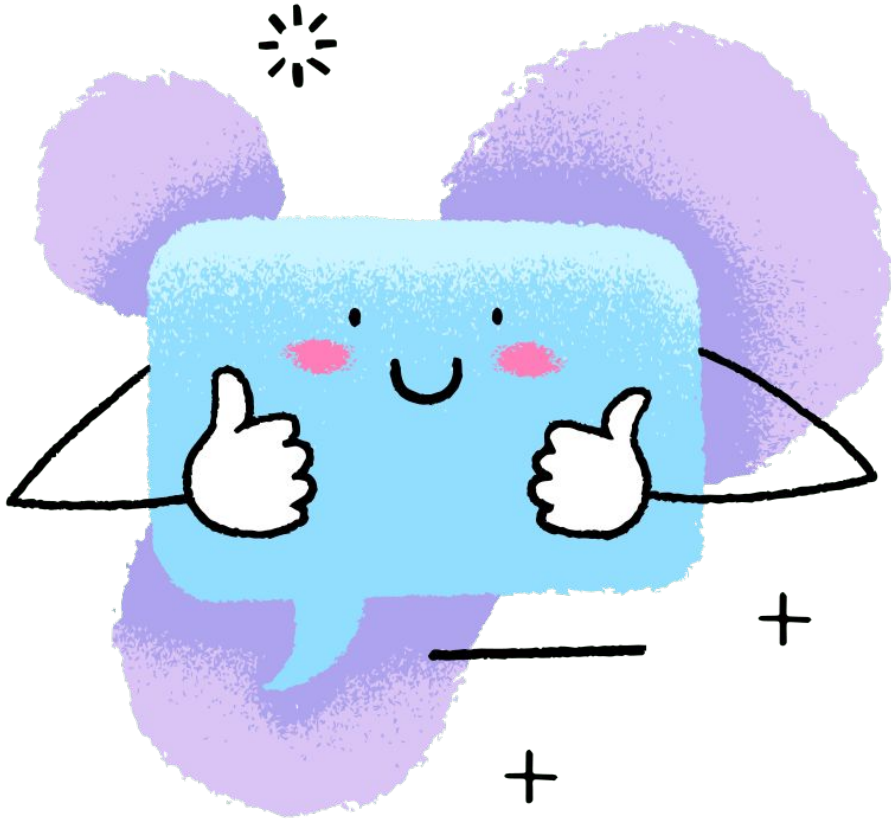
PySpark

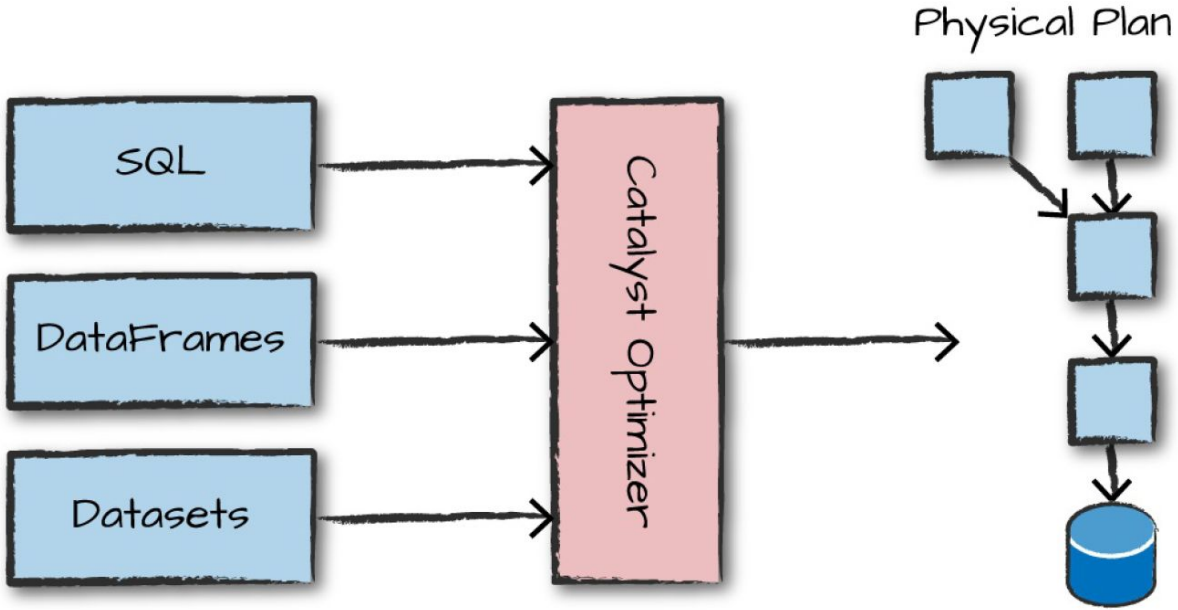


Что будет на уроке

1. Способы оптимизации: push down фильтрация, стратегия исполнения join. 4 этапа SQL-оптимизации
2. Работа со Spark UI
