

# Fundamentos de Big Data

Eduardo Viegas



ESCOLA  
**POLITÉCNICA**

# 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

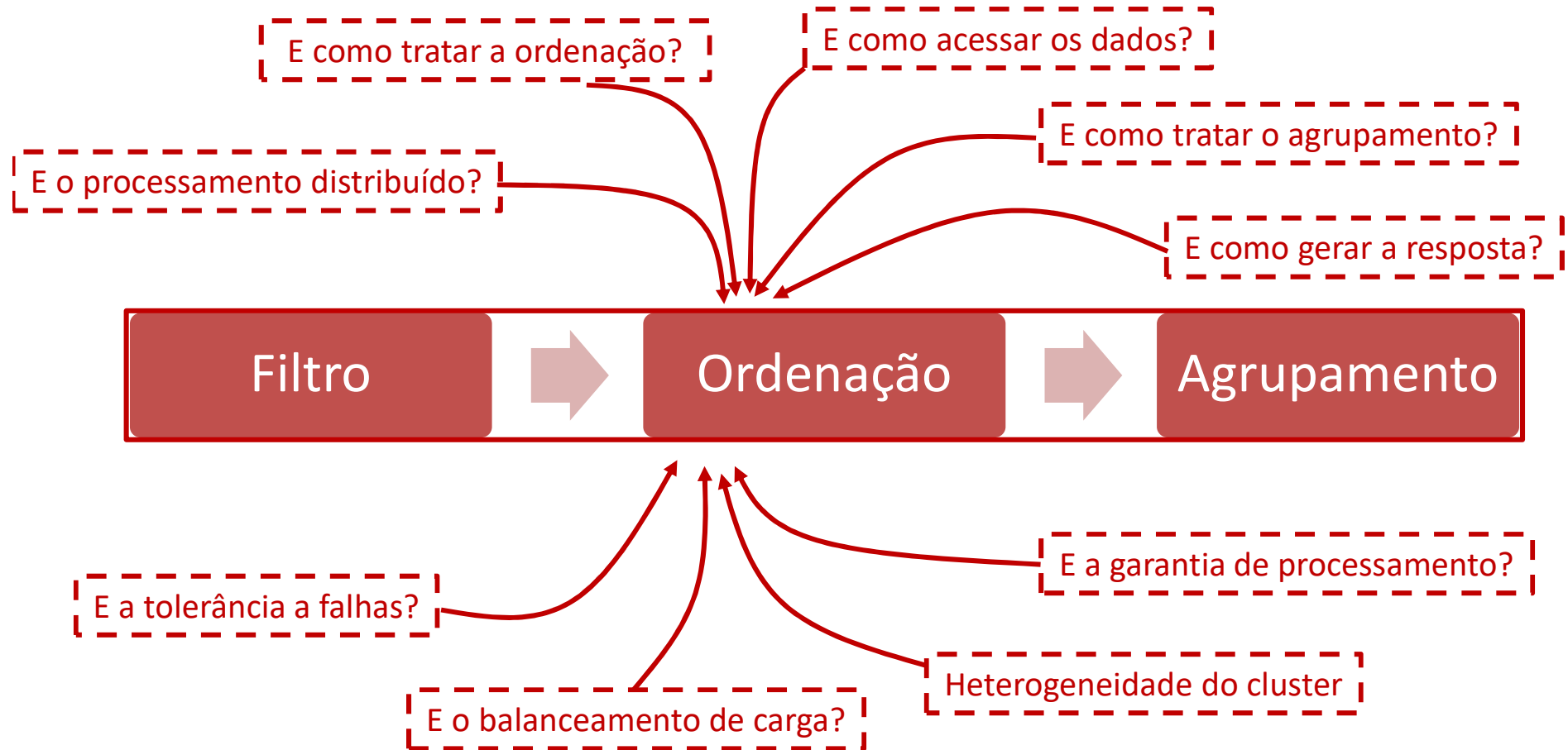
- Suponha que você possui uma base de dados com o seguinte formato
  - 1 TB de dados
  - +1 Bilhão de transações comerciais
  - Dados armazenados no HDFS, mais de 1000 máquinas

| 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?

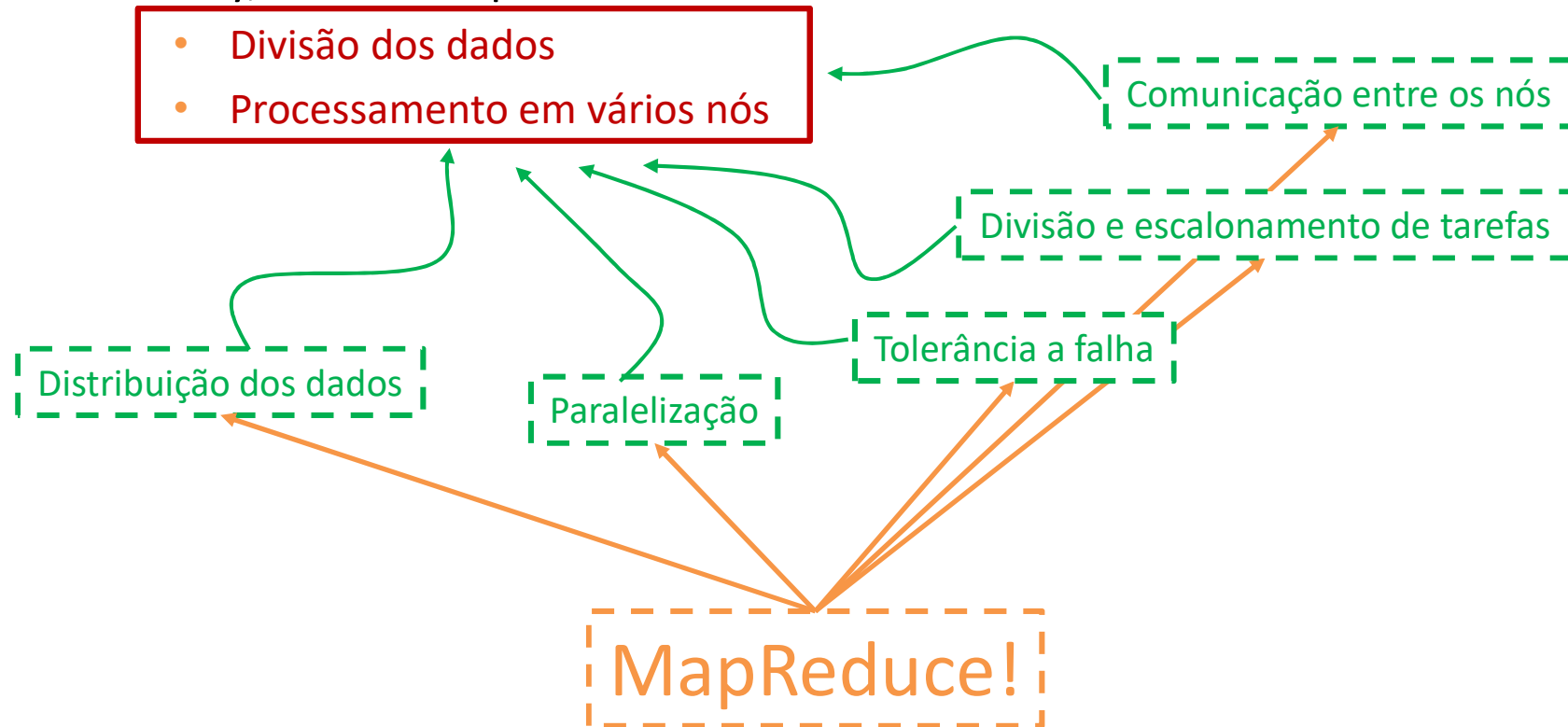
# Análise

- 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

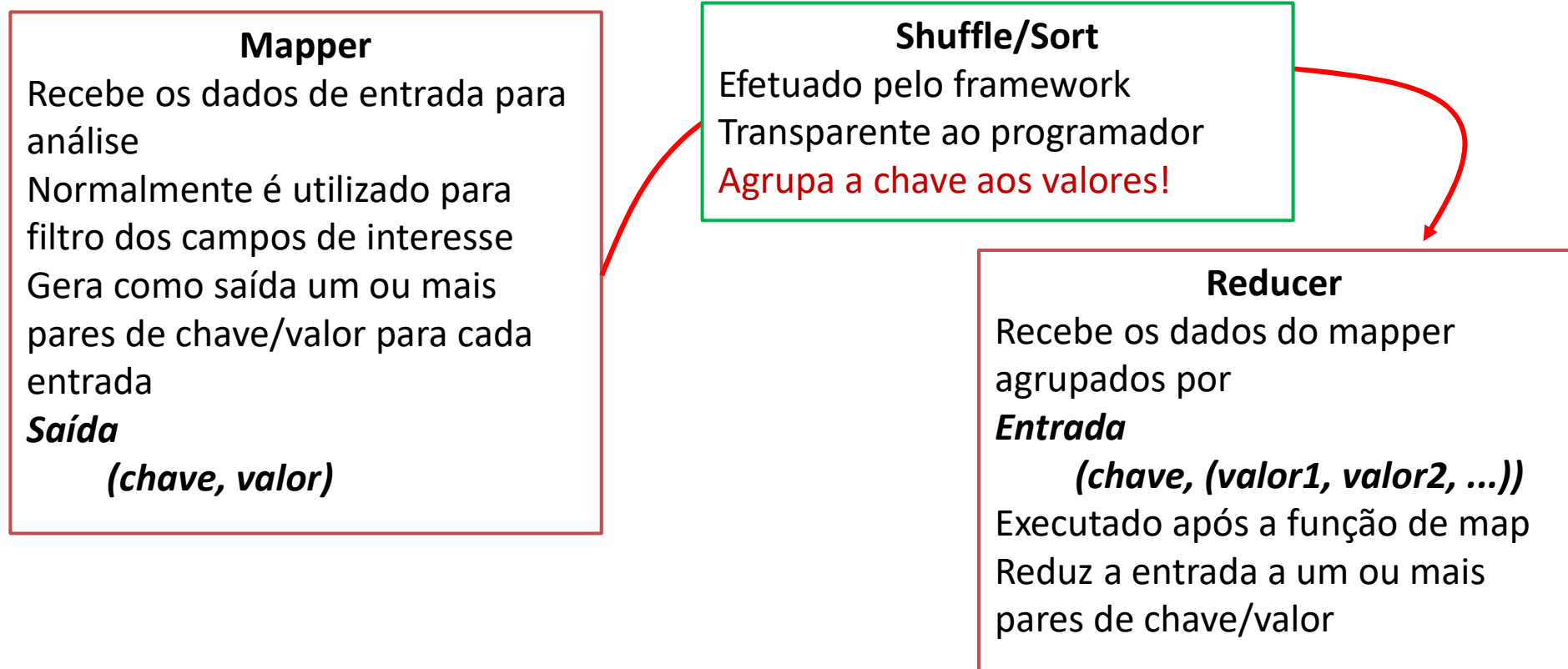
Combina os elementos de uma lista, através de uma função  $f$  para gerar um novo valor  
