唱吧数据库系统架构的变迁之路

高洁
ki.ustc@gmail.com

唱吧是什么？


Pad
16:41
北京榜
全国榜
新星榜
NOKIA

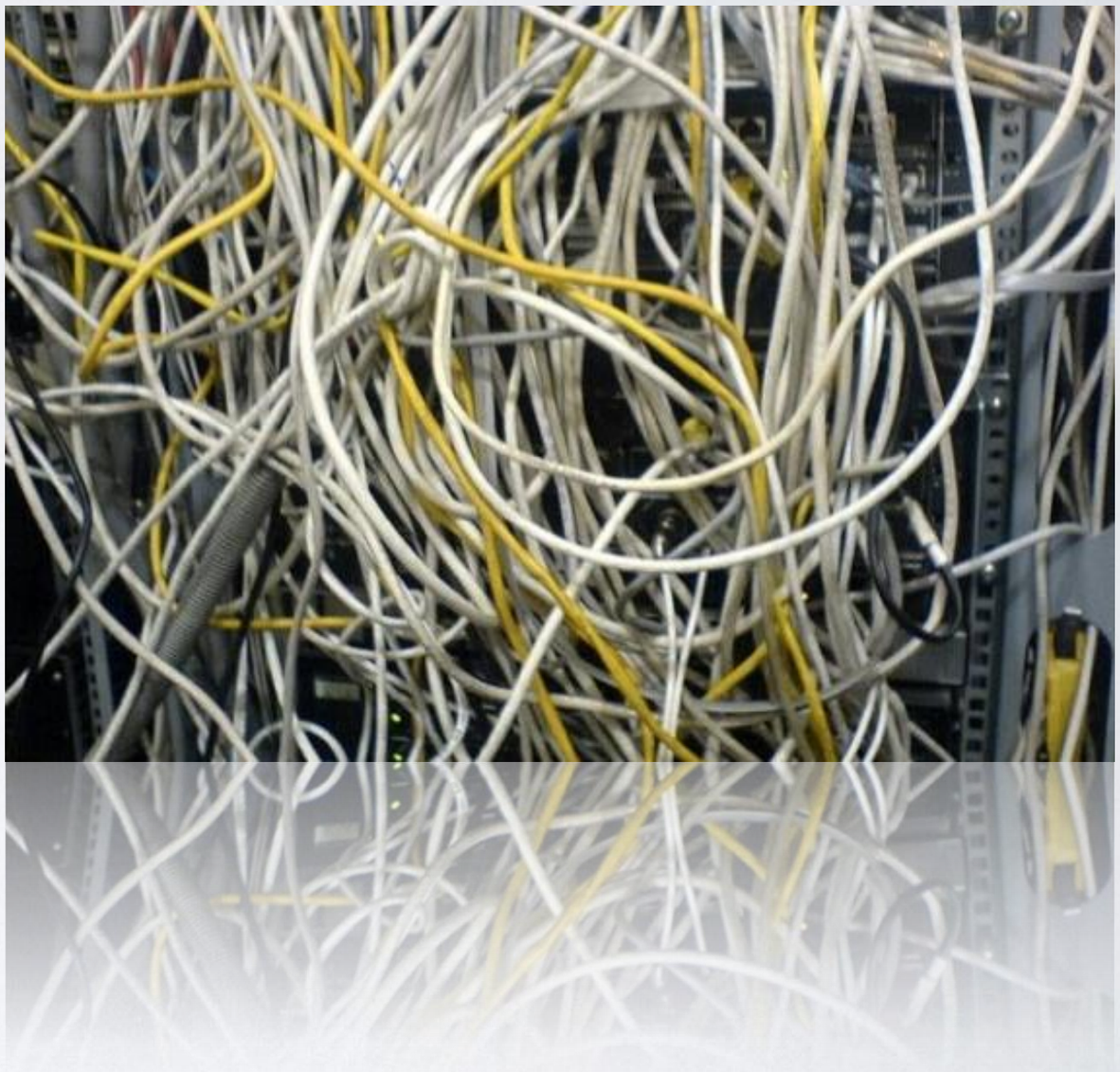
现状

- 2亿　　终端安装量
