

IMPACTO DOS PEQUENOS ARQUIVOS NO SPARK

Ingestão frequente
(streaming / micro-batch)



Muitas partições
no Spark



Particionamento



Escritas incrementais



COMPACTAÇÃO DE ARQUIVOS COM O COMANDO OPTIMIZE

CONSOLIDAÇÃO



SEGURANÇA ACID



SQL E PYSPARK



AGENDAMENTO



Optimize para Small Files Problem

Dataset de exemplo (problema de pequenos arquivos)

Tabela: vendas (Delta Lake)

id_venda	data	valor
1	01/05	100
2	01/05	120
3	01/05	90
4	01/05	200
5	02/05	150
6	02/05	80
7	02/05	110
8	03/05	130
...		

Arquivos físicos na tabela (muitos arquivos pequenos)

Partição data=2025-05-01: ~1200 arquivos
Partição data=2025-05-02: ~950 arquivos
Partição data=2025-05-03: ~800 arquivos

Problema: muitos arquivos pequenos causam

- Alto overhead no driver e no armazenamento
- Mais tempo de listagem de arquivos
- Menor paralelismo efetivo e menor desempenho

O que o OPTIMIZE faz?

- Combina arquivos pequenos em arquivos maiores (compaction)
- Reduz o número de arquivos lidos
- Melhora a eficiência de leitura e escrita

1 Como funciona o OPTIMIZE

- Lê os arquivos pequenos da tabela
- Combina os dados em arquivos maiores
- Grava novos arquivos grandes otimizados por partição
- Atualiza o log do Delta com os novos arquivos

Partição 1 Partição 2 Partição 3

Vários arquivos pequenos serão lidos

Compaction dos arquivos

Partição 1 Partição 2 Partição 3

512 MB 512 MB 512 MB

Menos arquivos e tamanho ideal

Delta Log

- Remove arquivos antigos
- Adiciona novos arquivos otimizados

DELTA LAKE

2 Antes e depois do OPTIMIZE

Antes do OPTIMIZE

Partição data=2025-05-01: ~1200 arquivos, Total: ~150 MB
Partição data=2025-05-02: ~950 arquivos, Total: ~110 MB
Partição data=2025-05-03: ~800 arquivos, Total: ~90 MB

Depois do OPTIMIZE

Partição data=2025-05-01: 512 MB, 1 arquivo
Partição data=2025-05-02: 512 MB, 1 arquivo
Partição data=2025-05-03: 512 MB, 1 arquivo

Muitos arquivos pequenos = mais overhead, mais I/O, mais tempo

Poucos arquivos grandes = menos overhead, mais eficiência, mais performance

3 Resumo rápido

Aspecto	Antes do OPTIMIZE	Depois do OPTIMIZE
Número de arquivos	Muito alto (milhares)	Baixo (poucos)
Tamanho dos arquivos	Muito pequenos (KBs)	Grandes (ex.: 128MB - 1GB)
Overhead (listar arquivos)	Alto	Baixo
Leitura (I/O)	Ineficiente	Eficiente
Escrita	Gera muitos arquivos	Gera arquivos otimizados
Performance	Mais lenta	Mais rápida
Custo (armazenamento e CPU)	Maior	Menor

Dica de ouro

Rode OPTIMIZE periodicamente para manter a tabela sempre saudável!

Boas práticas:

- Agende OPTIMIZE em horários de baixo uso
- Use ZORDER BY para melhorar ainda mais as consultas (opcional)
- Monitore o tamanho dos arquivos com DESCRIBE DETAIL tabela

Exemplo de comando

- Compactação simples
OPTIMIZE vendas;
-- Combina arquivos pequenos em arquivos grandes
- Compactação com ZORDER (opcional)
OPTIMIZE vendas
ZORDER BY (data);
-- Além de compactar, organiza os dados para acelerar filtros por data
- Verificar o resultado
DESCRIBE DETAIL vendas;
-- Veja o número de arquivos após o optimize

Importante: OPTIMIZE é executado de forma assíncrona e usa recursos do cluster. Ele reescreve os dados, por isso é uma operação de manutenção (não use durante picos de carga).

DELTA LAKE