Tecnologias de Banco de Dados

Msc. Eliezio Soares
eliezio.soares@ifrn.edu.br

INSTITUTO FEDERAL
Rio Grande do Norte
Campus
Currais Novos

Arquitetura e Instalação de Banco de Dados

- Características de SGBDs;
- Log de Transação;
- Write-Ahead Log;
- Checkpoint;
- Transação ACID;
- Shared Memory;
- Shared Buffer;
- Arquitetura Multiprocessos;
- Backend;
- Page Cache;

Banco de Dados e SGBD

Banco de Dados: é um conjunto de dados relacionados, representando um pedaço ou interpretação do mundo real, que possui uma estrutura lógica com significado inerente e comumente possui uma finalidade e usuários específicos.

(CAIUT, Fábio. 2015)

Sistema Gerenciador de Bancos de Dados (SGBD): é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados.
O principal objetivo de um SGBD é proporcionar uma forma de armazenar e recuperar informações de um banco de dados de maneira conveniente e eficiente.

(Silberschatz. 2012)

Características de um SGBD

- Independência de dados;
- Segurança mais especializada:;
- Acesso concorrente e compartilhamento dos dados;
- Restrições de Integridade;
- Recuperação de Falhas;
- Manipular grande quantidade de dados;
- Diminui o tempo de desenvolvimento de aplicações.

Você é responsável pelo desenvolvimento de um sistema de vendas. Como resolveria a seguinte situação?

Uma grande venda é feita, com mais de 200 itens. O registro representando o pedido é gravado, mas após gravar 50 itens ocorre uma queda de energia. O que deve ocorrer depois que o sistema voltar ao ar?

Propriedades ACID

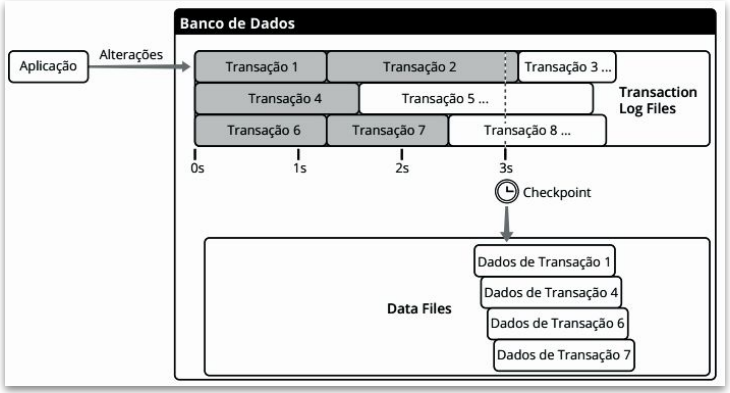
- **Atomicidade**
- **C**onsistência
- **I**solamento
- **D**urabilidade

Para garantir a atomicidade das transações, o SGBD utiliza o Log de Transações.

Características de um SGBD

- Quando há uma transação (update, insert, delete) as alterações são feitas nos arquivos de log de transação.
- Apenas as transações bem sucedidas (que efetuaram um COMMIT) são efetivadas nos arquivos de dados.

Figura 1 - Processamento de Transações em SGBD.



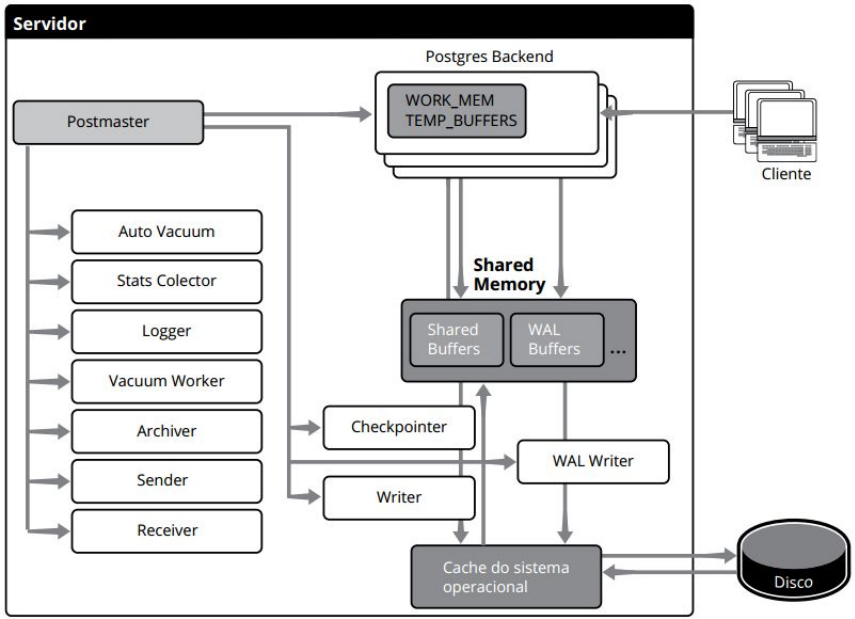
Fonte: Caiut, Fábio. 2015.