3. Сложные случаи: перекос в объеме данных между экзекьюторами. Функции repartition, coalesce

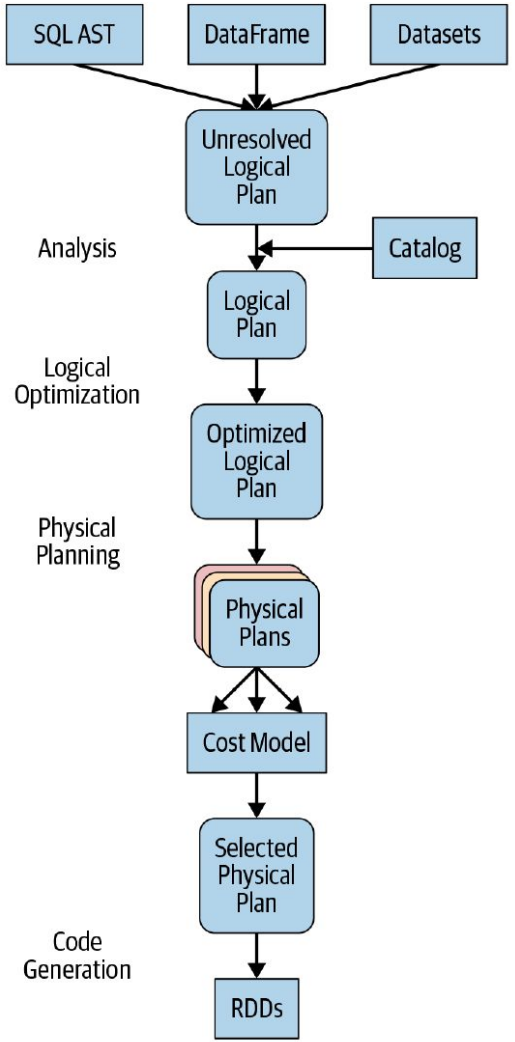
Оптимизатор запросов

Lazy исполнение трансформаций позволяет оптимизировать план



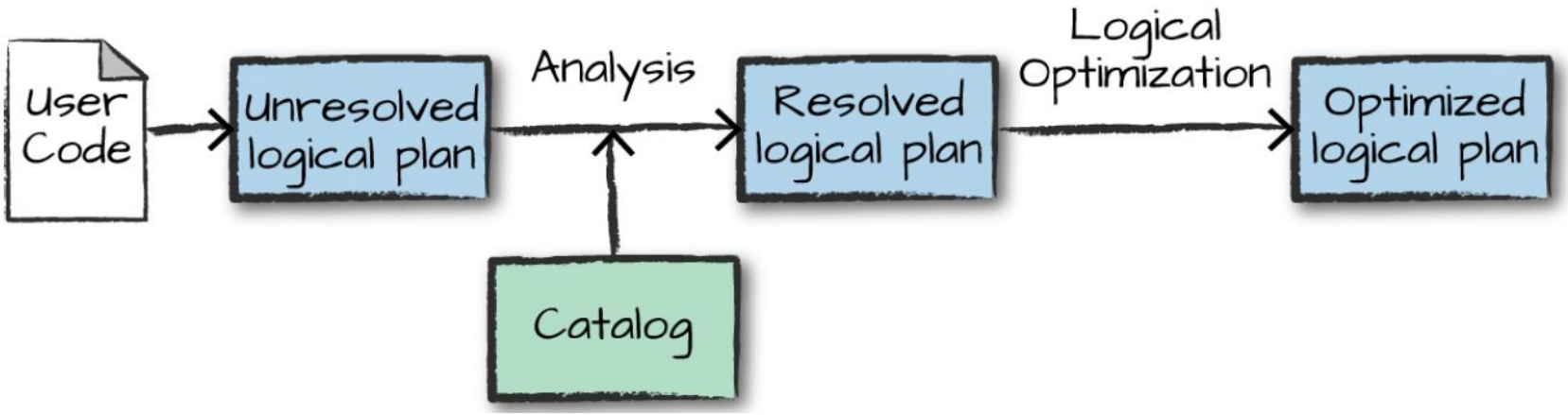
Catalyst Optimizer
transformational phases

