



新手机 新应用 新娱乐

# PostgreSQL Enterprise Appliance

Digoal.Zhou

7/6/2011

# Database LifeCycle



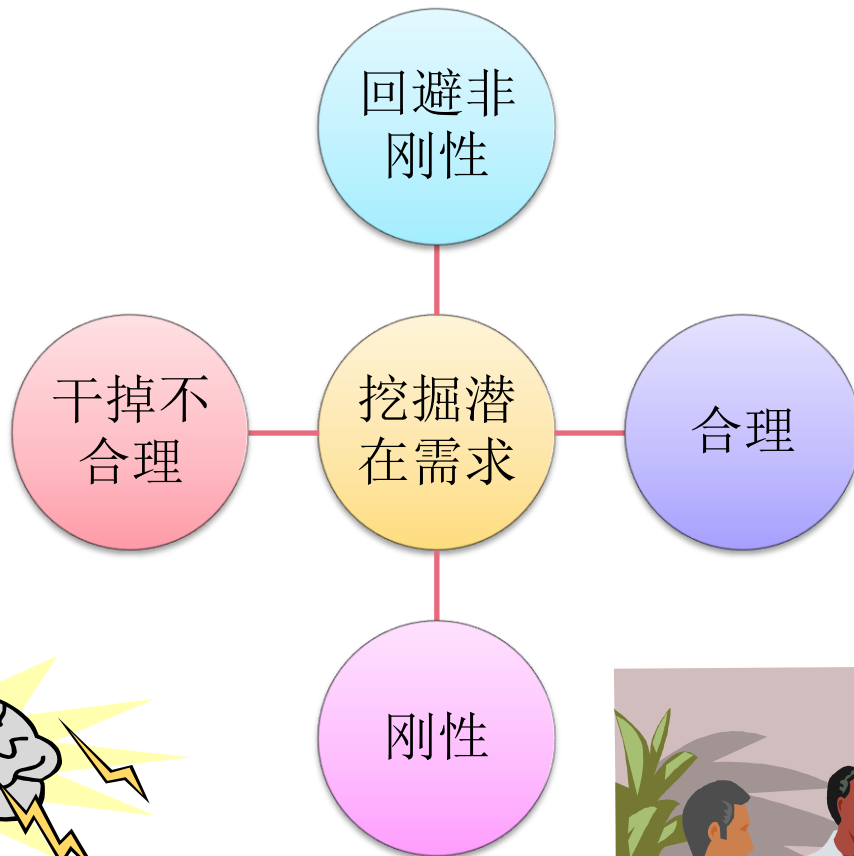
DBA



# 用户就是上帝



# 需求阶段



头脑风暴



需求评审



## ■ 业务类型列举浅析

### ■ APPStore型(静态)

- 资源基础信息
- 资源排行榜
- 资源打分,评论等
- 包月类信息
- 活跃数据<内存大小

### ■ SNS型(动态)

- 个人属性
- 社会关系(好友, 群组)
- 分享(微博, 即时消息, 图片, 视频, 音乐, 博文)
- 磁性应用(社区网游)
- 活跃数据>内存大小

### ■ 网游型(动态)

- 个人属性(经验值,等级,装备,金钱...)
- 关系(好友, 群组)

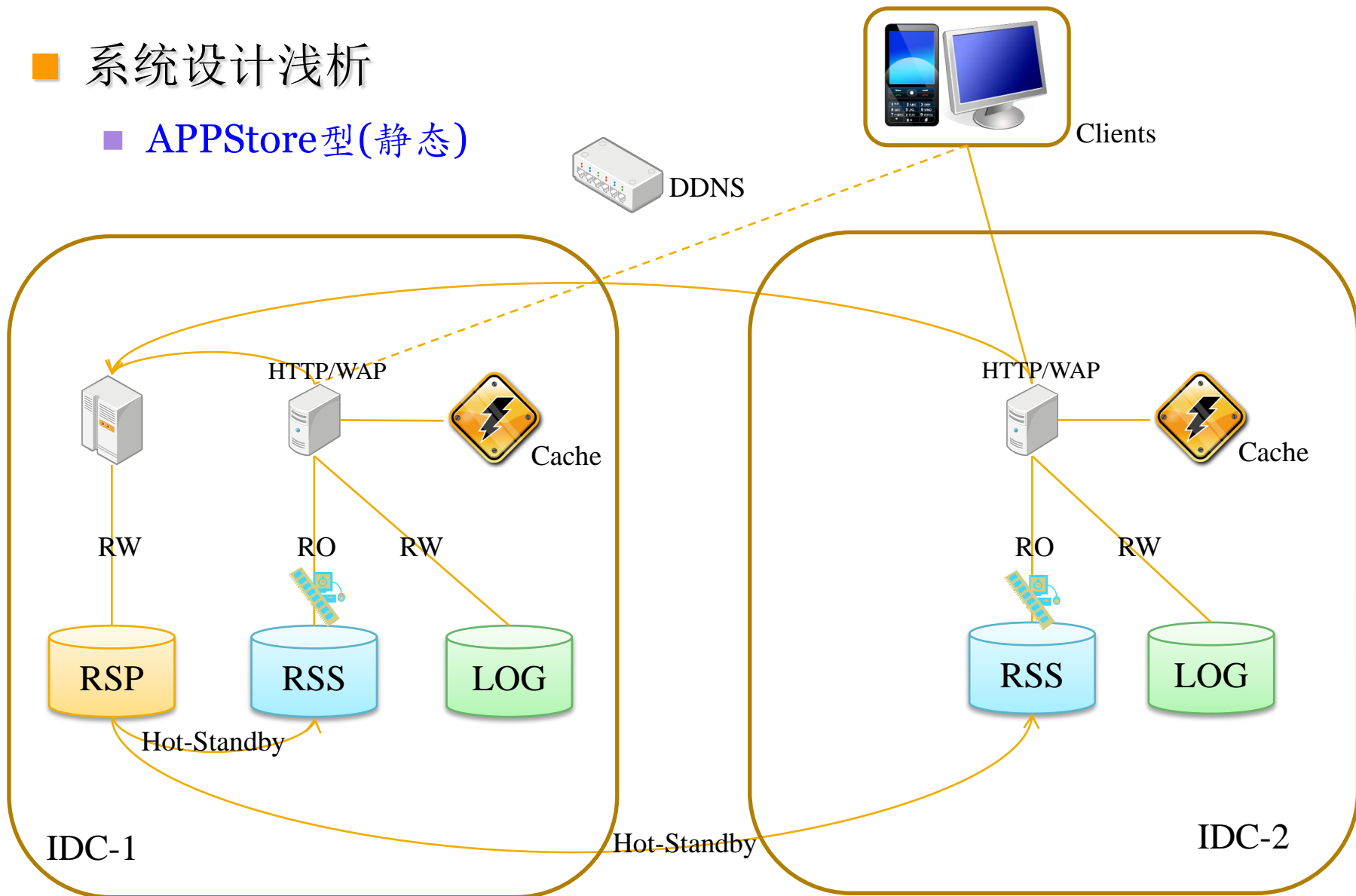
## ■ 数据库挑战浅析

### ■ APPStore型(静态)

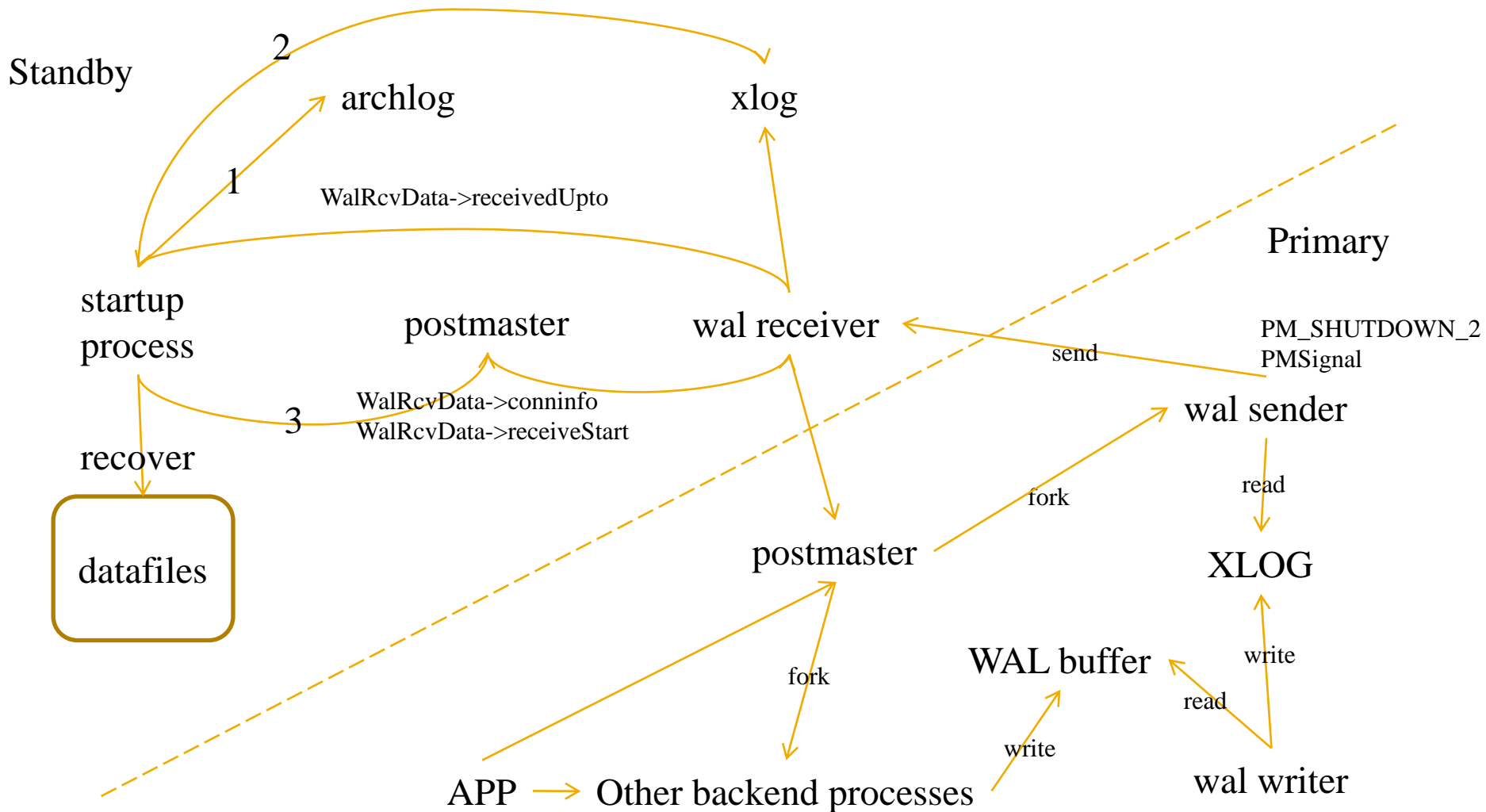
- 读请求(70%)
  - 白名单, 黑名单, 资源信息, 包月信息等
- 写请求(30%)
  - 资源信息增删改, 新增日志
- 请求数到达数据库极限
- 应用层缓存刷新或失效
- 硬件故障
- IDC故障
- 用户体验

