

内存数据库服务运营之路

@启盼cobain

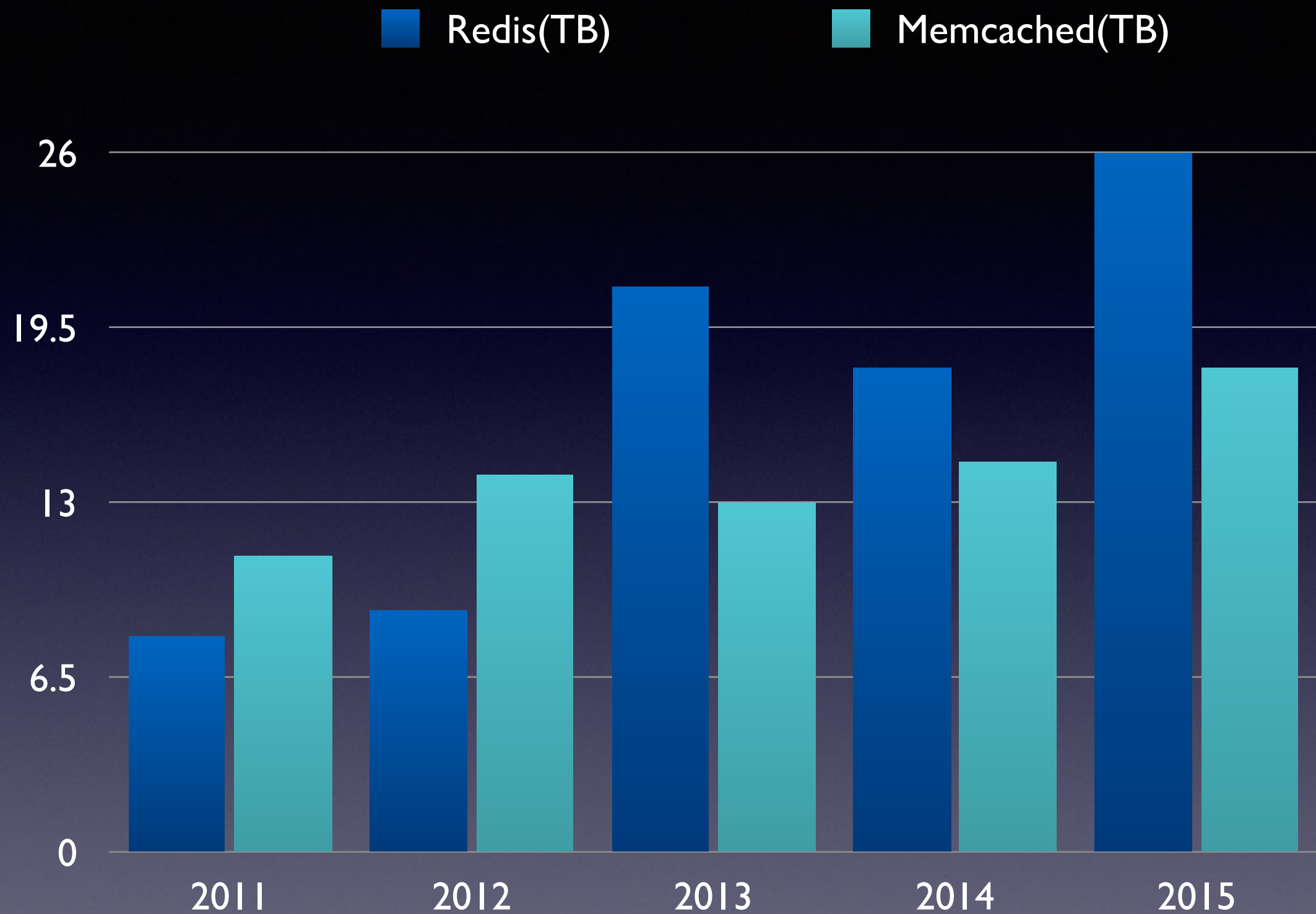
“RAM is the new disk...”

— Jim Gray

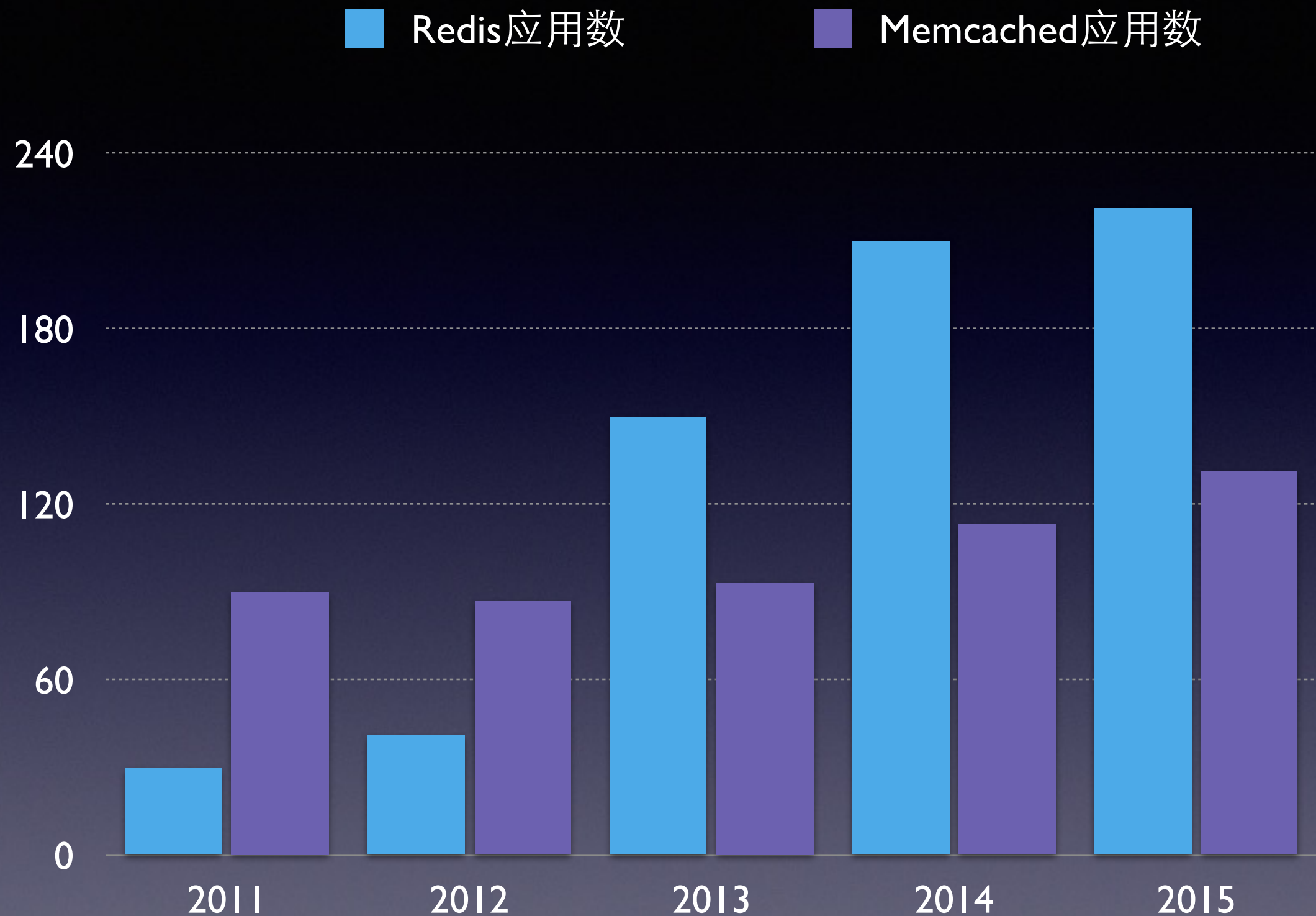
目录

- 新浪内存数据库服务发展历程
- 内存数据应用存储架构演进
 - 计数服务存储演化
 - 社交图谱存储演化
- OPS组织结构及运维系统架构演进
- 反思与总结

