



全球

World Of Tech 2017

2017年12月1日-2日 • 深圳中洲万豪酒店

软件开发技术峰会

DEVELOPMENT



云端微服务架构的 运维思考

熊普江

腾讯资深架构师

云端微服务架构的运维思考

01

架构演进的理解

02

微服务架构的特点与趋势

03

微服务架构运维的解决之道

01 架构演进的理解

- ◆ 移动互连时代全面来临
- ◆ 硬件技术发展迅速，服务器性能越来越强大
- ◆ 容器技术兴起，轻量协议成熟应用
- ◆ 应用云化逐渐普及
- ◆ 开发模式转变
- ◆ 精细化运营转变

信息化模式的转型

传统信息化模式

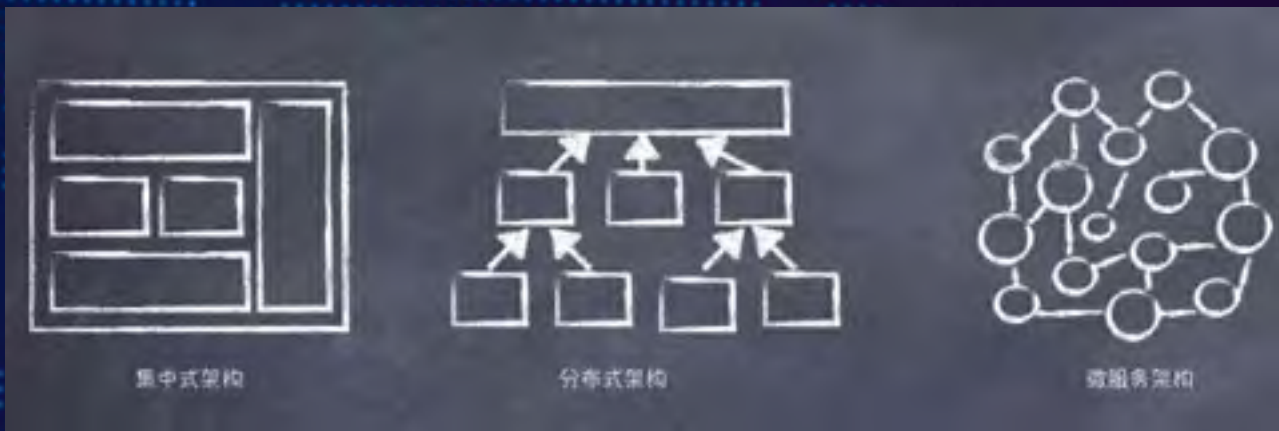


“平台+应用”的建设模式



从纵切到横切的转型

微服务架构是架构演进的必然结果