- 7000万　　注册用户
- 3000万　　每月活跃用户
- 500万　　每日活跃用户
- 15000　　夜间高峰qps

很多年前。。。

2012・5

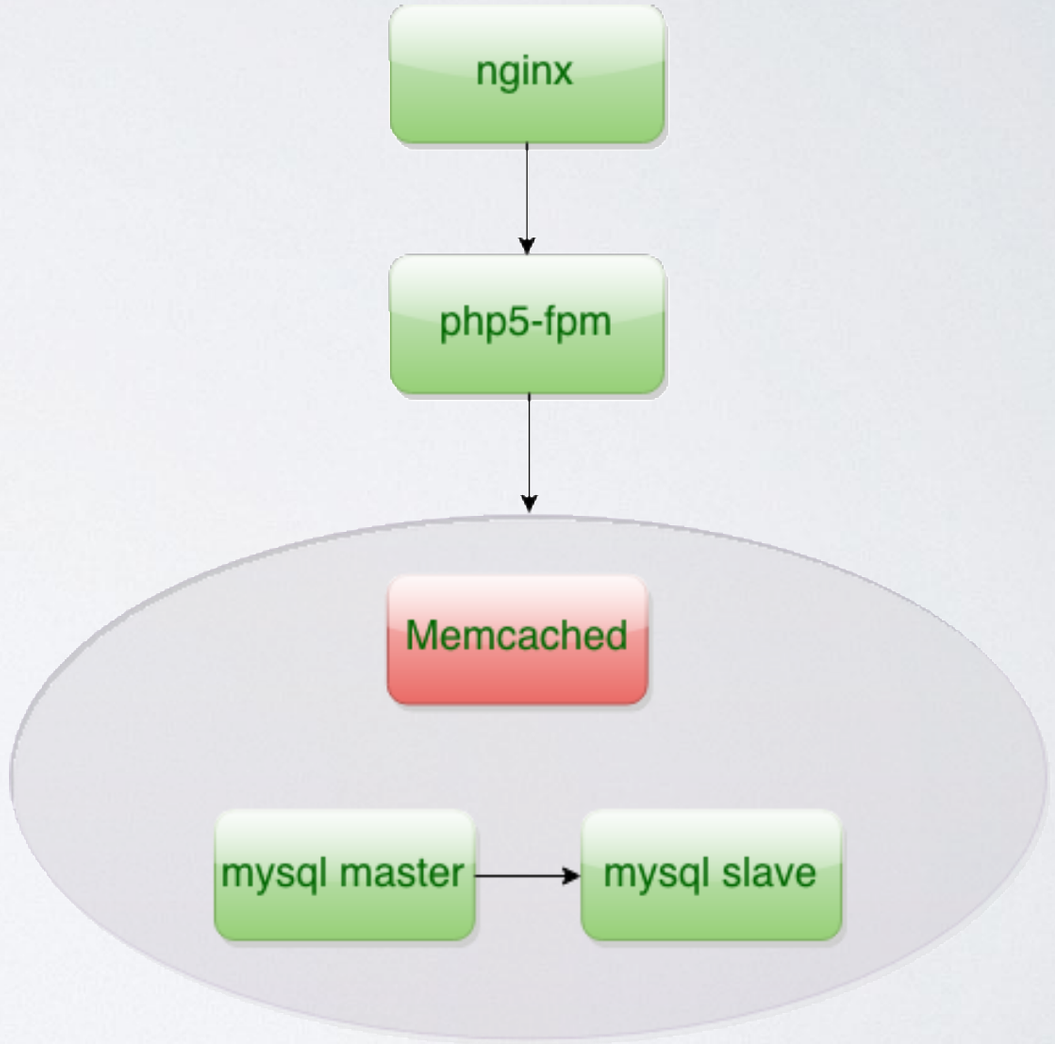
- 3台CentOS机器
