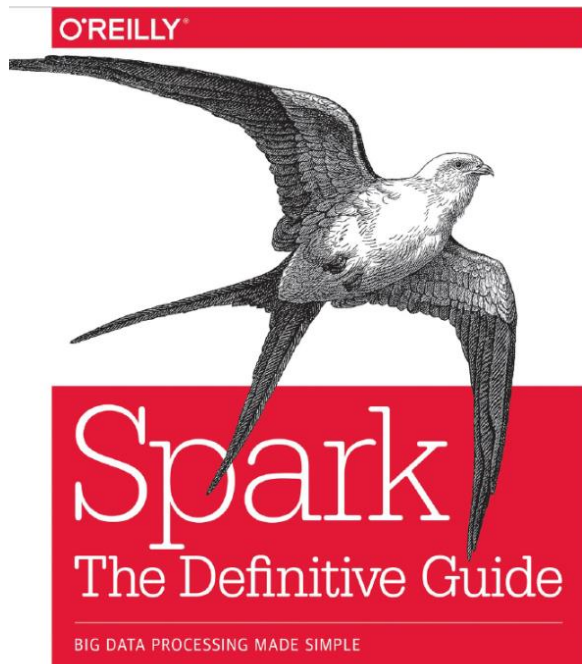


# Introdução ao Apache Spark

## Arquitetura do Spark - Parte 2



### Capítulos Abordados

#### 2. A Gentle Introduction do Spark

- Partitions
- Transformations / Actions
- Spark UI

#### 15. How Spark Runs on a Cluster

- Jobs / Stages / Tasks



# <Partitions/>

## Dados Particionados

RDD split into 5 partitions

item-1	item-6	item-11	item-16	item-21
item-2	item-7	item-12	item-17	item-22
item-3	item-8	item-13	item-18	item-23
item-4	item-9	item-14	item-19	item-24
item-5	item-10	item-15	item-20	item-25

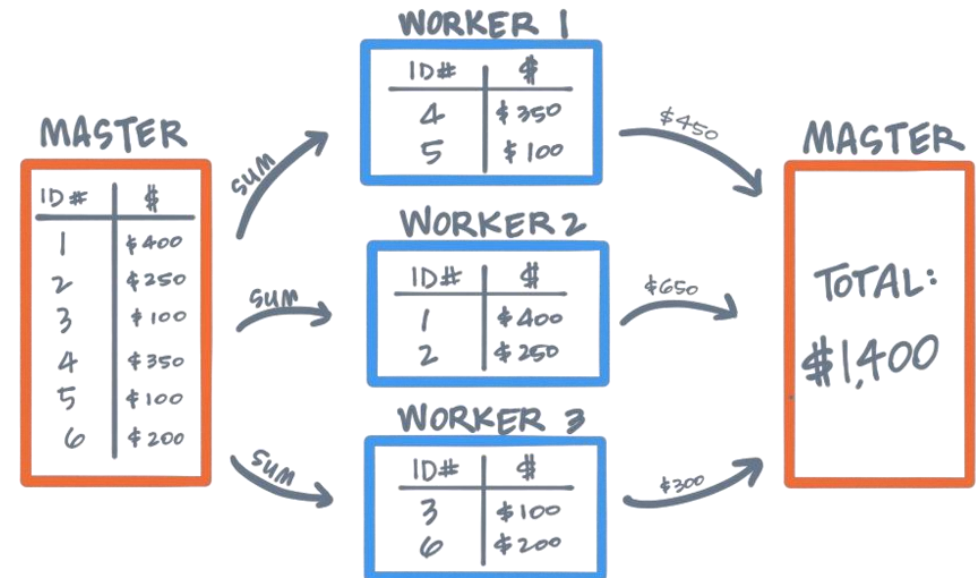
Worker  
Spark  
executor

Worker  
Spark  
executor

Worker  
Spark  
executor

```
val rdd = sc.parallelize(1 to 25, 5) // 25 items, 5 partitions
```

## Processamento Paralelo e Distribuído



```
val money = List(400,250,100,350,100,200)
val sum_money = sc.parallelize(money, 3).sum()
```