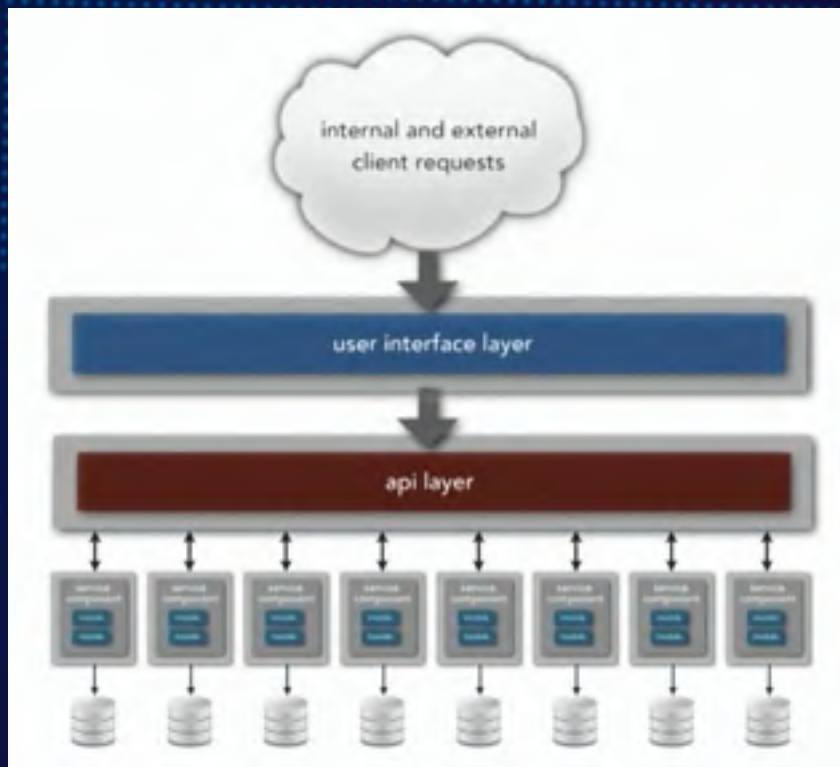


02

微服务架构的特点与趋势

- ◆ 一种架构风格
- ◆ 强调服务的粒度、强调敏捷和健壮
- ◆ 单一、独立，每个微服务专注并很好地完成一件任务
- ◆ 解耦、去中心化，组件化封装
- ◆ 多个微服务组合完成相对复杂的业务系统
- ◆ 高性能、高容错性

微服务架构解析

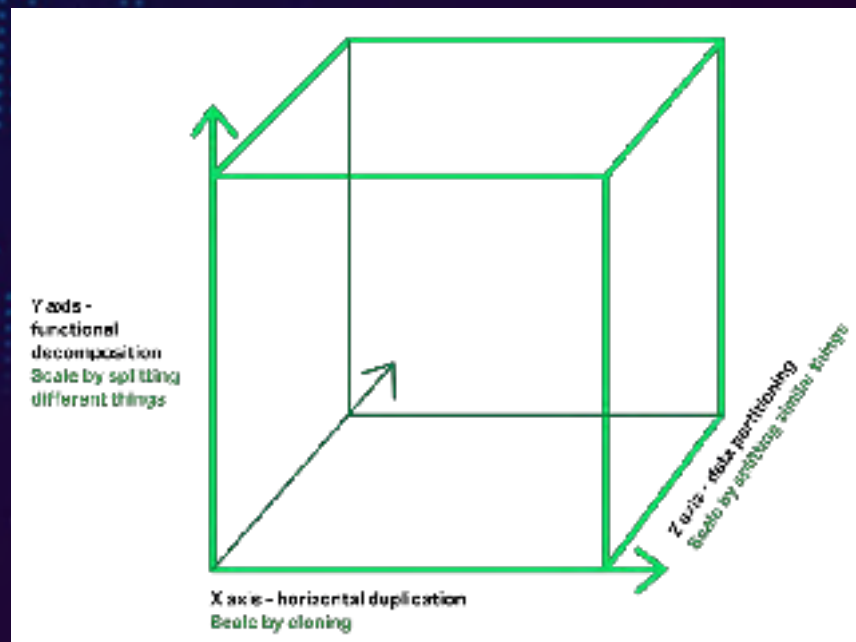


- ✓ 适用于构建复杂的应用
- ✓ 分布式（微服务间单一共同协议）
- ✓ 分别部署（自动化DevOps工具）
- ✓ 服务组件（基础公共、功能性）
- ✓ 微服务的边界（独立开发、测试，且发布、部署后不影响其他）
- ✓ 不共享任何事物
- ✓ API 层（微服务与外部服务接口）
- ✓ 优先开发新的微服务，而不是在微服务上增加新场景或功能

