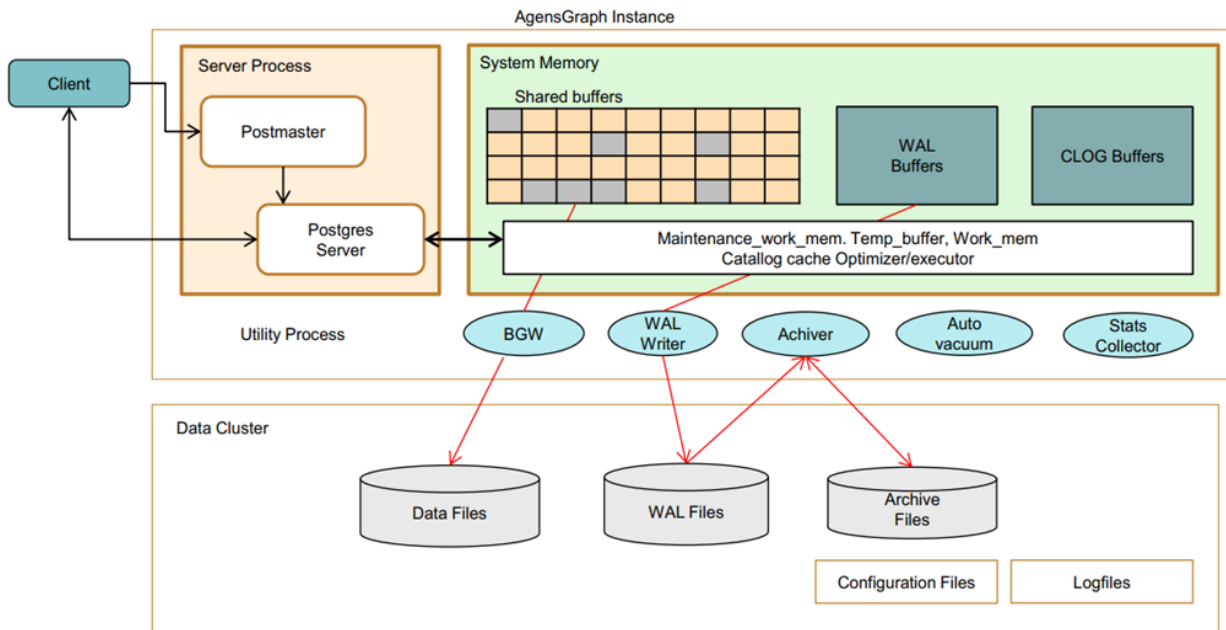


3장. AgensSQL 아키텍처

AgensSQL은 PostgreSQL에서 fork되었기 때문에 PostgreSQL의 아키텍처를 그대로 따릅니다.

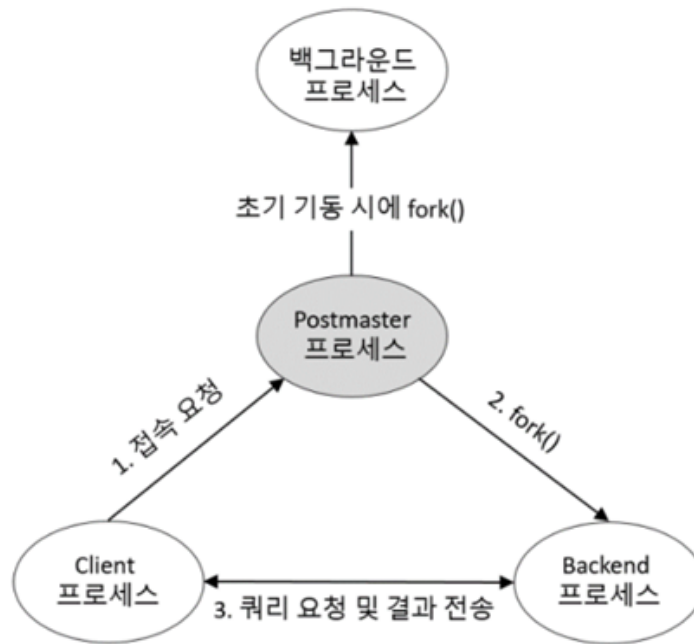


3.1 Process 아키텍처

PostgreSQL은 Client / Server 모델을 사용하며, 프로세스 기반의 DBMS입니다. PostgreSQL에서 사용하는 프로세스들은 아래와 같습니다.

