

Todo PostgreSQL

Arquitectura del sistema

PostgreSQL

INSTRUCTOR:

José Segovia

<info@todopostgresql.com>



Todo PostgreSQL

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

Todo PostgreSQL

Introducción

- PostgreSQL es un sistema multiproceso.
- Postmaster gestiona los multiples procesos.
- Procesos de usuario
 - Un usuario es atendido por un solo proceso.

Todo PostgreSQL

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

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

Conexión

1) Postmaster se encarga recibir peticiones de clientes:

- Puerto configurable.
- Socket UNIX y/o TCP.

2) Utilización de memoria compartida.

Todo PostgreSQL

Conexión

3)Pre-autentifica la conexión:

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

4)Creación de nuevos Backend.

5)Supervisión de los Backend.

Todo PostgreSQL

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

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

Todo PostgreSQL

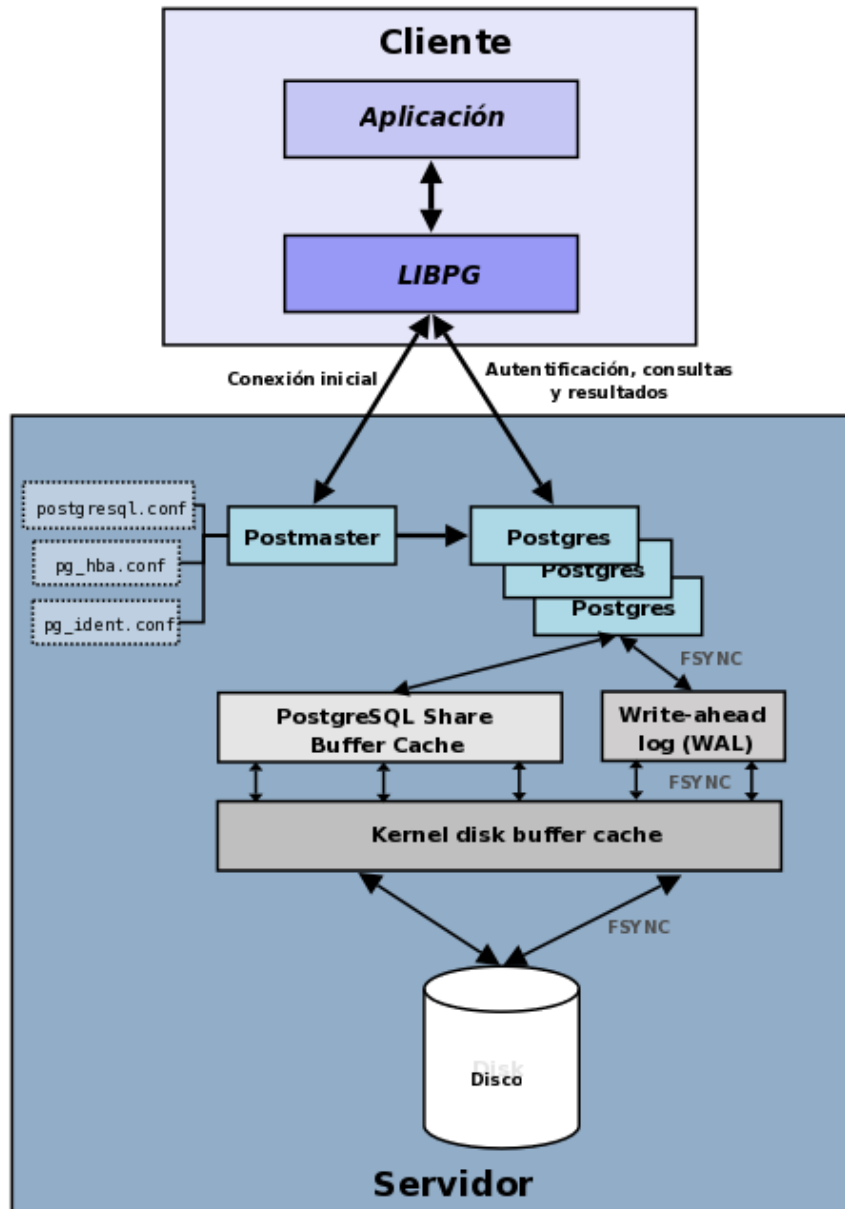
Funcionamiento

- 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.
- 6) 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.