# <Transformations and Actions/>

## Transformações

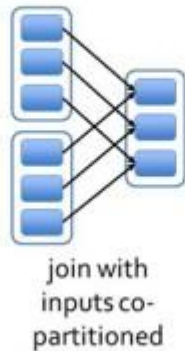
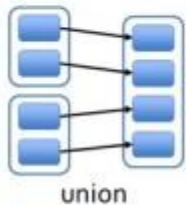
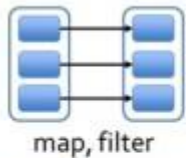
### Narrow

- *map*
- *filter*
- *flatMap*
- *mapValues*

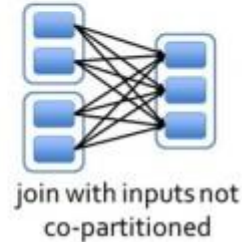
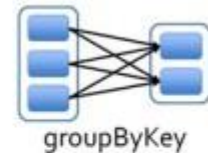
### Wide

- *groupByKey*
- *reduceByKey*
- *distinct*
- *join*
- *repartition*

"Narrow" deps:



"Wide" (shuffle) deps:



## Ações

### Lazy Evaluation

Transformações  
(Criam o plano lógico)



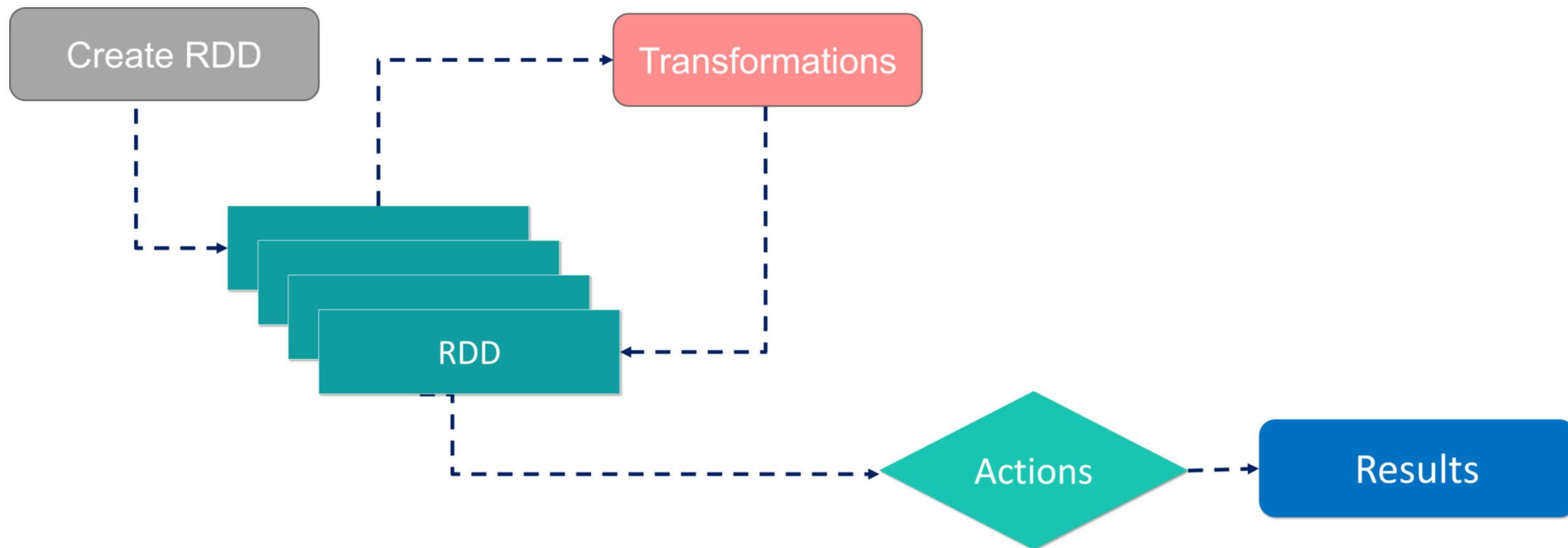
Action  
(Desencadeia a computação)

### Tipos de ações

- Disponibilizar dados no console
- Transformar dados em objetos nativos da linguagem (ex. List)
- Persistir os dados no output final (HDFS, SQL Server, Hive, Hbase, Kafka, etc.)

