

WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE



出品人及主持人：

李庆丰

新浪微博研发中心
研发总监

容器技术实践

微博混合云DCP平台介绍 与业务上云实践

付稳 @it_fuwen



付稳

新浪微博
技术专家

分享主题：

新浪微博混合云DCP平台介绍
与业务上云实践

分享主要内容

- **一、背景、挑战**
- **二、Weibo DCP设计与实现**
- **三、基础设施**
- **四、弹性调度**
- **五、春晚实战与总结**

一、背景、挑战

微博业务挑战



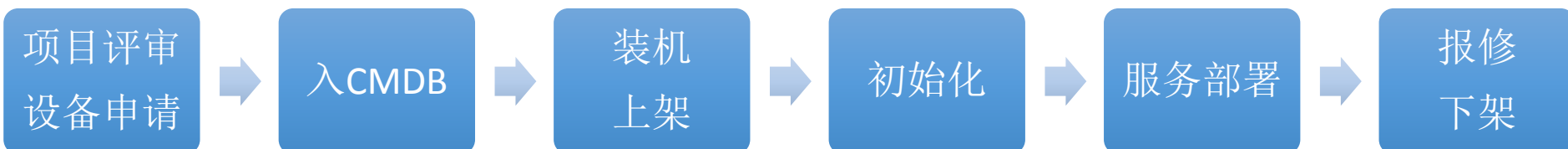
十亿级PV、千亿级数据、2000台以上的服务器规模、20个以上的大小服务模块、百亿级数据HBase存储、千台以上的Docker混合云集群，持续不断的技术挑战；

微博业务现状

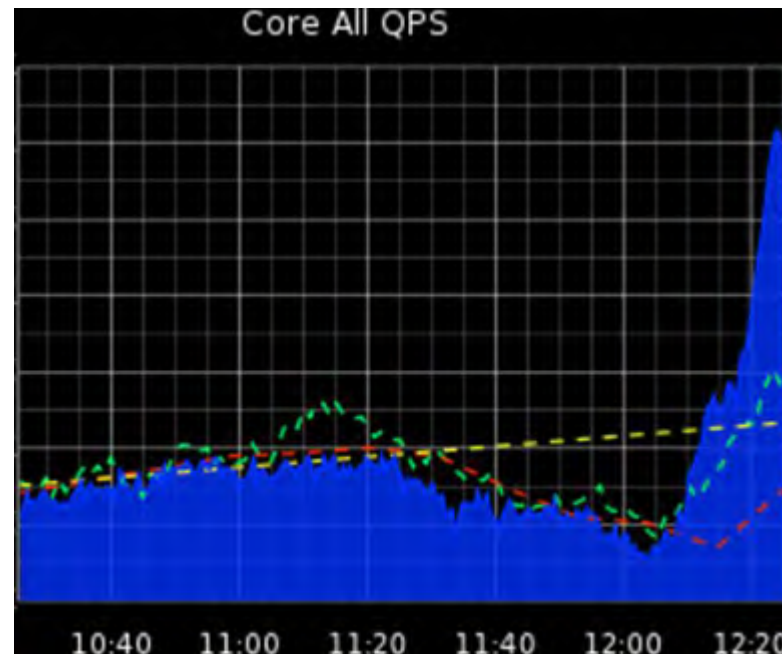
- 春晚峰值流量应对
 - 机架位不足，上千台服务器库存不足
 - 千万级采购成本巨大
 - 采购周期长，运行三个月只为一晚
- 白百合、李晨娱乐事件等热点突发峰值应对
 - 突发性强无预期、无准备
 - PUSH常规化，短时间大量设备扩容需求

如何10分钟内完成1000节点扩容能力？

服务扩缩容流程繁琐

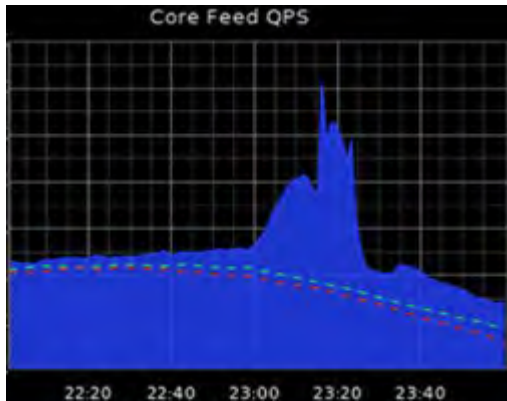


明星“出轨、在一起！”



微博业务现状与解决

弹性快速扩缩容



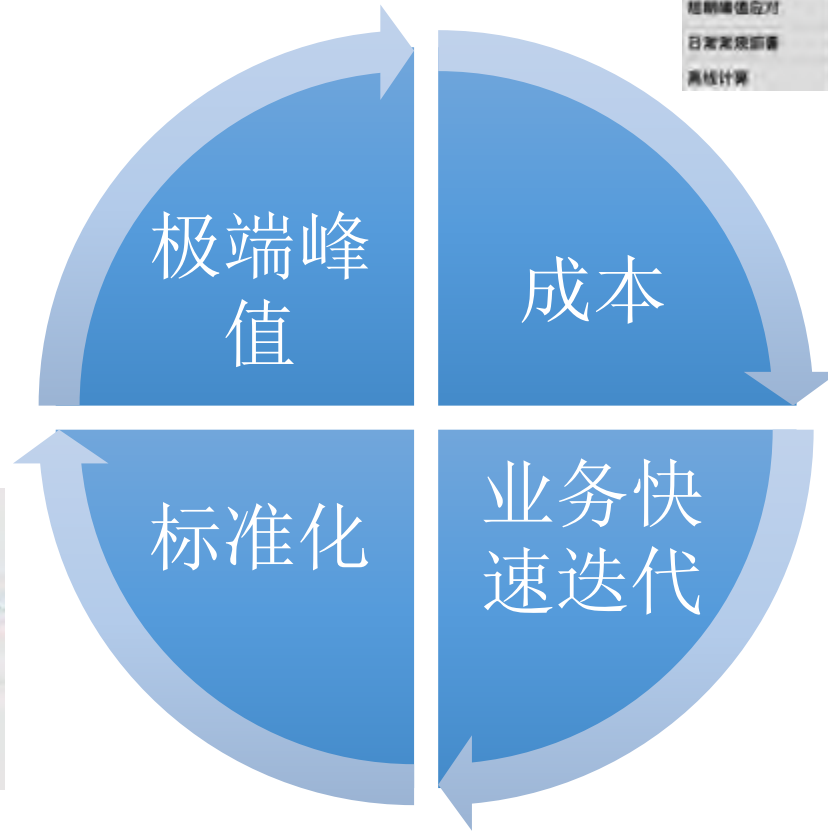
运维标准化

混合云弹性调度可伸缩业务成本节省数倍

场景	私有云方案	公有云方案	预估总成本对比 (私/公)
短期峰值应对	部分调整, 部分采购	按小时付费	数十倍
日常常规部署	按月摊销	包月付费	1.0-1.2
离线计算	按月摊销	按小时付费	数十倍

配置	按量 (元/小时)	包月 (元/月)
4核4G, 20G磁盘	1.13	302
4核8G, 20G磁盘	1.77	402
4核16G, 20G磁盘	3.05	602
8核8G, 20G磁盘	2.25	598
8核32G, 20G磁盘	6.09	1198
16核16G, 磁盘20G	4.52	1207
16核64G, 磁盘20G	12.17	2390

备注: 小时计费单位为包月3-4倍 (64G必须选择16核)



产品更新迭代快, 系统变更代码指数增长

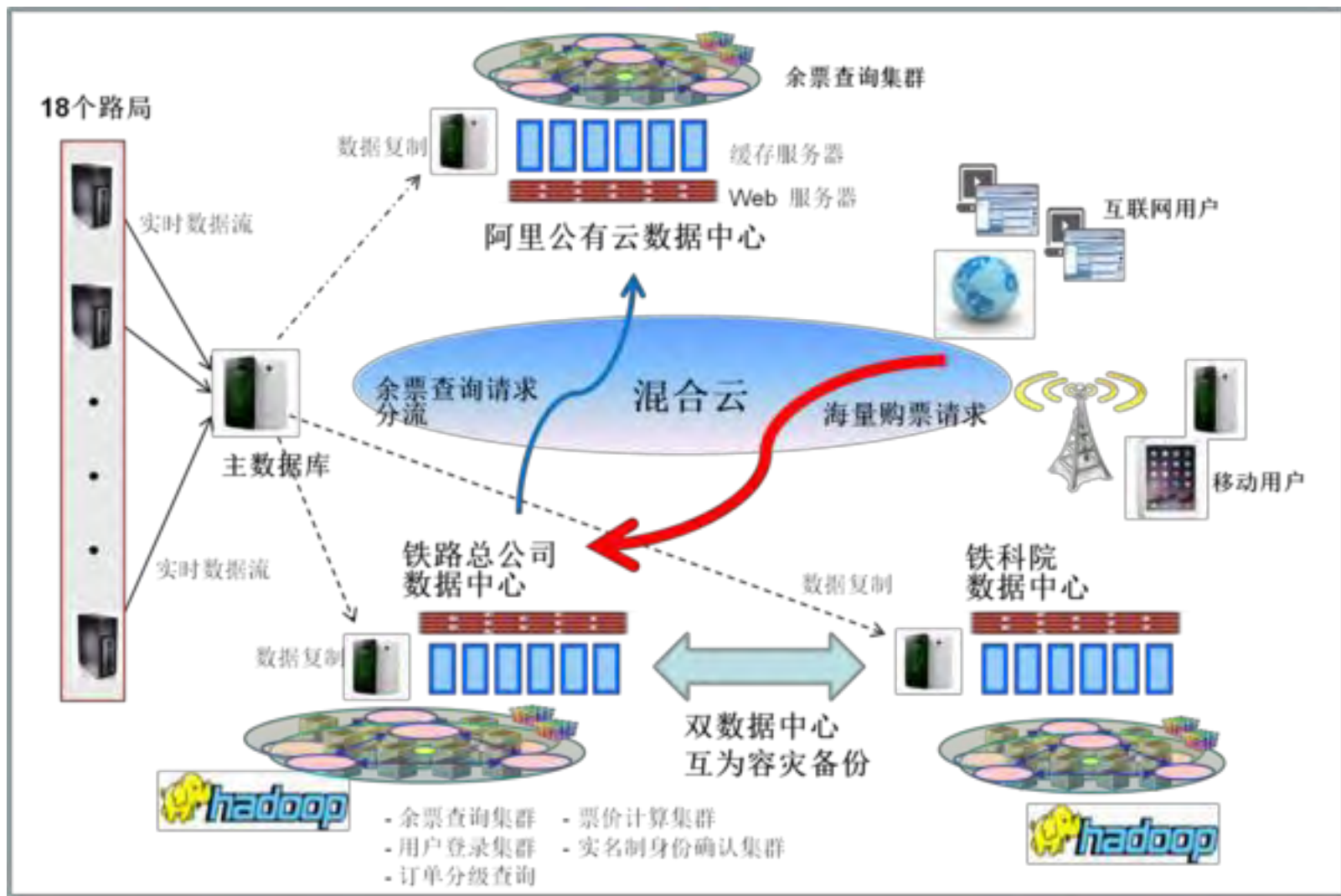
混合云趋势

混合云趋势：安全、可扩展性、成本...

- 阿里云、AWS等公有云平台趋于成熟
 - 国外Zynga、Airbnb、Yelp等使用AWS进行部署
 - 国内阿里云12306、高德、快的已部署，陌陌等部署中
 - 12306借助阿里云解决饱受诟病的春节余票查询峰值问题
- Docker、Mesos等容器新技术使大规模动态调度成为可能
 - 京东618大促借助Docker为基础的弹性云解决峰值流量问题