左图：web应用程序拆解微服务

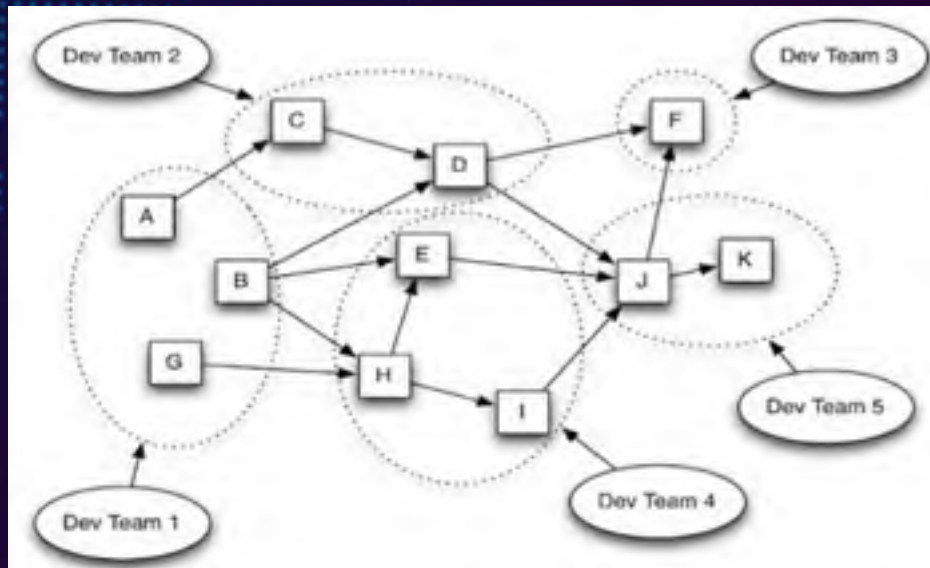
微服务架构的优势明显

- ✓ 单个微服务更易理解，方便开发与维护
- ✓ 应用解耦，对应用整体应用交付而言，开发迭代更敏捷
- ✓ 技术选择更加自由，微服务不再需要限定某个统一的技术实现
- ✓ 微服务独立部署，应用更稳定
- ✓ 扩展更加容易与快速



微服务架构带来的挑战

- ✓ 开发模式需要相应调整
- ✓ 服务数量急剧增多
- ✓ 容器编排与配置管理等会变得异常复杂
- ✓ 业务的容量管理变得更加困难，资源利用效率难以提升
- ✓ 监控的颗粒度增多，依赖及关联关系更加复杂
- ✓ 微服务故障恢复、调度需要更精细化



02 微服务架构下的运维思考

- 容量管理
- 容器编排与配置管理
- 服务监控
- 故障恢复与业务调度
- 快速扩展
- 资源利用效率

03

微服务架构运维的解决之道

- ◆ 以微信为例
- ◆ 微服务架构应用
- ◆ 微服务容量管理
- ◆ 监控、部署、调度
- ◆ 精细化运营
- ◆ 未来演进

微信服务为什么微服务云化

- 2011 微信诞生
 - 一直主张敏捷开发（态度、试错），小步快跑
 - 四大法器：大系统小做、让一切可扩展、有基础组件、轻松上线
-
- 2013~2014左右 微信微服务模块数达5K
 - 同物理机多服务混布，资源抢占问题突出
 - 微服务资源利用率要求

