



# PALOMINO<sup>o</sup>DB

Proven Database Excellence

## PostgreSQL 9.1

(Presentación para Libre Software World Conference)

Presentado por:

Emanuel Calvo @3manuek

DBA

[www.palominodb.com](http://www.palominodb.com)



Libre  
Software  
world conference



## Sobre el palestrante y la presentación:

- Operational DBA at PalominoDB
- Consultor Principal en Helpame
- DBA Mysql y Oracle en Aedgency.
- Twitter: @3manuek
- Mail: emmanuel@{palominodb.com|helpame.com}
- Esta presentación está basada en el artículo del mismo ponente:
  - <http://palominodb.com/blog/2011/09/28/postgresql-91-part-1-general-features>



Hey! You're da' DB guy!  
\*Fix my PANTS!\*

Shut off and do it rigth now!

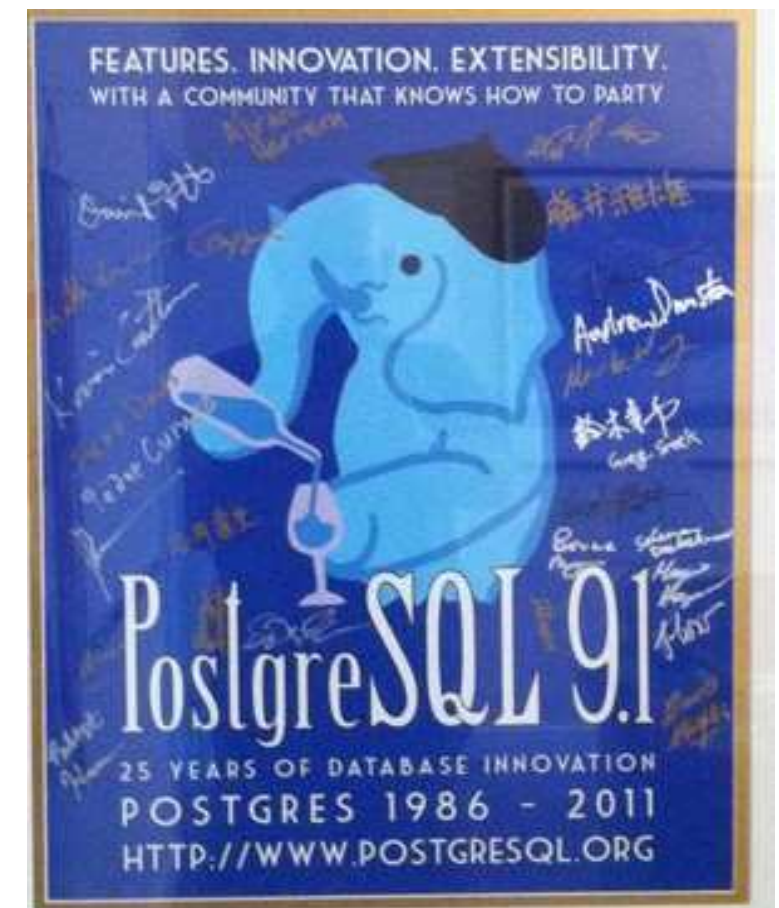


PALOMINO<sup>DB</sup>

Proven Database Excellence

# Nuevas características

- Replicación sincrónica
- Soporte de datos foráneos
- Soporte de internalización por columna
- Instantánea de aislamiento serializable (SSI)
- Tablas efímeras
- Expresiones comunes de tablas con escritura
- Algoritmo de KNN para índices GIST e GIN
- Soporte SE-LINUX
- Actualización de la versión del lenguaje Python
- PGXN
- Otras cosas adicionales...

