睿创神脑
RUICHUANGSHENNAO

PolarDB

PostgreSQL

PostgreSQL 实例结构

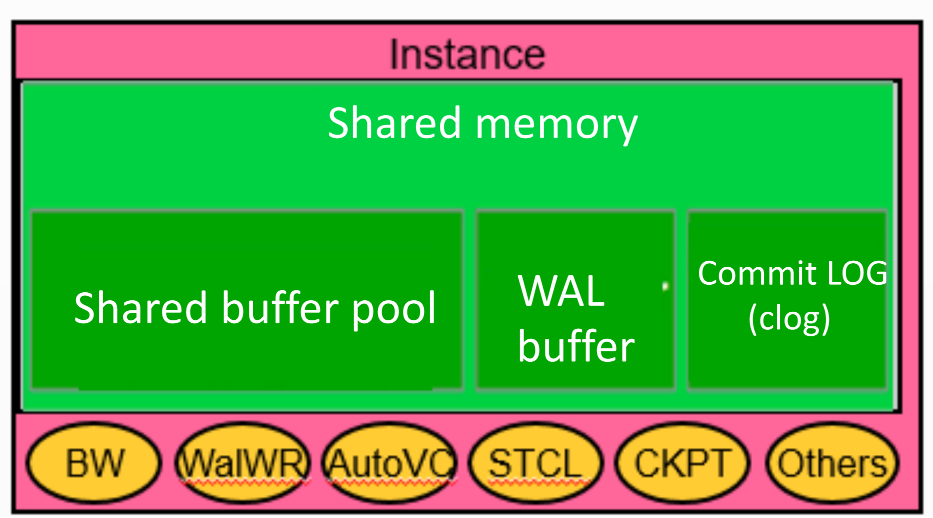
Objectives

- PostgreSQL实例简介
- PostgreSQL初始化参数文件
- PostgreSQL内存结构
- PostgreSQL后台进程

PostgreSQL实例

PostgreSQL实例

- 用来访问PostgreSQL数据库
- 一个实例对应一个数据库集簇
- 由内存和后台进程组成



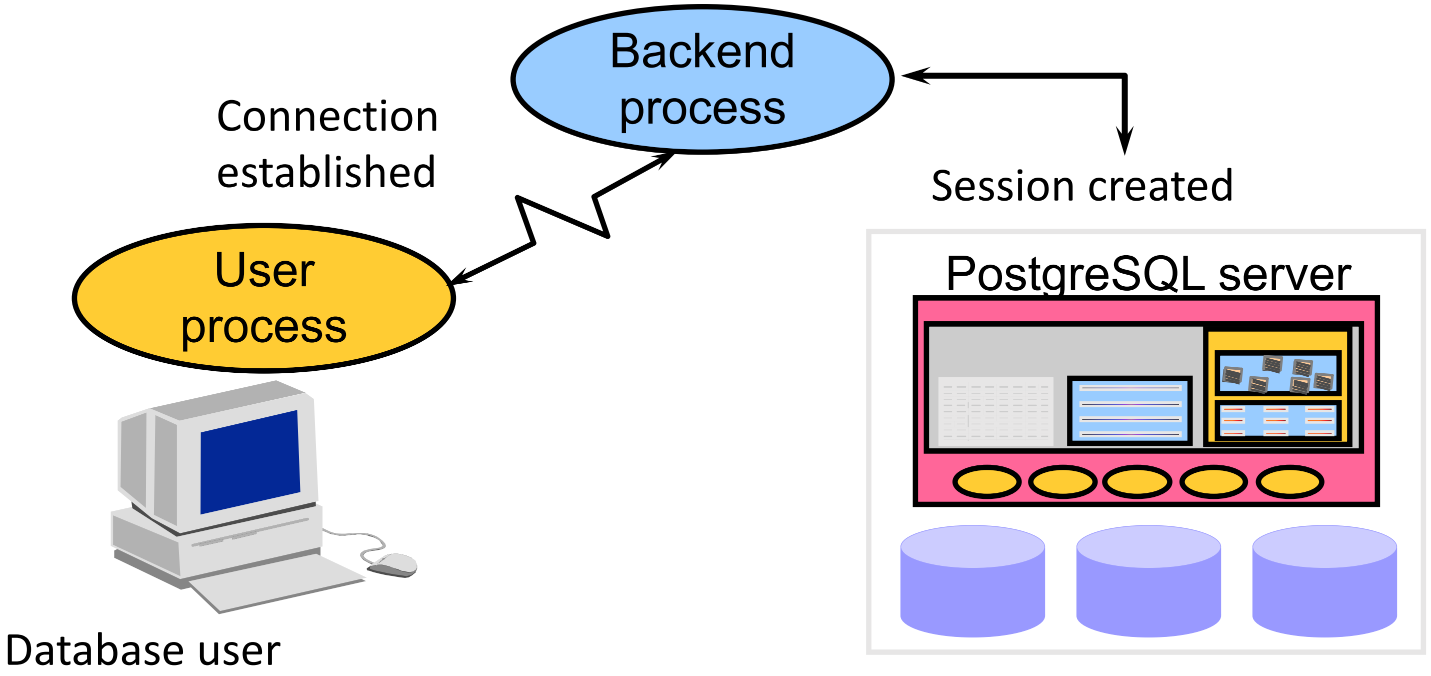
Memory structures

Background process structures

建立一个连接和创建一个会话

连接到一个PostgreSQL实例

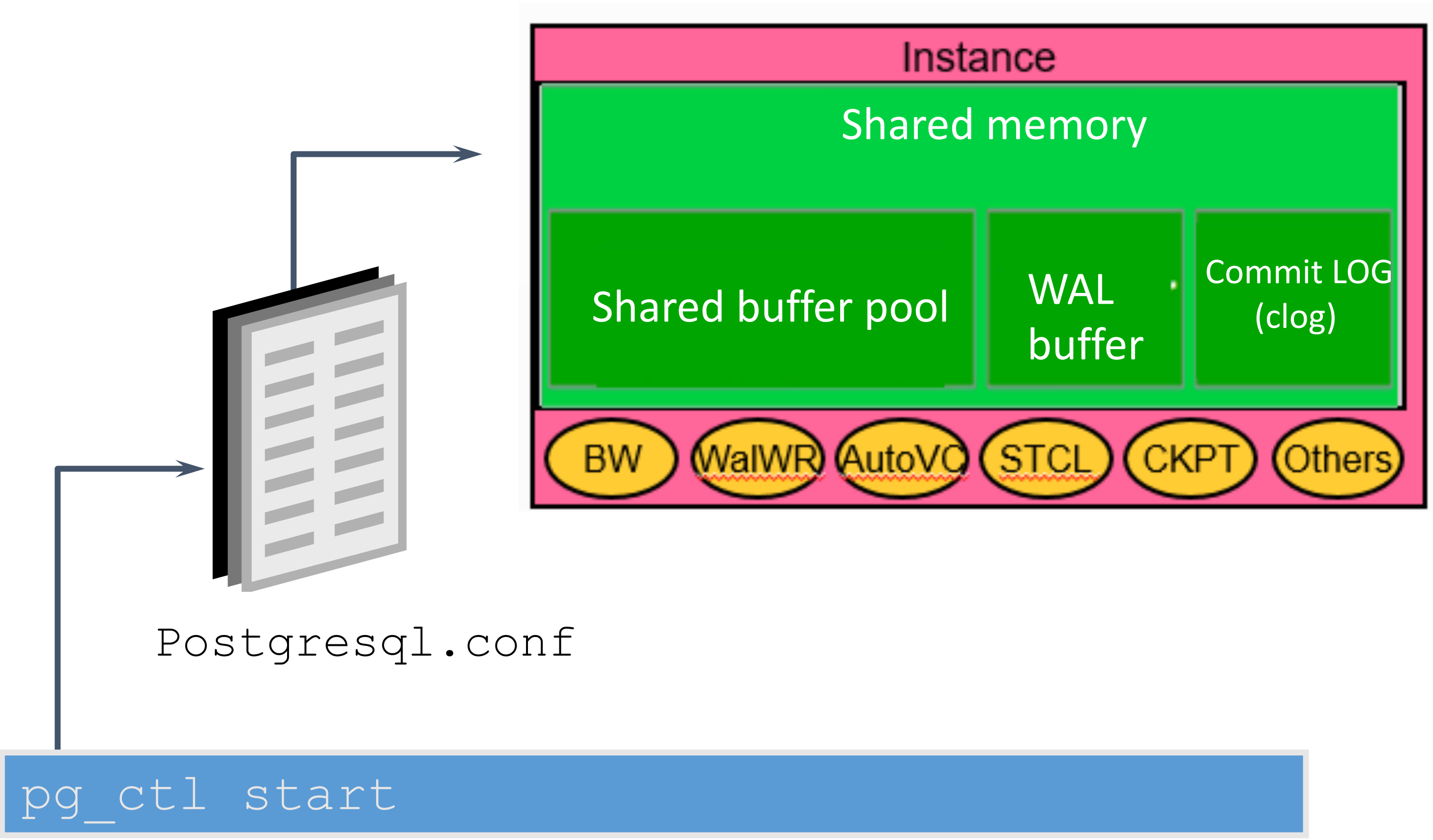
- 建立一个用户连接
- 创建一个会话



睿创神脑
RUICHUANGSHENNAO

初始化参数文件

PolarDB PostgreSQL



CUUG

初始化参数文件

两种类型的参数

- 显式：在文件中对参数进行设置
- 隐式：文件中没有设置，使用Postgres默认值

可以存在多个初始化参数文件

- 静态参数文件：postgresql.conf