Características do PostgreSQL

- Multi-Version Concurrency Control (MVCC)
- Point in Time Recovery (PITR)
- Tablespaces
- Replicação Síncrona e Assíncrona
- Transações Aninhadas (Savepoint)
- Backup Online
- Otimizador de Queries
- Log de Transações (WAL)

Arquitetura PostgreSQL

Figura 2 - Arquitetura do PostgreSQL



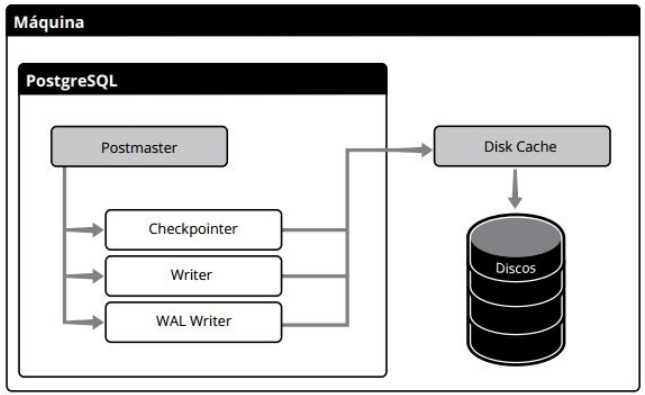
Fonte: Caiut, Fábio. 2015.

- Sistema **multiprocessos**: Para cada conexão, um processo é criado no SO para servir a esse cliente.
  - Cada processo é chamado de **backend**.
- Além dos backends, existem outros processos para atender as tarefas necessárias para o SGBD.

Arquiterura PostgreSQL

- **Postmaster** é o processo principal que inicia todos os demais.
  - **Checkpointer** é responsável por disparar a operação de Checkpoint aplicando as alterações do WAL para os arquivos de dados ao atingir um intervalo de tempo ou o limite do WAL.
  - **Write Ahead Log (WAL)** é como se conhece o log de transação. É responsável por gravar em disco as alterações dos buffers do WAL em intervalos pré-definidos.

Figura 3 - Processos mínimos.

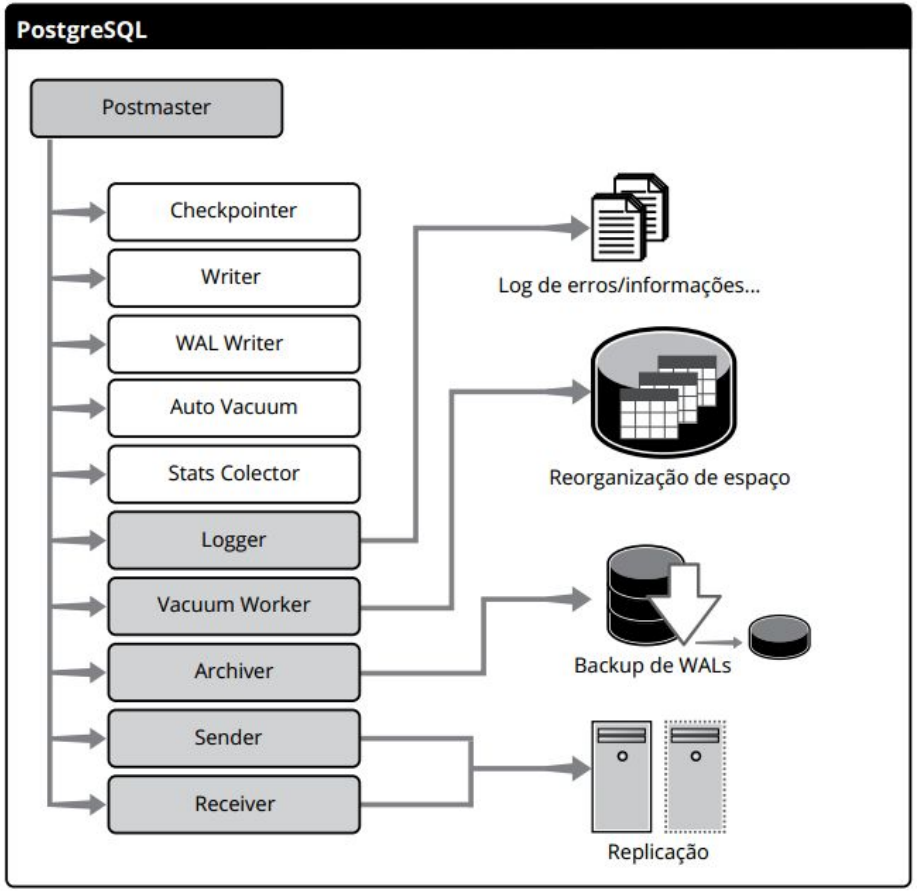


Fonte: Caiut, Fábio. 2015.