3.1.1. Postmaster Process

client의 권한을 확인한 후 새로운 postgres progress를 시작하여 client를 연결합니다. 1개의 Connection(=Client Process)마다 1개의 Backend Process를 생성(Postmaster Process에 의해 fork)합니다.



3.1.2. Multiple Postgres Process

프로세스명	설명
postmaster process	client의 권한을 확인한 후 새로운 postgres process를 시작하여 client를 연결합니다.
WAL writer process	client가 commit 을 요청하면 WAL writer는 해당 트랜잭션이 발생하는 모든 WAL 레코드를 WAL 파일로 쓰고 flush 한다
background writer process	shared buffer를 주기적으로 점검하여 dirty page를 데이터 파일에 기록한다
checkpointer process	checkpoint 발생시 더이상 버퍼 수정이 발생하지 않도록 합니다. background writer process가 dirty page를 flush 하도록 하고, WAL Writer process 가 checkpoint 레코드를 WAL파일에 기록하고 flush 하도록하여 데이터베이스의 일관성을 유지합니다.

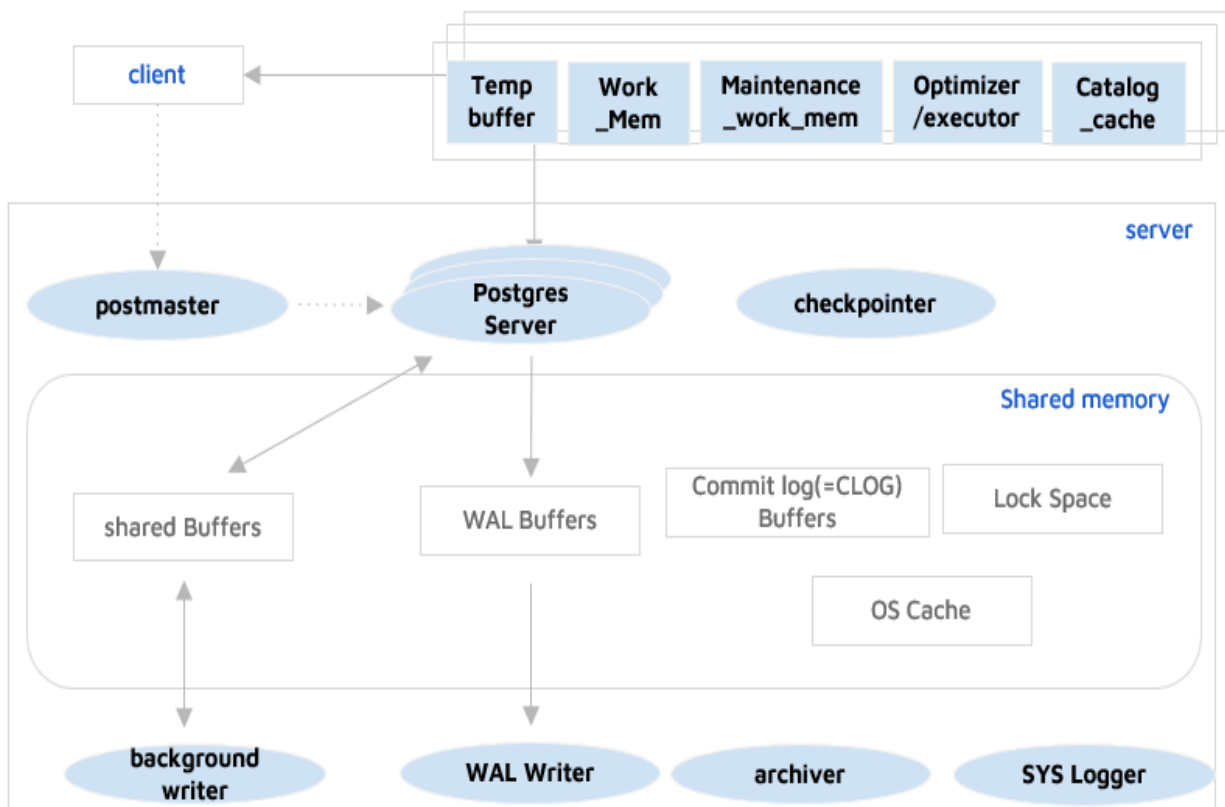
3.1.3. Optional Process

프로세스명	설명
autovacuum launcher process	하위 프로세스를 실행하여 데이터베이스에 대한 vacuum 작업을 한다
logger process	로그 파일에 log, warning, error message를 기록합니다. (WAL file과는 다른 log file 임)