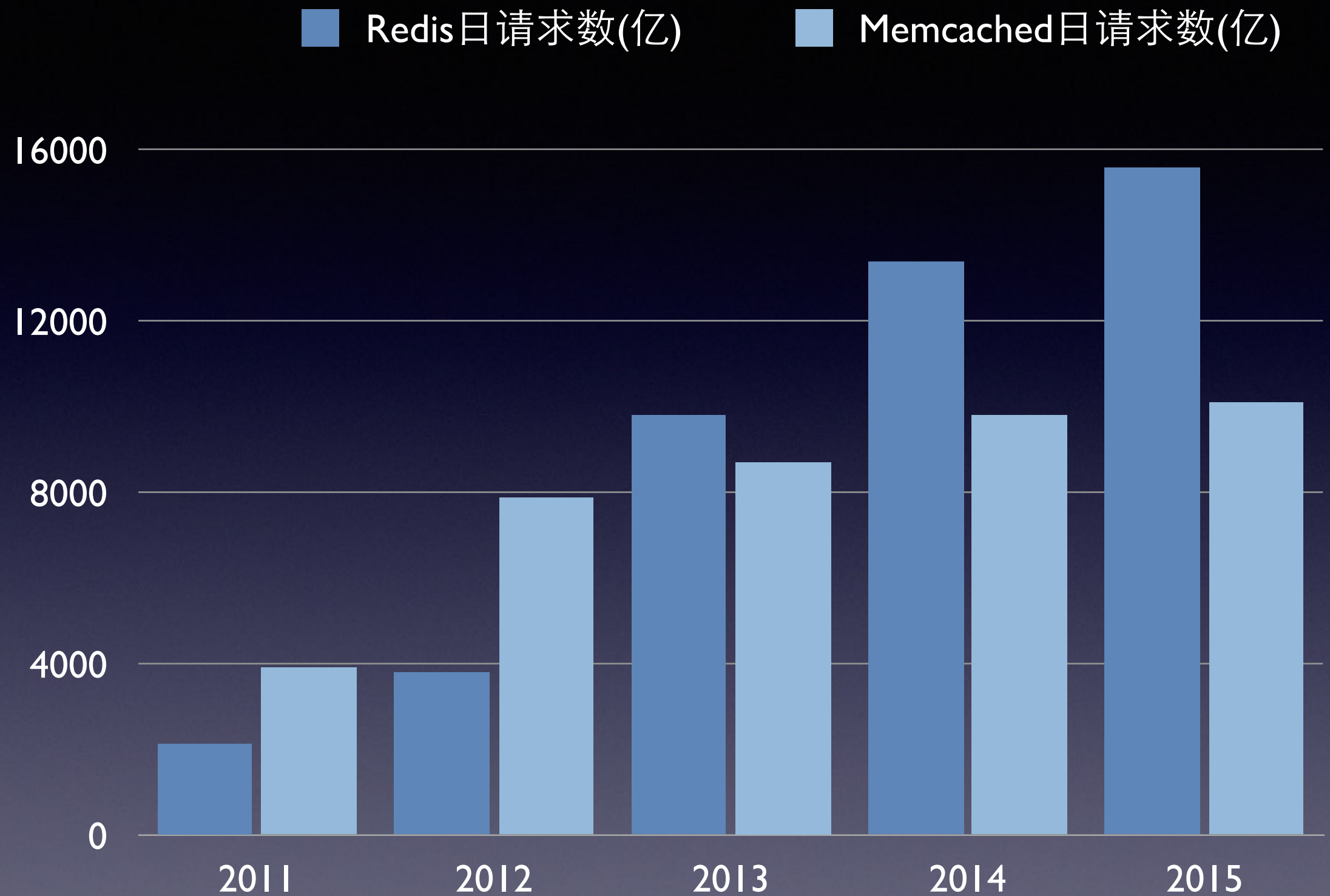
2011-2015



In memory application roadmap in Sina



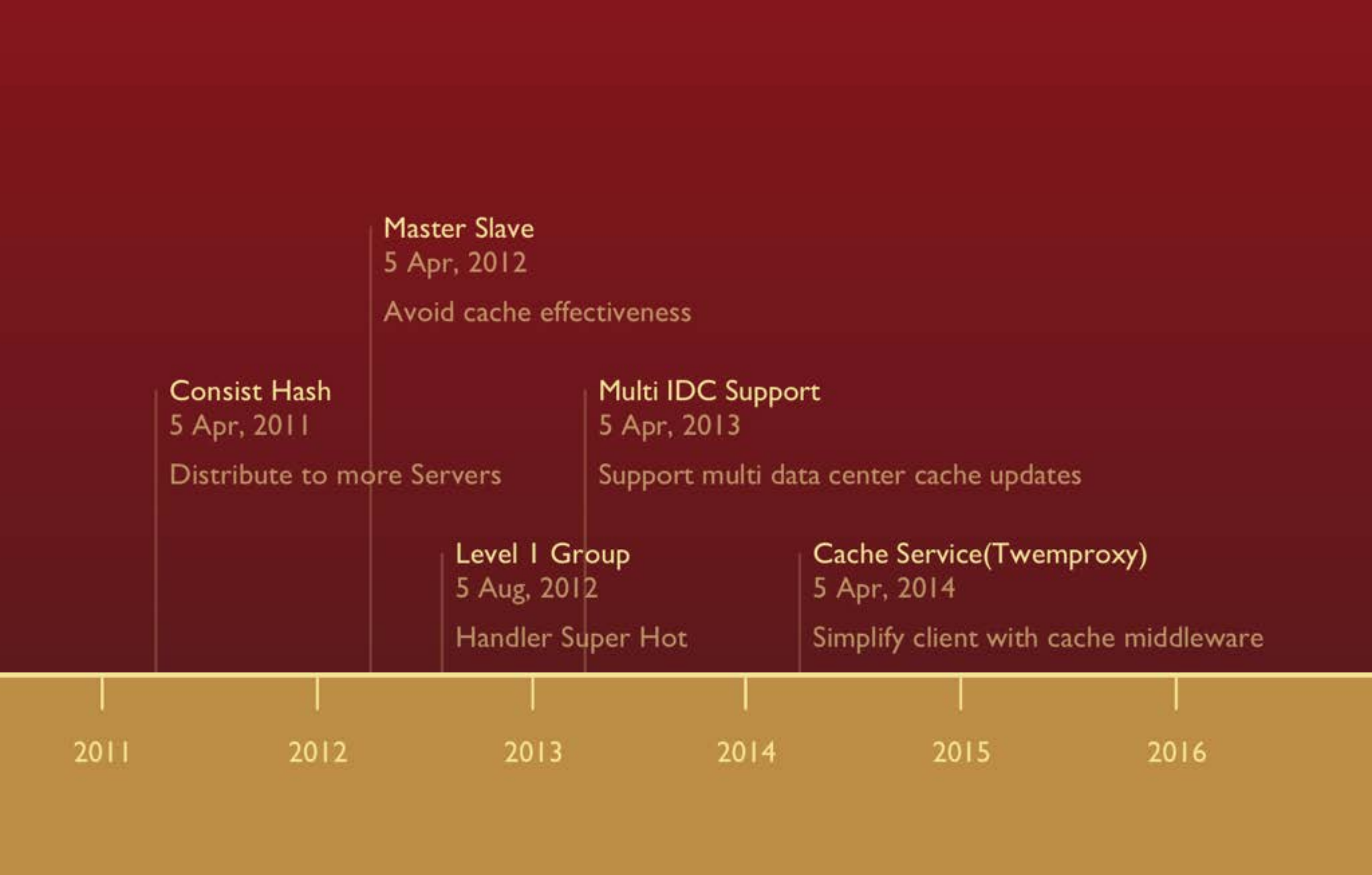
In memory application roadmap in Sina



In memory application roadmap in Sina

2011-2015

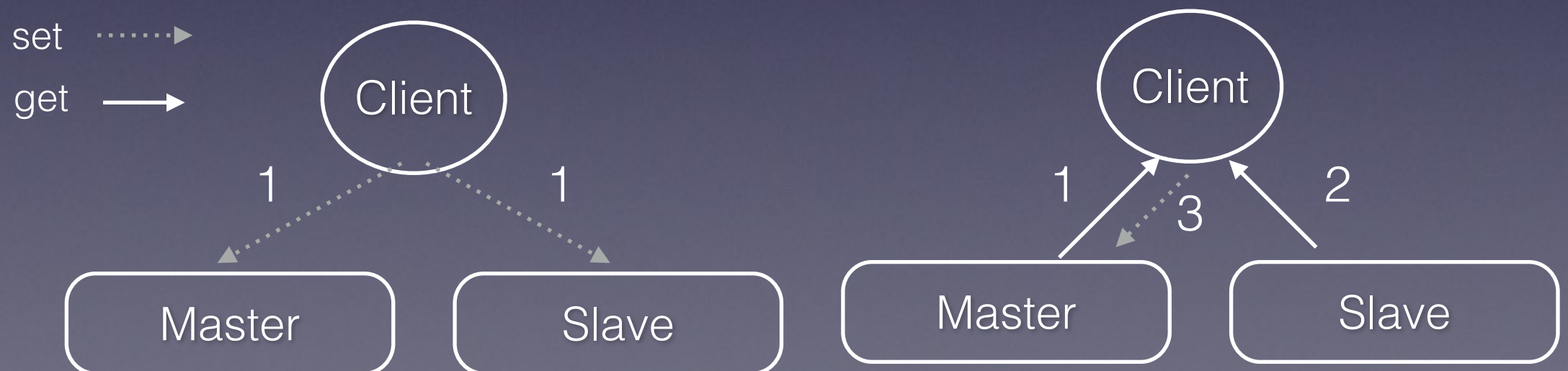
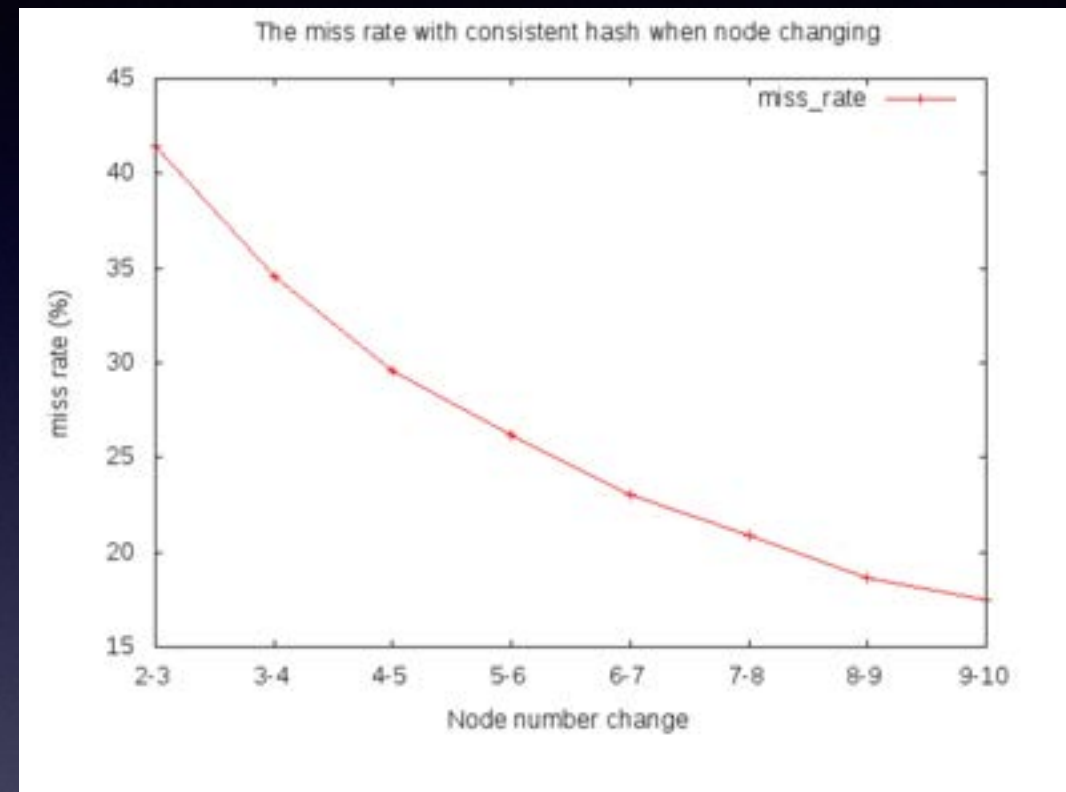
- 面临挑战
 - 2011 微博索引Memcached请求量翻番 -> 缓存失效风险
 - 2012 微博计数服务Redis改造上线 -> Redis高可用
 - 2013 异地机房部署 -> 跨机房数据更新
