Arquitectura del sistema PostgreSQL

info@todopostgresql.com

Todo PostgreSQL
by Abatic

abatic

1. Introducción
2. Procesos
3. Conexión
4. Funcionamiento

Todo PostgreSQL by Abatic

Introducción

PostgreSQL tiene una arquitectura multi-proceso.

- Está escrito en C.
- Escala muy bien en multi-procesador.
- Reduce la complejidad y problemática.
- Simplifica la seguridad.

Modelo “proceso por usuario”:

Cada conexión es atendida por un nuevo proceso

Todo PostgreSQL by Abatic

Introducción

- Soporta múltiples arquitecturas de CPU:

x86	x86_64	IA64	S/390	PowerPC 64
S/390x	Sparc	Sparc64	ARM	MIPS
MIPSEL	M68K	PA-RISC		

Todo PostgreSQL by Abatic

Introducción

- Soporta múltiples Sistemas Operativos:

Linux	Windows	FreeBSD	OpenBSD	NetBSD
OS X	AIX	HP/UX	Solaris	UnixWare

Existen algunas notas específicas según la plataforma donde se instala PostgreSQL que se recomienda revisar.

Todo PostgreSQL by Abatic

Procesos

Procesos “comunes”.

- El postmaster.
- Procesos de mantenimiento.
- Procesos de utilidad.
- Procesos con funcionalidad adicional (9.3+).

Todo PostgreSQL by Abatic

Procesos

Postmaster:

- Proceso maestro de la Base de Datos.
- Controla conexiones y gestiona el cluster.
- Se encarga de lanzar el resto de procesos.
- Hace las tareas de supervisor y se encarga de reiniciar los procesos que hayan finalizados.

Todo PostgreSQL by Abatic

Procesos

Postmaster:

- Es el encargado de escuchar nuevas conexiones, donde crea un nuevo proceso llamado “postgres” en segundo plano por cada sesión de usuario.
- Puede escuchar en múltiples direcciones IP, pero sólo en un único puerto TCP (o un único socket de dominio UNIX para conexiones locales).

Todo PostgreSQL by Abatic

Procesos

Postmaster:

- Gestiona exactamente un cluster de la base de datos, que consiste en un único directorio de datos ($PGDATA), éste puede contener múltiples BD.

Todo PostgreSQL
by Abatic

Procesos

Procesos de backend:

- Hacen todo el trabajo de la base de datos.
- Un proceso Postgres Backend por conexión.
- Se encarga de aceptar la conexión, autentifica al usuario.
- Controlado por el postmaster.
- Utiliza la memoria compartida (shared buffers) para poder interactuar con el resto de procesos.

Todo PostgreSQL
by Abatic

Procesos

- Comunicación mediante memoria compartida (System V o memoria de la aplicación).
- Procesos de utilidad
  - BGWriter + Checkpointer: bloques modificados de memoria a disco.
  - Stats collector.
  - Logs collector/logger.

Todo PostgreSQL by Abatic

Procesos

- Procesos de utilidad
  - WAL writer.
  - WAL Archiver.
  - Auto-Vacuum Launcher (vacuum).
  - WAL Sender.
  - WAL Receiver.

Todo PostgreSQL by Abatic

Procesos

Background writer:

- Proceso encargado de recoger todas las páginas modificadas desde shared buffer y escribirlas a disco.
- Puede provocar una sobrecarga de I/O en el sistema.

Checkpointer:

- Realiza un punto de control en la secuencia de escritura de los WAL.

Todo PostgreSQL by Abatic

Procesos

Stats collector:

- Obtiene estadísticas de uso de las relaciones.
- Son muy importantes de cara al rendimiento.
- Dan información al planificado de consultas sobre los datos y valores habituales que se encuentran en la base de datos.

Todo PostgreSQL by Abatic

Procesos

Logging collector:

- Redirige los mensajes de error a syslog, eventlog o a ficheros log.
- Los log son la bitácora de nuestra base de datos.

Todo PostgreSQL by Abatic

Procesos

WAL Writer:

- Proceso encargado de escribir y sincronizar los ficheros WAL hacia disco cuando ocurra una transacción.
- De esta forma permite que el sistema pueda recuperarse en caso de desastre.

