

China · Beijing

ArchSummit全球架构师 峰会北京站

姓名 许家滔

Title 微服务在微信的架构实践



[北京站]

主办方 **Geekbang** & **InfoQ**
极客邦科技



促进软件开发领域知识与创新的传播



关注InfoQ官方微信
及时获取ArchSummit
大会演讲视频信息



全球软件开发大会 [北京站]

2017年4月16-18日 北京·国家会议中心

咨询热线: 010-64738142



全球架构师峰会 2016 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线: 010-89880682

微服务在微信后台的架构实践

姓名 许家滔

Title 微服务在微信的架构实践

自我介绍

- 许家滔 (sunnyxu@tencent.com)
- 微信技术架构部，后台开发中心
- 基础微信业务 / 基础服务 / 基础组件
- 负责微信后台系统，一直专注于布式存储平台和后台服务框架等
- 开源 libco
github.com/tencent/libco

大 纲

第一部分 背景与概览

第二部分 基础架构

第三部分 监控与测试

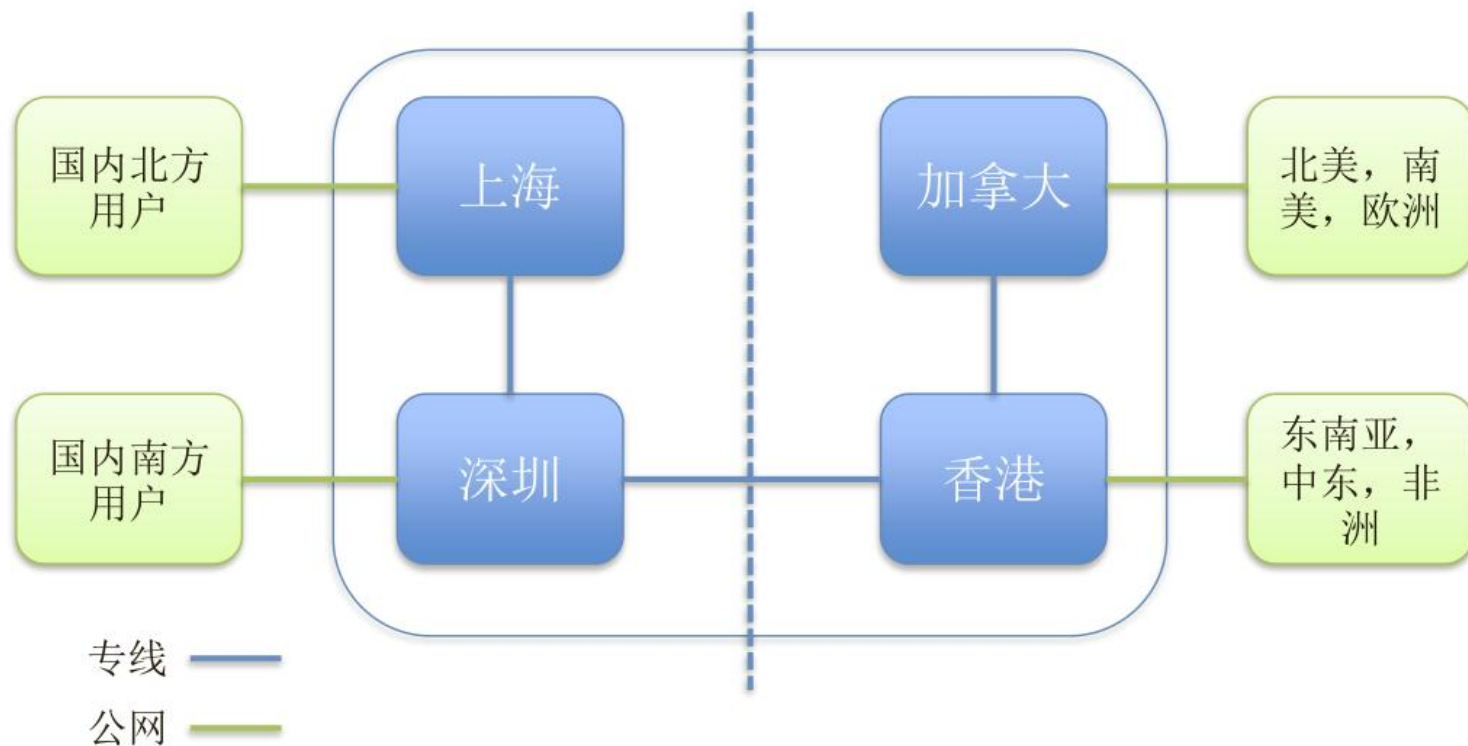
第四部分 跨业务服务关系

背景与概览

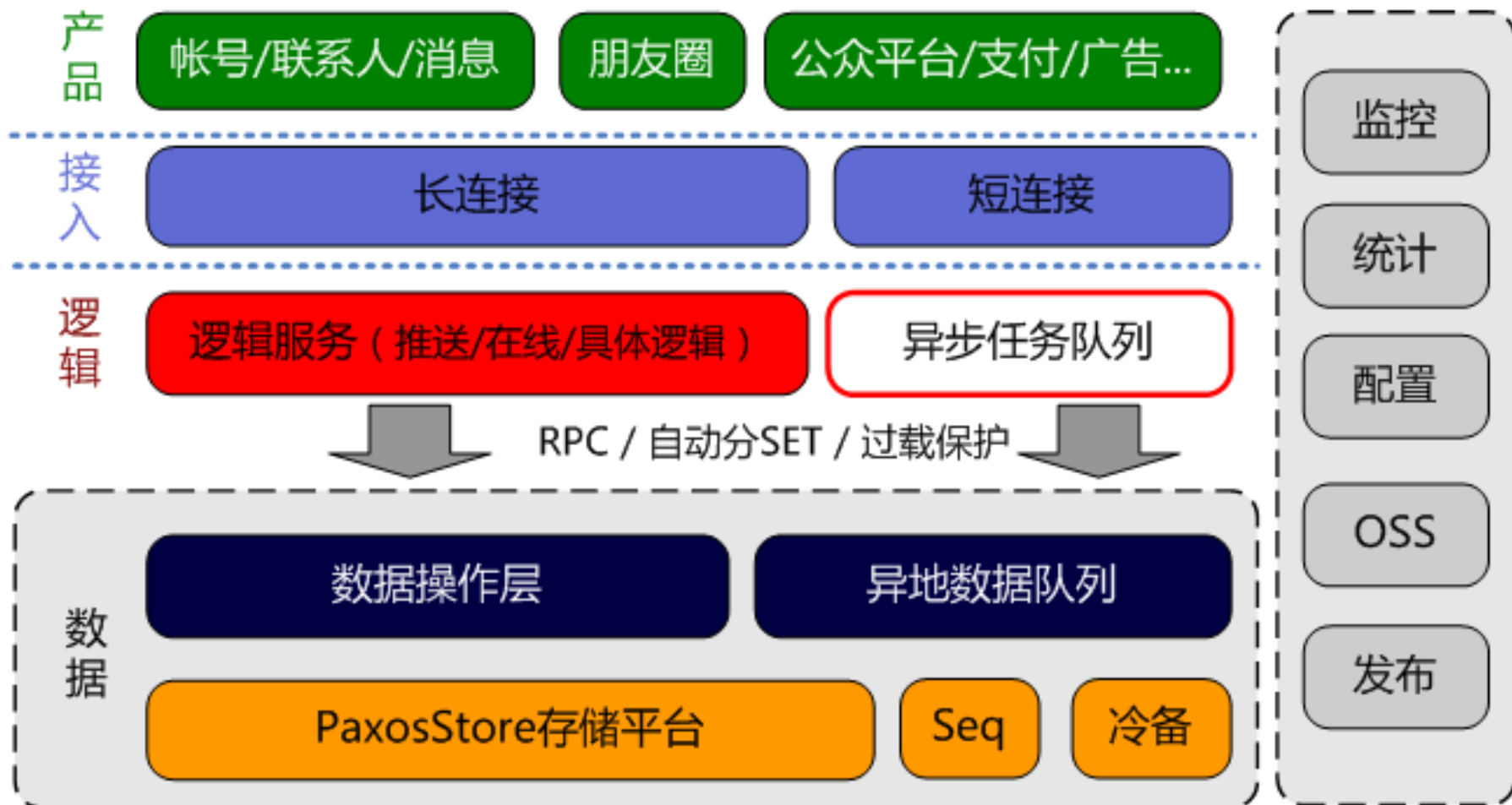
- 敏捷开发
- 统一的基础架构与运营管理
- 千级模块的庞大系统
基础微信/公众平台/微信支付等等
数万服务器

微信的后台架构与微服务理念有很多相通之处，尝试从微信后台架构出发描述互联网后台的微服务实践

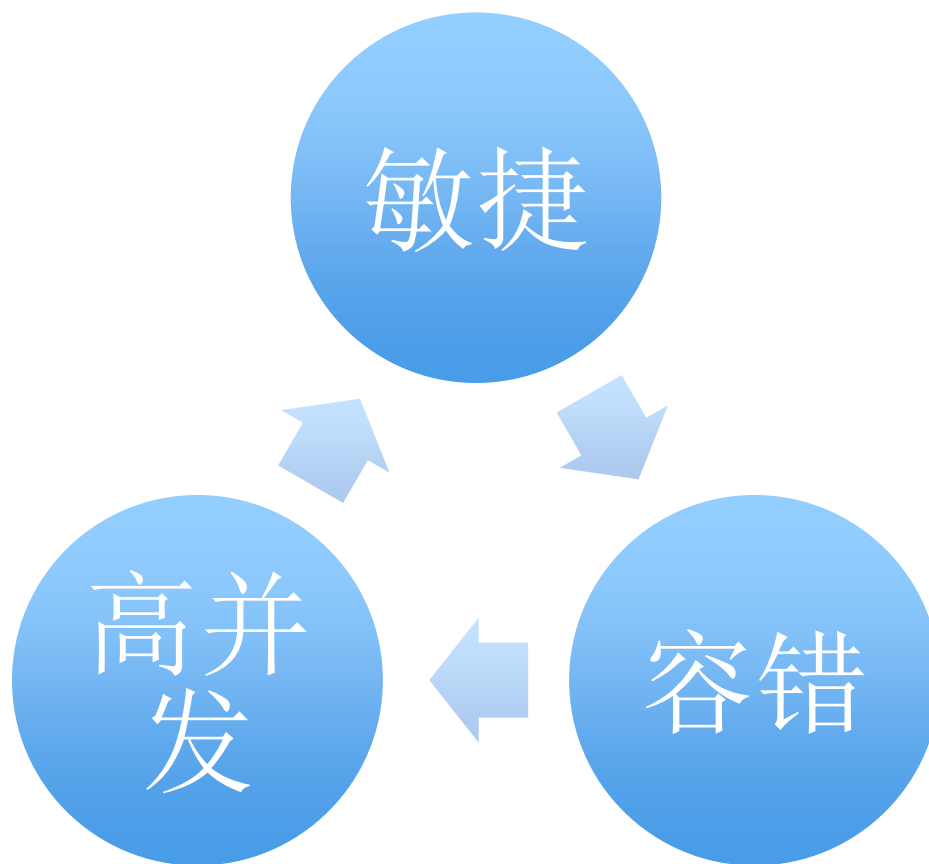
背景与概览



背景与概览



微信的海量场景下 微服务架构需要怎样的能力？



微服务基础架构

服务框架

- 服务布局（多城市/园区管理）
- 远程调用（**c++**大规模协程使用，protobuf，异步队列）
<https://github.com/tencent/libco>
- 容错（重试策略，**过载保护**，负载均衡，柔性可用）
- 部署管理，发布系统