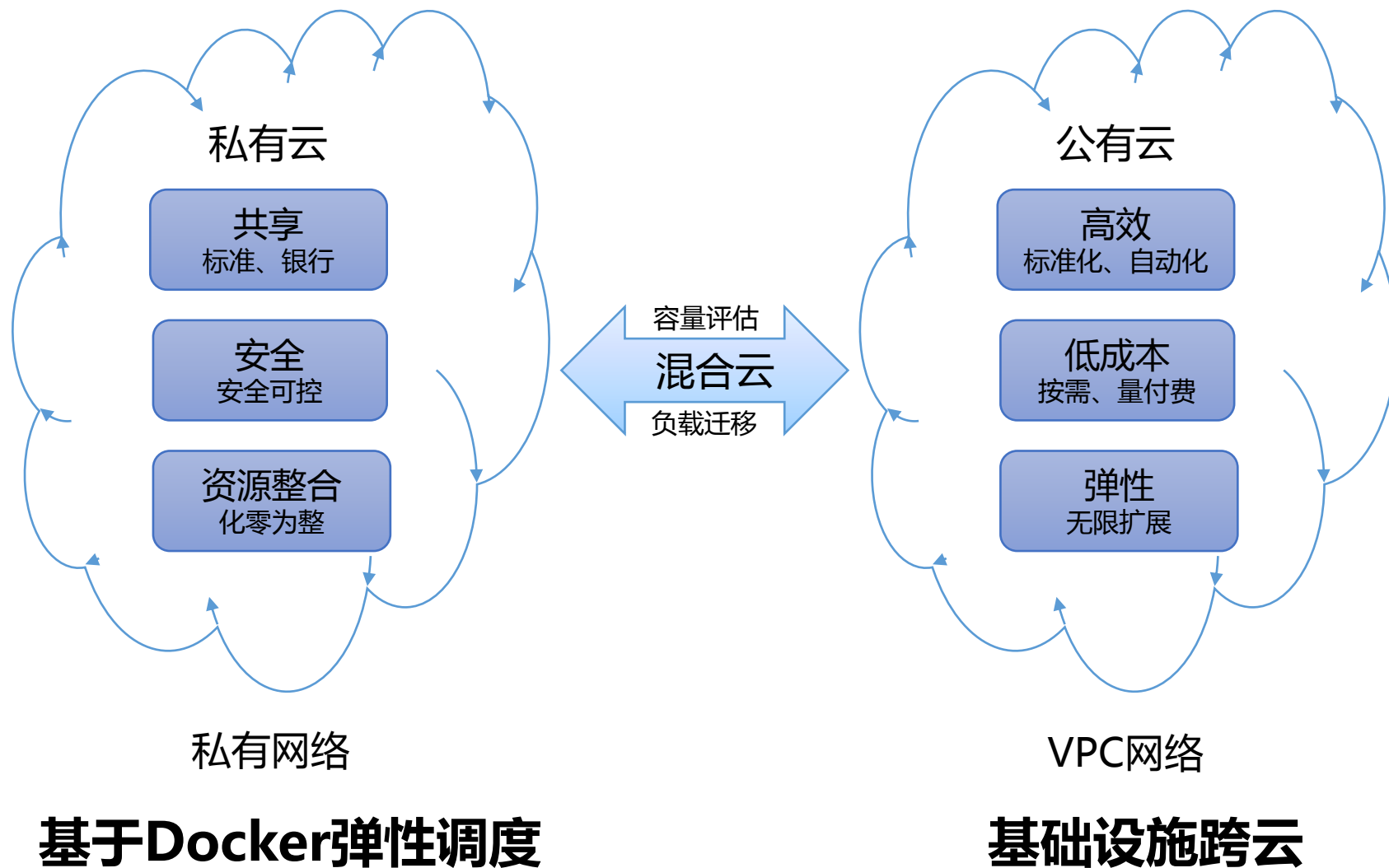


12306混合云案例



12306 两地三中心 混合云架构

混合云核心点

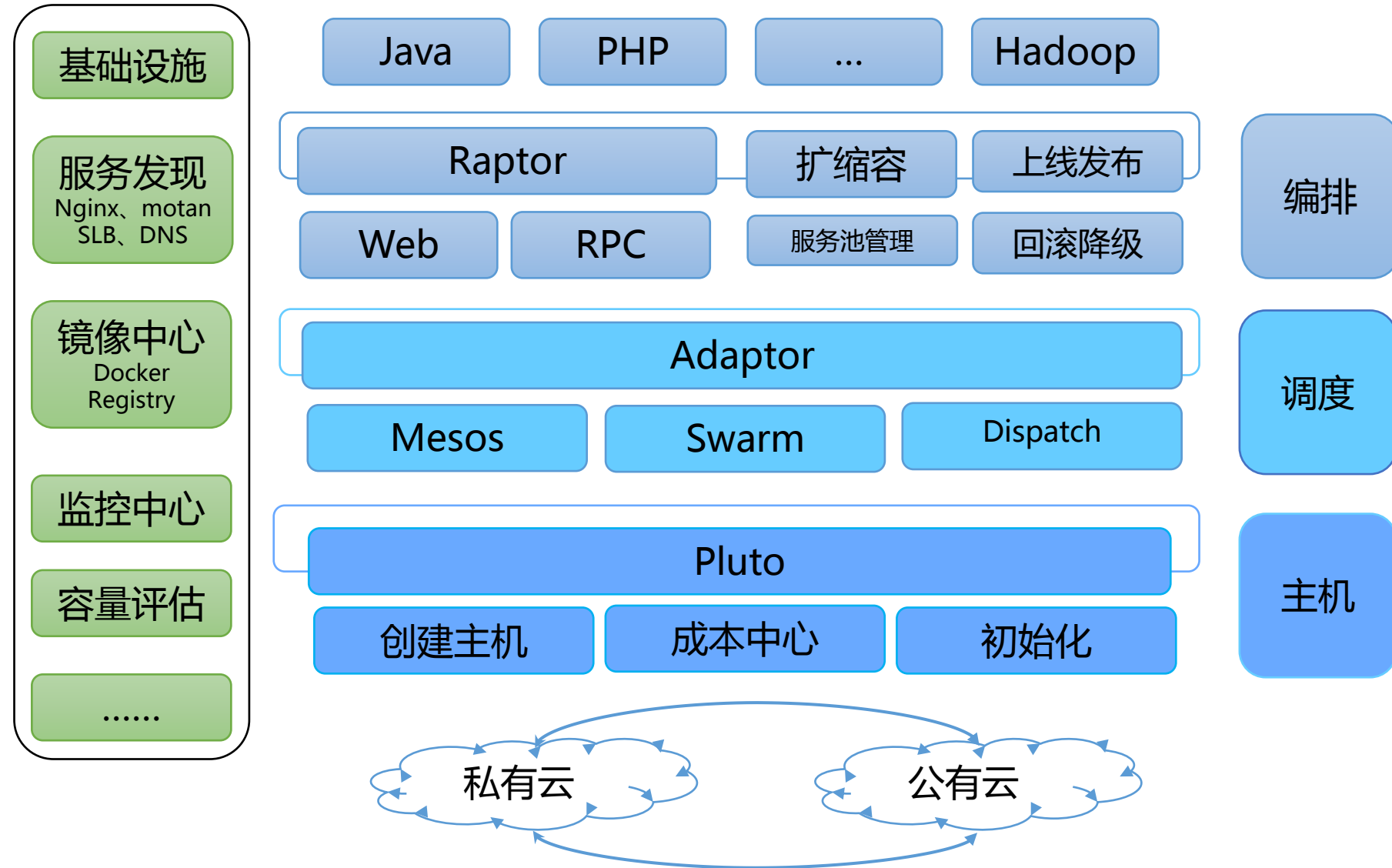


二、Weibo DCP设计与实现

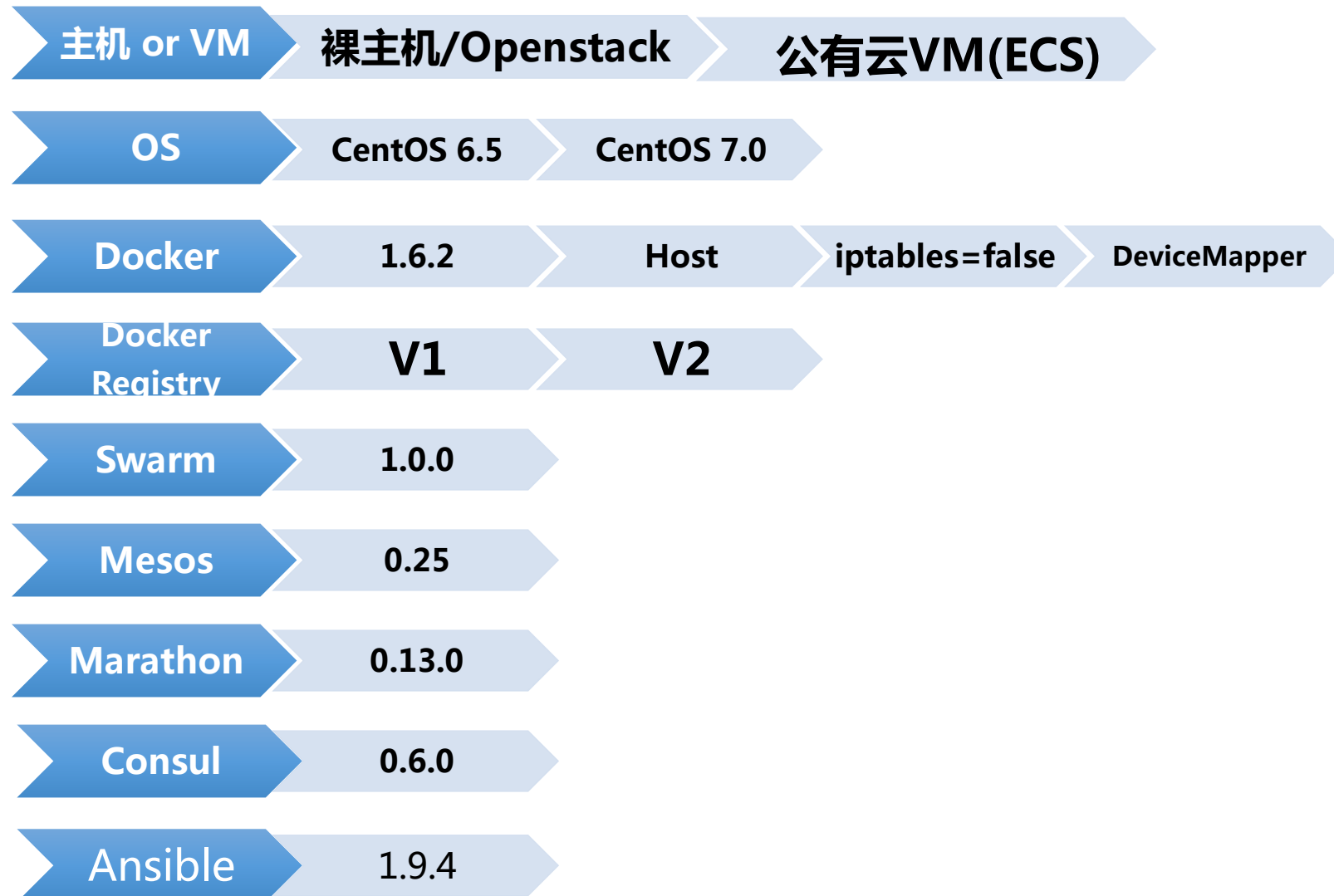
微博DCP技术架构演进



混合云DCP技术架构



混合云DCP技术栈

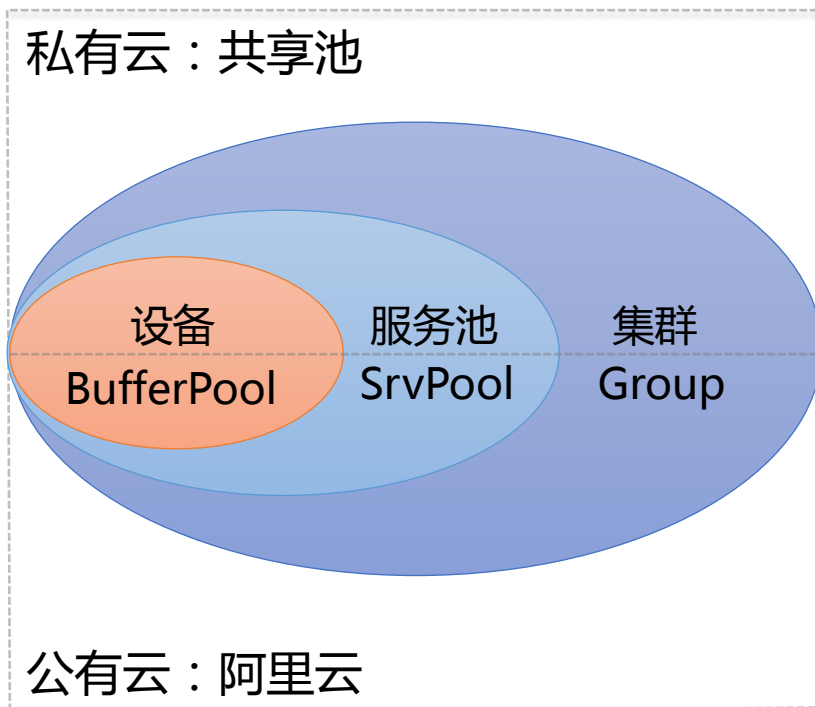


混合云DCP功能模块

业务方	红包飞	MAPI	广告	有信	Feed
	用户	通讯	平台架构		
P A A S	Swarm	Docker 调度策略	Mesos 调度管理	容量评估	Docker Register
	容器监控	调度监控			Docker 镜像市场
I A A S	共享池 管理	成本核算	ECS管理	SLB 管理	Consul 工具管理
	四七层解 决方案	专线保障	公有云 流量管理	审批流程	OS升级 自动化
基础 框架	软件安装	工程框架	安全保障	账户体系	Docker 工具体系
	监控体系	DNS管理	配置管理	阿里云 Yum/日志	

DCP-核心设计思想

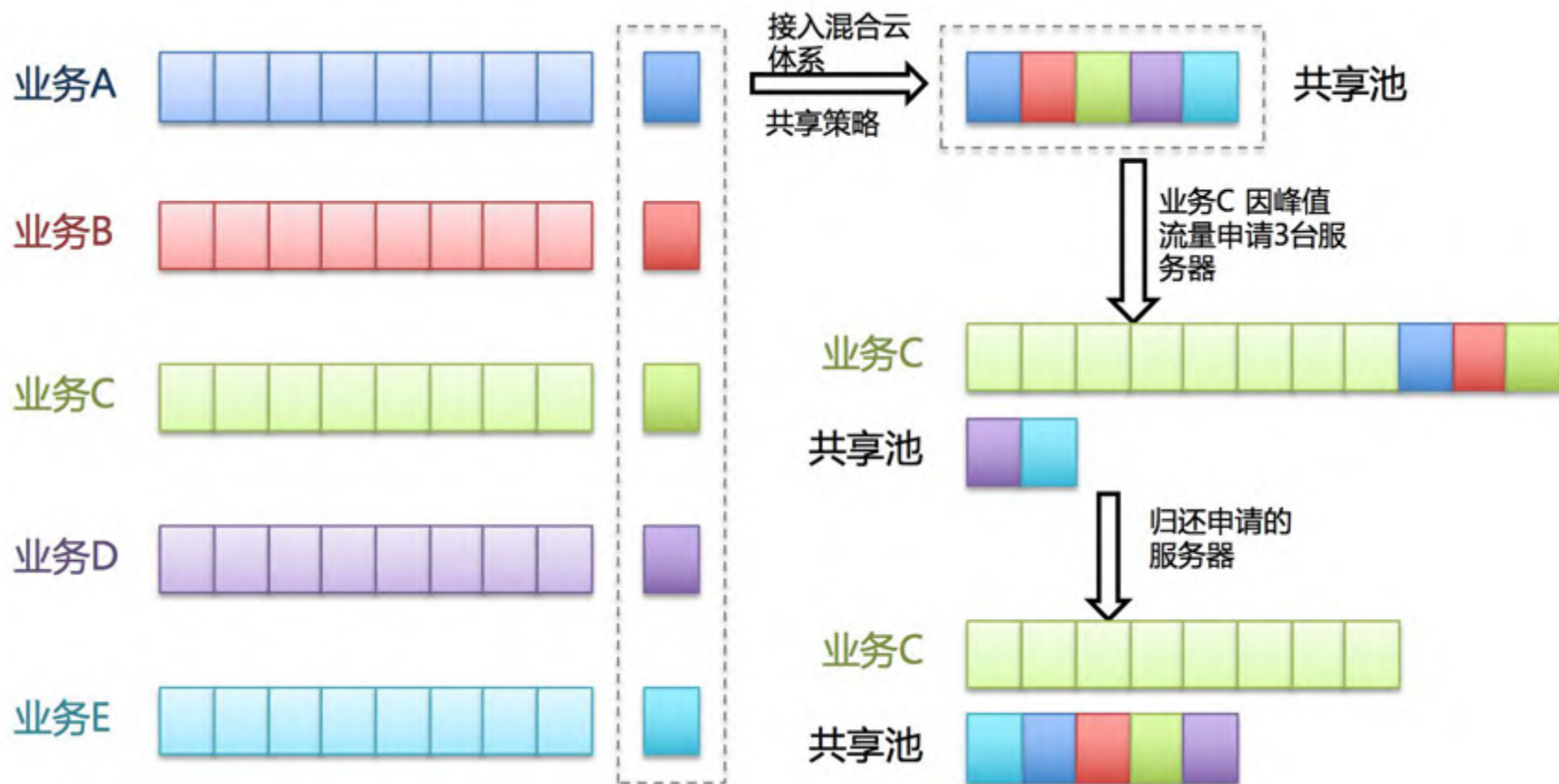
- 核心问题：设备从哪来？
- 设备方案：内网共享池 + 公有云=BufferPool
- 服务：IP + Port



层级关系

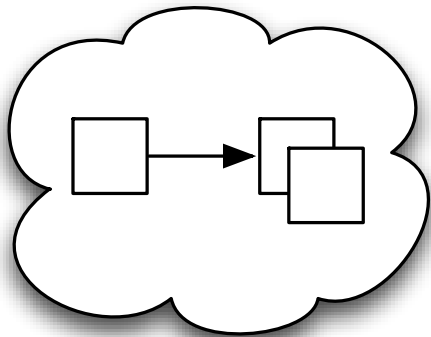
- DCP：分为多个集群
- 集群：为独立平台，对应业务线
 - 集群内：自由调度（跨池）
 - 集群外：配额调度
- 服务池：同一业务线的同构服务
- 设备：buffer池 = 共享池 + ECS