微服务架构在微信业务的应用与管理

- 从场景上看：

- ✓ 服务布局
- ✓ 容错（重试策略、过载保护、负载均衡、柔性可用）
- ✓ 容量管理与监控
- ✓ 部署管理与调度
- ✓ 精细化运营

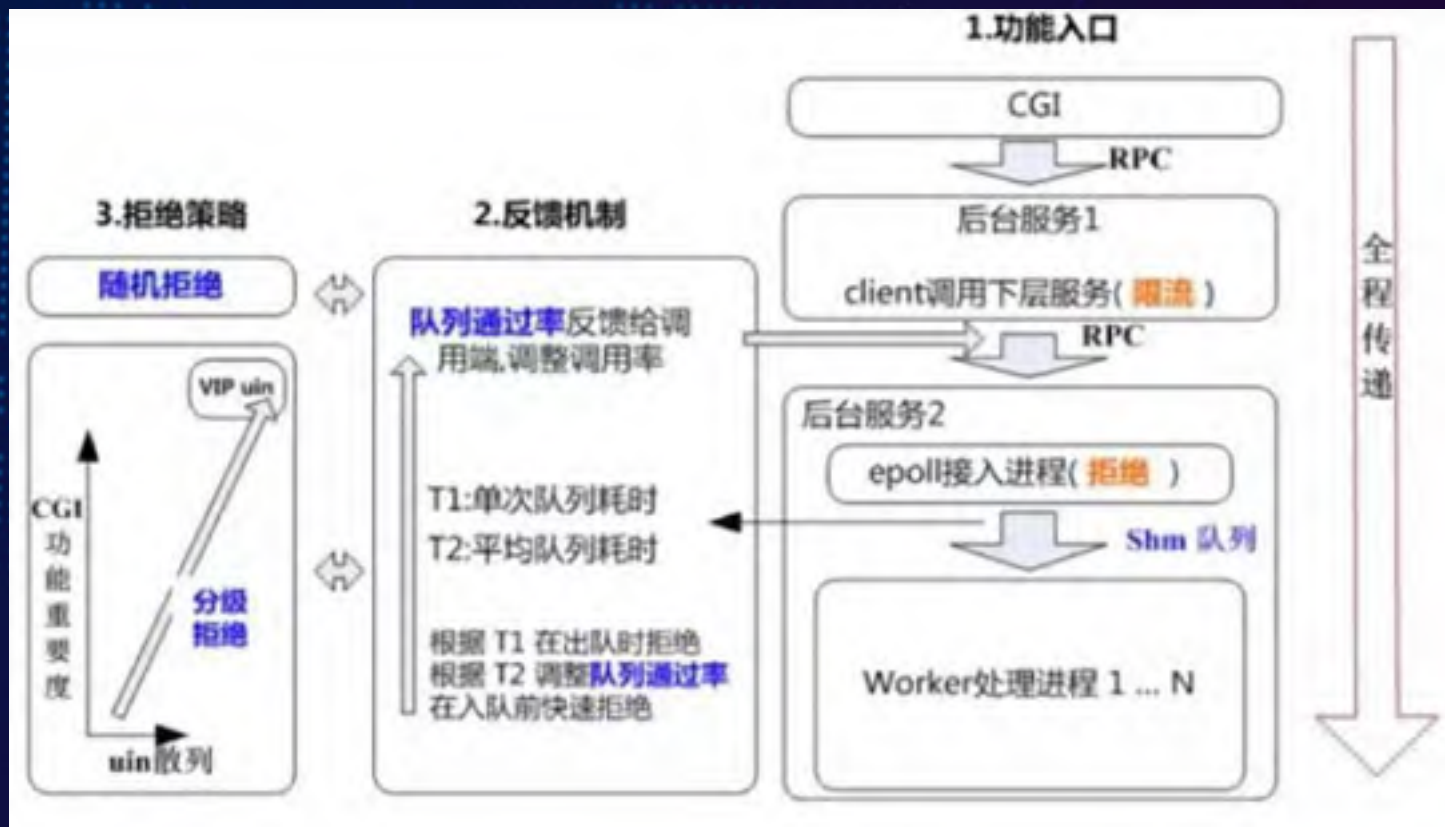
- 从技术分层上看：

- ✓ 接入层：服务布局，独占物理机、容量充足、变更少
- ✓ 逻辑层（最适合微服务化）：混合部署、容量不可控、变更频繁
- ✓ 存储层：服务布局，按需独占物理机、容量可控