服务布局

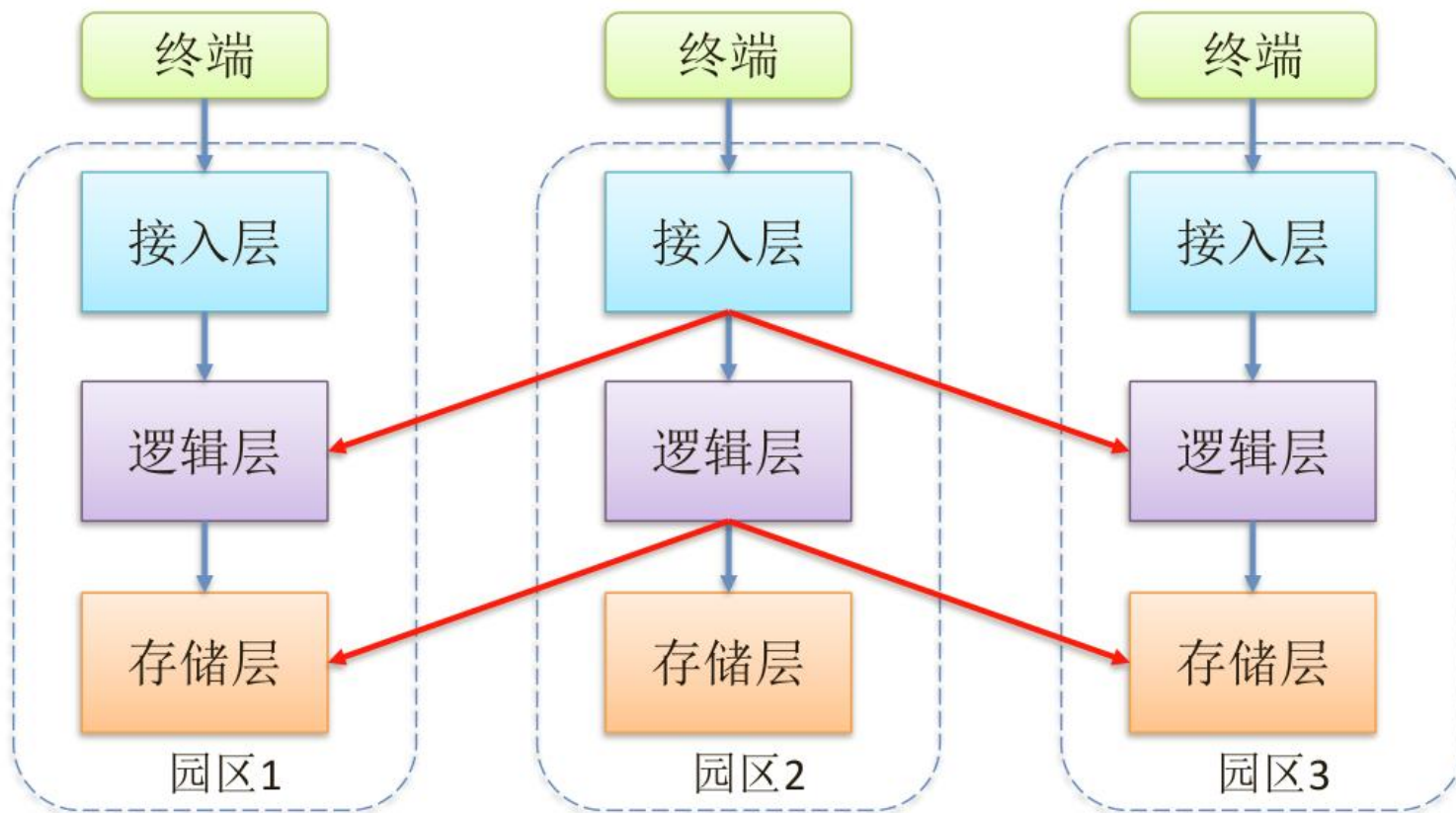
多地自治，园区互备

城市间后备的公网udp通道



服务布局

多地自治，园区互备



远程调用

当你掏出手机微信支付，背后会引发多少次RPC？

细腻的业务逻辑，高扇出的RPC调用

事件驱动的异步编程

callback / stat machine / future promise

如何重用，如何敏捷？

<https://github.com/tencent/libco>

远程调用 / libco

极简

一行代码10块钱的故事

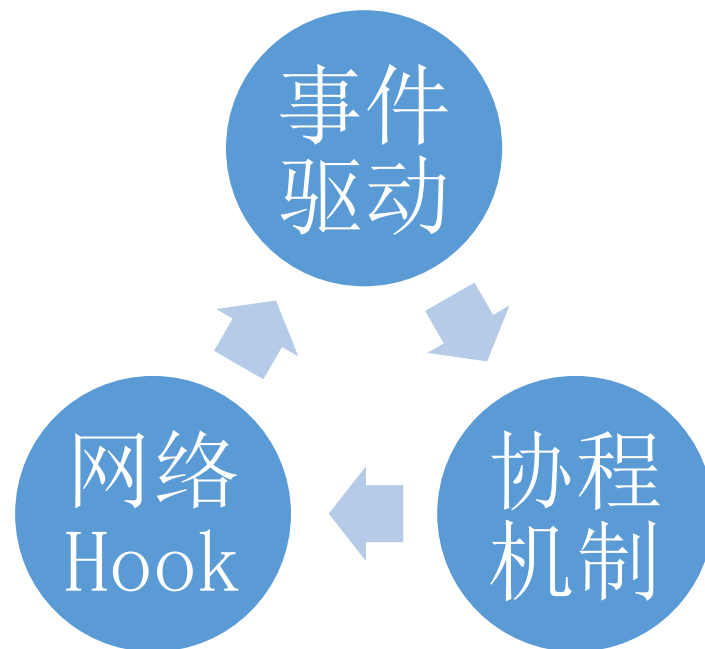
完备

时间轮盘与事件处理链

常用网络编程模式

同步原语

...



