



Configuração do banco de dados

Enterprise applications

NetApp

February 11, 2026

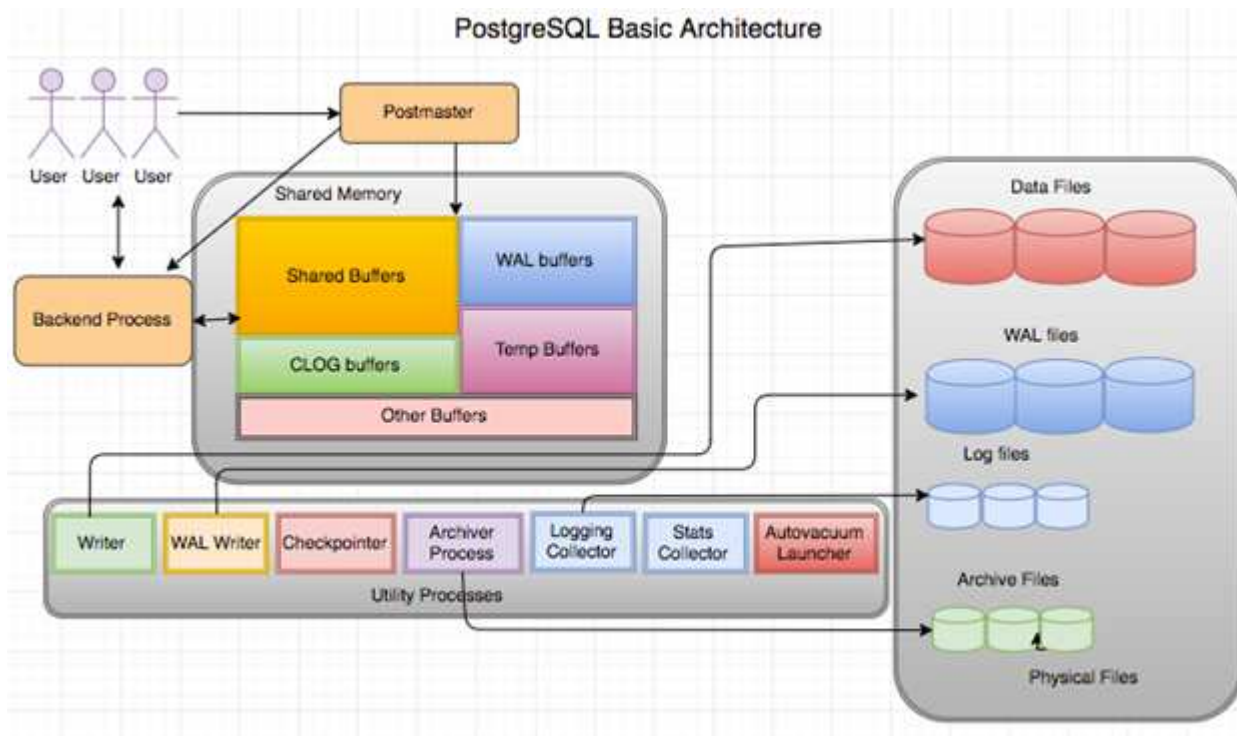
Índice

- Configuração do banco de dados 1
 - Arquitetura 1
 - Parâmetros de inicialização 2
- Definições 2
 - BRINDE 4
 - VÁCUO 4
- Tablespaces 4

Configuração do banco de dados

Arquitetura

PostgreSQL é um RDBMS baseado na arquitetura cliente e servidor. Uma instância do PostgreSQL é conhecida como um cluster de banco de dados, que é uma coleção de bancos de dados em vez de uma coleção de servidores.



Há três elementos principais em um banco de dados PostgreSQL: O postmaster, o front-end (cliente) e o back-end. O cliente envia solicitações para o postmaster com informações como protocolo IP e a que banco de dados se conectar. O postmaster autentica a conexão e a passa para o processo de back-end para comunicação adicional. O processo back-end executa a consulta e envia resultados diretamente para o front-end (cliente).