1. Analysis
2. Logical optimization
3. Physical planning
4. Code generation



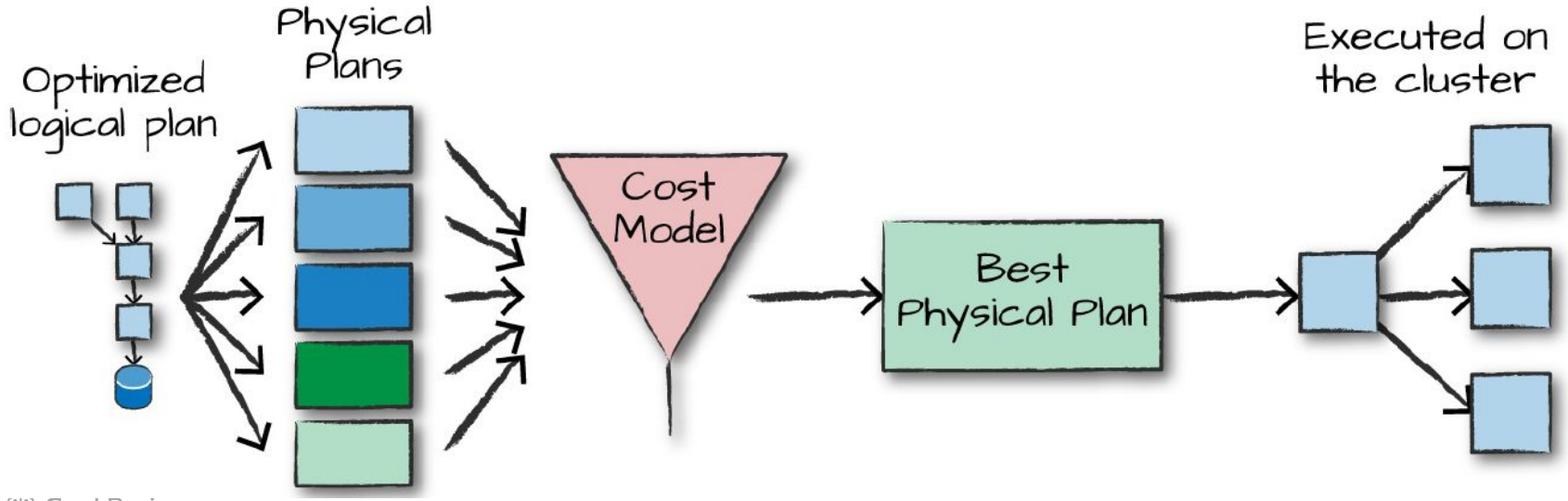
Logical plan

1) проверка наличия таблиц, колонок
