



데이터베이스 구성

Enterprise applications

NetApp
May 03, 2024

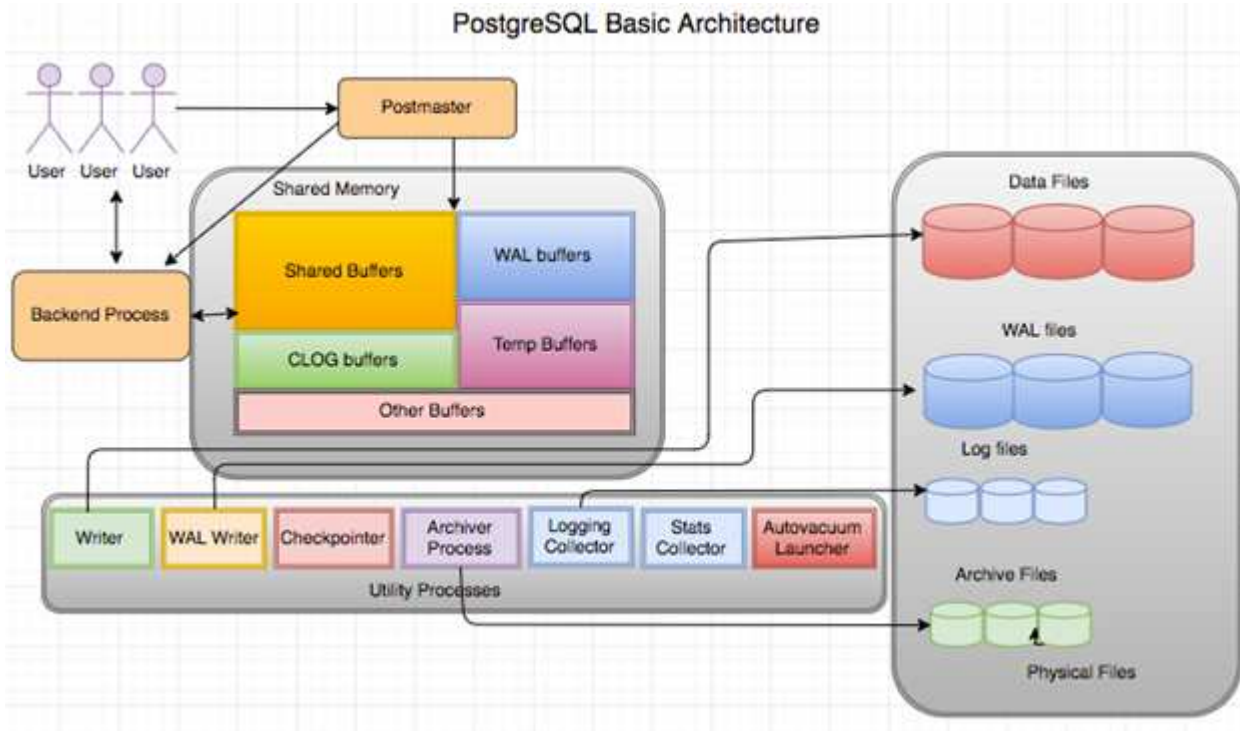
목차

데이터베이스 구성	1
PostgreSQL 아키텍처	1
PostgreSQL 초기화 매개 변수입니다	2
ONTAP를 사용한 PostgreSQL 데이터베이스 구성	2
PostgreSQL 테이블스페이스	4

데이터베이스 구성

PostgreSQL 아키텍처

PostgreSQL은 클라이언트 및 서버 아키텍처를 기반으로 하는 RDBMS입니다. PostgreSQL 인스턴스는 데이터베이스 클러스터라고 하며, 이 클러스터는 서버 컬렉션이 아닌 데이터베이스의 모음입니다.



