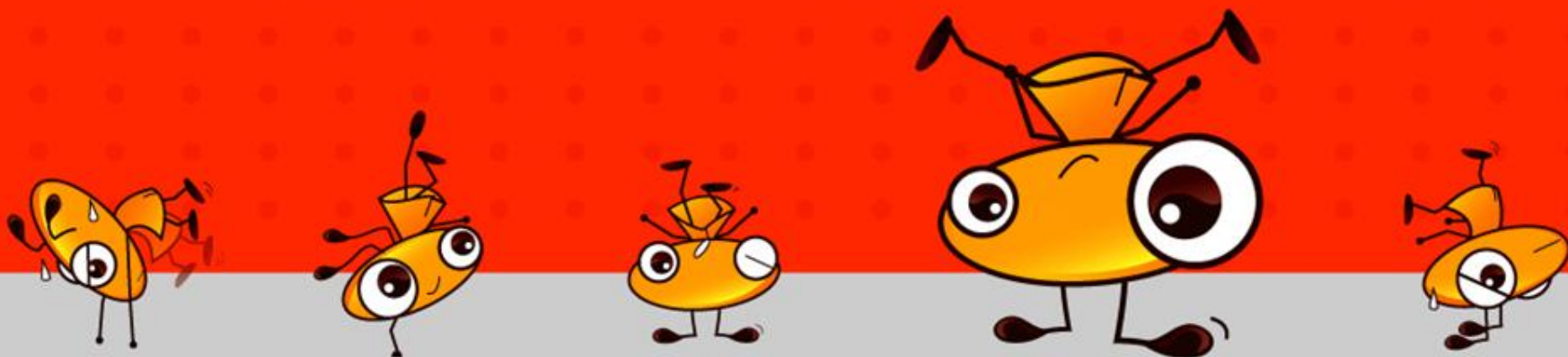


# 淘宝系统架构介绍

黄裳  
2009.10



# 互联网公司技术架构系列资料

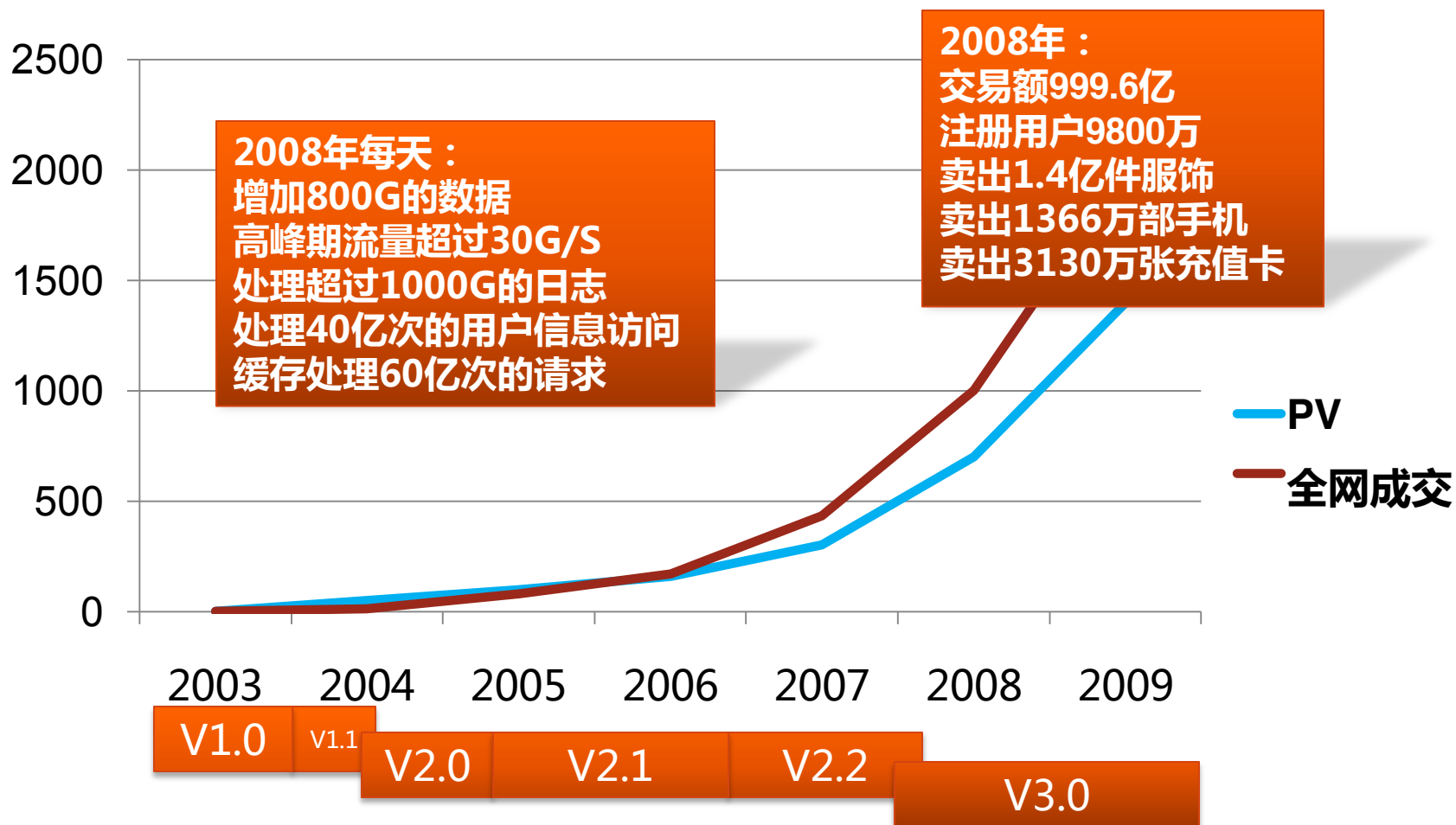


为您悉心整理

/\* 让工作重新关于成长和成就、关于快乐和分享、关于梦想和荣光 \*/