服务布局



过载保护

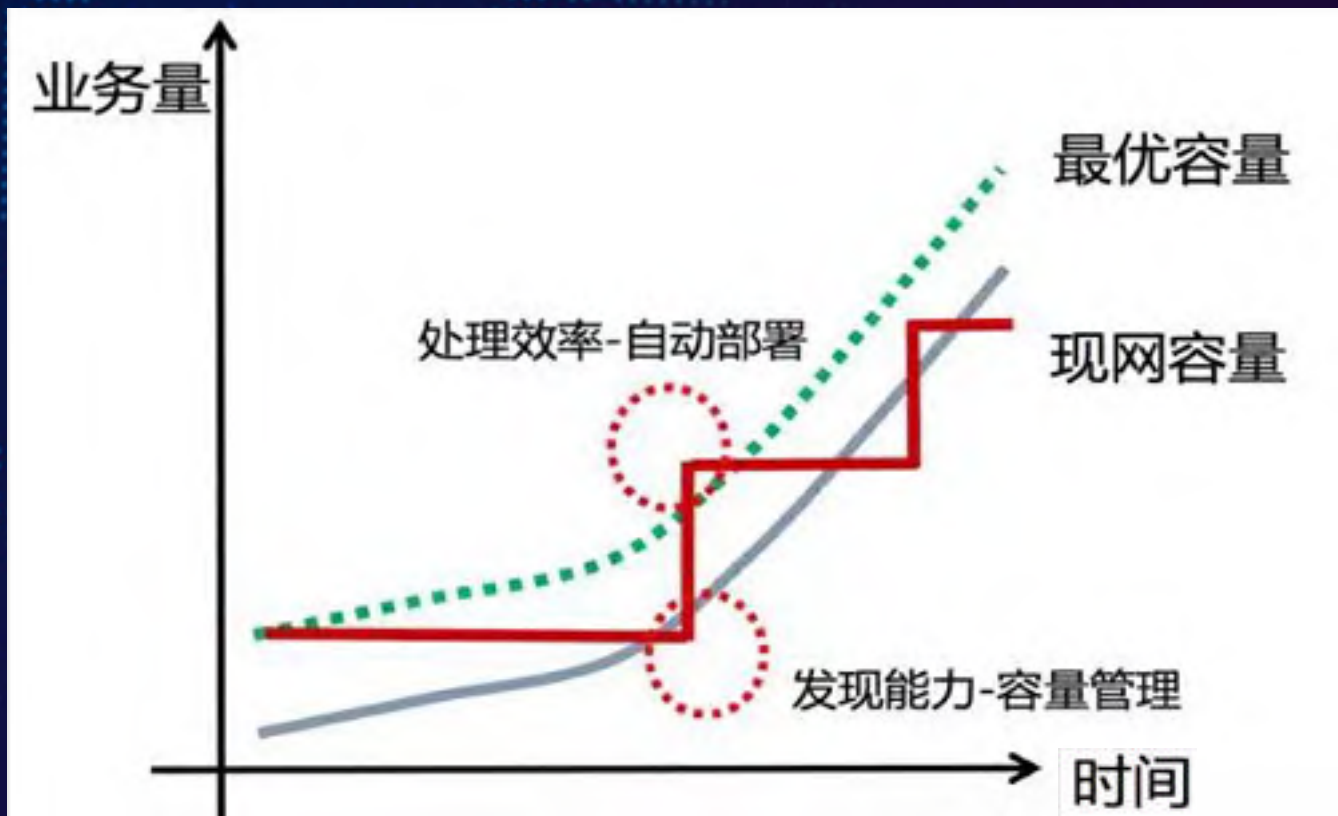


注：本图片引自许家滔~微服务在微信的架构实践

容量管理

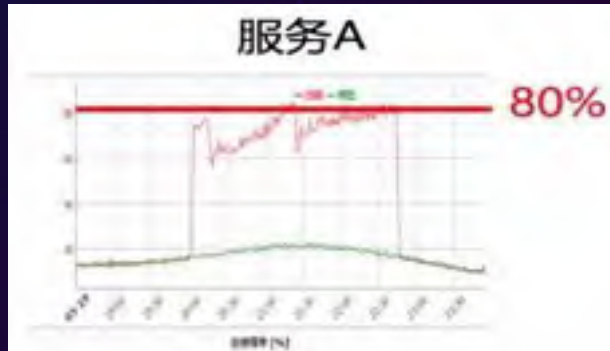
- 三个前提：
 - ✓ 微服务间资源隔离管理
 - ✓ 微服务的过载自我保护能力（框架）
 - ✓ 服务快速伸缩操作
- 业务支撑与容量曲线模型
- 建议的微服务容量评估模型

