



Alibaba  
technology  
Association

# 聚划算架构演化 & 系统优化

---

[dafeng@taobao.com](mailto:dafeng@taobao.com)



# Outline

- 聚划算业务介绍
- 系统架构变迁过程
- 研发实践优化
- 我们在路上

# 聚划算业务介绍



Alibaba  
technology Association

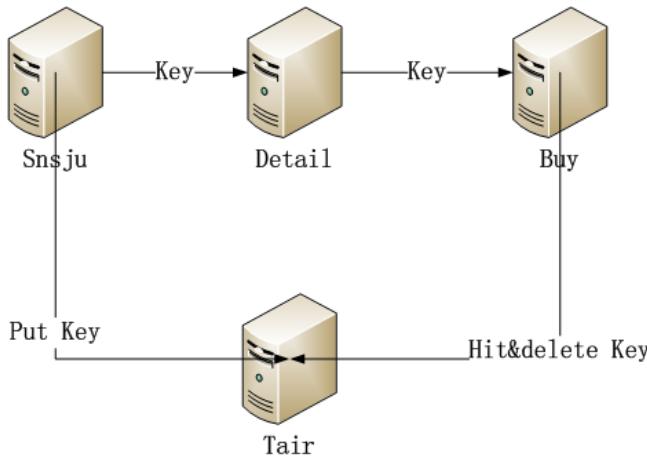
- 日均5kw的PV
- 日活跃用户600w
- 阿里百亿俱乐部
- 每日5k个热卖单品
- 日均订单50w笔以上
- 日均提供的远程调用1亿次

# 系统架构变迁过程



# 雏形期

- 架构特点
  - 单一应用
  - 分层结构，清晰
  - 新的功能快速迭代
  - 水平扩展能力



聚划算 淘宝团购

今日推荐 | 更多团购 | 精品回顾 | 我的聚划算 | 商家报名 | 城市招商

除甲醛空气净化器 ¥29.90

字母横领男士短袖T恤 ¥35.00

优雅显瘦女士连衣裙 ¥78.00

宝贝第一计划 宝宝款 ¥165.00

今日团购：【10:10开团】空气净化器，最低价只需306元包邮！(每个ID限购3件)[定金支付后不予退还]