DCP-资源共享

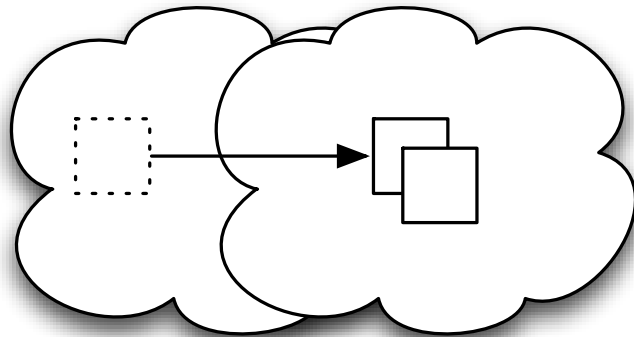


DCP-大规模集群扩容方式

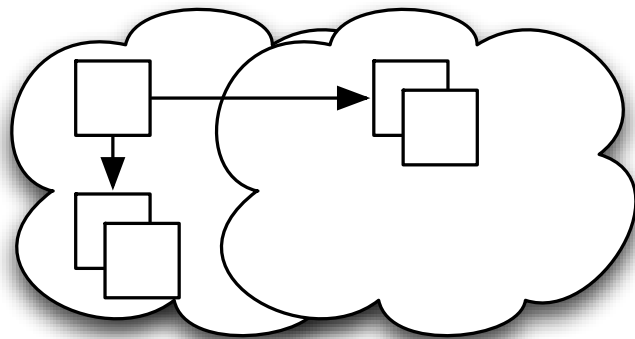
私有内弹性扩容



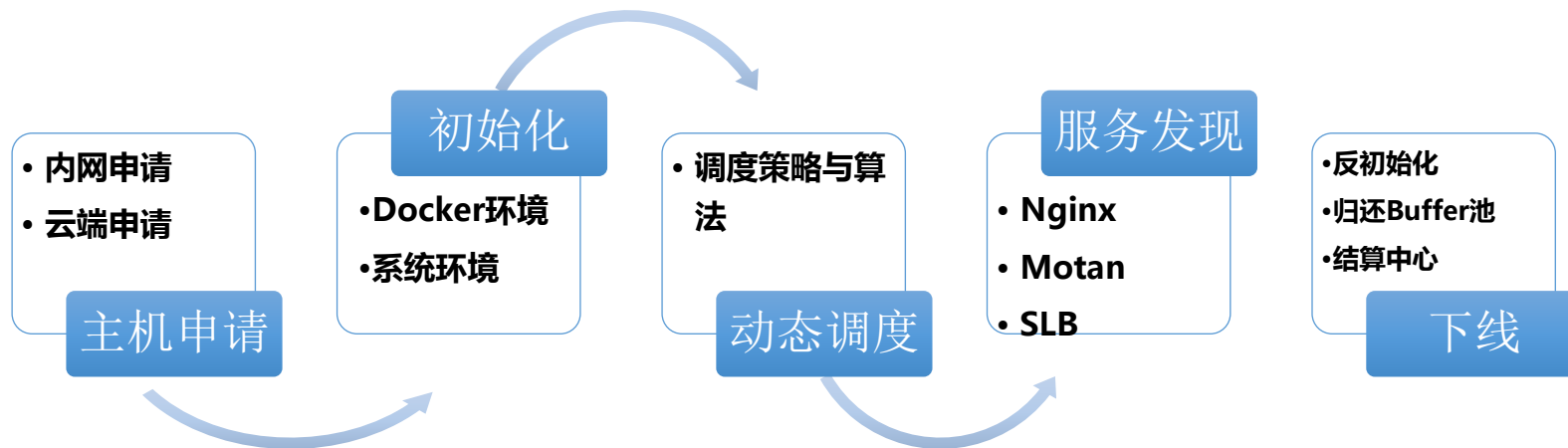
扩容到公有云弹性



私有云、公有云同时弹性扩容

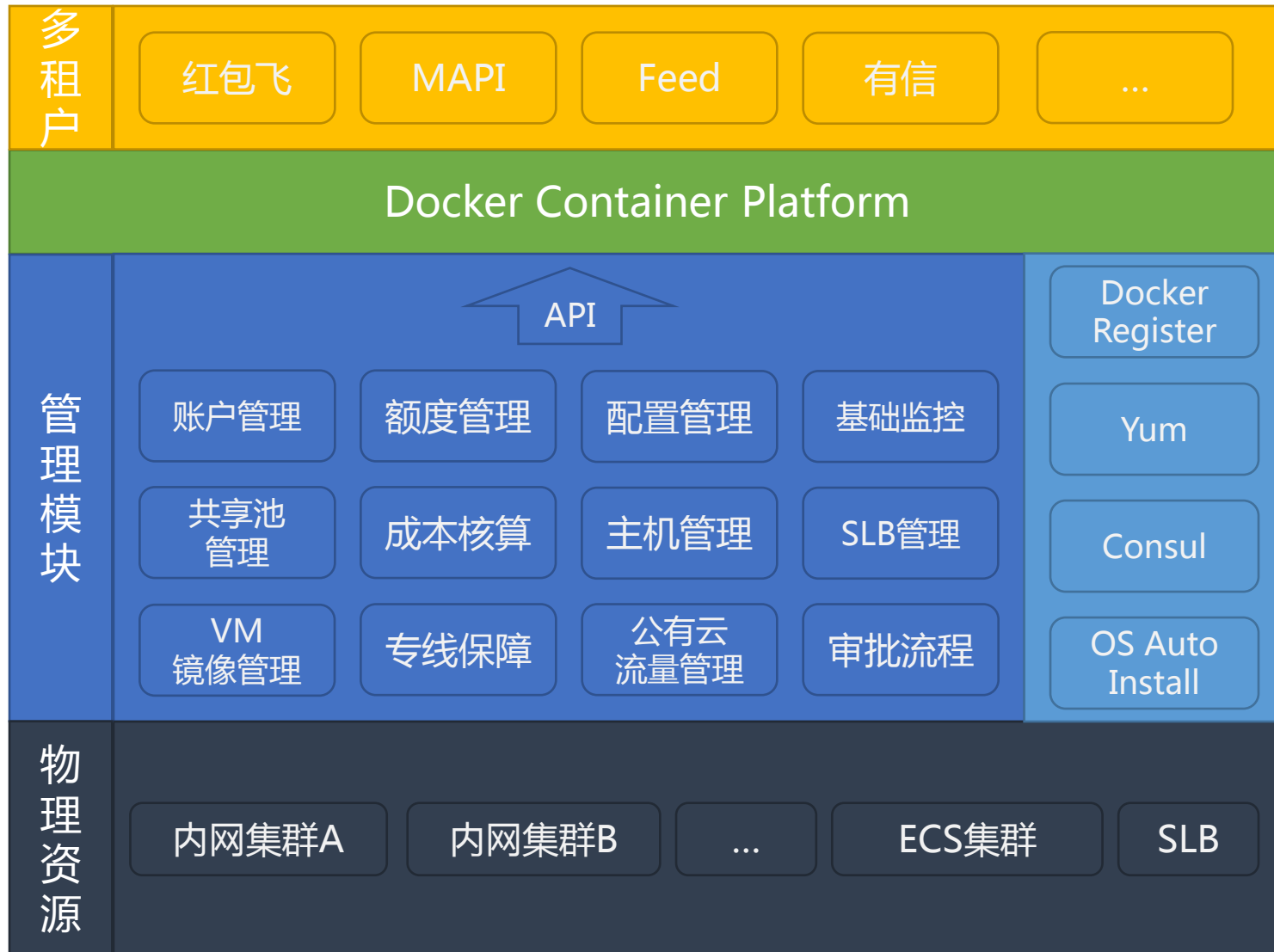


混合云DCP流程

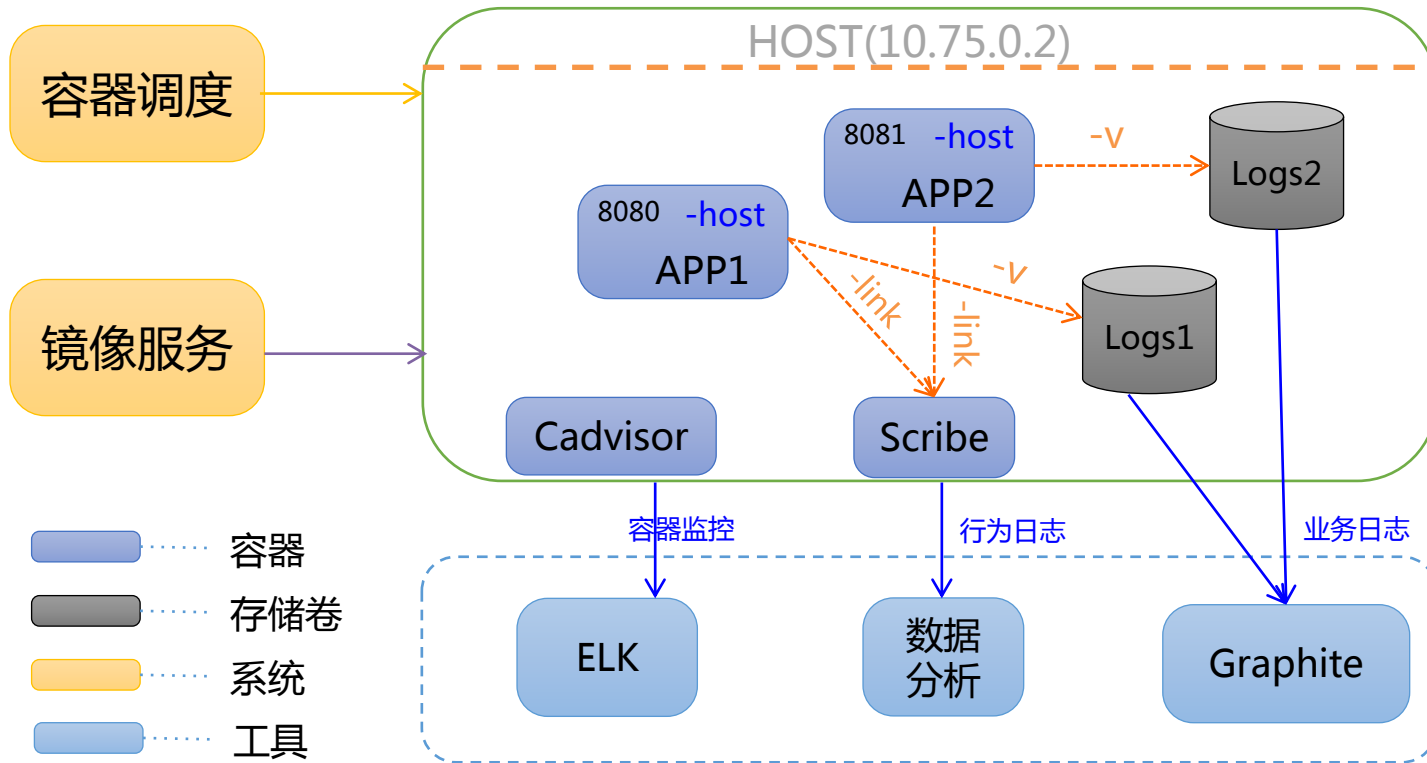


二、基础设施

统一资源管理



单机部署方案



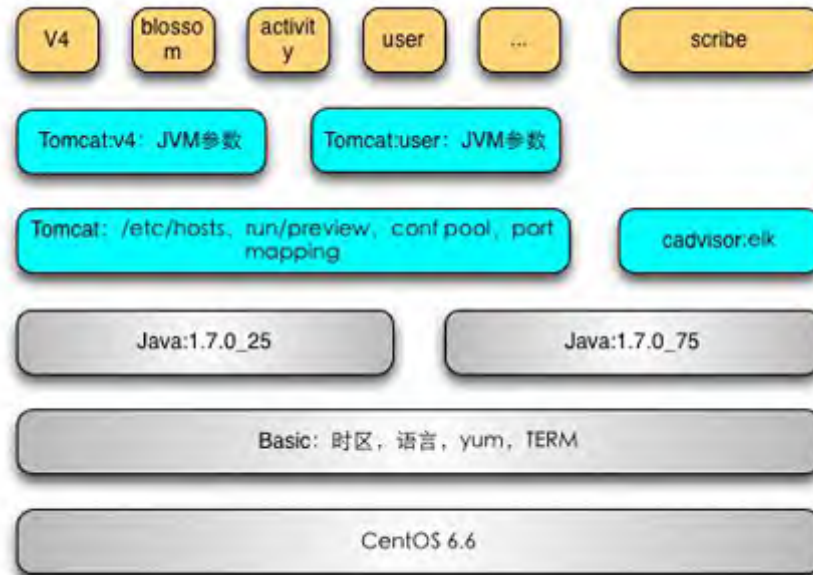
基础环境软件版本

Java		php	
cAdvisor 0.7.1.fix			
Mesos 0.25			
Swarm 1.0.0			
Consul 0.5.2			
Docker 1.6.2	Registry v2	Daemon Wrapper	Registry v1
devicemapper-direct-lvm		Docker 1.3.2	
CentOS 7.1.1503/3.10.0-229.el7.x86_64		devicemapper-loop-lvm	
		CentOS 6.6/2.6.32	

