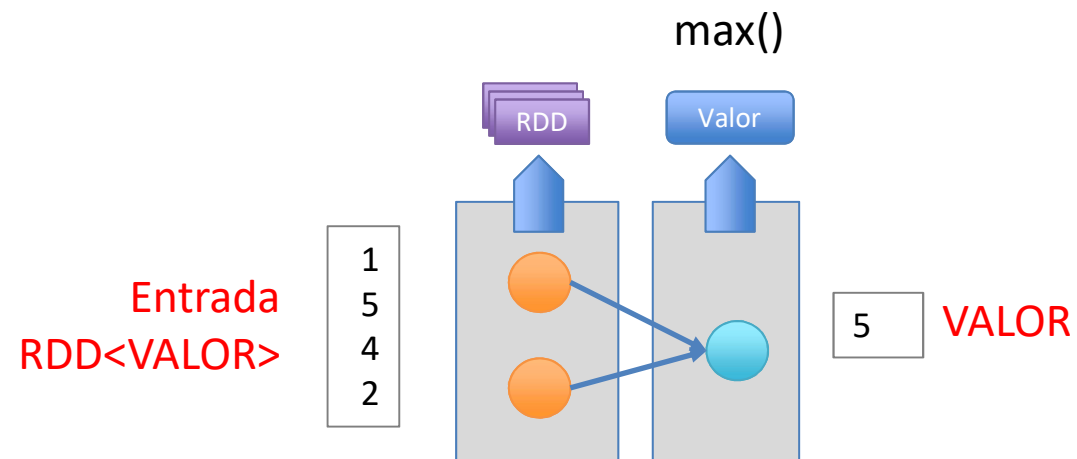
*Exemplo: ações*

REDUCE: Gera um valor agregando o RDD de entrada (função recebe 2 parâmetros)