Todo PostgreSQL by Abatic

Write-Ahead-Logging (WAL)

- WAL (en español, Registro de Escritura Anticipada). Realiza una escritura en el fichero de cada sentencia ejecutada antes de que tenga lugar y es utilizado para poder recuperar bases de datos en caso de rotura del sistema o fallo.

Todo PostgreSQL by Abatic

Procesos

Archive:

- Se encarga de **copiar los ficheros WAL almacenados en el directorio pg_wal**, a otros directorios.

Esta copia se realizará en el directorio especificado en el parámetro **archive_command** del archivo de configuración postgresql.conf.

Todo PostgreSQL by Abatic

pg_receivewal

- El proceso Archive no transmite los ficheros WAL en tiempo real, tiene un pequeño retraso.
- Para enviar los ficheros WAL al directorio local en tiempo real, disponemos de la herramienta pg_receivewal.
- Streaming Replication Protocol

Todo PostgreSQL by Abatic

Procesos

Autovacuum Launcher:

- Es opcional, por defecto se encuentra activo.
- Se encarga de automatizar comandos VACUUM y ANALIZE.
- Crea un proceso autovacuum worker por cada BD hasta alcanzar el límite indicado por el parámetro autovacuum_max_workers.

Todo PostgreSQL by Abatic

Procesos

Autovacuum Worker:

- Se encarga de chequear cada tabla e índice de una base de datos indicada ejecutando ANALYZE o VACUUM.

Todo PostgreSQL by Abatic

Procesos

WAL Receiver:
Proceso encargado de recibir los ficheros WAL enviados desde el servidor Master.

WAL Sender:

- Proceso encargado de enviar los ficheros WAL a los servidores Slaves.
- Hay uno por cada servidor Slave.