更多团购>>

定金 付定金

¥29.90

原价: ¥699.00 当前价: ¥399.00 可低至: ¥306.00

62人已参团, 距最低价还差130人

¥499.00 ¥399.00 ¥306.00

0人 50人 200人以上

剩余时间: 9小时 5分 46.9秒

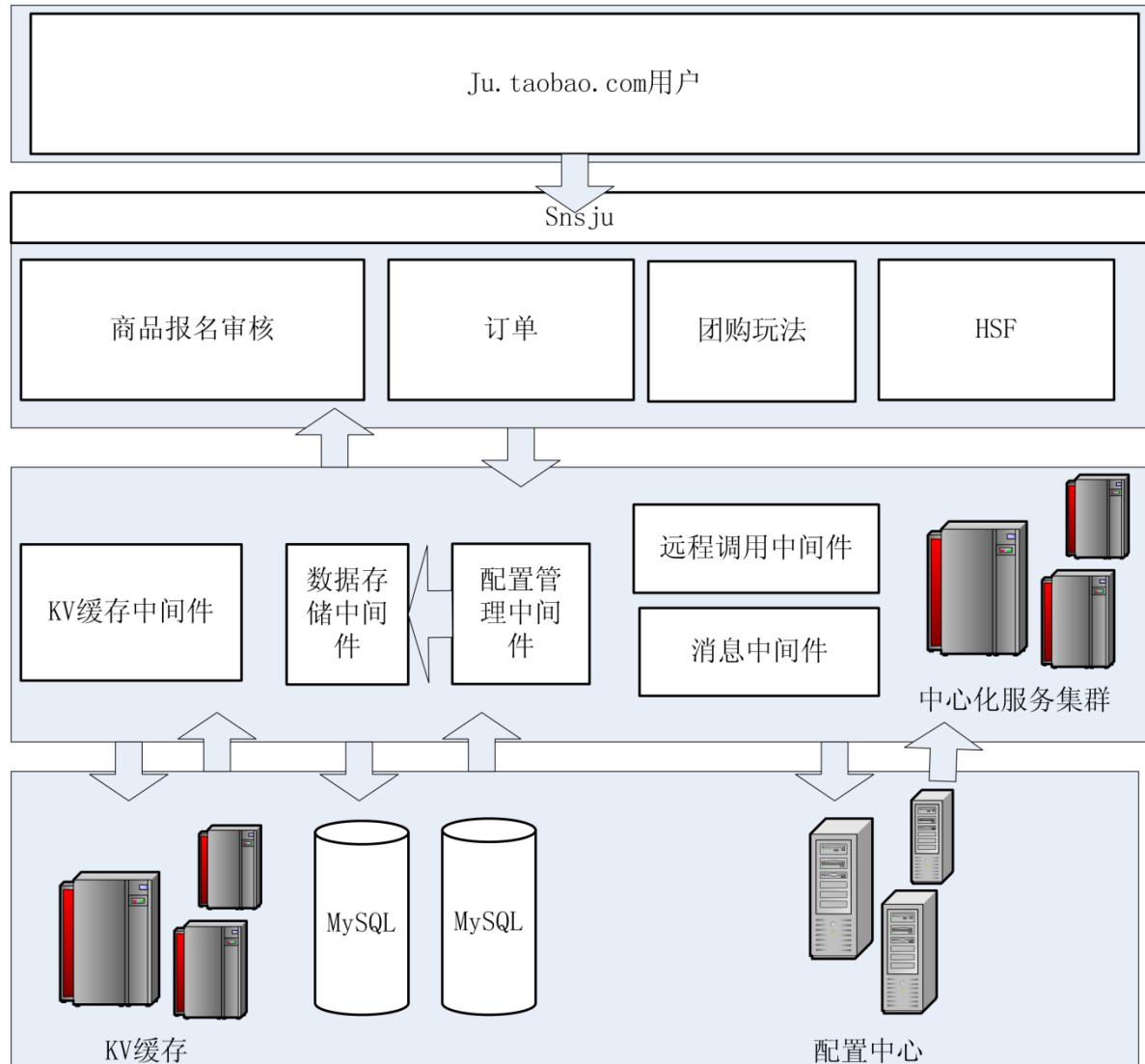
淘宝商城 www.tmall.com 品质保证

正品保障 提供发票 七天退换

分享到: 旺旺/QQ 微博 淘江湖 开心 人人 豆瓣 新浪微博

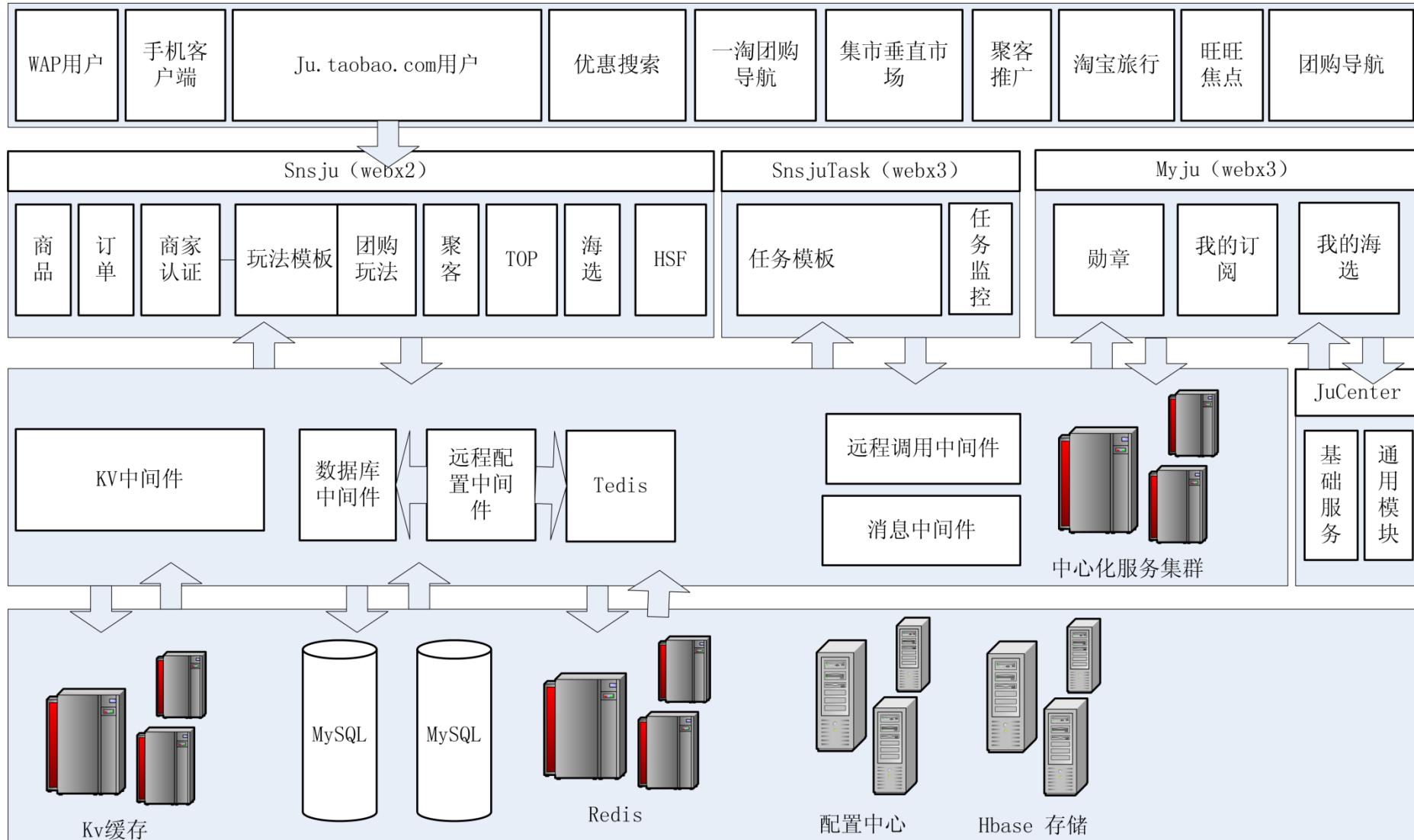
# 雏形期架构

- 遇到的问题
  - 流量尖峰
  - 商品数量陡增
  - 各种新业务模式





# 高速发展期架构





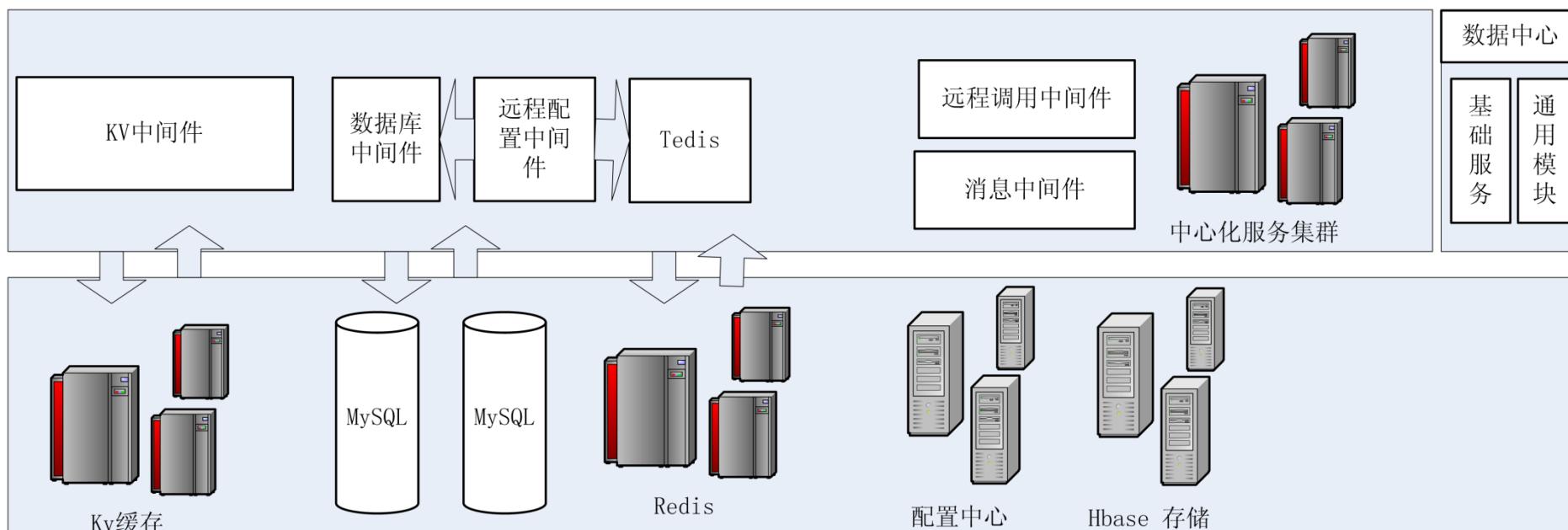
# 高速发展期

- 遇到的问题
  - 技术如何更好的支持快速迭代
  - 数据量，流量发展使得旧架构出现瓶颈
  - 团队扩张&合作

The screenshot shows a promotional page for dried fruit on the Taobao platform. The main banner features the text "聚划算 品质团购每一天 [杭州]" and "100%经典基因 SUPERJEANS 简约牛仔系列登场!". The product listing for "丝路宝典特级红马奶子带籽葡萄干2包 共440g (每个ID限购10件)" is displayed with a price of ¥19.80. The page includes a large image of the raisins, purchase statistics ("24116人已购买"), and a timer indicating remaining time. To the