- 1台 百兆路由器



- 就这样一团糟

第一次整理和维护

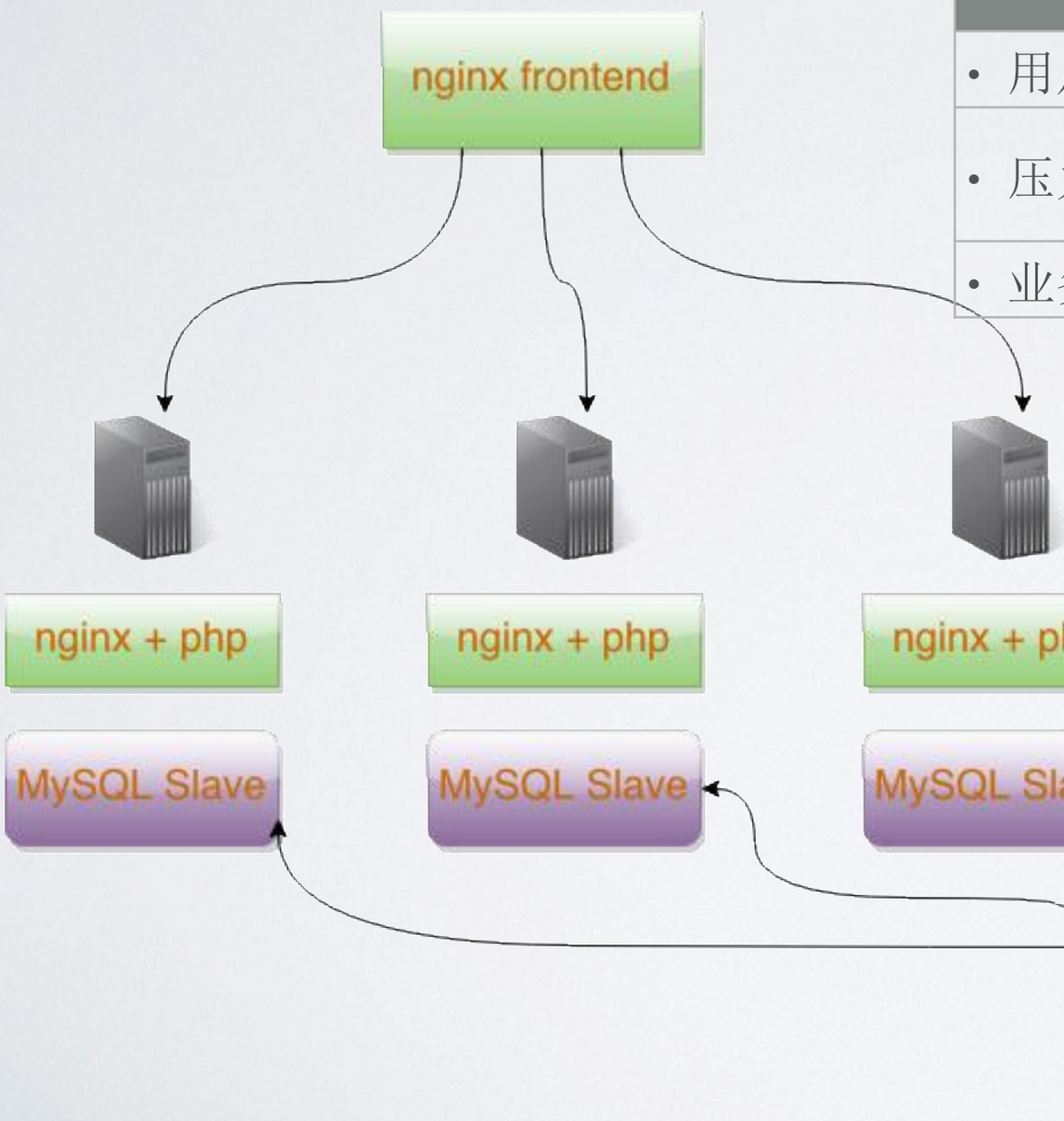
- 简单读写分离
- 主要精力：优化代码



上线第二天

不堪重负。。。

进一步扩充



出现问题	解决方案
