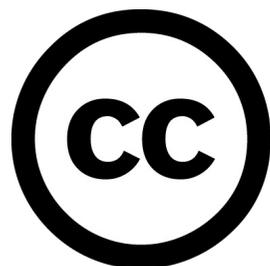


# atmantree.com

Los autores del presente documento lo ha publicado  
bajo las condiciones que especifica la licencia



Creative Commons

Attribution-NonCommercial-ShareAlike 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

En caso de dudas escriba a:

[info@atmantree.com](mailto:info@atmantree.com)

# Introducción a pgpool-II

Guía paso a paso



# Agenda

- ¿Qué es pgpool-II?
  - Arquitectura
- Instalación
  - Paquetería
  - Compilación
  - pgpool-II admin
- Configuración
  - Replicación
  - Maestro Esclavo
  - Stream
  - Otros usos de pgpool-II
- Otros detalles de configuración
- Recomendaciones para el uso eficaz de pgpool
- Ejercicio realizado durante el taller

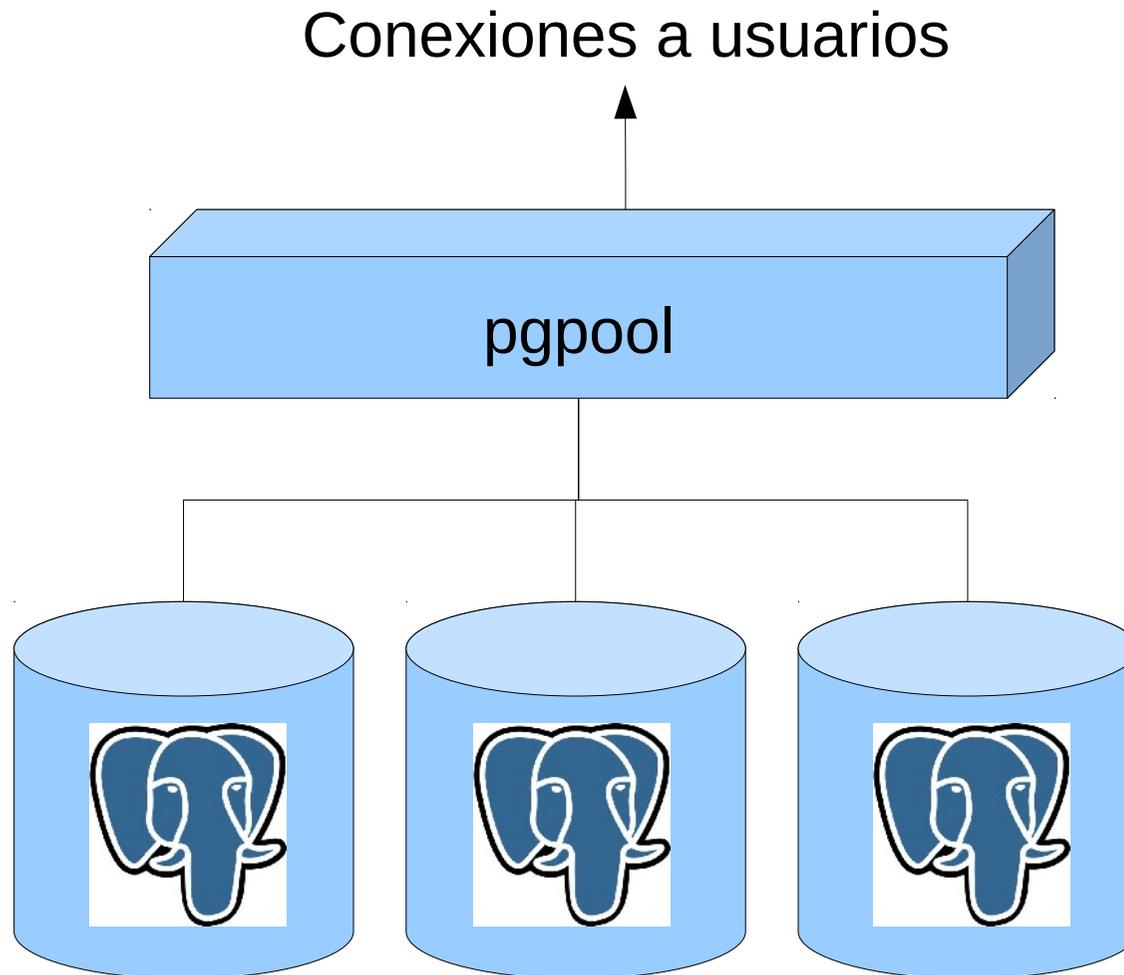
# ¿Qué es pgpool-II?