right, there are other product cards for "丝路宝典特级红马奶子带籽..." (¥19.80), "有机红枣特级若羌灰枣80g..." (¥93.90), and "百年吉多精选桂圆干1000g" (¥38.80). A "查看更多商品(232) + " button is also visible.



# 中心化发展期架构





- 遇到的问题
  - 应用如何拆分的大讨论
  - 依赖关系
  - 应用爆炸
  - 资源和效率

今日团购 > 吃喝玩乐 > 餐饮美食 聚新鲜 电视团

[杭州] [余杭区] [上越日本料理]: 双人套餐/刺身五品/寿司拼盘/日式炸大虾/脆皮鸡翅/蔬菜沙拉/加州手卷/鳗鱼手卷/味增汤! 仅售88元! 原价260元 精雕细琢, 色诱你的胃!

¥ 88.00 4.0 折 参团

原价 ¥ 260.00 节省 ¥ 172.00 订阅同类

已售: 1442 数量有限, 赶快下单吧

⌚ 13:12:52.2

分享按钮: 微信 QQ 淘 人人 开

本团购参加店铺满就送活动  
单笔订单满118元, 免运费  
满118元不免邮地区: 内蒙古;甘肃;青海;宁夏;新疆;西藏;台湾;香港;澳门;海外

券 ¥ ☺ 24



# 演化总结

- 服务化中心化趋势
- 解耦强依赖
- 公共组件的抽象
- 合理使用不同层次的缓存
- 警惕缓存命中率



# 研发实践优化

- 让缓存有序&结构化
- 大对象，大列表
- 热点数据搜索
- 分布式任务系统
- 数据分布式复制
- 缓存占空
- 数据异步化写入
- PageCache

# 场景:结构化缓存

- 用户最近浏览宝贝
- “十分热卖”
- 交易排行榜





- 什么是tedis?

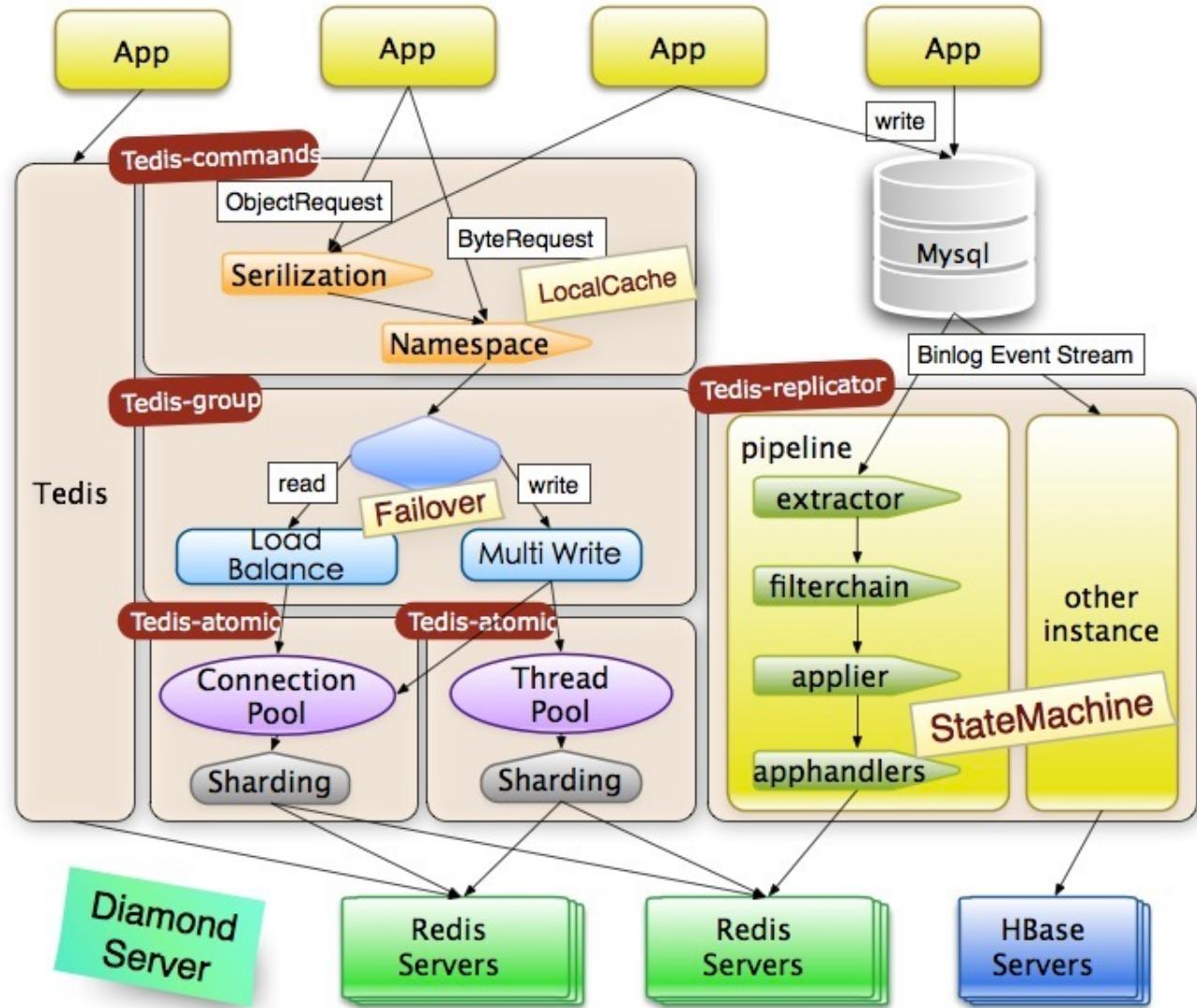
- 高性能内存数据库Redis的一个客户端；或者将Tedis理解为淘宝在小量数据的高性能存储、操作和运算等场景的解决方案，目前正在聚划算团队率先实践。