Uma instância do PostgreSQL é baseada em um modelo multiprocesso em vez de um modelo multithread. Ele gera vários processos para diferentes trabalhos, e cada processo tem sua própria funcionalidade. Os principais processos incluem o processo do cliente, o processo do escritor DE WAL, o processo do escritor de fundo, e o processo do checkpointer:

- Quando um processo de cliente (primeiro plano) envia solicitações de leitura ou gravação para a instância do PostgreSQL, ele não lê ou grava dados diretamente no disco. Primeiro, ele armazena os dados em buffers compartilhados e buffers de log write-ahead (WAL).
- Um processo de escritor WAL manipula o conteúdo dos buffers compartilhados e buffers WAL para escrever nos logs DO WAL. Os logs WAL são normalmente Registros de transações do PostgreSQL e são sequencialmente escritos. Portanto, para melhorar o tempo de resposta do banco de dados, o PostgreSQL primeiro grava nos logs de transação e reconhece o cliente.
- Para colocar o banco de dados em um estado consistente, o processo de gravação em segundo plano verifica o buffer compartilhado periodicamente para páginas sujas. Em seguida, ele limpa os dados para os arquivos de dados armazenados em volumes NetApp ou LUNs.

- O processo de checkpointer também é executado periodicamente (com menos frequência do que o processo em segundo plano) e impede qualquer modificação nos buffers. Ele sinaliza para o processo de gravação WAL para escrever e lavar o Registro do ponto de verificação para o final dos logs DO WAL que são armazenados no disco NetApp. Ele também sinaliza o processo de gravação em segundo plano para escrever e lavar todas as páginas sujas para o disco.

Parâmetros de inicialização

Você cria um novo cluster de banco de dados usando o `initdb` programa. `initdb` Um script cria os arquivos de dados, tabelas do sistema e bancos de dados de modelos (`template0` e `template1`) que definem o cluster.

A base de dados de modelos representa uma base de dados de stock. Ele contém definições para tabelas de sistema, exibições padrão, funções e tipos de dados. `pgdata` atua como um argumento para o `initdb` script que especifica a localização do cluster do banco de dados.

Todos os objetos de banco de dados no PostgreSQL são gerenciados internamente pelos respectivos OIDs. Tabelas e índices também são gerenciados por OIDs individuais. As relações entre objetos de banco de dados e seus respectivos OIDs são armazenadas em tabelas de catálogo de sistema apropriadas, dependendo do tipo de objeto. Por exemplo, OIDs de bancos de dados e tabelas de heap são armazenados em `pg_database` e `pg_class`, respectivamente. Você pode determinar os OIDs emitindo consultas no cliente PostgreSQL.

Cada banco de dados tem suas próprias tabelas individuais e arquivos de índice que são restritos a 1GB. Cada tabela tem dois arquivos associados, sufixos respectivamente com `_fsm` e `_vm`. Eles são referidos como o mapa de espaço livre e o mapa de visibilidade. Esses arquivos armazenam as informações sobre a capacidade de espaço livre e têm visibilidade em cada página no arquivo de tabela. Os índices têm apenas mapas de espaço livre individuais e não têm mapas de visibilidade.

O `pg_xlog/pg_wal` diretório contém os registros de escrita antecipada. Os logs write-ahead são usados para melhorar a confiabilidade e o desempenho do banco de dados. Sempre que você atualizar uma linha em uma tabela, o PostgreSQL primeiro grava a alteração no log write-ahead e, mais tarde, grava as modificações nas páginas de dados reais em um disco. O `pg_xlog` diretório geralmente contém vários arquivos, mas `initdb` cria apenas o primeiro. Arquivos extras são adicionados conforme necessário. Cada arquivo xlog tem 16MBcm de comprimento.

Definições

Existem várias configurações de ajuste do PostgreSQL que podem melhorar o desempenho.

Os parâmetros mais utilizados são os seguintes:

- `max_connections` = <num>: O número máximo de conexões de banco de dados a ter ao mesmo tempo. Use este parâmetro para restringir a troca para o disco e eliminar o desempenho. Dependendo do requisito do aplicativo, você também pode ajustar esse parâmetro para as configurações do pool de conexões.
- `shared_buffers` = <num>: O método mais simples para melhorar o desempenho do seu servidor de banco de dados. O padrão é baixo para a maioria dos hardwares modernos. Ele é definido durante a implantação para aproximadamente 25% da RAM disponível no sistema. Essa configuração de parâmetro varia dependendo de como ele funciona com instâncias de banco de dados específicas; você pode ter que

aumentar e diminuir os valores por tentativa e erro. No entanto, a configuração alta pode degradar o desempenho.