Ex.

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**



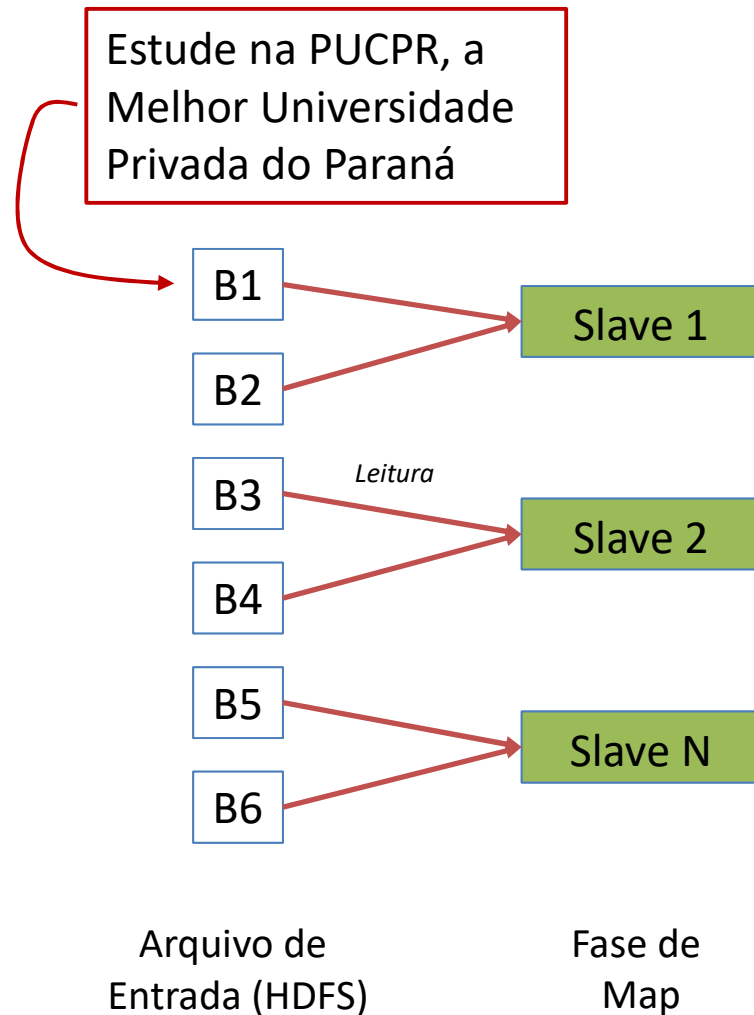
# MapReduce

- Considere um arquivo com o seguinte formato

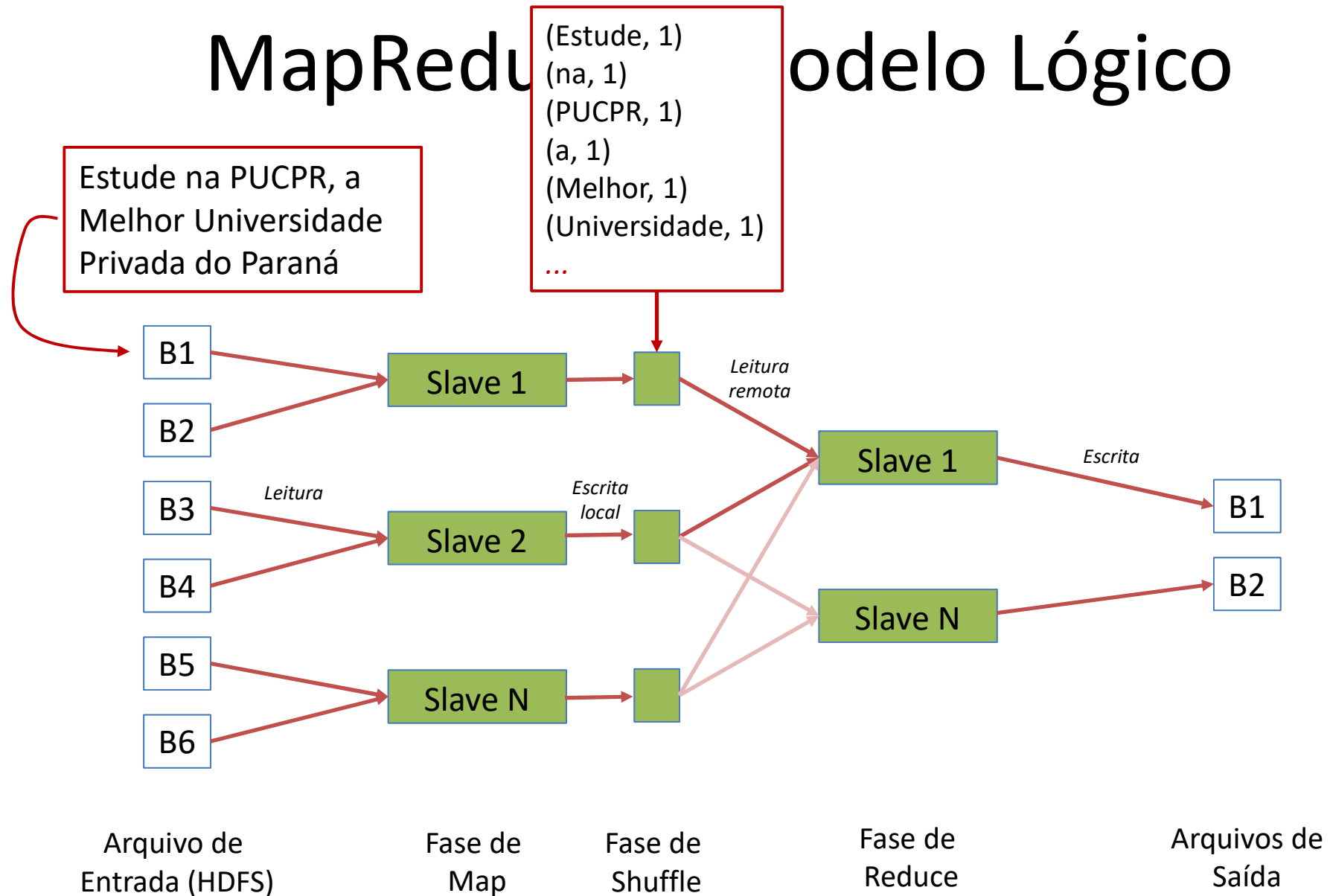
```
Afghanistan;2016;010410;Sheep, live;Export;6088;2339;Number of items;51;01_live_animals
Afghanistan;2016;010420;Goats, live;Export;3958;984;Number of items;53;01_live_animals
Afghanistan;2008;010210;Bovine animals, live pure-bred breeding;Import;1026804;272;Number of items;3769;01_live_animals
Albania;2016;010290;Bovine 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?