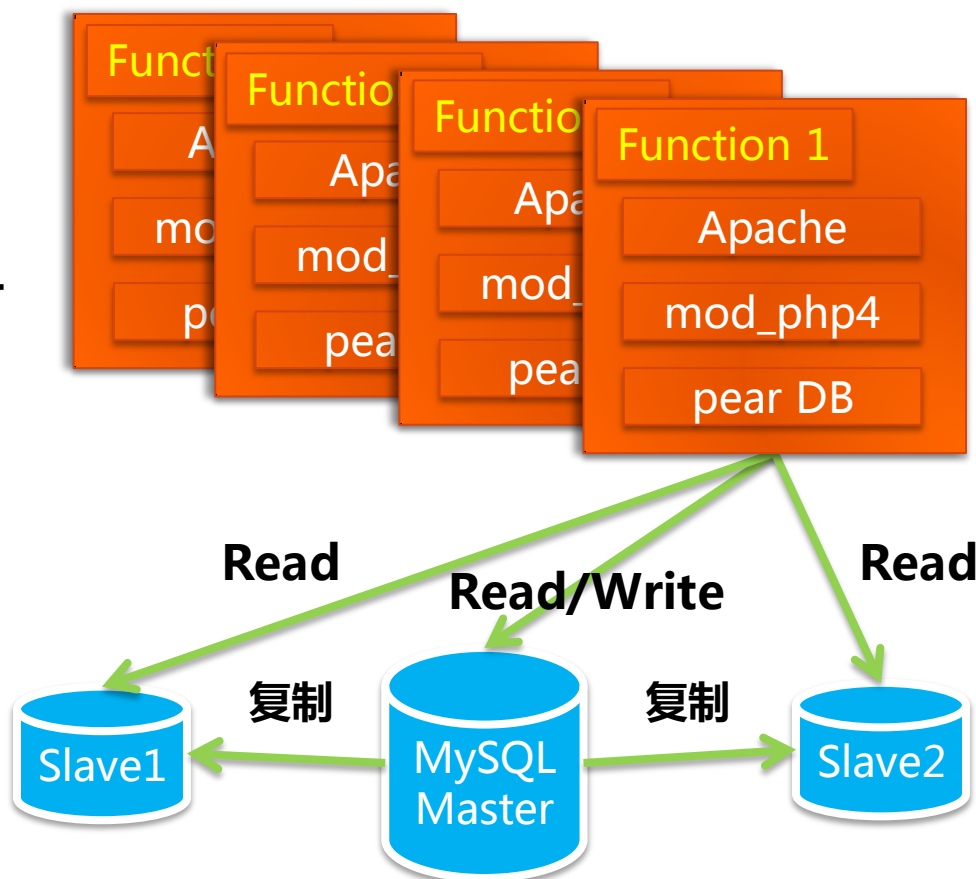
# 淘宝发展历程





# V1.0 2003.5 – 2004.1

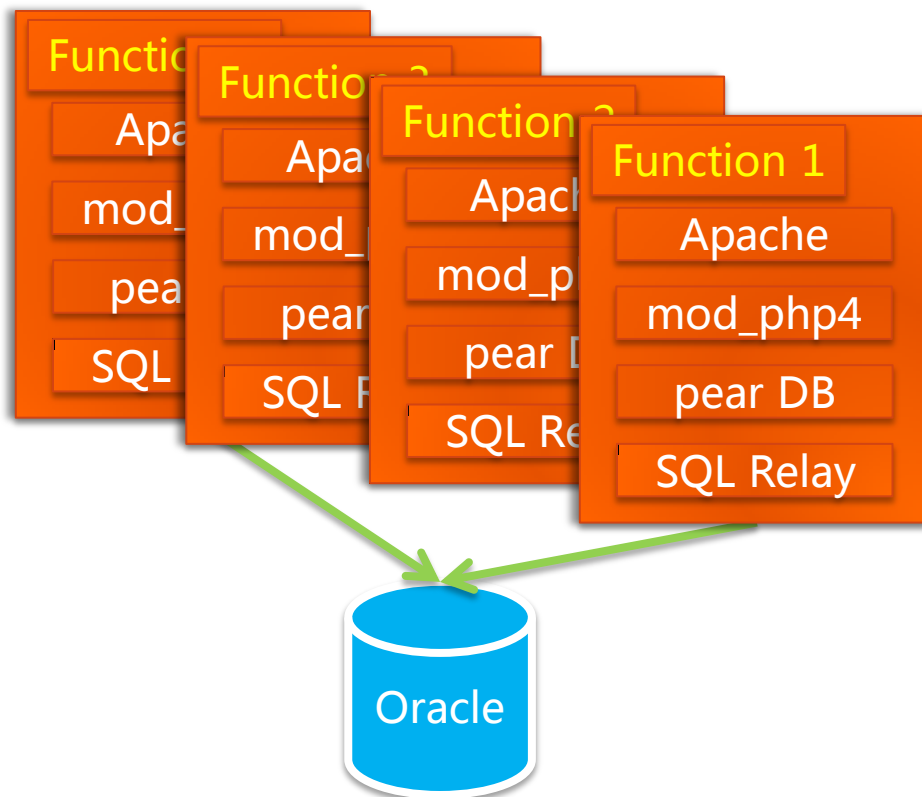
- 非典时期
- 马云住宅
- LAMP
- MySQL读写分离





# V1.1 2004.1 – 2004.5

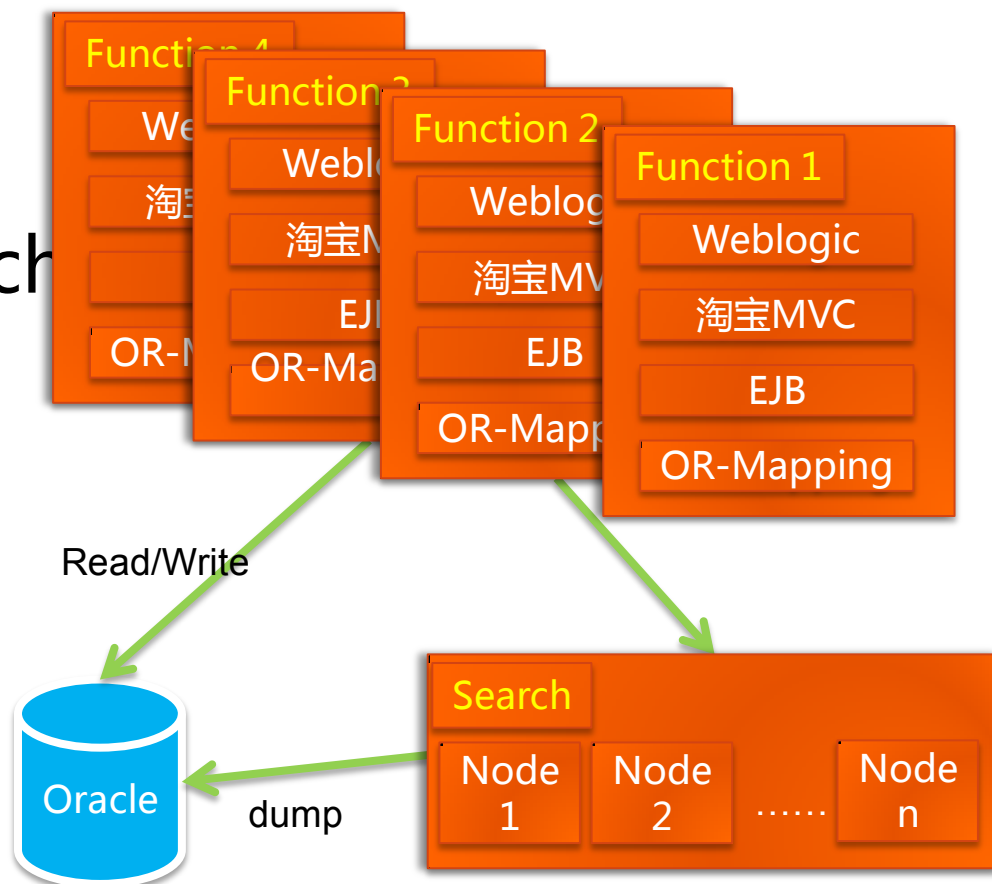