容量管理-业务支撑与容量关系

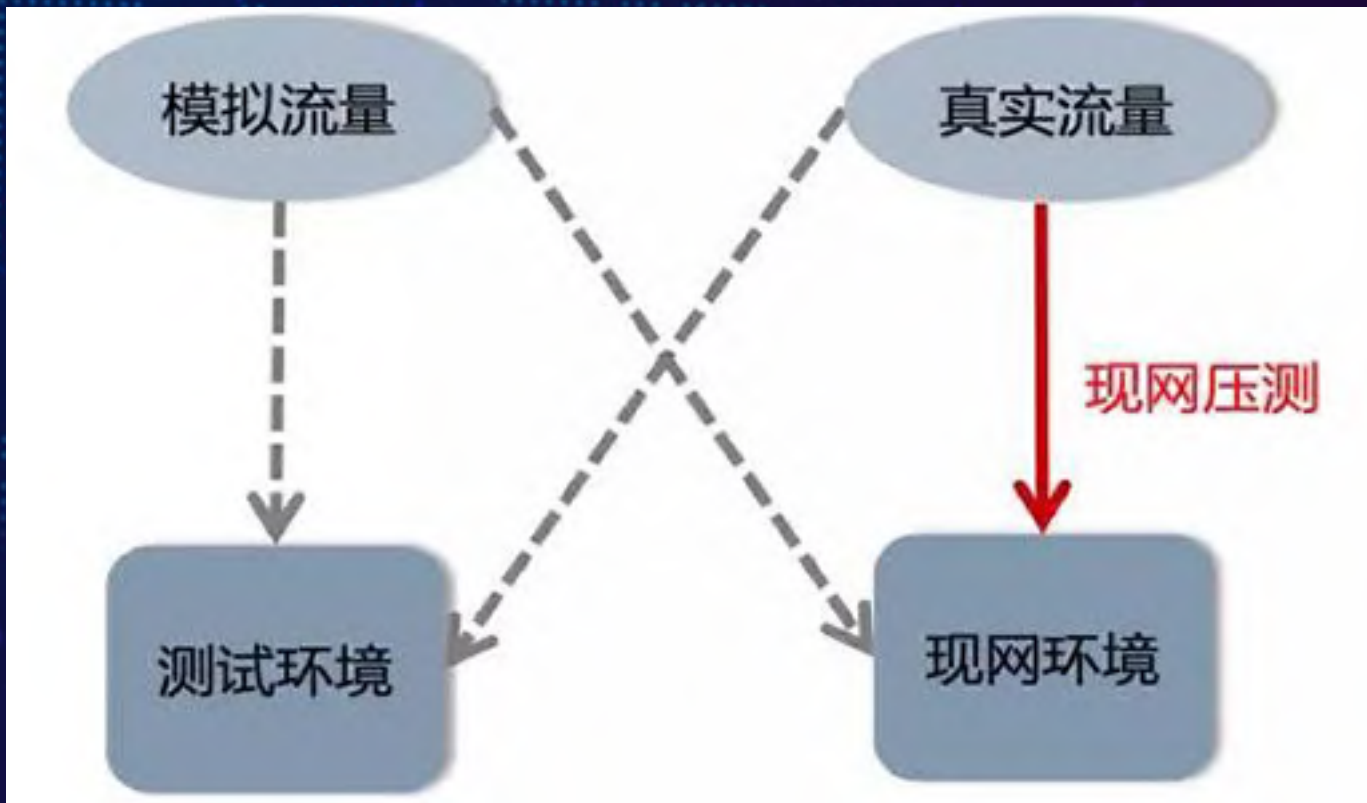


注：本页及后续幻灯部分图片引自吴磊~微信高效运维之路

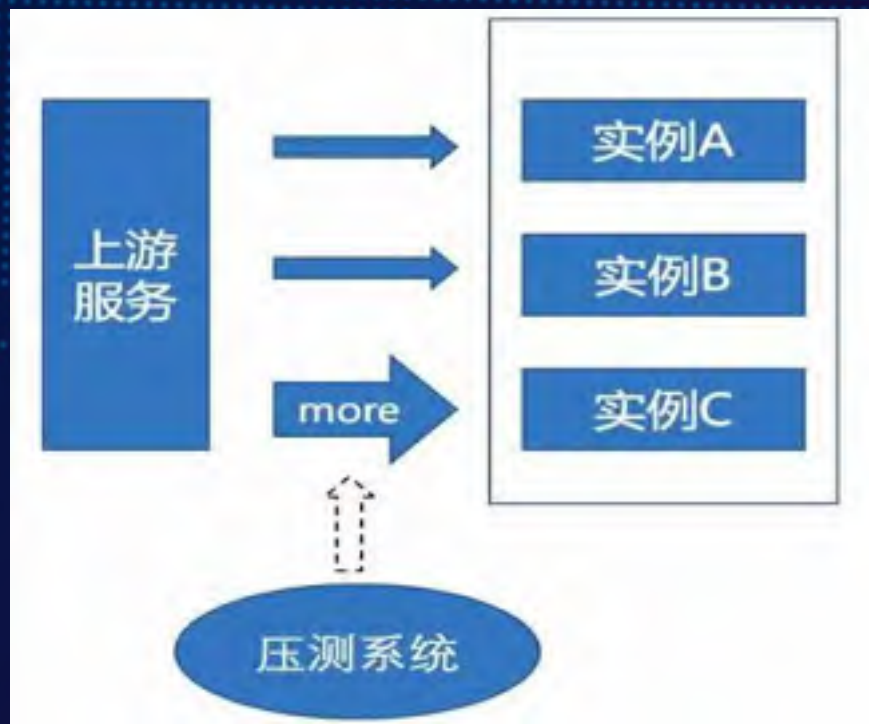