- 动态参数文件：postgresql.auto.conf
- ~~可选参数文件：postgresql.conf.user~~
- 可选DB/USER定制参数: pg_db_role_setting

读取顺序

- postgresql.conf →postgresql.auto.conf→pg_db_role_setting ~~postgresql.conf.user~~

睿创神脑
RUICHUANGSHENNAO

postgresql.conf参数文件

PolarDB PostgreSQL

- 文本文件
- 使用操作系统编辑器修改
- 手动修改
- 更改将在下次启动时生效
- 仅在实例启动期间读取
- 默认位置为$PGDATA

postgresql.auto.conf参数文件

- 文本文件
- 由Postgres服务器维护
- 支持用文本编辑器修改（不推荐）
- ALTER SYSTEM命令修改的参数保存在该文件
- 能够在关闭和启动期间持续进行更改
- 可以实现自我调整参数值
- 默认位置为$PGDATA

postgresql.auto.conf参数文件

- 改变一个参数的值，会在文件中自动添加参数

```
ALTER SYSTEM SET archive_mode = on;
```

- 恢复一个参数默认值，会在文件中自动删除参数

```
ALTER SYSTEM RESET archive_mode;
```

参数生效条件

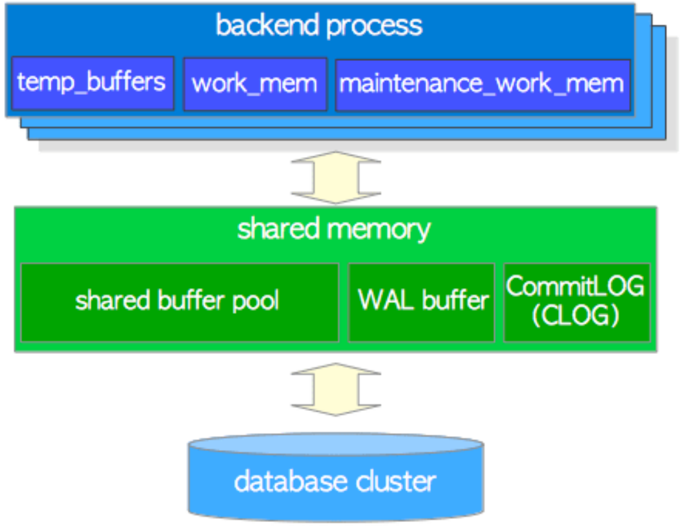
- sighup：表示需要超级管理员修改，reload就能够生效。
- superuser：表示使用超级管理员可以为普通用户、数据库、或者超级管理员自己修改。（注：有些参数是可以针对用户、数据库、实例）
- postmaster：表示需要超级管理员修改，需要重启才能够生效。
- user：表示普通用户可以修改该参数值，立即生效。

```
postgres=#  select name,setting,context from pg_settings
            where name in ('port','work_mem','log_statement','log_checkpoints');
      name        | setting |   context
------------------+---------+------------
 log_checkpoints  | off     | sighup
 log_statement    | all     | superuser
 port             | 1922    | postmaster
 work_mem         | 4096    | user
```

