

Transformações Narrow e Wide

Atributo	Narrow Transformations	Wide Transformations
Definição	1 part. entrada → 1 saída	N part. entrada → N saída
Exemplos	map, filter, union, select	groupBy, join, distinct
Custo	Baixo: Processamento local	Alto: Exige Shuffle
Dependência	Pipelineable	Barrier

Transformações Narrow e Wide

Dataset de exemplo (dados distribuídos)

Dataset completo

id	valor
1	10
2	20
1	30
2	40

Distribuído em 2 partições

Partição 1

id	valor
1	10
2	20

Partição 2

id	valor
1	30
2	40

1 Exemplo de NARROW (filter)

```
df.filter("valor > 15")
```

- ✓ Cada partição é processada sozinha
- ✓ Nenhum dado sai da partição
- ✓ Nenhuma comunicação entre nós

Resultado em cada partição

Partição 1 (resultado)

id	valor
2	20

Partição 2 (resultado)

id	valor
1	30
2	40

Interpretação

- cada partição → independente
- 1 entrada → 1 saída

✓ NARROW = eficiente por natureza

2 Exemplo de WIDE (groupBy)

```
df.groupBy("id").sum()
```

O que o Spark PRECISA fazer

- Para somar por id, ele precisa juntar:
 - todos os id = 1
 - todos os id = 2
- Mas eles estão espalhados nas partições.

Antes do shuffle (dados espalhados)

Partição 1		Partição 2	
id	valor	id	valor
1	10	1	30
2	20	2	40

Após shuffle (redistribuição)

Partição 1		Partição 2	
id	valor	id	valor
1	10	1	20
1	30	2	40

Após cálculo (resultado final)

Partição 1		Partição 2	
id	soma_valor	id	soma_valor
1	40	2	60

Interpretação

- cada saída depende de várias entradas
- houve redistribuição de dados (shuffle)
- existe comunicação entre nós

✓ WIDE = envolve SHUFFLE (caro)

3 Insight MUITO importante

Se os dados já estivessem assim:

Partição 1		Partição 2	
id	valor	id	valor
1	10	2	20
1	30	2	40



👍 O groupBy("id")
poderia rodar sem shuffle

Regra de ouro

NARROW

- processamento local
- sem shuffle
- mais rápido

WIDE

- redistribuição de dados
- com shuffle
- mais custoso

💡 **Insight final:** Performance em Spark não depende só do código — depende principalmente de como os dados estão distribuídos.

Otimização Spark

Entendendo e reduzindo o gargalo do Shuffle

01

Network I/O



02

Disk Spill



03

Sintomas Críticos



04

Estratégias



Performance

Gestão de Memória: Cache

Cache & Persist



Storage Levels



Unpersist



Melhores Práticas