# <RDDs Life Cycle/>

A big data company



# <Jobs, stages and tasks/>

## Jobs:

Criado a partir de uma ação.

Exemplos:

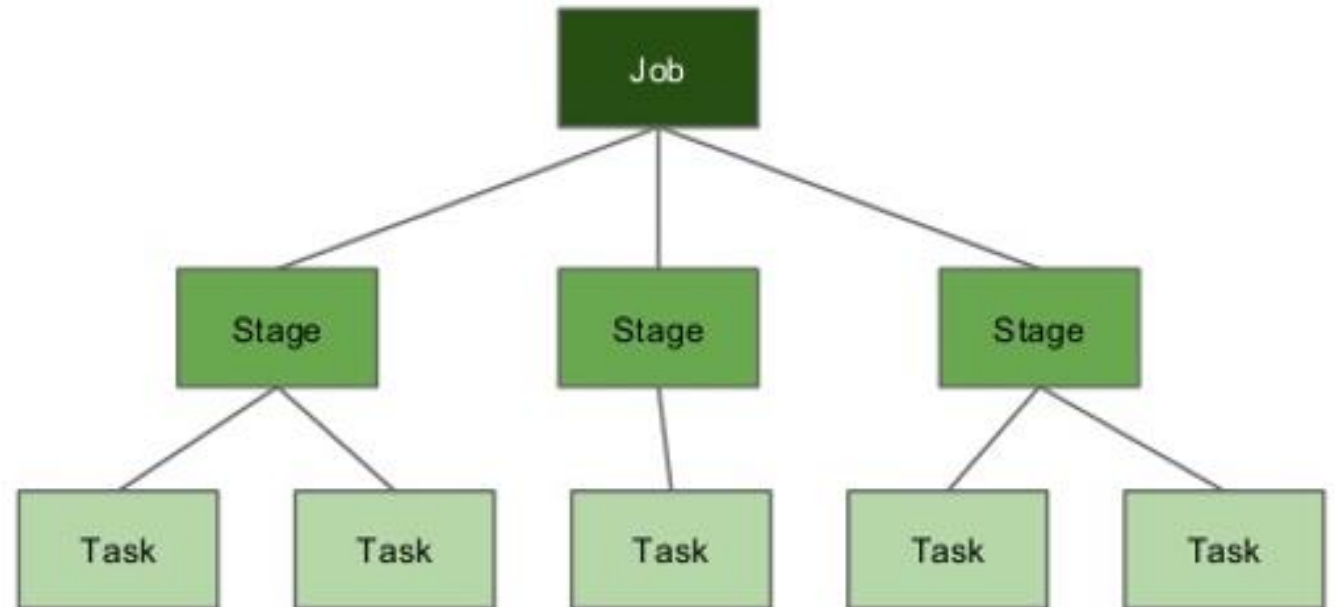
- `df.show()`
- `rdd.count()`
- `df.write.insertInto()`