## ■ 系统设计浅析

### ■ APPStore型(静态)



## 系统设计浅析 – 异步流复制介绍





## Parameter Tuning :

### Primary

max\_wal\_senders

wal\_sender\_delay ( The sleep is interrupted by transaction commit )

wal\_keep\_segments

**vacuum\_defer\_cleanup\_age ( the number of transactions by which VACUUM and HOT updates will defer cleanup of dead row versions. )**

### Standby

hot\_standby

# wal apply & SQL on standby conflict reference parameter

max\_standby\_archive\_delay

( the maximum total time allowed to apply any one WAL segment's data. )

max\_standby\_streaming\_delay

( the maximum total time allowed to apply WAL data once it has been received from the primary server )

wal\_receiver\_status\_interval

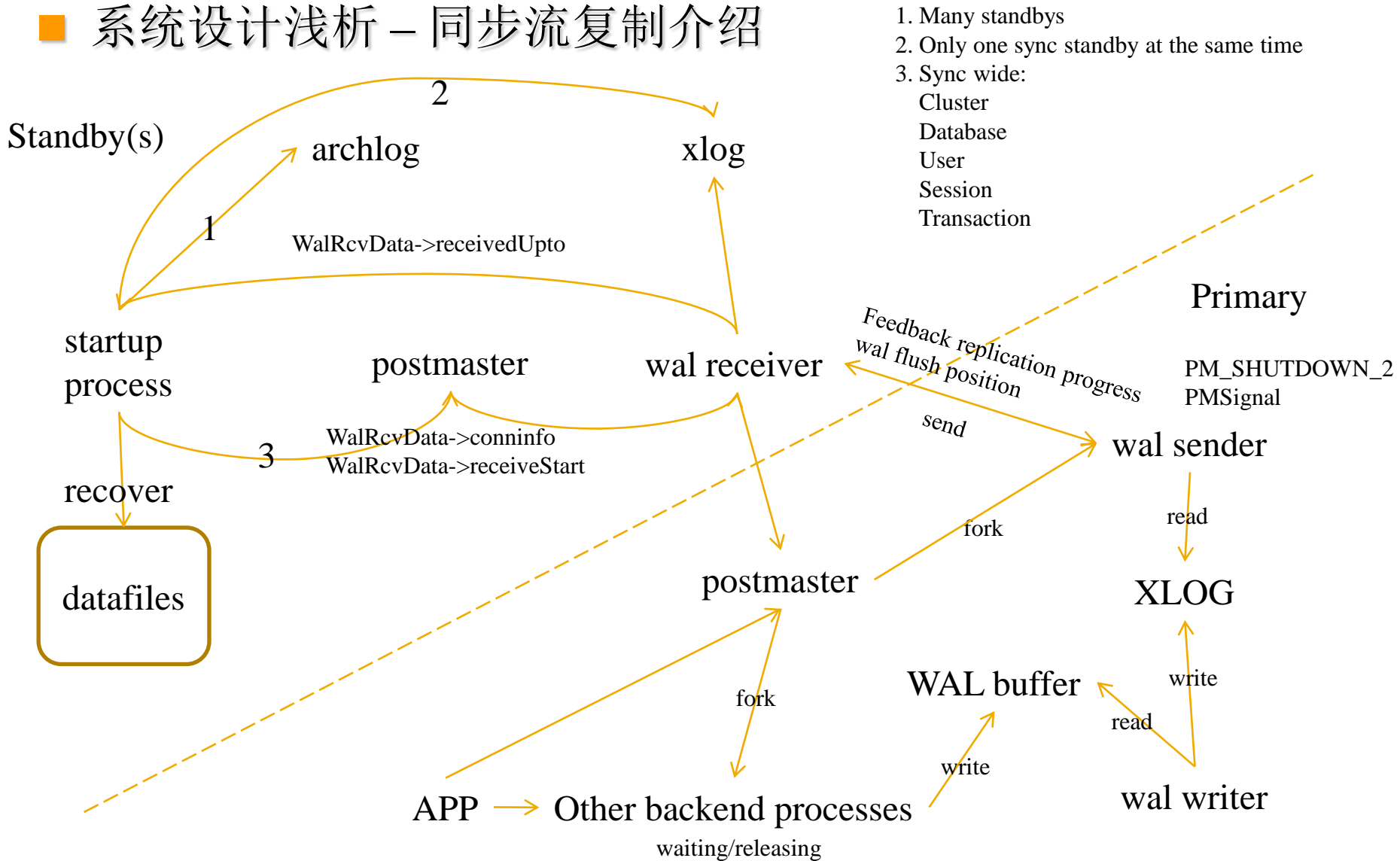
( minimum frequency, The standby will report the last transaction log position it has written, the last position it has flushed to disk, and the last position it has applied.)

**hot\_standby\_feedback**

(send feedback to the primary about queries currently executing on the standby. )

# 设计阶段

## 系统设计浅析 – 同步流复制介绍



Parameter Tuning :

Primary

max\_wal\_senders

wal\_sender\_delay

wal\_keep\_segments

vacuum\_defer\_cleanup\_age

synchronous\_replication

synchronous\_standby\_names

( primary\_conninfo in standby's primary\_conninfo )