 - 2013 微博核心缓存请求量再次翻番 -> 缓存面临超级热点
 - 2013 年内存容量增长三倍 -> Redis应用内存疯长
 - 2013 2014 App数量及请求量增长四倍 -> 运维效率
 - 2015 2016 ?



Memcached Roadmap in Sina

应对缓存实效风险

- 共享内存，更多分片
(without multiget)
- 主备缓存(with multiget)



应对跨机房数据更新

- 消息总线 (e.g. WMB)

- 优势：不依赖队列外其他组件
- 劣势：消息总线复杂性较高

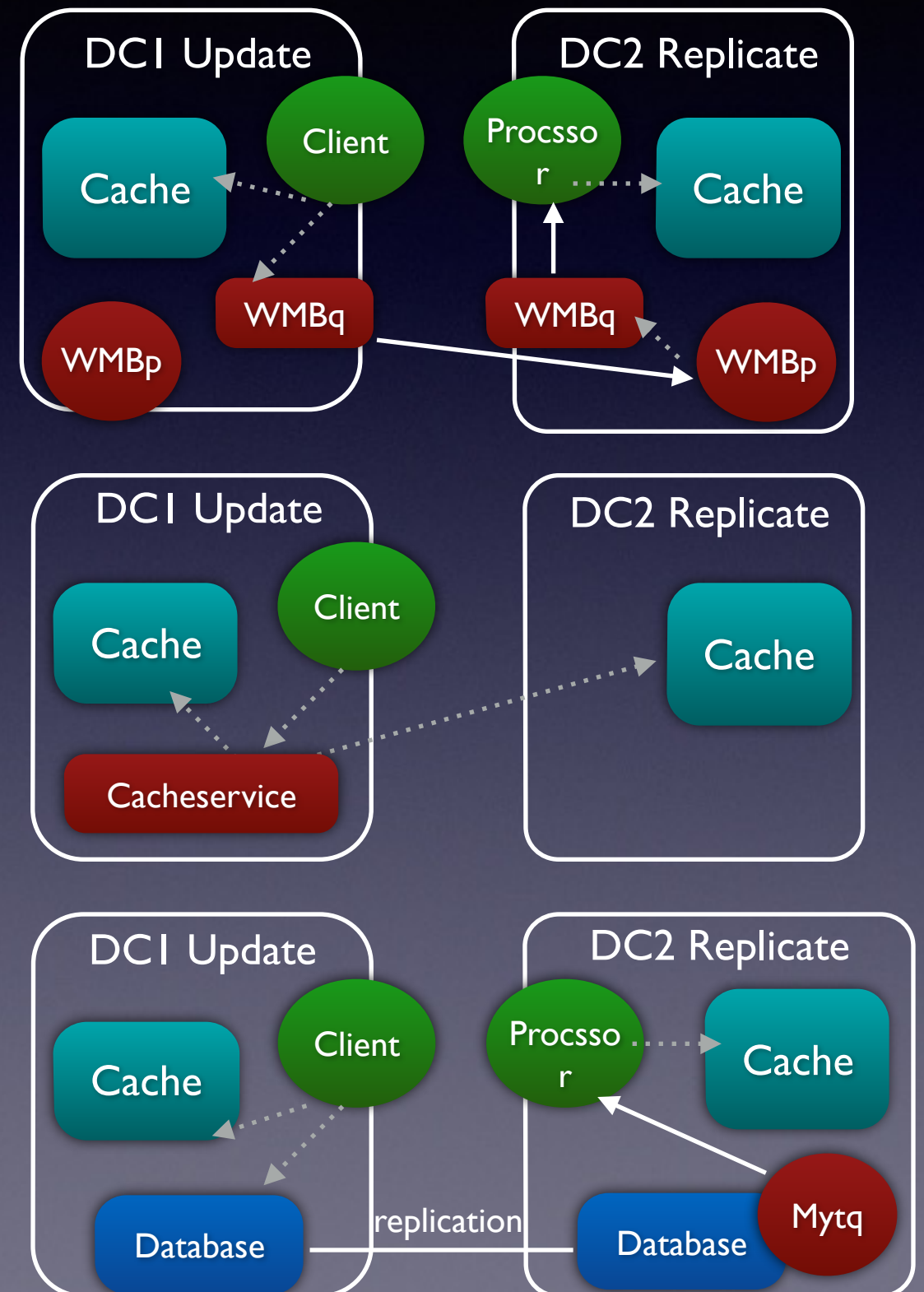
- 中间件更新 (e.g. Cacheservice)

- 优势：应用透明，相对运维友好
- 劣势：可靠性较低

- 数据库复制 (e.g. MySQL Replication + Mytriggerq + Processor)