Funcionamiento

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

Todo PostgreSQL



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

Procesos

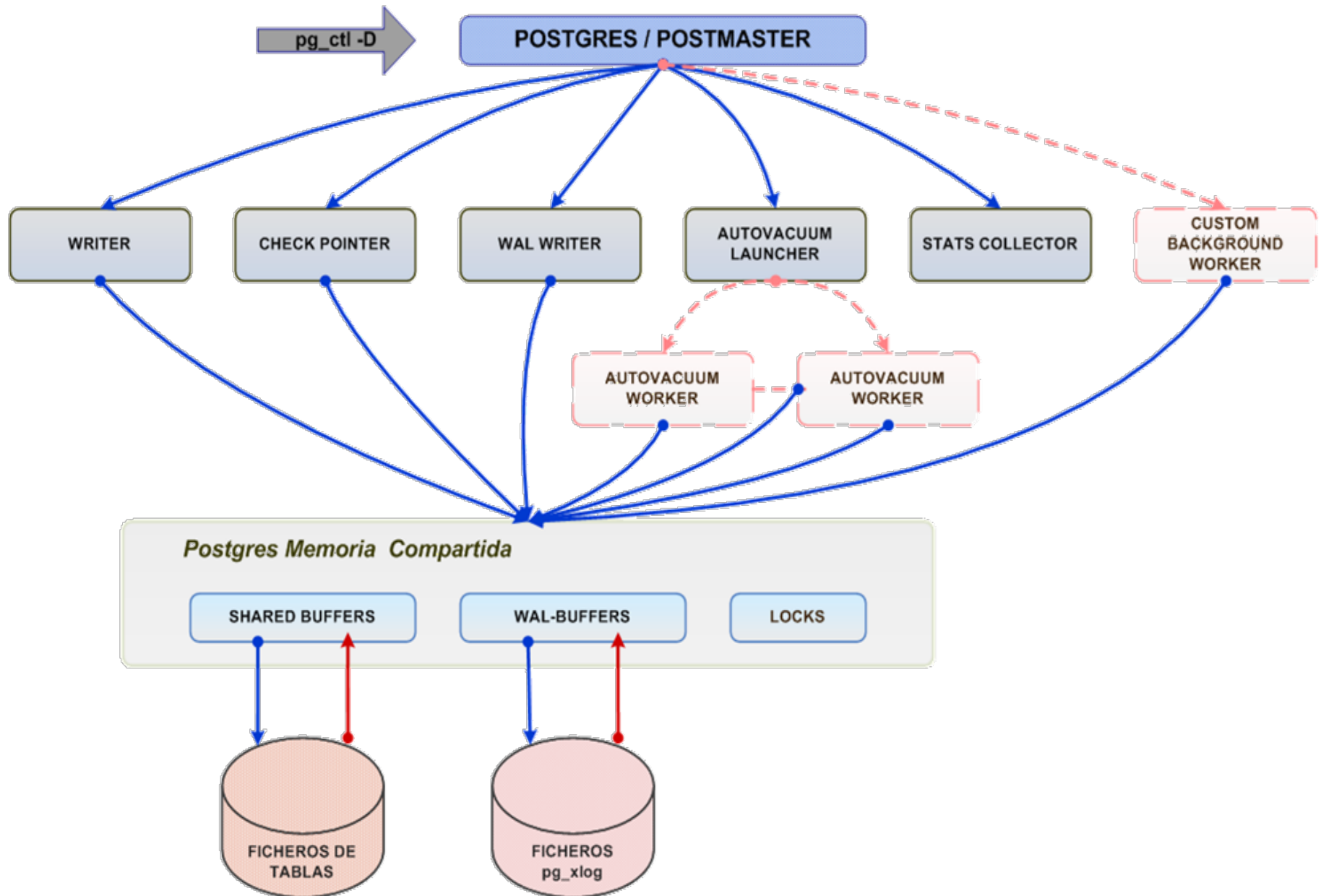
- Comunicación mediante memoria compartida (System V o memoria de la aplicación).
- Procesos de utilidad
 - ♦ BGWriter + Checkpointer: bloques modificados de memoria a disco.
 - ♦ Stats collector.
 - ♦ Logs collector/logger.

Todo PostgreSQL

Procesos

- Procesos de utilidad
 - ♦ WAL writer.
 - ♦ WAL Archiver.
 - ♦ Auto-Vacuum Launcher (vacuum).
 - ♦ WAL Sender: envía las modificaciones de una BD a otra de replica.

Todo PostgreSQL



Todo PostgreSQL

Procesos

Postmaster:

- Proceso principal de PostgreSQL.
- Es el encargado de escuchar en un puerto o socket por conexiones entrantes de clientes.
- Crea procesos hijos, llamados postgres.