Standby

hot\_standby

max\_standby\_archive\_delay

max\_standby\_streaming\_delay

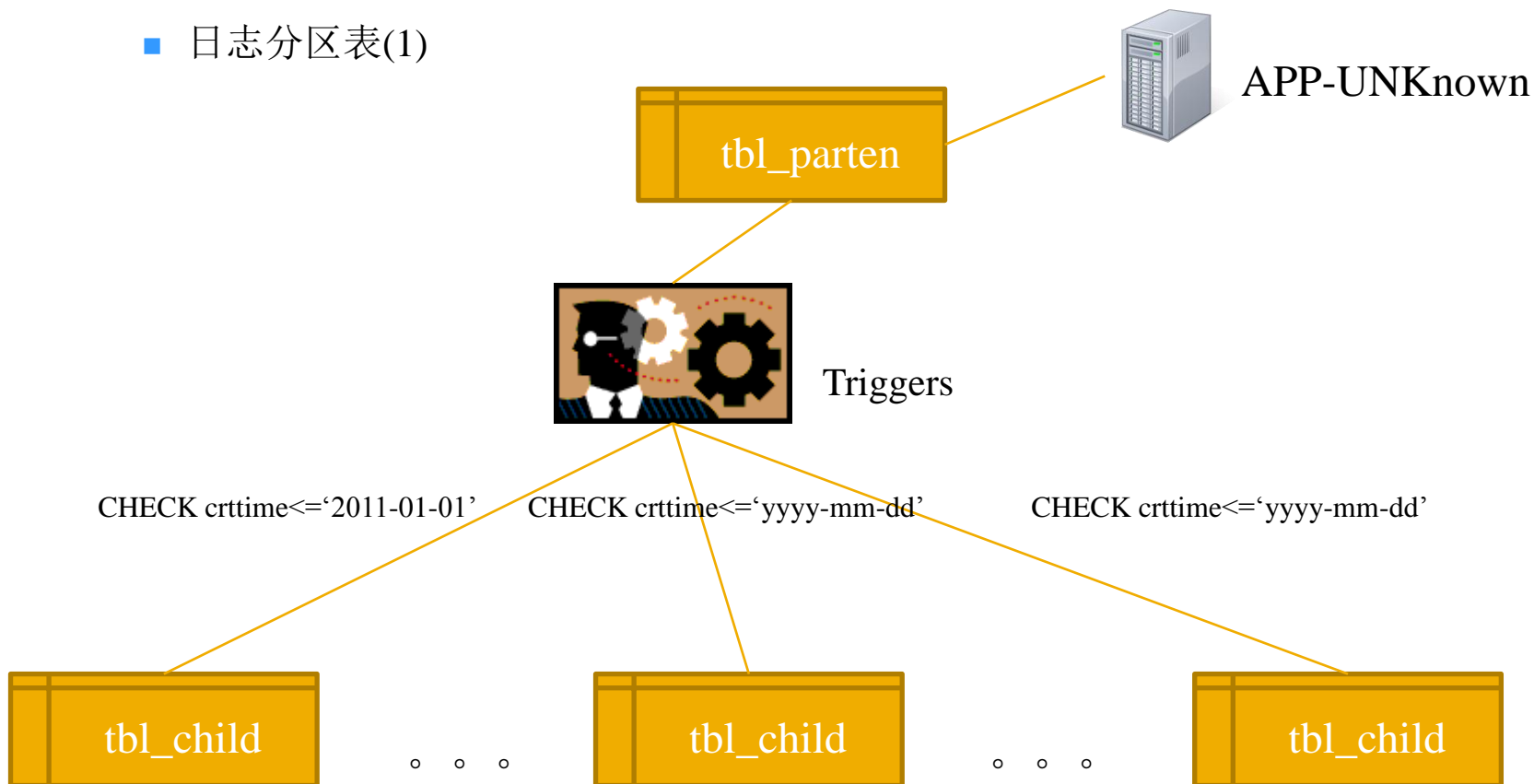
wal\_receiver\_status\_interval

hot\_standby\_feedback

## ■ 系统设计浅析

### ■ APPStore型(静态)

#### ■ 日志分区表(1)

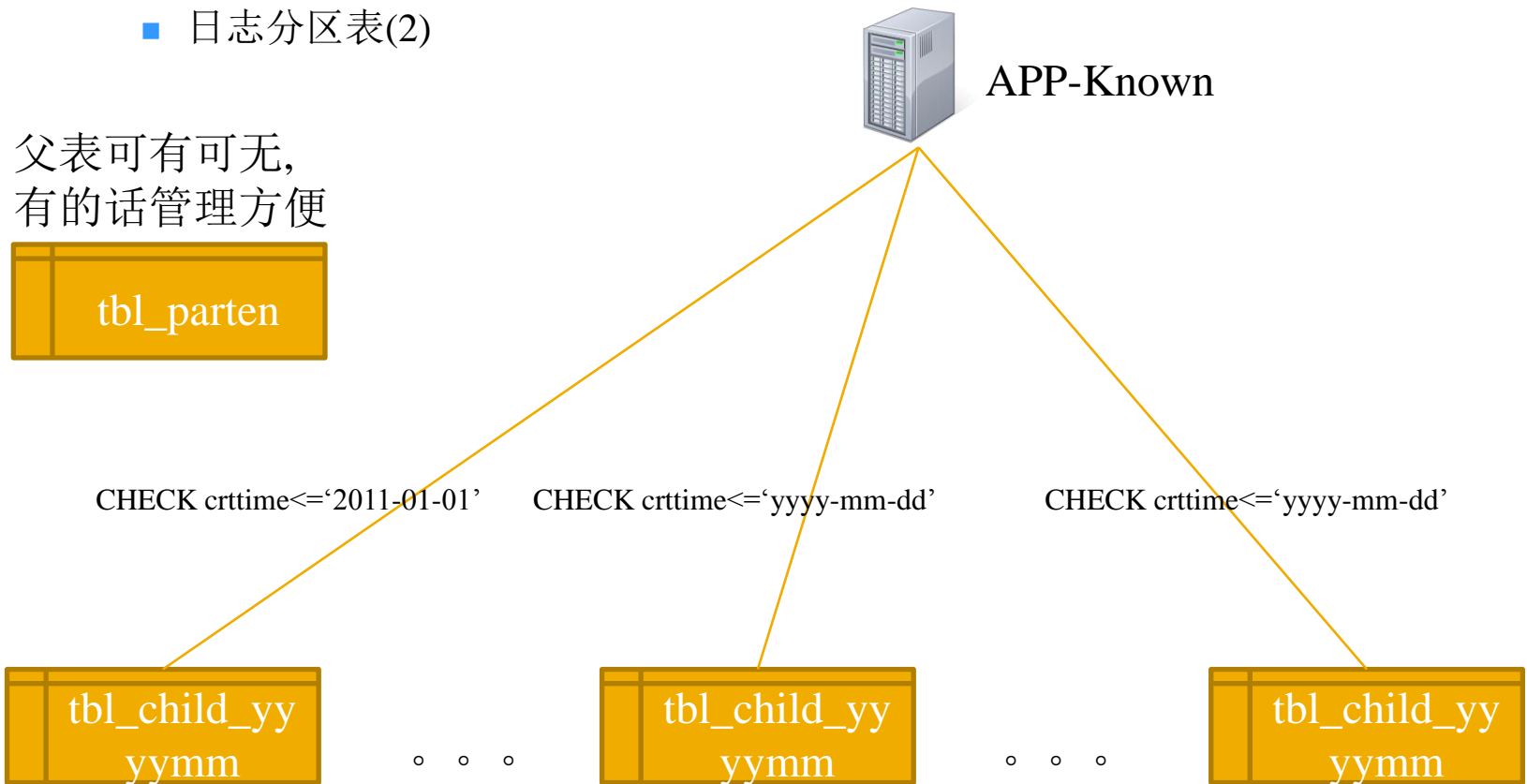


## ■ 系统设计浅析

### ■ APPStore型(静态)

#### ■ 日志分区表(2)

父表可有可无,  
有的话管理方便



## ■ 数据库挑战浅析

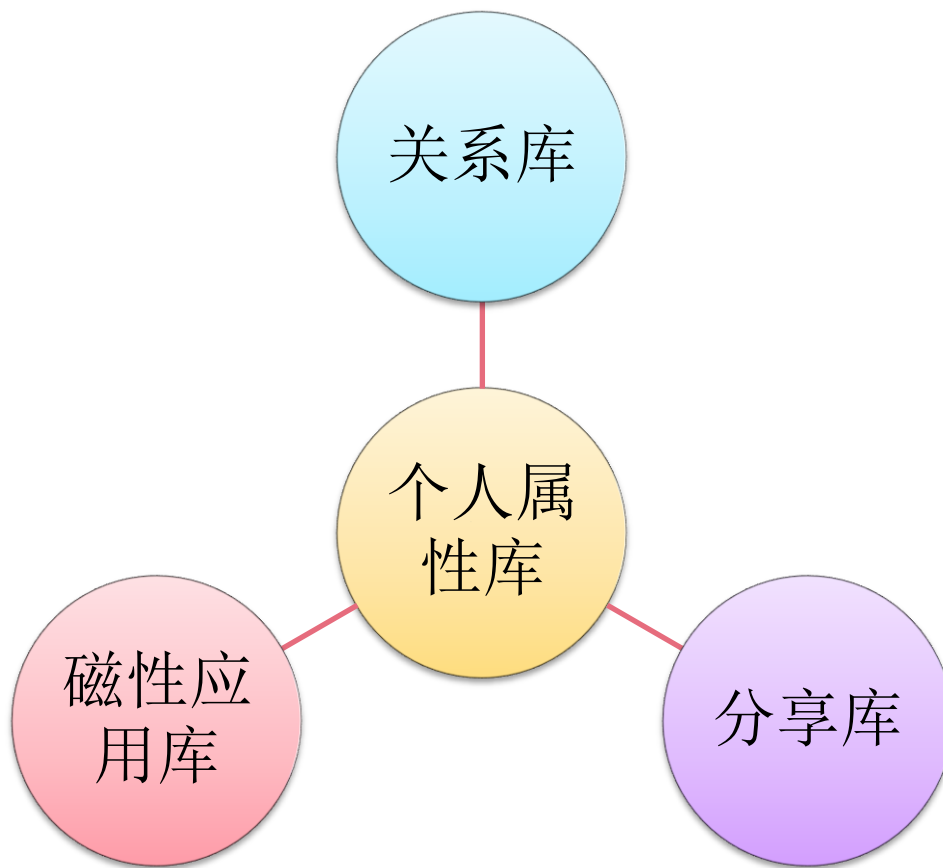
### ■ SNS型(动态)

- 写请求(50%)
- 读请求(50%)
- 个人属性数据>内存大小
- 社会关系数据>内存大小
- 分享数据>内存大小
- 磁性应用数据>内存大小
- 活跃数据>内存大小