PostgreSQL 데이터베이스에는 포스트마스터, 프론트 엔드(클라이언트) 및 백 엔드의 세 가지 주요 요소가 있습니다. 클라이언트는 IP 프로토콜 및 연결할 데이터베이스와 같은 정보를 사용하여 Postmaster에 요청을 보냅니다. 포스트마스터는 연결을 인증하고 추가 통신을 위해 백 엔드 프로세스에 전달합니다. 백 엔드 프로세스는 쿼리를 실행하고 결과를 프론트 엔드(클라이언트)로 직접 보냅니다.

PostgreSQL 인스턴스는 다중 스레드 모델 대신 다중 프로세스 모델을 기반으로 합니다. 이 프로세스는 서로 다른 작업에 대해 여러 프로세스를 생성하며 각 프로세스에는 고유한 기능이 있습니다. 주요 프로세스에는 클라이언트 프로세스, Wal 작성기 프로세스, 백그라운드 작성기 프로세스 및 체크포인트 프로세스가 포함됩니다.

- 클라이언트(포그라운드) 프로세스가 PostgreSQL 인스턴스에 읽기 또는 쓰기 요청을 보내면 디스크에 직접 데이터를 읽거나 쓰지 않습니다. 먼저 공유 버퍼와 Wal(Write-Ahead Logging) 버퍼에 데이터를 버퍼링합니다.
- Wal 작성기 프로세스는 Wal 로그에 쓸 공유 버퍼 및 Wal 버퍼의 내용을 조작합니다. WAL 로그는 일반적으로 PostgreSQL의 트랜잭션 로그이며 순차적으로 기록됩니다. 따라서 데이터베이스의 응답 시간을 향상시키기 위해 PostgreSQL은 먼저 트랜잭션 로그에 기록하고 클라이언트를 확인합니다.
- 데이터베이스를 일관된 상태로 만들기 위해 백그라운드 작성기 프로세스는 공유 버퍼에 더티 페이지가 있는지 주기적으로 검사합니다. 그런 다음 데이터를 NetApp 볼륨 또는 LUN에 저장된 데이터 파일로 플러싱합니다.
- 또한 체크포인트 프로세스는 백그라운드 프로세스보다 빈도가 낮아 주기적으로 실행되며 버퍼가 수정되지 않습니다. NetApp 디스크에 저장된 Wal 로그 끝에 체크포인트 레코드를 쓰고 플러시하도록 Wal 작성기 프로세스에 신호를 보냅니다. 또한 백그라운드 작성기 프로세스에 모든 더티 페이지를 기록하고 디스크에 플러시하도록 신호를

보냅니다.

PostgreSQL 초기화 매개 변수입니다

를 사용하여 새 데이터베이스 클러스터를 생성합니다 `initdb` 프로그램. AN `initdb` 스크립트는 클러스터를 정의하는 데이터 파일, 시스템 테이블 및 템플릿 데이터베이스(`template0` 및 `Template1`)를 생성합니다.

템플릿 데이터베이스는 재고 데이터베이스를 나타냅니다. 시스템 테이블, 표준 보기, 함수 및 데이터 형식에 대한 정의가 포함되어 있습니다. `pgdata` 에 대한 인수 역할을 합니다 `initdb` 데이터베이스 클러스터의 위치를 지정하는 스크립트입니다.

PostgreSQL의 모든 데이터베이스 개체는 해당 OID에 의해 내부적으로 관리됩니다. 테이블 및 인덱스도 개별 OID에 의해 관리됩니다. 데이터베이스 개체와 해당 OID 간의 관계는 개체 유형에 따라 적절한 시스템 카탈로그 테이블에 저장됩니다. 예를 들어 데이터베이스 및 힙 테이블의 OID는 에 저장됩니다 `pg_database` 및 `pg_class`. PostgreSQL 클라이언트에서 쿼리를 실행하여 OID를 확인할 수 있습니다.