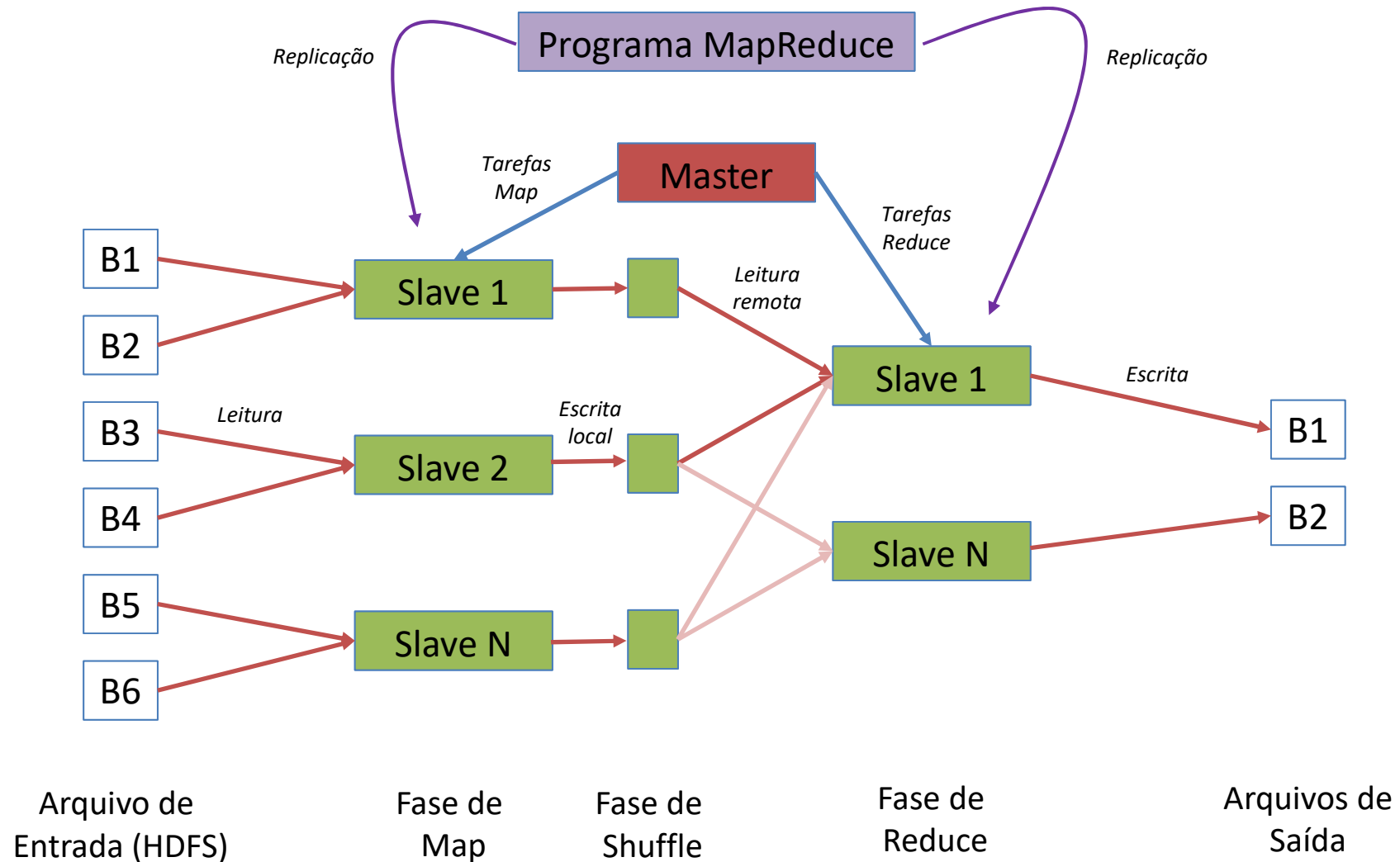
# MapReduce – Modelo Lógico



# MapReduce Modelo Lógico



# MapReduce – Modelo Lógico

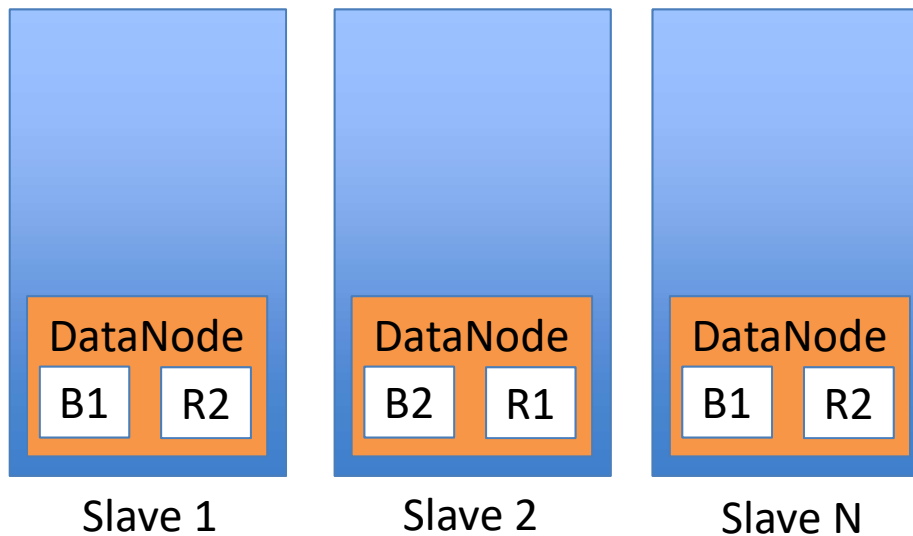


# MapReduce - Arquitetura

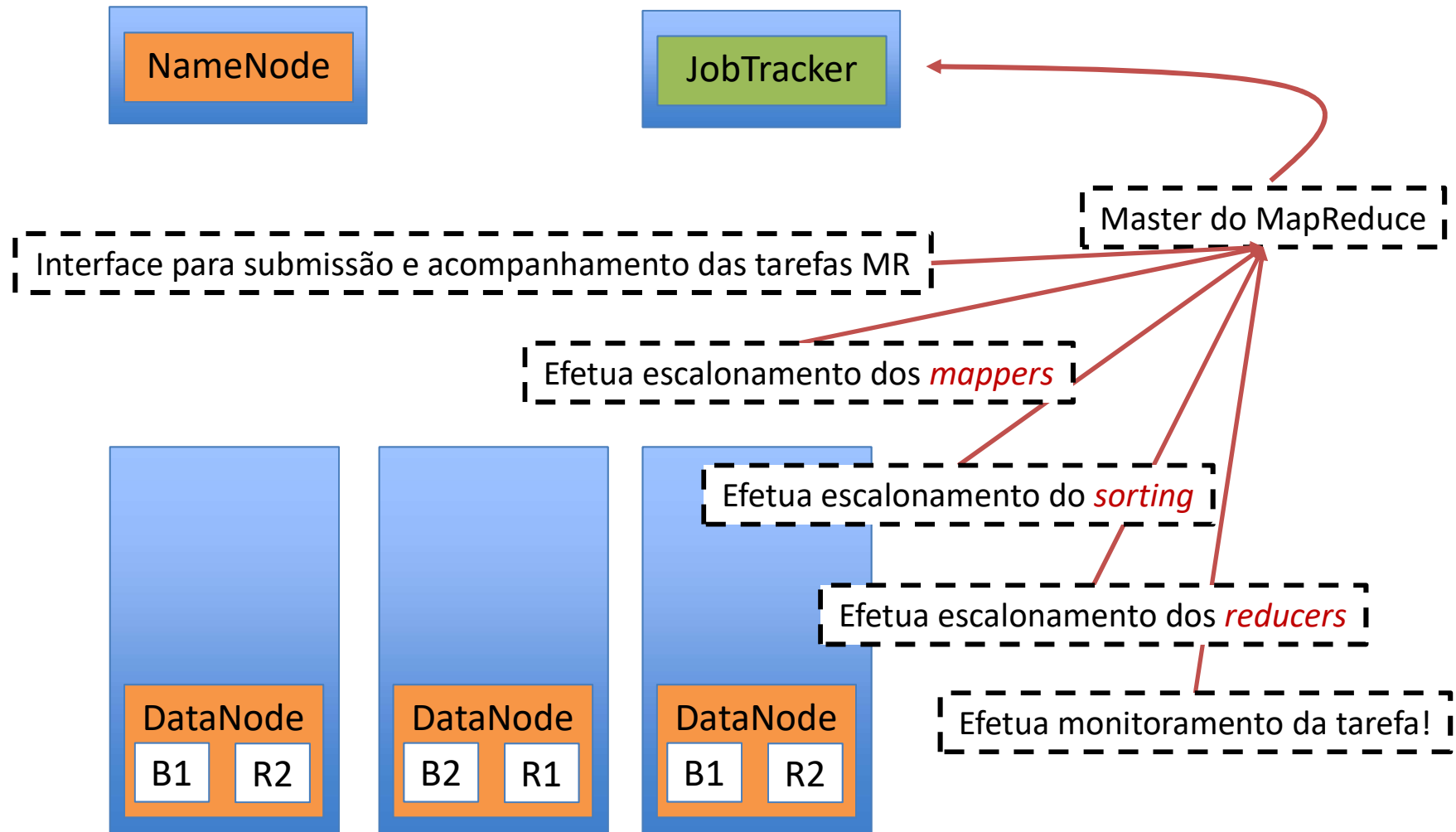


Cada nó é parte de um cluster HDFS

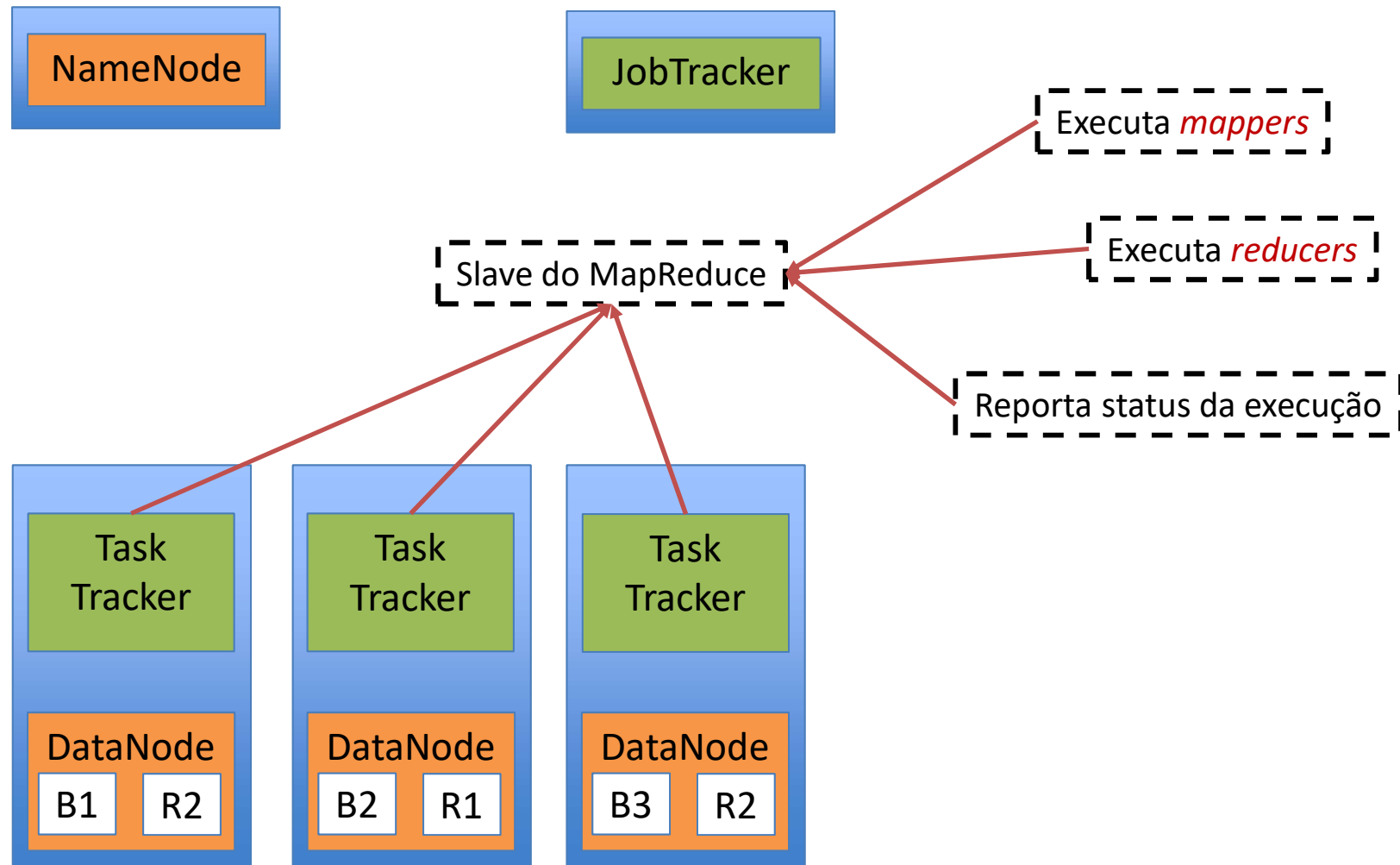
Dados de entrada são armazenados no HDFS e replicados nos nós