## ■ 系统设计浅析

### ■ SNS型(动态)

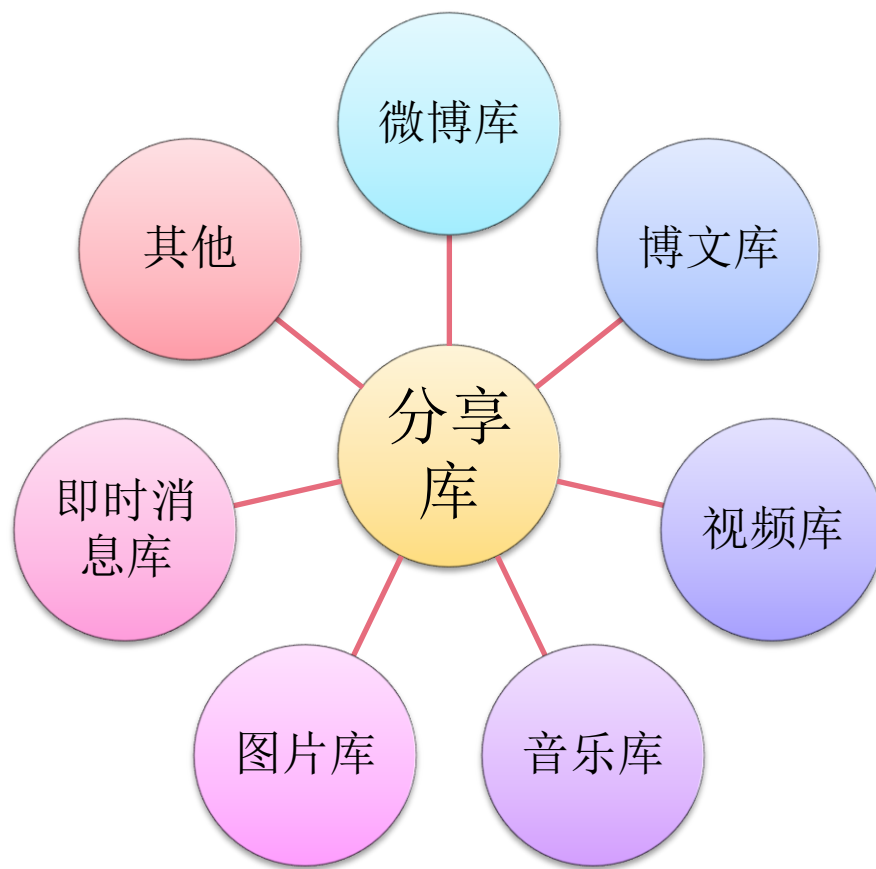
#### ■ 数据库功能性拆分



## ■ 系统设计浅析

### ■ SNS型(动态)

#### ■ 数据库功能性拆分

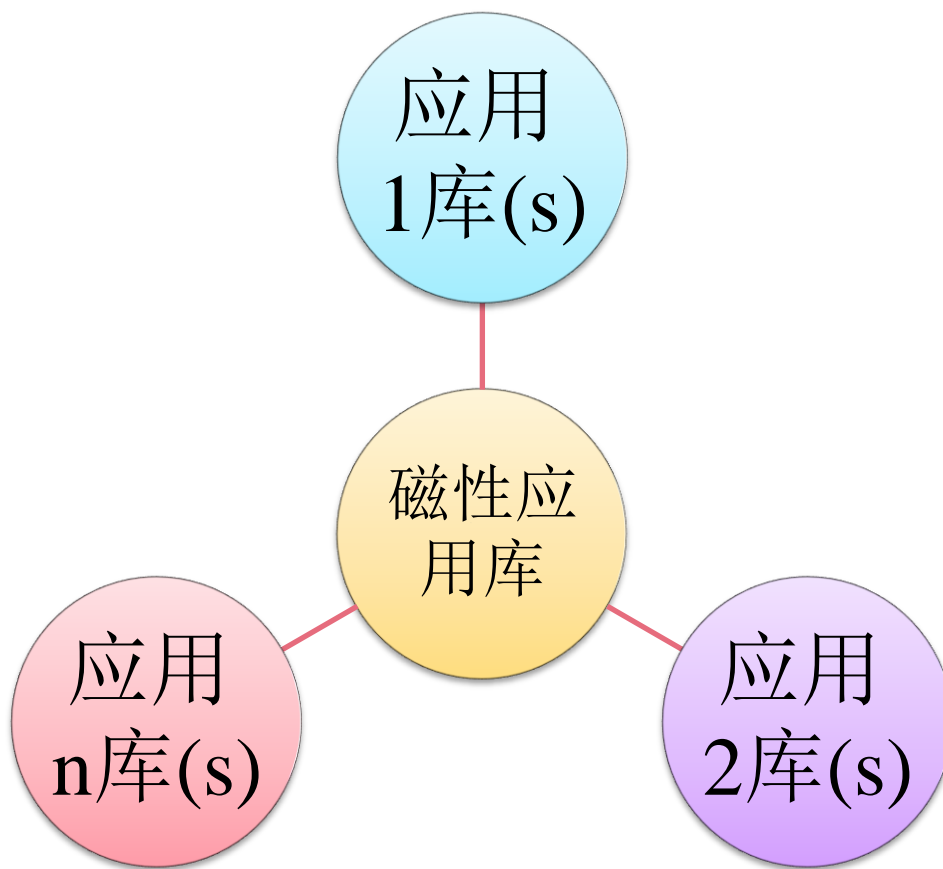




## ■ 系统设计浅析

### ■ SNS型(动态)

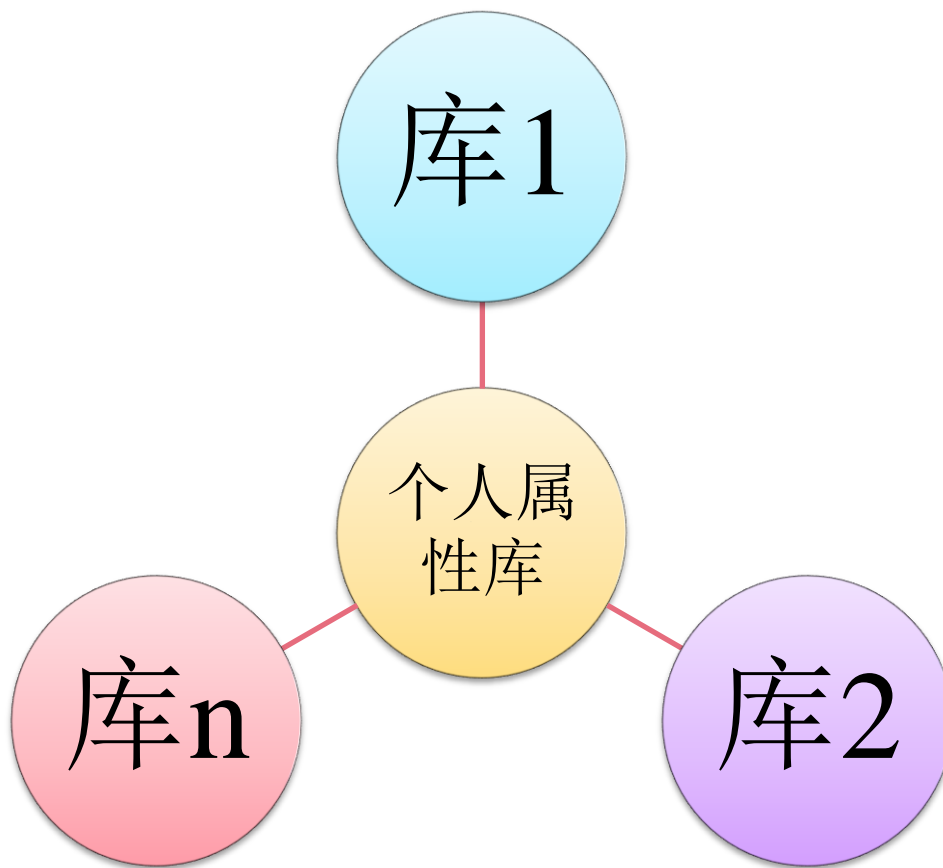
- 数据库功能性拆分
- 一般游戏库还可分区域



## ■ 系统设计浅析

### ■ SNS型(动态)

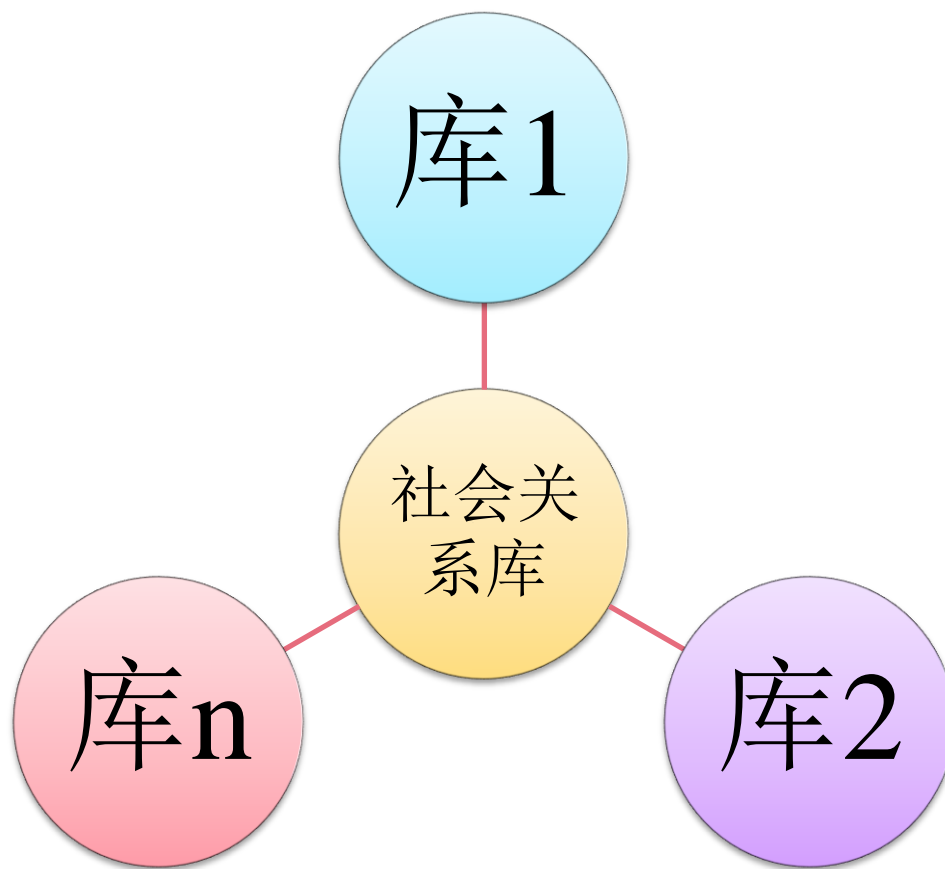
#### ■ 数据库水平拆分



## ■ 系统设计浅析

### ■ SNS型(动态)

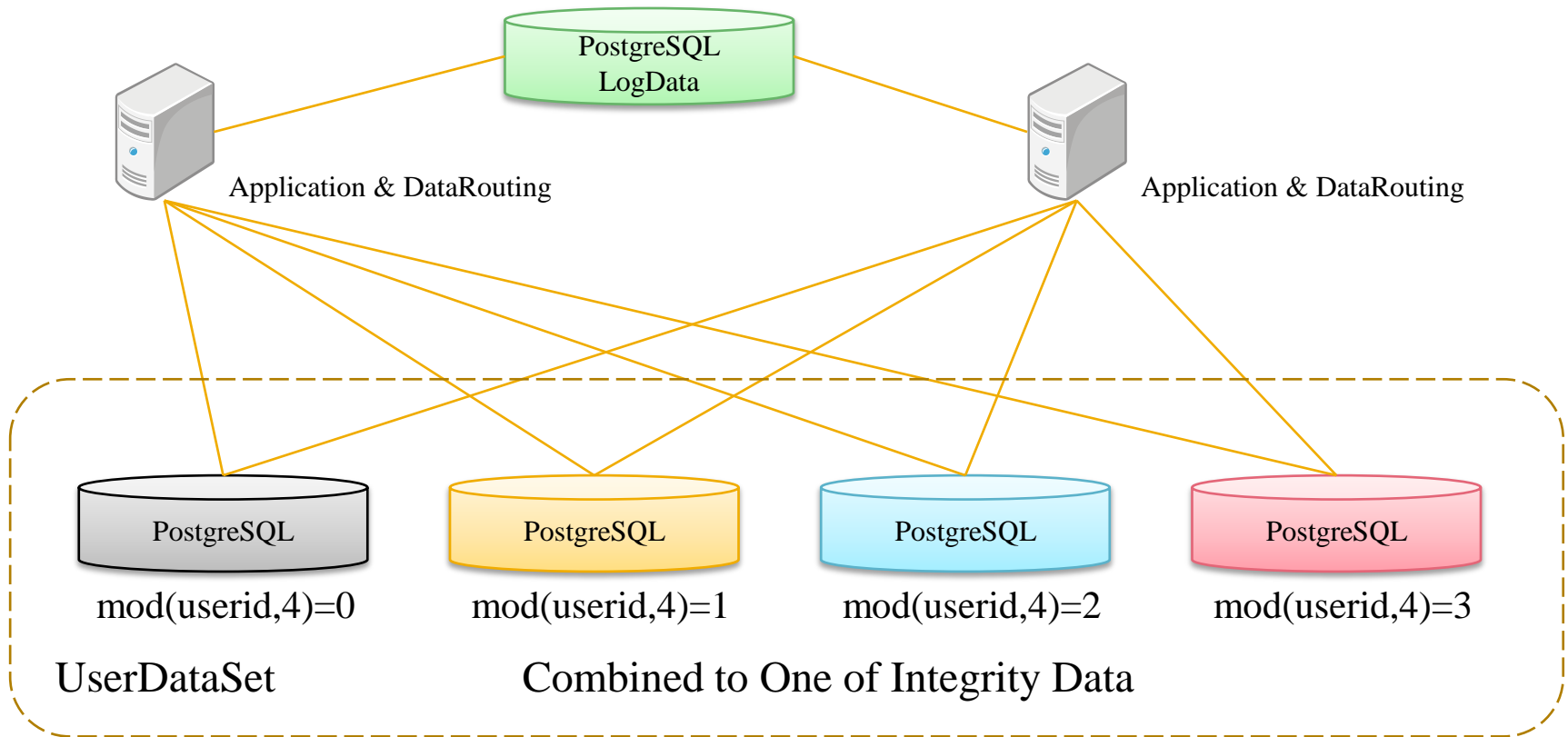
#### ■ 数据库水平拆分



## ■ 系统设计浅析

### ■ SNS型(动态)

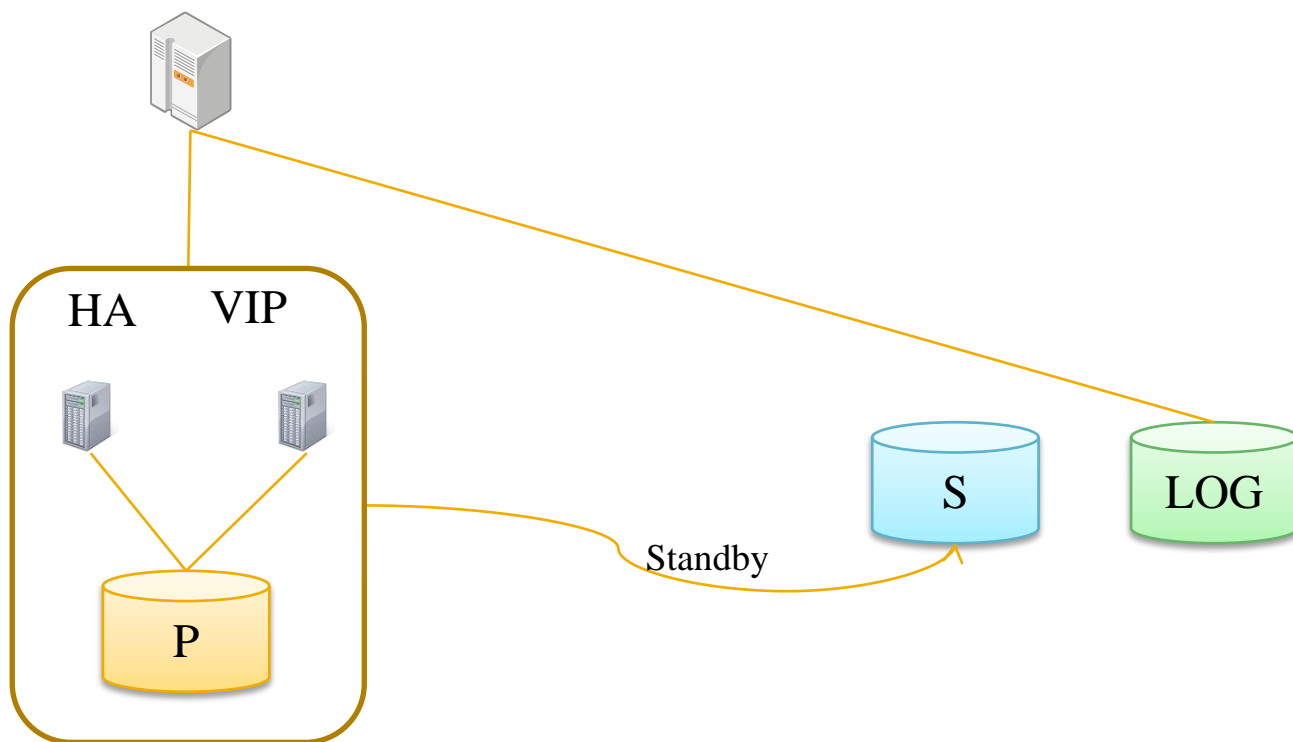
#### ■ 数据库水平拆分



## ■ 系统设计浅析

### ■ SNS型(动态)

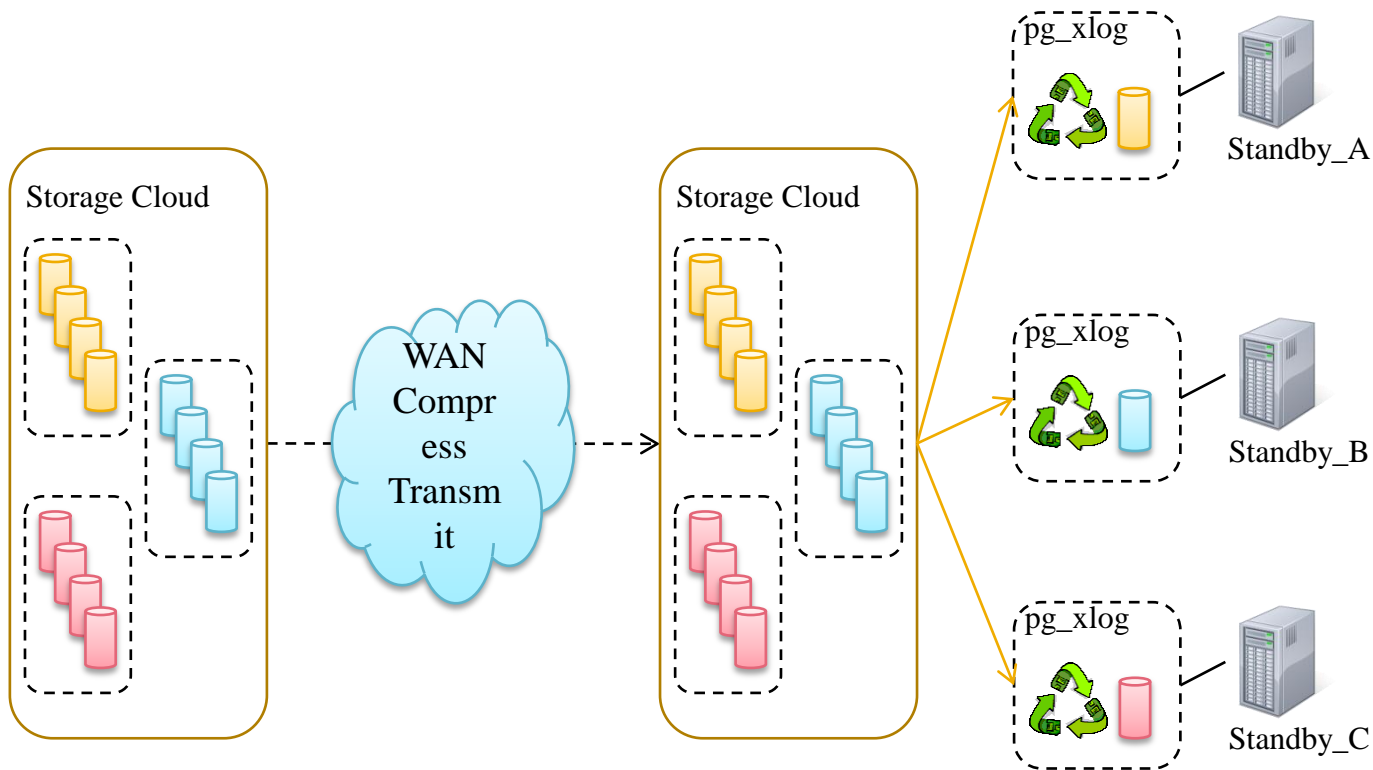
#### ■ IDC内部高可用设计



## ■ 系统设计浅析

### ■ SNS型(动态)

#### ■ 异地容灾设计

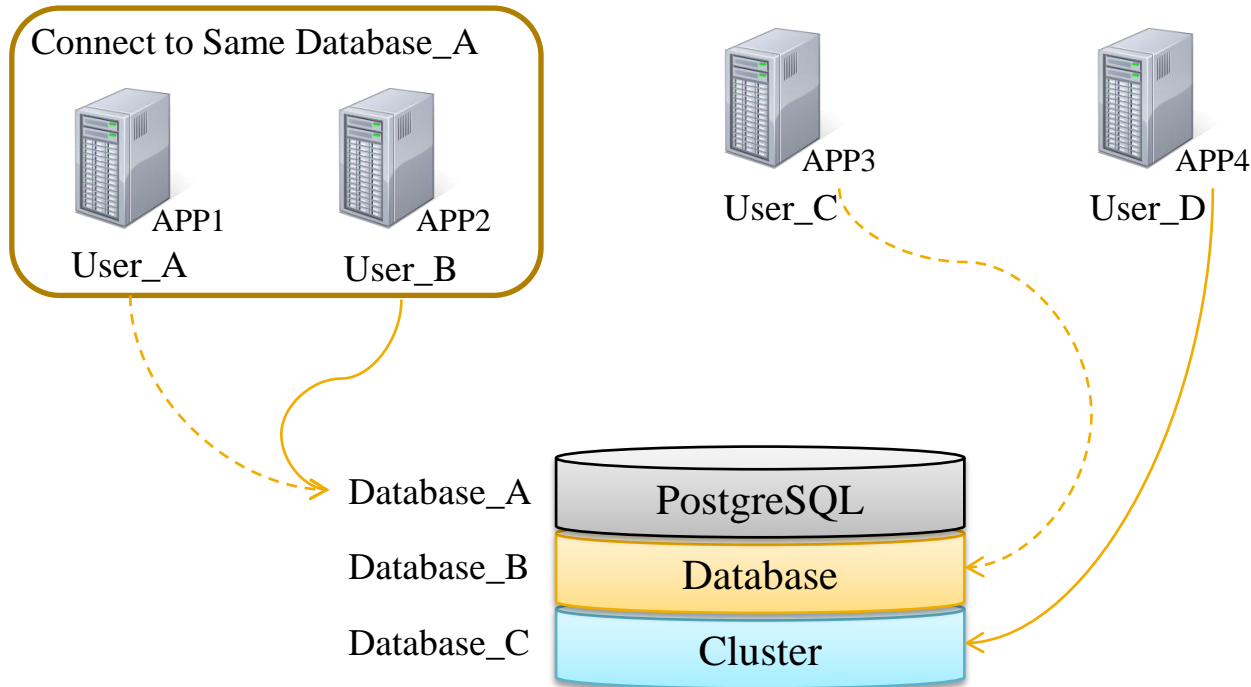


## ■ 系统设计浅析

### ■ SNS型(动态)

#### ■ 混合场景设计

Tuning settable run-time parameter using ALTER DATABASE or ALTER ROLE.



postgresql.conf :  
synchronous\_commit = on

User\_A :  
synchronous\_commit = off  
User\_B :  
Inherit postgresql.conf value.

Database\_B :  
synchronous\_commit = off  
Database\_C :  
Inherit postgresql.conf value.

## ■ SCHEMA设计

- 字段类型,长度,存储格式。
- fillfactor , autovacuum\_enabled , threshold , STATISTICS 相关参数。
- 索引,约束,PK,UK,FK,CHECK。

## ■ 函数开发

- Estimate data exists use perform replace select count(\*)
- Eliminate long transaction
- Eliminate complex operation

## ■ SQL优化

- Use prepared statement
- 批量提交
- Eliminate big result set return at once结果集指针
- Eliminate use database operation resource when application can do it
