

PostgreSQL HA

No hay bala de plata



PostgreSQL HA

- Diogo Biazus
 - Colaborador
 - Instructor
 - Empresario



PostgreSQL HA

- Los tres mandamientos de la alta disponibilidad:
 - Redundancia
 - Redundancia
 - Redundancia

PostgreSQL HA

- Objectivos
 - Aumentar el MTBF
 - Diminuir el tiempo de indisponibilidad

PostgreSQL HA

- ¿Como hacer?
 - Redundancia del hardware
 - Redundancia de base de datos

PostgreSQL HA

- Redundancia del hardware
 - Gran Rendimiento
 - Fácil de aplicar
 - Menos flexible

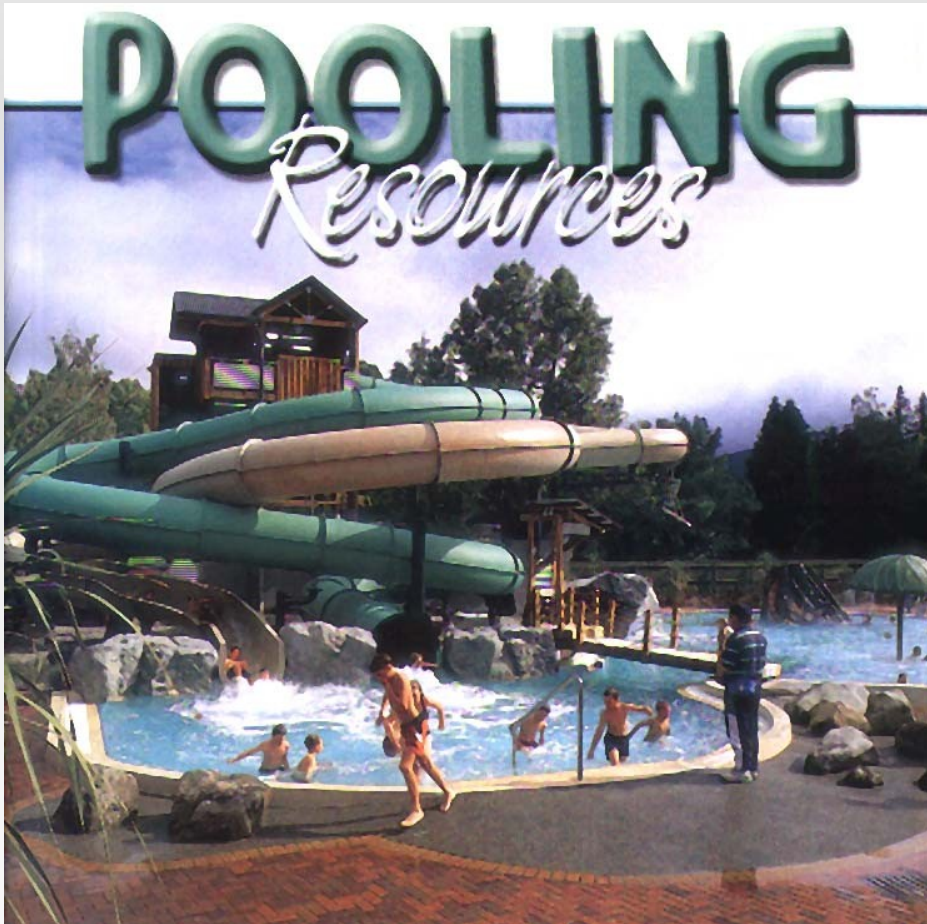
PostgreSQL HA

- Receta de redundancia del hardware
 - Ingredientes:
 - Un storage
 - Dos o más servidores como le gusta
 - Heartbeat
 - Modo de preparo:
 - Deje un servidor ativo
 - En caso de fallo otro servidor puede assumir usando la misma unidad del storage

PostgreSQL HA

- Redundancia de la base de datos
 - Complejidad de la aplicación varía de "no tan simple" a "¡necesidad de editar XMLs mano".
 - Es generalmente más flexible que la redundancia de hardware

PostgreSQL HA



- Pooling
 - "Man in the middle"
 - Pgpool
 - Sequoia

PostgreSQL HA

- Pgpool
 - Trabaja con PostgreSQL, código C
 - Fácil instalación y configuración
 - Código estable
 - Flexible, pero necesitamos otras herramientas
- Sequoia
 - Trabaja con cualquier base de datos, código java
 - Configuración de poco más complicado, pero bien documentado
 - Para los clientes JDBC, pero la solución es completa

PostgreSQL HA

- Problemas do pooling
 - Uso de campos seriales
 - Sincronización de tiempo
 - Una pieza más a tener problemas

PostgreSQL HA



- Replicacion multi-master
 - PgCluster
 - Cybercluster
 - Bucardo

PostgreSQL HA

- PgCluster / Cybercluster
 - Sincrónico con balance de carga
 - Lento
 - Requiere versión modificada del servidor / inestable
 - Soporta DDL, es bastante completa
- Bucardo
 - Asíncrono sin balance de carga
 - No es compatible con claves compuestas o DDL
 - Programable a través de Perl
 - Daemom funcionando en paralelo (trigger based)

PostgreSQL HA



- Replicacion master-slave
 - Slony - I

PostgreSQL HA

- Slony – I
 - Asíncrono sin balance de carga
 - Daemon funcionando en paralelo (trigger based)
 - No se elegirá a un nuevo master automáticamente

