

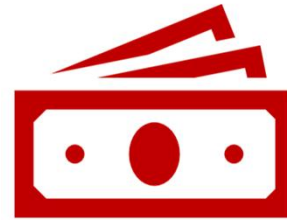
“

Em ambientes
distribuídos,
escrever código
que funciona
não é suficiente —
precisamos
escrever código
eficiente ”

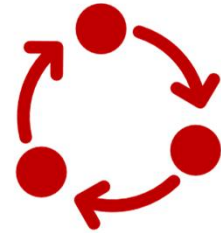


A Importância da Otimização em Big Data

Custo Operacional



Escalabilidade



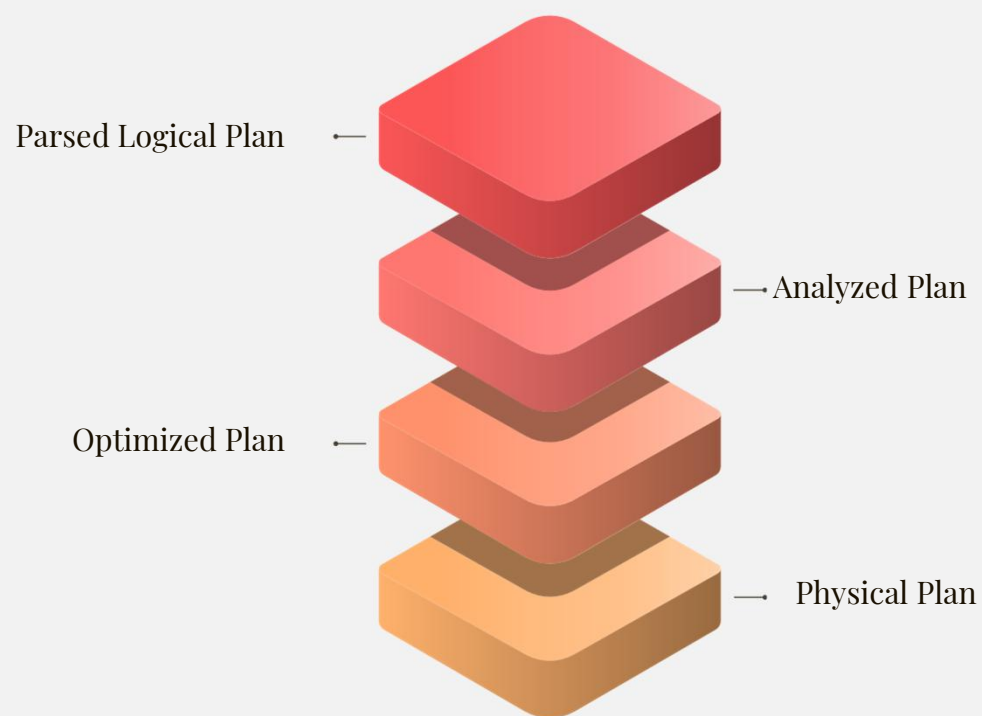
SLA e Latência



Competição por Recursos



Planos de Execução Spark



Analizando performance na prática com comando explain()

Item	Descrição	Por que importa	Buscar ou evitar?	Como aparece no plano	O que causa
 Shuffles (Exchange)	Redistribuição de dados entre nós do cluster	Muito custoso (rede + disco + serialização). Principal gargalo em jobs grandes	 Evitar ao máximo (ou reduzir)	Exchange, ShuffleExchangeExec	groupBy, join, distinct, repartition
 Predicate Pushdown	Filtros aplicados diretamente na fonte de dados	Reduz leitura, memória e tempo de execução	 Garantir que ocorra	PushedFilters: [...]	Filtros em colunas (WHERE, .filter()) em formatos como Parquet/Delta
 WholeStageCodegen	Geração de código Java otimizado em runtime (pipeline de operadores)	Aumenta performance reduzindo overhead de execução	 Ideal que apareça (na maioria dos casos)	WholeStageCodegen, *(1)	Operações SQL/DataFrame nativas (sem UDF)
 Seleção de Join	Algoritmo escolhido para o join	Impacto enorme na performance	 Depende (queremos o mais eficiente)	BroadcastHashJoin, SortMergeJoin, etc	join entre datasets
 Debugging do Plano	Comparação entre plano lógico e físico	Ajuda a entender otimizações e problemas	 Sempre analisar	== Physical Plan ==, == Logical Plan ==	Qualquer transformação

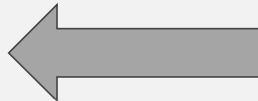
**EXPLAIN SELECT * FROM target t INNER JOIN source s
ON t.dep = s.dep AND t.id = s.id**

-- Plan without Storage Partition Join

== Physical Plan ==

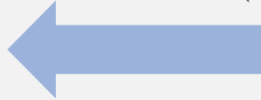
* Project (12)

+ - * SortMergeJoin **Inner** (11)



: - * Sort (5)

: + - Exchange (4)



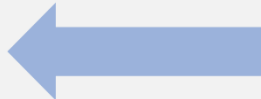
: + - * Filter (3)

: + - * ColumnarToRow (2)

: + - BatchScan (1)

+ - * Sort (10)

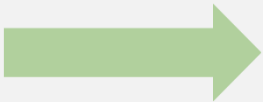
+ - Exchange (9)



+ - * Filter (8)

+ - * ColumnarToRow (7)

+ - BatchScan (6)



Catalyst Optimizer



RBO



CBO



Adaptive Query Execution



AQE: "coalesce shuffle partitions"



SortMergeJoin por BroadcastJoin

