

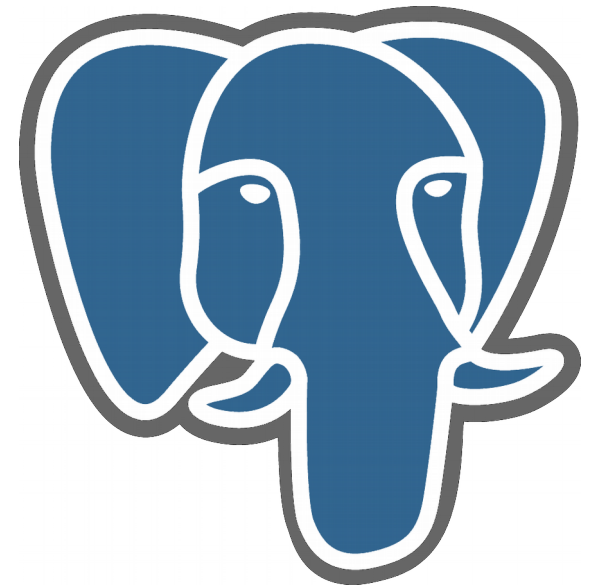


# **Administración Avanzada de PostgreSQL**

## **Conceptos de clusterización**

INSTRUCTOR:

José Segovia <[info@todopostgresql.com](mailto:info@todopostgresql.com)>



# **Conceptos de clusterización en PostgreSQL**

# Clusterización

- Se trata de un **conjunto de máquinas individuales** interconectadas entre si que se comportan como un todo.
- A cada servidor se le conoce como **nodo**.
- No confundir con el **cluster de PostgreSQL** (una instancia).

# **Posibilidades de Alta Disponibilidad con PostgreSQL**

# Opciones de Alta Disponibilidad

- Existen diversas herramientas para implementar Alta disponibilidad y escalar PostgreSQL en varios servidores.
- Almacenamiento compartido.
- Redundancia de datos (Replica).
  - Streaming Replication.
  - Maestro – Esclavo (Slony).
  - Multi – Maestro (Bucardo).

Característica	Failover del disco compartido	Replicación del sistema de archivos	Envío de registro de transacciones	Repetición maestro-Standby basada en disparadores	Middleware de replicación basado en declaraciones	Replicación Asíncrono Multimaster	Replicación síncrona multimaster
Implementación más común	NAS	DRBD	Streaming Repl.	Slony	Pgpool-II	Bucardo	
Método de comunicación	Disco compartido	Bloques de disco	WAL	Filas de la tabla	SQL	Filas de la tabla	Filas de tabla y bloqueos de fila
No requiere hardware especial		•	•	•	•	•	•
Permite múltiples servidores maestros					•	•	•
No hay sobrecarga del servidor maestro	•		•		•		
No hay que esperar varios servidores	•		Con sincronización desactivada	•		•	
El fallo maestro nunca perderá datos	•	•	Con sincronización activada		•		•
En espera aceptan consultas de sólo lectura			Con calor	•	•	•	•
Granularidad por tabla				•		•	•
No es necesaria la resolución de conflictos	•	•	•	•			•

# Todo PostgreSQL

Program	License	Maturity	Replication Method	Sync	Connection Pooling	Load Balancing	Query Partitioning
<b>PgCluster</b>	BSD	Not production ready	Master-Master	Synchronous	No	Yes	No
<b>pgpool-I</b>	BSD	Stable	Statement-Based Middleware	Synchronous	Yes	Yes	No
<b>Pgpool-II</b>	BSD	Recent release	Statement-Based Middleware	Synchronous	Yes	Yes	Yes
<b>slony</b>	BSD	Stable	Master-Slave	Asynchronous	No	No	No
<b>Bucardo</b>	BSD	Stable	Master-Master, Master-Slave	Asynchronous	No	No	No
<b>Londiste</b>	BSD	Stable	Master-Slave	Asynchronous	No	No	No
<b>Mammoth</b>	BSD	No longer maintained	Master-Slave	Asynchronous	No	No	No
<b>rubyrep</b>	MIT	No longer maintained	Master-Master, Master-Slave	Asynchronous	No	No	No
<b>Bi-Directional Replication</b> 📁	PostgreSQL (BSD)	Recent release	Master-Master (no triggers needed)	Asynchronous	No	No	No
<b>pg_shard</b> 📁	LGPL	Recent release	Statement-based Middleware (as an extension)	Synchronous	No	Yes	Yes
<b>pglogical</b> 🔗	PostgreSQL	Recent release	Master-Slave	Asynchronous	No	No	No
<b>Postgres-XL</b> 🔗	PostgreSQL	Recent release	MPP Postgres, scalable writes & reads	Synchronous	Yes	Yes	Yes
<b>Citus</b> 📁	AGPL	Recent release	MPP Postgres, scalable writes & reads	Asynchronous or Synchronous	Yes	Yes	Yes