# MapReduce - Arquitetura



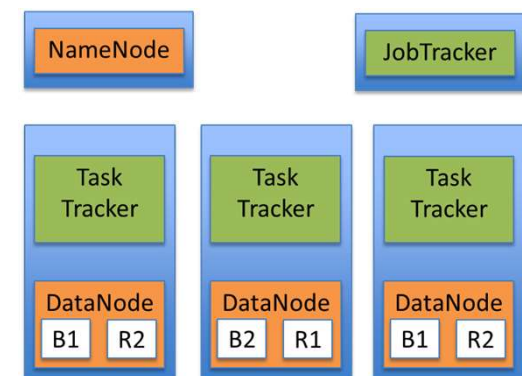
# MapReduce - Arquitetura





# 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

## ■ Principio da localidade

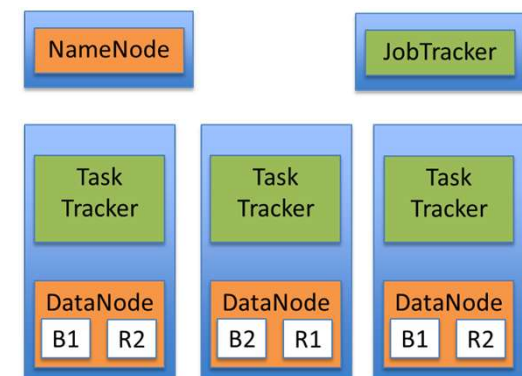
- Framework evita mover os dados pela rede
- TaskTracker é executado geralmente no mesmo nó do DataNode, bloco é lido localmente!