- Eliminate full table scan, use index when need it



- Loadrunner
- Pgbench
- 平均, 最高, 最低TPS, QPS, SQL响应时间.

## ■ 调整系统参数

### ■ /etc/sysctl.conf

- kernel.shmni
- kernel.shmall
- kernel.sem
- fs.file-max
- fs.aio-max-nr
- net.ipv4.ip\_local\_port\_range
- net.ipv4.tcp\_tw\_recycle
- net.ipv4.tcp\_max\_syn\_backlog
- net.ipv4.ip\_conntrack\_max
- net.ipv4.tcp\_timestamps
- net.core.rmem\_default
- net.core.rmem\_max

- net.core.wmem\_default
- net.core.wmem\_max
- vm.overcommit\_memory
- vm.overcommit\_ratio
- vm.lowmem\_reserve\_ratio

### ■ /etc/security/limits.conf

- nofile
- nproc
- core
- memlock

- 调整数据库参数
- # **Connection**
- listen\_addresses
- max\_connections
- superuser\_reserved\_connections
- unix\_socket\_directory
- unix\_socket\_permissions
- tcp\_keepalives\_idle
- tcp\_keepalives\_interval
- tcp\_keepalives\_count
- # **Memory**
- shared\_buffers
- maintenance\_work\_mem
- max\_stack\_depth ( ulimit -s )
- # **Kernel Resource Usage**
- max\_files\_per\_process
- # **Cost-based Vacuum Delay**
- vacuum\_cost\_delay
- vacuum\_cost\_limit
- # **Background Writer**
- bgwriter\_delay
- bgwriter\_lru\_maxpages
- bgwriter\_lru\_multiplier
- # **Asynchronous Behavior**
- effective\_io\_concurrency
  - (asynchronous I/O requests.  
Currently, this setting only affects  
bitmap heap scans)

## ■ # WAL Settings

■ wal\_level

■ synchronous\_commit

■ wal\_sync\_method

- (open\_datasync, fdatasync, fsync, fsync\_writethrough, open\_sync)