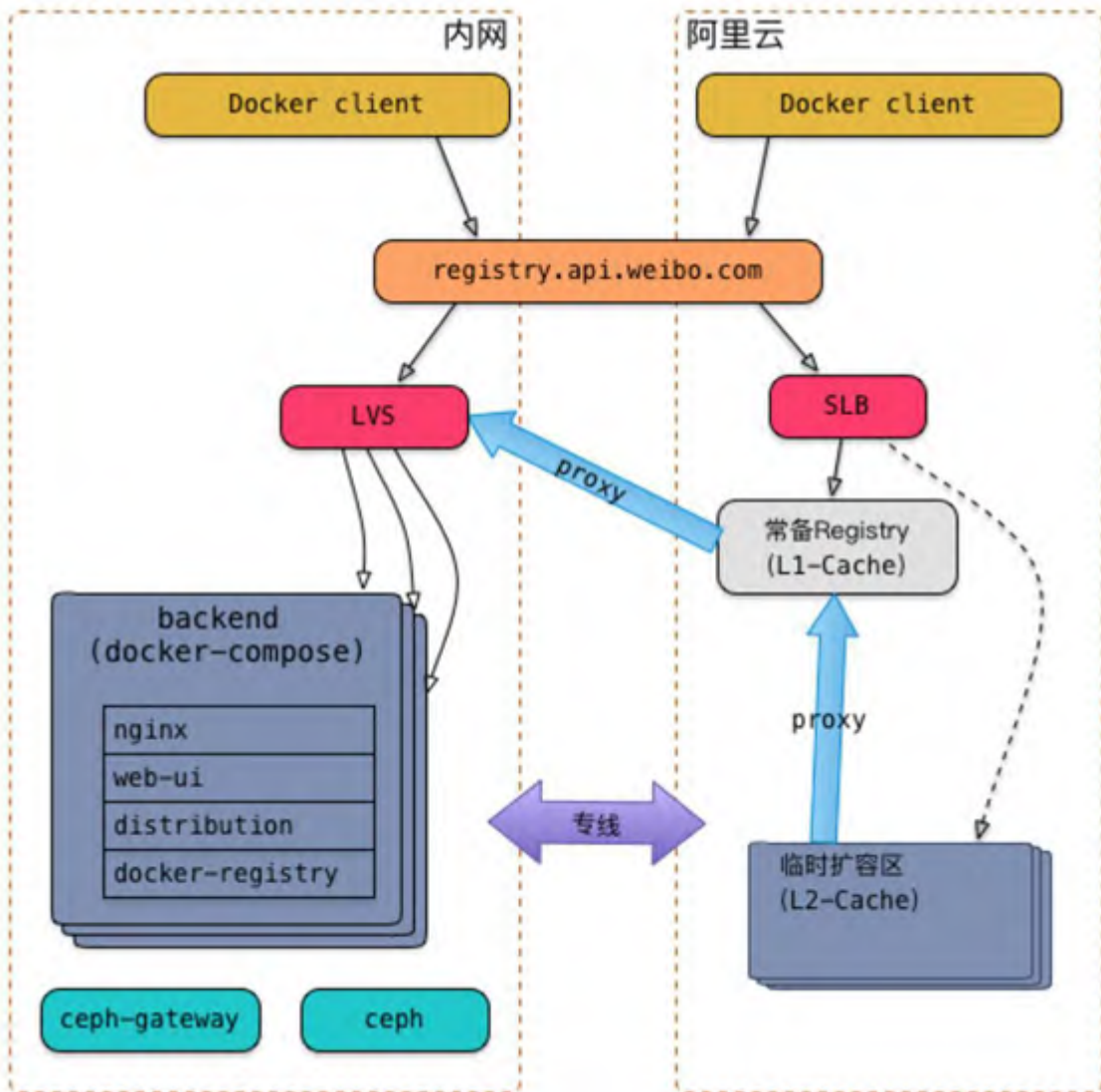
镜像分层服务

● 镜像服务

- 分层设计，逐层复用
- 基础环境/运行时/容器/业务
- 优化大小，dockerignore
- 禁止使用latest

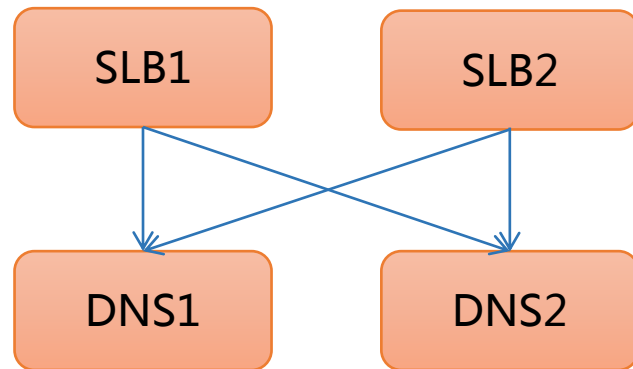


Docker Registry – 部署架构

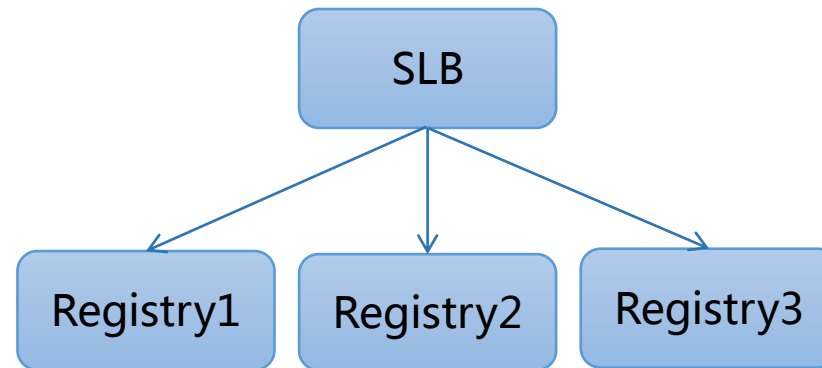


DCP中SLB的应用

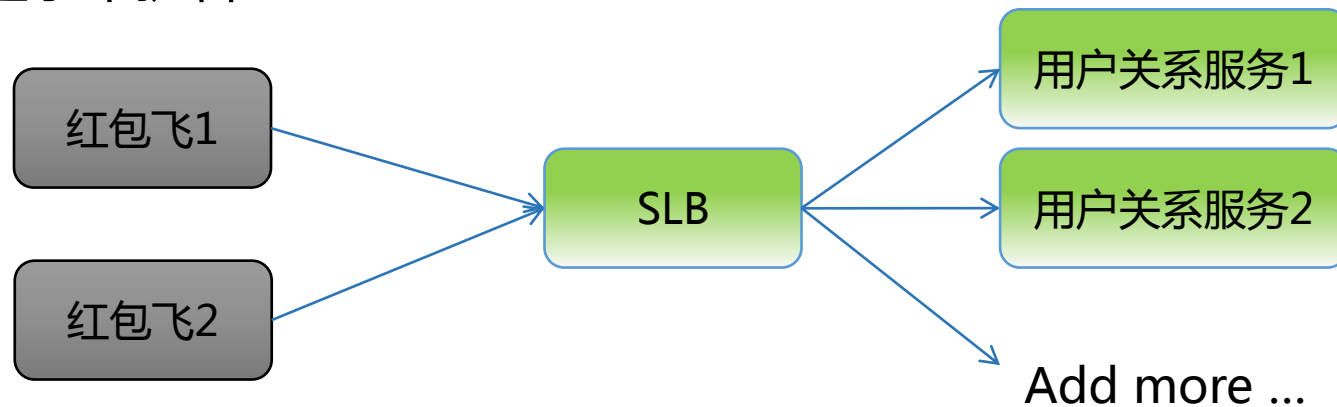
- DNS高可用



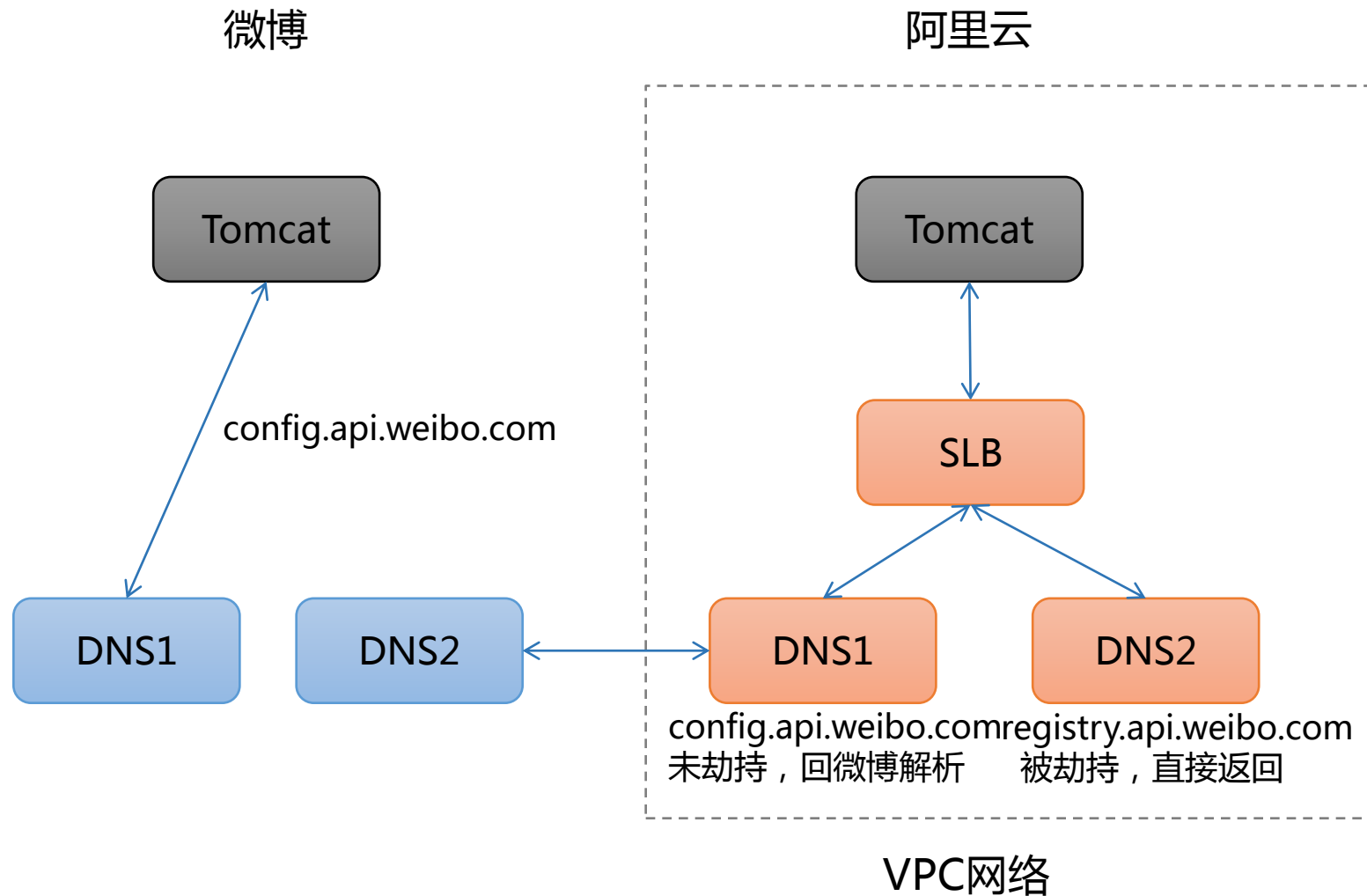
- Registry负载均衡



- 快速水平扩容

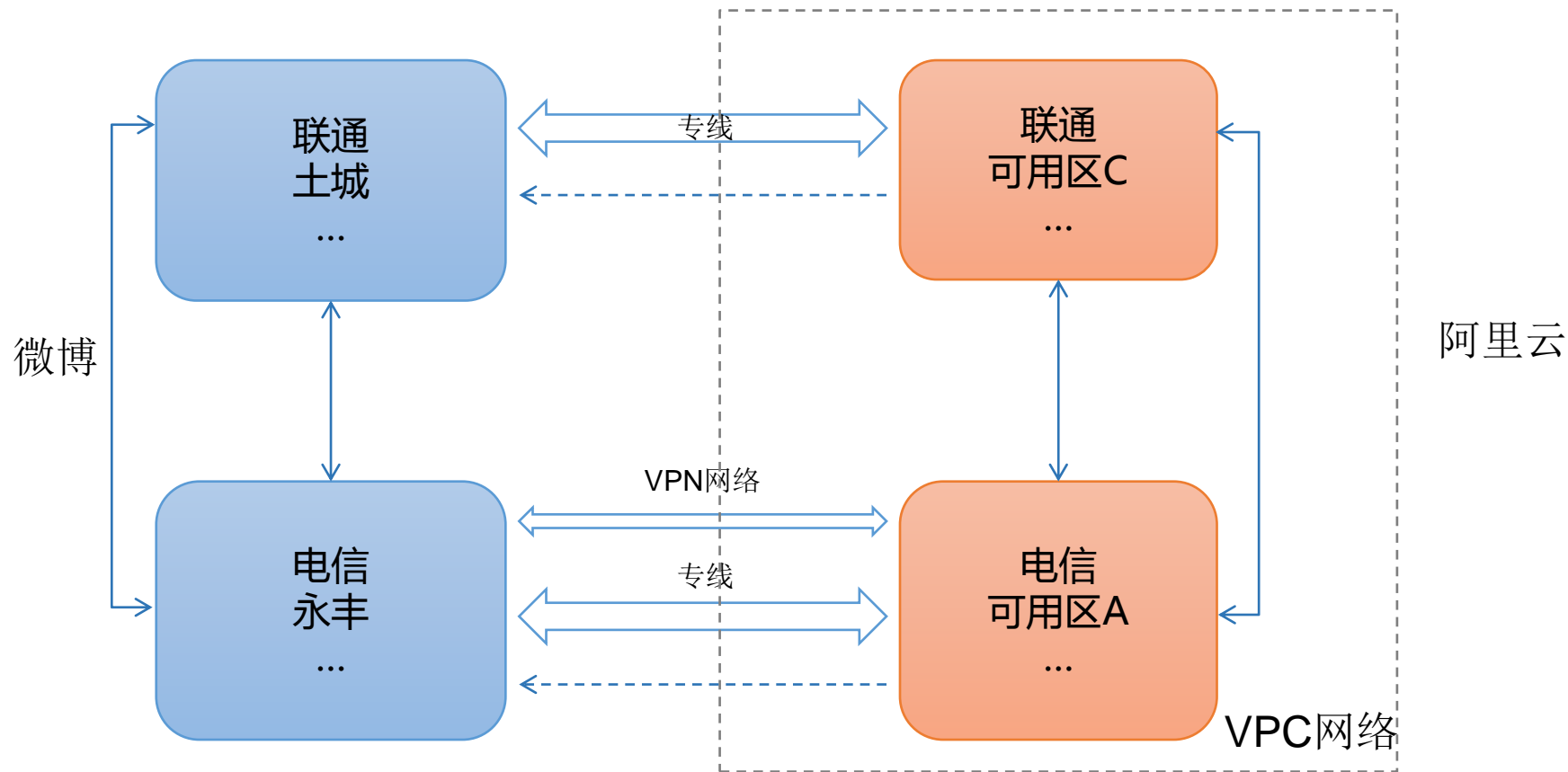


DNS智能解析



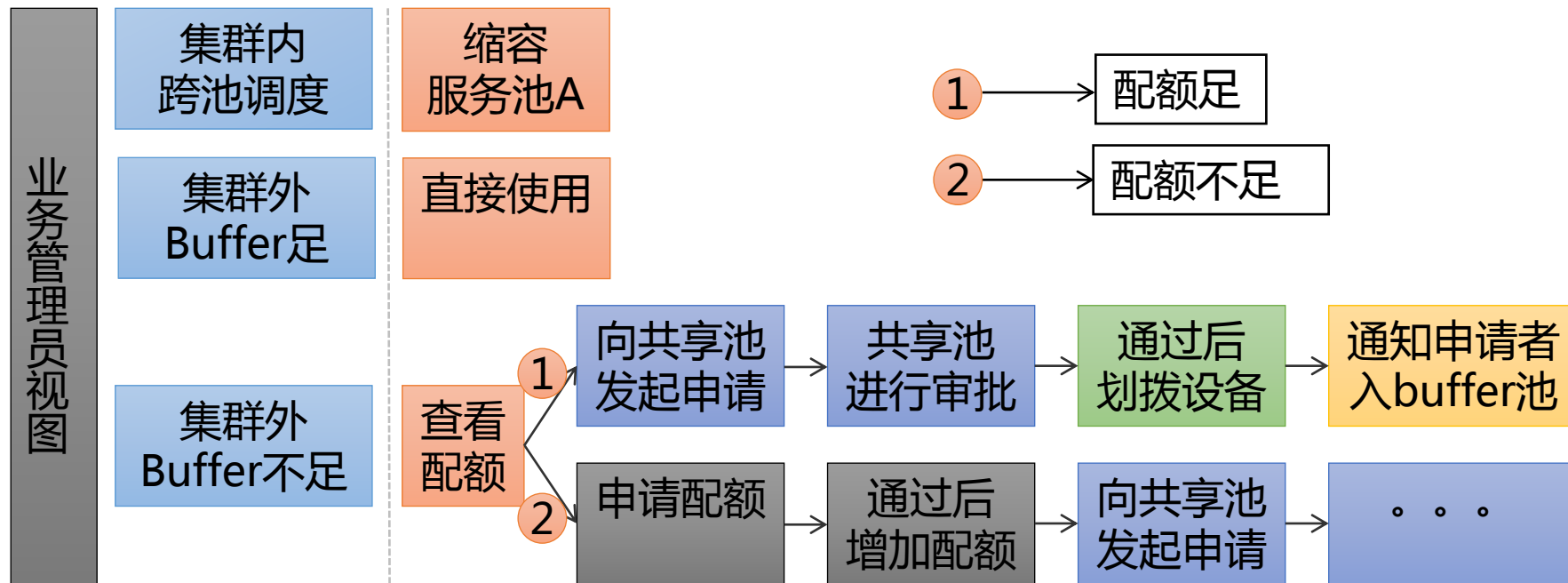
专线网络架构

- 通过路由配置分散两条专线压力，可随时切换
- VPN做备用
- 不同业务划分网段，便于监控专线带宽使用情况



DCP-弹性扩容第一步：主机申请

- 私有云：共享池（离线集群，低负载集群，错峰）
- 公有云：阿里云（动态创建）



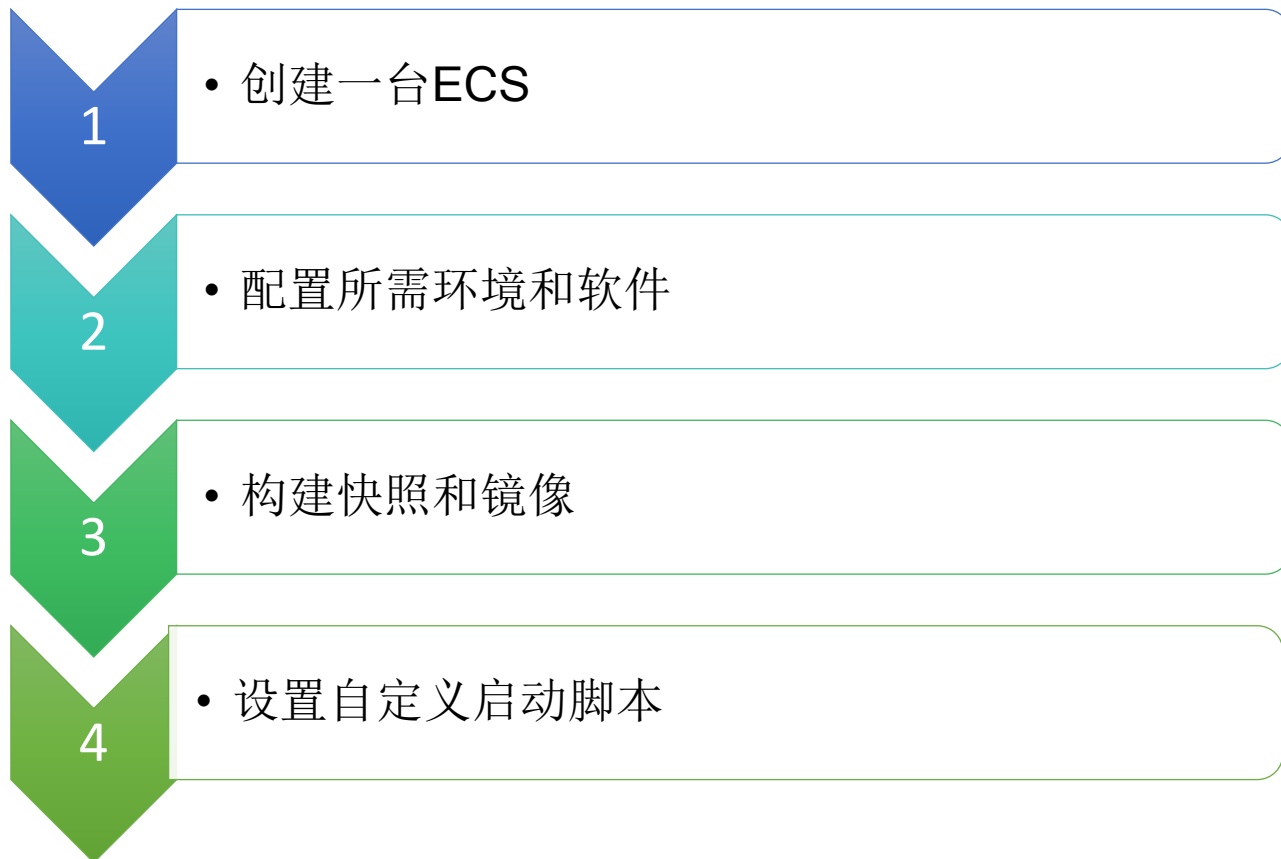
ECS批量创建

- 封装阿里云接口
 - 阿里云golang sdk
 - goroutines并发
 - chan异步调用
 - websocket和前端同步状态

- 遇到的问题
 - ECS前后端（瑶池和后羿）状态不同步，sleep解决
 - 并发数限制，单vpc下有锁，50秒拿不到锁即超时

VM镜像

- 各业务方管理自己的VM镜像，加快构建环境的速度



DCP - 设备申请案例：阿里云主机

交易单

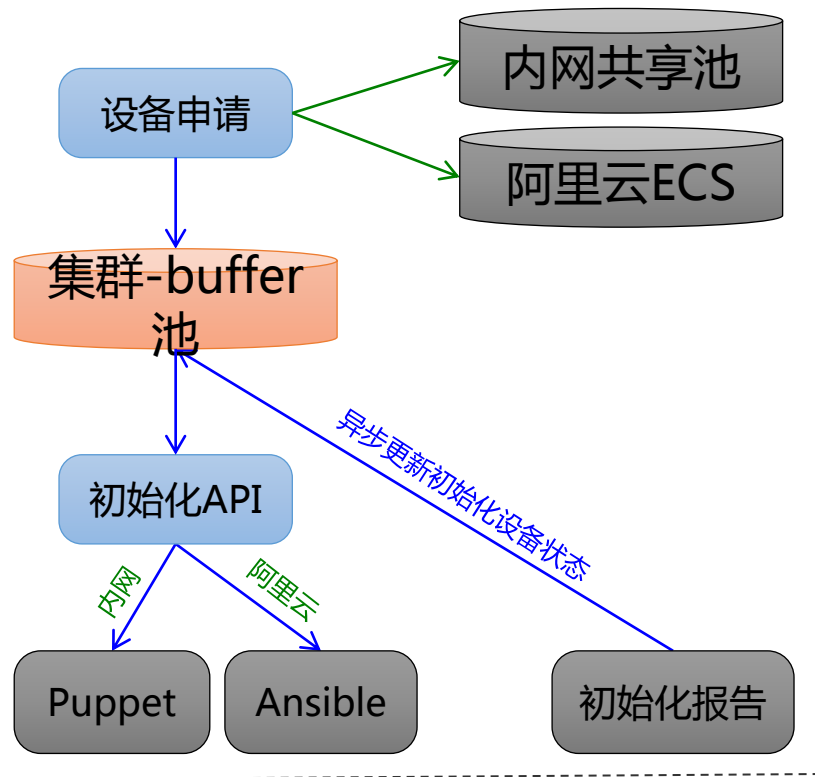
#	交易单	发起方	接收方	原因	交易数量	机型	CPU	MEM	开始时间	结束时间	状态	#
---	-----	-----	-----	----	------	----	-----	-----	------	------	----	---

发起申请

发起方	<input type="text" value="WeiboPlatform_Platform"/>	可用地域	<input type="text" value="北京地区"/>
可用区域	<input type="text" value="北京可用区C"/>	VPC	<input type="text" value="weibo_vpc"/>
交换机	<input type="text" value="Weibo_Switch_C_1"/>	安全组	<input type="text" value="beijing-a-default"/>
机器规格	<input type="text" value="16Cores-16GB"/>	CPU	<input type="text" value="16"/>
内存	<input type="text" value="16"/>	cmdb服务类型	<input type="text" value="JAVA应用"/>
申请机器数量	<input type="text" value="100"/>	镜像	<input type="text" value="weibo_hongbao_v5_4"/>
结束时间	<input type="text" value="2016-01-21T01:29:00Z"/>		