# Todo PostgreSQL

# **Soluciones incorporadas en la Base de Datos**

- log shipping
- replicación



# **Soluciones externas a la Base de Datos**

- Almacenamiento compartido en red (NAS)
- DRBD
- PgPool

# PgPool

- PgPool: es un intermediario entre la aplicación y la BD que **permite administrar un pool (conjunto) de conexiones persistentes siempre abiertas y disponible para su uso.**
- Inconveniente: **consume una cantidad de RAM significativa**, debido a que pgPool inicia sus propios procesos.

# PgPool

- Ventajas

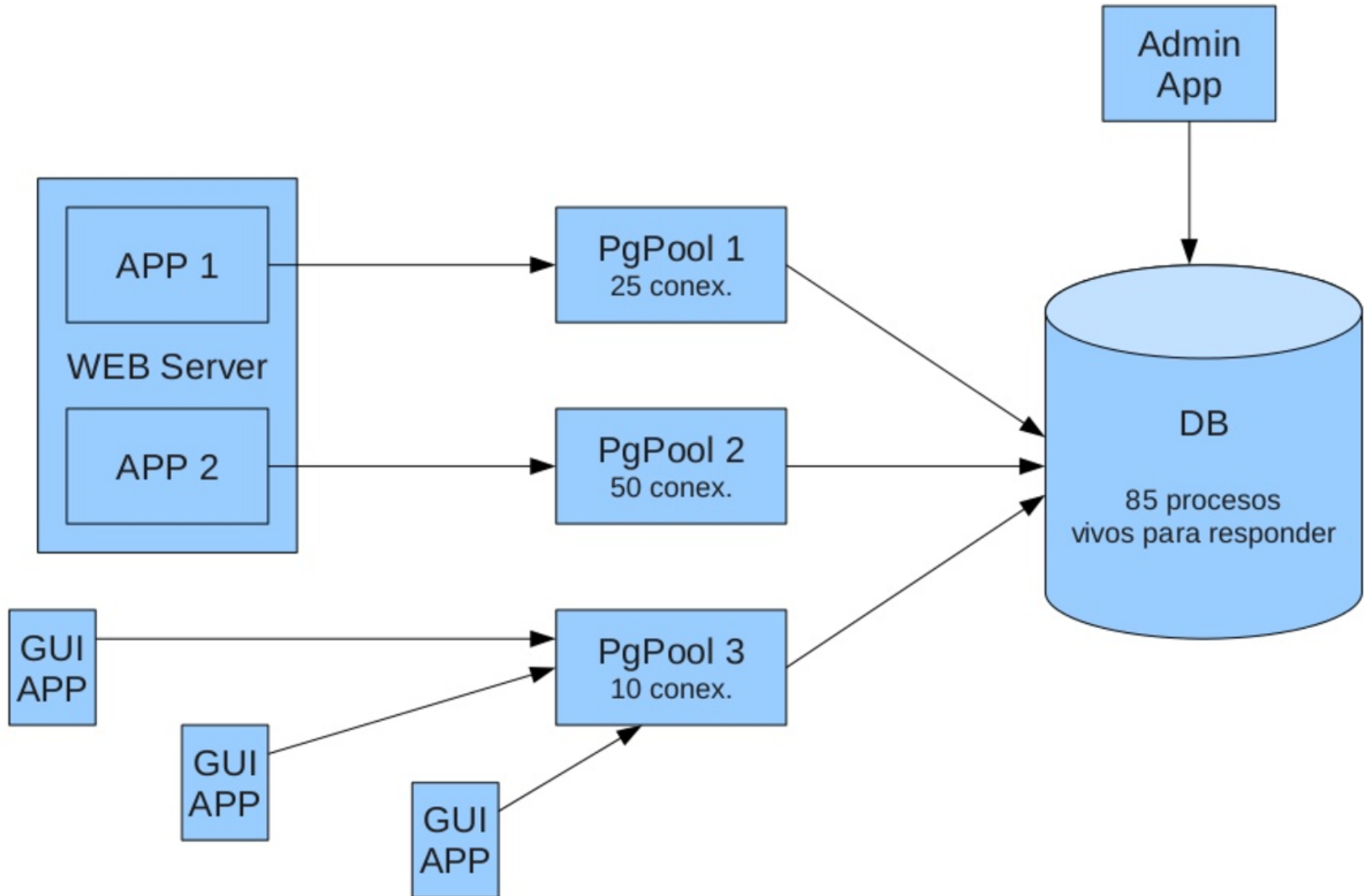
1. **Conexión rápida a la BD**, sin demorar en abrir procesos por conexión y reducir el costo.