pgpool-II es un middleware que trabaja entre servidores y clientes de PostgreSQL. El mismo provee las siguientes funcionalidades:

- Agrupado de Conexiones (connection pooling)
- Replicación
- Balance de Cargas (load balance)
- Limitación de Conexiones
- Distribución de Consultas (parallel query)

# ¿Qué es pgpool-II?

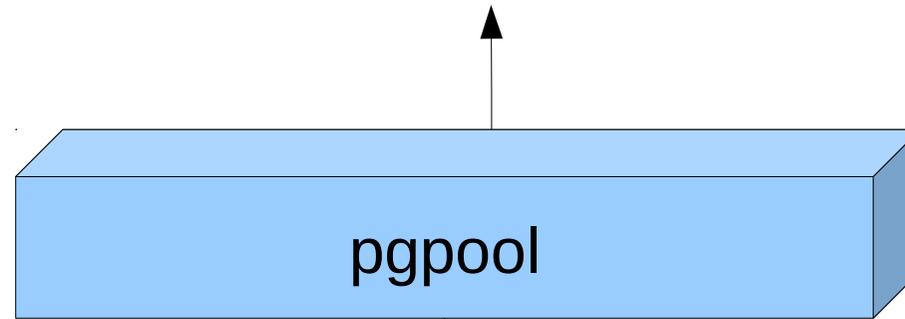
## Arquitectura



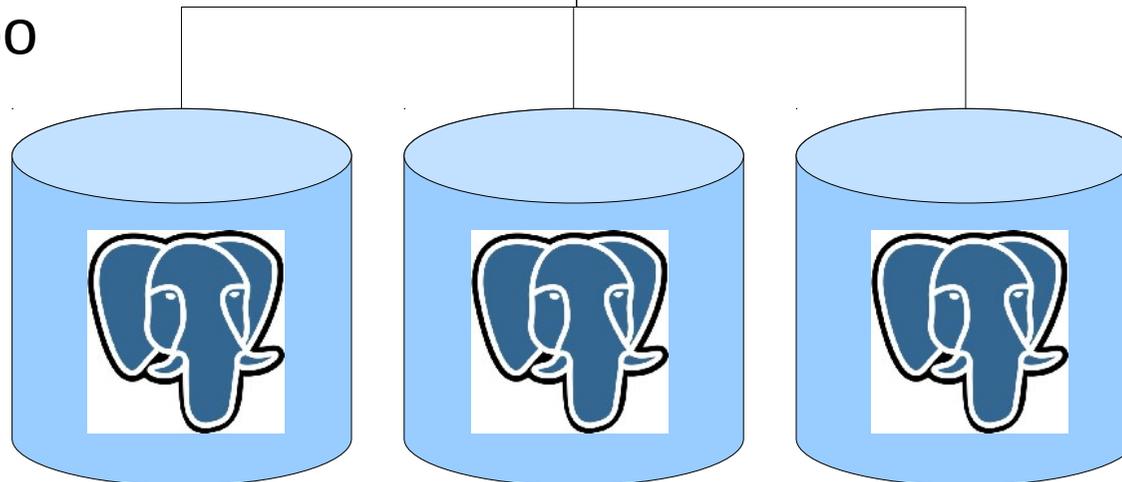
# ¿Qué es pgpool-II?