DCP-弹性扩容第二步：初始化

DCP初始化流程



1. 运维环境安装
2. Docker环境安装

配置管理



- 微博内网
 - 已有模块稳定
 - 无法SSH

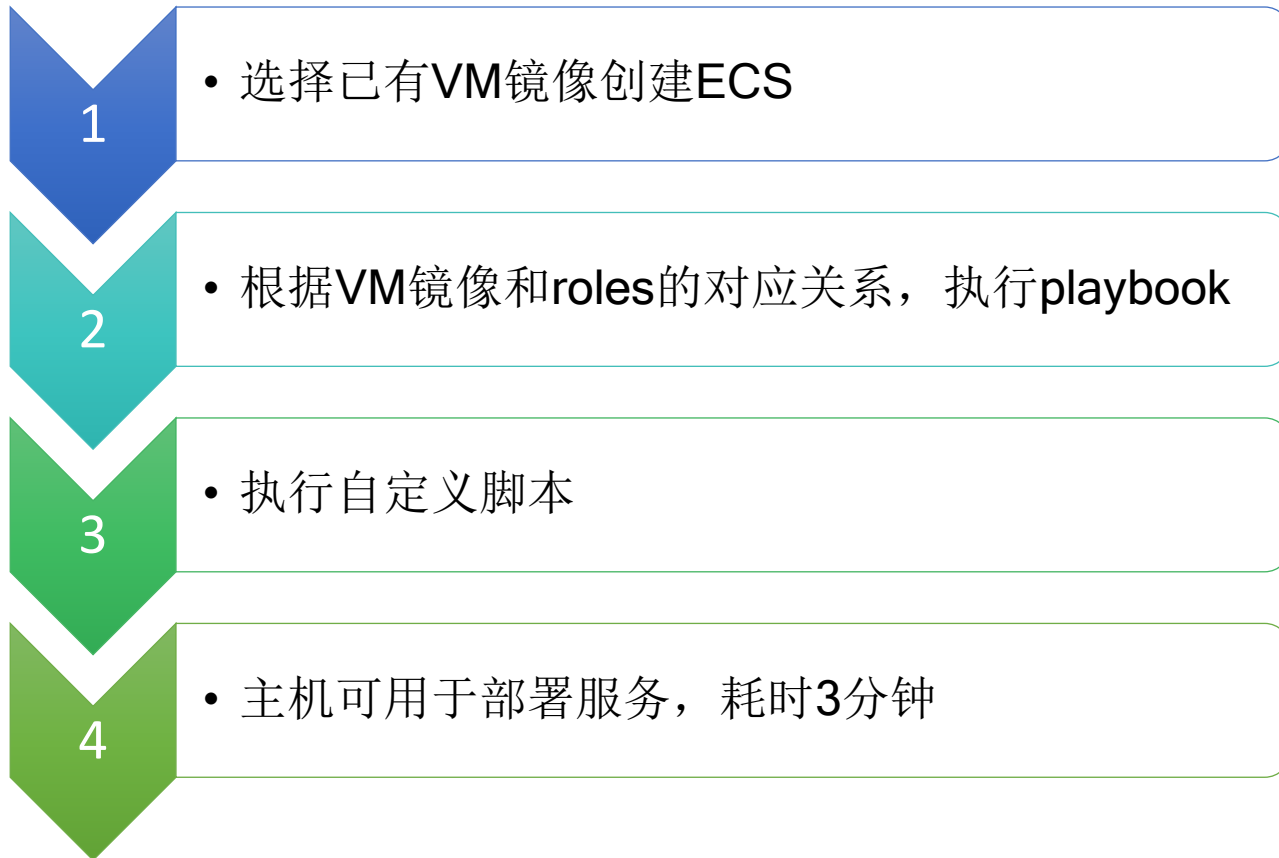


ANSIBLE

- 阿里云
 - 依赖少
 - 集成SSH登录权限
- 性能问题
 - 异步队列
 - 高并发下水平扩容
 - 分布式改造

配置初始化

- ECS创建完成后，自动执行初始化



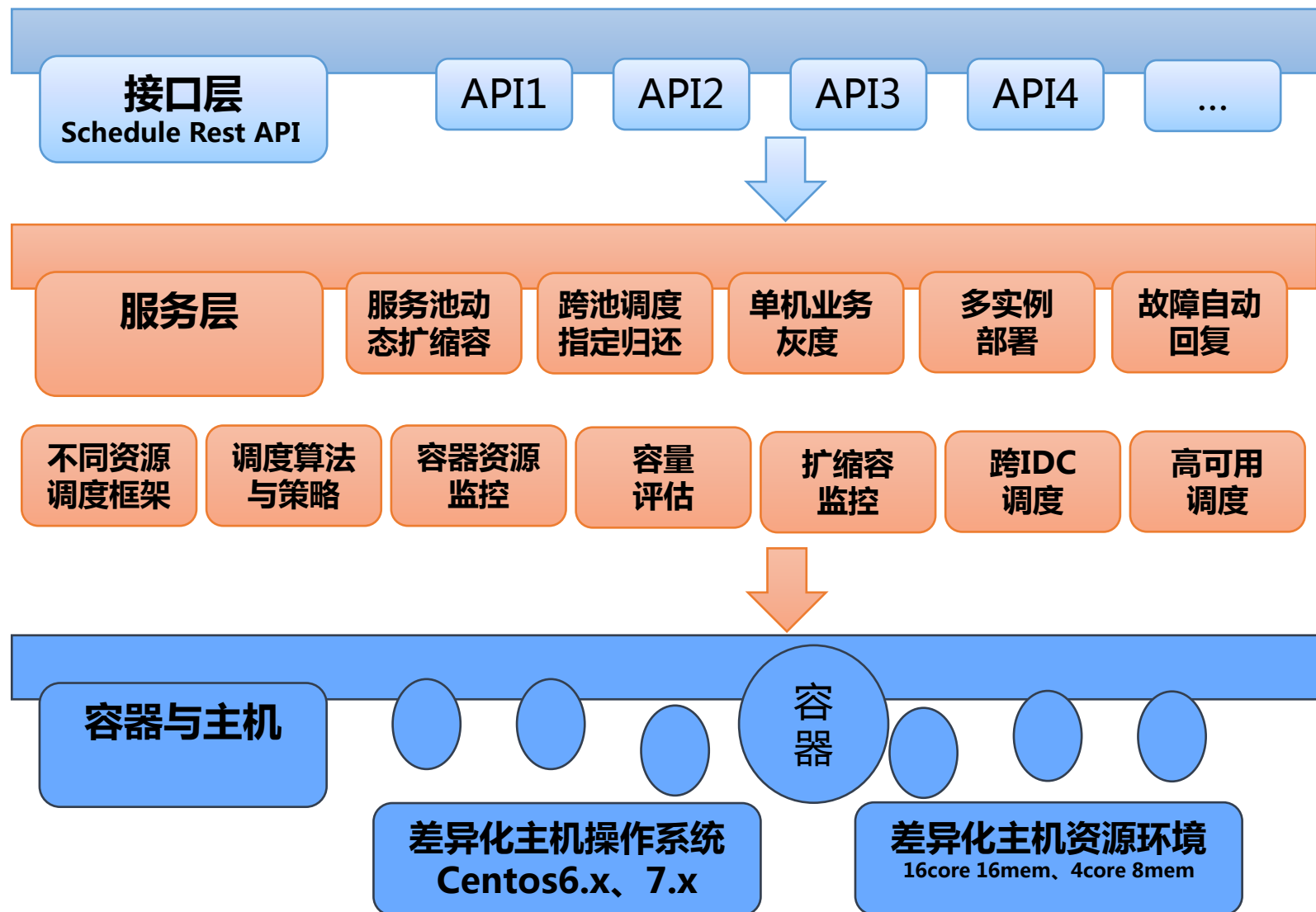
二、弹性调度

弹性调度 - 选型

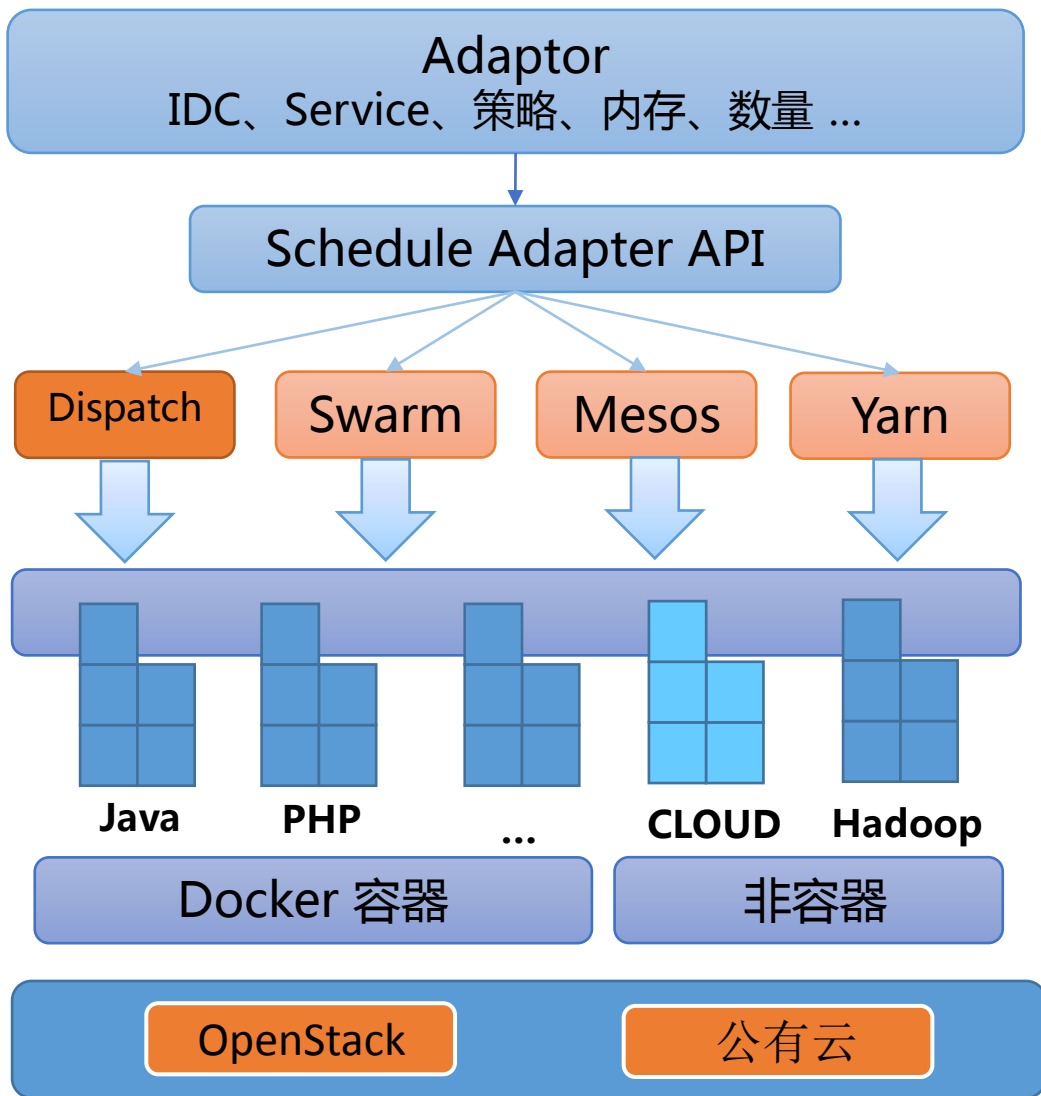
-	Swarm	Mesos	k8s
架构特点	针对Docker体系	借鉴Borg理念	源自Borg, 原生Docker
与Mesos结合	社区完善中	成熟	社区积极推荐
隔离机制	Docker	Mesos/Docker/其它	Docker
资源类型	内存 CPU 端口	内存 CPU 端口 Ulimit	内存 CPU 端口 Ulimit
调度非Docker	No	Yes	No
主机分组	Docker进程标签	Slave配置	Slave配置
打分策略	资源使用情况	资源使用情况	资源使用情况+应用节点均衡
端口编排	No	Yes	Yes
网络模式	Docker原生	支持自建	支持自建
主高可用	双主切换	zk	etcd
扩缩容	Roam二次开发	API修改实例数	API修改实例数
应用节点健康检测	No	Yes	Yes
应用节点自动故障转移	No	Yes	Yes
服务发现	Consul	zk	etcd
负载均衡	No	Haproxy	Kube-proxy
DNS	No	No	skydns
使用复杂度	低	中	中
集群规模	小	中	中, 在提升
生产使用	尚无大规模使用	业界大规模使用	业界大规模使用

弹性调度 - 选型

需求：
快速迭代
实现内网
计算资源
统一管理
调配，公有云上获得计算资源，快速自动化资源调度与应用部署



弹性调度系统



服务发现

调度框架

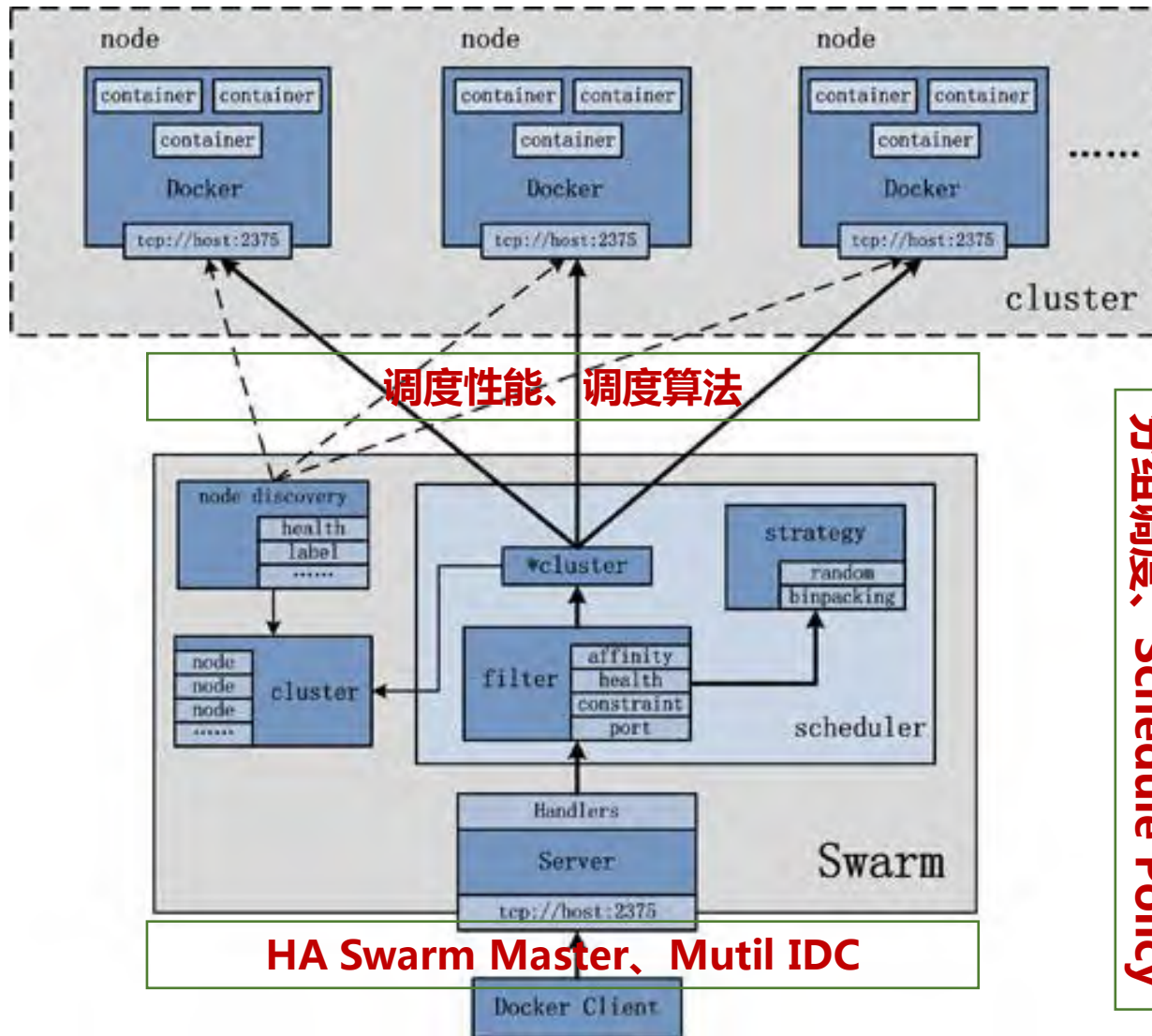
自定义调度

资源管理

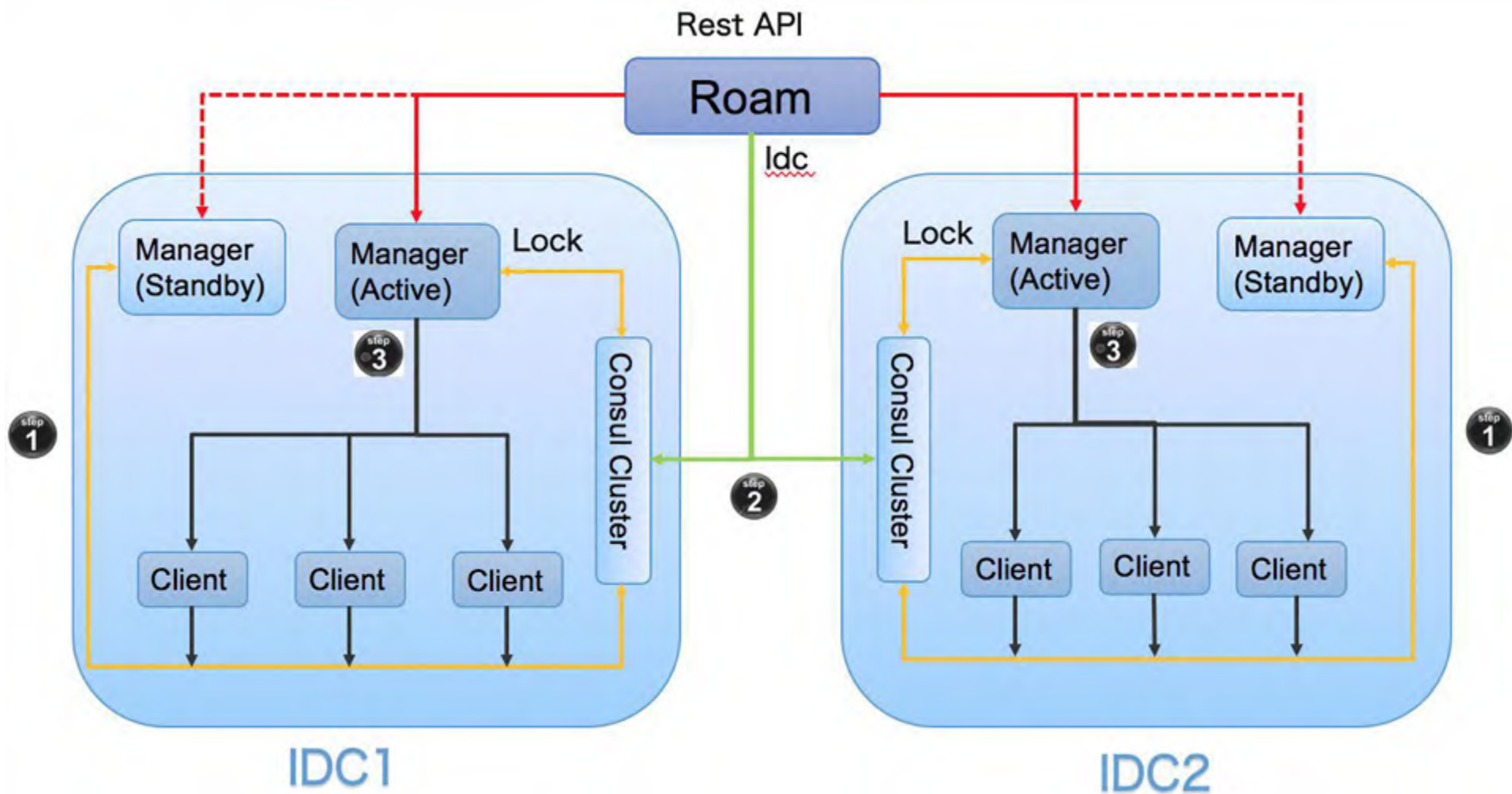
容量评估

监控报警