archiver process	WAL process에 의해 완전히 채워진 WAL 파일을 구성 가능한 위치에 복사합니다.
stats collector process	table과 index에 대한 액세스 수, table의 총 row 수에 대한 정보를 지속적으로 수집하며 VACUUM/ANALYZE 및 ANALYZE와 함께 동작합니다.
WAL sender/receiver processes	Streaming Replication 기능의 일부를 수행합니다.

3.2 Memory 아키텍처(=Shared Memory)

모든 세션들이 공유하여 사용하는 Shared Memory와 세션별로 할당되는 Backend Memory로 나뉩니다.



3.2.1. Shared Memory

Shared Memory는 모든 세션들이 공유하여 사용하는 공간으로 전역적으로 발생하는 데이터베이스의 작동과 관리에 필요한 공간이 할당됩니다.

항목	설명
Shared Buffer	데이터 영역에서 참조된 데이터를 저장하는 공간
WAL Buffer	데이터 변경시, 변경된 내역을 저장하는 공간. Crash 발생시 복구에 사용
Clog Buffer	트랜잭션 상태 정보를 저장하는 공간
Lock Space	트랜잭션 간의 Lock 정보를 저장하는 공간
Other Buffers	통계정보, two-phases-commit 등의 버퍼 공간

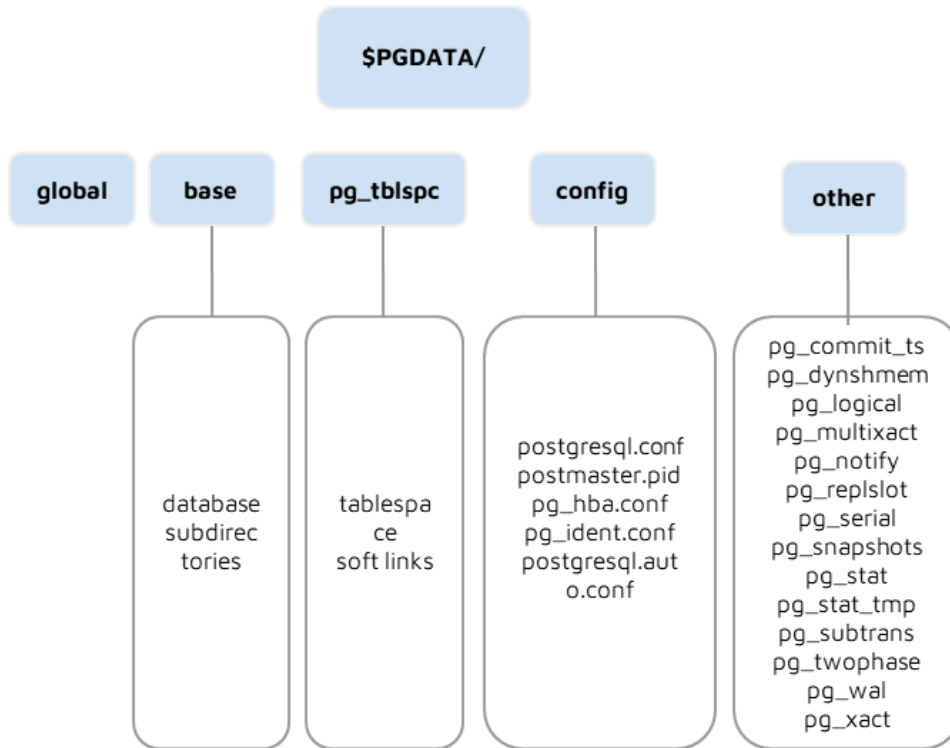
3.2.2. Backend Memory

postgres process는 각 client에 요청을 처리하는 server process에 할당되며, 각 server process에 의해 각각의 세션들에 Backend Memory가 자체적으로 할당됩니다.