• 用户量和访问量猛增	增加服务器
• 压力增大	多mysql从库，提高php性能
• 业务处理不合理	优化代码和缓存机制

上线第一周后

这样就好用了吗？

问题应接不暇

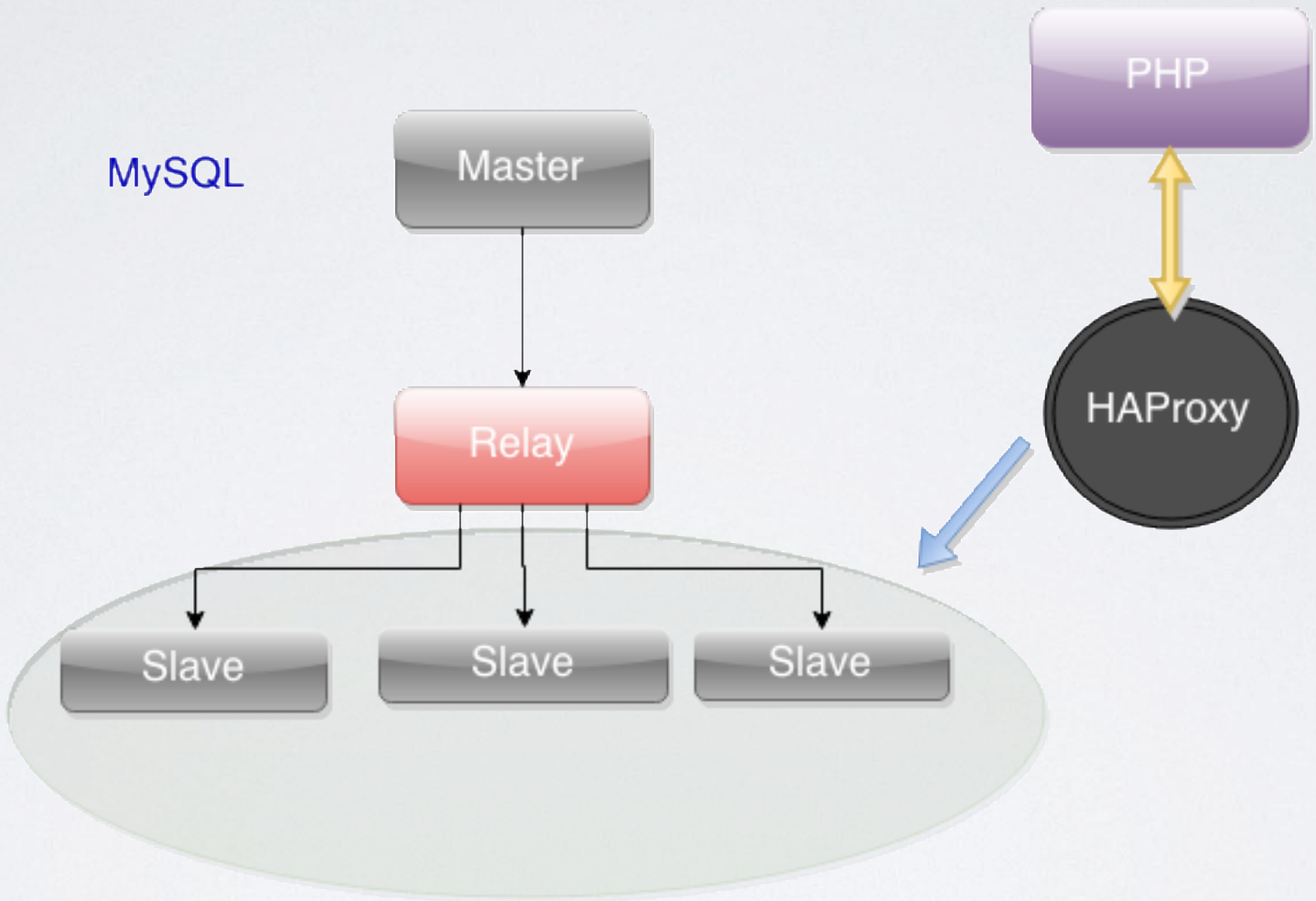
- 经典的主从延时问题