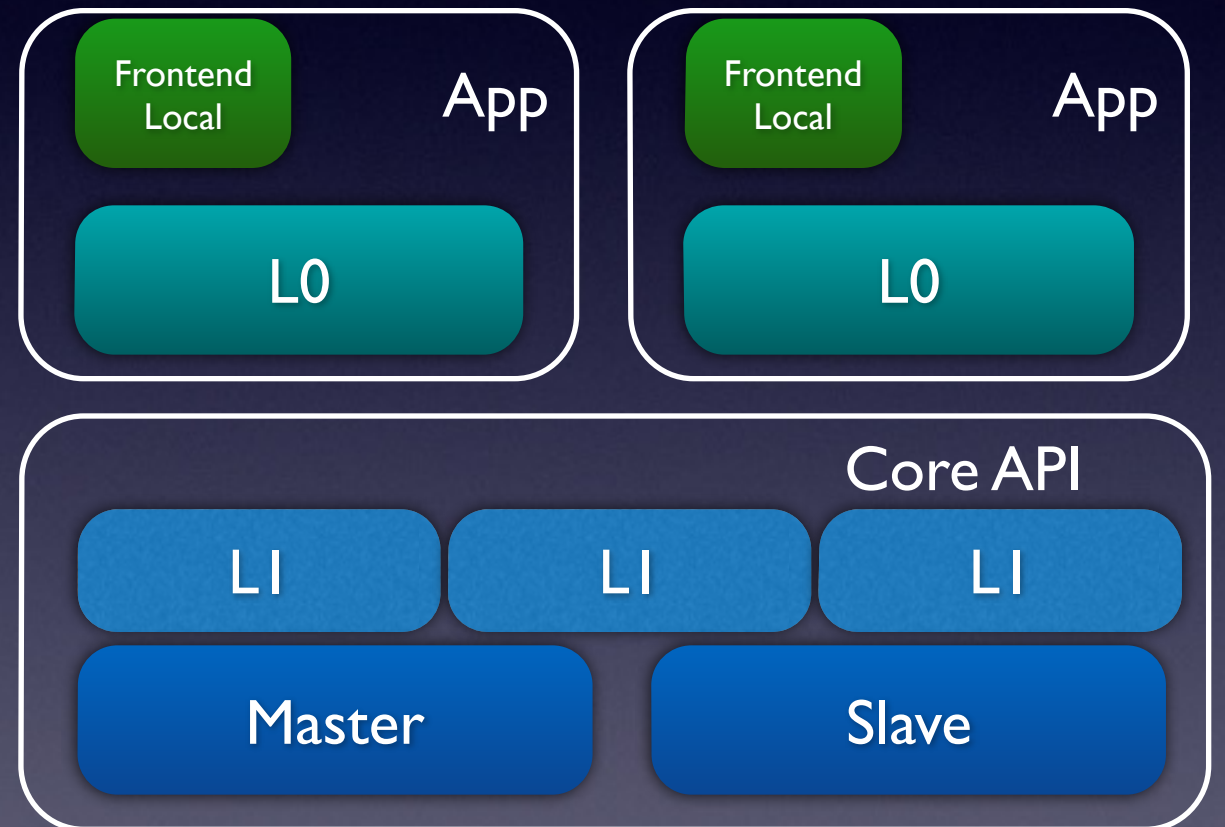
- 优势：可靠
- 劣势：维护额外数据库同步，消息格式转化相对受限。

set→
get →



应对超级热点

- 多级缓存
- 核心缓存快速构建数据副本
 - Memcached LI Group

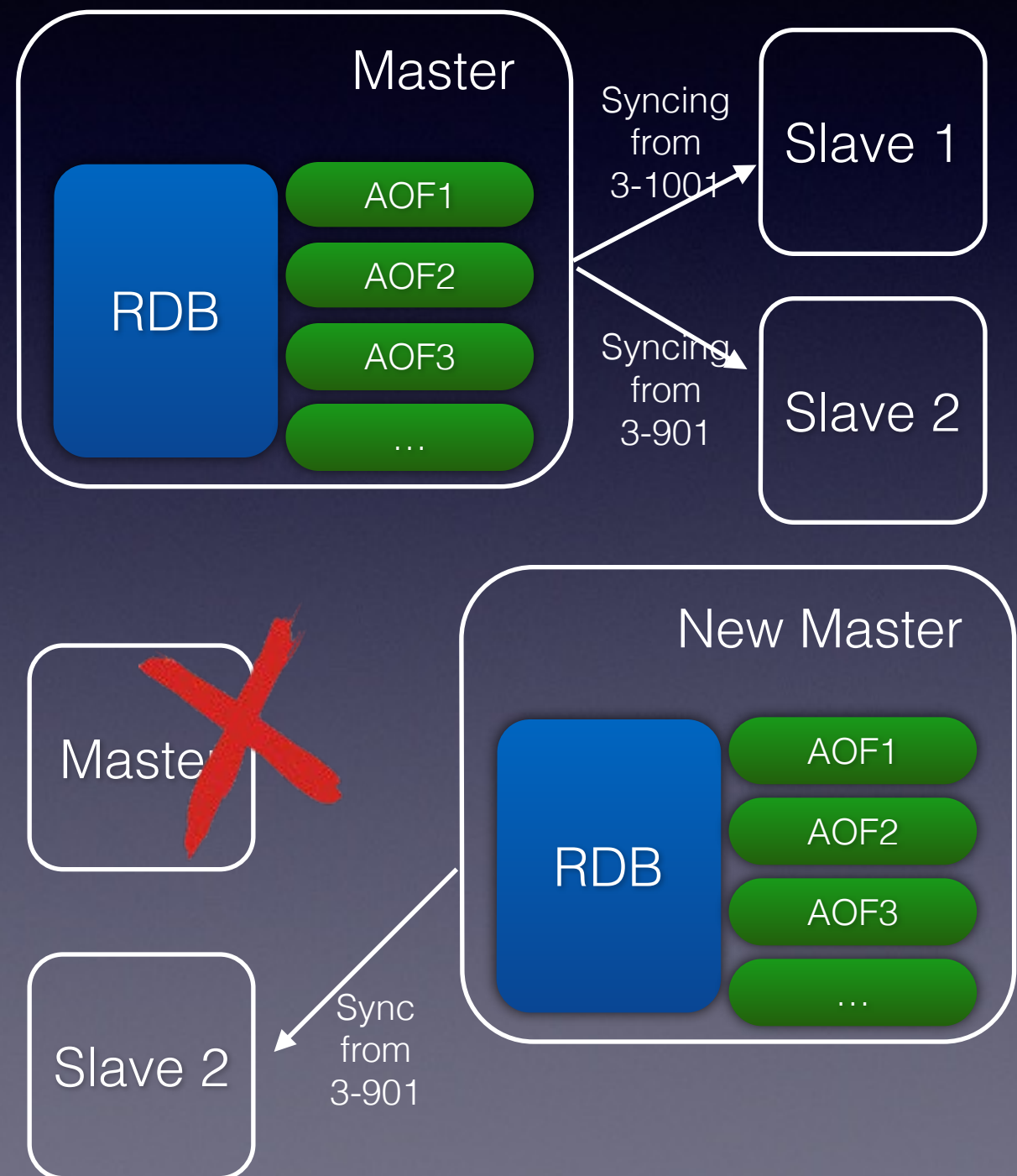




Redis Roadmap in Sina

Redis高可用

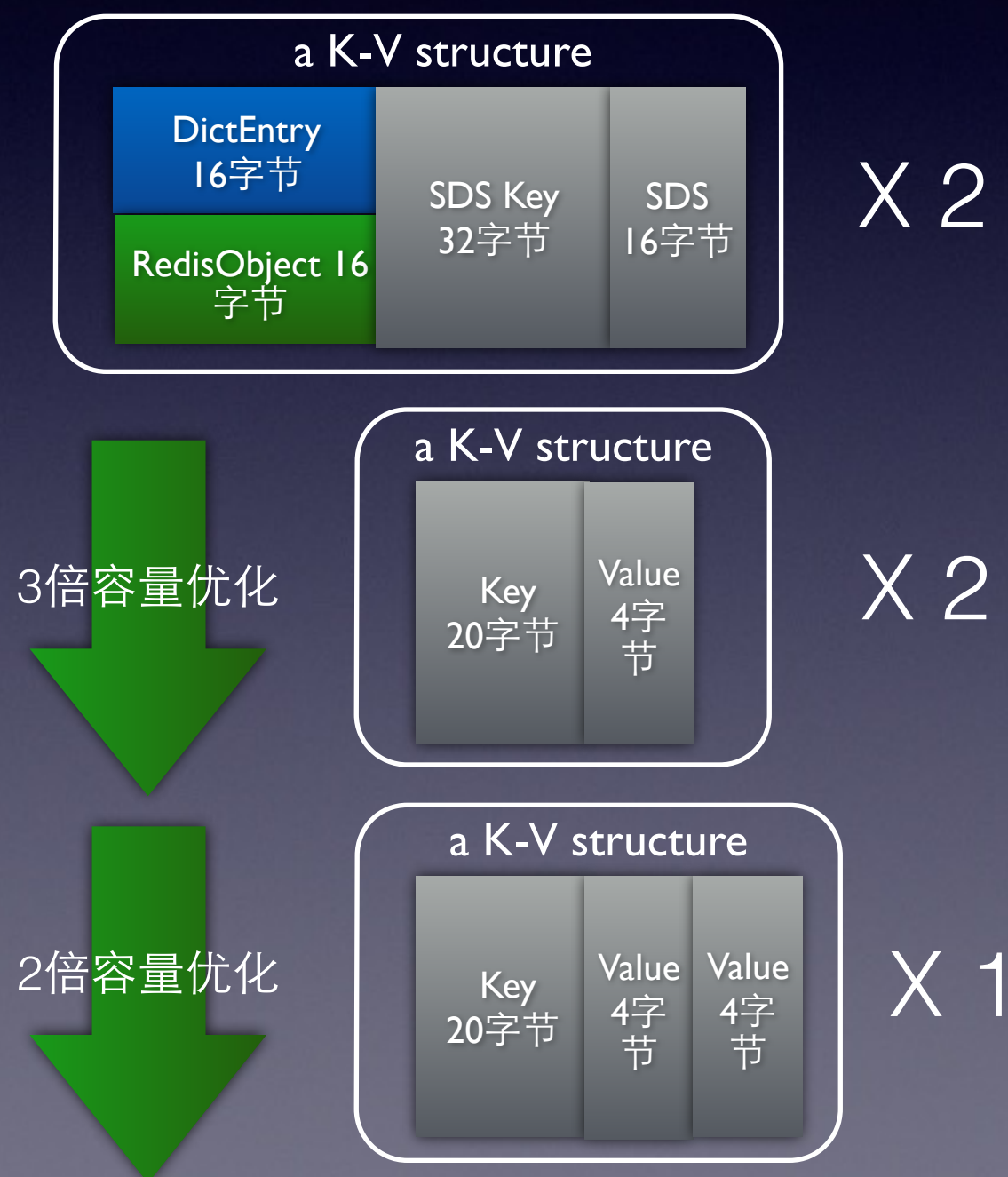
- 一主多从
- Slave故障自动摘除
- Master故障选主后闪恢复



Redis内存疯涨