**Guarda las conexiones y las reutiliza.**

2. Es necesario **limitar las conexiones concurrentes a la BD** para balancear el uso de recursos en el servicio.

3. Licencia **BSD**.

# PgPool



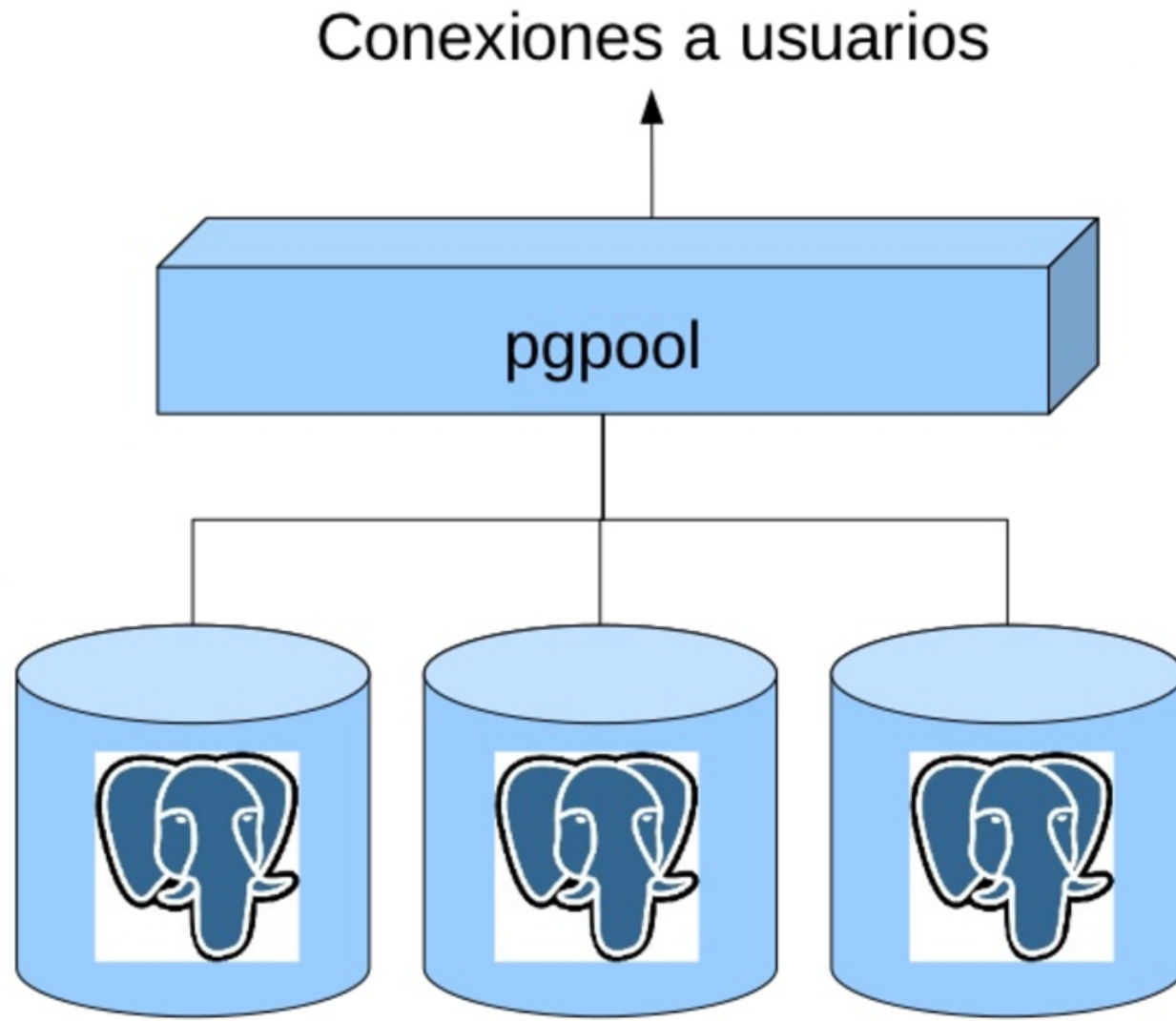
Todo PostgreSQL

# PgPool

4. Puede **gestionar varios servidores**.

5. Para reducir la carga, **distribuye las consultas "SELECT" entre los distintos servidores mejorando el rendimiento general del sistema** proporcionalmente al número de servidores PostgreSQL. Funciona mejor en una situación en la que hay **muchos usuarios que ejecutan muchas consultas al mismo tiempo**.