- 实时性要求高的查询出错
- 负载不均衡

第二次优化

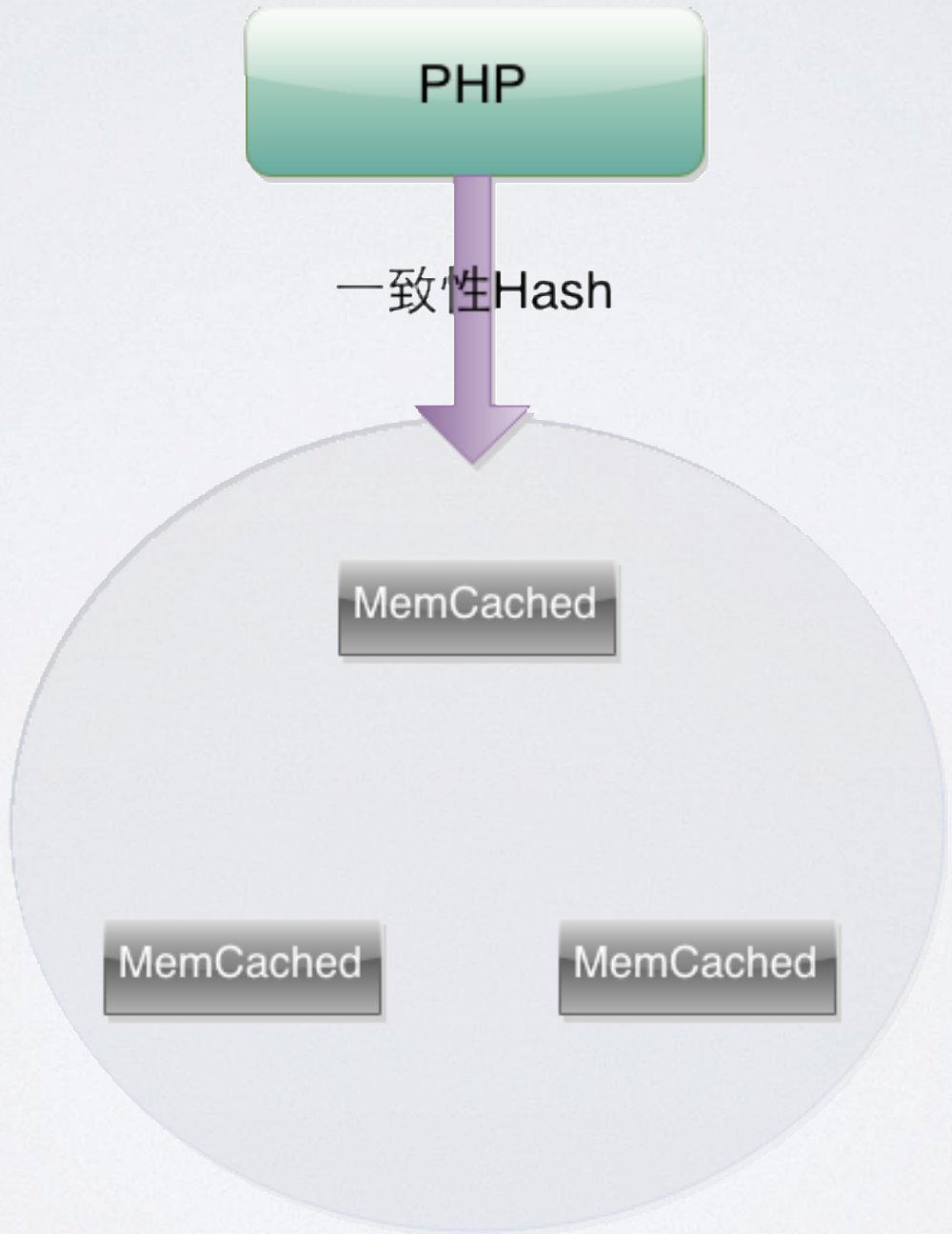
- MySQL引入relay转发
- 使用haproxy统一管理从库
- 多台Memcached，CRC hash