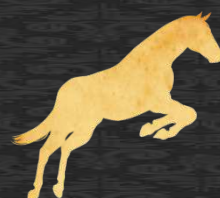


PALOMINO DB

Proven Database Excellence

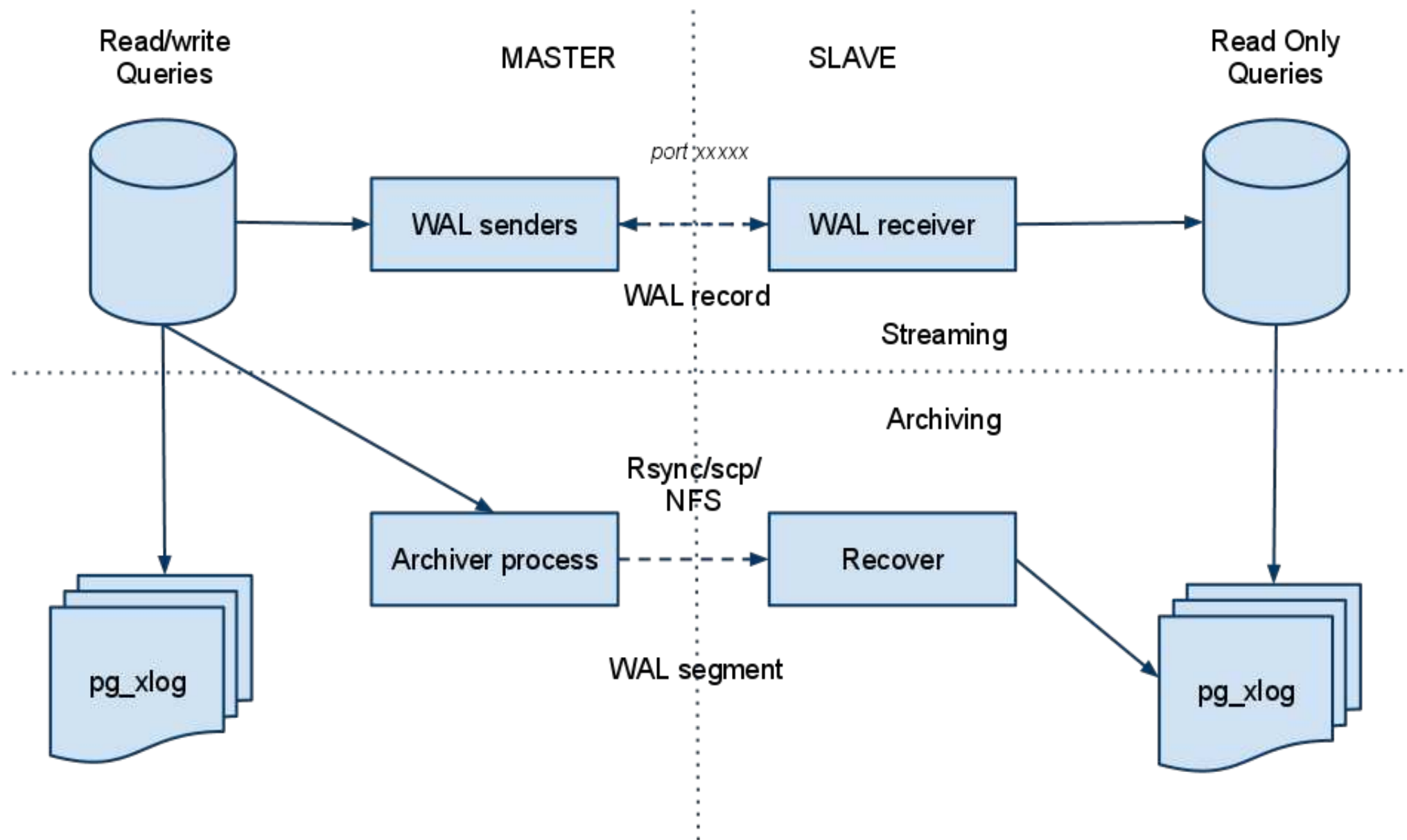
# Replicación Sincrónica

- Control de la sincronización por transacción a través de `synchronous_commit`.
- Mejoras respecto a la versión 9.0.
- Funciones para pausar y mostrar el estado del esclavo
  - `pg_xlog_replay_pause()`
  - `pg_is_xlog_replay_paused()`
  - `pg_xlog_replay_resume()`
  - `pg_last_xact_replay_timestamp()`
  - `Pg_last_xact_replay_location()`
- `pg_ctl` incluye “promote” para el esclavo
- Clonación de esclavo sin downtime (`pg_basebackup`)
- Mayor información del estado de los esclavos (`pg_stat_replication`).
- [`pg\_stat\_database\_conflicts`](#) para saber el estado de consultas canceladas.