Los usuarios no se conectan directamente a los nodos

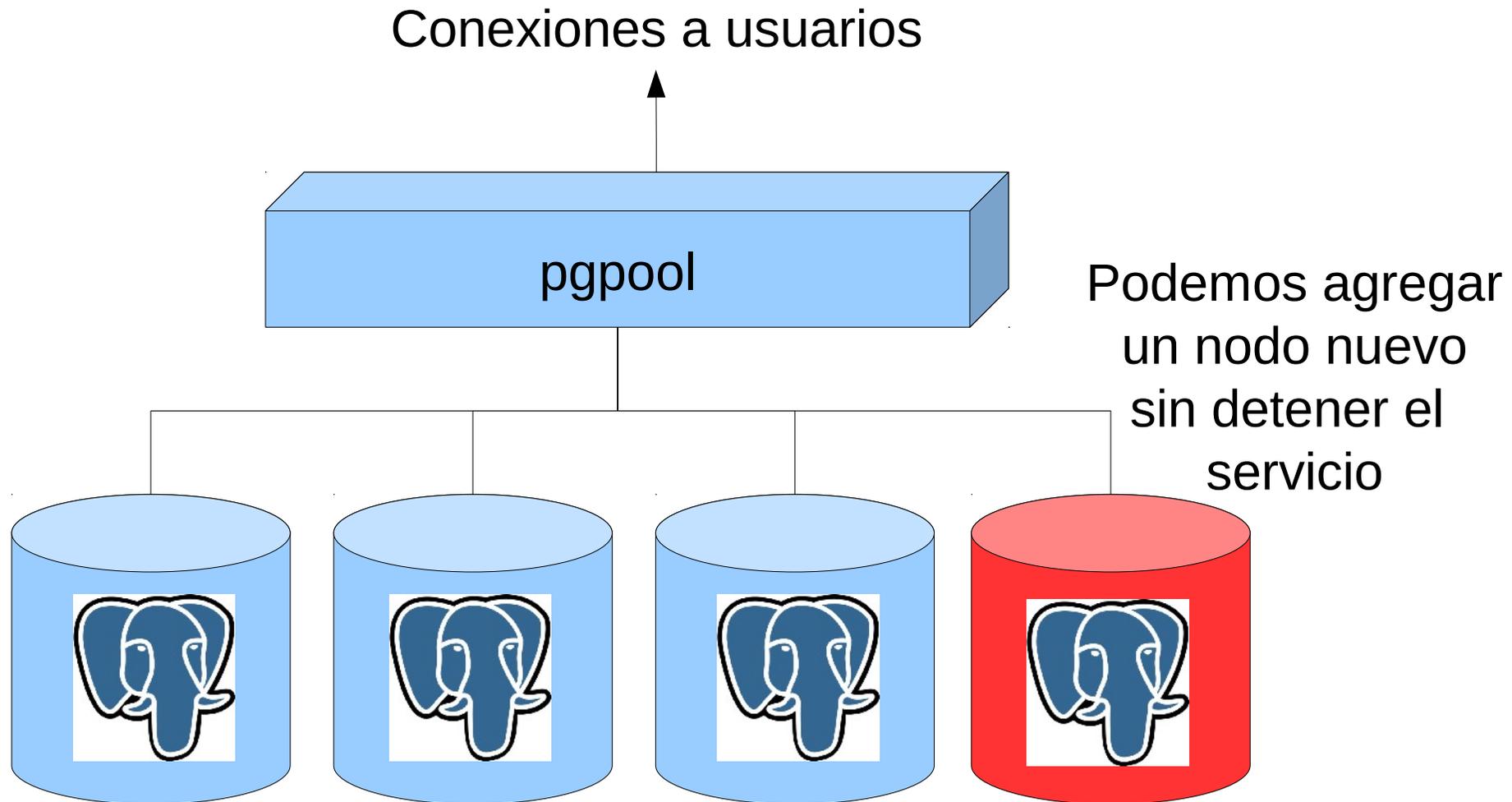
Conexiones a usuarios



Tenemos un grupo de servidores bajo el pool



# ¿Qué es pgpool-II?



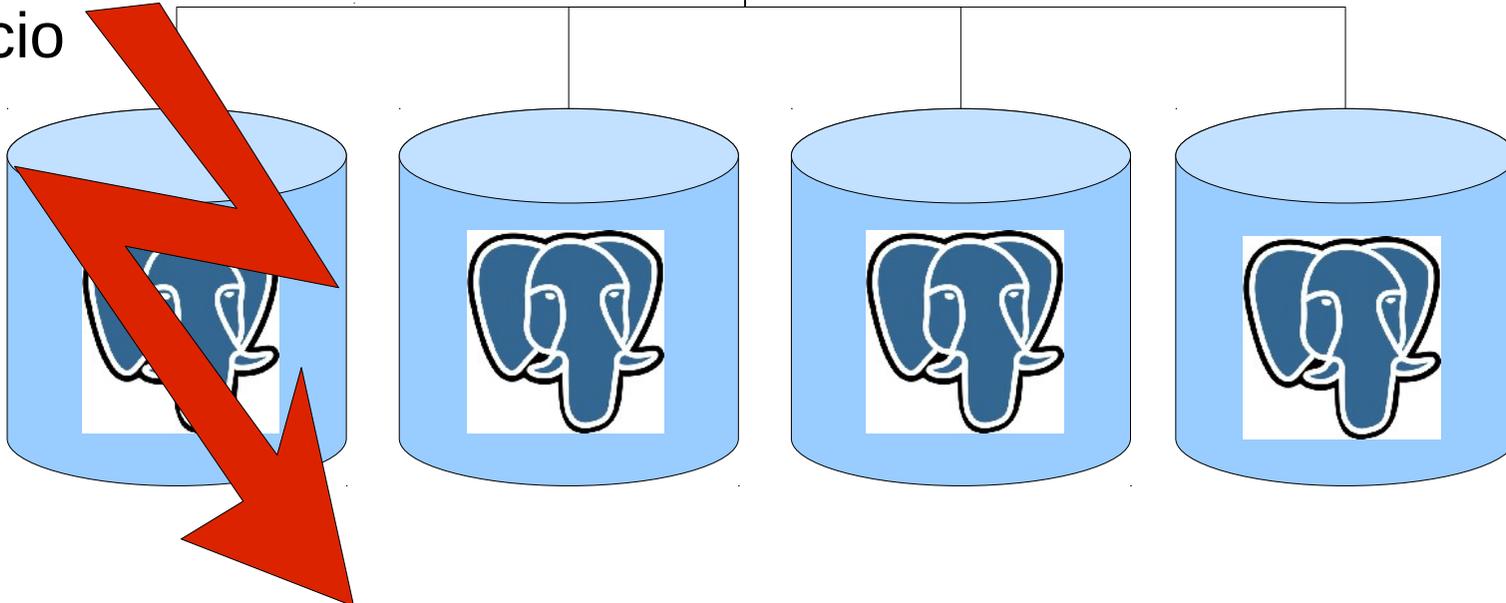
# ¿Qué es pgpool-II?