上线3个月，发布android

MySQL



MemCached



此阶段的总结

- 缓存的合理运用，降低MySQL命中
- SQL优化，减少读写锁
- MySQL数据库读压力调整
- HAproxy负载均衡

问题依旧

- user表的读取压力仍然过大
- user_relation表的读取和更新频繁
- 从库写入压力大，锁表严重

第三次重构

- MySQL不能适应所有场景

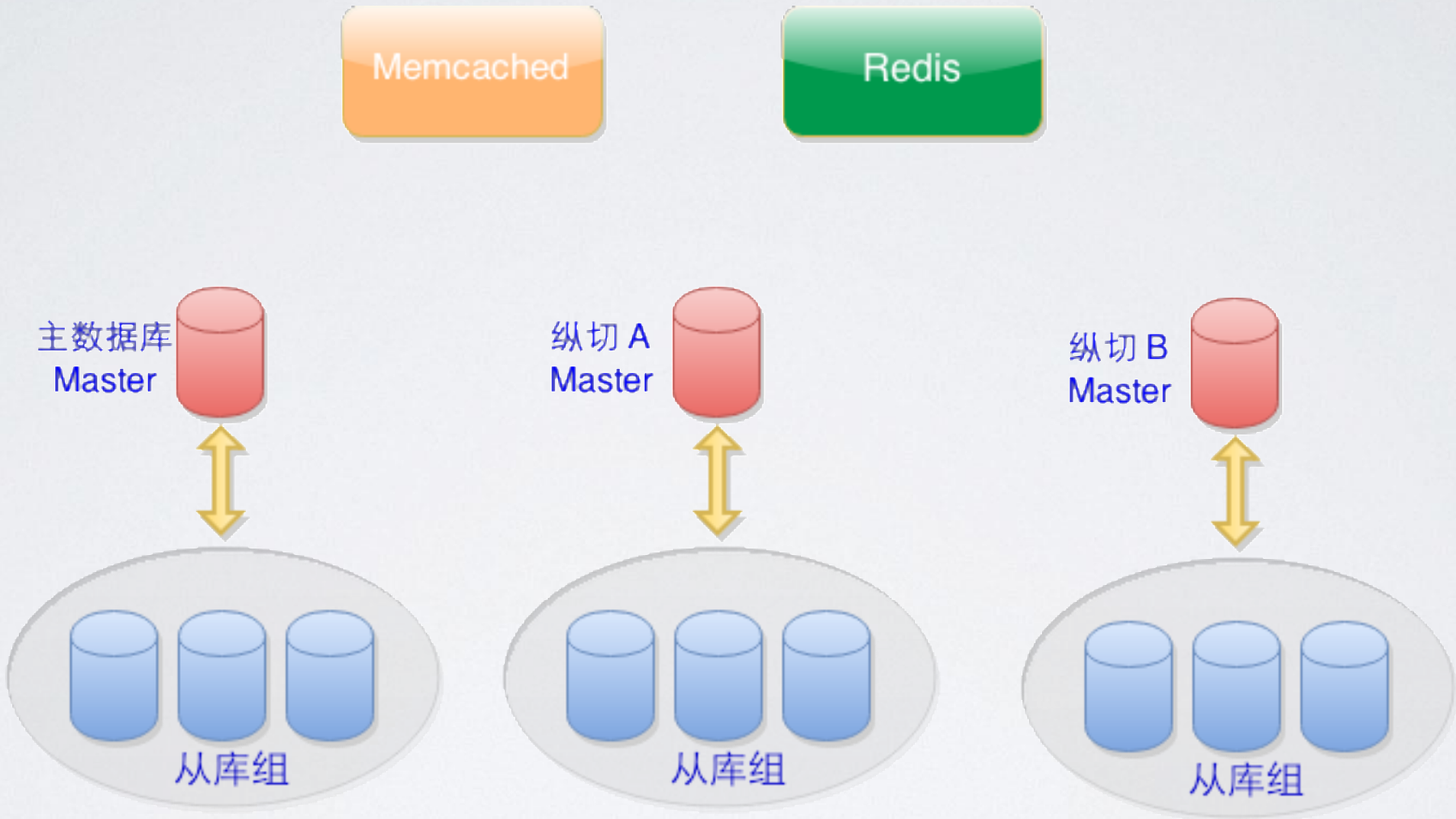
2012年年底

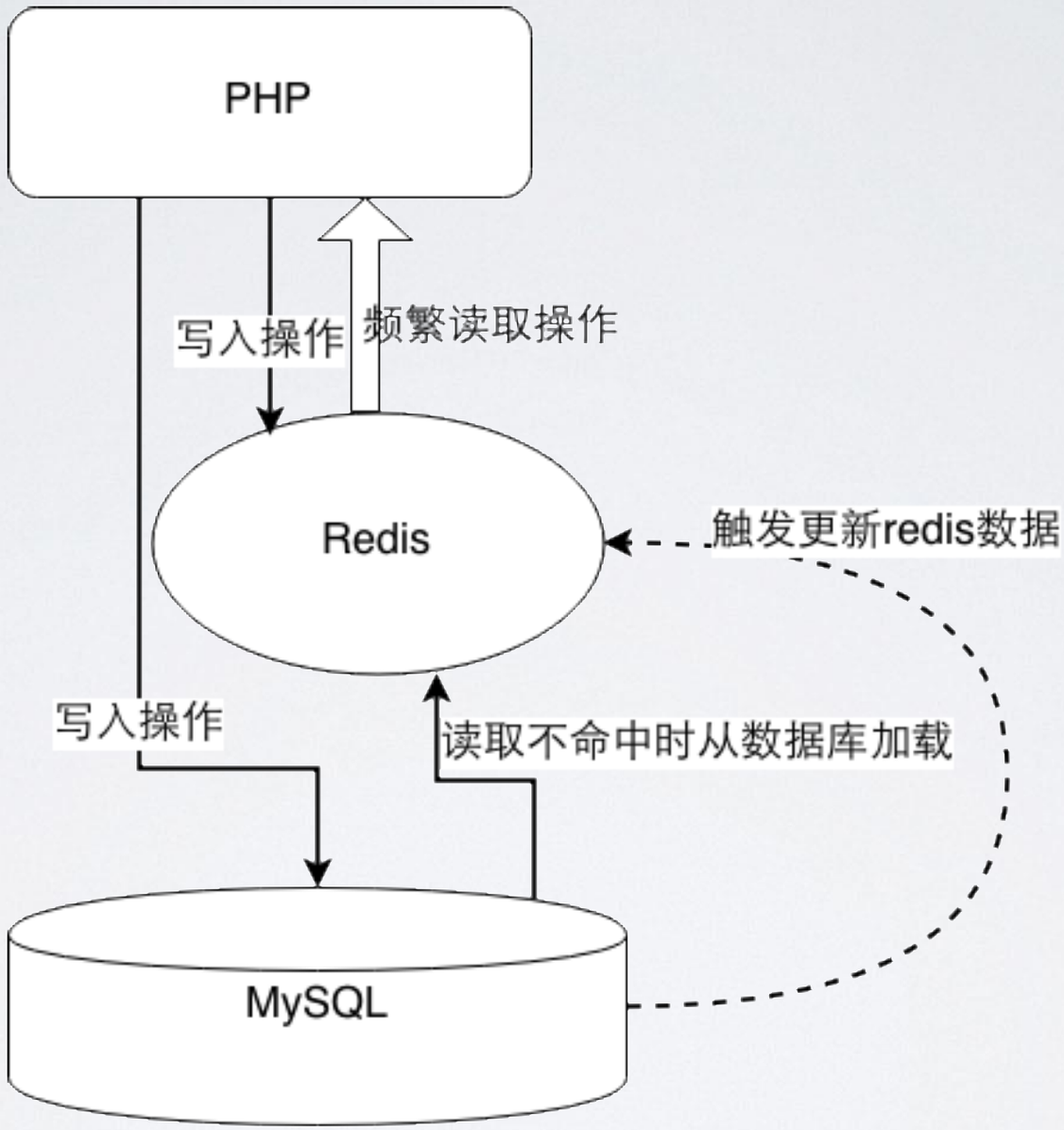
方案一　引入Redis

- 用redis的hash数据结构导入user表
- 用sorted set实现用户关系(user_relation)
- 用redis实现用户消息通知
- 用redis替代memcache缓存歌曲听过次数等数据，不再同步到数据库

方案二 纵切数据库

- 把user_relation，notice等表剥离到独立的数据库
- 拆分的数据库逐渐变成冷备，仅保留写入
- 引入MemcacheDB消息队列，对非及时数据库写入用异步完成


Memcached
Redis
主数据库
Master
纵切 A
Master
纵切 B
Master
从库组
从库组
从库组


PHP
写入操作
频繁读取操作
Redis
触发更新redis数据
写入操作
读取不命中时从数据库加载
MySQL

方案三　异地机房的尝试

- 独立的业务，放置到天津机房
- 跨机房数据访问：
  异地MySQL从库同步
  异地Redis从库
  Webdis实现redis远程写入
  访问安全性
- 网络波动造成的影响
- 失败的尝试

此阶段的总结

- 广泛应用redis到适合nosql的场景
