



全球运维大会

2016

DevOps 2.0: 重塑运维价值

北京站

会议时间：12月16日 - 12月17日

会议地点：北京国际会议中心

主办单位：



乐视高可用电商云平台架构与实现

邓良驹 研发工程师



目录



1

电商系统发展与问题

2

微服务化与服务治理

3

乐视高可用电商云架构解析

4

抢购案例介绍

5

高可用RDS介绍

6

采坑经验



电商系统发展过程

• 初创期

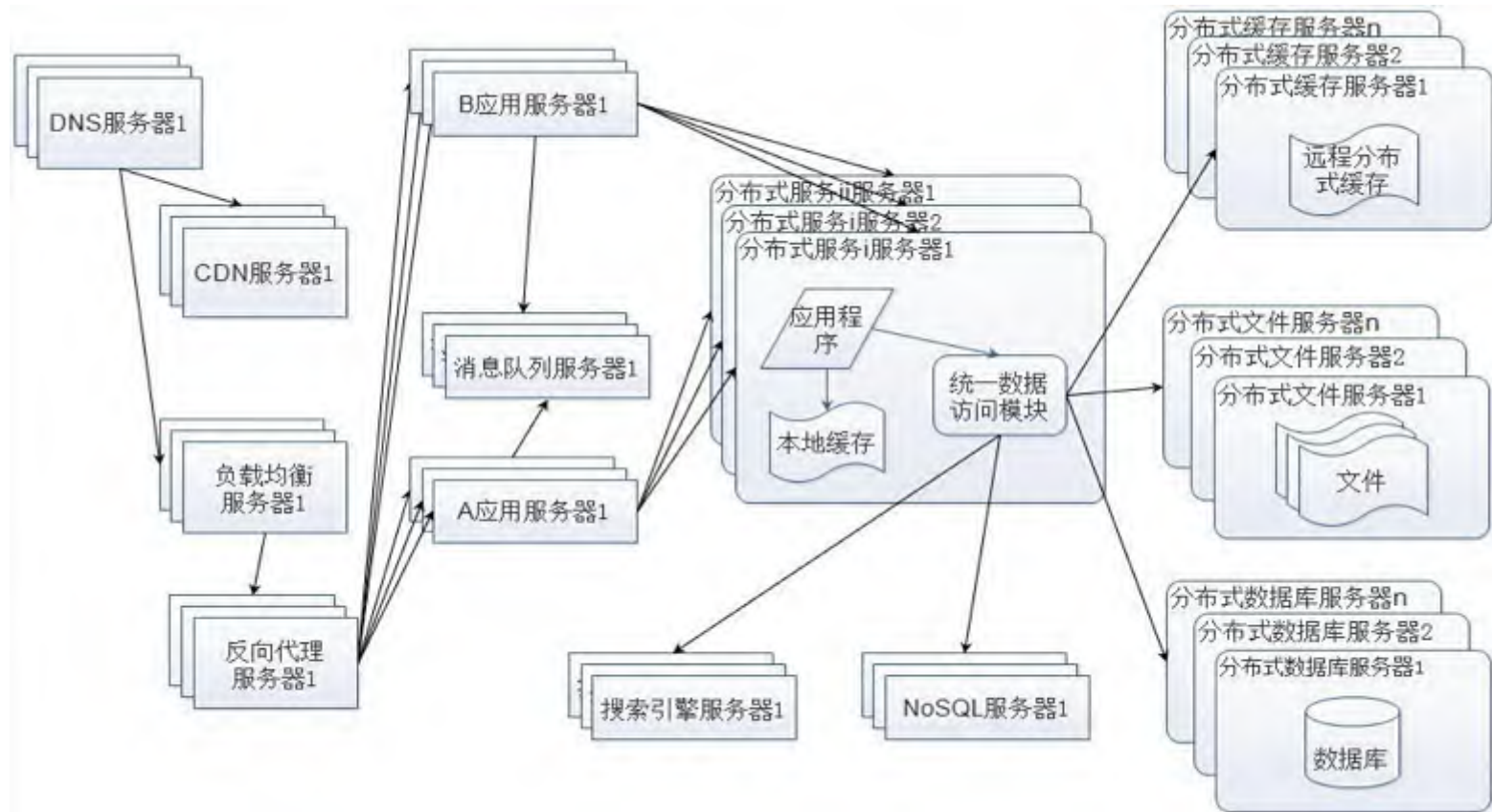
商品类型少，业务复杂度低，系统架构简单。采用高可用数据库、分布式缓存、文件存储等基本组件就可满足需求。