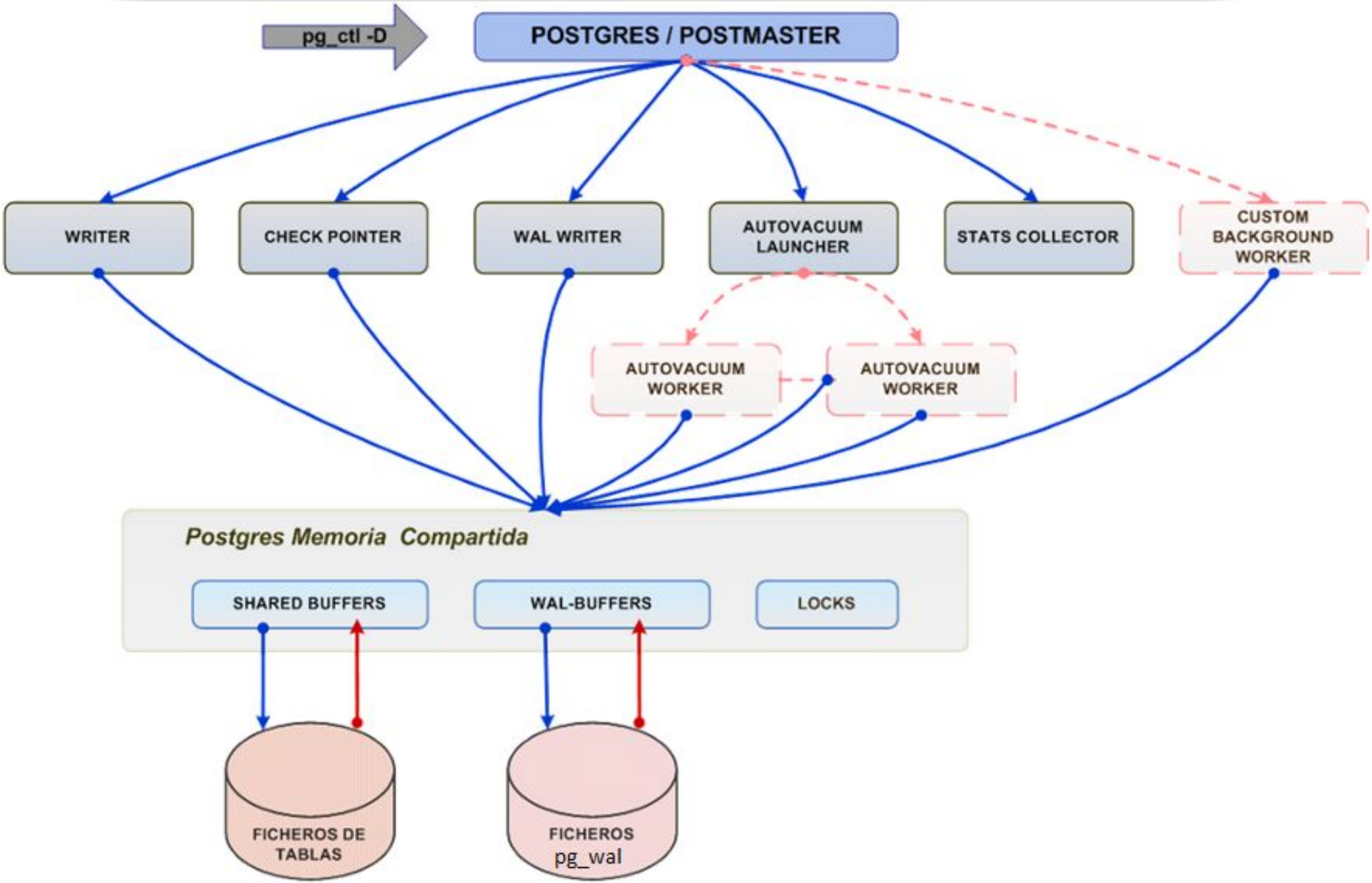
Todo PostgreSQL by Abatic

Procesos

Background workers:

- Un mecanismo para lanzar procesos de usuario cuyo ciclo de vida está gestionado por el postmaster.
- Permite ejecutar código de usuario de forma similar a como lo hacen los procesos de PostgreSQL.

Todo PostgreSQL by Abatic



Todo PostgreSQL
by Abatic

Buffering de Lectura/Escritura de Disco

- **Shared Buffers**
  - Se encuentran en Memoria.
  - Lecturas lógicas de la caché.
  - Reduce lecturas desde el S.O.
  - Páginas de 8 KB.

Todo PostgreSQL by Abatic

Buffering de Lectura/Escritura de Disco

- **WAL Buffers**
  - Se encuentran en Memoria.
  - Durabilidad de las transacciones.

Todo PostgreSQL by Abatic

Conexión

1) Postmaster se encarga recibir peticiones de clientes:
   - Puerto configurable.
   - Socket UNIX y/o TCP.

2) Utilización de memoria compartida.

Todo PostgreSQL by Abatic

Conexión

3) Pre-autentifica la conexión:

- Ajustes del fichero pg_hba.conf.
- Métodos externos.

4) Creación de nuevos Backend.

2) Supervisión de los Backend.

Todo PostgreSQL by Abatic

Funcionamiento

Cuando una aplicación desea conectarse a PostgreSQL.

1) Inicia una conexión con el servidor mediante el protocolo FE/BE, que funciona sobre TCP/IP.

2) La aplicación cliente utilizara un driver JDBC, ODBC o la librería pq de PostgreSQL.

Todo PostgreSQL by Abatic

Funcionamiento

3) Una vez realizada la conexión, se realiza la autenticación con el usuario. Y el proceso postmaster creará un nuevo backend (proceso postgres), que será el responsable de atender la conexión del usuario y ejecutar las consultas.

1) El backend opera siempre a través de la memoria de PostgreSQL (shared buffers).

Todo PostgreSQL by Abatic

Funcionamiento

5) Cuando un dato debe leerse del disco, se llevará la tabla del disco a la memoria compartida de los procesos, y éstos la leerán o modificarán sobre el shared buffers.