2) pushing down predicates or selections

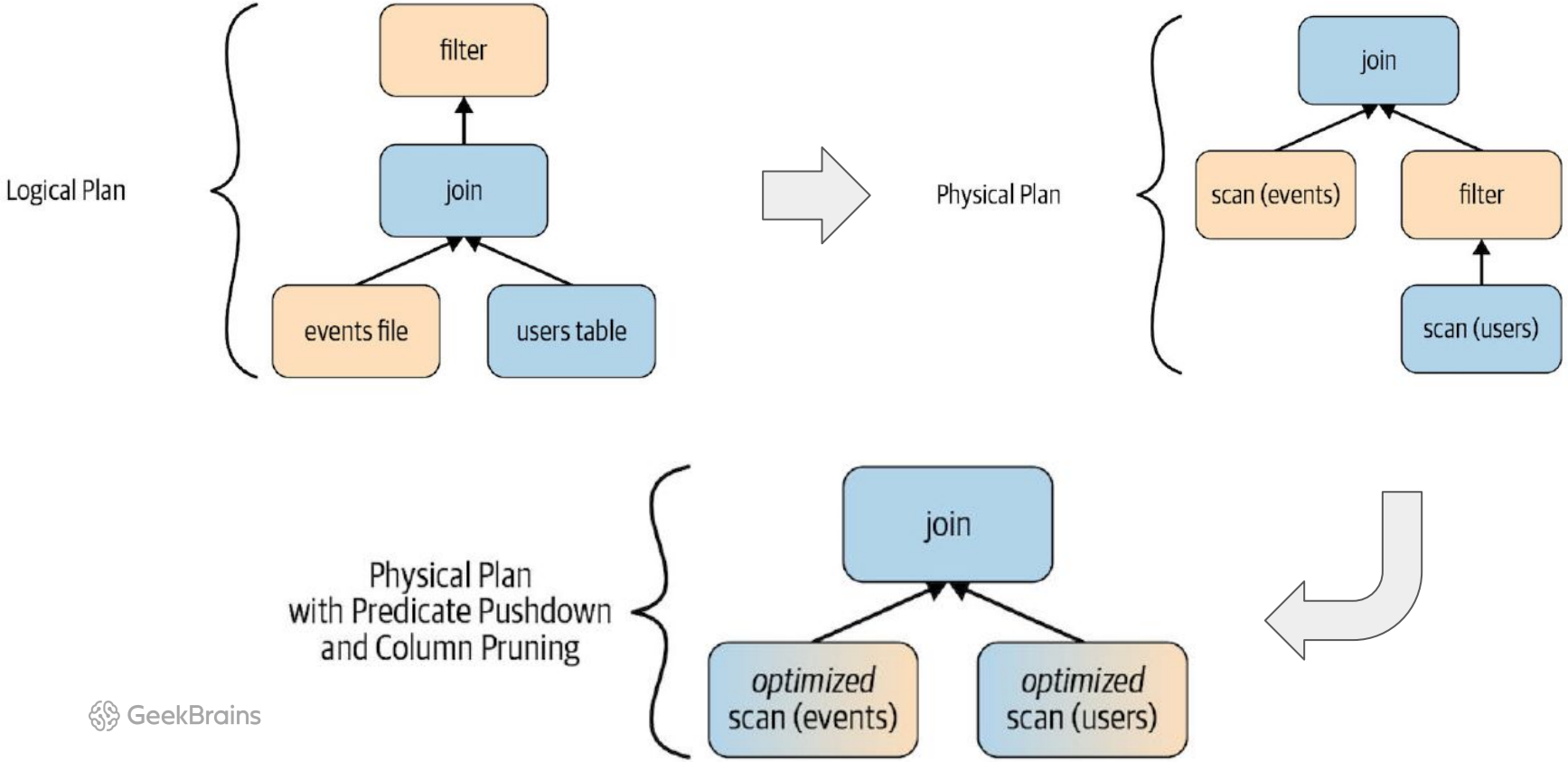


Physical Planning

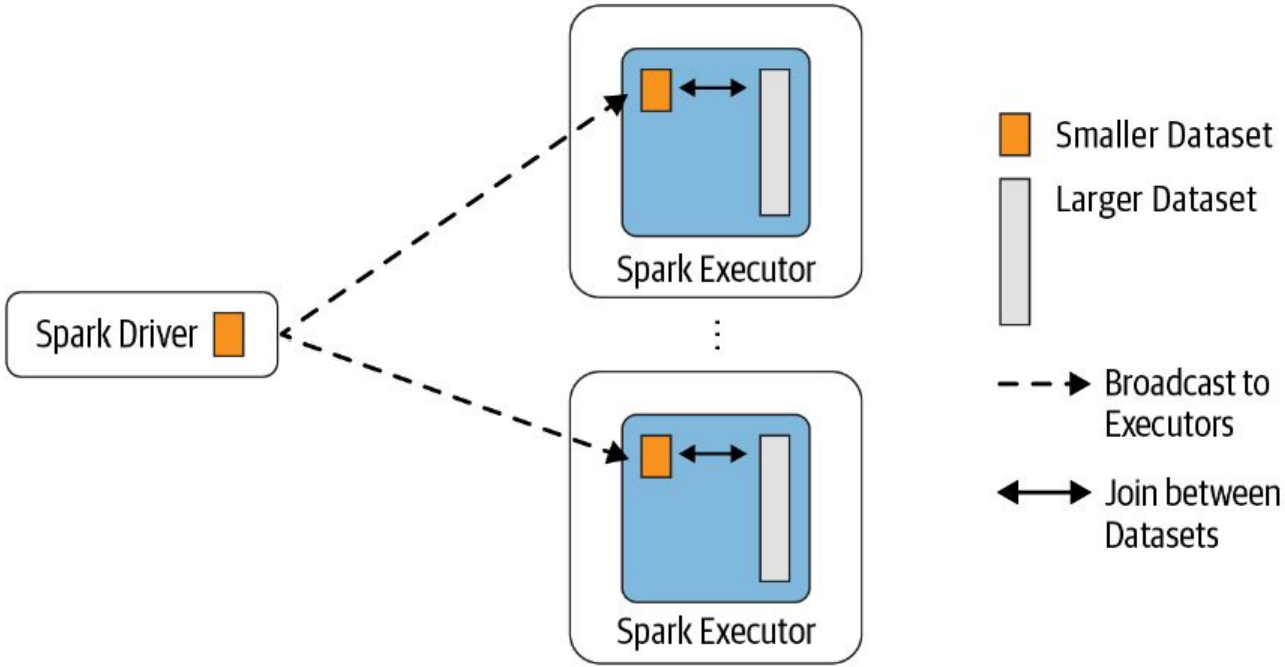
Использует знание о размере и распределении партиций