<https://github.com/tencent/libco>

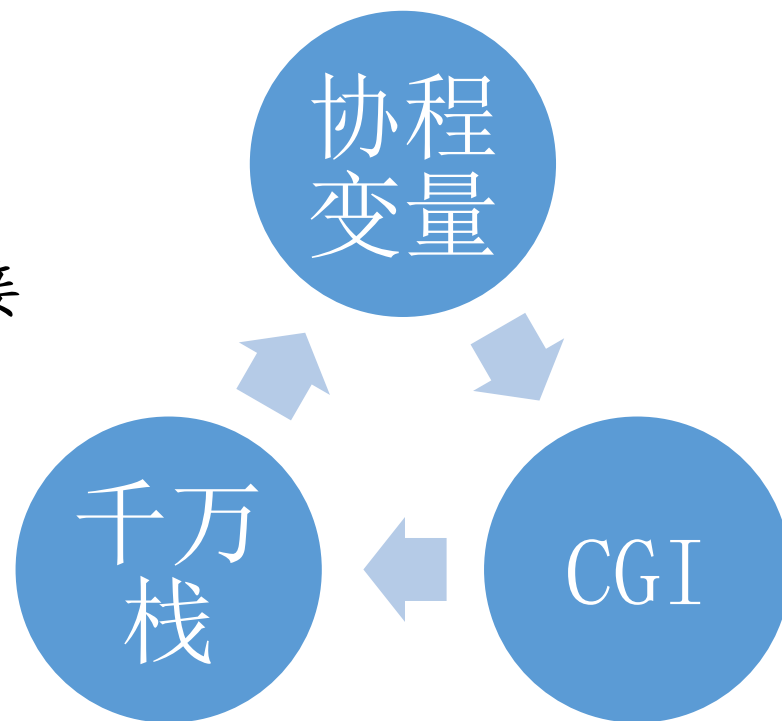
远程调用 / libco

协程变量

支持标准CGI开发

支持gethostbyname

高级用户可以开启千万连接

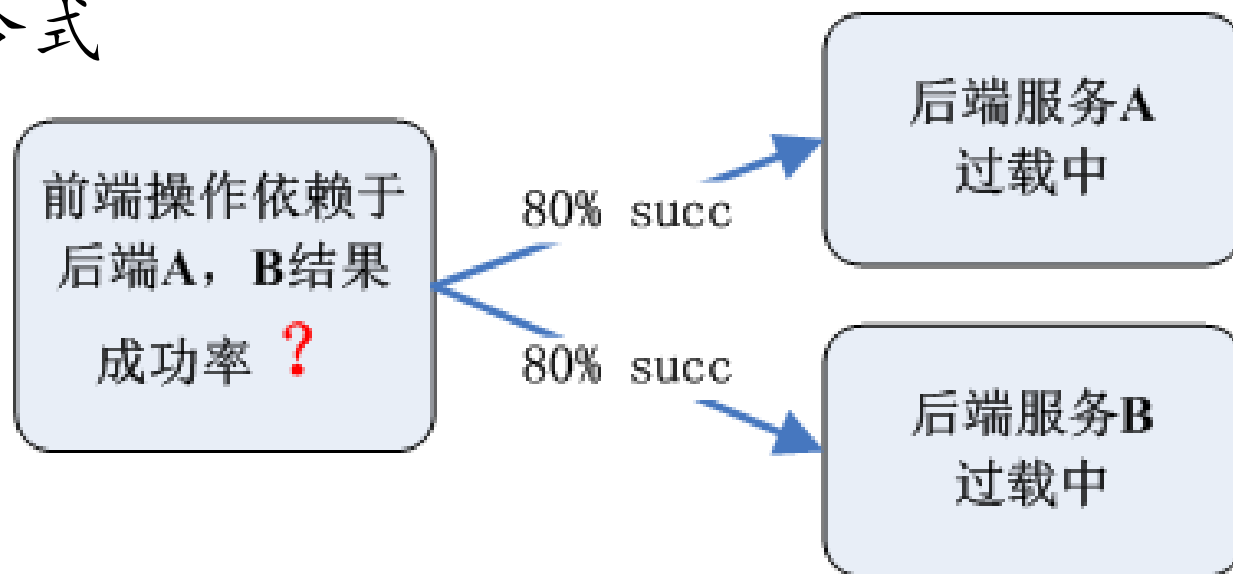


<https://github.com/tencent/libco>

微服务基础架构

过载保护

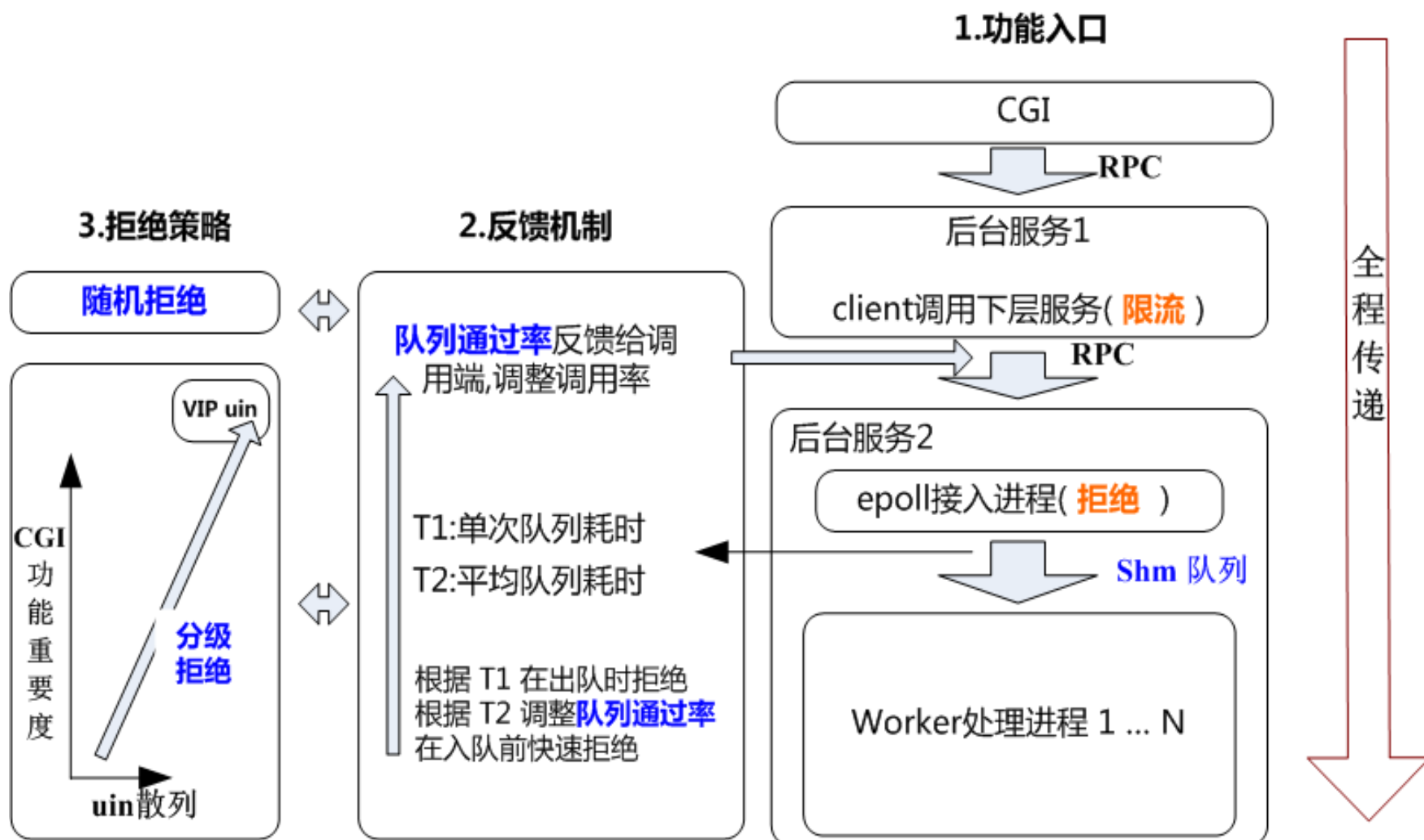
- 轻重分离
- 队列式
- 组合命令式