모든 데이터베이스에는 1GB로 제한된 고유한 개별 테이블 및 인덱스 파일이 있습니다. 각 테이블에는 각각 접미사가 붙은 두 개의 관련 파일이 있습니다 `_fsm` 및 `_vm`. 이를 여유 공간 지도와 가시성 지도라고 합니다. 이러한 파일은 사용 가능한 공간 용량에 대한 정보를 저장하며 테이블 파일의 각 페이지를 볼 수 있습니다. 인덱스에는 개별 가용 공간 맵만 있으며 가시성 맵은 없습니다.

를 클릭합니다 `pg_xlog/pg_wal` 디렉토리는 미리 쓰기 로그가 포함됩니다. 미리 쓰기 로그는 데이터베이스의 안정성과 성능을 개선하는 데 사용됩니다. 테이블의 행을 업데이트할 때마다 PostgreSQL은 먼저 미리 쓰기 로그에 변경 내용을 쓰고 나중에 실제 데이터 페이지에 대한 수정 내용을 디스크에 씁니다. 를 클릭합니다 `pg_xlog` 디렉토리는 일반적으로 여러 파일이 포함되어 있지만 `initdb`는 첫 번째 파일만 만듭니다. 필요에 따라 추가 파일이 추가됩니다. 각 `xlog` 파일의 길이는 16MB입니다.

ONTAP를 사용한 PostgreSQL 데이터베이스 구성

성능을 향상시킬 수 있는 PostgreSQL 튜닝 구성에는 여러 가지가 있습니다.

가장 일반적으로 사용되는 매개 변수는 다음과 같습니다.

- `max_connections` = <num>: 한 번에 가질 최대 데이터베이스 연결 수입니다. 이 매개 변수를 사용하여 디스크로의 스와핑을 제한하고 성능을 중단합니다. 응용 프로그램 요구 사항에 따라 연결 풀 설정에 대해 이 매개 변수를 조정할 수도 있습니다.
- `shared_buffers` = <num>: 데이터베이스 서버의 성능을 개선하는 가장 간단한 방법입니다. 최신 하드웨어의 기본값은 `LOW` 입니다. 배포 중에 시스템에서 사용 가능한 RAM의 약 25%로 설정됩니다. 이 매개 변수 설정은 특정 데이터베이스 인스턴스에서 작동하는 방식에 따라 달라집니다. 시행 착오로 값을 늘리거나 줄여야 할 수도 있습니다. 그러나 높게 설정하면 성능이 저하될 수 있습니다.
- `effective_cache_size` = <num>: 이 값은 PostgreSQL의 최적화 프로그램에 PostgreSQL이 데이터를 캐싱하는 데 사용할 수 있는 메모리 양을 알려주고 인덱스를 사용할지 여부를 결정하는 데 도움이 됩니다. 값이 클수록 인덱스를 사용할 가능성이 높아집니다. 이 매개 변수는 에 할당된 메모리 크기로 설정해야 합니다 `shared_buffers` 사용 가능한 OS 캐시 양에 더합니다. 이 값은 전체 시스템 메모리의 50%를 초과하는 경우가 많습니다.
- `work_mem` = <num>: 이 매개 변수는 정렬 작업 및 해시 테이블에 사용할 메모리 양을 제어합니다. 응용 프로그램에서 무거운 정렬을 수행하는 경우 메모리 양을 늘려야 할 수 있지만 주의해야 합니다. 이는 시스템 차원 매개 변수가 아니라 작업당 매개 변수입니다. 복합 쿼리에 여러 개의 정렬 작업이 있는 경우 여러 개의 `work_mem`

메모리 유닛을 사용하며 여러 백 엔드에서 동시에 이 작업을 수행할 수 있습니다. 값이 너무 큰 경우 이 쿼리로 인해 데이터베이스 서버가 스왑될 수 있습니다. 이전 버전의 PostgreSQL에서는 이 옵션을 `sort_mem`이라고 했습니다.