问题-容器调度-Swarm



问题 - 多IDC、高可用、可扩展



Swarm – 调度策略 – 不适用

- 调度=主机 or 容器过滤 + 策略选择
- 过滤器filter
 - Node Filters : health (会根据节点状态进行过滤, 会去除故障节点)、
constraint (约束过滤器、Label分组调度)
 - Container Configuration Filters : affinity (亲和性过滤器)、dependency (依赖过滤器)
、Port (会根据端口的使用情况过滤)
- 调度策略
 - 根据各个节点的可用的CPU, Mem及正在运行的容器的数量来计算应该运行容器的节点进行打分
，剔除掉资源不足的主机，然后策略选择：spread、binpack、random
 - Binpack : 在同等条件下，选择资源使用最多的节点
 - Spread : 在同等条件下，选择资源使用最少的节点
 - Random : 随机选择一个
- 调度颗粒度
 - Memory : docker run -m 1g ...
 - CPU : docker run -c 1 ...

Swarm-调度算法-打分机制

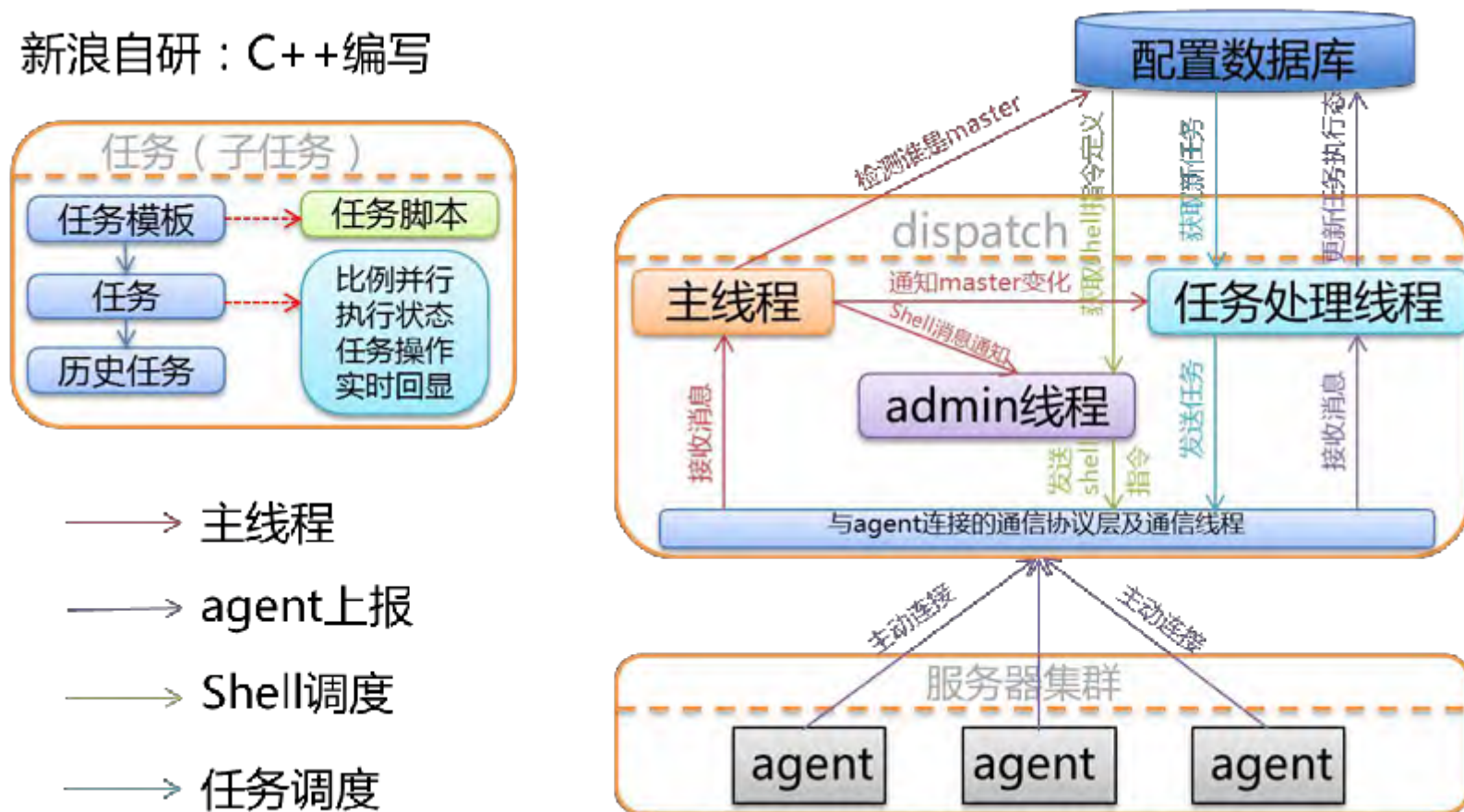
```
if config.CpuShares > 0 {  
    cpuScore = (node.UsedCpus + config.CpuShares) * 100 / nodeCpus  
    //cpuScore= ( 物理机已用CPU+本次需用CPU ) *100/物理机CPU  
}  
if config.Memory > 0 {  
    memoryScore = (node.UsedMemory + config.Memory) * 100 / nodeMemory  
    //memScore= ( 物理机已用内存+本次需用内存 ) *100/物理机内存  
}
```

```
if cpuScore <= 100 && memoryScore <= 100 {同时满足可用内存、CPU  
    weightedNodes = append(weightedNodes, &weightedNode{Node: node, Weight: cpuScore + memoryScore})  
}
```