- tedis做了什么？

- tedis是对开源redis的java客户端jedis的封装，在jedis的基础上我们封装了一套更易于使用的byte api接口和object api接口，同时做好了默认的object序列化工作。在部署上，目前采用master-master结构，实现多写和随机读机制，既每个写请求会发到所有服务器，每个读请求随机选取一个服务器，当在某台服务器读失败后，将该台机器加到重试队列中，直到该服务器恢复正常，客户端的请求才会重新到该服务器。

# Tedis

- 系统结构





- **redis适合做什么？**

- 在主页中显示最新的项目列表
- 删除和过滤
- 排行榜及相关问题
- 按照用户投票和时间排序
- 过期项目处理
- 计数
- 特定时间内的特定项目
- 实时分析正在发生的情况，用于数据统计与防止垃圾邮件等
- Pub/Sub
- 队列
- 缓存

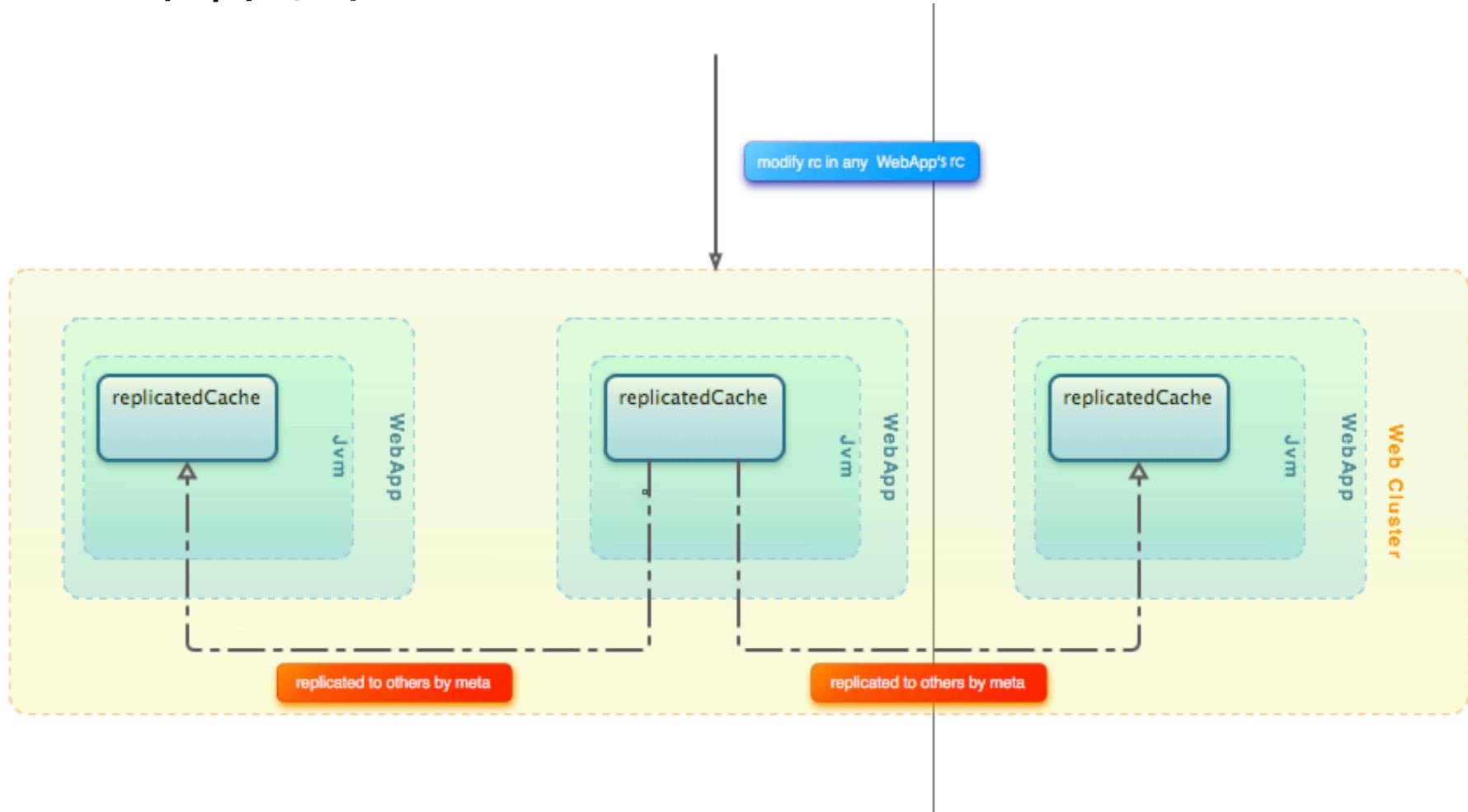


- 列表很大怎么办？
- 不能忽视的序列化和反序列化开销
- 一个经常使用的大数据结构
  - Ip库
  - 类目库
  - 城市库



- 什么是RC?
  - 基于jvm 本地缓存的实时同步解决方案，解决了内存业务对象集合在集群范围内不同服务器之间的一致性问题。
- RC做了什么?
  - 在应用访问集群中任意一台机器的时候.对其的本地缓存进行操作，本地缓存在修改的同时给广播中心发送消息。广播组件通过获得所有加入该组的机器列表给所有机器发一个消息通知所以机器都需要进行更新。其它机器在接收到消息以后做同样的操作。进行更新本地缓存