- MySQL迁移至Oracle
- 引入SQL Relay中间件





## V2.0 2004.2-2005.03

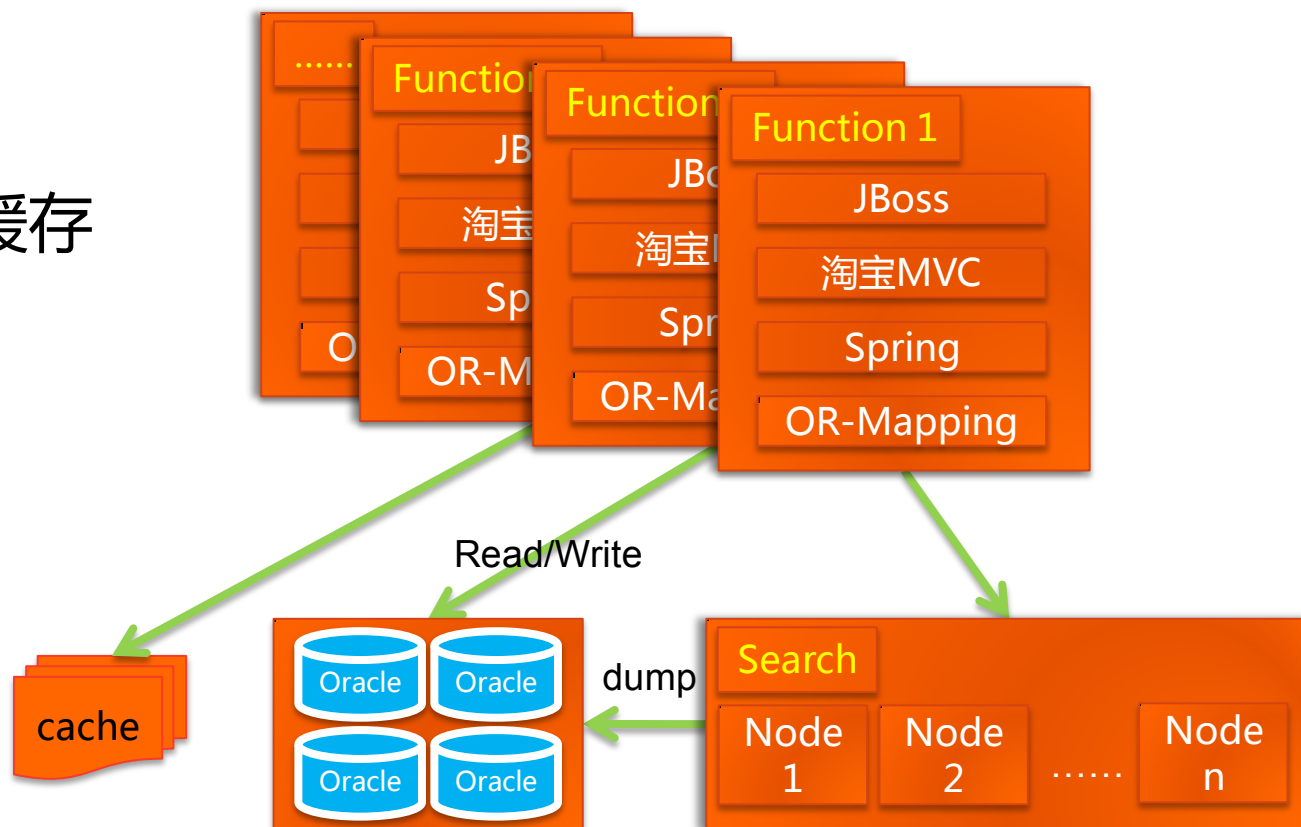
- php迁移至java
- MVC框架WebX
- 项目管理工具AntX
- 引入搜索引擎ISearch