■ wal\_buffers

■ wal\_writer\_delay

■ commit\_delay

■ commit\_siblings

## ■ # WAL Checkpoints

■ checkpoint\_segments

■ checkpoint\_timeout

■ checkpoint\_completion\_target

■ checkpoint\_warning

## ■ # WAL Archiving

■ archive\_mode

■ archive\_command

■ archive\_timeout

## ■ # Streaming Replication

## ■ # Synchronous Replication

## ■ # Standby Servers

## ■ # Planner Method

## ■ # Planner Cost Constants

■ random\_page\_cost

■ effective\_cache\_size

## ■ # Genetic Query Optimizer

## ■ # Other Planner Options

■ default\_statistics\_target

■ constraint\_exclusion

- # Log
- log\_destination
- logging\_collector
- log\_directory
- log\_truncate\_on\_rotation
- log\_rotation\_age
- log\_rotation\_size
- log\_min\_duration\_statement
- log\_checkpoints
- log\_lock\_waits
- log\_statement
- # RUNTIME STATISTICS
- # AUTOVACUUM PARAMETERS
- autovacuum
- log\_autovacuum\_min\_duration
- autovacuum\_vacuum\_threshold
- autovacuum\_analyze\_threshold
- autovacuum\_vacuum\_scale\_factor
- autovacuum\_analyze\_scale\_factor
- autovacuum\_freeze\_max\_age
- autovacuum\_vacuum\_cost\_delay
- autovacuum\_vacuum\_cost\_limit
- # CLIENT CONNECTION DEFAULTS
- deadlock\_timeout
- statement\_timeout

# 运维阶段

- 监控
- 权限管理
- 审计
- 备份，恢复测试
- 性能报告
- 优化，与开发人员一起提高系统性能
- 扩容
- 资源调配(如业务量下滑，对应的数据库不再需要高端的硬件设备，涉及数据迁移，数据库合并等)
- 日志分析
- 空间回收(如删除归档,备份并删除不再需要的数据)

## ■ 监控

### ■ 监控要素

- 系统：CPU，内存，交换分区，网络，IOPS，磁盘，Kernel报错，硬件故障等。
- 数据库：慢查询，DEAD TUPLE比例，锁等待事件，buffer使用情况，严重异常日志等。