## Arquitectura

Conexiones a usuarios

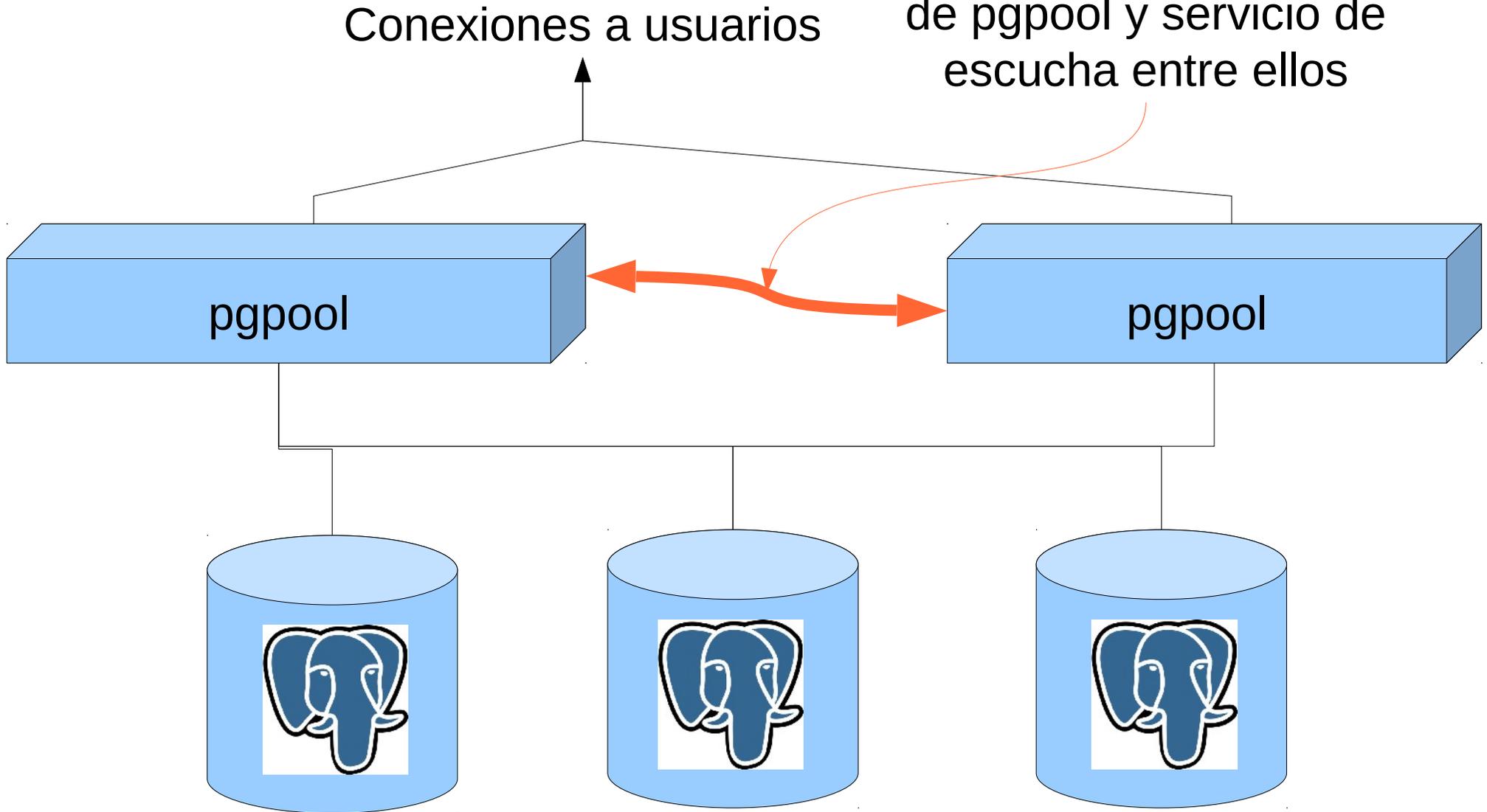


Si falla un nodo  
no se detiene la  
continuidad del  
servicio