- `effective_cache_size = <num>`: Este valor diz ao otimizador do PostgreSQL quanta memória o PostgreSQL tem disponível para armazenar dados em cache e ajuda a determinar se deve usar um índice. Um valor maior aumenta a probabilidade de usar um índice. Este parâmetro deve ser definido para a quantidade de memória alocada para `shared_buffers` mais a quantidade de cache de SO disponível. Muitas vezes, esse valor é mais de 50% da memória total do sistema.
- `work_mem = <num>`: Este parâmetro controla a quantidade de memória a ser usada em operações de classificação e tabelas de hash. Se você fizer uma classificação pesada em seu aplicativo, talvez seja necessário aumentar a quantidade de memória, mas tenha cuidado. Não é um parâmetro amplo do sistema, mas um parâmetro por operação. Se uma consulta complexa tiver várias operações de ordenação nela, ela usará várias unidades de memória `work_mem`, e vários back-ends poderiam estar fazendo isso simultaneamente. Essa consulta pode muitas vezes levar o servidor de banco de dados a trocar se o valor for muito grande. Essa opção foi anteriormente chamada `sort_mem` em versões mais antigas do PostgreSQL.
- `fsync = <boolean> (on or off)`: Este parâmetro determina se todas as suas páginas WAL devem ser sincronizadas com o disco usando `fsync()` antes que uma transação seja confirmada. Desligá-lo às vezes pode melhorar o desempenho de gravação e ativá-lo aumenta a proteção contra o risco de corrupção quando o sistema trava.
- `checkpoint_timeout`: O processo de checkpoint elimina dados comprometidos no disco. Isso envolve muitas operações de leitura/gravação no disco. O valor é definido em segundos e valores mais baixos diminuem o tempo de recuperação de falhas e o aumento de valores pode reduzir a carga nos recursos do sistema reduzindo as chamadas de ponto de verificação. Dependendo da criticidade do aplicativo, uso, disponibilidade do banco de dados, defina o valor de `checkpoint_timeout`.
- `commit_delay = <num>` E `commit_siblings = <num>`: essas opções são usadas em conjunto para ajudar a melhorar o desempenho, escrevendo várias transações que estão cometendo de uma só vez. Se houver vários objetos `commit_siblings` ativos no momento em que a transação está sendo feita, o servidor aguarda por `commit_delay` microssegundos para tentar cometer várias transações de uma só vez.
- `max_worker_processes` / `max_parallel_workers`: Configurar o número ideal de trabalhadores para processos. `Max_Parallel_workers` corresponde ao número de CPUs disponíveis. Dependendo do design do aplicativo, as consultas podem exigir um número menor de trabalhadores para operações paralelas. É melhor manter o valor para ambos os parâmetros o mesmo, mas ajustar o valor após o teste.
- `random_page_cost = <num>`: Este valor controla a forma como o PostgreSQL visualiza leituras de discos não sequenciais. Um valor maior significa que o PostgreSQL é mais provável de usar uma varredura sequencial em vez de uma varredura de índice, indicando que seu servidor tem discos rápidos. Modificar esta configuração depois de avaliar outras opções como otimização baseada em plano, aspiração, indexação para alterar consultas ou esquema.
- `effective_io_concurrency = <num>`: Este parâmetro define o número de operações de e/S de disco simultâneas que o PostgreSQL tenta executar simultaneamente. Aumentar esse valor aumenta o número de operações de e/S que qualquer sessão individual do PostgreSQL tenta iniciar em paralelo. O intervalo permitido é de 1 a 1.000, ou zero para desativar a emissão de solicitações de e/S assíncronas. Atualmente, essa configuração afeta somente digitalizações de heap bitmap. As unidades de estado sólido (SSDs) e outro storage baseado em memória (NVMe) costumam processar muitas solicitações simultâneas. Assim, o melhor valor pode ser nas centenas.

