



Otimizando suas Querys no PostgreSQL

XIV Fórum Goiano de Software Livre
Jean Pierre Monteiro Queiroz



Agenda

- **Otimizador/Planejador**
- **Analyze**
- **Ferramentas de análise**
- **Índices**
- **Paralelismo**
- **Dicas Consultas Eficientes**

Otimizador/Planejador

- O que é o plano de execução ?

- Explain SELECT pessoa.nome,pessoa.cpf FROM pessoa WHERE pessoa.nome ILIKE '%pedro%'

Seq Scan on pessoa (cost=0.00..2346.66 rows=492 width=39)

Filter: ((nome)::text ~~* '%pedro%'::text)

- Explain Analyze SELECT pessoa.nome,pessoa.cpf FROM pessoa WHERE pessoa.nome ILIKE '%pedro%'

Seq Scan on pessoa (cost=0.00..2346.66 rows=492 width=39) (actual time=26.338..158.150 rows=423 loops=1)

Filter: ((nome)::text ~~* '%pedro%'::text)

Rows Removed by Filter: 48670

Planning time: 0.666 ms

Execution time: 158.259 ms

Analyze

- Analyze
- Vacuum Analyze



Como identificar Query's lentas ?

- pgBadger
- pg_stat_statements

1 - Adicione a configuração abaixo no postgresql.conf:

```
shared_preload_libraries = 'pg_stat_statements'
```

```
# Increase the max size of the query strings Postgres records
```

```
track_activity_query_size = 2048
```

```
# Track statements generated by stored procedures as well
```

```
pg_stat_statements.track = all
```

2 - reiniciar o serviço do postgresQL

3 - CREATE EXTENSION pg_stat_statements;

Índices

- **Busca sequencial vs Busca indexada**
- **Índice Btree**
- **Índice GIN**
- **O planejador opta por realizar uma busca sequencial em tabelas menores.**
- **O planejador utilizará o índice apenas quando sua Query's for seletiva. Se você criar um índice em uma coluna situação por exemplo, que contemple 50 % dos registros, o planejador irá realizar uma busca sequencial em toda a tabela.**

Consultas paralelas no PostgreSQL

- `max_parallel_workers_per_gather`
- Nível de Banco de Dados
- Nível de Sessão
- Sequential Scan Parallelisation
- Parallel Aggregate Functions
- Parallel Joins



Dicas de melhores práticas

- O planejador altera a ordem dos joins conforme o parâmetro **join_collapse_limit** que por padrão é 8. Caso a query realize até 8 joins o planejador irá alterar a ordem de acordo com o melhor plano. Não é aconselhado alterar esse valor, caso a query tenha mais de 8 uma dica é colocar primeiro as tabelas que sofre filtros no predicado.
- EXISTS vs IN
- WITH Queries (Common Table Expressions).
- Para carga de dados utilize o COPY e não o INSERT.

 Até aqui é isso

Dúvidas ?



jean.aparecida@gmail.com



<https://twitter.com/jpierremq>



<https://github.com/jeanpierremq>