Todo PostgreSQL

Procesos

Postgres:

- Es el proceso hijo que se encarga de autenticar peticiones de conexión.
- Gestiona consultas y manda los resultados a las aplicaciones clientes.

Todo PostgreSQL

Procesos

Background writer:

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

Checkpoint:

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

Todo PostgreSQL

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

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

Procesos

Autovacuum Worker:

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

Write-Ahead-Logging (WAL)

- Los procesos escriben la información en el buffer.
- WAL Writer: escribe a disco en pg_xlog (pg_wal).
 - Cuando se realiza un commit.
 - Al llegar a un tamaño según la configuración (max_wal_size).
 - Se realiza en grupo de segmento de 16Mb.

Todo PostgreSQL

Write-Ahead-Logging (WAL)

- WAL Archive: se encarga de **copiar los ficheros WAL almacenados en pg_xlog**, en otra ubicación al llenarse el directorio pg_xlog.

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

Todo PostgreSQL

Procesos

Stats collector:

- Es opcional, por defecto se encuentra activo.
- Recolecta información acerca de la actividad del cluster.
 - Números de accesos a una tabla o índice.
 - Número de tuplas de una tabla.
 - Información acerca de las acciones realizadas por ANALYZE o VACUUM por cada tabla.
- Añade una sobre carga para ejecutar consultas, pero permite poder elegir la mejor opción al planificador de consultas.

Todo PostgreSQL

Procesos

WAL Receiver:

Proceso encargado de recibir los ficheros WAL enviados desde la base de datos primaria.

WAL Sender:

- Proceso encargado de enviar los ficheros WAL a la base de datos en Standby.
- Hay uno por cada servidor Standby.

Todo PostgreSQL

Procesos

Logger:

- Es opcional, por defecto se encuentra desactivado.
- Todos los procesos de utilidad junto con los procesos de backends y postmaster se encargan de adjuntarle información a este proceso.
- Toda la información será almacenada en pg_log (log).
- Un incremento en la cantidad a registrar producirá un aumento en la I/O.

Procesos

Logger:

- Es opcional, por defecto se encuentra desactivado.
- Todos los procesos de utilidad junto con los procesos de backends y postmaster se encargan de adjuntarle información a este proceso.
- Toda la información será almacenada en pg_log.
- Un incremento en la cantidad a registrar producirá un aumento en la I/O.

Buffering de Lectura/Escritura de Disco

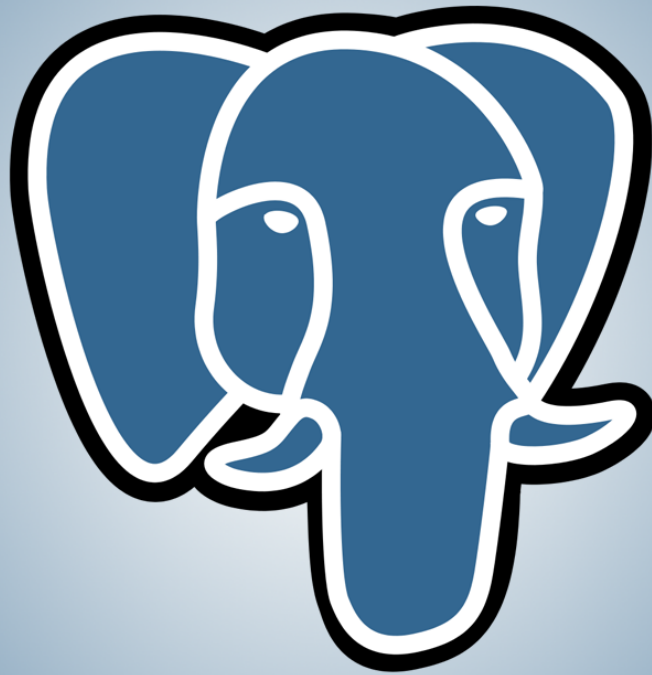
- Shared Buffers
 - Se encuentran en Memoria.
 - Lecturas lógicas de la caché.
 - Reduce lecturas desde el S.O.
 - 1000 shared buffers.
 - Páginas de 8 KB.

Buffering de Lectura/Escritura de Disco

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

Commit y Checkpoint

- 1) Se han realizado cambios en la base de datos.
- 2) Se realiza el Commit.
- 3) Se escribe desde memoria al disco los archivos WAL (pg_xlog o pg_wal).
- 4) Se realiza el Checkpoint, escribe desde la memoria hacia los archivos de datos.



todopostgresql.com