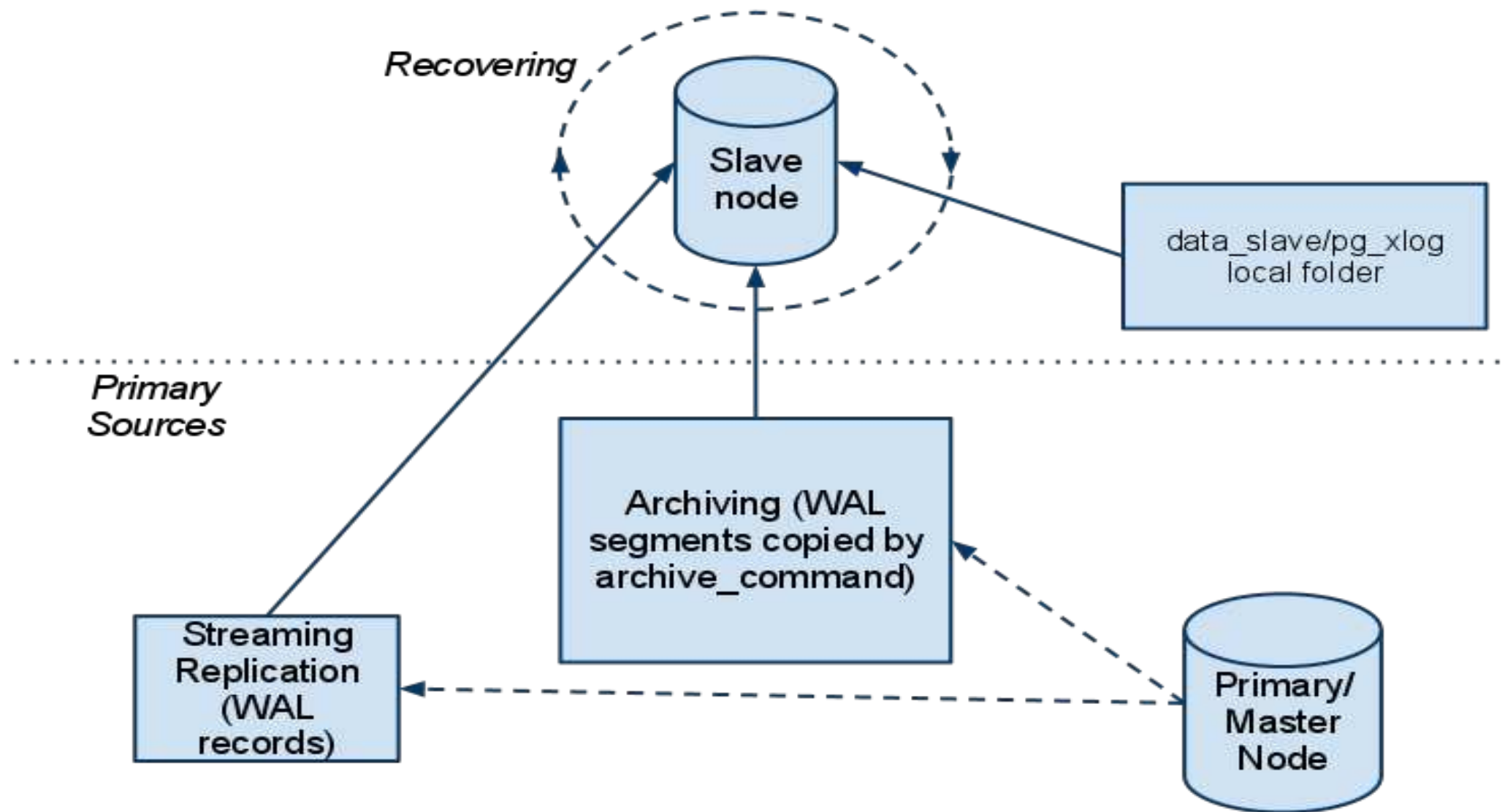




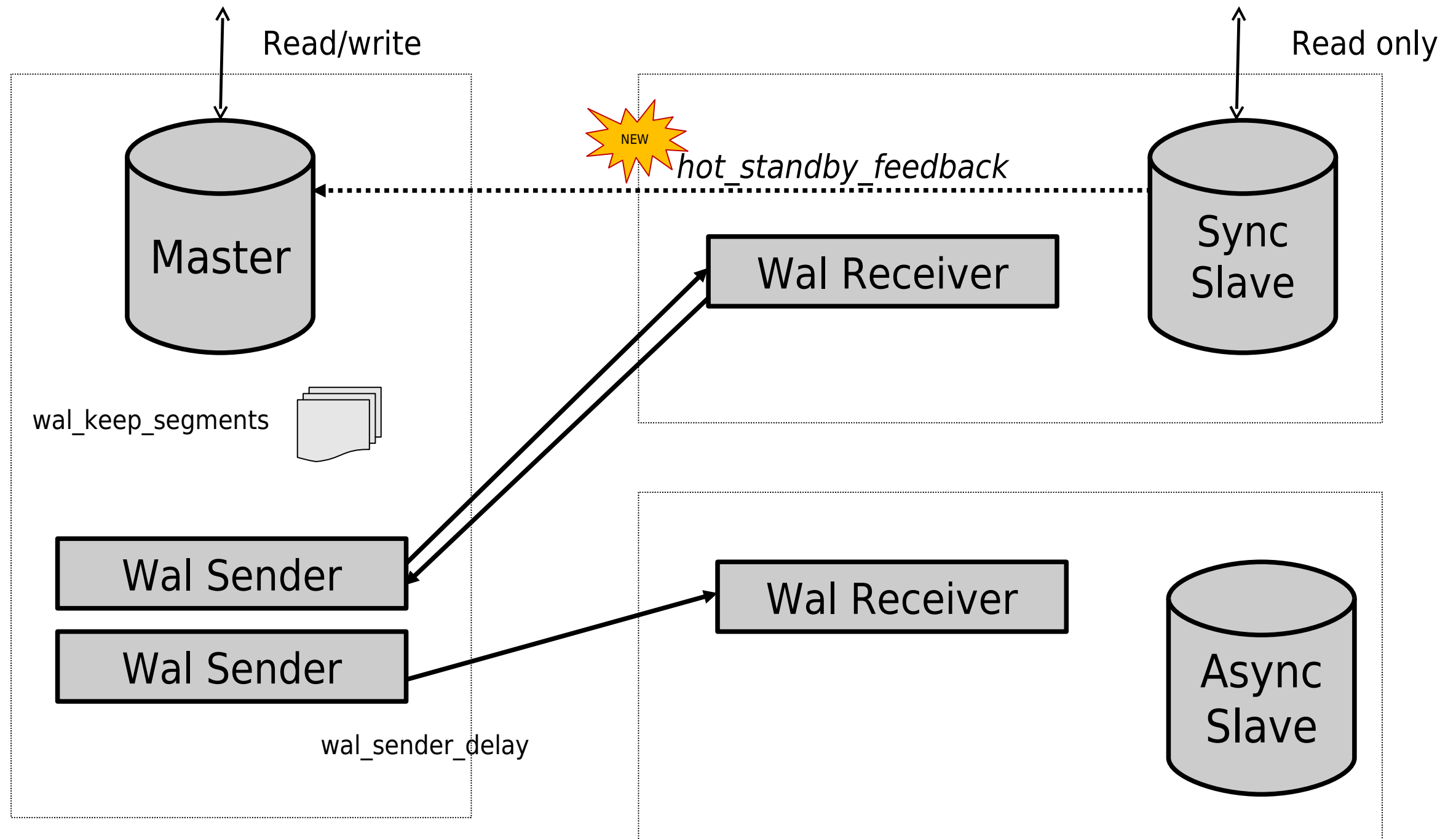
# Replicación



# Fuentes de los servidores Standby



# Replicación Sincrónica (una vista más detallada)



# Limitaciones en la replicación

- Solo un servidor puede ser sincrónico.
- No soporta multi-master.
- No soporta replicación parcial.
- No soporta distribución de datos.





# Soporte datos foráneos (MED – Management of External Data)

```
CREATE EXTENSION file_fdw;  
  
CREATE SERVER pglog FOREIGN DATA WRAPPER file_fdw;  
  
CREATE FOREIGN TABLE gaiyin (  
    nombre text,  
    numero integer,  
    letra char(1)  
) SERVER pglog  
OPTIONS ( filename 'c:\temp\file.csv', format 'csv' );
```

```
postgres=# explain select * from gaiyin;  
          QUERY PLAN
```

```
-----  
Foreign Scan on gaiyin (cost=0.00..1.10 rows=1 width=44)  
  Foreign File: c:\temp\file.csv  
  Foreign File Size: 36  
(3 filas)
```

<http://wiki.postgresql.org/wiki/SQL/MED>