### ■ 常用软件

- 实时告警
- 趋势图
- sendmail

**Nagios**<sup>®</sup>

**cacti**

## ■ 权限管理

- 最小权限连接: `pg_hba.conf`
- 最小权限对象: `grant / revoke`

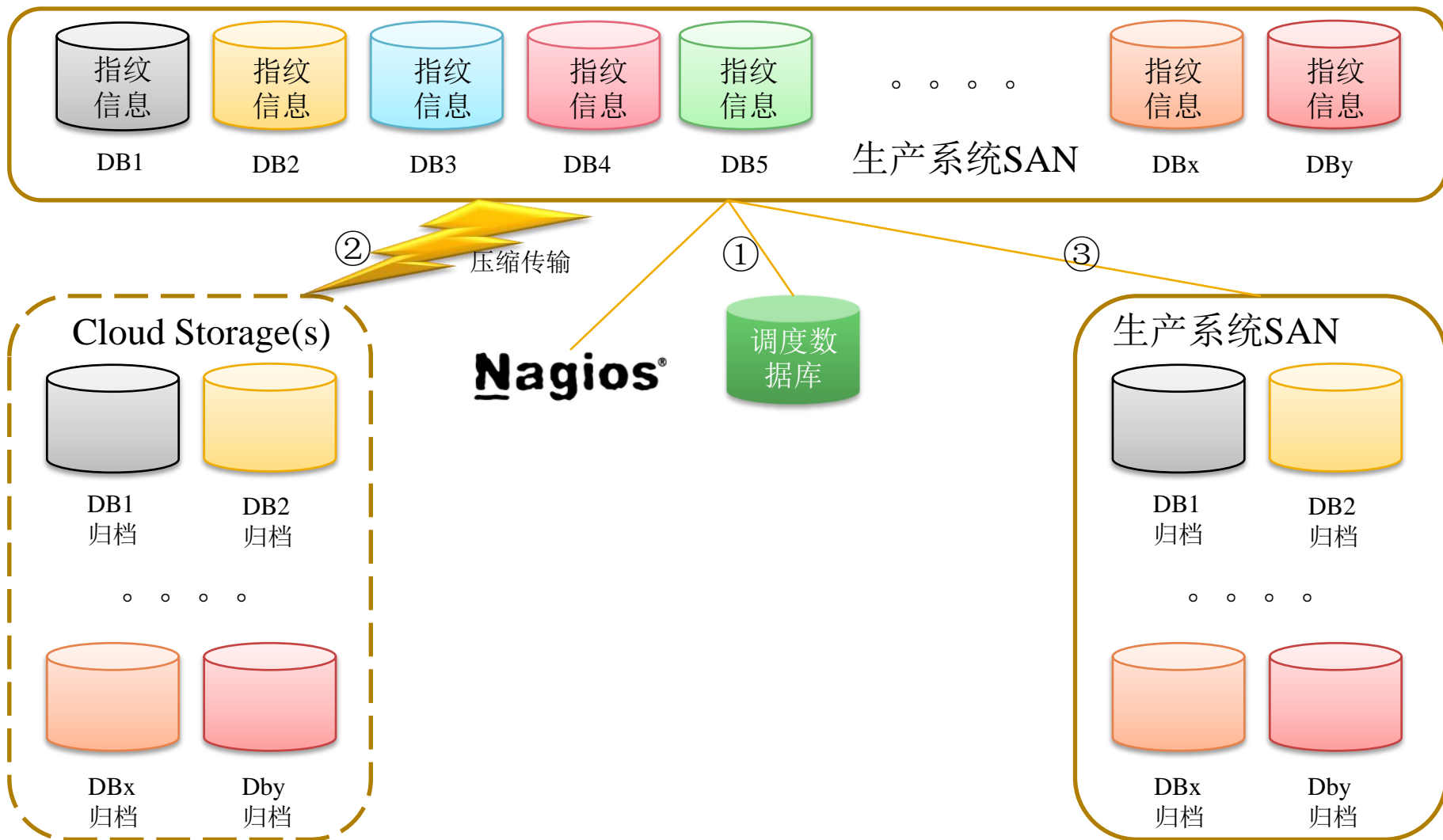
## ■ 审计

- 登录审计
- DDL操作审计
- 按用户审计(目前PostgreSQL还不支持)