# V2.1 2004.10 – 2007.01

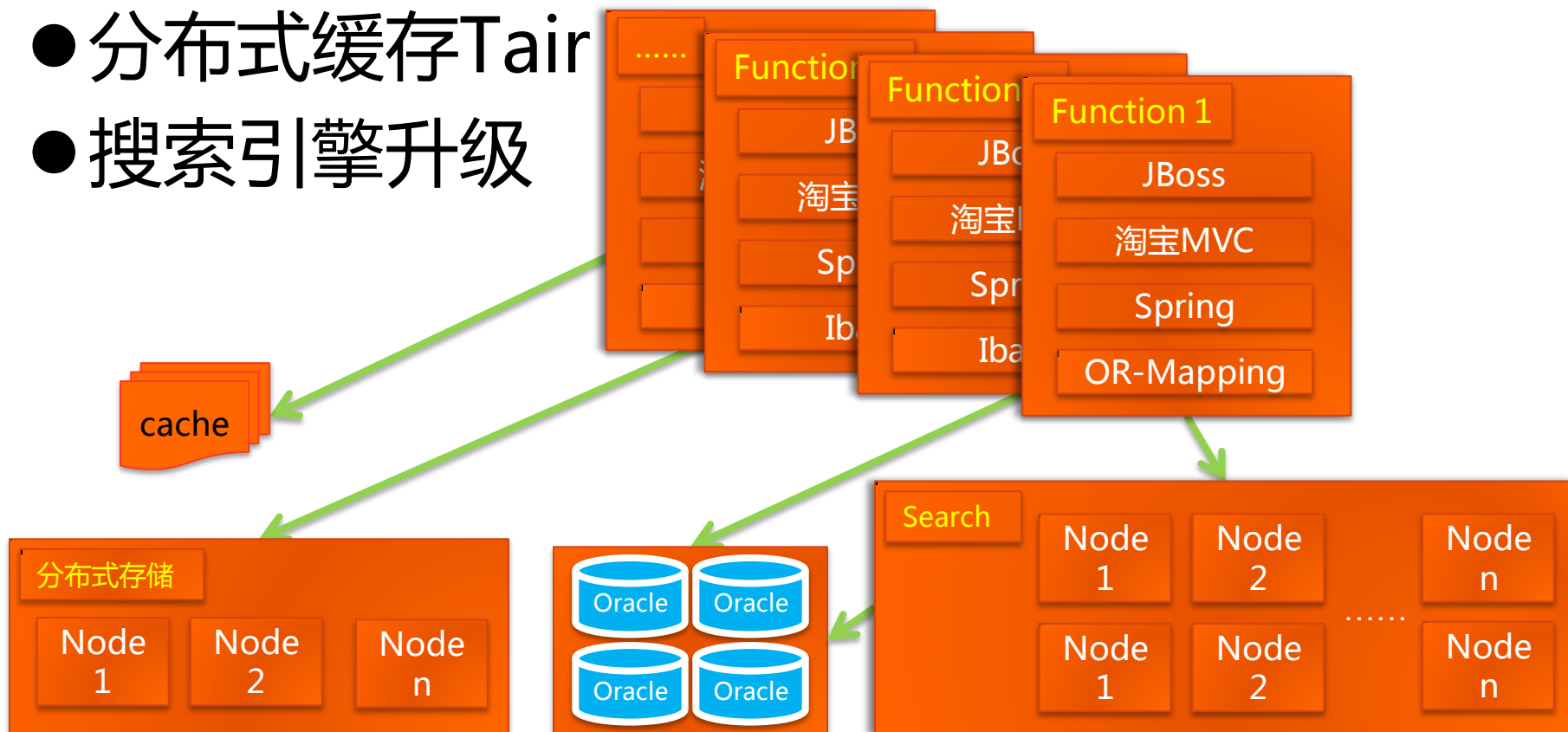
- weblogic迁移至jboss
- 支持分库的数据访问框架
- 抛弃EJB
- 引入Spring
- 基于BDB的缓存
- 建立CDN





## V2.2 2006.10 – 2007.12

- 分布式存储TFS
- 分布式缓存Tair
- 搜索引擎升级







# 需求

- 高稳定性
  - 高数据安全
  - 高可用性
- 高容量,高性能
  - 高并发处理能力
  - 高存储容量
  - 低响应时间
- 低成本
  - 硬件
  - 人力





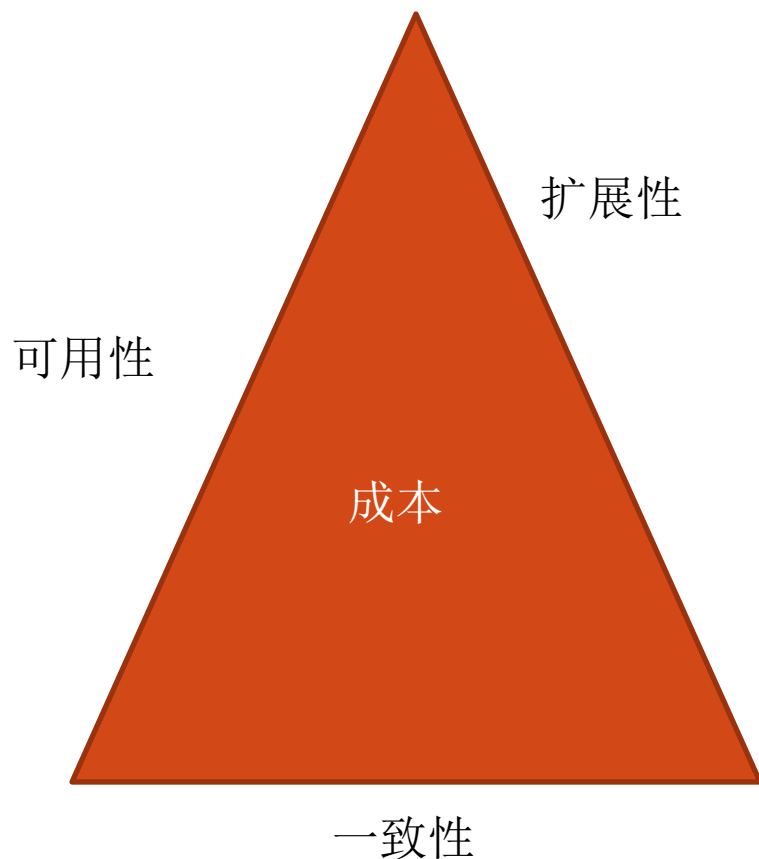
# 处理策略

- 高稳定性
  - 备份
  - 隔离
- 高容量，高性能
  - “人”多力量大
  - 分割
  - 异步
- 成本
  - 提高效率
  - 自动化





# 整体策略：三角原则（CAP）



效果 = 周长  
周长和三个边长为线性关系

成本 = 面积  
面积和三个边长成指数关系

当边长小于某阈值  
效果 = 最小边长度



## V3.0 2007.12 -

- 应用透明伸缩
  - Session框架
  - 高性能服务框架HSF
  - 消息系统Notify
  - 业务中心建立
- 数据透明伸缩
  - 分布式数据层TDDL
- 稳定性
  - 容灾
- 成本
  - 自动化
  - 数据迁移到MySQL