PostgreSQL HA

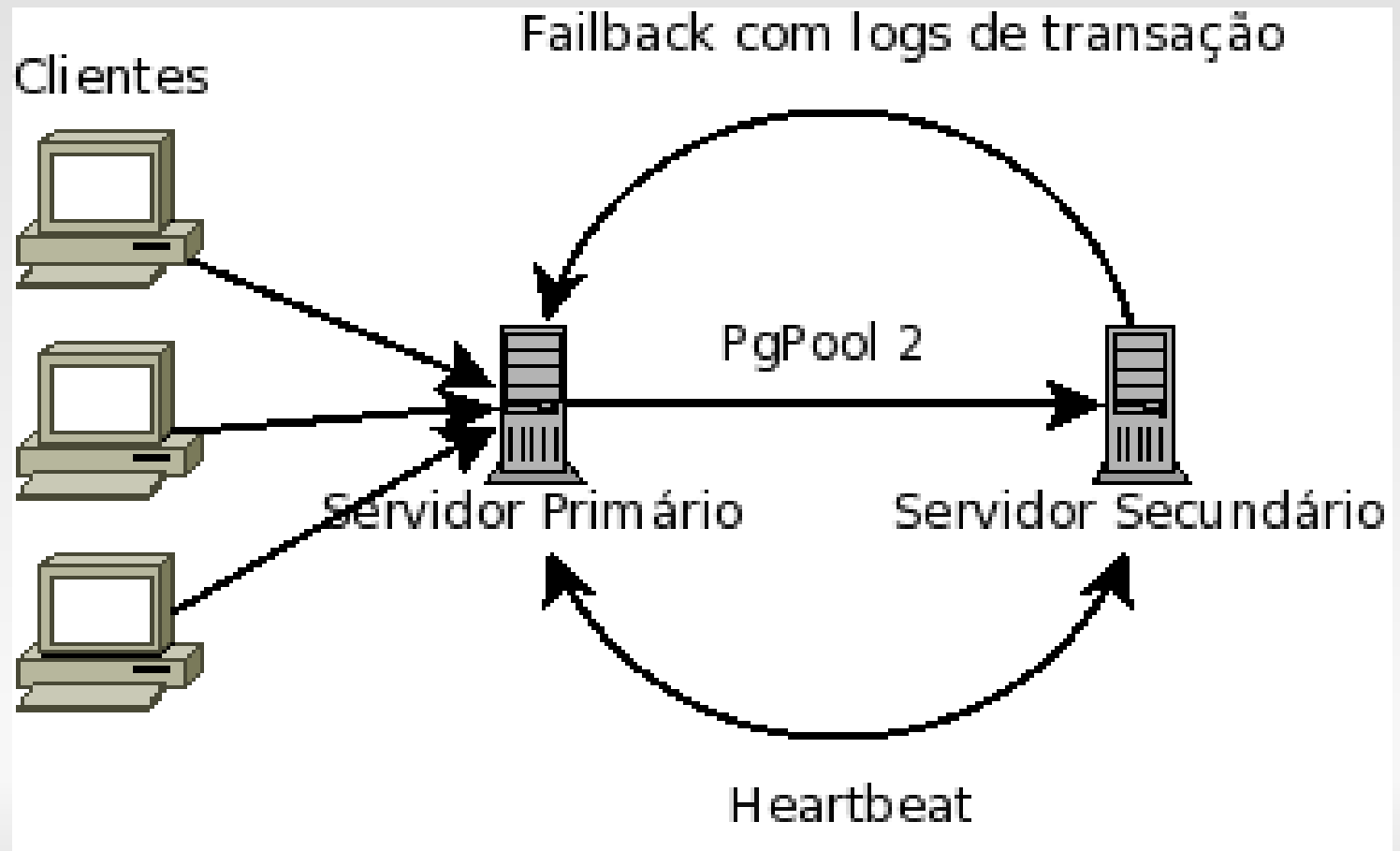
- ¿Que elegir?
 - PgCluster: No necesito mucho rendimiento, pero necesito sincronismo (LAN)
 - Bucardo: No necesito de sincronismo y no tengo clave compuesta (LAN o WAN)

PostgreSQL HA

- Pgpool: Muy flexible, necesito rendimiento y no voy a alterar mi aplicativo.
- Sequoia: Quiero una solución completa e puedo utilizar JDBC
- Slony – I: Muy utilizado en conjunto con el Pgpool. Robusto y veloz. Simplifica e mejora el failback.

PostgreSQL HA

- Estudo de caso
- PGPool + Heartbet
- 60 GB
- OLTP



PostgreSQL HA

- PGPool - pgpool.conf
 - listen_addresses = '*'
 - port = 5432
 - pcp_port = 9898
 - replication_mode = true
 - backend_hostname0 = 'localhost'
 - backend_port0 = 9999
 - backend_weight0 = 1

PostgreSQL HA

- Heartbet - ha.cf

bcast eth0

auto_failback off

node host1

- authkeys

auth 1

1 sha1 teste

- Haresources

dbserver-slave.buffon.com.br IPaddr::192.168.0.31/24

PostgreSQL HA

- Failover
 - Automatico
- Failback
 - WAL PITR

PostgreSQL HA

Gracias
diogob@gmail.com