항목	설명
temp_buffer	임시 테이블들을 저장하는 공간
work_mem	정렬, 조인, Hash Table 작업 등을 위해서 사용하는 공간
maintenance_work_mem	Vacuum, Create Index 등의 작업을 위해 사용하는 공간
optimizer / executor	수행할 쿼리들에 대한 최적의 실행 계획 수립 및 실행을 담당하는 영역
catalog cache	메타데이터를 조회하거나 이용하기 위한 공간

3.3 Database Cluster 구조

Database Cluster를 initdb를 통해 초기화를 진행하면, 기본적으로 아래와 같이 파일과 디렉토리가 생성됩니다.



항목	설명
PG_VERSION	PostgreSQL의 주 버전 번호가 포함된 파일
base	데이터베이스 별 하위 디렉토리를 포함하는 하위 디렉토리
global	pg_database와 같은 클러스터 전체 테이블을 포함하는 하위 디렉토리
pg_commit_ts	트랜잭션 커밋 타임 스탬프 데이터가 포함된 하위 디렉토리
pg_clog	트랜잭션 커밋 상태 데이터를 포함하는 하위 디렉토리
pg_dynshmem	동적 공유 메모리 하위 시스템에서 사용하는 파일을 포함하는 하위 디렉토리
pg_logical	논리적 디코딩을 위한 상태 데이터가 포함된 하위 디렉토리
pg_multixact	다중 트랜잭션 상태 데이터를 포함하는 하위 디렉토리 (공유 행 잠금에 사용됨)
pg_notify	LISTEN / NOTIFY 상태 데이터를 포함하는 하위 디렉토리

pg_replslot	복제 슬롯 데이터가 포함 된 하위 디렉토리
pg_serial	커밋 된 직렬화 가능한 트랜잭션에 대한 정보가 포함 된 하위 디렉터리
pg_snapshots	내 보낸 스냅 샷이 포함 된 하위 디렉토리
pg_stat	통계 하위 시스템에 대한 영구 파일이 포함 된 하위 디렉토리
pg_stat_tmp	통계 하위 시스템에 대한 임시 파일이 포함 된 하위 디렉토리
pg_subtrans	서브 트랜잭션 상태 데이터를 포함하는 서브 디렉토리
pg_tblspc	테이블 스페이스에 대한 심볼릭 링크를 포함하는 하위 디렉토리
pg_twophase	준비된 트랜잭션에 대한 상태 파일이 포함 된 하위 디렉터리
pg_xlog	WAL (Write Ahead Log) 파일이 포함 된 하위 디렉토리
postgresql.conf	기본 설정 파일
postgresql.auto.conf	ALTER SYSTEM에서 설정 한 구성 매개 변수를 저장하는 데 사용되는 파일
pg_hba.conf	PostgreSQL에 접속하는 클라이언트에 대한 인증 설정을 설명하는 파일. HBA는 호스트 기반 인증(host-based authentication)의 약어
pg_ident.conf	클라이언트의 인증 방식으로 Ident 인증을 사용하는 경우, ident 사용자 이름을 PostgreSQL의 역할 이름에 매핑하는데 사용하는 파일
current_logfiles	현재 기록되는 로그 파일명 명시
postmaster.opts	서버가 마지막으로 시작된 명령 줄 옵션을 기록하는 파일
postmaster.pid	현재 포스트 마스터 프로세스 ID (PID), 클러스터 데이터 디렉토리 경로, 포스트 마스터 시작 타임 스탬프, 포트 번호, Unix 도메인 소켓 디렉토리 경로 (Windows에서는 비어 있음), 첫 번째 유효한 listen_address (IP 주소 또는 * 또는 그렇지 않은 경우 비어 있음)를 기록하는 잠금 파일 TCP에서 수신) 및 공유 메모리 세그먼트 ID (이 파일은 서버 종료 후에 존재하지 않음)