- JobTracker divide as tarefas baseado na localização dos dados

## ■ Paralelismo

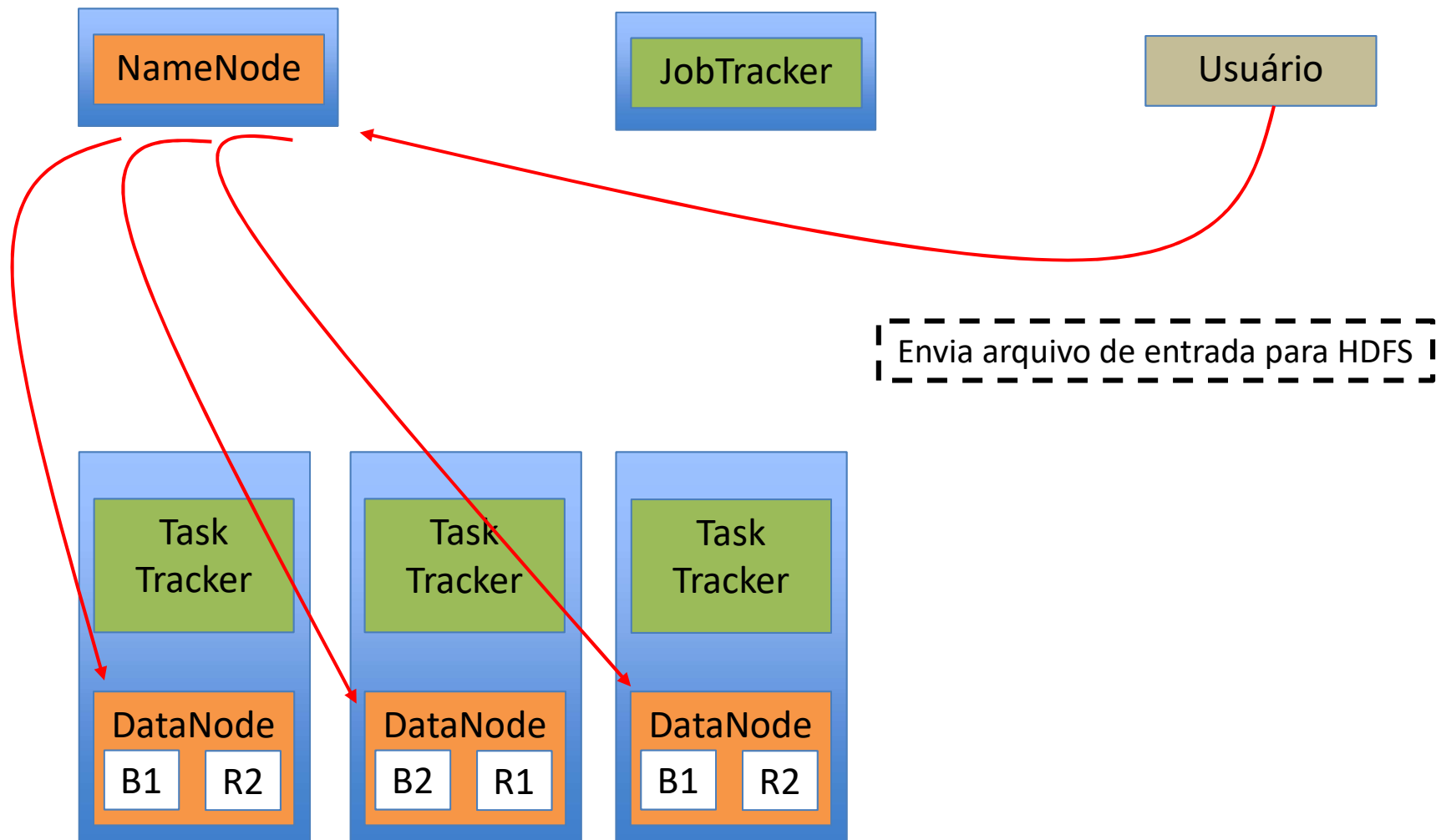
- Função de map é executada paralelamente sobre cada bloco
- Função de reduce é executada em paralelo sobre chaves diferentes
- Função reduce é executada apenas após a execução da função map

## ■ Tolerância a falha

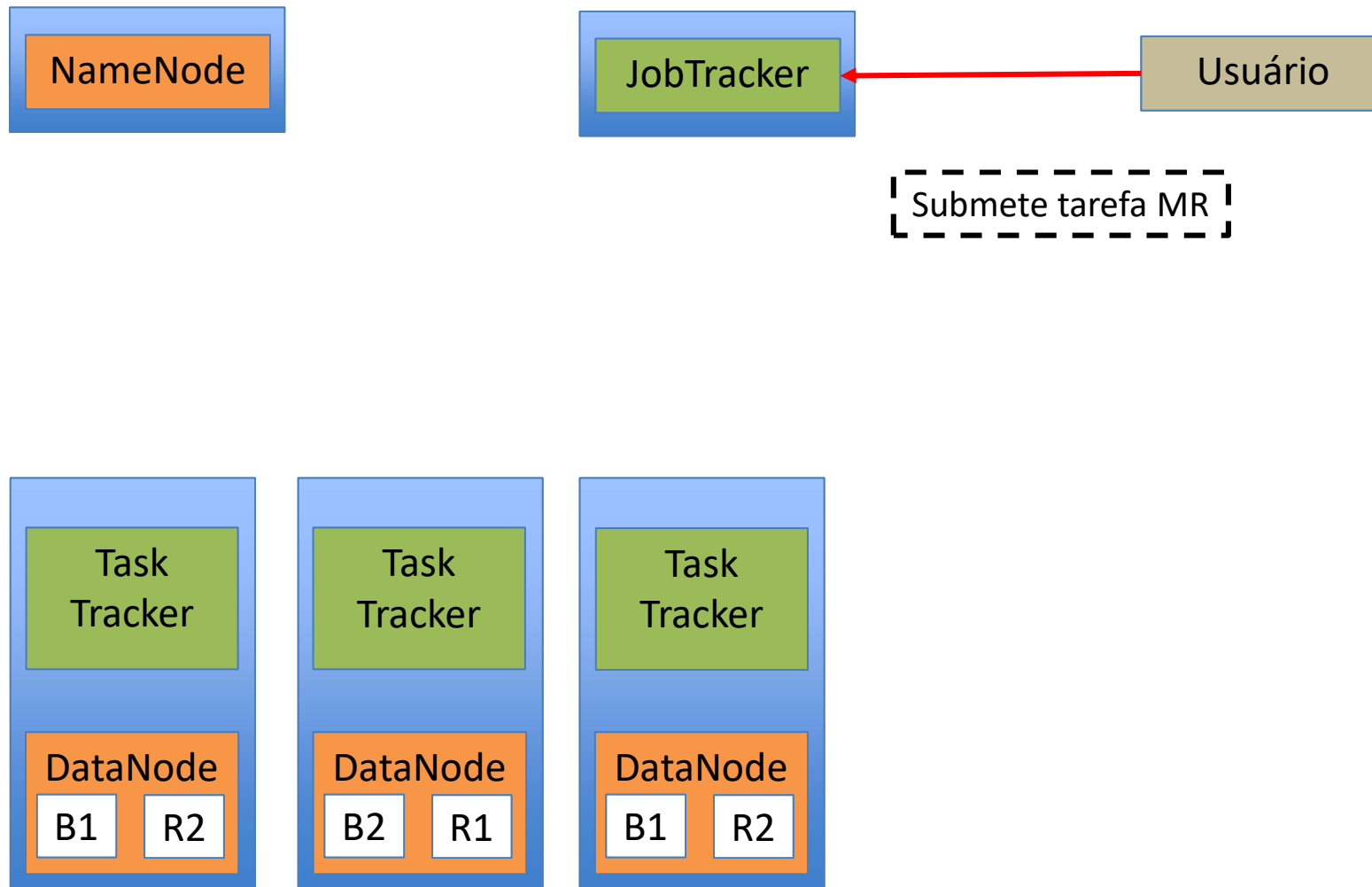
- JobTracker monitora TaskTrackers
- Falhas são tratadas pela reexecução sobre o bloco!