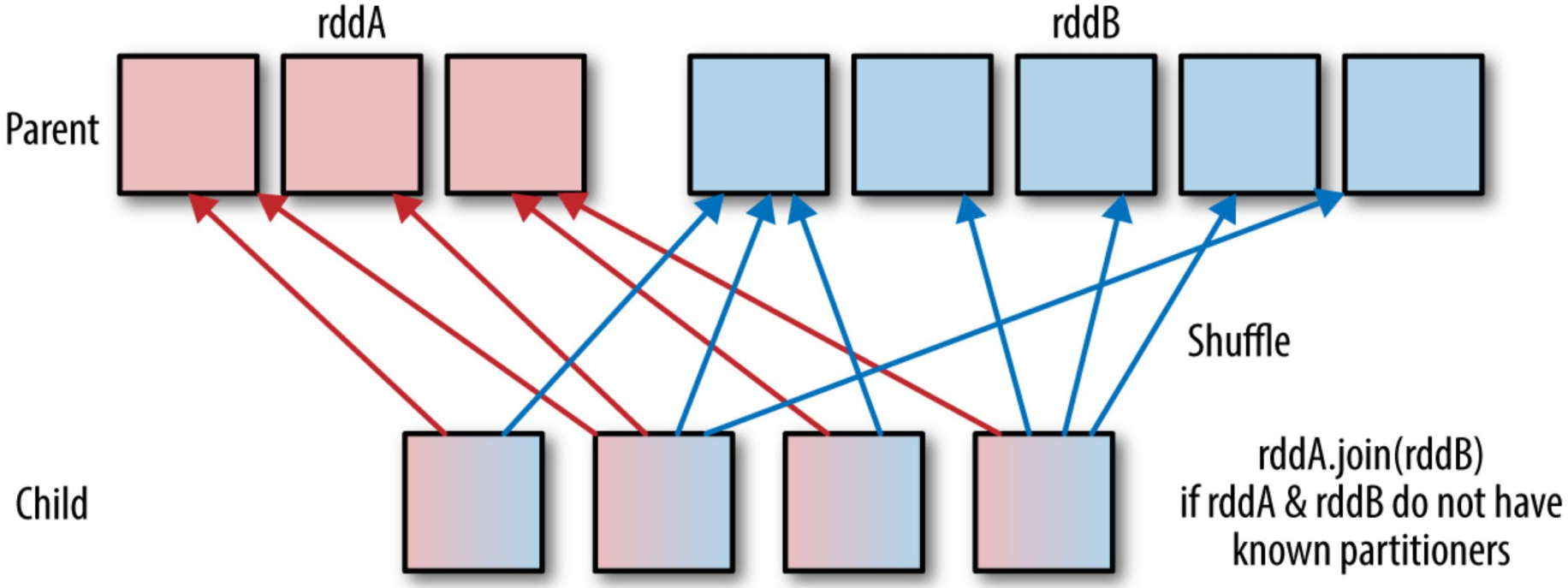
Планы запроса из примера



Broadcast Hash Join