## V3.0 应用透明伸缩

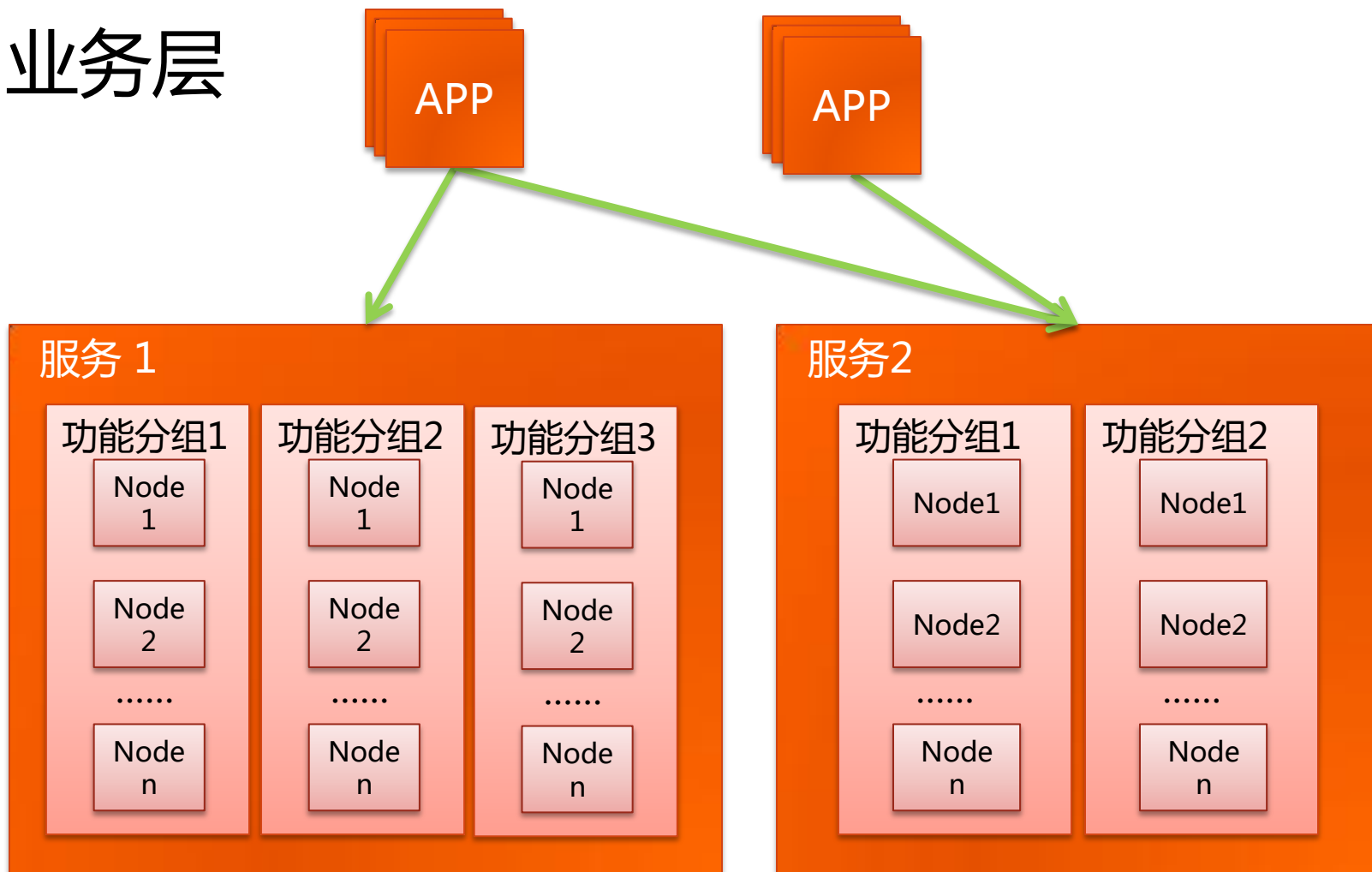
- 展现层-会话处理很重要
  - 粘性session
  - session复制
  - 集中式session
  - 不用session





# V3.0 应用透明伸缩

## ● 业务层





# V3.0 应用透明伸缩

- 业务处理
  - 统一和隔离
  - 负载均衡
  - 一片私有的“云”





# V3.0 数据透明伸缩

- 容量和性能的扩展
  - 垂直分割
    - 按功能
  - 水平分割
    - 按规则
- 路由
  - 透明化
- 复制
  - 空间换时间

核心  
业务  
数据



.....



业务1

.....



业务2

.....