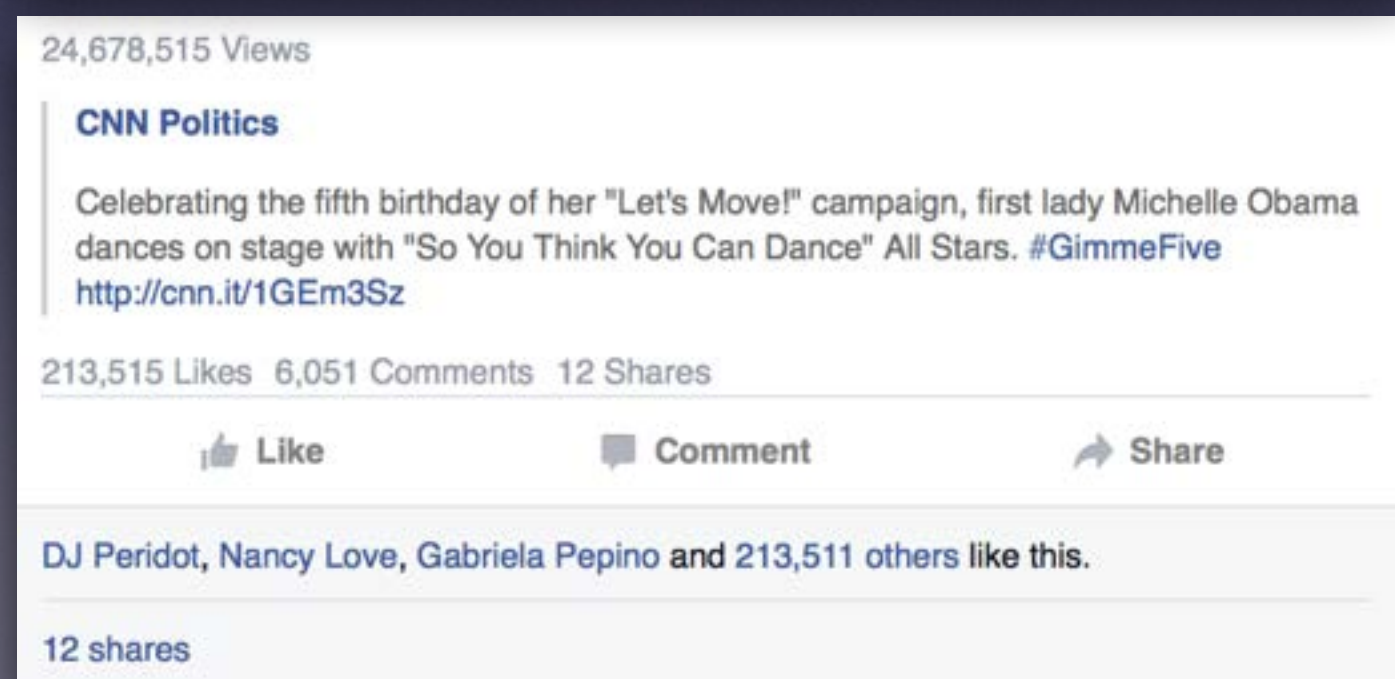
存储一个Key 20字节，两个Value 4字节计数

- Cache化改造
 - store:cache 9:1 -> 6:4
- 数据结构优化
 - Redisscounter
 - Counterservice



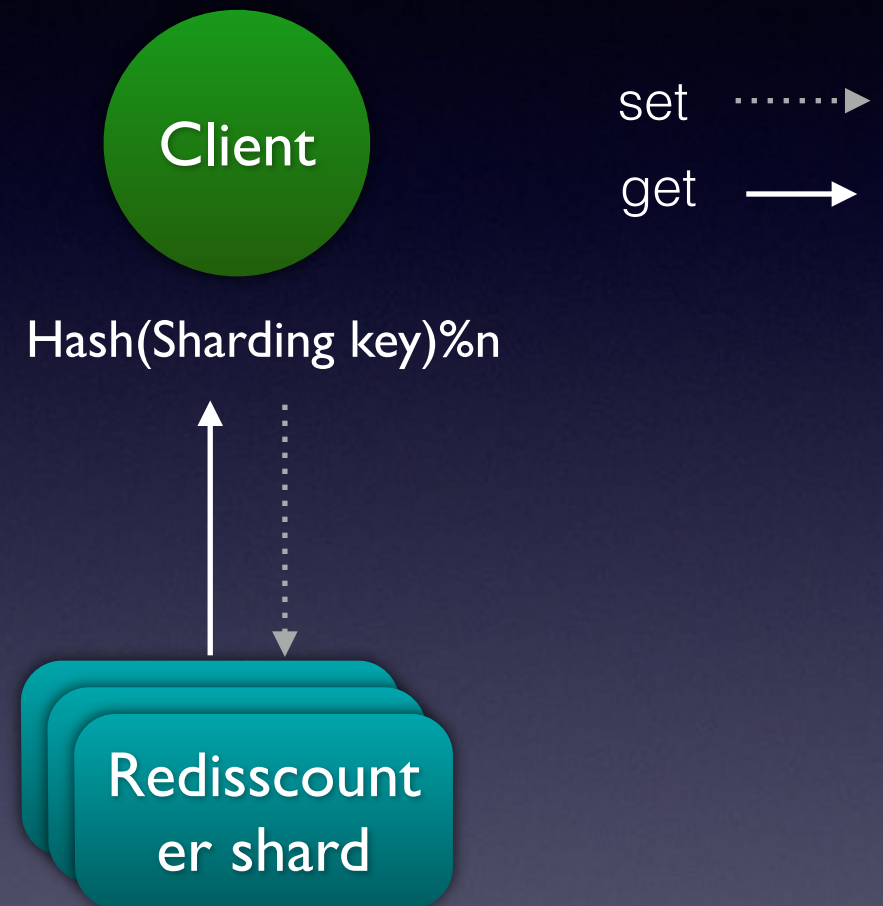
应用存储架构演进

- 计数服务存储演化



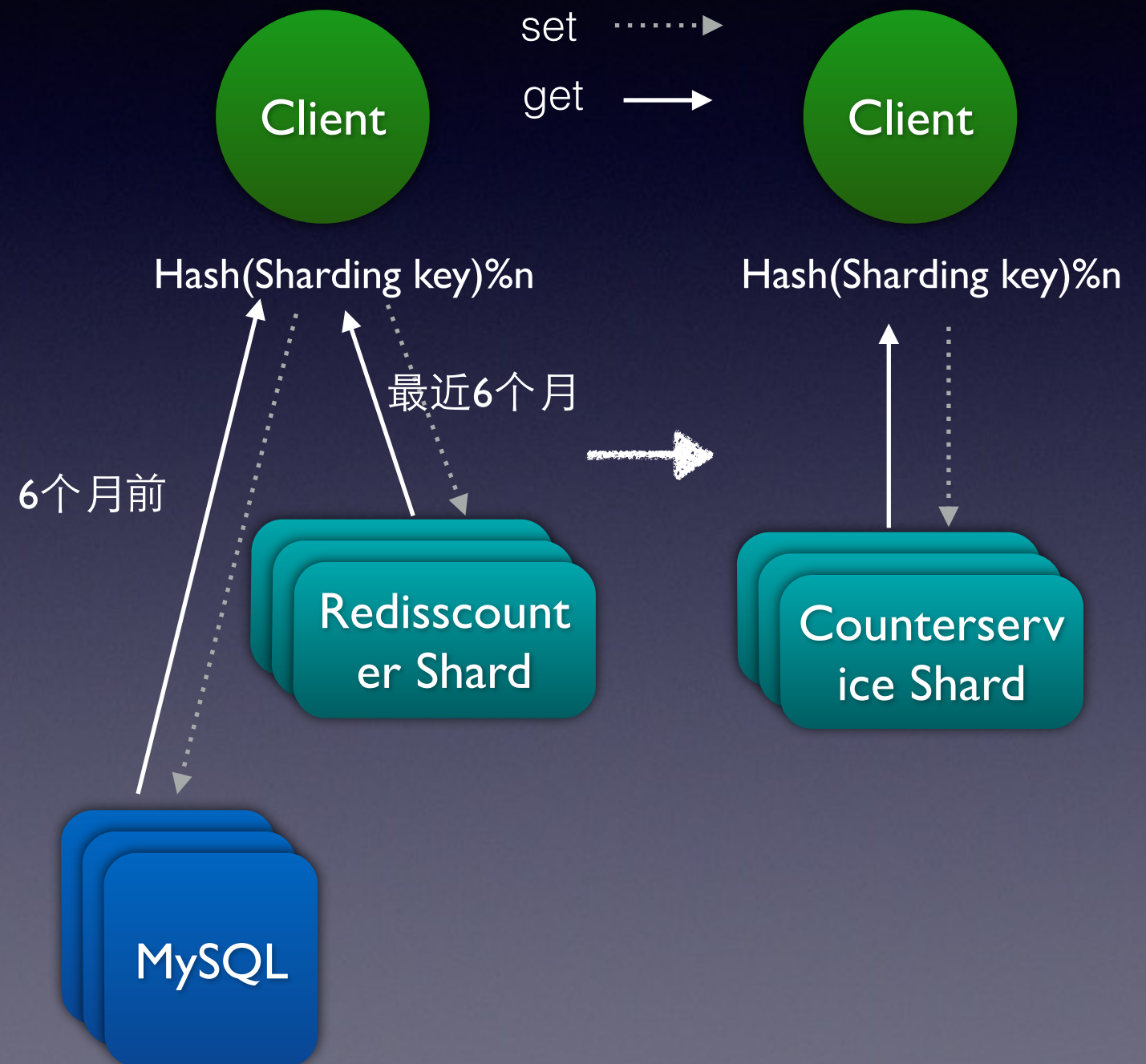
计数服务存储演化

- 十亿级计数
 - 用Redis存储1亿计数需要多大空间?
 - 用户纬度增长(e.g 用户关注数, 粉丝数, 微博数)