Memory Architecture

PostgreSQL中的内存架构可以分为两大类：

- Local memory area –由每个后端进程分配给自己使用.
- Shared memory area –由PostgreSQL服务器的所有进程使用.



Memory Architecture

Local Memory Area：

sub-area	description
work_mem	Executor uses this area for sorting tuples by ORDER BY and DISTINCT operations, and for joining tables by merge-join and hash-join operations.
maintenance_work_mem	Some kinds of maintenance operations (e.g., VACUUM, REINDEX) use this area.
temp_buffers	Executor uses this area for storing temporary tables.

Memory Architecture

Shared Memory Area：

sub-area	description
shared buffer pool	PostgreSQL loads pages within tables and indexes from a persistent storage to here, and operates them directly.
WAL buffer	To ensure that no data has been lost by server failures, PostgreSQL supports the WAL mechanism. WAL data (also referred to as XLOG records) are transaction log in PostgreSQL; and WAL buffer is a buffering area of the WAL data before writing to a persistent storage.
commit log	Commit Log(CLOG) keeps the states of all transactions (e.g., in_progress,committed,aborted) for Concurrency Control (CC) mechanism.

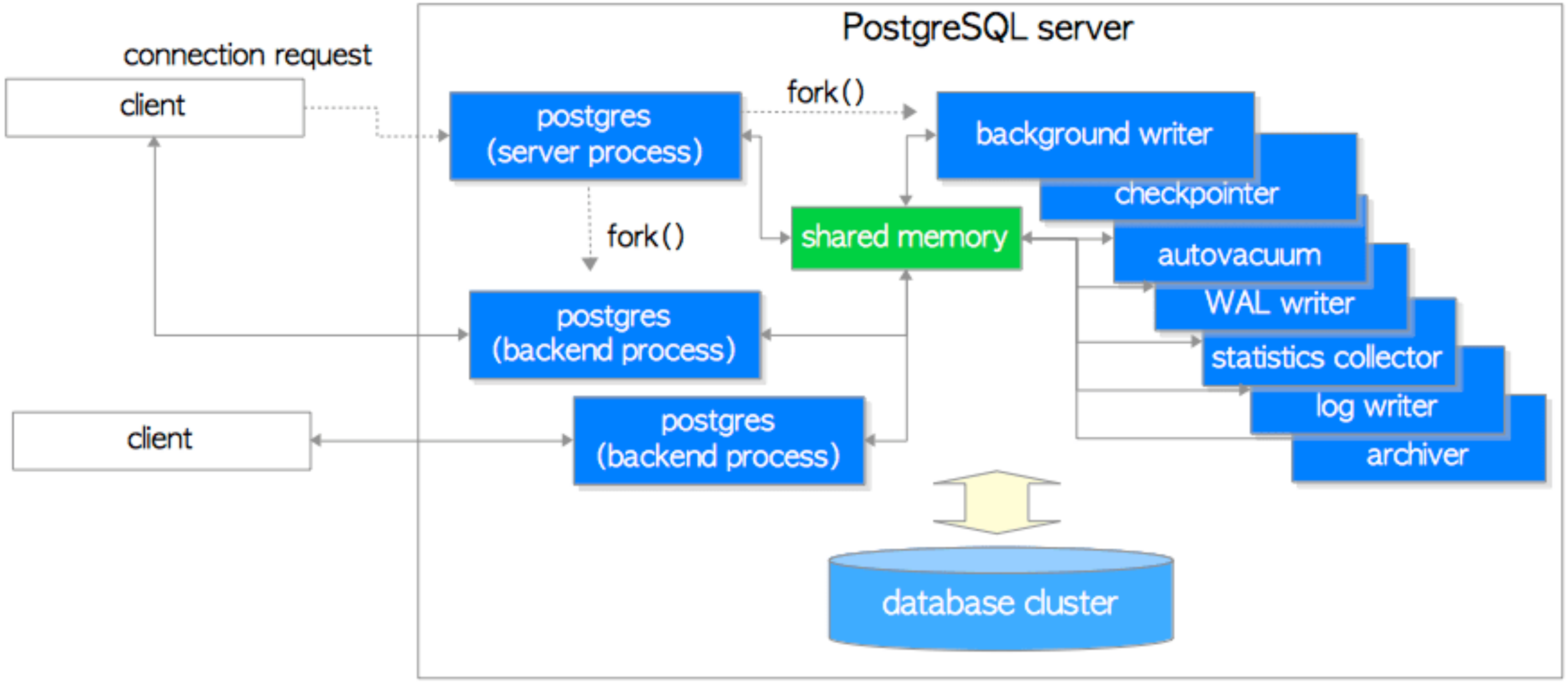
Process Architecture

PostgreSQL进程结构：

- **postgres server process**： postgres服务器进程是与数据库群集管理相关的所有进程的父进程。
- **backend process**：每个后端进程处理连接的客户端发出的所有查询和语句。
- **background processes**：各种后台进程执行用于数据库管理的每个特性的进程（例如清空和检查点进程）。
- **replication associated processes**：在与复制相关联的进程中，它们执行流式复制。
- **background worker process-**：在版本9.3支持的后台工作进程中，它可以执行用户实现的任何处理。

Process Architecture

PostgreSQL进程结构：



Postgres Server Process

Postgres Server Process：

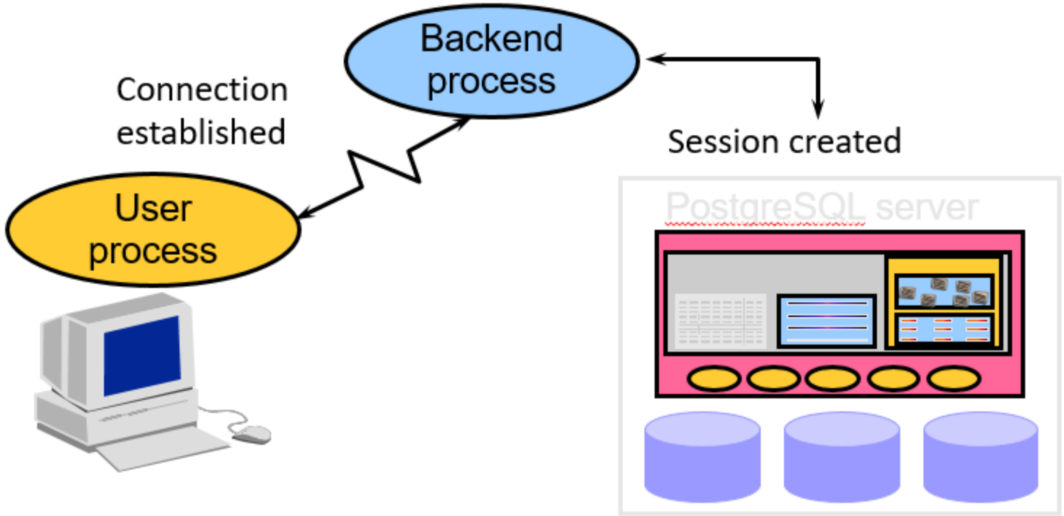
- postgres服务器进程是PostgreSQL服务器中所有进程的父进程。早期版本它被称为“postmaster”。
- pg_ctl实用程序启动该进程，然后派生出各个后台进程。同时分配共享内存区域。
- 一个postgres服务器进程侦听一个网络端口，默认端口为5432。