Arquiterura PostgreSQL

- **Postmaster** é o processo principal que inicia todos os demais.
  - **Writer (ou background writer / bgwriter)** procura por páginas de dados modificadas na memória (shared_buffer) e as escreve para o disco em lotes pequenos.
  - **AutoVacuum** automatiza a execução de Vacuum.
  - **Stats Collector** é responsável por coletar estatísticas de uso do servidor relacionadas a tabelas, índices, blocos em discos etc.

Figura 4 - Todos os processor utilitários.



Fonte: Caiut, Fábio. 2015.

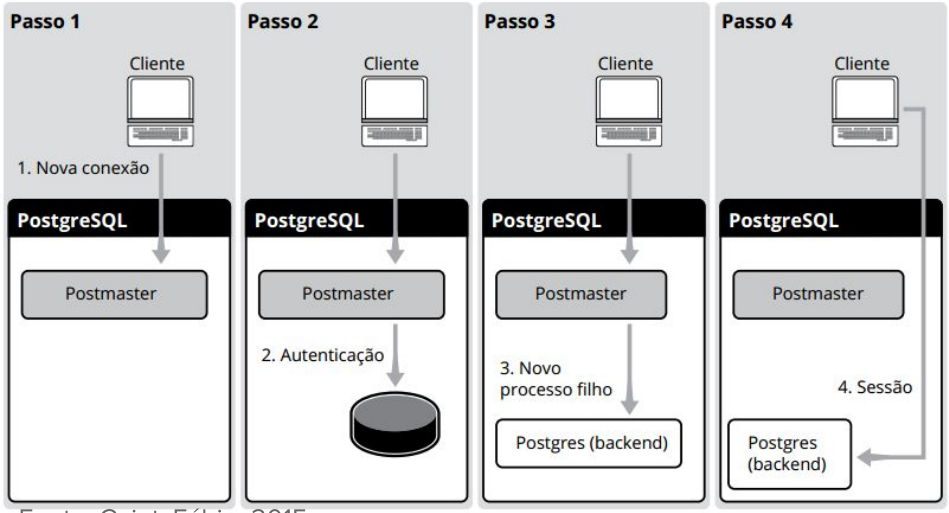
- **Logger** é responsável por registrar o que acontece no PostgreSQL.
- **Vacuum Worker** é o que de fato executa o procedimento de Vacuum.
- **Archiver** realiza o backup dos segmentos de WAL que já foram completamente preenchidos.
- **Sender** e **Receiver** permitem o recurso de replicação binária.

Arquiterura PostgreSQL

Novos processos são criados para cada conexão ao SGBD.

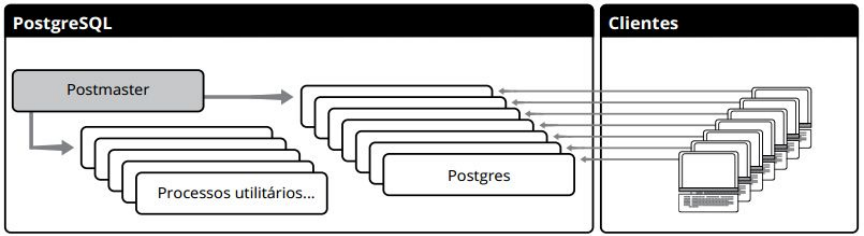
A partir de então cada backend lida diretamente com o cliente, sem intermédio do Postmaster.

Figura 5 - Conexão com clientes e criação de Backends.



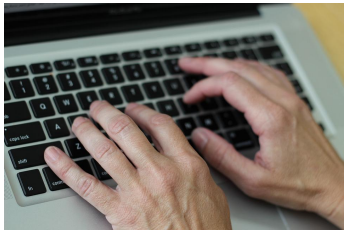
Fonte: Caiut, Fábio. 2015.

Figura 6 - Backends atendendo clientes.



Fonte: Caiut, Fábio. 2015.

Mãos a obra!