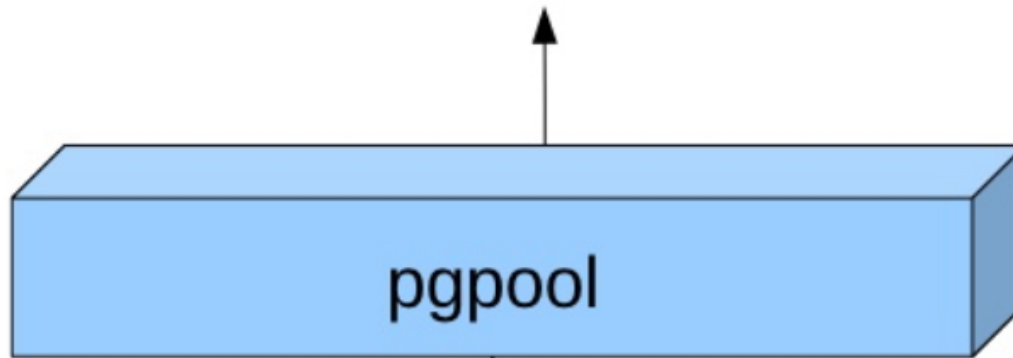
# PgPool



Todo PostgreSQL

# PgPool

Conexiones a usuarios



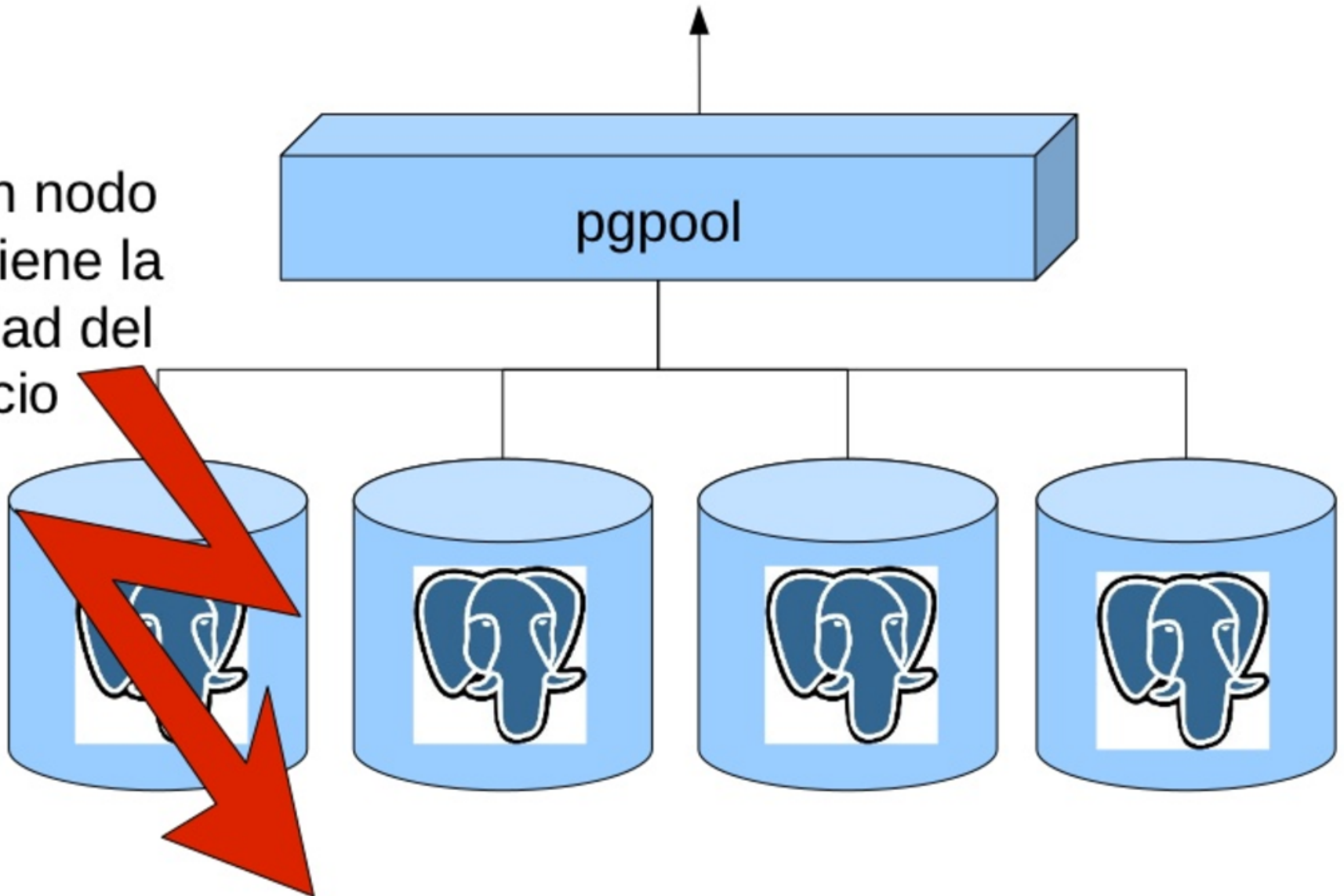
Podemos agregar  
un nodo nuevo  
sin detener el  
servicio

Todo PostgreSQL

# PgPool

Conexiones a usuarios

Si falla un nodo  
no se detiene la  
continuidad del  
servicio



Todo PostgreSQL



# PgPool

- Instalación y configuración

1. descargar el código fuente y descomprimir (tar xf pgpool-II-3.6.5.tar.gz).