# ¿Qué es pgpool-II?

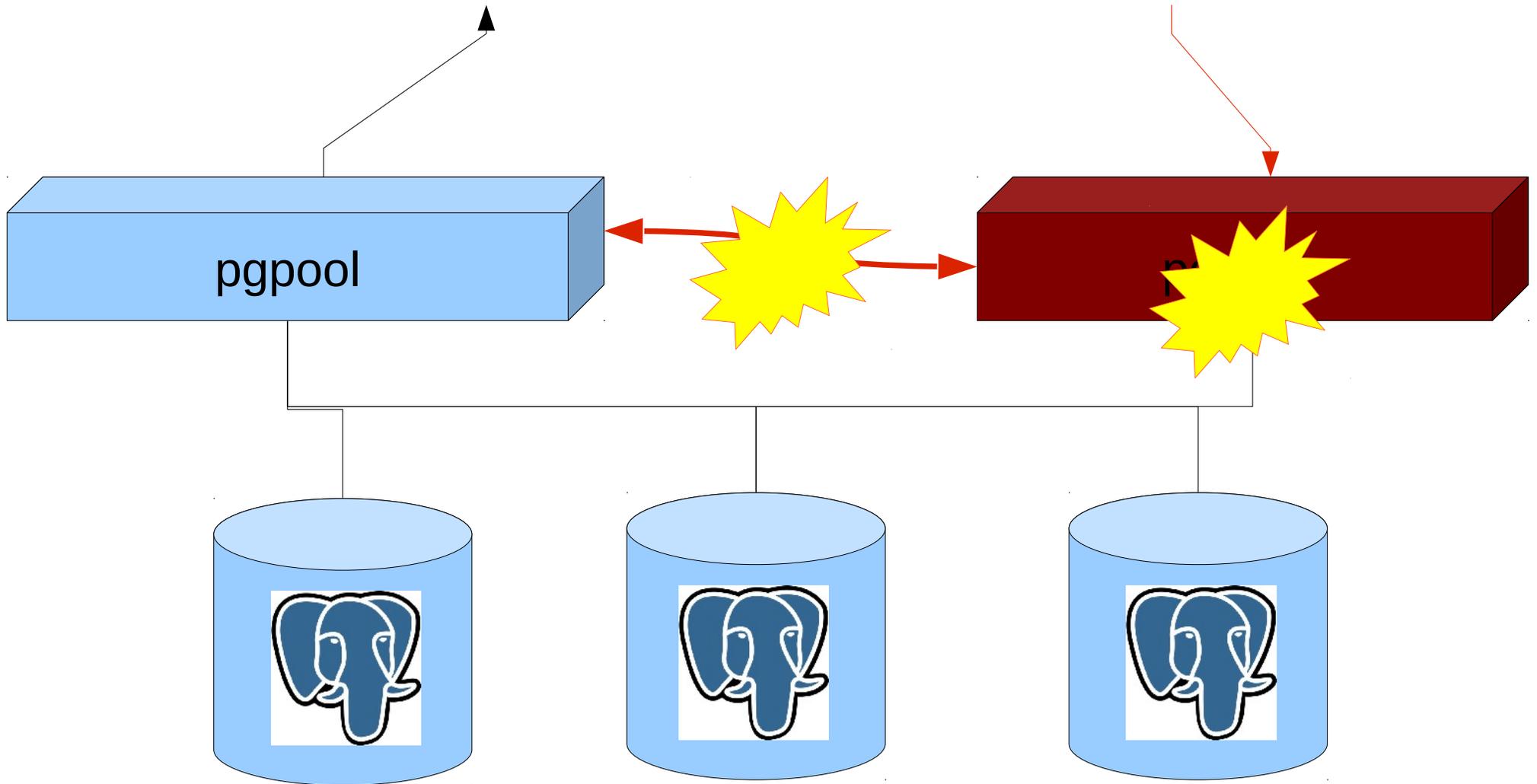
Podemos incluso tener redundancia de servicios de pgpool y servicio de escucha entre ellos