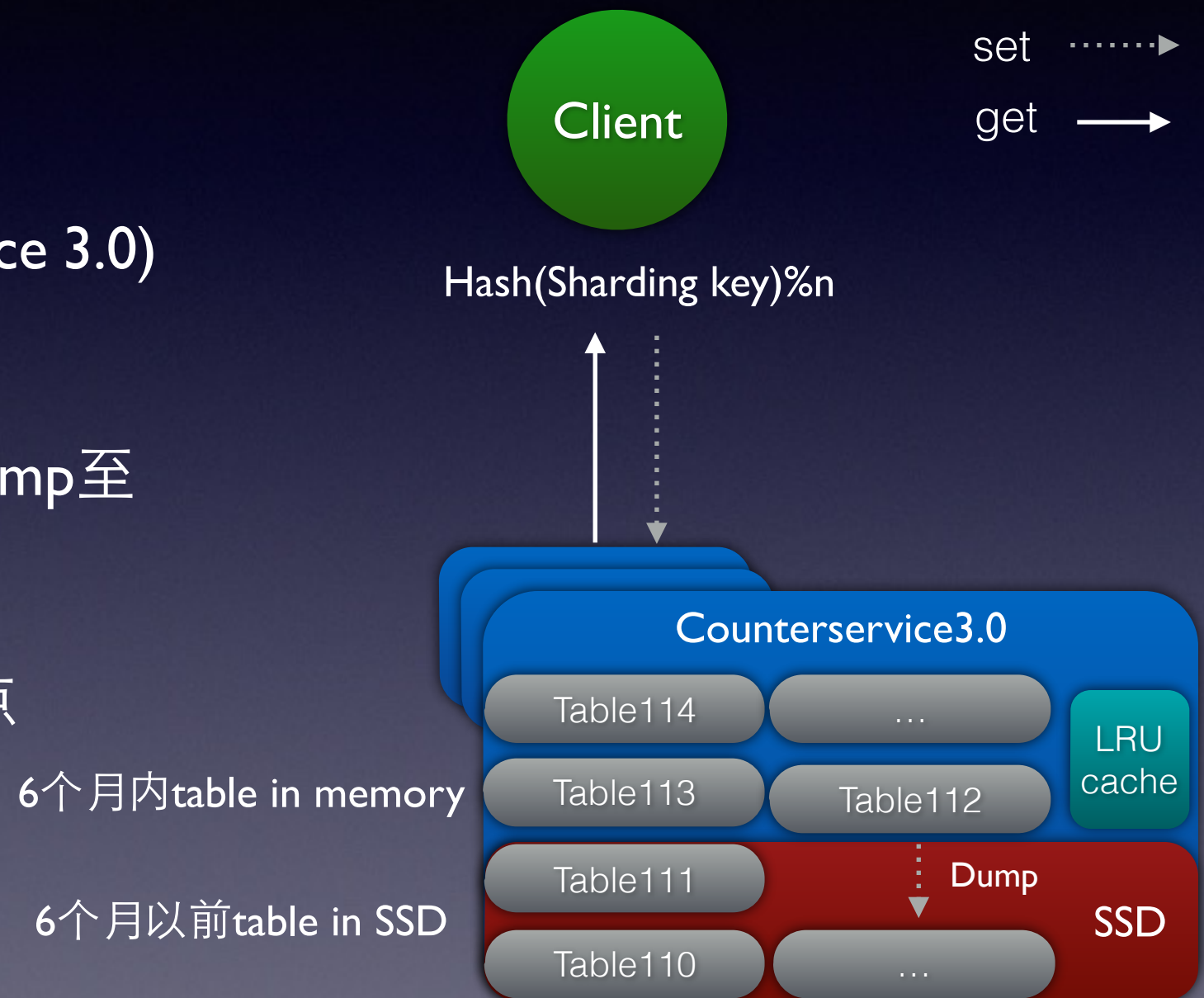
计数服务存储演化

- 百亿级计数 (Counterservice 2.0)
- 微博纬度增长(e.g 微博转发评论数)
- MySQL热点更新响应不够稳定



计数服务存储演化

- 千亿级计数 (Counterservice 3.0)
- 内存table写满后自动dump至SSD
- LRU cache防止历史热点



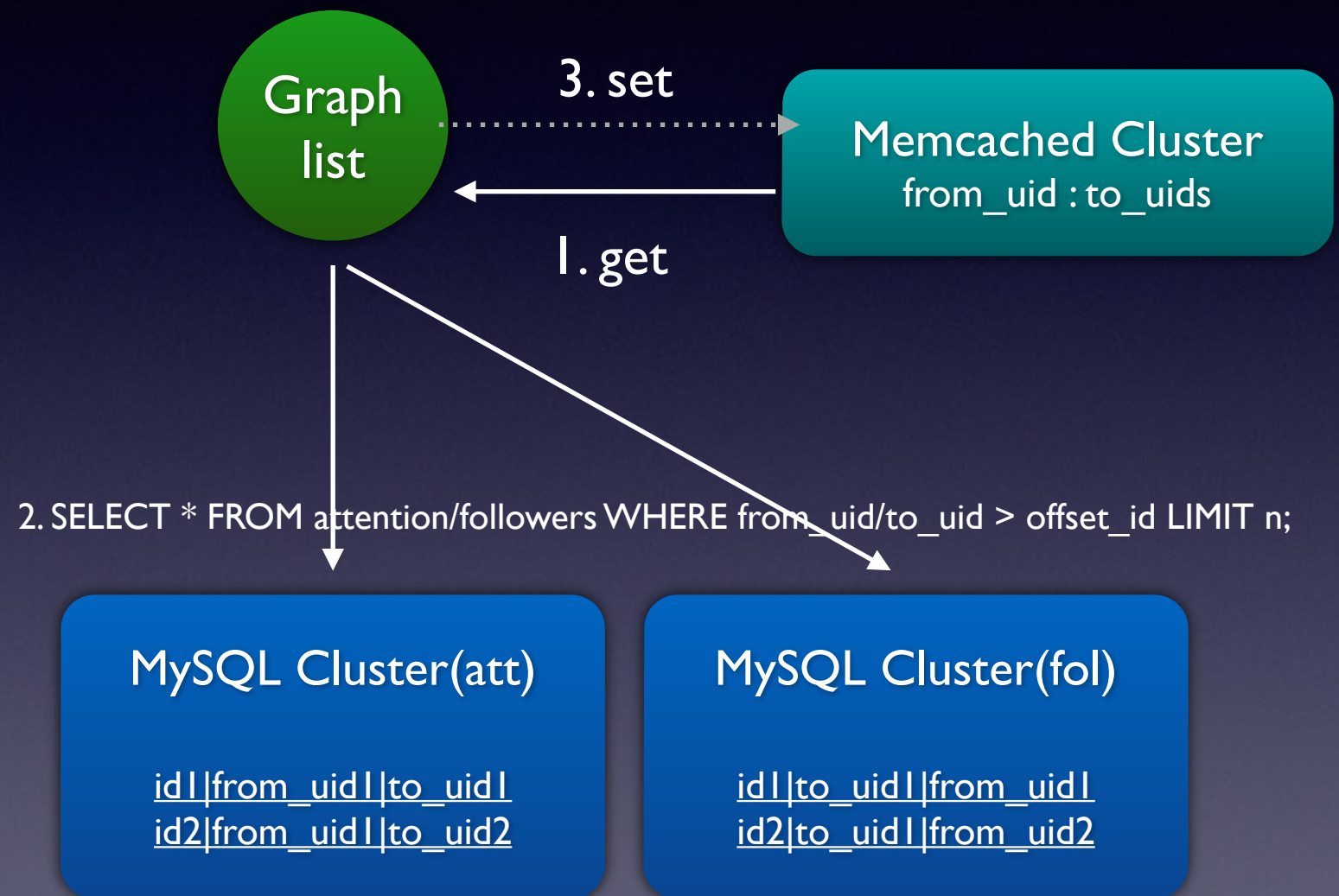
应用存储架构演进

- 社交图谱存储演化



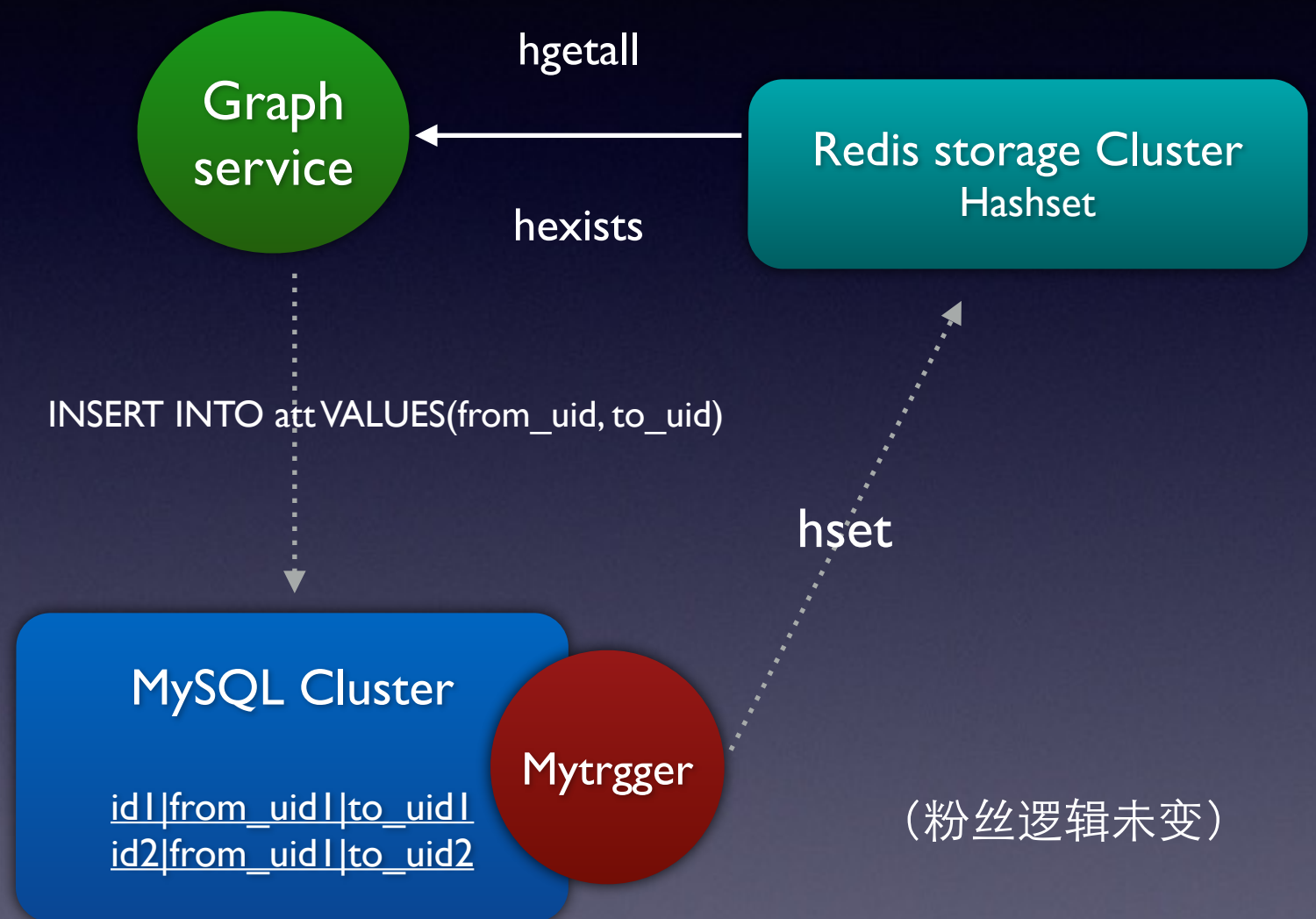
社交图谱存储演化

- 社交图谱 - 初级阶段
- graph_list
- attention/followers
- Memcached + MySQL