- 部署结构





- RC适合做什么?
  - 频繁读取的稳定数据
  - 集群环境下配置同步
  - 热点数据大规模读取
  - 无法忽视kv缓存的网络以及序列化开销场景



# 场景：热点查询

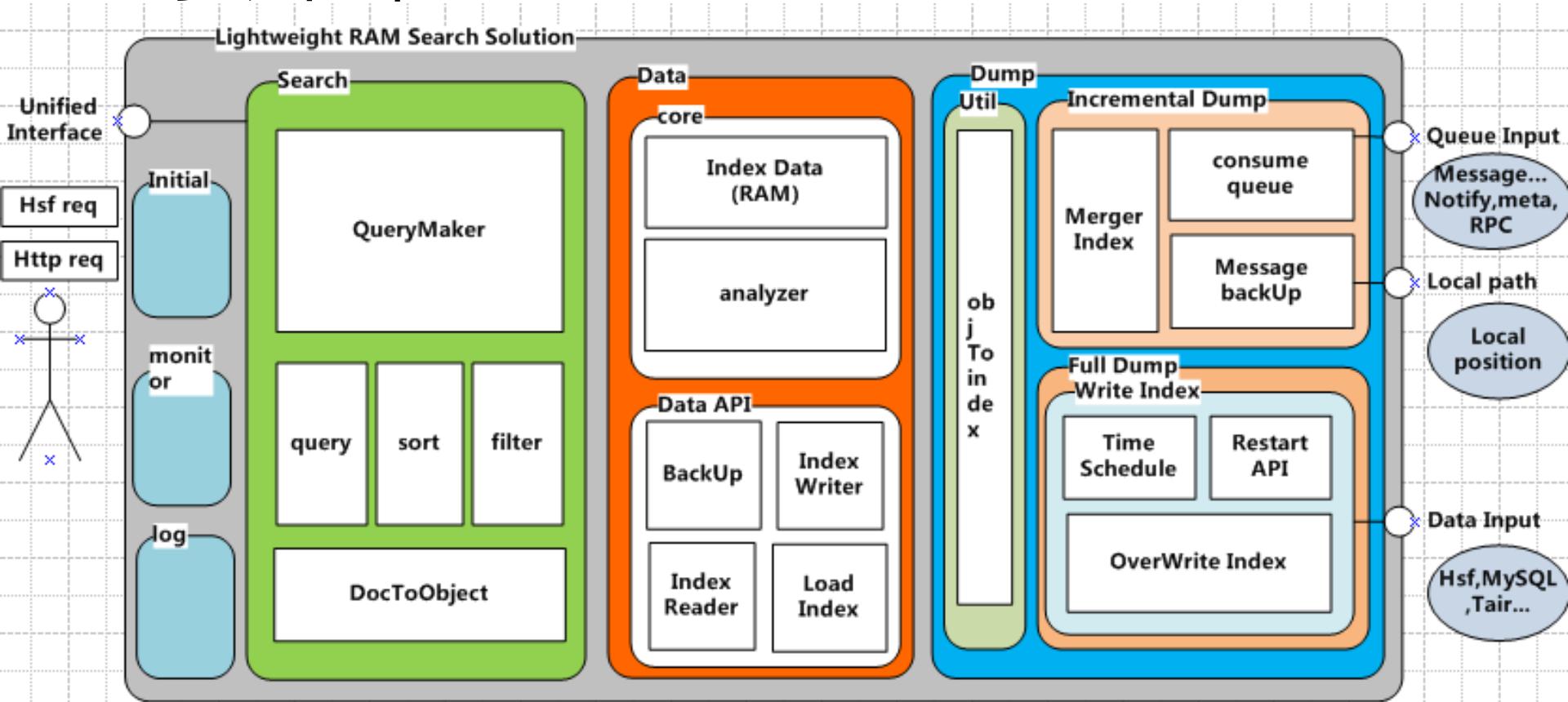
- 针对内存数据的搜索
- 数据热点集中，查询场景多样化

The screenshot shows a user interface for an e-commerce platform. At the top, there's a navigation bar with categories: '杭州生活 (34)' (highlighted in red), '全部', '餐饮美食', '休闲娱乐', '旅游酒店', '生鲜果蔬', '生活服务', and '车房'. Below this is another row with '商品团购 (134)' (highlighted in red), '全部', and various sub-categories: '服饰', '时尚', '电视', '食品', '鞋包', '母婴', '居家', '聚名品', and '聚家装'. At the bottom of the interface, there are several sorting and filtering options: '默认', '销量 ↓', '价格 ↑', '折扣 ↓', '服务保障' (highlighted in red), '价格区间', and '只看明日聚透'.



- 什么是Laser?
  - 基于lucene的本地搜索容器，是针对中小规模数据量，大流量读，小流量更新的搜索解决方案。
- Laser做了什么?
  - Laser主要是针对索引的创建，更新，查询操作的封装，整个容器采用内存索引，最大限度的提高查询的效率，通过全量和增量模块保持数据的实时，同时针对容器重启情况下的数据预热，内存监控，以及诸多提高dump&query开发效率的工具包。