- 把mysql定位为“离线”存储
- 合理使用多级缓存的机制
- 在业务逻辑上减少联表查询，高效利用索引

第四次优化 业务逐渐复杂

- QPS 请求量
- IO 磁盘
- 内存
- 网络 带宽
- 软件

2013年开始，一直持续至今

QPS

存储	理论值	线上场景
MySQL	2000~5000读写	2000左右
Memcached	5~10万读写	10K
Redis	6万 k/v 读写 3万 hash 读写	8~10K
APC Cache	10万+读写	10K+

I/O

- RAID 5
- RAID10
- SSD
- RamDisk

用 最快的存储设备 处理 访问最频繁的数据。

内存

存储	占用	说明
MySQL	较小	innodb pool query cache
Memcached	较大	速度极快 快满时会会出现缓存失效等状况
Redis	较大	60%以后性能下降 80%以后非常危险 回收能力差

网络

- 带宽会成为redis&memcached等服务的瓶颈
- Memcached不能滥用，拥堵网络

软件

在合适的场景选用合适的工具。

MySQL引擎选用

Engine	运用场景
Blackhole	黑洞，在relay从库上用于消除磁盘IO
innodb	事务；主库上提高写入速度；
myisam	从库查询；count(*)；
TokuDB	在表的总行数过亿以后速度远超innodb
XtraDB	升级版的innodb
HandlerSocket	直接读取innodb文件，无锁，快；不稳定

MySQL本身优化

- 版本选取： MySQL5.0， 5.5， MariaDB 10，Percona
- my.cnf参数优化
- Emoji 字符支持
- 水平拆分表（支付order，主题，评论等数据）