PALOMINO<sup>DB</sup>

Proven Database Excellence

# Soporte Datos foráneos - mysql\_fdw

apt-get install libmysqld-dev

\* Compilar como un contrib (make; make install)

CREATE EXTENSION mysql\_fdw;

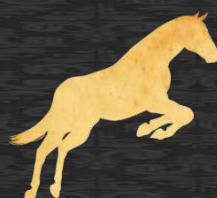
CREATE SERVER mysql\_srv  
FOREIGN DATA WRAPPER mysql\_fdw  
OPTIONS (address 'localhost', port '3306');

CREATE USER MAPPING  
FOR PUBLIC SERVER mysql\_srv  
OPTIONS (username 'root', password 'postgres');

CREATE FOREIGN TABLE prueba\_my  
(texto text, numero int, floaty float)  
SERVER mysql\_srv OPTIONS (table 'palomino.prueba');

CREATE FOREIGN TABLE prueba\_my (  
id integer,  
name text,  
address text)  
SERVER mysql\_srv  
OPTIONS (table 'palomino.prueba');

Fuente: <http://pgsnake.blogspot.com/2011/08/postgresql-91-meet-mysql.html>



PALOMINO<sup>DB</sup>

Proven Database Excellence

# Soporte datos foráneos (conexión a otras bases)

Oracle  
Mysql  
OBDC  
Redis  
CouchDB  
Filetext  
Twitter  
LDAP  
...

[http://wiki.postgresql.org/wiki/Foreign\\_data\\_wrappers](http://wiki.postgresql.org/wiki/Foreign_data_wrappers)  
<http://repositorio.siu.edu.ar/trac/postgresql/wiki/Dblink>



PALOMINO<sup>DB</sup>

Proven Database Excellence

# Soporte de internalización por columna

```
CREATE TABLE prueba_texto (  
  _id serial PRIMARY KEY,  
  texto_en text COLLATE 'en_US',  
  texto_es text COLLATE 'es_ES'  
);
```

```
SELECT  
  texto_es < ('rendevouz' COLLATE "es_ES")  
FROM prueba_texto;
```

```
SELECT  
  texto_es COLLATE "en_EN" < texto_en  
FROM prueba_texto;
```

```
CREATE INDEX ON prueba  
  (texto COLLATE "en_US);
```

- < 8.4 el COLLATE estaba ligado al sistema operativo, siendo muy poco flexible.
- En esta versión podemos tener incluso columnas con ordenamientos específicos.
- Se pueden crear índices utilizando un ordenamiento específico.
  - En este caso tener cuidado, porque el utilizar otro COLLATE hará que no utilice el índice.