容量管理-容量评估挑战



容量管理-容量评估模型



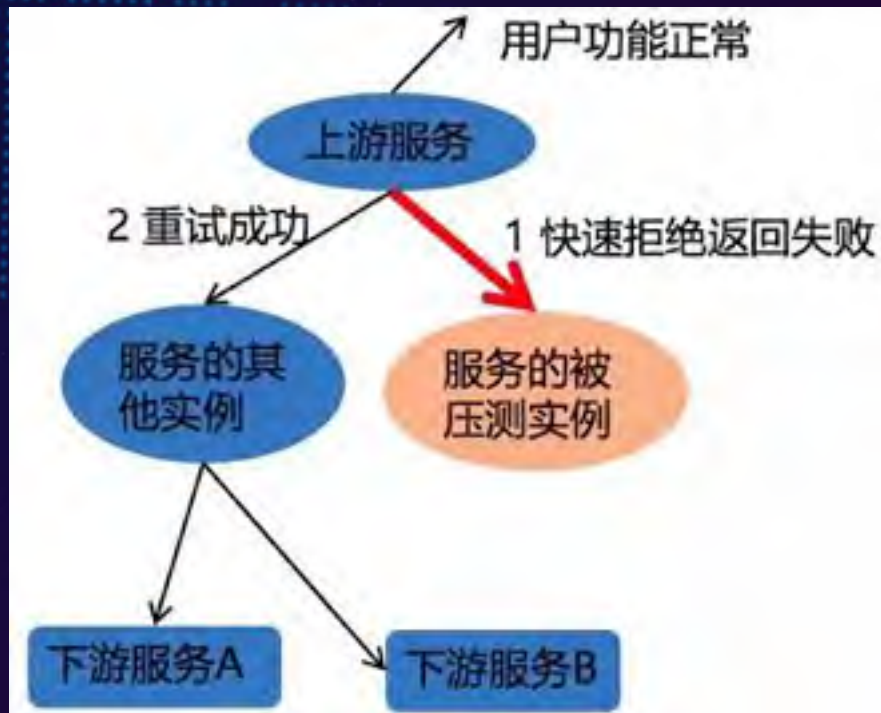
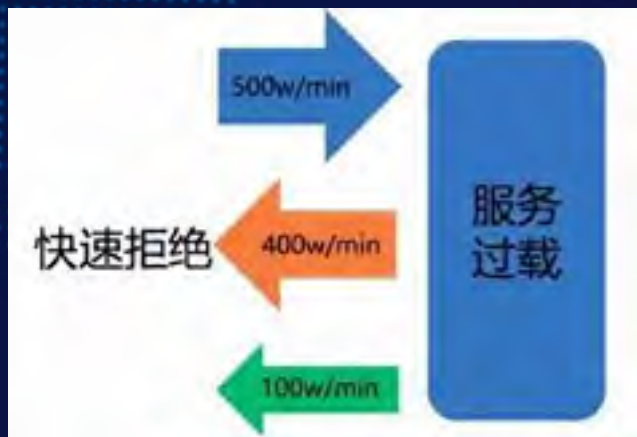
容量管理-现网压测容量评估