资源只与容器Create时配置有关，与运行时实际使用资源情况无关。无论容器是否由Swarm创建，无论容器处在何种状态，只要配置了资源限额，调度时均会计算在内！

任务调度框架 - Dispatch

新浪自研：C++编写



DCP-弹性扩容第二步：服务扩容

- 扩容

- 输入：服务池名称、服务类型，容器类型、数量、镜像地址等
- 输出：扩容报告

- 记账中心

- 资源拥有者、使用方、型号、使用时间、日期、信用等

- 容器服务类型：按照dPxx(dockerPxx)命名

dP01
8核12G

dP02
12核12G

dP03
12核16G

dP04
16核16G

DCP-扩容操作任务化

任务详情

社区 文档 王关胜

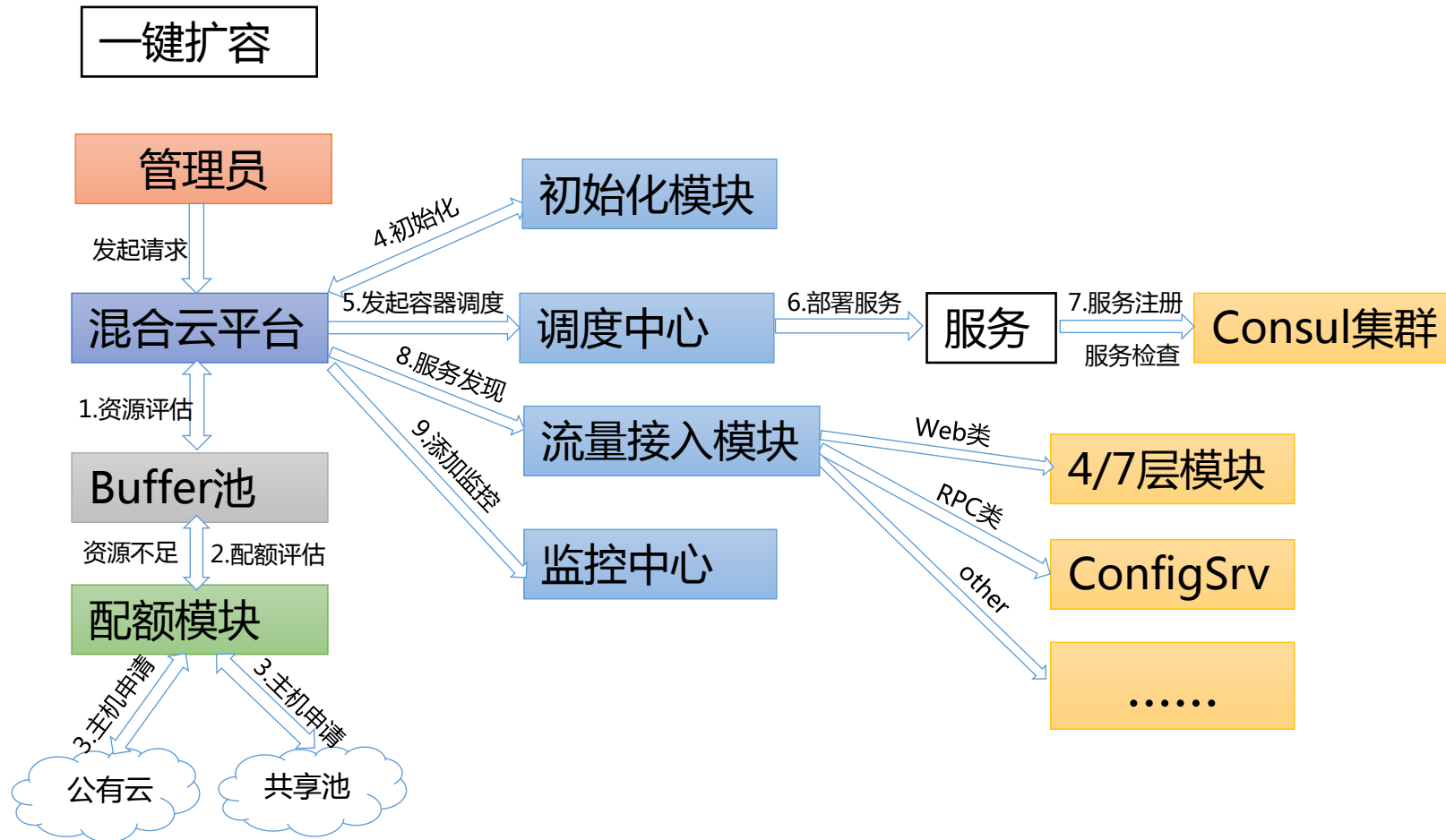
任务信息 任务参数和属性

任务名称	auto_deploy_openapi_StartDocker_20160207_201117		
版本号			
保留版本号			
备注			
任务类型	auto_StartDocker	应用名称	openapi_webv2_docker
应用路径	/data1/weibo	超时时间	700
是否启动	是	执行比例	100
是否自动执行	否	自动执行步长	2000

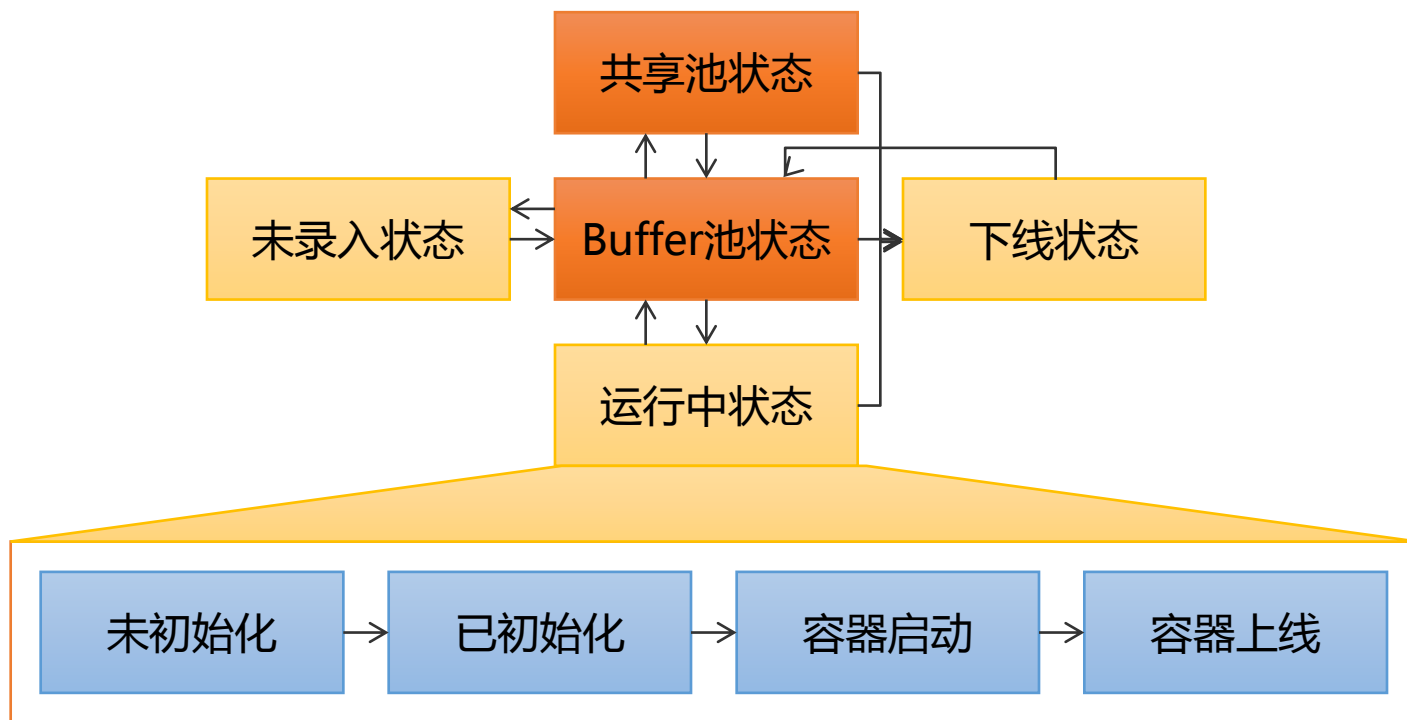
任务概览 概要统计

应用池子名称	总台数	未分发	成功	执行中	跳过	超时	失败	进度
openapi_webv2-aliyun_tc-core-inner-docker	44	0	33	0	11	0	0	100%

DCP-弹性扩容：流程



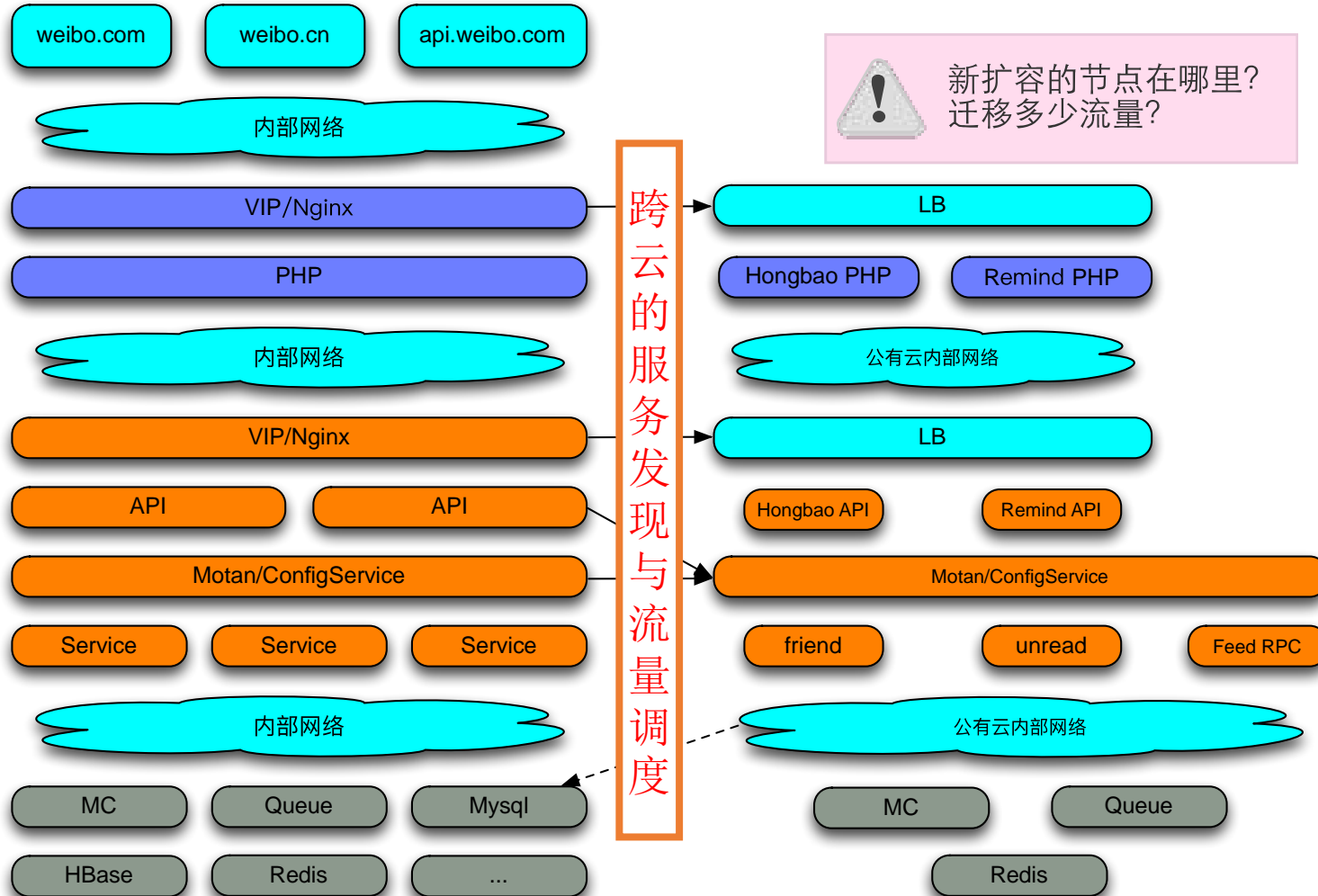
主机设备生命周期



正常主机状态变迁

关键点：服务发现

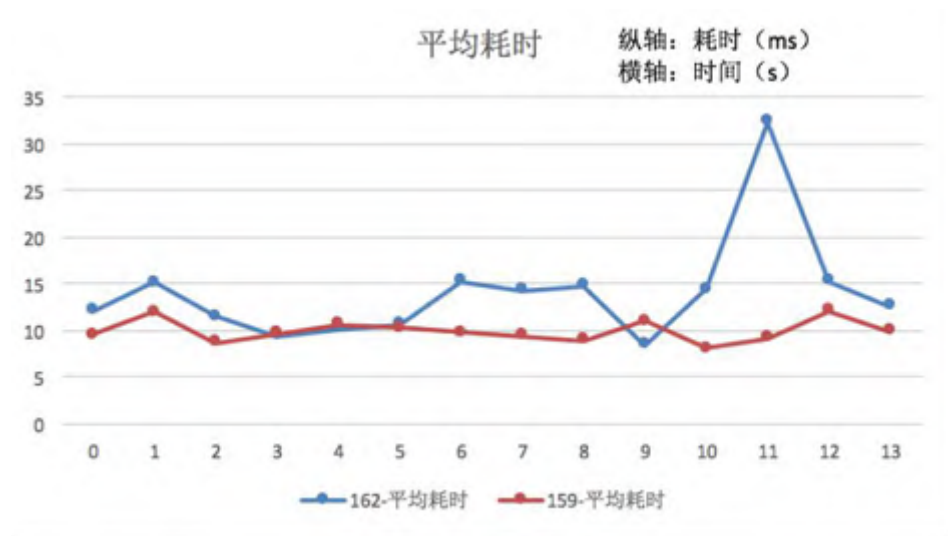
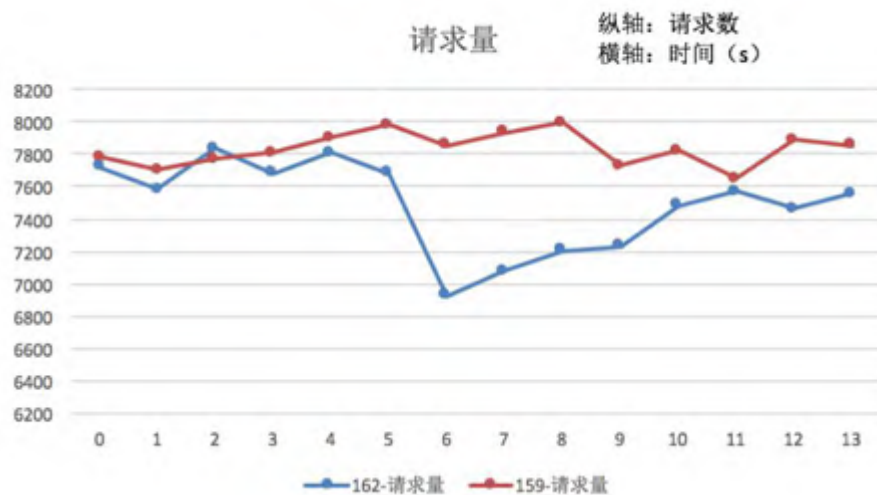
流量要快速、安全的切到弹性节点