## Stages:

Grupo de tarefas iguais sendo executadas em paralelo

## Tasks:

Unidade de computação aplicada a uma única partição



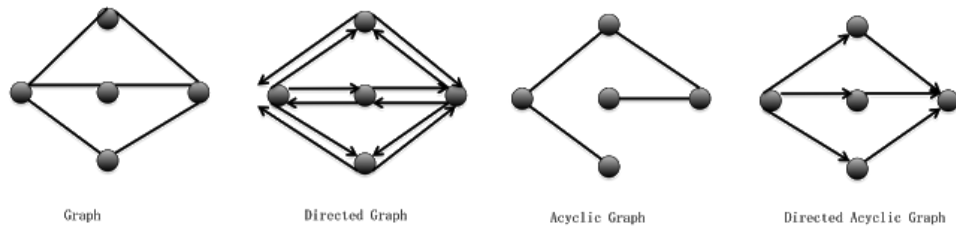


# <Jobs, stages and tasks/>

## Directed Acyclic Graph (DAG)

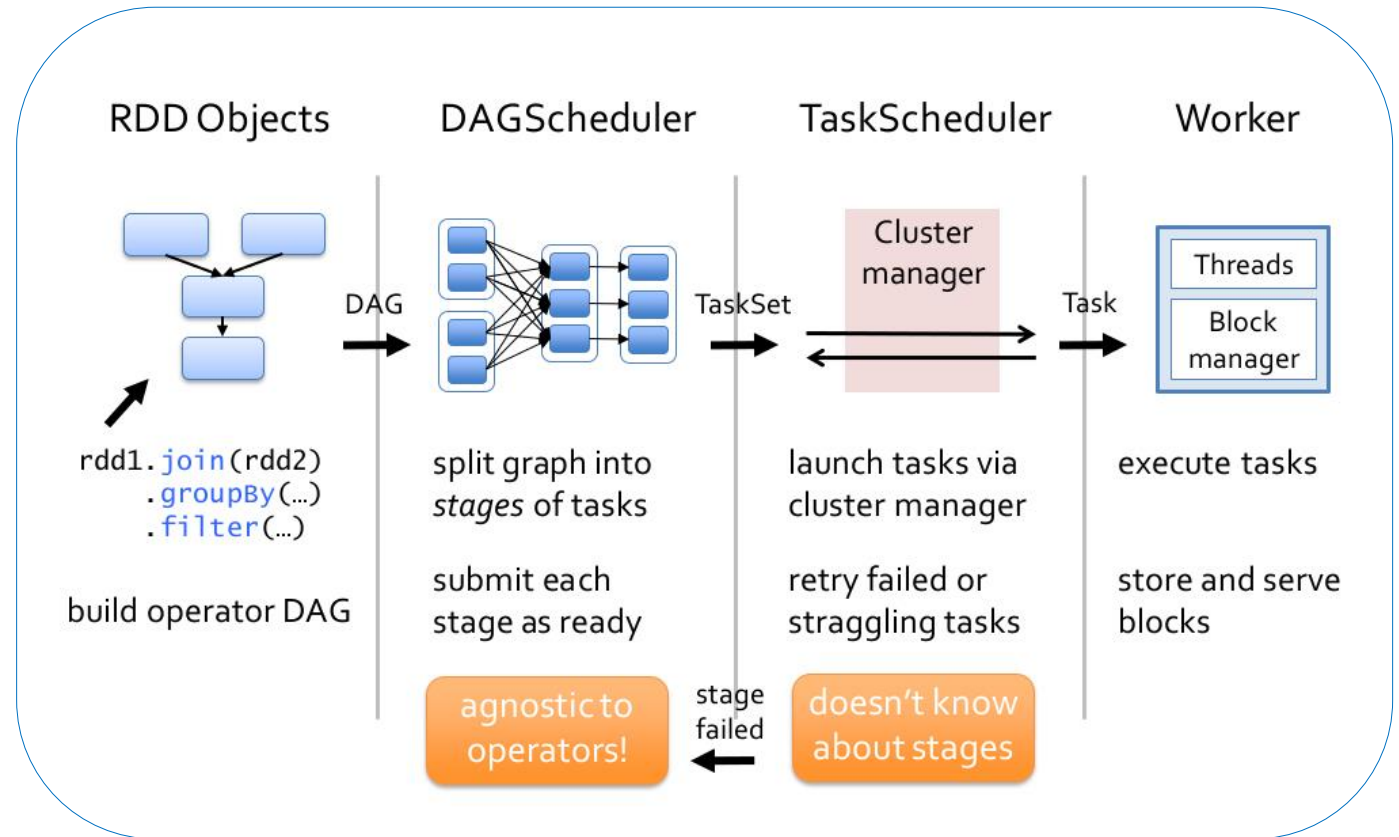
Vértices: RDDs

Arestas: Transformações



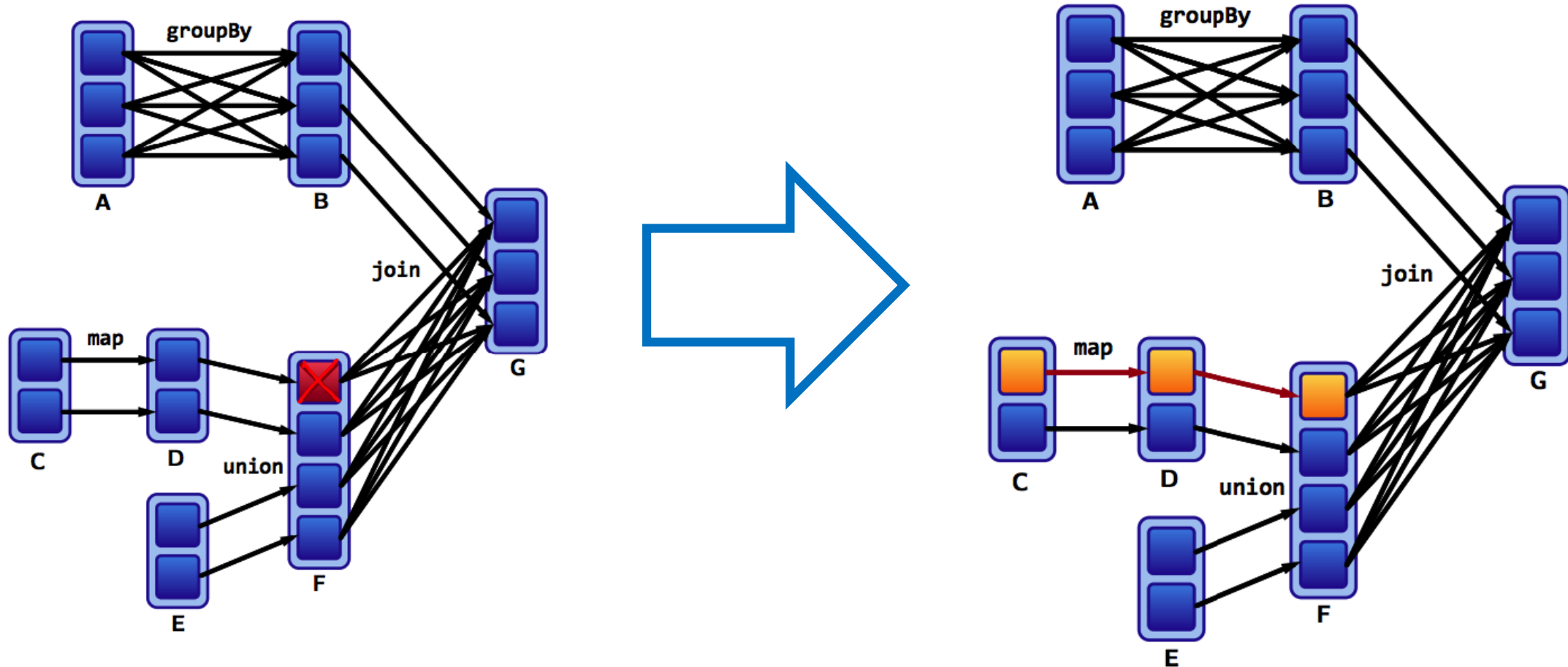