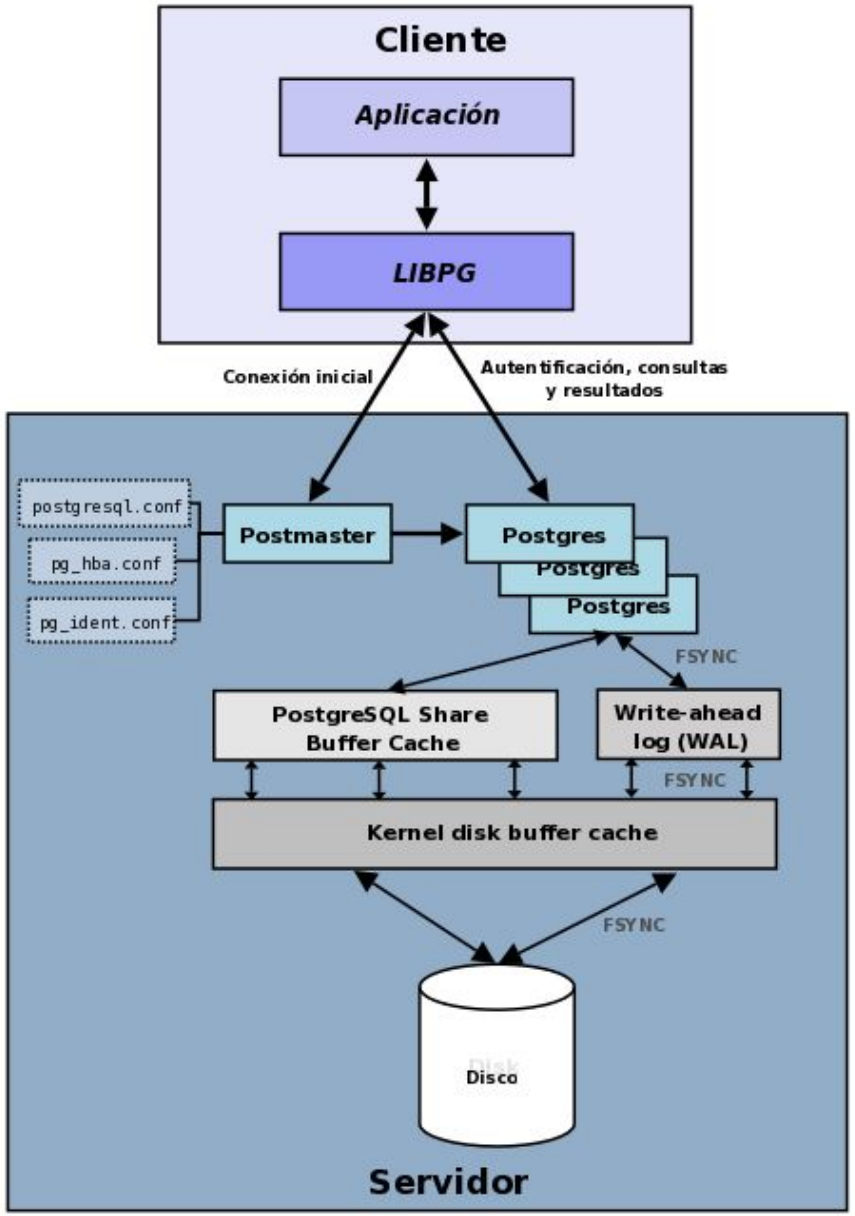
1) Cuando un dato debe ser escrito, éste se modifica igualmente sobre shared buffers y, cuando la operación se marca como lista para ser aplicada (COMMIT) se generará los registros WAL.

Todo PostgreSQL by Abatic

Funcionamiento

7) Estos WAL se escriben en disco mediante una operación de fsync, que indica al S.O. que realice una operación de write-though de dichos cambios en los ficheros WAL.

Todo PostgreSQL by Abatic



El proceso principal es el encargado de crear procesos Postgres.

Los diferentes procesos interactuarán mediante el Buffer Caché.

Para conocer qué procesos están interactuando en nuestro sistema:

```
ps -aux | grep postgres
```

Todo PostgreSQL by Abatic

Cursos de PostgreSQL para DBA y Developers

Todo PostgreSQL
by Abatic

TODOPOSTGRESQL.COM