MySQL从库监控

- 自动检测从库状态，以及主从延迟
- haproxy自动剔除和加入从库
- 服务的监控报警

其他开源软件的广泛应用

- 结合MemcacheDB和Beanstalk，实现消息队列和异步的作业系统
- Solr用于昵称、曲库等数据的搜索
- MongoDB等其他NoSQL的尝试
- 基于leveldb在ssd磁盘上实现评论数据的存储
- FastDFS用于静态资源文件的分布式存储
- …

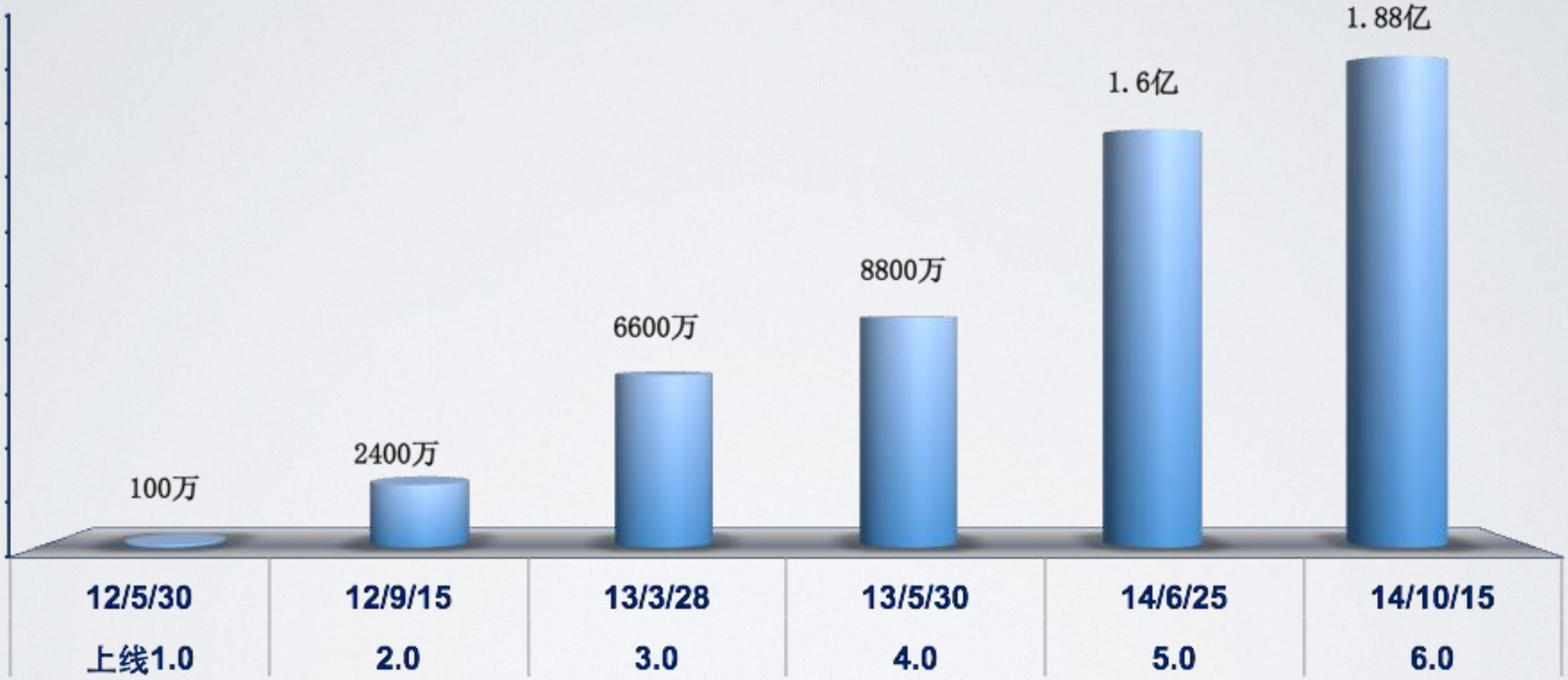
Redis集群的建设

- 单机性能的瓶颈
- 多instances的优势和劣势
- 用集群替代分散的“分布式”Redis
- 备份和扩容
- 中转 Redis Proxy实现（类比twemproxy）

总结

- 找到系统的瓶颈
- 选取合适的数据库
- 充分利用内存
- 考虑扩展性

应对唱吧用户和业务飞速增长



“谢谢大家。”

- Gao Jie