社交图谱存储演化

- 社交图谱 - 中期阶段
- check attention/followers
- Redis Hashset
- MySQL+Mytrigger+Redis



社交图谱存储演化

- 社交图谱 - 进阶

- 内存占用成本

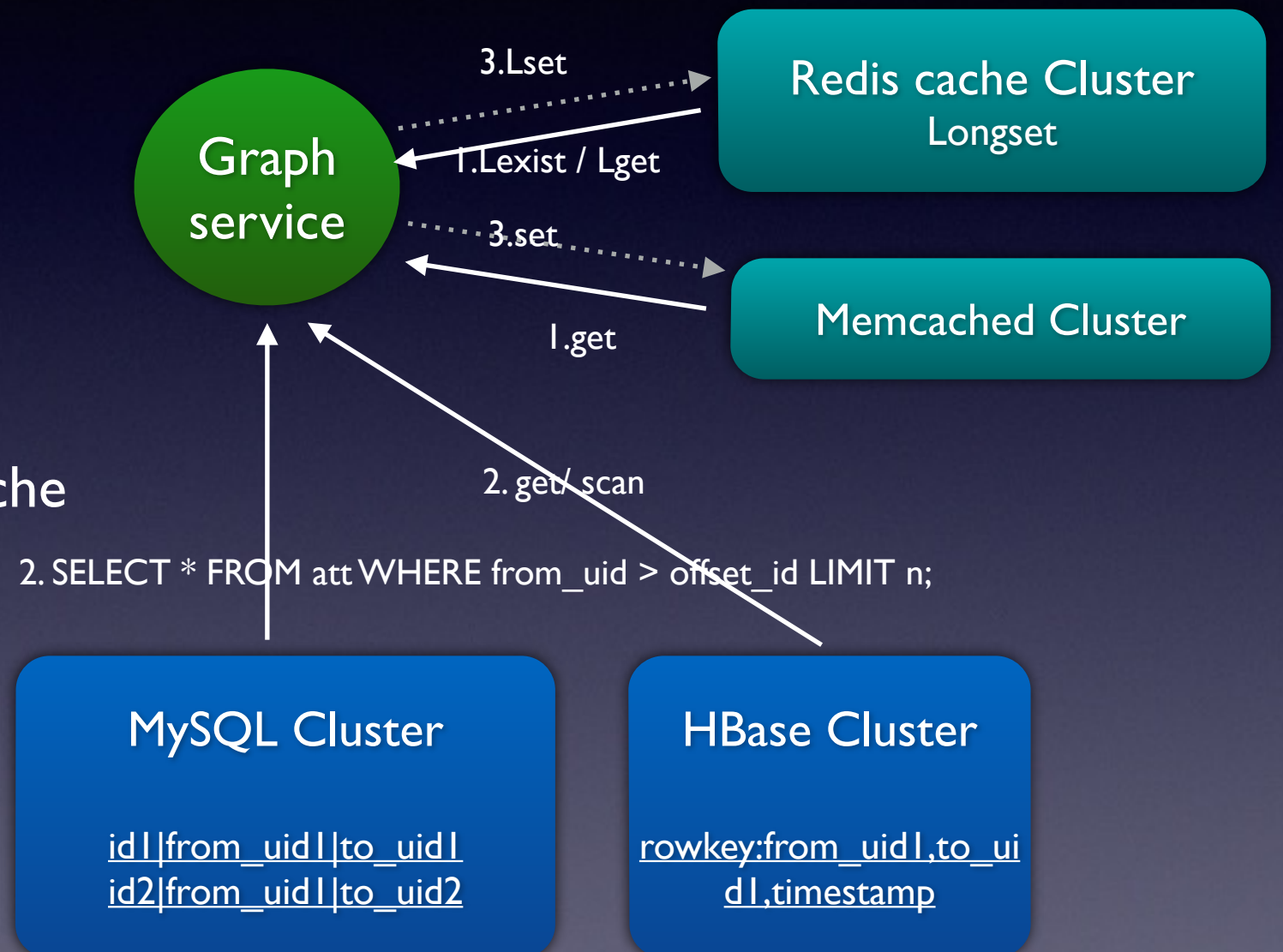
- Redis Storage到Redis Cache

- Hset性能瓶颈

- Longset

- 优化长尾存储

- HBase



社交图谱存储演化

- 未来计划 – 持续分级存储
 - 存储层冷热分离至HBase
 - 缓存层冷热分离至SSD

WORKED FINE IN DEV

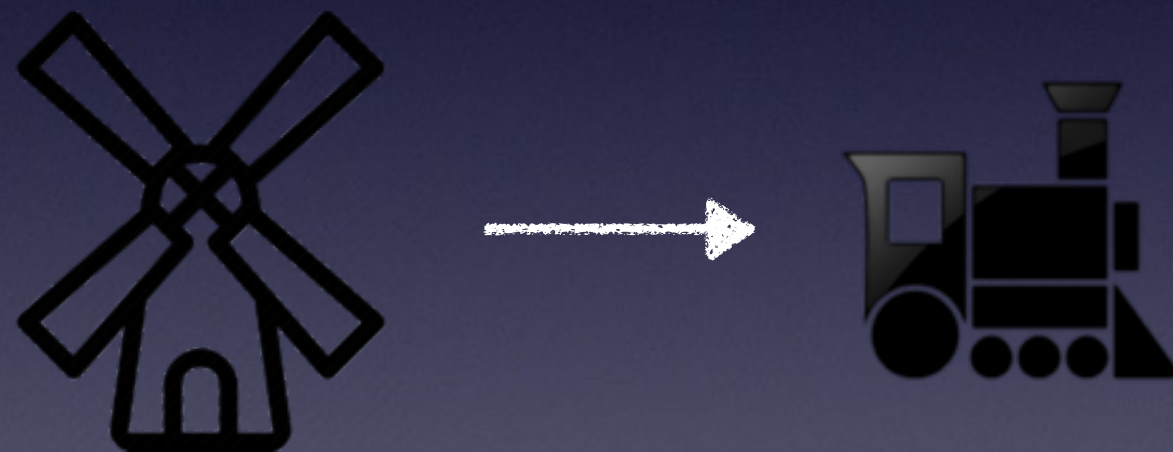
DBA

PROBLEM NOW

OPS系统更要给力...