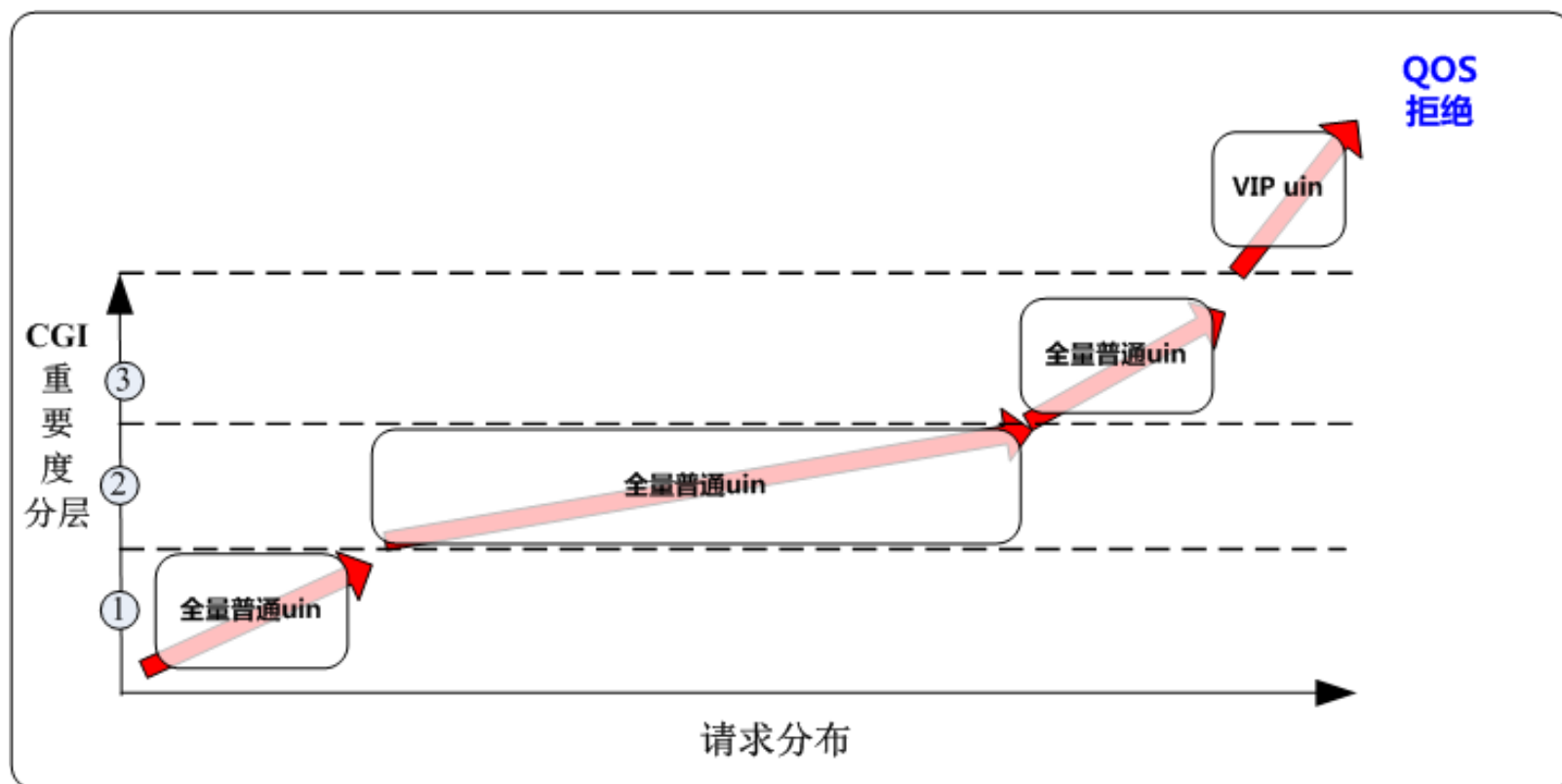
过载保护的核心是反馈

推荐《失控》

微服务基础架构 / 过载保护



微服务基础架构 / 过载保护



微服务基础架构

数据存储

- 一致性的权衡
- PaxosStore
- 异步队列
- 其他

微信PaxosStore

- 同步复制，多主多写（园区间数据一致）
- 支持键值/列表等数据结构
- 支持表格，单表亿行，并可执行类SQL事务
- 插件化设计，统一的分布框架下面定制的存储引擎
- 低成本，灵活组合不同的资源（内存/ssd/sata）
- 数千台机器组成的集群