# SSI

- Serializable Snapshot Isolation (traducirlo? ... ¬¬)
  - Aislamiento de Instantánea serializable.
- Postgres ahora soporta 3 niveles transaccionales:
  - READ COMMITTED
  - REPETEABLE READ (viejo SERIALIZABLE)
    - Es lo que sería SNAPSHOT ISOLATION
  - SERIALIZABLE
- No es necesario “select for update”
- Simplificación de la programación
- Tiene coste, utiliza bloqueo predicativo, incrementa el número de rollbacks debido a conflictos.

<http://wiki.postgresql.org/wiki/SSI>

<http://wiki.postgresql.org/images/4/4f/SSI-PGConfEU2011.pdf>



PALOMINO<sup>DB</sup>

Proven Database Excellence

# Tablas efímeras

- Llamadas UNLOGGED TABLES.
- Estas tablas no pueden recuperarse ya que no figuran en la WAL.
- Minimizan el impacto de escritura en disco.
- No son replicadas mediante streaming.
- Truncadas en caso de crash.
- Casos especiales sobre contención de CLOG

<http://rhaas.blogspot.com/2011/11/hint-bits.html>



PALOMINO<sup>DB</sup>

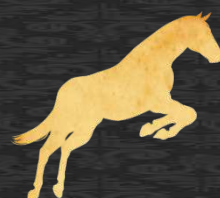
Proven Database Excellence

# Expresiones comunes de tablas con escritura

- WCTE
- CTE fue implementado desde 8.4
- Información avanzada: <http://wiki.postgresql.org/images/d/da/Ciolfi-debug-wcte-2011.pdf>

```
WITH table_temp(i,j,k) AS  
(  
    DELETE FROM prueba WHERE i =9 RETURNING *  
)  
INSERT INTO prueba_del  
SELECT i,j,k  
FROM table_temp;
```

```
WITH RECURSIVE t(a,b) AS (  
    VALUES(0,1)  
    UNION ALL  
    SELECT greatest(a,b), a + b AS a FROM t  
    WHERE b < $1  
)  
SELECT a FROM t;
```



# KNN para GiST y GIN

- Búsqueda de valores cercanos utilizando índices.
  - Facilita la búsqueda de puntos cercanos.
- Operador “<->”

Artículo ejemplo:

<http://www.palominodb.com/blog/2011/10/12/indexing-text-columns-gist-or-gin-optimize-ilike-using-pgtrgm-postgres-91-part-1>

<http://wiki.postgresql.org/images/4/46/Knn.pdf>



PALOMINO<sup>DB</sup>

Proven Database Excellence



# Soporte SE-LINUX

- Integración con SE-LINUX.
- Tienen prioridad los permisos otorgados desde el SO en vez de los de la Base de Datos.
- Este módulo debe ser activado explícitamente.
- Se basa en la creación de SECURITY LABELS
- NOTA: Se requiere que las librerías de SELINUX del kernel sean mayores a las 2.0.93. Estas librerías vienen por defecto en los kernels mayores a 2.0.33

[http://wiki.postgresql.org/wiki/SEPostgreSQL\\_Documentation](http://wiki.postgresql.org/wiki/SEPostgreSQL_Documentation)



PALOMINO<sup>DB</sup>

Proven Database Excellence

## Características menores (ALTER ENUM)

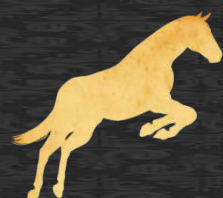
- `ALTER TYPE tipo_enum ADD VALUE new_value [{BEFORE|AFTER} existing_value];`  
`CREATE TYPE lista AS ENUM ('tipoa','tipob','tipoc');`  
`ALTER TYPE lista ADD VALUE 'tipoa2' AFTER 'tipoa';`