OPS运维系统架构演进

- 小作坊运维
- 运维系统1.0
- 运维系统2.0



OPS运维系统架构演进

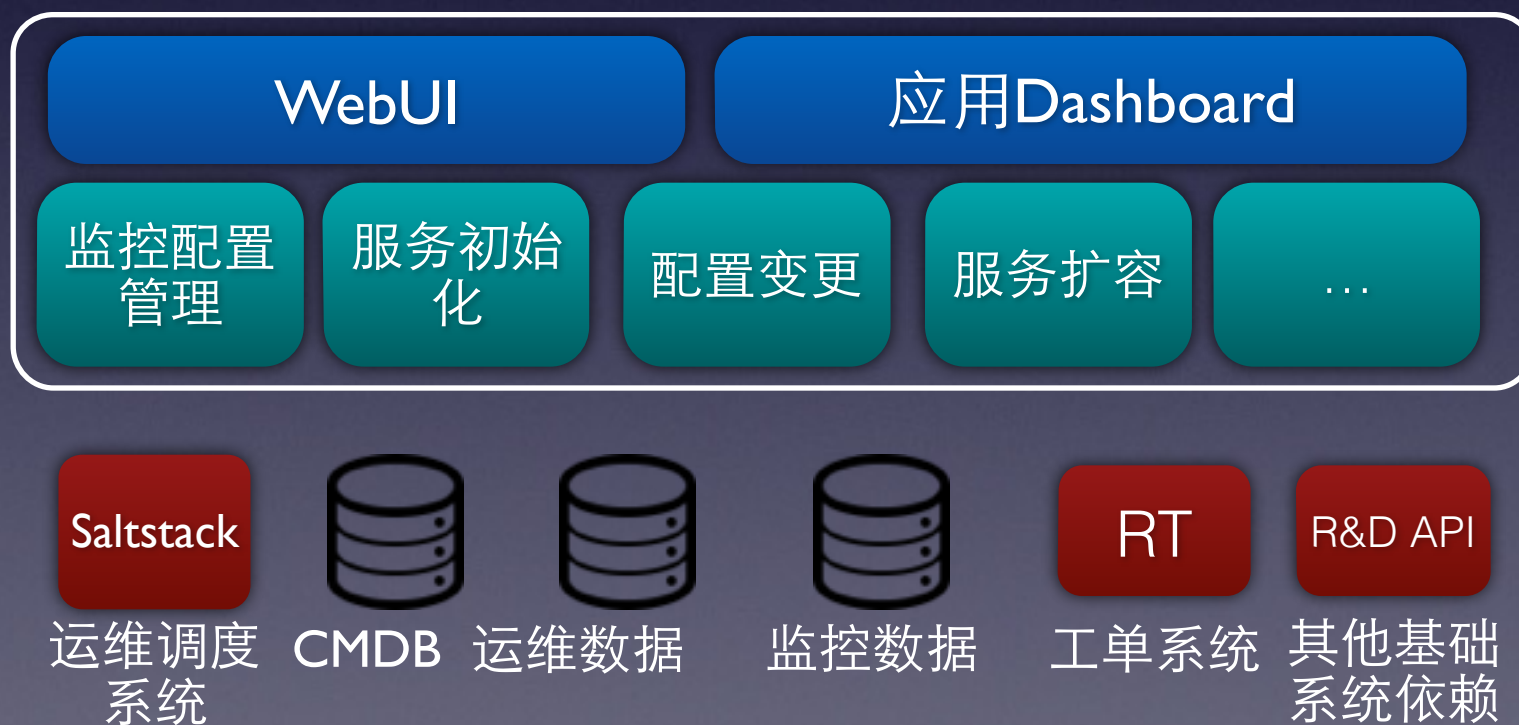
- 小作坊运维
 - 独立监控报警系统
 - 多套运维系统共存
 - 工单进展难以跟踪
 - 运维系统多语言开发
php python



OPS运维系统架构演进

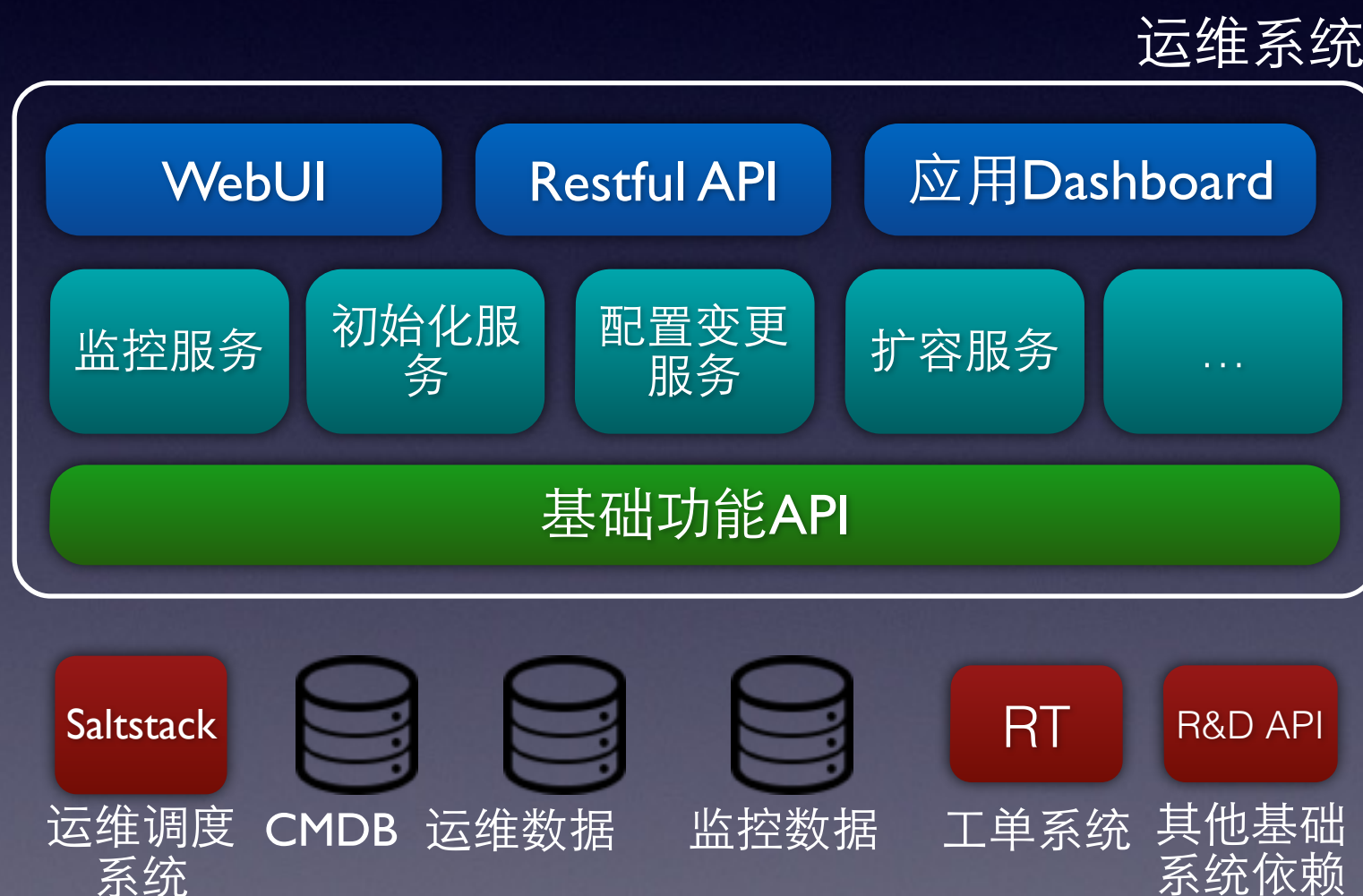
- 运维系统1.0
 - 统一中心管理
 - 单模块架构
 - 运维系统统一python

运维系统



OPS运维系统架构演进

- 工业时代2.0
 - 模块化，服务化，可持续迭代
 - 数据驱动



反思与总结

- 应用驱动，不幻想需求
- 没有银弹，避免滥用
- 架构尽量化繁为简
- 运维友好，保证服务生命力
- 分享，好的技术应该推广
- 服务意识，关注应用视角

反思与总结

- Think big, act small!

Q&A

We are Hiring!
recruiting-apac@appannie.com