# ¿Qué es pgpool-II?

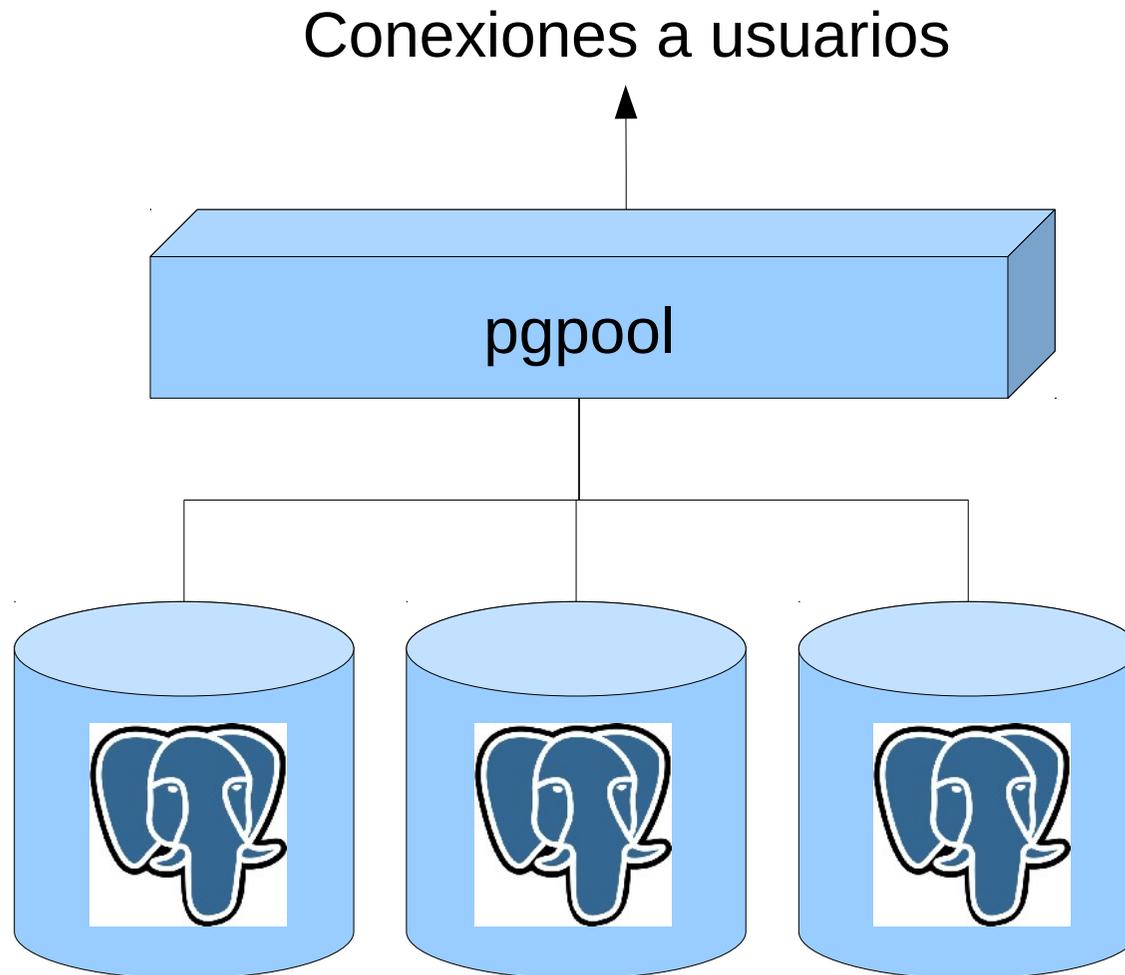
Si falla un pgpool tenemos  
seguiremos conectados  
mediante el otro pgpool

Conexiones a usuarios



# ¿Qué es pgpool-II?

## Arquitectura



# Instalación

## Paquetería

### Instrucción

```
$ aptitude -t squeeze-backports install pgpool2
```

### A Favor:

Disponibilidad y funcionamiento

### En Contra:

Versión disponible algo antigua

# Instalación

## Compilación (recomendado)

### Pre-requisitos

```
$ aptitude install postgresql-9.0 postgresql-server-dev-9.0
```

```
$ aptitude install build-essential libssl-dev libreadline5-dev
```

```
$ aptitude install zlib-bin zlib1g-dev libpq-dev
```

```
$ aptitude install krb5-multidev comerr-dev libkrb5-dev
```

(pueden escribir todos los pre-requisitos en una sola línea)