Shuffle Sort Merge Join



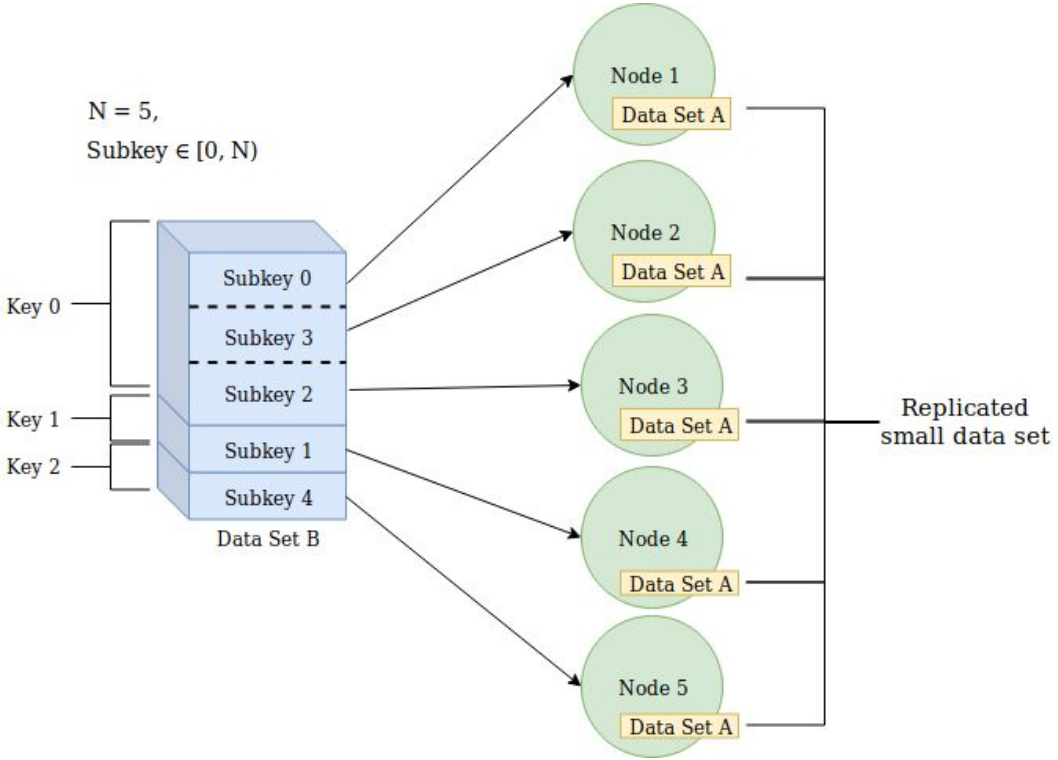
Чем опасны перекосы в данных?