# PGXN

<https://github.com/pgxn/pgxn-client>

<http://pgxn.org/faq/>

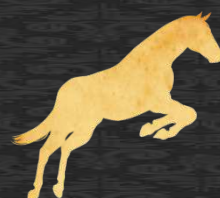


PALOMINO<sup>DB</sup>

Proven Database Excellence

# Cambios varios

- Casteos para tipos compuestos como al estilo función/atributo fueron dehabilitados.
  - *valor\_compuesto.text*, *text(valor\_compuesto)*
- CREATE EXTENSION
- Puesta a punto automática de *wal\_buffers*
- Mejoras en índices GiST y GIN
- Evento INSTEAD OF para disparadores (útil para creación de vistas actualizables)
- Soporte de compilación para Windows 64 bits
- *standard\_conforming\_strings* ahora está establecido por defecto en 'on'
  - Antes 'O'Connor' , ahora 'O'Connor' o E'O'Connor'
- Cambios en PL/SQL (RAISE sin parámetros)
- Performance mejorada para tablas heredadas (Merge Append node – Ordenamiento de tablas heredadas dentro del append en vez de fuera).
- Algoritmo Hash para FULL OUTER JOIN.
- Información detallada: <http://www.postgresql.org/docs/9.1/static/release-9-1.html>





## Cambios varios (II)

- Permisos de replicación (no hace falta que el usuario sea SUPERUSER)
- Mejoras en la seguridad de conexión entre servidores replicados para evitar ataques internos.
- Dependencias funcionales de una PK reconocidas por el GROUP BY.
  - Antes: ERROR: column "tabla.col\_pk" must appear in the GROUP BY clause or be used in an aggregate function.



## Novedades varias

- PLV8 ( JavaScript V8 procedural language)
- Postgres-XC <http://postgres-xc.sourceforge.net/>
- PgPool II 3.1
- Nuevas versiones de londiste, plproxy, bucardo e/otros.



# PostgreSQL 9.2

Características integradas / a integrar:

- \* Index only scans
- \* Slave replication cascade
- \* Fix para las unlogged tables en concurrencia masiva
- \* ...

[http://wiki.postgresql.org/wiki/PostgreSQL\\_9.2\\_Development\\_Plan](http://wiki.postgresql.org/wiki/PostgreSQL_9.2_Development_Plan)

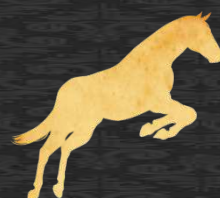
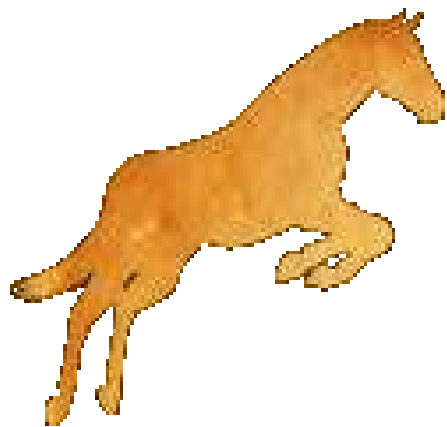
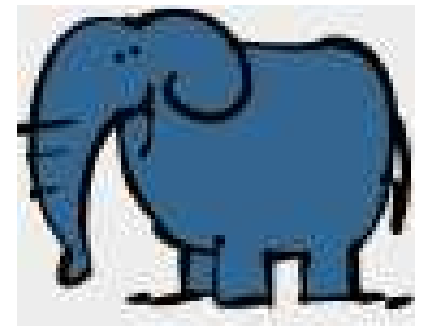


PALOMINO<sup>DB</sup>

Proven Database Excellence

Gracias por asistir!!

Presentación hecha en Libre/Open Office







GO...  
Save yourself...  
I'll keep them...

kill your time on [9GAG.COM](http://9GAG.COM)



PALOMINO<sup>DB</sup>

Proven Database Excellence