# Instalación

## Compilación

### Instrucción

```
$ wget http://pgfoundry.org/frs/download.php/3114/pgpool-II-3.1.tar.gz  
$ tar -xzf pgpool-II-3.1.tar.gz  
$ cd pgpool-II-3.1  
$ ./configure --prefix=/opt/pgpool3/  
$ make  
$ make install  
$ cd sql/pgpool-regclass
```

# Instalación

## Compilación

### Instrucción

```
$ make
```

```
$ make install
```

```
$ psql -p ##### -U postgres -h ##### -f pgpool-regclass.sql template1
```

```
$ psql -p $$$$ -U postgres -h $$$$$$ -f pgpool-regclass.sql template1
```

```
$ cd ../../
```

*(sustituya los valores velados con ##### y \$\$\$\$ por los correspondientes a las instancias de postgres que utilizará como nodos del cluster)*

# Instalación

## Compilación

### Instrucción

```
$ cd sql/pgpool-recovery
```

```
$ make
```

```
$ make install
```

```
$ psql -p ##### -U postgres -h ##### -f pgpool-recovery.sql template1
```

```
$ psql -p $$$$ -U postgres -h $$$$$$ -f pgpool-recovery.sql template1
```

```
$ cd ..
```

```
$ psql -p ##### -U postgres -h ##### -f insert_lock.sql template1
```

```
$ psql -p $$$$ -U postgres -h $$$$$$ -f insert_lock.sql template1
```

```
$ cd
```

# Instalación

## Compilación

### Instrucción

- Configure los archivos en `/opt/pgpool3/etc/`
- Levante pgpool con el comando:  
`$ /opt/pgpool3/bin/pgpool -n -f /opt/pgpool3/etc/pgpool.conf`

Fin :-D

# Instalación

## Compilación

A Favor:

Versión Actualizada y últimos avances

En Contra:

Pre-requisitos y configuración más explícita.

# Instalación

## pgpool Admin

```
$ wget http://pgfoundry.org/frs/download.php/3149/pgpoolAdmin-3.1.0.tar.gz
```

```
$ tar -xzf pgpoolAdmin-3.1.0.tar.gz
```

```
$ cd pgpoolAdmin-3.1.0/conf/
```

```
$ nano pgmgt.conf.php
```

Y configure los datos para acceder a los nodos y a pgpool.

Posteriormente proceda a instalar el pgpoolAdmin según sus preferencias dentro del servidor web Apache.

# Configuración

- pgpool tiene 3 archivos de configuración
  - pgpool.conf
  - pcp.conf
  - pool\_hba.conf
- Dependiendo de cada escenario de HA deberemos configurar el archivo pgpool.conf
- Para los configurar los accesos al pool se utiliza el archivo pool\_hba.conf
- Es altamente recomendable que todos los accesos a los backends del pool se restrinjan solamente a pgpool

# Escenario 1 (Replicación)

- Para configurar pgpool para replicación debemos tener en cuenta inicialmente los siguientes valores para la configuración el archivo `pgpool.conf`

```
replication_mode = on
```

```
insert_lock = on
```

```
black_function_list = 'nextval,setval'
```

```
master_slave_mode = off
```

# Escenario 2 (Maestro-Esclavo)

- Para configurar pgpool para Maestro-Esclavo debemos tener en cuenta inicialmente los siguientes valores para la configuración el archivo pgpool.conf

```
replication_mode = off
```

```
insert_lock = off
```

```
black_function_list = 'currval,lastval,nextval,setval'
```

```
master_slave_mode = on
```

# Escenario 3 (Stream)

- Para configurar pgpool para Stream debemos tener en cuenta inicialmente los siguientes valores para la configuración el archivo pgpool.conf

```
log_standby_delay = 'if_over_threshold'
```

```
replication_mode = off
```

```
insert_lock = off
```

```
black_function_list = 'currval,lastval,nextval,setval'
```

```
master_slave_mode = on
```

```
sr_check_period = 10
```

```
delay_threshold = 10000000
```

# Otros Detalles de Configuración

- Se recomienda revisar los archivos de configuración de ejemplo que trae los fuentes de pgpool con un comparador como Meld o K3Diff
- pgpool tiene cantidad de variantes dentro de su configuración que se recomienda ampliamente que visite <http://www.pgpool.net/pgpool-web/pgpool-II/doc/pgpool-en.html> para más detalles.