- `fsync = <boolean> (on or off)`: 이 매개변수는 트랜잭션이 커밋되기 전에 `fsync()`를 사용하여 모든 Wal 페이지를 디스크에 동기화할지 여부를 결정합니다. 이 기능을 끄면 쓰기 성능이 향상될 수 있으며 이 기능을 켜면 시스템이 충돌할 때 손상 위험으로부터 보호됩니다.
- `checkpoint_timeout`: 체크포인트 프로세스는 커밋된 데이터를 디스크로 플러시합니다. 여기에는 디스크에서 많은 양의 읽기/쓰기 작업이 사용됩니다. 이 값은 초 단위로 설정되고 값이 낮을수록 충돌 복구 시간이 줄어들고 값을 늘리면 체크포인트 호출을 줄여 시스템 리소스의 부하가 감소할 수 있습니다. 애플리케이션 중요도, 사용량, 데이터베이스 가용성에 따라 `checkpoint_timeout` 값을 설정합니다.
- `commit_delay = <num>` 및 `commit_siblings = <num>`: 이러한 옵션은 한 번에 커밋되는 여러 트랜잭션을 작성함으로써 성능을 향상시키는 데 사용됩니다. 트랜잭션이 커밋되는 즉시 여러 개의 `commit_siblings` 개체가 활성 상태인 경우 서버는 `commit_delay microseconds`가 한 번에 여러 트랜잭션을 커밋할 때까지 기다립니다.
- `max_worker_processes / max_parallel_workers`: 프로세스에 대한 최적의 작업자 수를 구성합니다. `max_parallel_workers`는 사용 가능한 CPU 수에 해당합니다. 응용 프로그램 설계에 따라 병렬 작업의 경우 쿼리에 필요한 작업자의 수가 줄어들 수 있습니다. 두 매개 변수의 값을 동일하게 유지하되 테스트 후 값을 조정하는 것이 좋습니다.
- `random_page_cost = <num>`: 이 값은 PostgreSQL이 비순차적 디스크 읽기를 보는 방식을 제어합니다. 값이 높을수록 PostgreSQL은 인덱스 검사 대신 순차 검사를 사용할 가능성이 높으므로 서버에 빠른 디스크가 있음을 나타냅니다. 계획 기반 최적화, 진공 청소, 쿼리 또는 스키마를 변경하는 인덱싱 등의 다른 옵션을 평가한 후 이 설정을 수정합니다.
- `effective_io_concurrency = <num>`: 이 매개변수는 PostgreSQL이 동시에 실행하려고 시도하는 동시 디스크 I/O 작업의 수를 설정합니다. 이 값을 높이면 개별 PostgreSQL 세션이 병렬로 시작하려고 하는 입출력 작업 수가 증가합니다. 허용되는 범위는 1에서 1,000까지이며, 비동기 I/O 요청 발급을 비활성화하려면 0입니다. 현재 이 설정은 비트맵 힙 스캔에만 영향을 줍니다. SSD(Solid-State Drive)와 기타 메모리 기반 스토리지(NVMe)는 동시 요청을 많이 처리할 수 있으므로 수백 개의 요청이 최고의 가치를 실현할 수 있습니다.

PostgreSQL 구성 매개 변수의 전체 목록은 PostgreSQL 설명서를 참조하십시오.

토스트

TOAST는 특대형 특성 저장 기술을 의미합니다. PostgreSQL은 고정된 페이지 크기(일반적으로 8KB)를 사용하며 튜플이 여러 페이지에 걸쳐 있을 수 없습니다. 따라서 큰 필드 값을 직접 저장할 수 없습니다. 이 크기를 초과하는 행을 저장하려고 할 때 TOAST는 큰 열의 데이터를 작은 "조각"으로 나눈 다음 TOAST 테이블에 저장합니다.

토스트 속성의 큰 값은 결과 집합이 클라이언트로 전송될 때만(선택한 경우) 당겨집니다. TOAST(Out-of-Line Storage)가 없을 때보다 테이블 자체가 훨씬 작고 공유 버퍼 캐시에 더 많은 행을 저장할 수 있습니다.