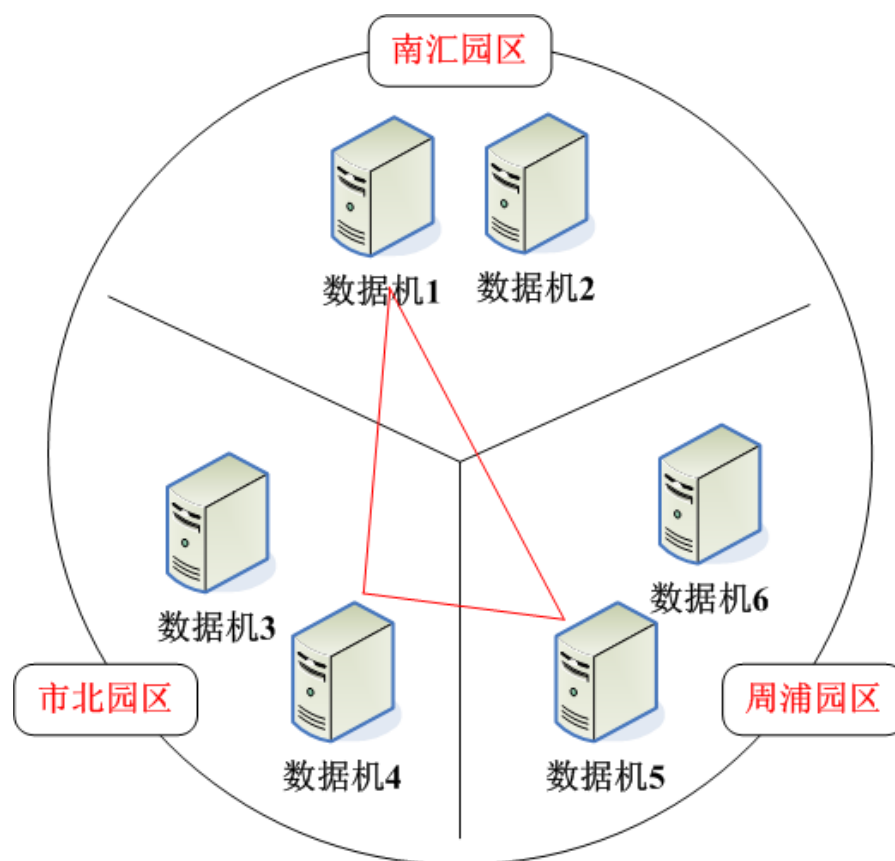
微信PaxosStore

如何设计数据容灾以获得自由切换的能力
以同城为大前提的低延迟环境
非租约Paxos

负载均衡

单机/园区分级容灾，kv64架构
单机故障分摊25%流量
园区故障分摊33%流量

SET部署,一致性哈希均匀分布
具备细粒度的伸缩性



为何需要这么重的方式？

应用简单快速

数据容灾

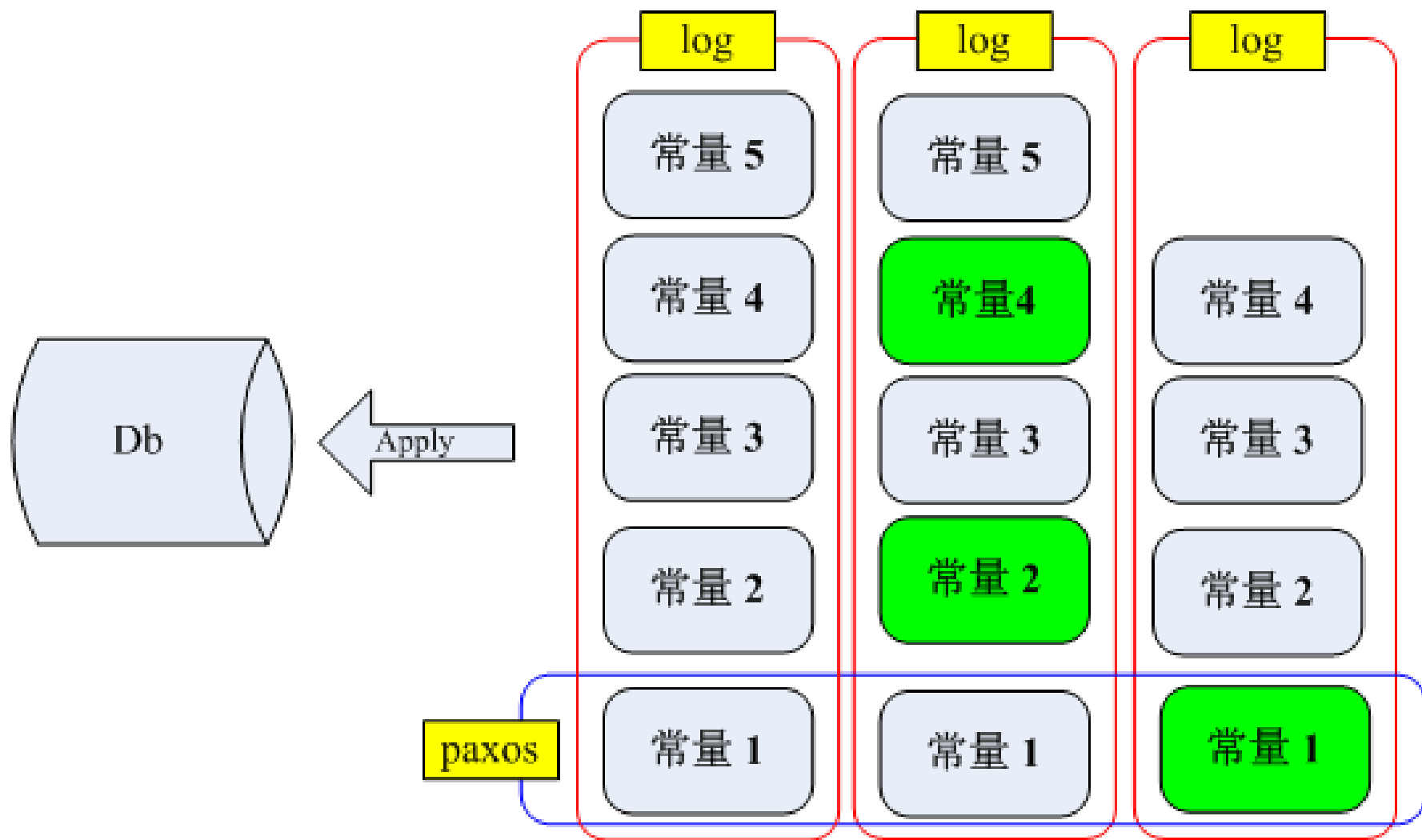
服务可用性

.....