Consulte a documentação do PostgreSQL para obter uma lista completa dos parâmetros de configuração do PostgreSQL.

BRINDE

TOAST representa a técnica de armazenamento de atributos oversized. PostgreSQL usa um tamanho de página fixo (geralmente 8KB) e não permite que tuplas abranjam várias páginas. Portanto, não é possível armazenar grandes valores de campo diretamente. Quando você tenta armazenar uma linha que excede esse tamanho, O TOAST divide os dados de colunas grandes em "pedaços" menores e os armazena em uma MESA DE BRINDE.

Os grandes valores dos atributos tostados são retirados (se selecionados) apenas no momento em que o conjunto de resultados é enviado ao cliente. A tabela em si é muito menor e pode caber mais linhas no cache de buffer compartilhado do que poderia sem qualquer armazenamento fora de linha (TOAST).

VÁCUO

Na operação normal do PostgreSQL, tuplas que são excluídas ou tornadas obsoletas por uma atualização não são removidas fisicamente de sua tabela; elas permanecem presentes até que O VÁCUO seja executado. Portanto, você deve executar O VÁCUO periodicamente, especialmente em tabelas atualizadas com frequência. O espaço que ocupa deve ser recuperado para reutilização por novas linhas, para evitar a interrupção do espaço em disco. No entanto, ele não retorna o espaço para o sistema operacional.

O espaço livre dentro de uma página não é fragmentado. VACUUM reescreve todo o bloco, empacotando eficientemente as linhas restantes e deixando um único bloco contíguo de espaço livre em uma página.

Em contraste, O VACUUM FULL compacta ativamente as tabelas escrevendo uma versão completamente nova do arquivo de tabela sem espaço morto. Esta ação minimiza o tamanho da mesa, mas pode levar muito tempo. Ele também requer espaço em disco extra para a nova cópia da tabela até que a operação seja concluída. O objetivo do VÁCUO de rotina é evitar a atividade TOTAL DO VÁCUO. Este processo não só mantém as tabelas em seu tamanho mínimo, mas também mantém o uso em estado estável do espaço em disco.

Tablespaces

Dois espaços de tablespaces são criados automaticamente quando o cluster de banco de dados é inicializado.

O `pg_global` espaço de tabela é usado para catálogos de sistemas compartilhados. O `pg_default` espaço de tabela é o espaço de tabela padrão dos bancos de dados `template1` e `template0`. Se a partição ou o volume em que o cluster foi inicializado ficar sem espaço e não puder ser estendido, um espaço de tabela pode ser criado em uma partição diferente e usado até que o sistema possa ser reconfigurado.

Um índice que é muito usado pode ser colocado em um disco rápido e altamente disponível, como um dispositivo de estado sólido. Além disso, uma tabela que armazena dados arquivados que raramente são usados ou não críticos ao desempenho pode ser armazenada em um sistema de disco mais lento e menos caro, como unidades SAS ou SATA.

As tablespaces são uma parte do cluster do banco de dados e não podem ser tratadas como uma coleção autônoma de arquivos de dados. Eles dependem dos metadados contidos no diretório de dados principal e, portanto, não podem ser anexados a um cluster de banco de dados diferente ou fazer backup individualmente. Da mesma forma, se você perder um espaço de tabela (por meio da exclusão de arquivos, falha de disco, etc.), o cluster do banco de dados pode se tornar ilegível ou incapaz de iniciar. Colocar um espaço de tabela em um sistema de arquivos temporário como um disco RAM corre o risco de a confiabilidade de todo o cluster.

Depois que ele é criado, um espaço de tabela pode ser usado a partir de qualquer banco de dados se o

usuário solicitante tiver Privileges suficiente. PostgreSQL usa links simbólicos para simplificar a implementação de espaços de tabela. O PostgreSQL adiciona uma linha à `pg_tablespace` tabela (uma tabela abrangente) e atribui um novo identificador de objeto (OID) a essa linha. Finalmente, o servidor usa o OID para criar um link simbólico entre o cluster e o diretório fornecido. O diretório `$PGDATA/pg_tblspc` contém links simbólicos que apontam para cada um dos espaços de tablespaces não incorporados definidos no cluster.

Informações sobre direitos autorais

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES DOCUMENTOS, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.