服务发现-业界常用方案

问题：Reload损耗

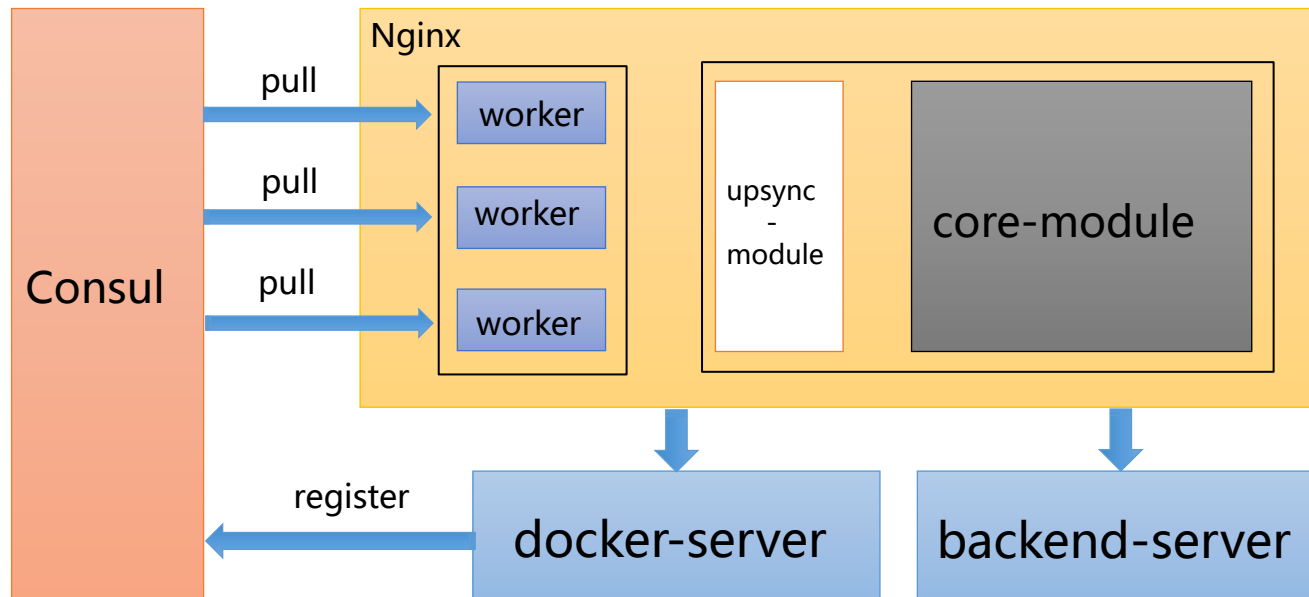
- 开源解决方案大多利用Nginx的Reload机制
- 性能损耗情况:
 - ◆ 请求量：普通reload会导致吞吐量下降10%
 - ◆ 平均耗时：差异不大



服务发现：微博方案

微博方案 - nginx-upsync-module

- Nginx Plus的开源版
- 支持基于Consul自动服务发现
- 开源：<https://github.com/weibocom/nginx-upsync-module>



Ngix Upsync-自适应后端处理能力

➤ 弹性节点的处理能力不对等

➤ server 10.xx.xx.xx:xxxx max_fails=0 fail_timeout=30s

weight=20; #同样的权重导致单点性能恶化

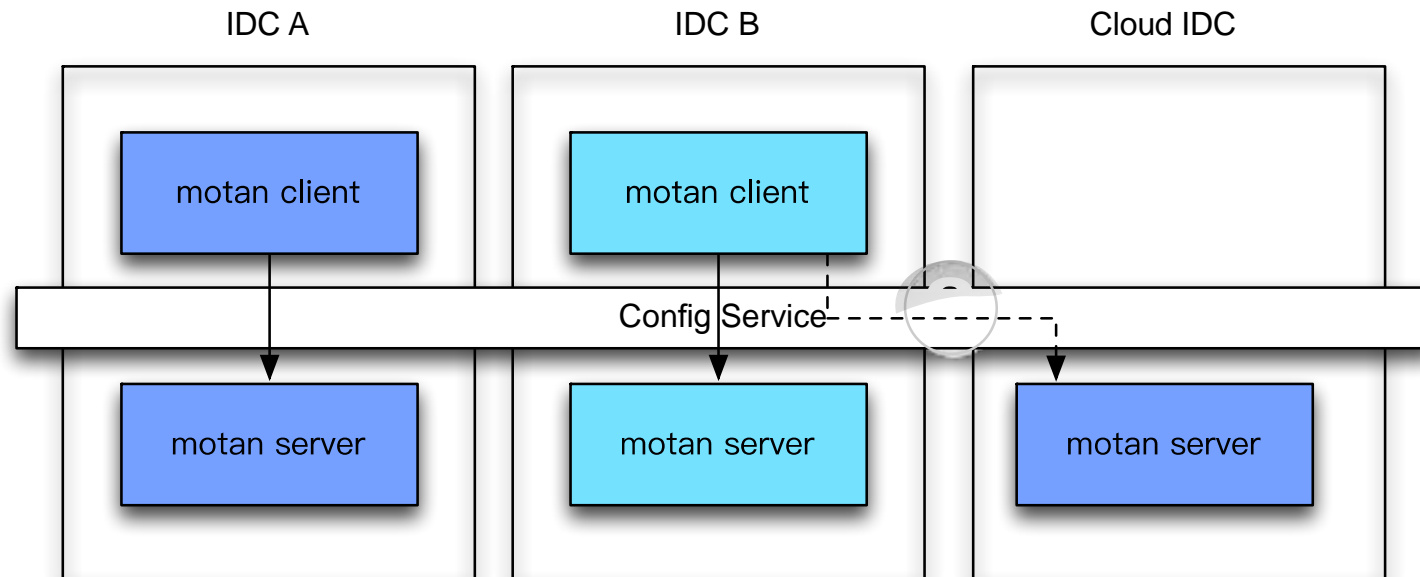
➤ 节点注册计算能力

➤ 所有节点默认权重是20；

➤ 公有云有20%性能损耗，权重=16；

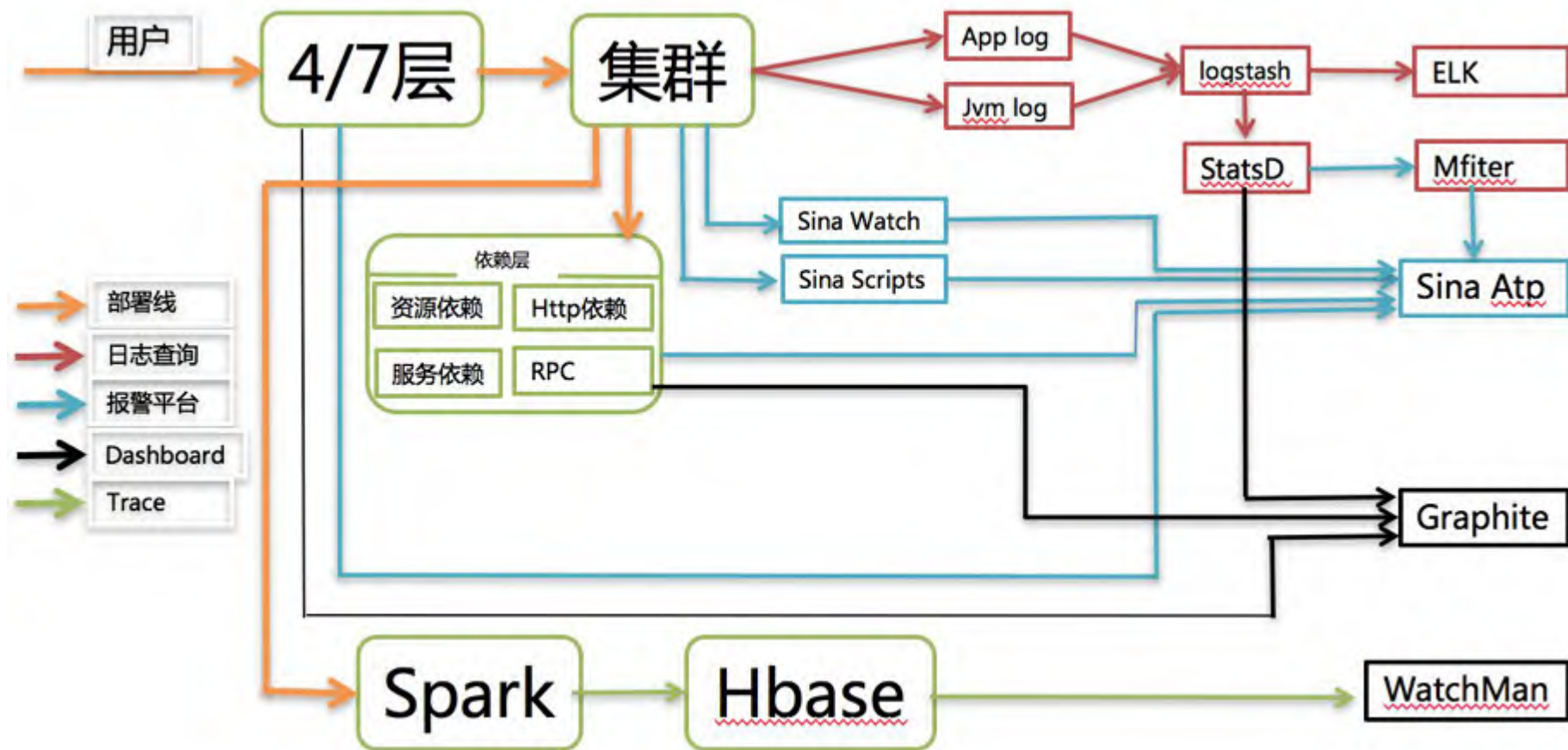
RPC-Motan RPC服务发现

- 开源：微服务RPC框架Motan
 - <https://github.com/weibocom/motan>
- Motan流量路由模型优先匹配单IDC
 - IDC间切换一般仅在故障处理时采用
- 新版Motan已支持按流量权重配置定向路由

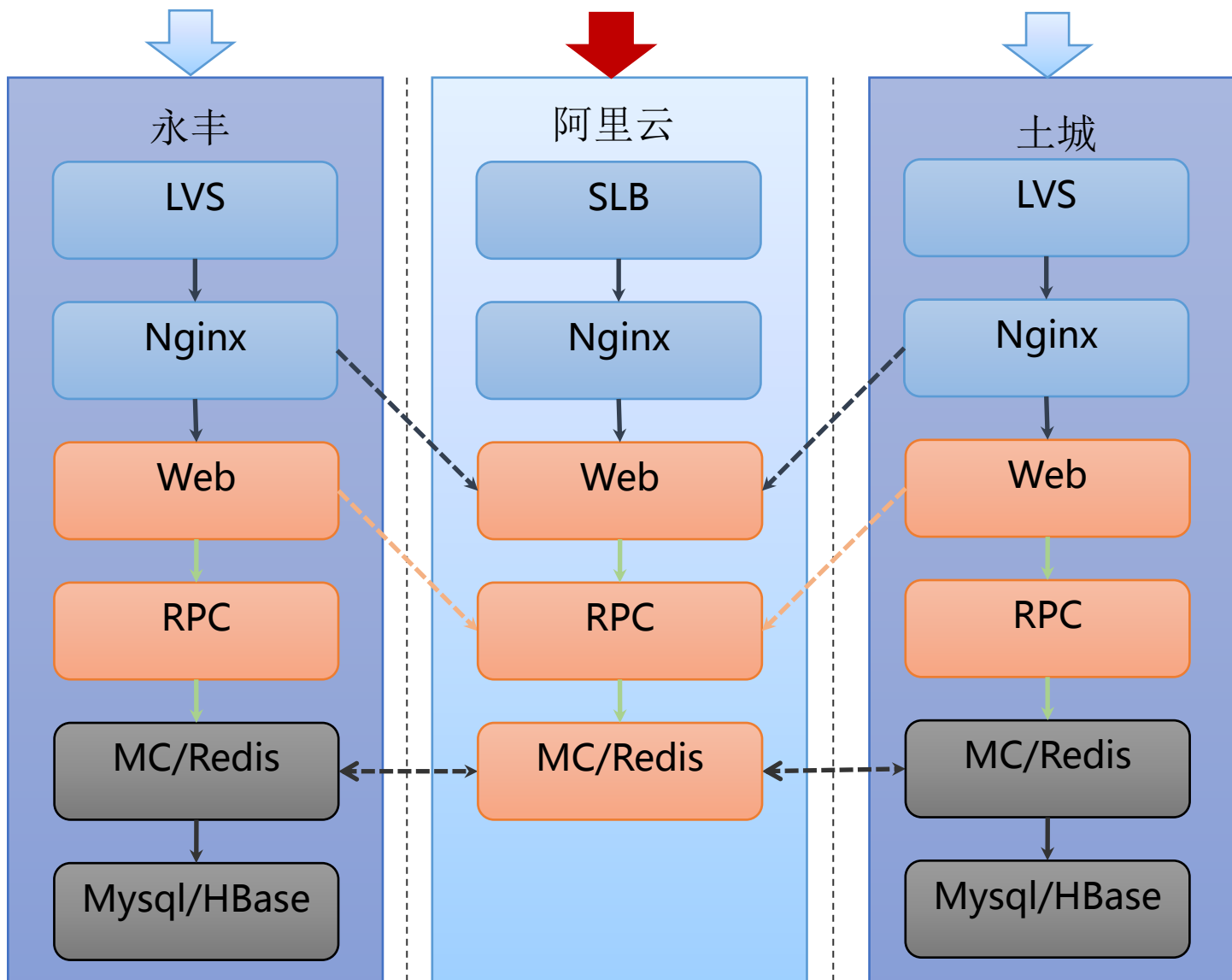


五、春晚实战与总结

DCP-三节保障：监控体系



三节保障与阿里云部署



微博混合云DCP成果

● 混合云进展：

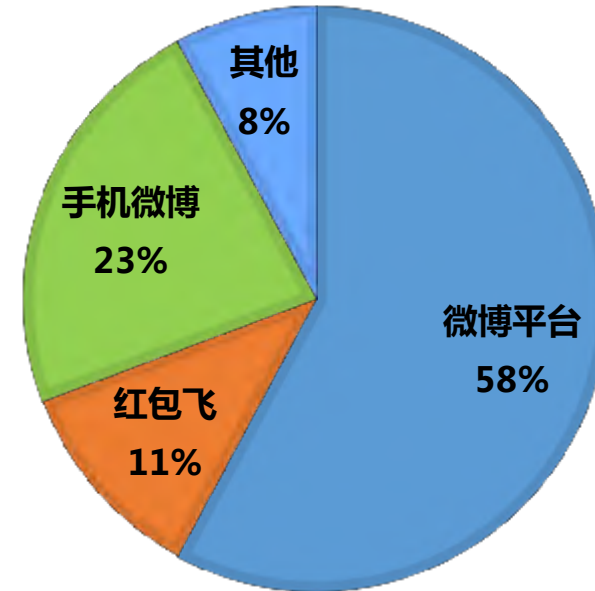
- 容器数：5000+
- 晚高峰自动扩容500+

● 春晚备战：

- **10分钟混合云扩容1000节点技术能力**
- 2017春晚峰值历史新高，完成**4700台**阿里云ECS扩容，实现无降级平滑过渡，公有云高峰支持微博50%主体流量。完成eod流、红包飞、手机微博公司各主要业务线均完成上云支持

主要业务方

■ 微博平台 ■ 红包飞 ■ 手机微博 ■ 其他



总结 – 春晚问题与总结

问题	解决
阿里云部署缓存ECS PPS打满	扩容缓存MC L1一倍
Feed性能恶化，部分ECS性能差	单机监控发现性能恶化ECS，执行503
资源DNS解析失败	不使用SLB负载均衡，直接配置
环网带宽优化	网络链路优化 业务架构优化

OpenDCP : 基于Docker技术的混合云管理平台

OpenDCP : <https://github.com/weibocom/opendcp>

- 综合性的运维管理平台。涵盖运维配置、发布、上线变更等运维管理主要功能，而不局限于容器集群管理，可适配Kubernetes、Mesos、Swarm等
- 功能覆盖镜像市场、多云对接、服务编排、服务发现等云资源管理主要环节
- 支持阿里云、AWS、私有云等主流云厂商
- 支持Nginx、SLB等服务发现方式
- 支持Java、PHP、C/C++、Go等主流语言

欢迎大家到OpenDCP开源社区沟通交流：

itfuwen@163.com @it_fuwen



Thank you !