## V3.0 数据分割策略

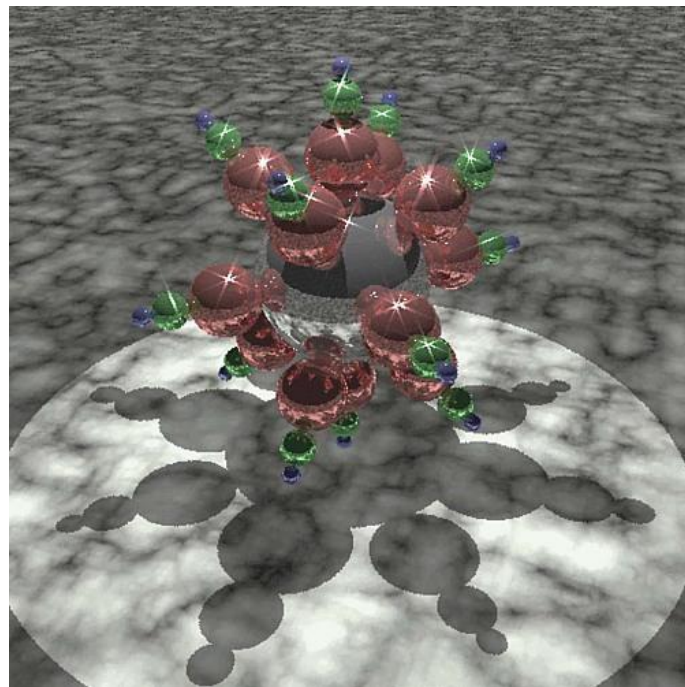
- 复杂结构的水平分割处理方案
  - $n$ 策略
  - $2n$ 策略
  - $n+1$ 策略
  - 读写分离策略