数据存储的
基本要求

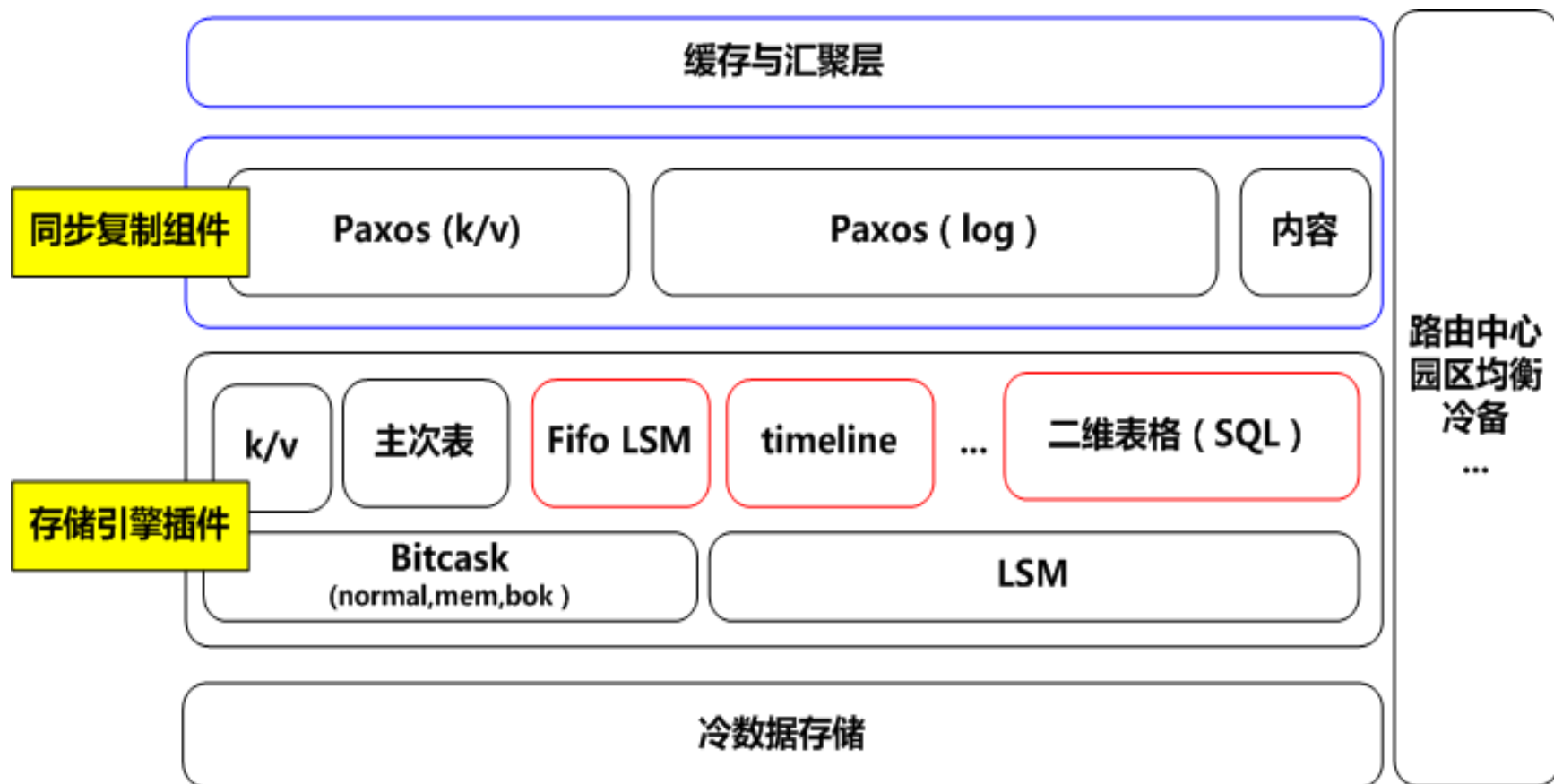
Raft ?



微信PaxosStore

- Paxos工程实现在海量服务上的权衡与创新
 - 多主多写（去租约无leader模型）
 - 过载保护
 - 无切换不可用时间
 - 隔离性：细粒度，单机kw+的Paxos(log)实例
 - 高性能：协议交互优化