- Se um nó falha, outro nó com as replicas dos blocos reexecuta a tarefa



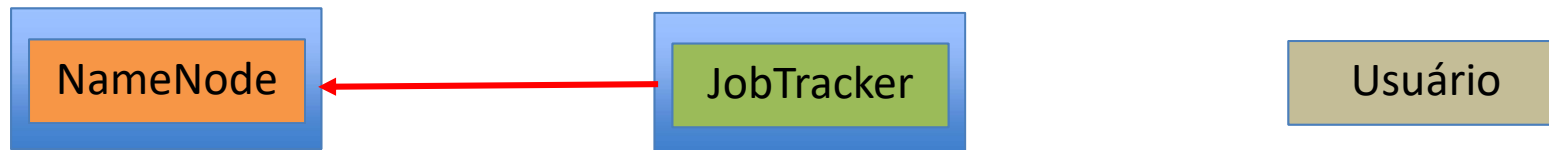
# MapReduce - Processo



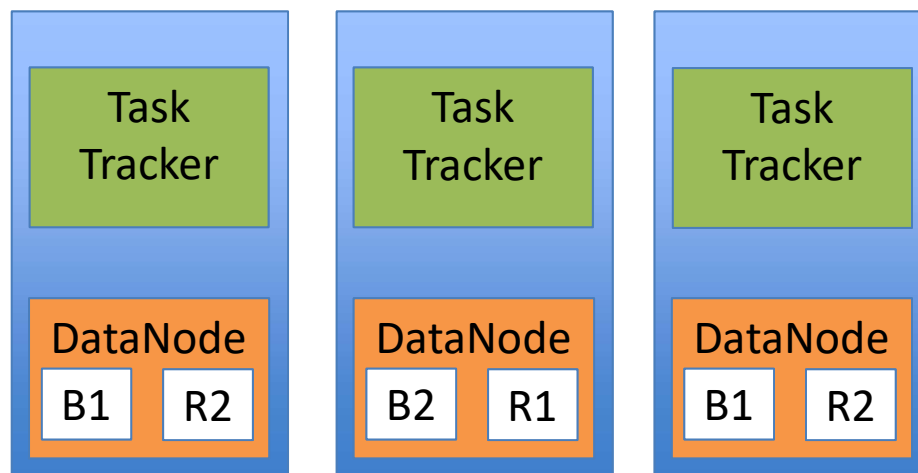
# MapReduce - Processo



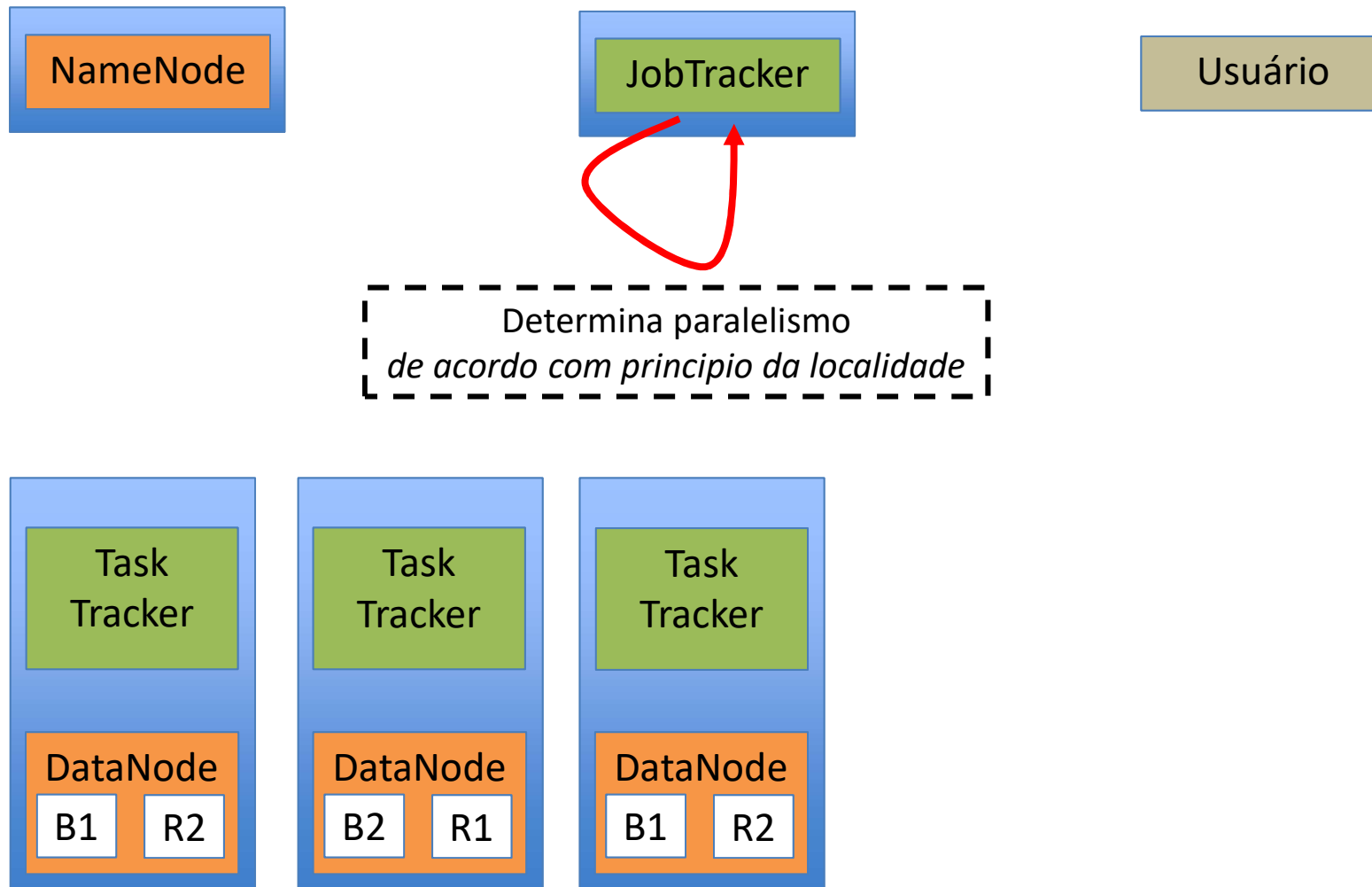