## • 系统框架



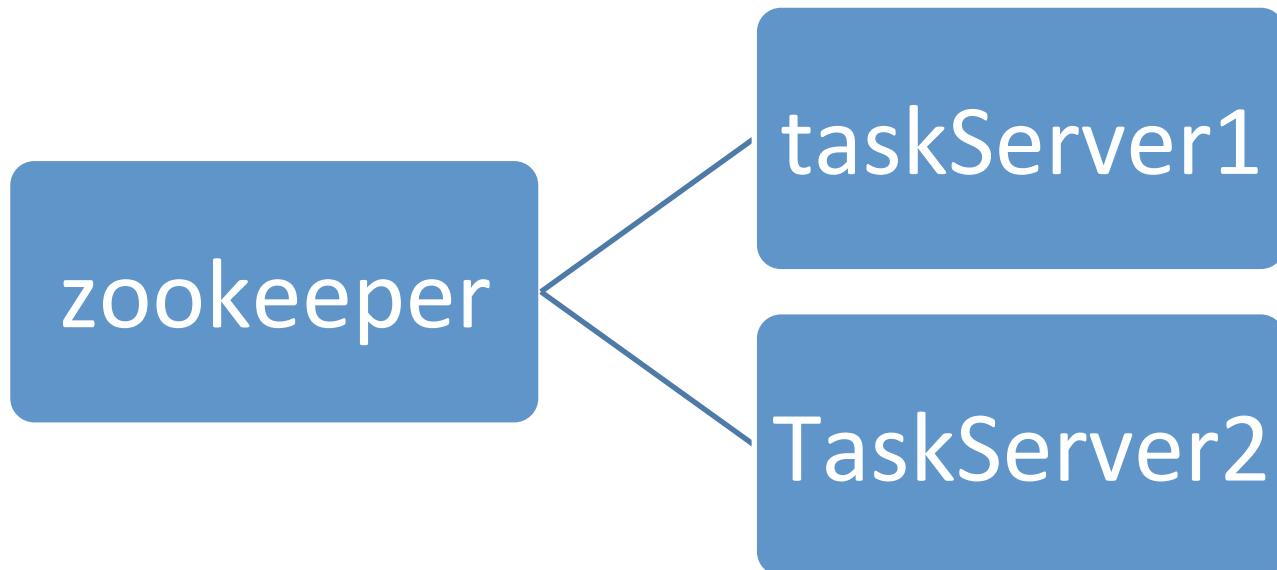


# Laser

- Laser适用的场景
  - 热点数据的读取查询
  - Kv缓存无法适应的大列表展示
  - 针对lucene索引对象的查询客户端



- 让任务系统在分布式环境跑起来

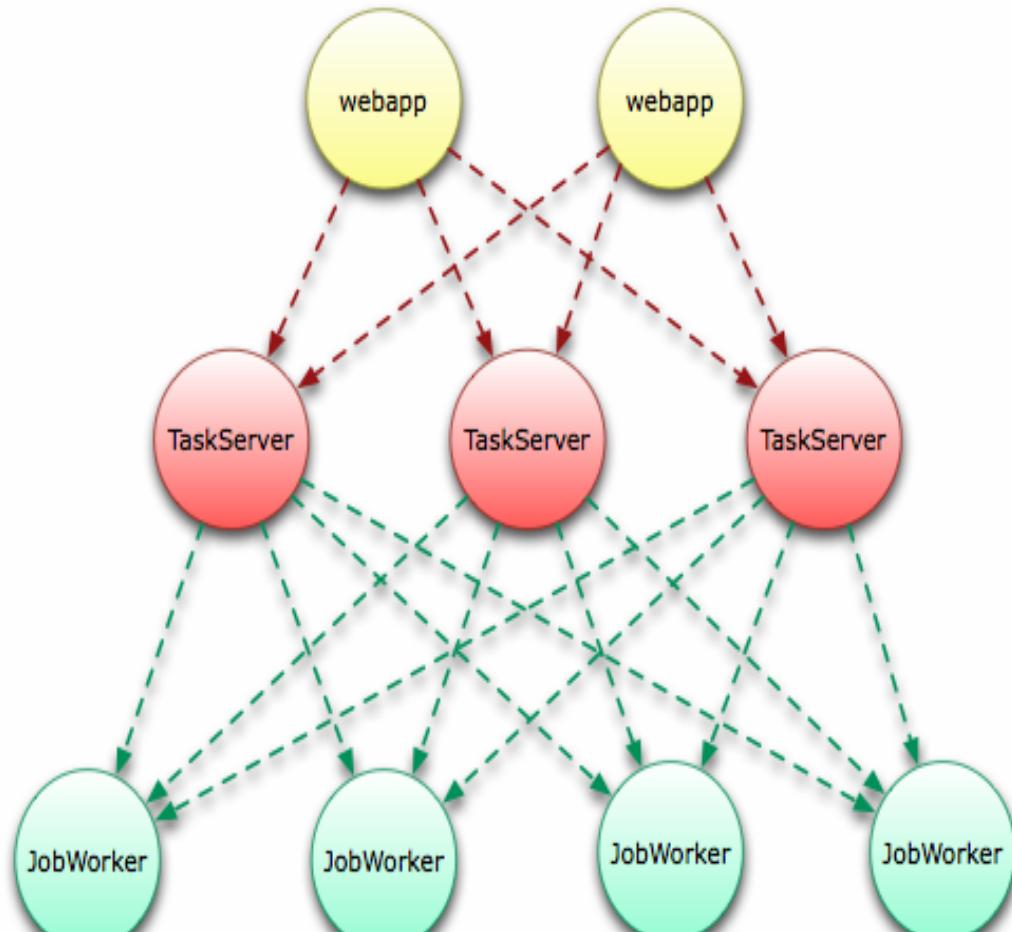




- Yardman是什么?
  - yardman是一个分布式的任务系统。具体的说是由任务管理，任务调度，任务执行3个系统组成的集群环境下，分布式多任务协调系统。
- Yardman做了什么?
  - 通过ops页面，yardman获得需要执行的任务，以及对于的任务执行程序，TaskServer按照配置执行任务触发，停止等调度。JobWorker通过拉机制实时去taskServer获取需要执行的job。扩展jobWorker就可以实现水平扩展