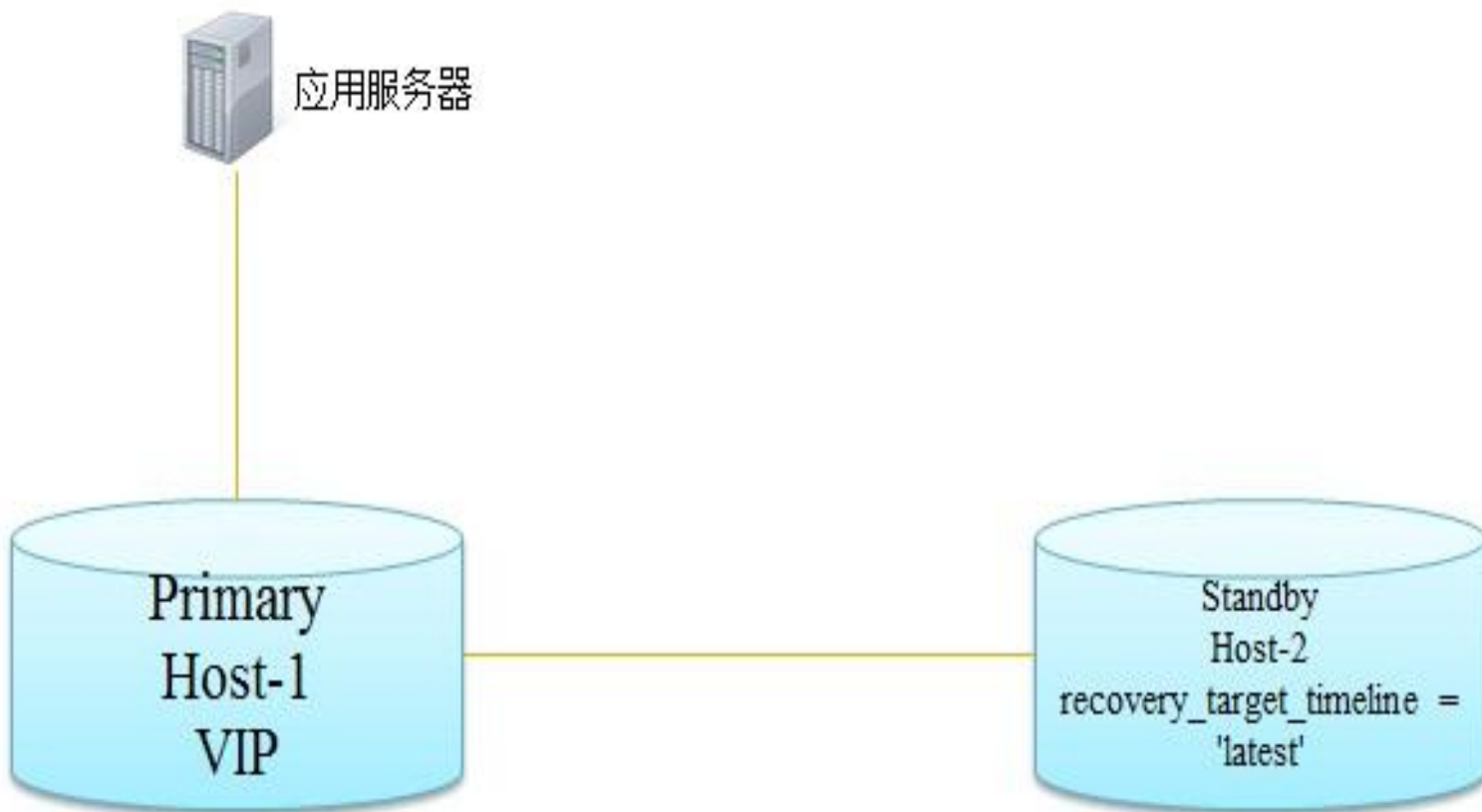
# 运维阶段

## ■ 备份

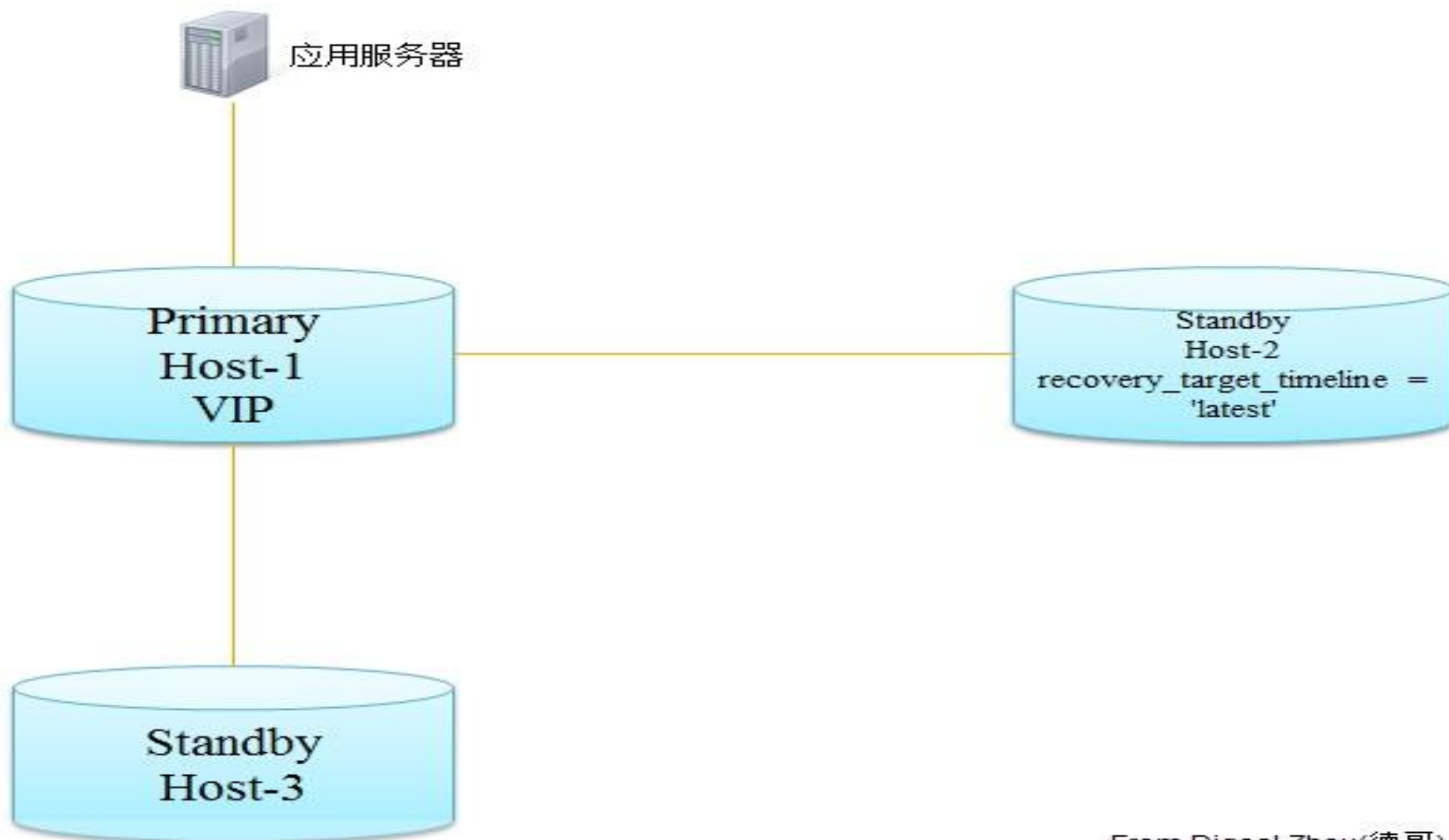


# 运维阶段

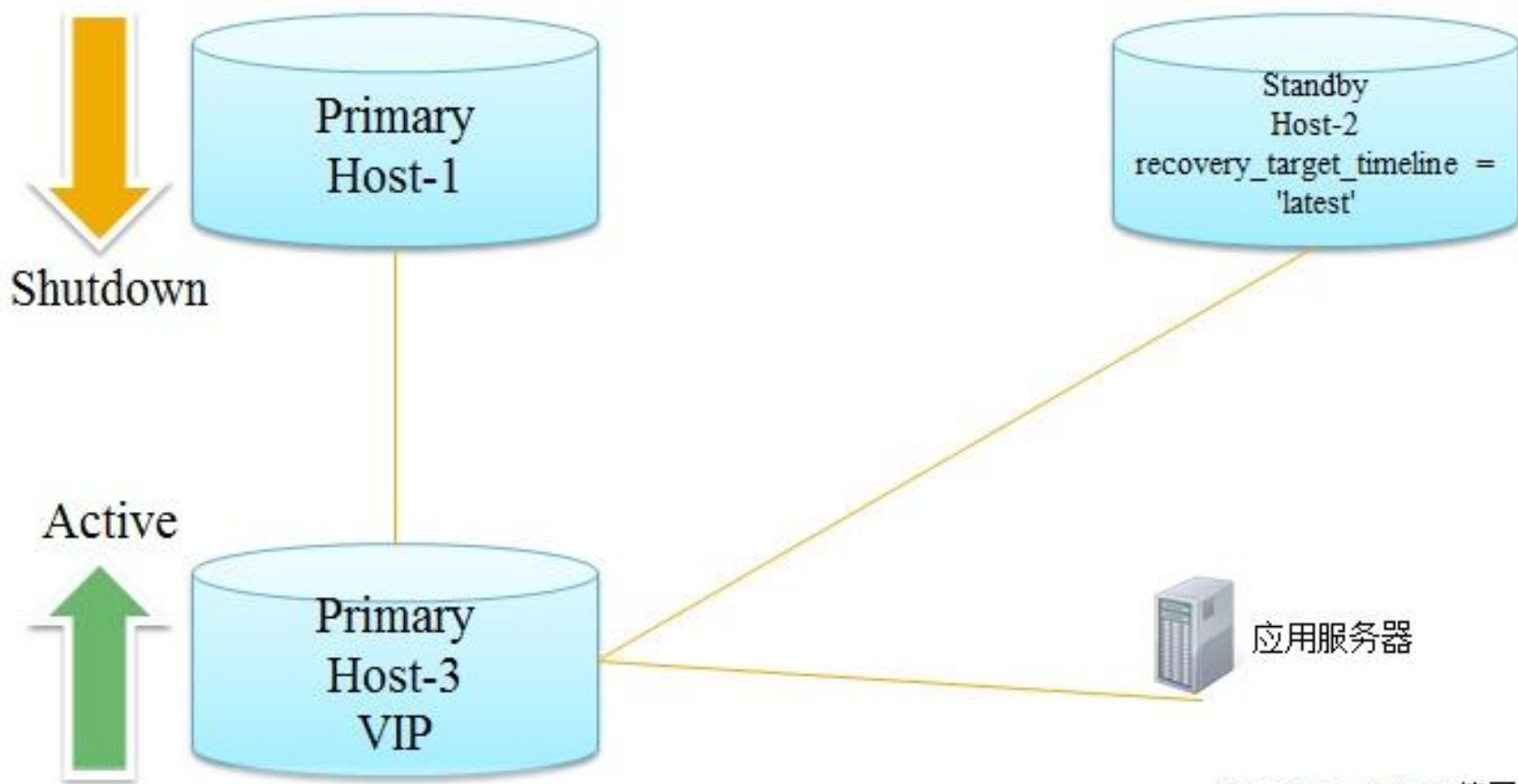
## ■ 扩容(接近平滑)



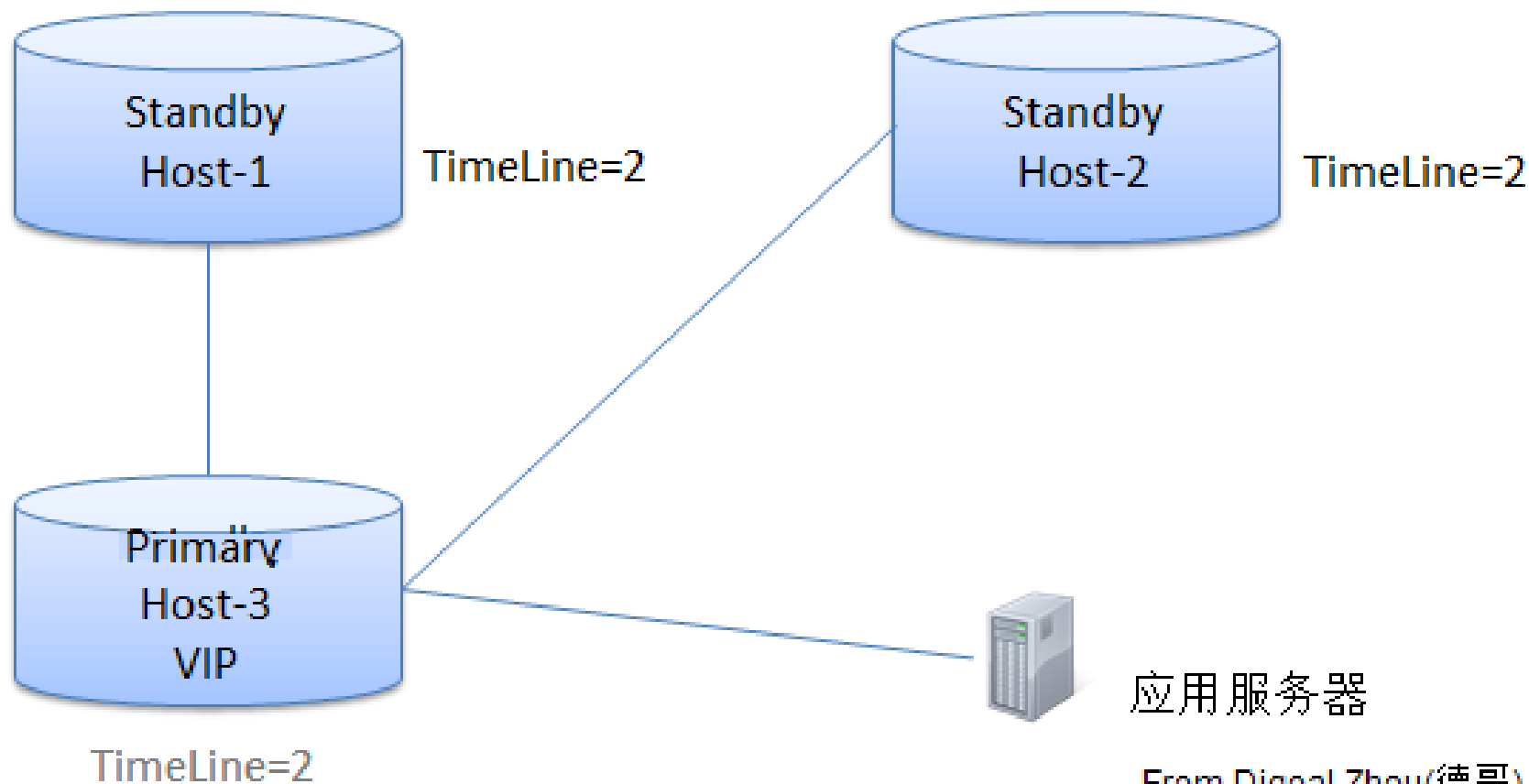
# 运维阶段



# 运维阶段



# 运维阶段



From Digoal.Zhou(德哥)

# Thanks

- Thanks all people contribute to PostgreSQL.



- Digoal.Zhou
- Blog
- <http://blog.163.com/digoal@126>