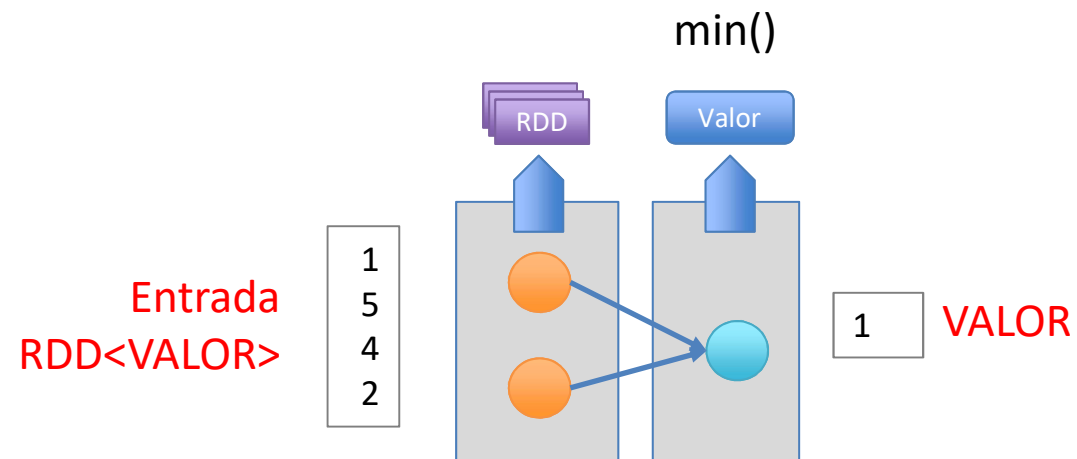
*Exemplo: ações*

MAX: Gera o maior valor do RDD de entrada



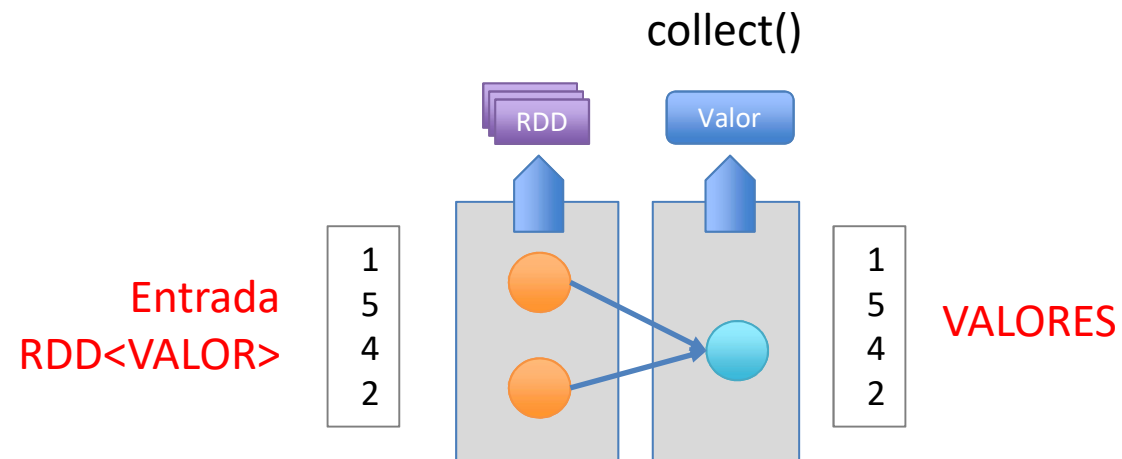
*Exemplo: ações*

MIN: Gera o menor valor do RDD de entrada



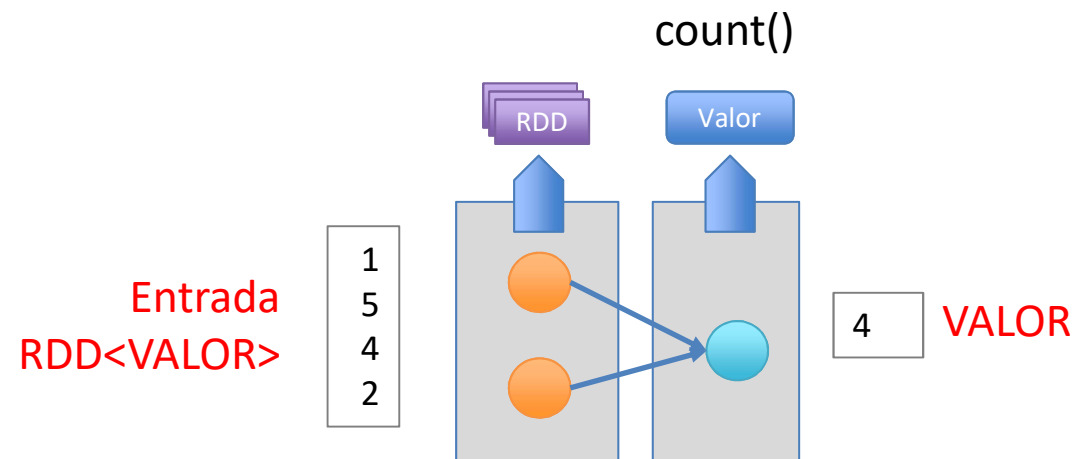
*Exemplo: ações*

COLLECT: Recebe todos os valores do RDD como uma lista no driver



*Exemplo: ações*

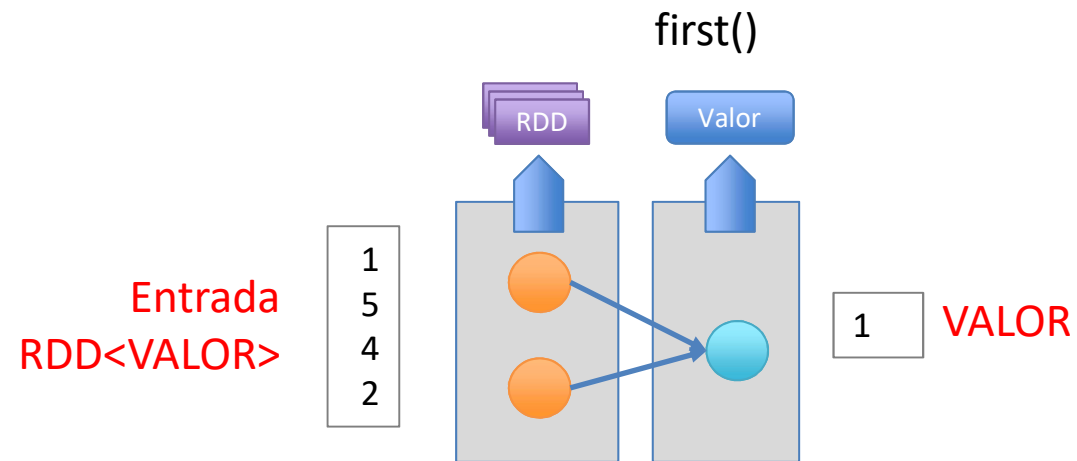
COUNT: Recebe um inteiro com a quantidade de elementos no RDD





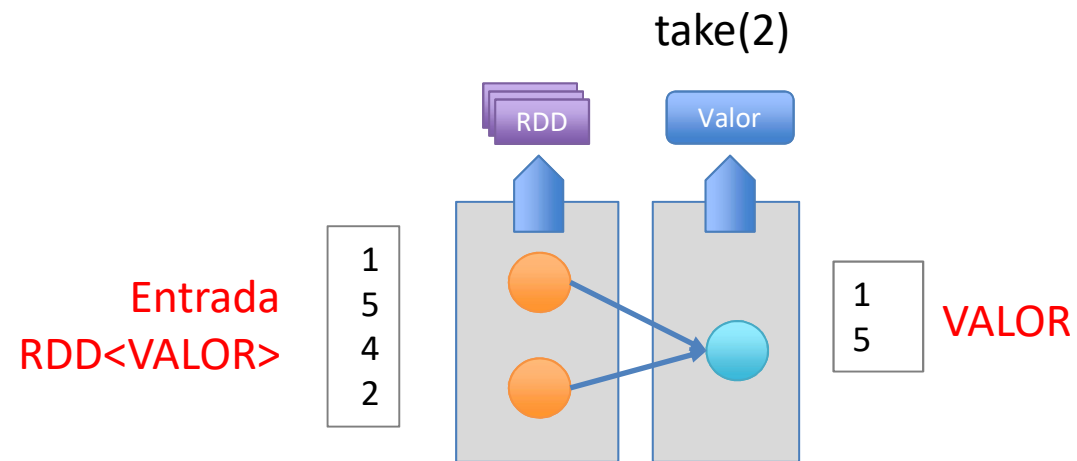
*Exemplo: ações*

FIRST: Recebe o primeiro valor do RDD



*Exemplo: ações*

FIRST: Recebe os N primeiros valores do RDD em uma lista



# Apache Spark - Atividade

Campo	Descrição
Dia	Dia da ocorrência
Mês	Mês da ocorrência
Ano	Ano da ocorrência
Bloco	Região da ocorrência
Tipo	Tipo da ocorrência criminal
Descrição	Breve descrição da ocorrência
Descrição da localização	Descrição sobre a localização da ocorrência, e.g. Rua
Latitude	Localização da ocorrência
Longitude	Localização da ocorrência