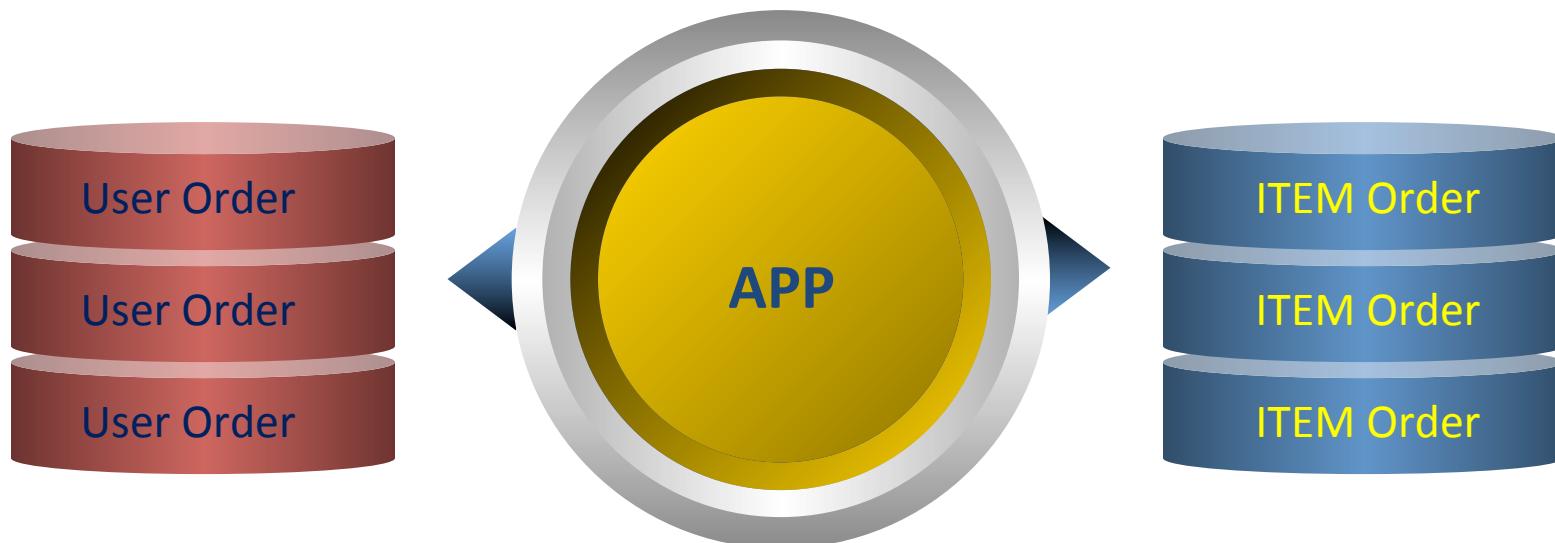
- 主要目标：将任务部署在不同的机器上运行

任务类型:	日常任务
机器分组选择:	
程序类型:	java程序
程序名称:	*
运行参数:	
责任人:	*
部门:	*
任务优先级:	<input checked="" type="radio"/> 中级 <input type="radio"/> 低级 <input type="radio"/> 高级
调度方式:	<input checked="" type="radio"/> 定时调度 <input type="radio"/> 循环执行 <input type="radio"/> 手动执行 <input type="radio"/> Expression
定时规则:	运行周期: 每个小时 启动时间: 0 00:00:00 运行次数: 1 结束时间: 任务结束时间, 任务
任务描述:	