진공

일반적인 PostgreSQL 작업에서는 업데이트로 삭제되거나 폐기된 튜플은 테이블에서 물리적으로 제거되지 않으며 진공이 실행될 때까지 그대로 유지됩니다. 따라서, 특히 자주 업데이트되는 테이블에서 정기적으로 진공을 실행해야 합니다. 그런 다음 디스크 공간 중단을 방지하기 위해 새 행에서 재사용할 수 있도록 해당 공간을 재확보해야 합니다. 그러나 운영 체제에 공간을 반환하지 않습니다.

페이지 내의 여유 공간은 조각화되지 않습니다. Vacuum은 전체 블록을 다시 쓰므로 나머지 행을 효율적으로 압축하고 페이지에 연속된 단일 여유 공간 블록을 남깁니다.

반면, VACUUM FULL은 데드 공간 없이 완전히 새로운 버전의 테이블 파일을 작성하여 테이블을 적극적으로 압축합니다. 이렇게 하면 테이블 크기가 최소화되지만 시간이 오래 걸릴 수 있습니다. 또한 작업이 완료될 때까지 테이블의 새

복사본을 위한 추가 디스크 공간이 필요합니다. 일상적인 진공의 목적은 진공이 완전히 작동하지 않도록 하는 것입니다. 이 프로세스는 테이블을 최소 크기로 유지할 뿐만 아니라 디스크 공간의 안정적 상태 사용도 유지합니다.

PostgreSQL 테이블스페이스

데이터베이스 클러스터가 초기화될 때 두 개의 테이블스페이스가 자동으로 생성됩니다.

를 클릭합니다 `pg_global` 테이블스페이스는 공유 시스템 카탈로그에 사용됩니다. 를 클릭합니다 `pg_default` 테이블스페이스는 `Template1` 및 `template0` 데이터베이스의 기본 테이블스페이스입니다. 클러스터가 초기화된 파티션이나 볼륨에 공간이 부족하여 확장할 수 없는 경우 다른 파티션에 테이블스페이스를 생성하여 시스템을 재구성할 때까지 사용할 수 있습니다.

자주 사용되는 인덱스는 솔리드 스테이트 디바이스처럼 빠른 고가용성 디스크에 배치할 수 있습니다. 또한 드물게 사용되거나 성능이 중요하지 않은 아카이브 데이터를 저장하는 테이블을 SAS 또는 SATA 드라이브와 같이 더 저렴하고 느린 디스크 시스템에 저장할 수 있습니다.

테이블스페이스는 데이터베이스 클러스터의 일부이며 데이터 파일의 자동 모음으로 취급할 수 없습니다. 기본 데이터 디렉토리에 포함된 메타데이터에 따라 달라지므로 다른 데이터베이스 클러스터에 연결하거나 개별적으로 백업할 수 없습니다. 마찬가지로 파일 삭제, 디스크 오류 등을 통해 테이블스페이스를 잃으면 데이터베이스 클러스터를 읽을 수 없거나 시작할 수 없게 됩니다. RAM 디스크와 같은 임시 파일 시스템에 테이블스페이스를 배치하면 전체 클러스터의 안정성이 저하됩니다.

생성된 후 요청 사용자에게 충분한 권한이 있는 경우 모든 데이터베이스에서 테이블스페이스를 사용할 수 있습니다. PostgreSQL은 심볼 링크를 사용하여 테이블스페이스의 구현을 간소화합니다. PostgreSQL은 에 행을 추가합니다 `pg_tablespace` 테이블(클러스터 전체 테이블) 및 새 OID(개체 식별자)를 해당 행에 할당합니다. 마지막으로, 서버는 OID를 사용하여 클러스터와 지정된 디렉토리 사이에 심볼 링크를 생성합니다. 디렉터리 `$PGDATA/pg_tblspc` 클러스터에 정의되어 있지 않은 각 테이블스페이스를 가리키는 심볼 링크를 포함합니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.