### DAG Optimizer

Rearranjo da ordem das transformações para alcançar o mesmo resultado

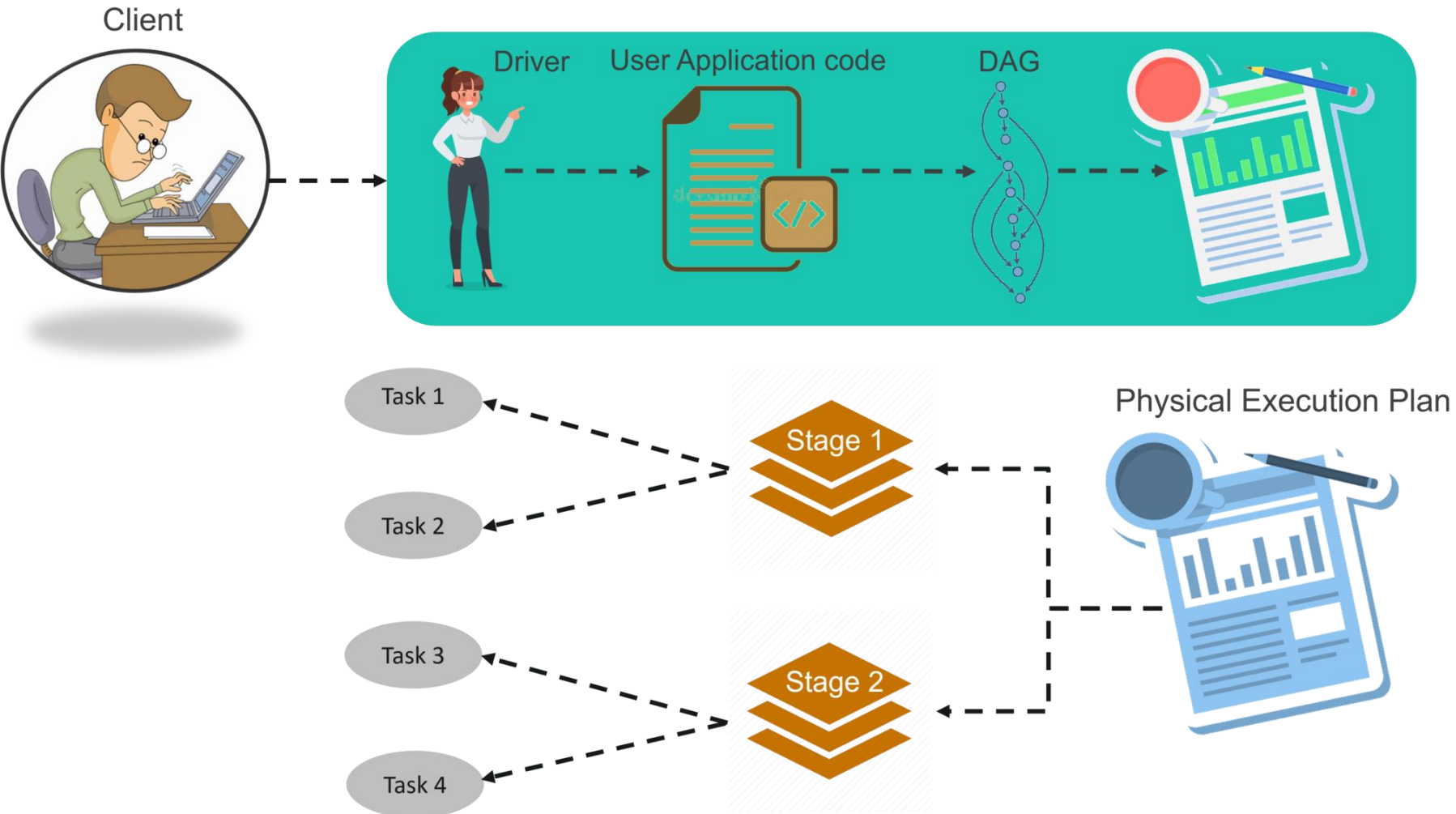


# <Fault Tolerance/>

## Recomputação apenas dos dados perdidos (Partição perdida)



# <Putting all together.../>

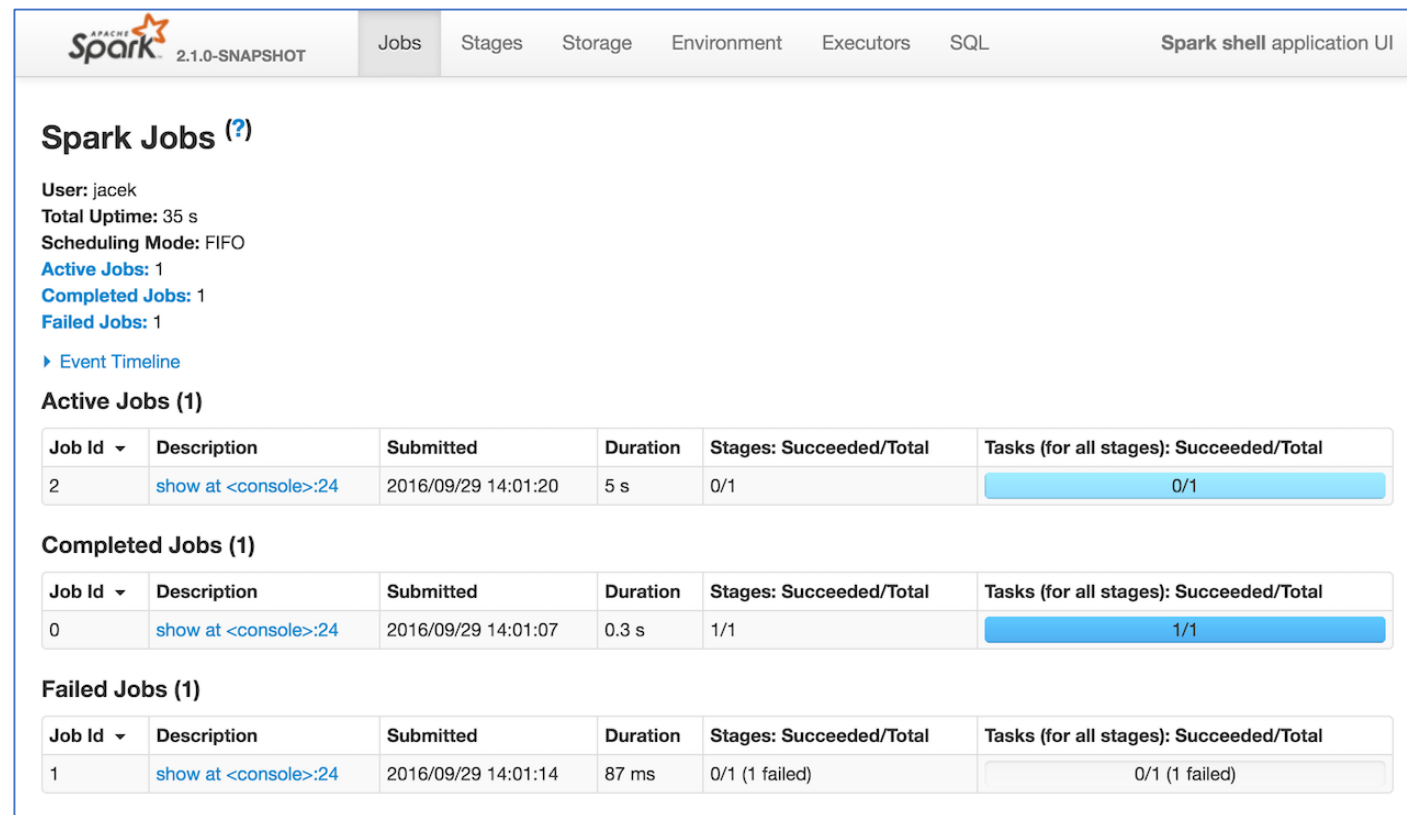




# Spark User Interface

Acesso: <http://localhost:4040>  
(default)

1. Jobs
2. Stages
3. Storage (RDDs Persistidos)
4. Environment (Configurações da App.)
5. Executors (Estado dos executores e do Driver da App).



APACHE Spark 2.1.0-SNAPSHOT

Jobs Stages Storage Environment Executors SQL Spark shell application UI

### Spark Jobs (?)

User: jacek  
Total Uptime: 35 s  
Scheduling Mode: FIFO  
Active Jobs: 1  
Completed Jobs: 1  
Failed Jobs: 1  
[Event Timeline](#)

#### Active Jobs (1)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	<a href="#">show at &lt;console&gt;:24</a>	2016/09/29 14:01:20	5 s	0/1	0/1

#### Completed Jobs (1)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	<a href="#">show at &lt;console&gt;:24</a>	2016/09/29 14:01:07	0.3 s	1/1	1/1

#### Failed Jobs (1)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	<a href="#">show at &lt;console&gt;:24</a>	2016/09/29 14:01:14	87 ms	0/1 (1 failed)	0/1 (1 failed)