# MapReduce - Processo



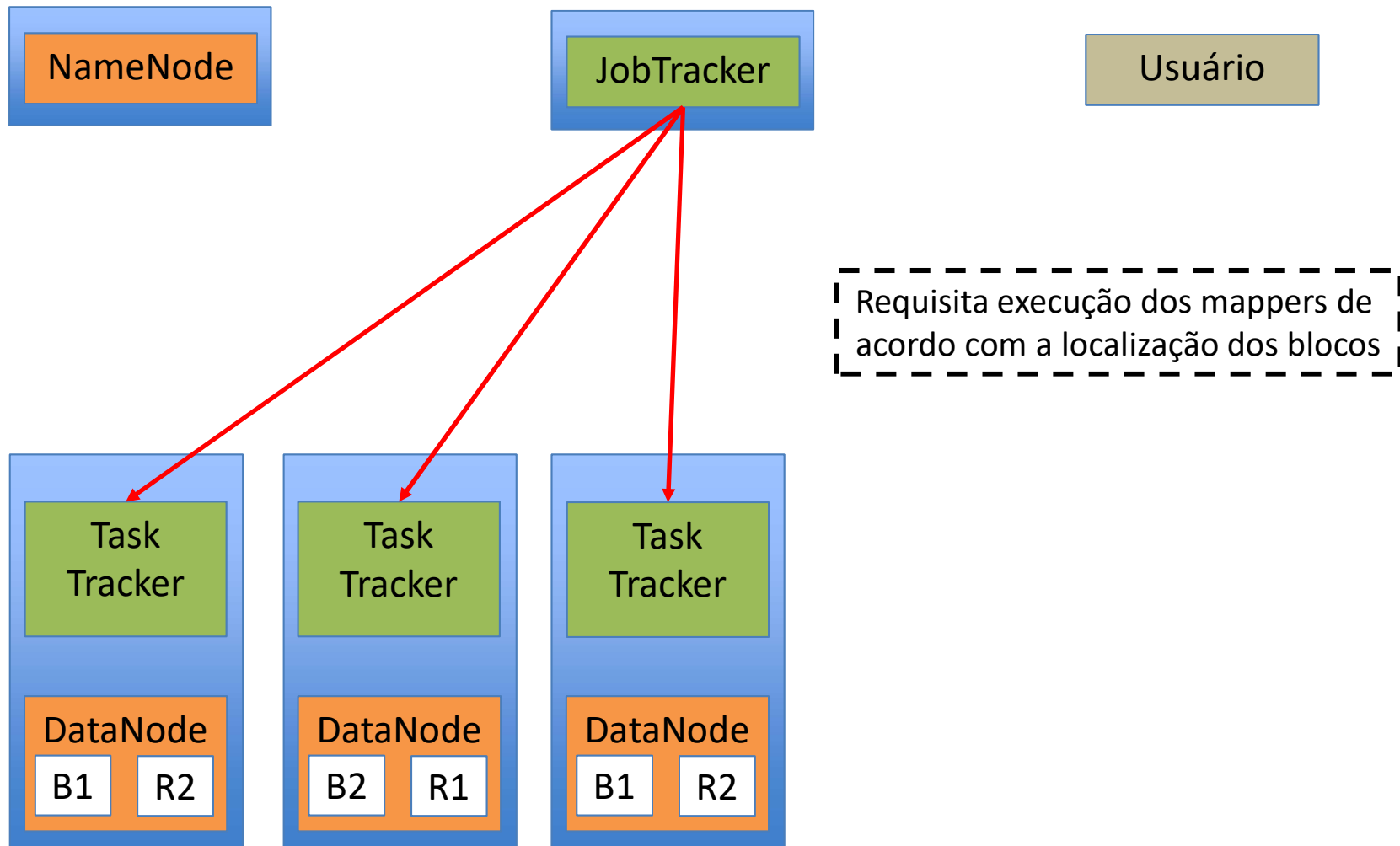
Requisita metadados do arquivo de entrada



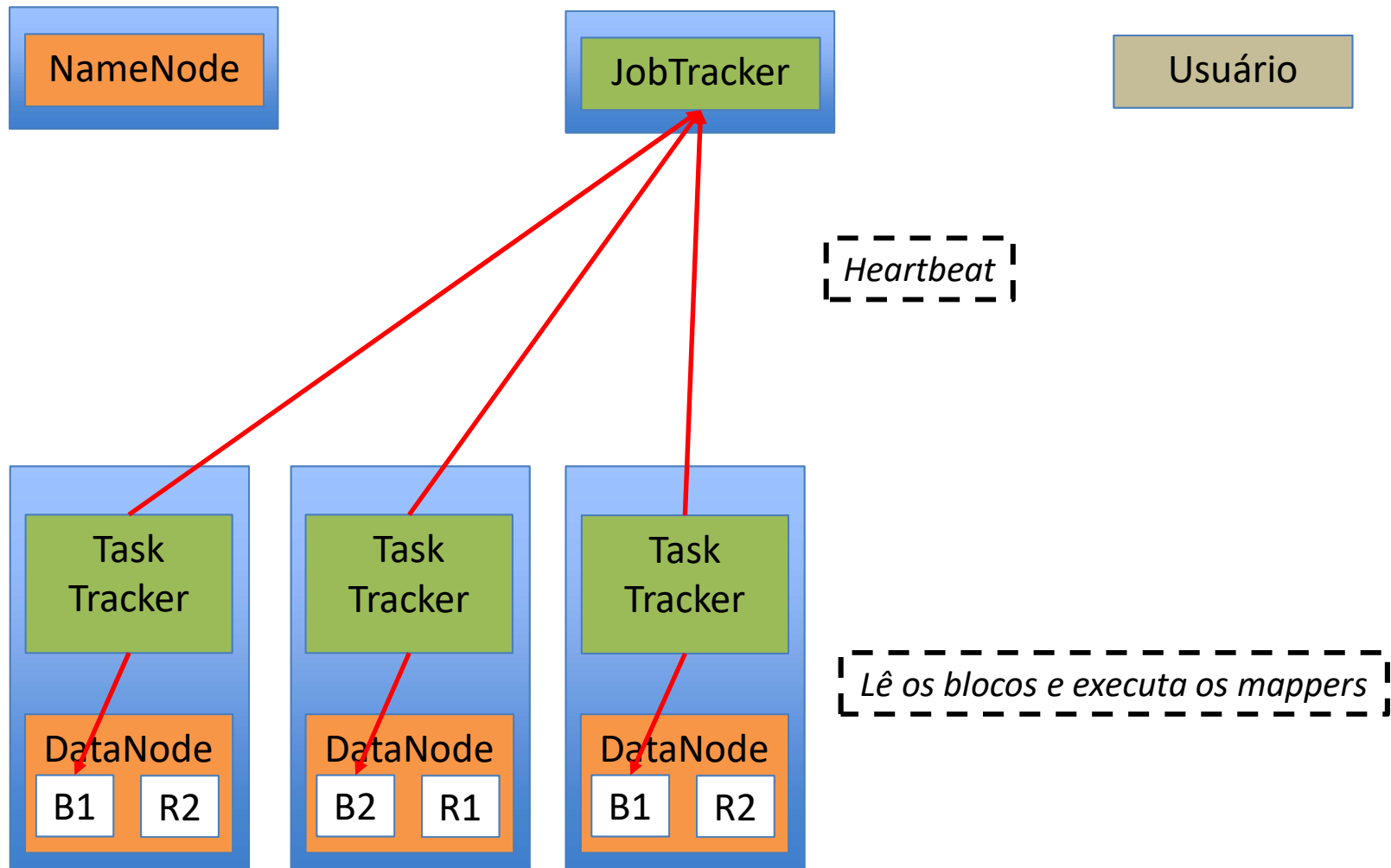
# MapReduce - Processo



# MapReduce - Processo

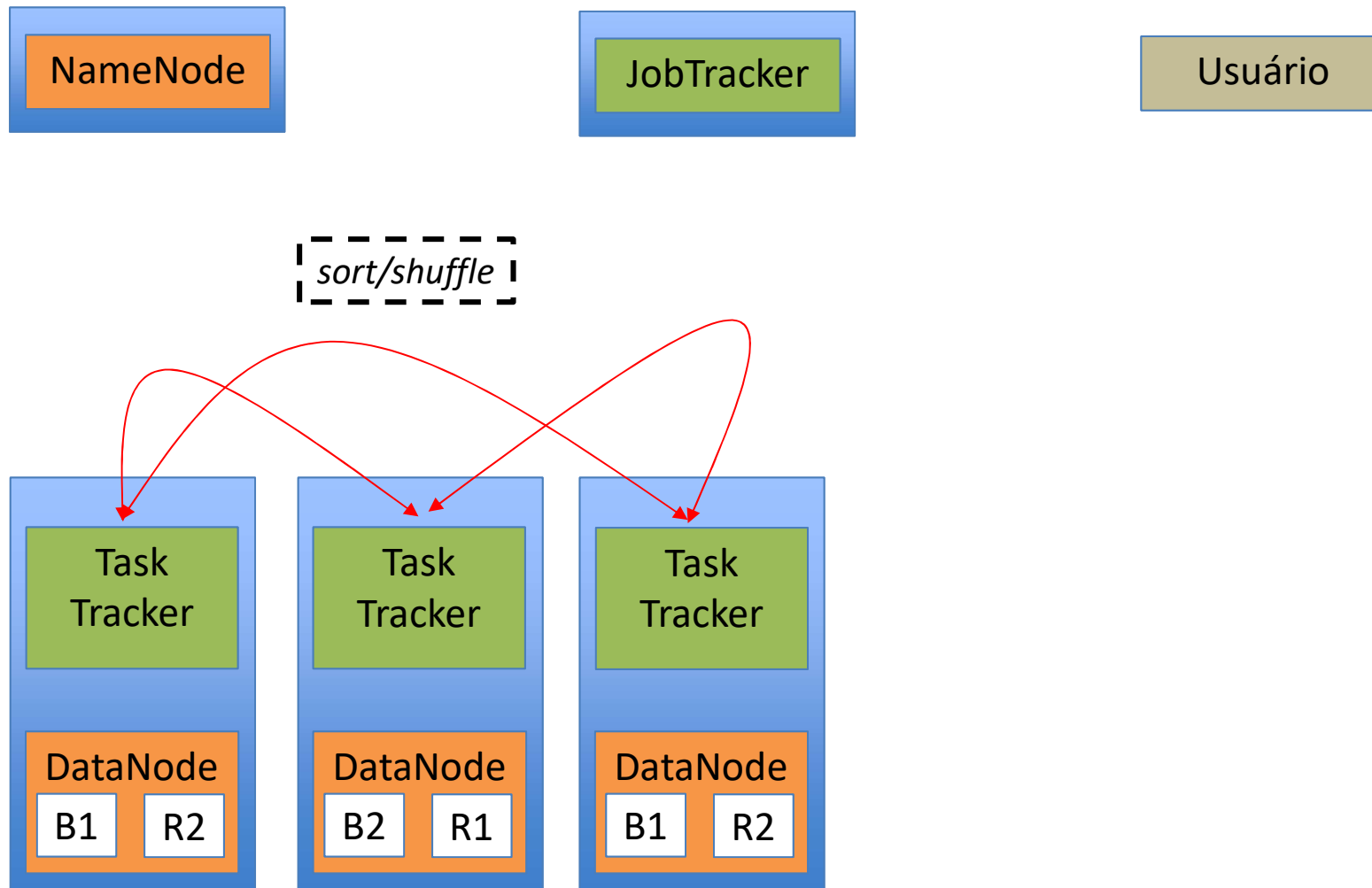


# MapReduce - Processo

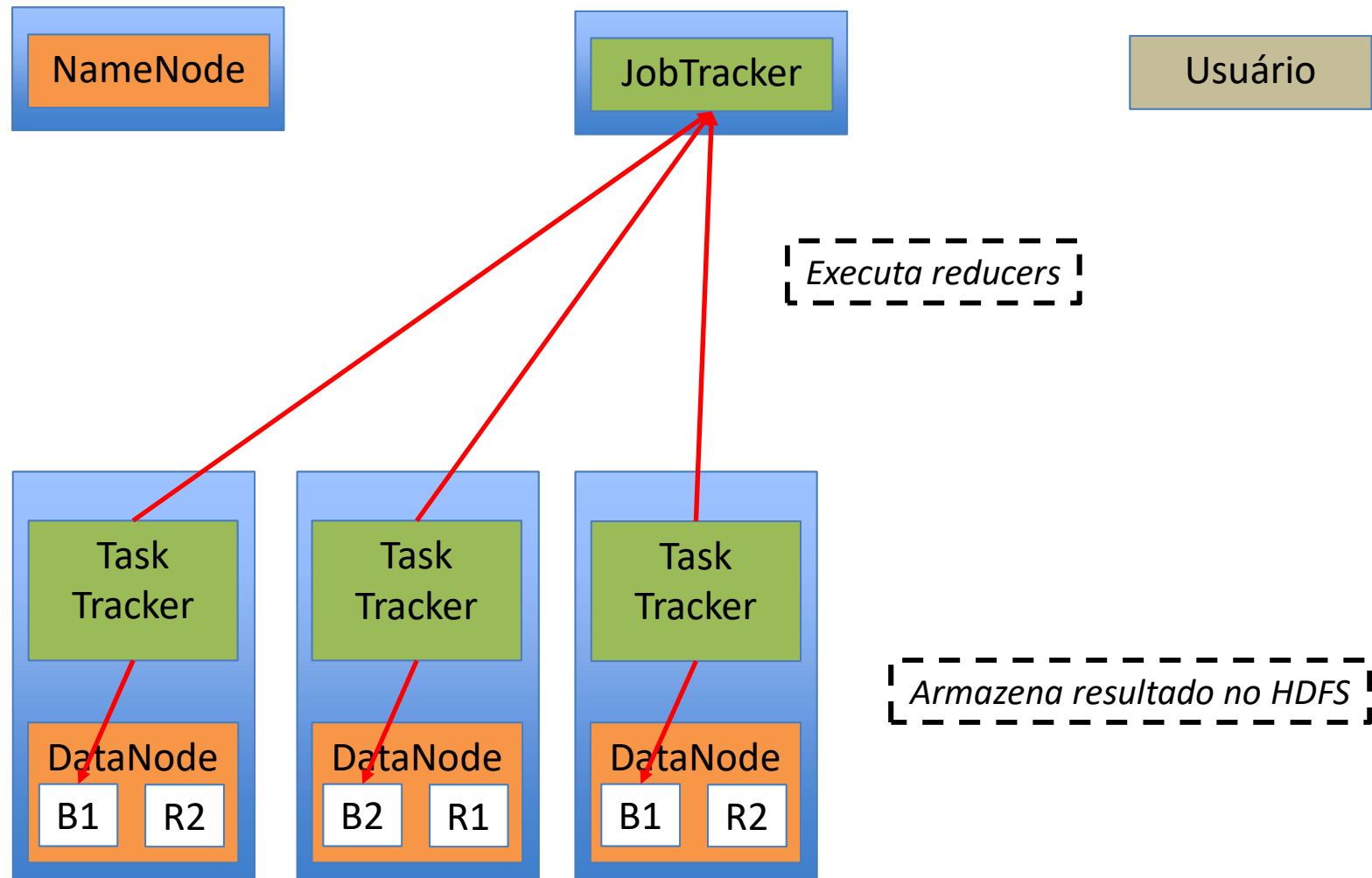




# MapReduce - Processo

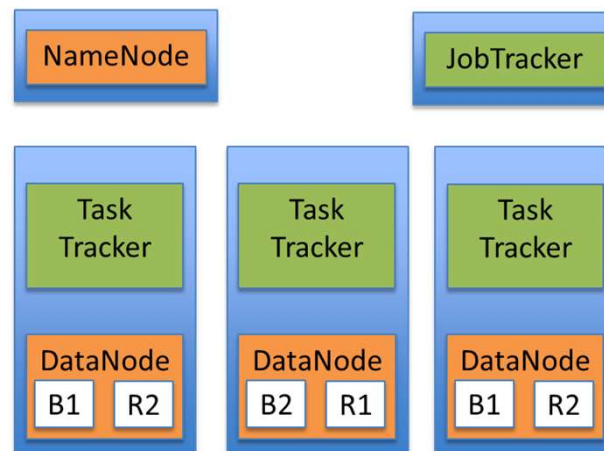


# MapReduce - Processo



# 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 jobtracker
  - 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              |