- Bottleneck времени выполнения
- выше риски падений и потери данных

```
rdd_71_1    Memory Deserialized 1x Replicated   1264.7 MB   0.0 B   node4:38759
rdd_71_10   Memory Deserialized 1x Replicated   11.6 MB     0.0 B   node1:58115
rdd_71_11   Memory Deserialized 1x Replicated   25.7 MB     0.0 B   node1:53968
rdd_71_2    Memory Deserialized 1x Replicated   72.6 MB     0.0 B   node4:54133
rdd_71_4    Memory Deserialized 1x Replicated   1260.9 MB   0.0 B   node2:33179
rdd_71_5    Memory Deserialized 1x Replicated   56.8 MB     0.0 B   node2:54222
rdd_71_7    Memory Deserialized 1x Replicated   54.5 MB     0.0 B   node4:34149
rdd_71_8    Memory Deserialized 1x Replicated   1277.8 MB   0.0 B   node1:43572
rdd_71_9    Memory Deserialized 1x Replicated   1255.8 MB   0.0 B   node1:58518
```

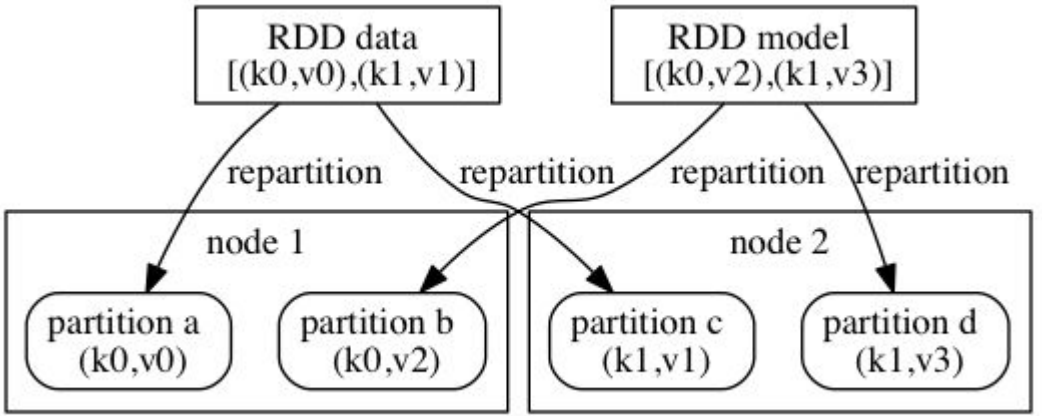
Что делать с перекосом?

- **Солить**
- repartiotion
- coalesce



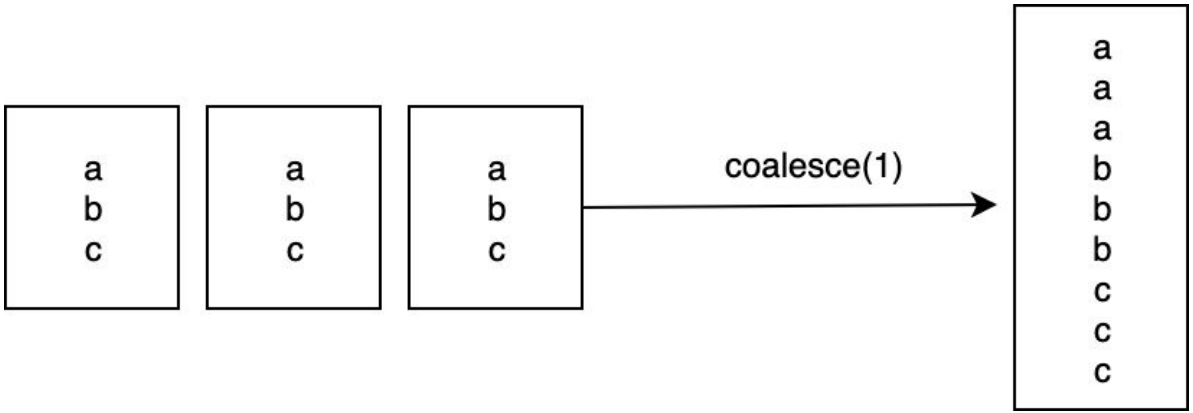
Что делать с перекосом?

- Солить
- **repartition**
- coalesce



Что делать с перекосом?

- Солить
- repartition
- **coalesce**



Спасибо!

Каждый день вы становитесь лучше :)