# 数据优化策略：时空原则+28原则

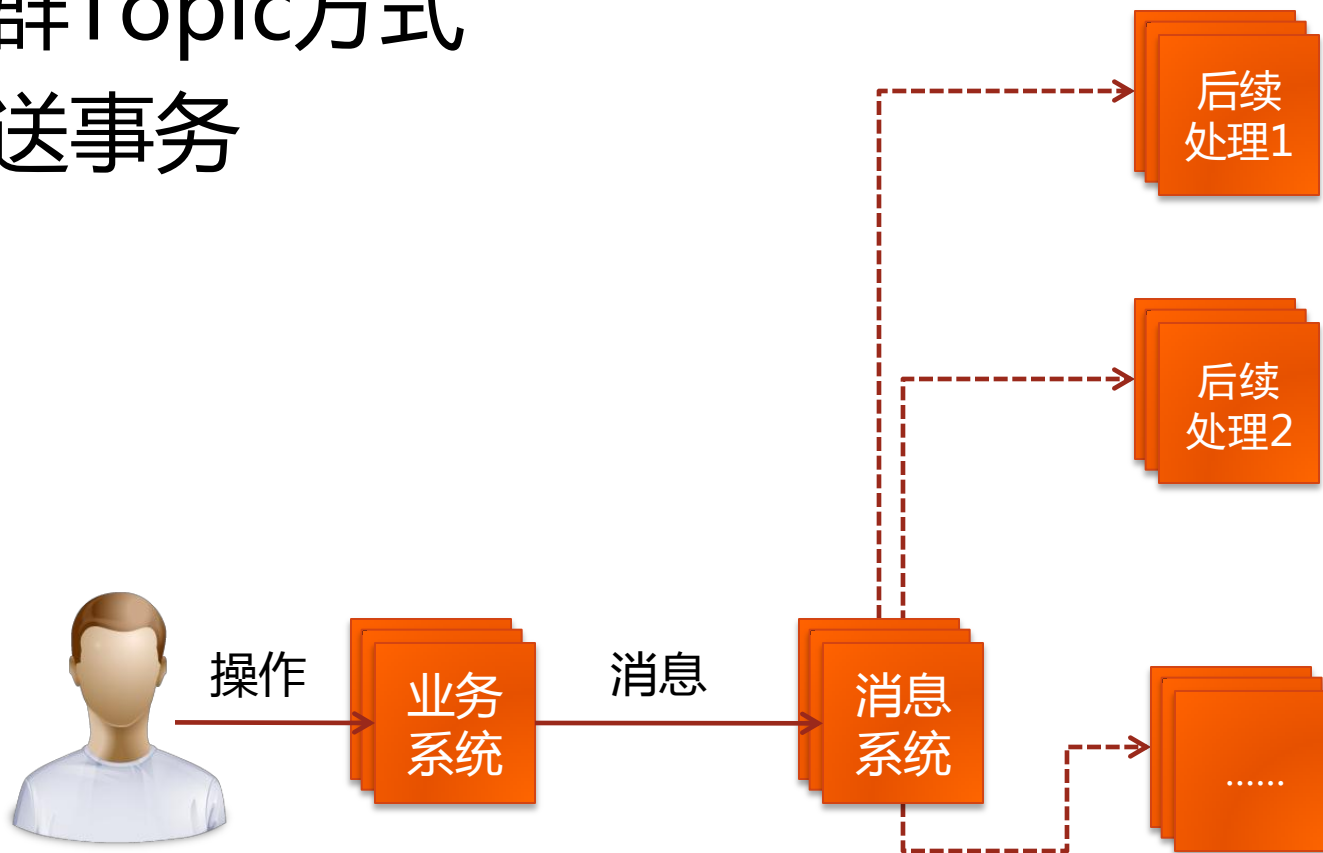
- 老生常谈
  - 空间换时间
  - 时间换空间
- 效果和成本的平衡





# V3.0 消息系统

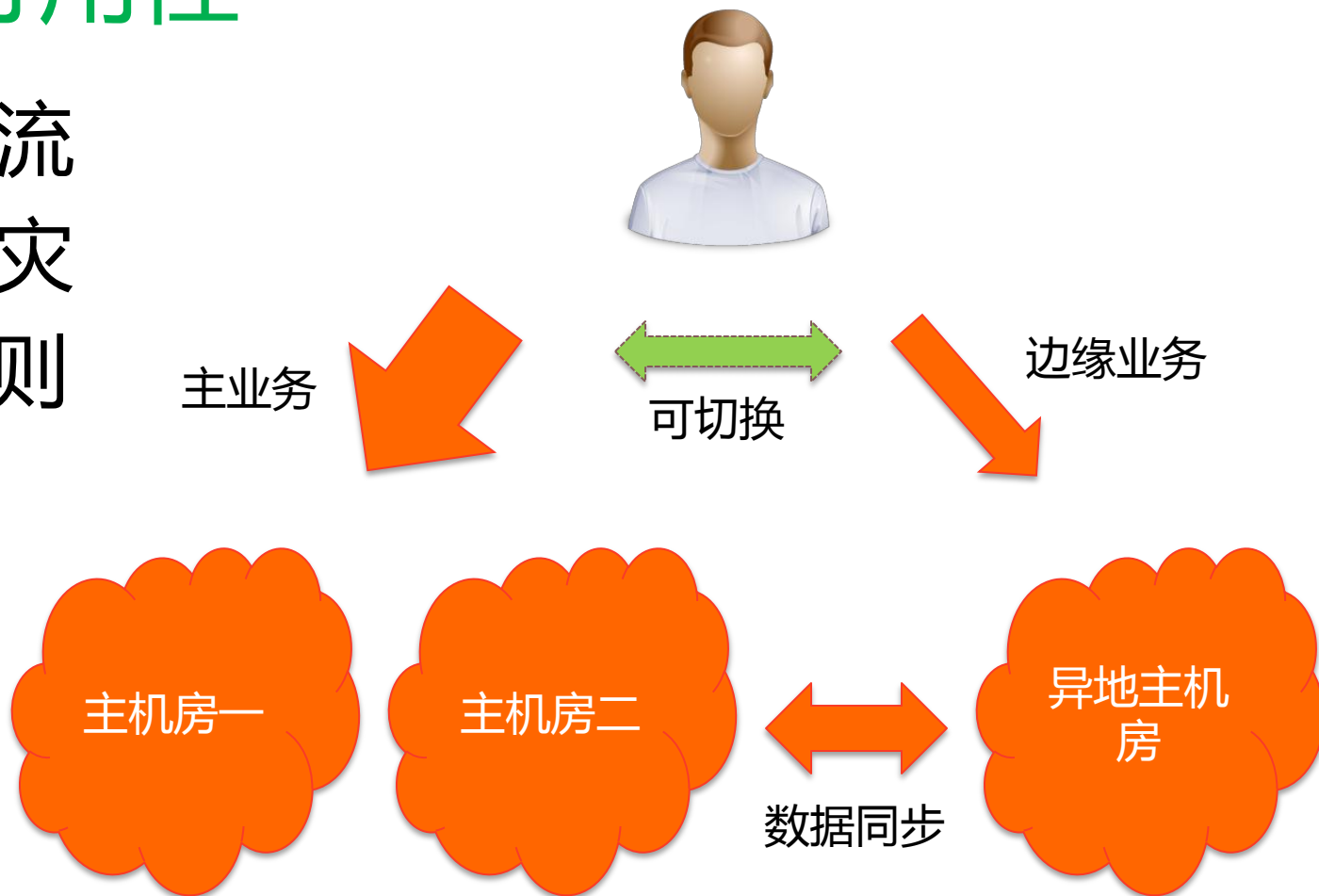
- 集群Topic方式
- 发送事务





# V3.0 可用性

- 同城分流
- 异地容灾
- n+1原则





# V3.0 自动化



***A: I wanna the fried chicken!***  
***B: Sir, we will be arriving at the nearest KFC soon.***



# 总结

- 适当放弃一致性
- 备份和隔离解决稳定性问题
- 分割和异步解决性能问题
- 自动化降低人力成本
- 产品化管理



# Questions?