# Recomendaciones para el uso eficaz de pgpool

- Diseñar su cluster
- Planificar su implantación
- Entonar cada nodo de PostgreSQL
- Balancear el cluster según las capacidades de cada nodo, a mayor potencia mayor peso.
- Bloquear accesos externos a los nodos del cluster
- Configurar los accesos en el archivo `pgpool_hba.conf`

# Recomendaciones para el uso eficaz de pgpool

- Diseñe e implemente su mecanismo de recuperación de eventualidades conforme a sus requerimientos de QA
- Ante las dudas o ambigüedades que se le presenten durante la instalación y configuración evite la tentación de adivinar, use la documentación, listas de correo y otros espacios de discusión
- Recuerde, la alta disponibilidad se mide por sus resultados en continuidad de servicio.

# Ejercicio Realizado en el Taller

- Si va a repetir el ejercicio en entorno gráfico asegúrese de configurar la herramienta phpPgAdmin con acceso a los nodos del cluster y al puerto de pgpool.
- En caso de hacer el ejemplo desde la interfaz de líneas de comandos abra tantas terminales como sesiones de psql vaya a tomar en consideración (1 para pgpool + 1 por cada nodo del cluster)

# Ejercicio Realizado en el Taller

- Nos conectamos a los backends (nodos) y a pgpool
- Creamos una base de datos nueva desde la conexión a pgpool
- Validamos que la misma se haya creado en los backends

# Ejercicio Realizado en el Taller

- Nos conectamos a los backends (nodos) y a pgpool

```
$ psql -p ##### -h #####
```

- Desde la conexión a pgpool creamos una base de datos nueva

```
postgres# CREATE DATABASE bd1 ENCODING  
'utf8';
```

- Validamos que la misma se haya creado en los backends

```
postgres# \l
```

# Ejercicio Realizado en el Taller

- Nos conectamos a nuestra nueva base de datos
- Creamos una tabla en nuestra base de datos
- Revisamos el contenido de la tabla en pgpool y en cada uno de los backends (nodos)
- Hacemos la misma prueba con una tabla con fechas y horas.

# Ejercicio Realizado en el Taller

```
# \c bd1
```

```
# CREATE TABLE tabla1 (id serial, nombre  
varchar(70), apellido varchar (50));
```

```
# INSERT INTO tabla1 (nombre,apellido)  
VALUES ('Juan','Salao');
```

Ejecutar en cada uno de los nodos

```
# SELECT * FROM tabla1;
```

# Ejercicio Realizado en el Taller

```
# CREATE TABLE prueba_tiempo (id int,  
tiempo timestamp without time zone);
```

```
# INSERT INTO prueba_tiempo values  
(1,now());
```

Ejecutar en cada uno de los nodos

```
# SELECT * FROM prueba_tiempo;
```

# Ejercicio Realizado en el Taller

- Creamos una función
- Ejecutamos la función
- Validamos el resultado
- Ejecutamos la función con el “hint” para que pgpool no realice balance en la ejecución.

# Ejercicio Realizado en el Taller

```
# CREATE or replace function insertar_fechas (i int)
returns void as $$
    declare
        x int;
    begin
        x:=0;
        for x in 1..i loop
            insert into prueba_tiempo values(x,now());
            RAISE NOTICE 'x es %', x;
        end loop;
    end
$$ language plpgsql;
```

# Ejercicio Realizado en el Taller

```
# SELECT insertar_fechas(5);
```

Ver resultados en pgpool

Ver resultados en cada uno de los backends

(efectos: now() toma un valor durante la transacción por ende se repite 5 veces la misma hora, debido a que pgpool balanceó la consulta los resultados fueron enviados a uno solo de los backends)

# Ejercicio Realizado en el Taller

Limpiamos el contenido de la tabla desde la conexión a pgpool

```
# TRUNCATE prueba_tiempo;
```

Ejecutamos la función con el “hint” para pgpool

```
# /* NO LOAD BALANCE */ SELECT  
insertar_fechas(5);
```

# Ejercicio Realizado en el Taller

Ejercicio que queda indicado:

Igual que el anterior pero utilice la función que se le indica a continuación para la tabla:

```
# CREATE TABLE prueba_nombre (nombre  
VARCHAR);
```

# Ejercicio Realizado en el Taller

```
# CREATE or replace function insertar_nombre (i  
varchar) returns void as
```

```
$$
```

```
declare
```

```
begin
```

```
insert into prueba_nombre values(i);
```

```
RAISE NOTICE 'se inserto el nombre %', i;
```

```
end
```

```
$$
```

```
language plpgsql;
```

# FIN (por ahora)



# Créditos

@carlosgr\_arahat

@lennincaro

@leninmhs

@deglip

@gregoria126

@cnti

Fuentes:

documentación pgpool;

documentación proyecto postgresql-r;

wikipedia