Backend Processes

Backend Processes：

- 后端进程（也称为postgres）由postgres服务器进程启动，并处理一个连接的客户端发出的所有查询。

- PostgreSQL允许多个客户端同时连接；配置参数max_connections控制客户端的最大数量。

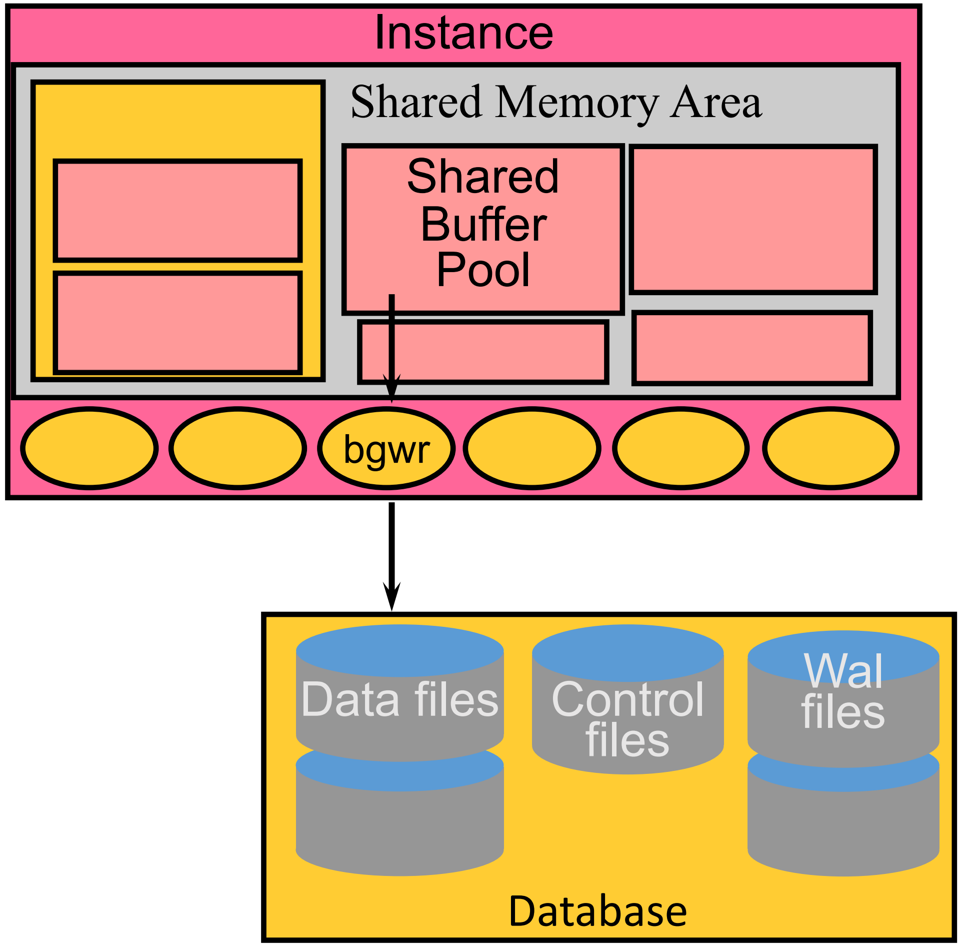


Background Processes

Background Processes：

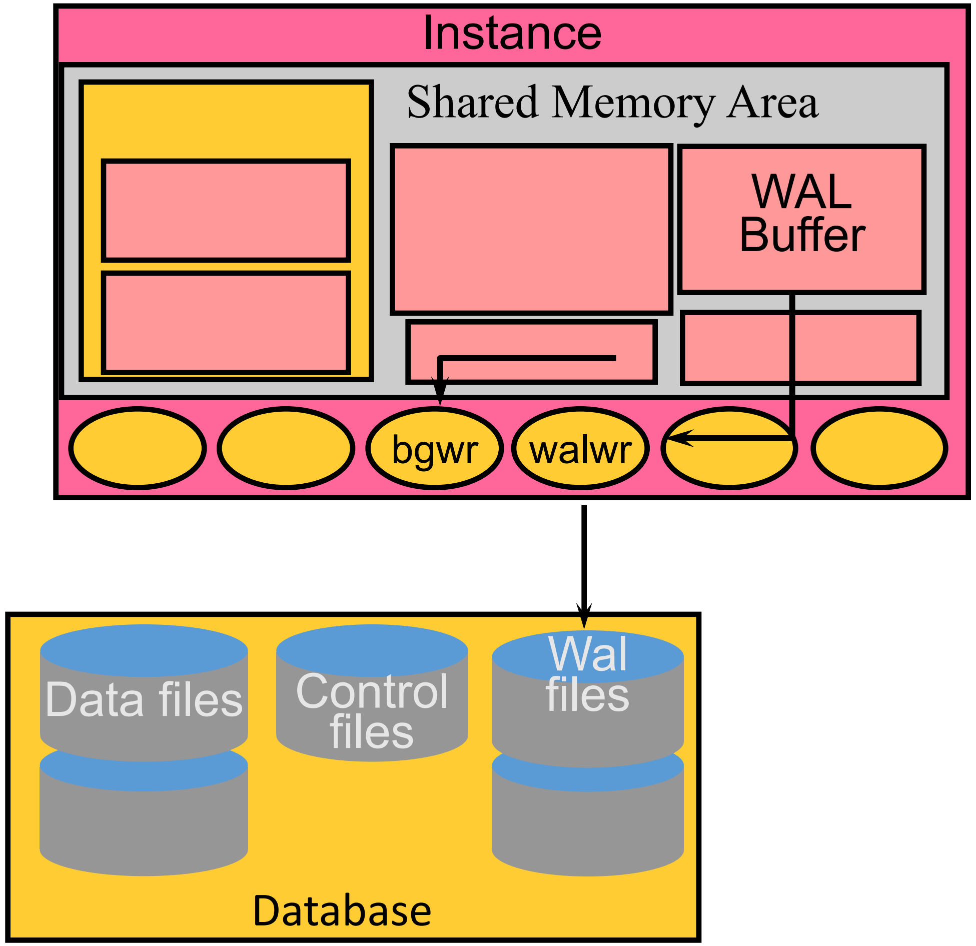
process	description
background writer	In this process, dirty pages on the shared buffer pool are written to a persistent storage (e.g., HDD, SSD) on a regular basis gradually. (In version 9.1 or earlier, it was also responsible for checkpoint process.)
checkpointer	In this process in version 9.2 or later, checkpoint process is performed.
autovacuum launcher	The autovacuum-worker processes are invoked for vacuum process periodically. (More precisely, it requests to create the autovacuum workers to the postgres server.)
WAL writer	This process writes and flushes periodically the WAL data on the WAL buffer to persistent storage.
statistics collector	In this process, statistics information such as for pg_stat_activity and for pg_stat_database, etc. is collected.
logging collector (logger)	This process writes error messages into log files.
archiver	In this process, archiving logging is executed.