# 场景：跨库数据复制

- 2张不同纬度的分表
- 不影响性能，实现分库
- 最终一致性



# 基于Binlog的跨库事务

- 优点：
  - A. 和业务隔离，侵入性小。
  - B. 对db影响较小。
- 缺点：

小于等待时间的增减操作会导致暂时的不一致

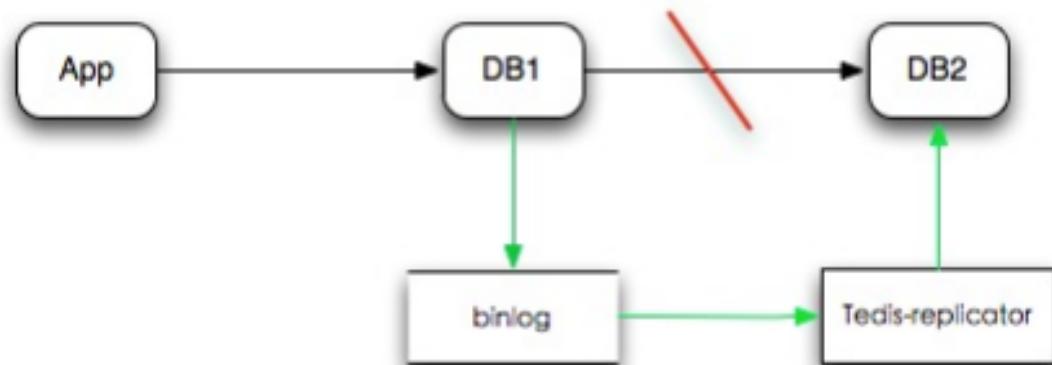
正常流程：



异常流程：（插入DB2数据失败）

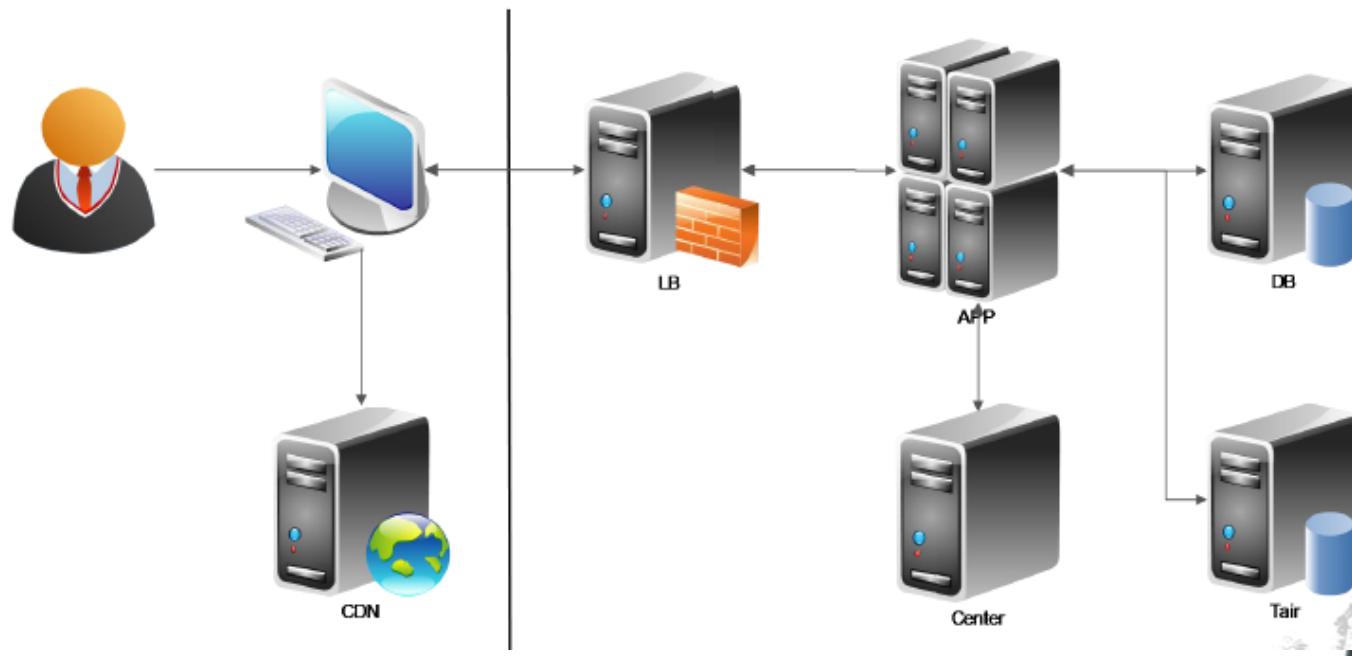


补偿措施：

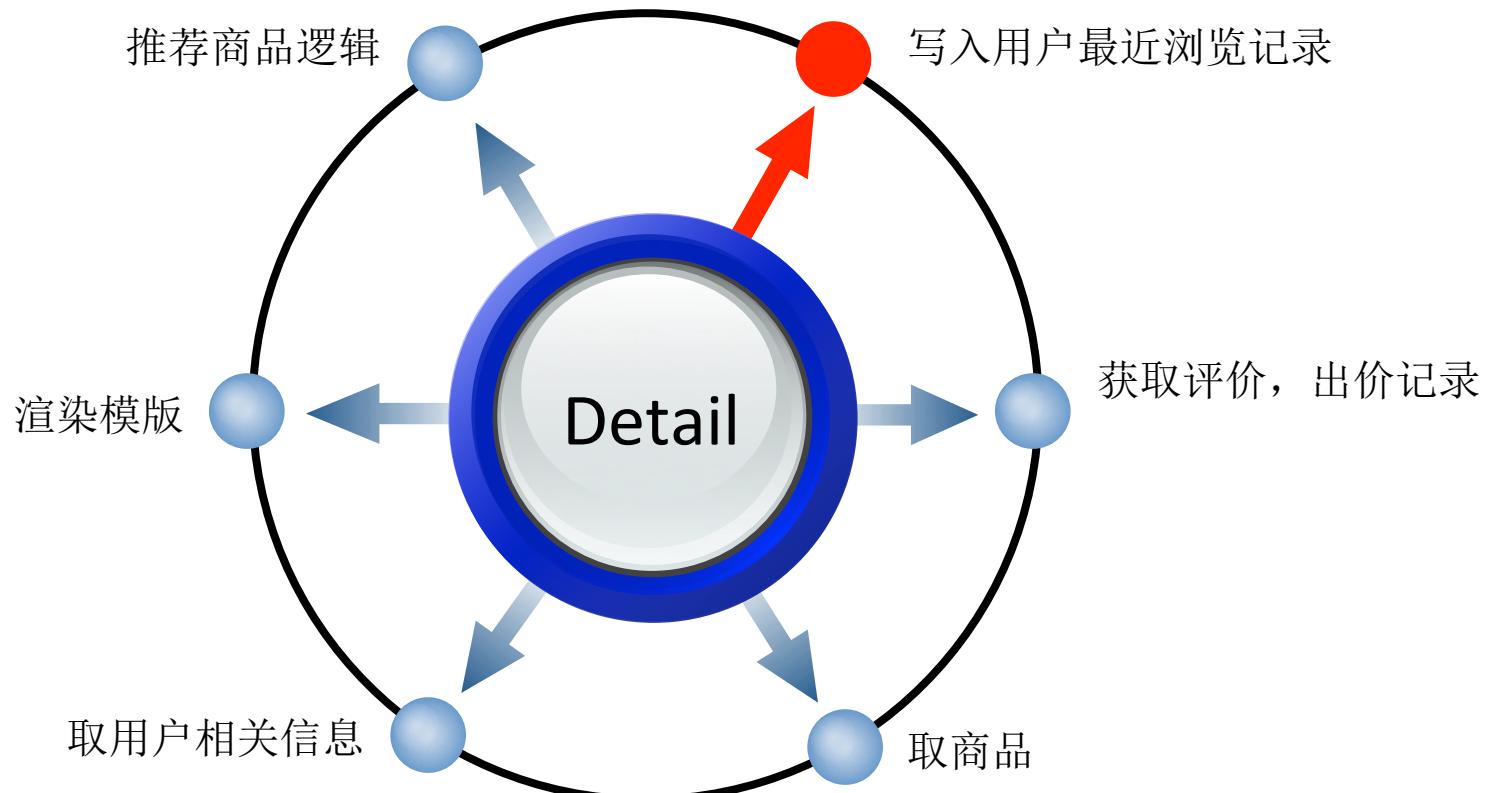


# 场景：缓存被击穿

- 网络攻击访问不存在的数据
- 区分不存在数据&未缓存数据

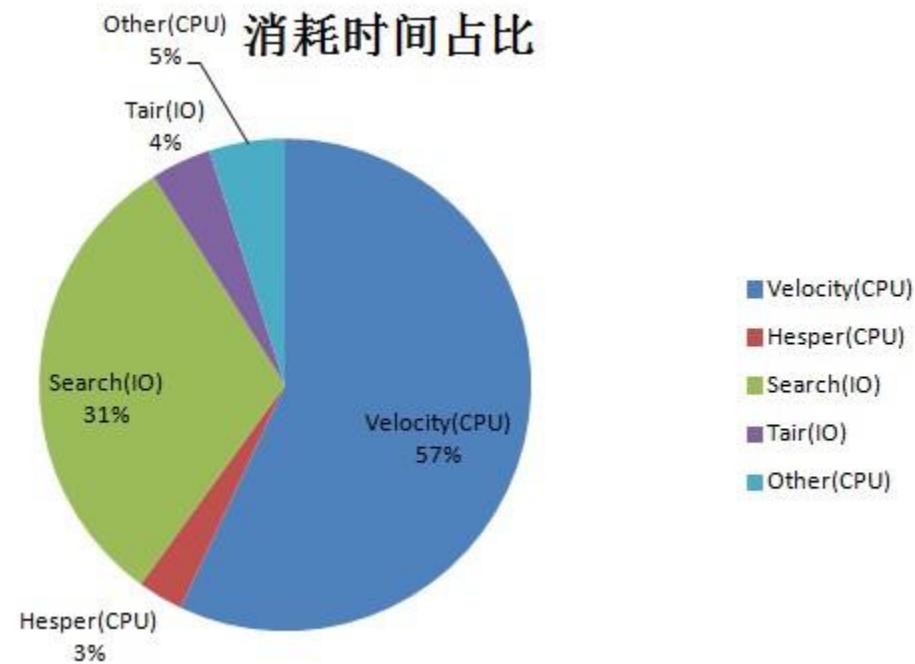
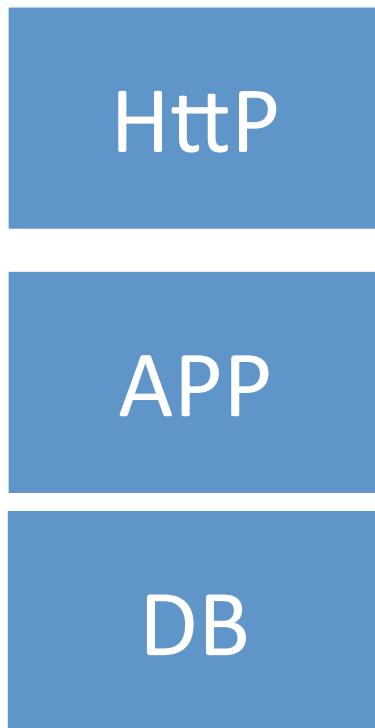


- 让请求快起来



# 场景：继续提升

- 如何让我们的QPS更高？
- 系统资源消耗在哪里？





- 应用拆分颗粒度问题，分久必合
- 虚拟机OR实体机？
- 统一的数据中心
- 应用云的理想