2. Revisar la configuración (./configure). Es posible personalizar la construcción con uno o más opciones de comandos, por ejemplo especificar un directorio diferente para los binarios.

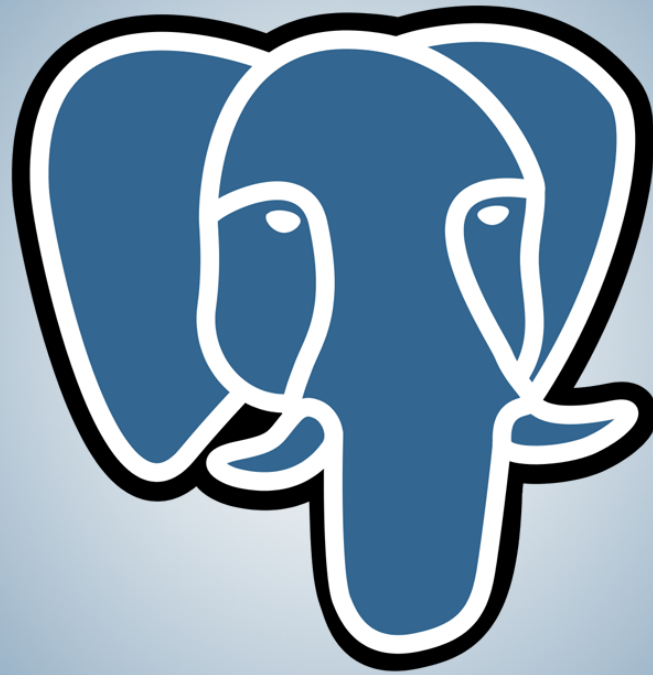
# PgPool

- Instalación y configuración

3. compilar (make) e instalar (make install)

4. Instalar pg\_pool\_recovery

```
cd pgpool-II-x.x.x/src/sql/pgpool-recovery  
make  
make install
```



[todopostgresql.com](http://todopostgresql.com)