微信PaxosStore / 系统架构

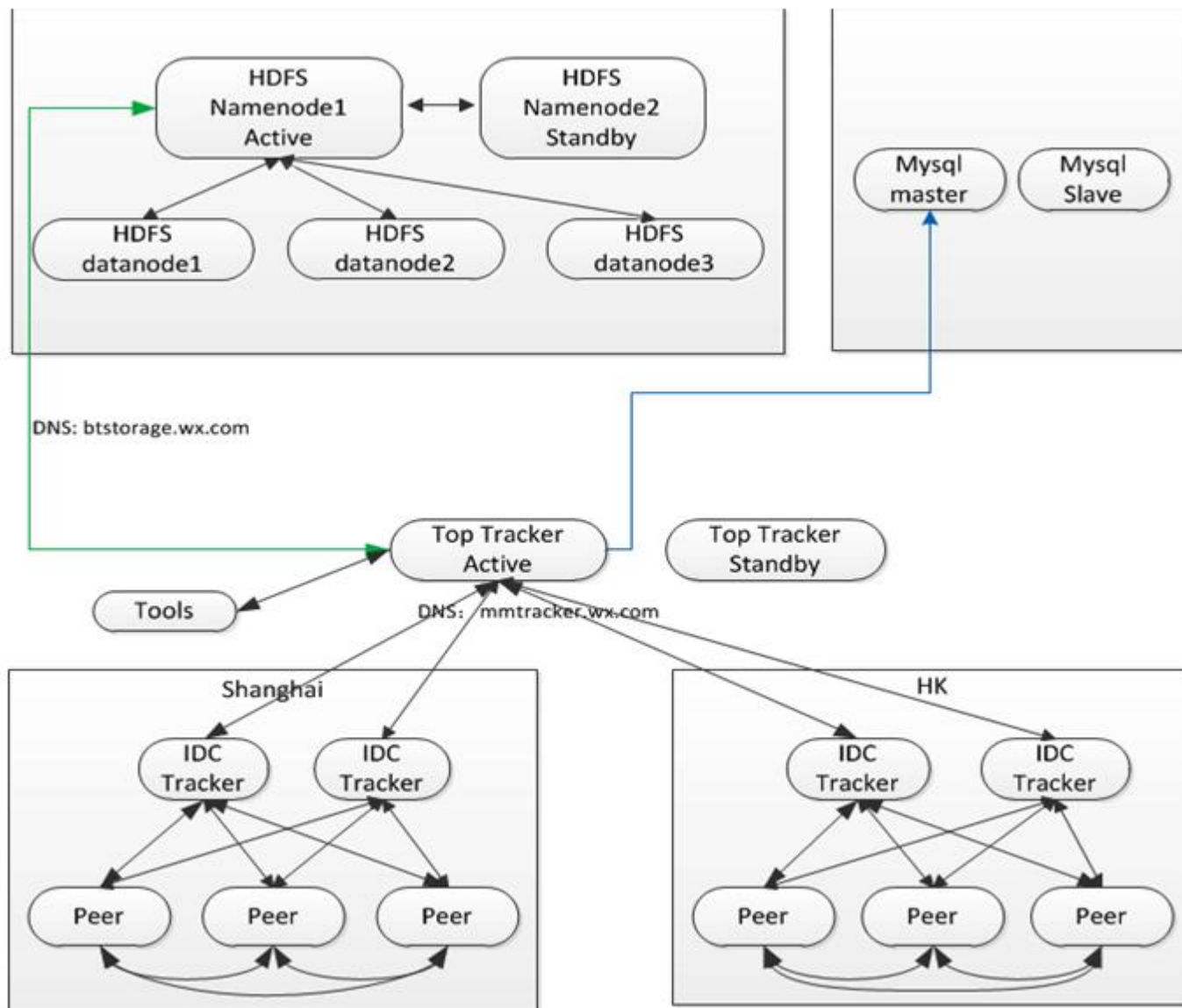


基础架构

发布系统

- 三军未动，粮草先行
- 基于**BitTorrent**的传输方案
- Facebook/Twitter/EBay

发布系统

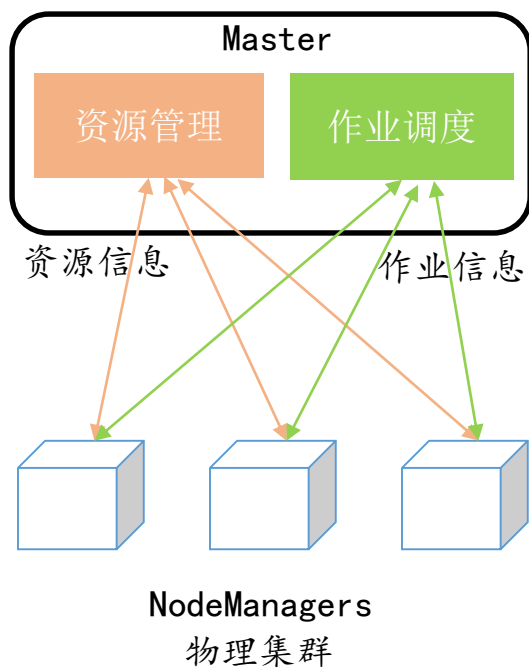


基础架构

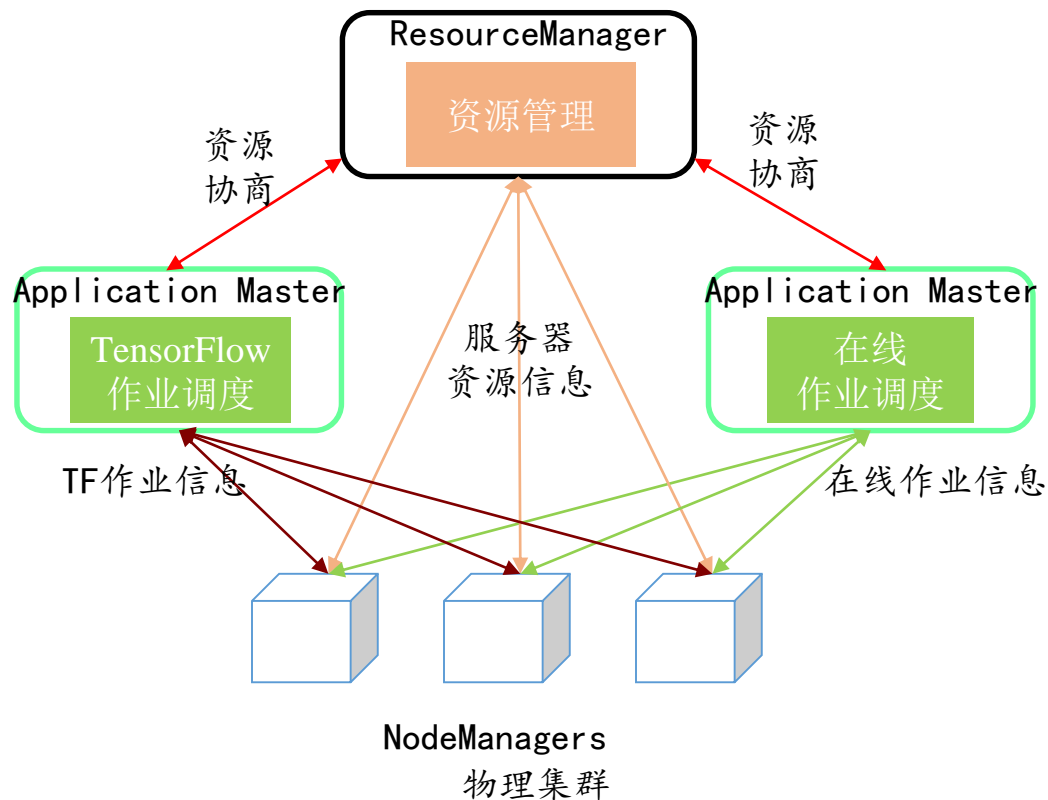
容器与集群管理

- Cgroup
- Docker
- 微信Yard

微信Yard



中央调度器
(JobTracker, Torca)



双层调度器
(Mesos, Yarn)

基础架构

需求

- 高可靠、高可用：完善的容灾能力，能管理微信在线服务
- 可扩展、伸缩性好：能方便的接入各种不同的离线计算框架
- 独立性：各个作业资源相互独立，不争用
- 大规模：能管理数万级别服务器的集群和作业

选择

双层调度器模型

监控与测试

性能监控与故障分析

- 分钟级的调用关系
- 可用性指标
- 手动介入原则

测试方法

- 分布式系统测试代价高
- Mock
- 单机集成

跨业务服务关系

代码管理

- 依赖管理
- 编译系统

服务解耦

- Gateway
- broker

THANKS



[北京站]

主办方 **Geekbang** > **InfoQ**
极客邦科技