background writer



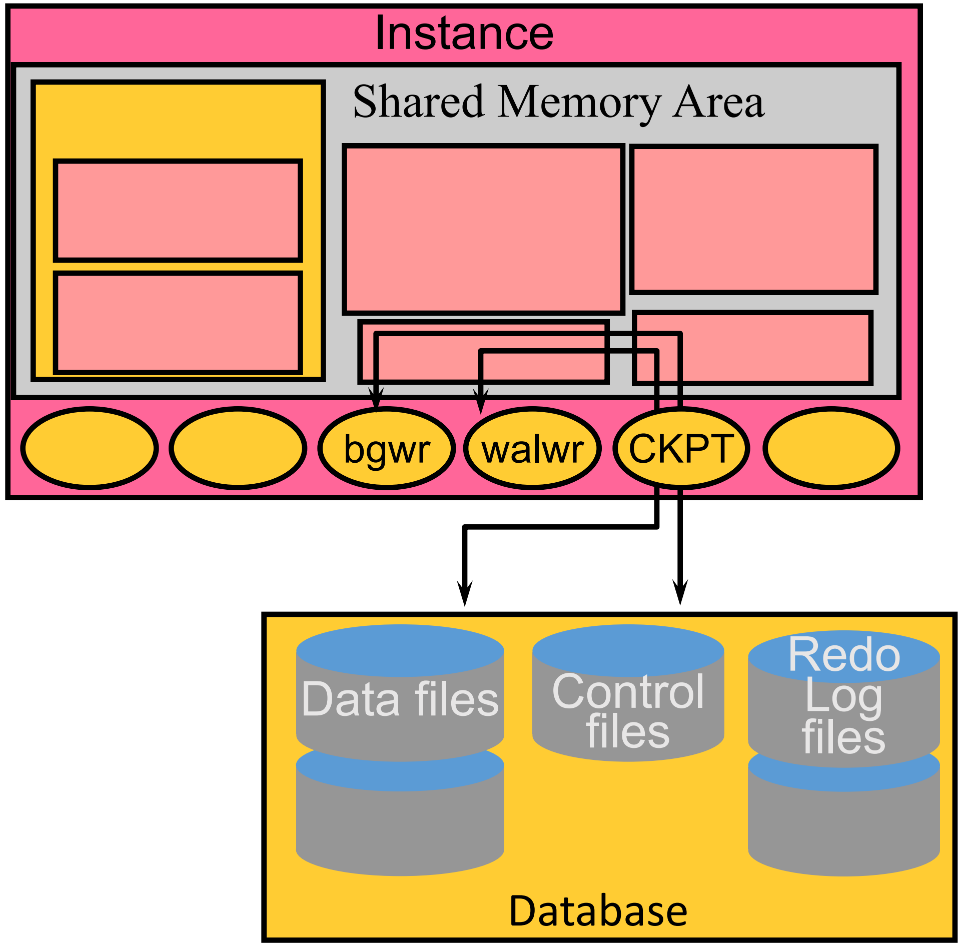
- bgwr writes when:
  - Dirty buffers reach threshold
  - There are no free buffers
  - Timeout occurs

WAL writer



- WAL writer
  - At commit
  - When one-third full
  - There are no free buffers
  - Timeout
  - Before bgwr writes

Checkpoint (CKPT)



- Responsible for:
  - Write dirty buffer
  - Updating wal segment file with checkpoint information
  - Updating control files with checkpoint information
  - Pg_start_backup()

Background Processes

Background Processes：

```
postgres> pstree -p 9687
-+= 00001 root /sbin/launchd
 \-+- 09687 postgres /usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data
   |--= 09688 postgres postgres: logger process
   |--= 09690 postgres postgres: checkpointer process
   |--= 09691 postgres postgres: writer process
   |--= 09692 postgres postgres: wal writer process
   |--= 09693 postgres postgres: autovacuum launcher process
   |--= 09694 postgres postgres: archiver process
   |--= 09695 postgres postgres: stats collector process
   |--= 09697 postgres postgres: postgres sampledb 192.168.1.100(54924) idle
   \--= 09717 postgres postgres: postgres sampledb 192.168.1.100(54964) idle in transaction
```

睿创神脑
RUICHUANGSHENNAO

PolarDB PostgreSQL

总结

- PostgreSQL实例简介
- PostgreSQL初始化参数文件
- PostgreSQL内存结构
- PostgreSQL后台进程

练习

- 1、在postgresql.conf, postgresql.auto.conf, alter user, alter database 中各设置work_mem的不同值. 观察参数优先级.
- 2、设置max_connection, 观察这个参数允许的最大值与什么参数有关.
- 3、设置synchronous_commit的不同值, 观察不同值对tpcb压测结果带来的影响.

睿创神脑
RUICHUANGSHENNAO

PolarDB PostgreSQL