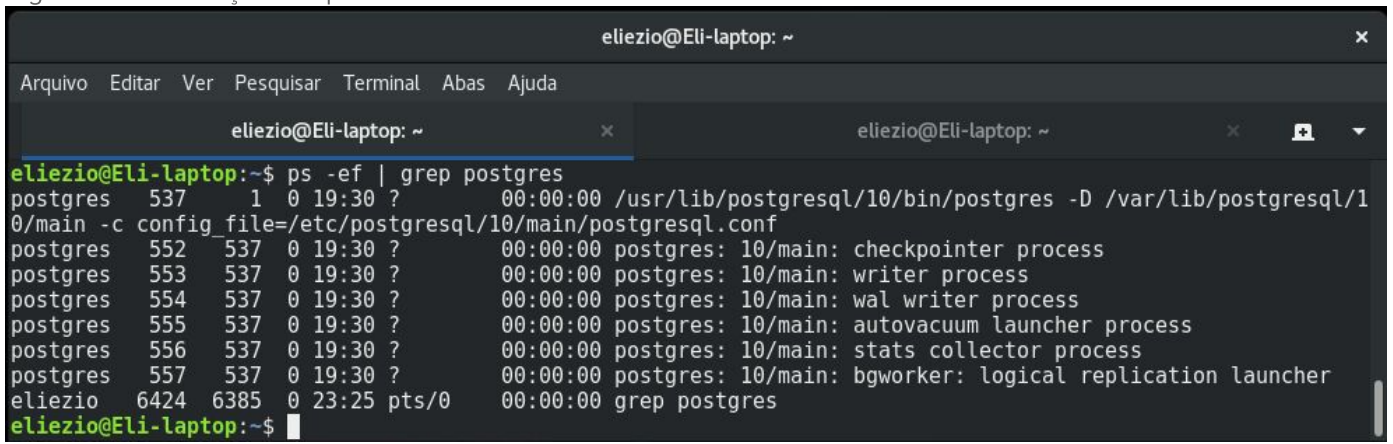


Em um terminal linux, digite o comando **ps -ef | grep postgres**

ps: mostra os processos em execução
-e: exibe todos os processos
-f: exibe detalhes de cada linha
| grep postgres: filtra os processos com nome postgres

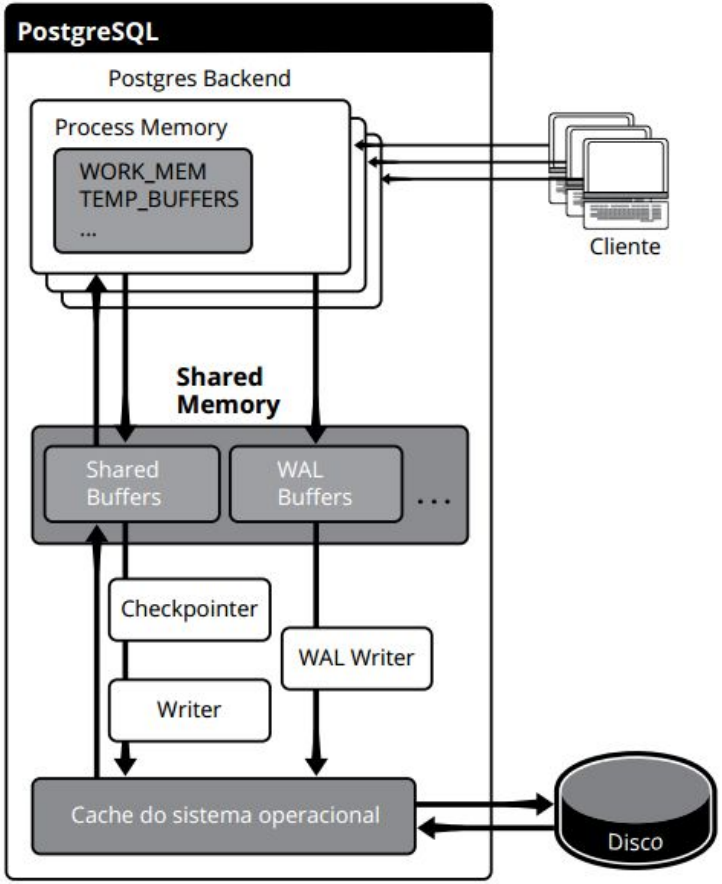
Tente iniciar uma nova conexão e verifique se um backend é gerado no servidor.

Figura 7 - Visualização dos processos em um terminal linux.



Fonte: Elaborado pelo autor.

Figura 8 - Arquitetura de Memória no PostgreSQL.



Fonte: Caiut, Fábio. 2015.

O Sistema Operacional normalmente aloca um segmento de memória exclusivamente para um novo processo.

- **Shared Memory** é uma área compartilhável entre processos postgres.
  - **Shared Buffer** é o que de fato executa o procedimento de Vacuum.
  - **Wal Buffer** utilizado pelo Wal Writer.
- **Cache do SO** (kernel buffer cache, disk cache, page cache) intermedia toda a operação de E/S o recurso de replicação binária, pois o kernel sempre o utiliza para realizar as operações.

Bibliografia Utilizada

CAIUT, Fábio. Administração de banco de dados. 1ª Edição. Rio de Janeiro. RNP/ESR, 2015.

ELMASRI, Ramez. NAVATHE, Shamkant B. Sistemas de Banco de Dados. Projeto de Banco de Dados. 6ª Edição. São Paulo. Pearson Addison Wesley, 2011.