解决：

- ✓ 压测引发故障？能否及时发现？
- ✓ 压测引发底层问题？
- ✓ 压测出现故障能否快速恢复？

容量管理-压测微服务自我保护



微服务监控



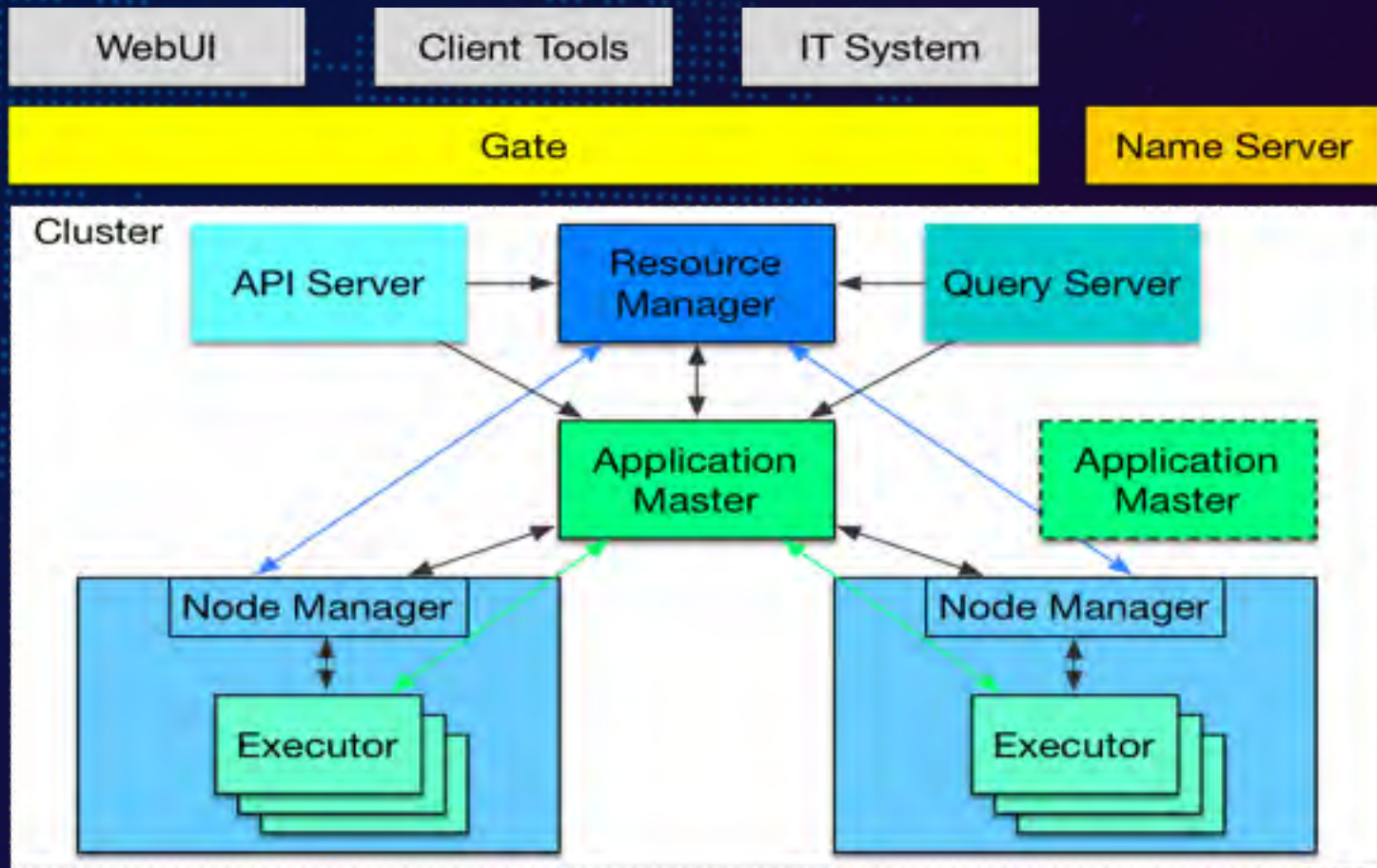
微服务监控-秒级监控能力



容器编排、业务部署与调度-yard

- 基于svrkit框架自研
- 参考borg/yarn/k8s/mesos等主流调度系统
- 覆盖80%+微服务

Yard调度系统架构



容器编排能力开放输出--腾讯云K8S



腾讯云基于K8S 做了什么

控制台

01

- a、集群管理：通过控制台管理集群，不需手动配置搭建
- b、封装服务、配置项、pod等概念封装到控制台
- c、提供服务事件、容器日志和webshell等工具

02

对接腾讯云IAAS层

- a、基于VPC的容器网络，可和外部CDB等资源打通
- b、支持cbs盘的volume和storage class
- c、基于腾讯云LB的ingress controller

03

镜像仓库

- a、仓库优化，包括添加缓存层以及和cos存储的对接
- b、CI/CD：代码源对接github，通过镜像触发器可与企业自建CI流程打通
- c、提供mirror registry

微服务精细化运营

- 容器化混合布署，全盘调度，充分利用资源
- 削峰填谷，最大化资源效率



微服务精细化运营-削峰填谷

- 在线业务削峰/错峰：
 - ✓ 有效识别业务高峰期及配置可错峰微服务
 - ✓ 消峰/错峰微服务在高峰期前获取服务资源，峰期后释放资源
- 离线计算填谷：
 - ✓ 离线任务类别
 - ✓ 离线任务运行时段控制，如：
 - 01:00 ~ 08:00 不限任务
 - 08:00 ~ 20:00 任务入队控制
 - ✓ 离线任务的资源占用控制
 - cpu.shares + memory.limit_in_bytes + blkio
 - ✓ 离线任务的优先级

微服务未来演进



- 微服务及云资源的CMDB如何构建？
- 更酷的微服务管理工具？（如服务可视化、消息跟踪、智能故障检测...）
- 微服务应用商店？

场景决定未来

一切细微之处，皆是演进之端

云端让业务价值无限延伸



腾讯云

Tencent 腾讯

Thank you!