• 发展期

数据量、业务复杂度、系统复杂度、计算资源需求都剧增。则需要业务拆分并独立部署，采用CDN、高可用数据库、分布式缓存、分布式消息队列、分布式文件存储等。



网站基本架构



高速发展期电商系统面临的问题

- **随着业务的快速扩张、计算资源需求也扩张，但利用率低下**

另一面是企业想降低运维成本

更多地资源意味着要投入更多的人力

应付促销等活动而冗余的资源大部分时间都闲置



高速发展期电商系统面临的问题

- **系统复杂度的直线上升，耦合严重，牵一发动全身**

一直被业务牵着走，少有时间对系统进行及时梳理和重构

甚至有相似度极高的代码存在多份拷贝

也许还一时招不到人来搞定能应付业务规模的系统，例如应对抢购等高并发场景



高速发展期电商系统面临的问题

• 安全性问题愈加严重

业务做大后容易招致DDoS、恶意刷单、恶意渗透、用户隐私泄露等安全风险

大多数与用户隐私、交易相关的平台都被“脱裤”过，严峻形势几乎可以用“无一幸免”来形容

《网络安全法(草案)》让企业不能再以“致歉”了事

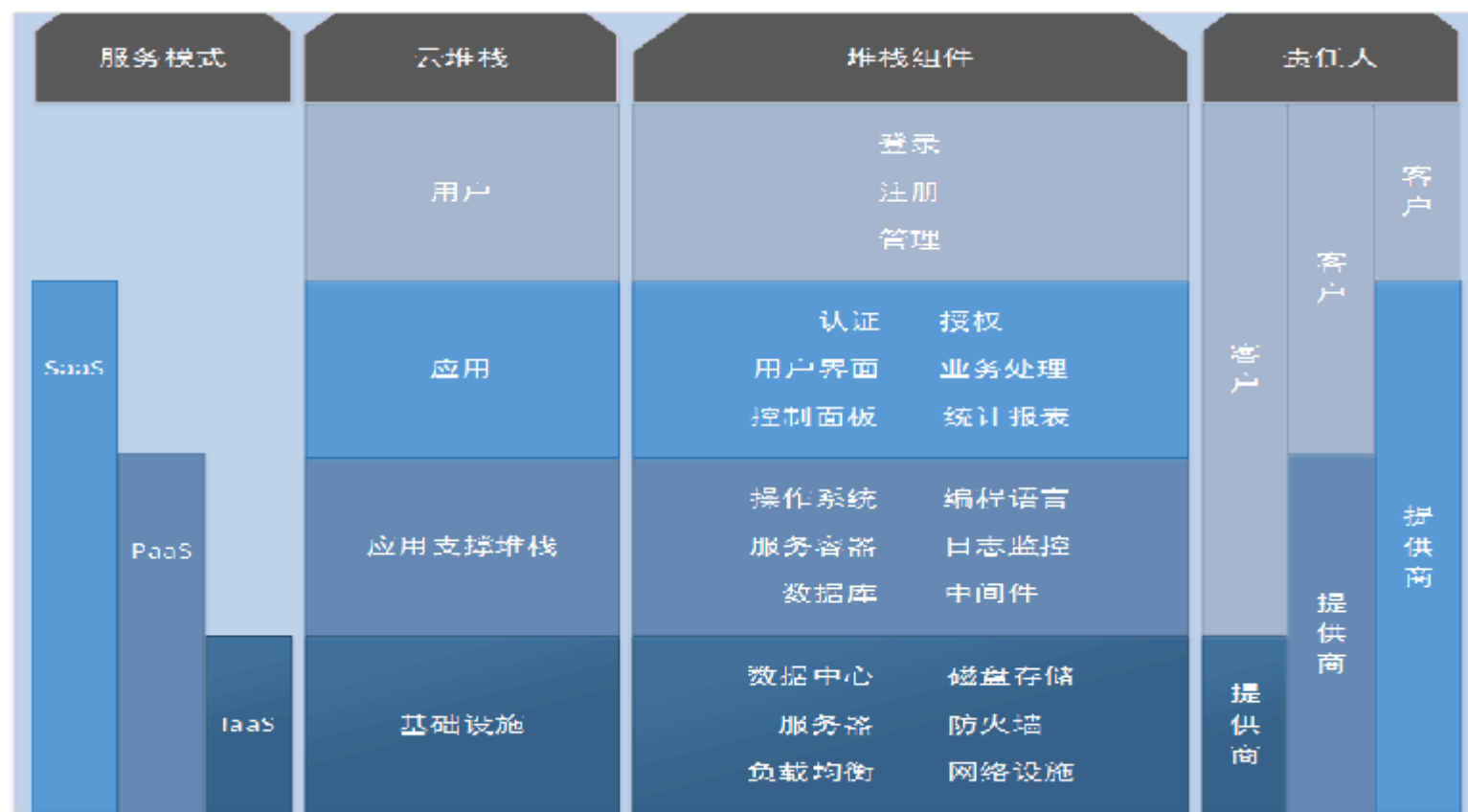


如何走出困境

- 一般企业的核心竞争力在商业市场上，而不是在技术保障上。尤其是小微电商或传统电商都难有相应的技术实力。
- 云计算方兴未艾，**按需自服务、宽带接入、资源池化、快速伸缩、服务可计量**等五大特点都让上述各种问题可以良好解决
- 几乎各大云计算平台都针对电商行业推出了相应的云计算产品
- 微服务架构可以降低单个系统复杂度，借助云计算平台可以良好地实施



云计算服务简介



向云迁移

- 为了更好地发挥云计算平台优势，业务系统需要适时重构

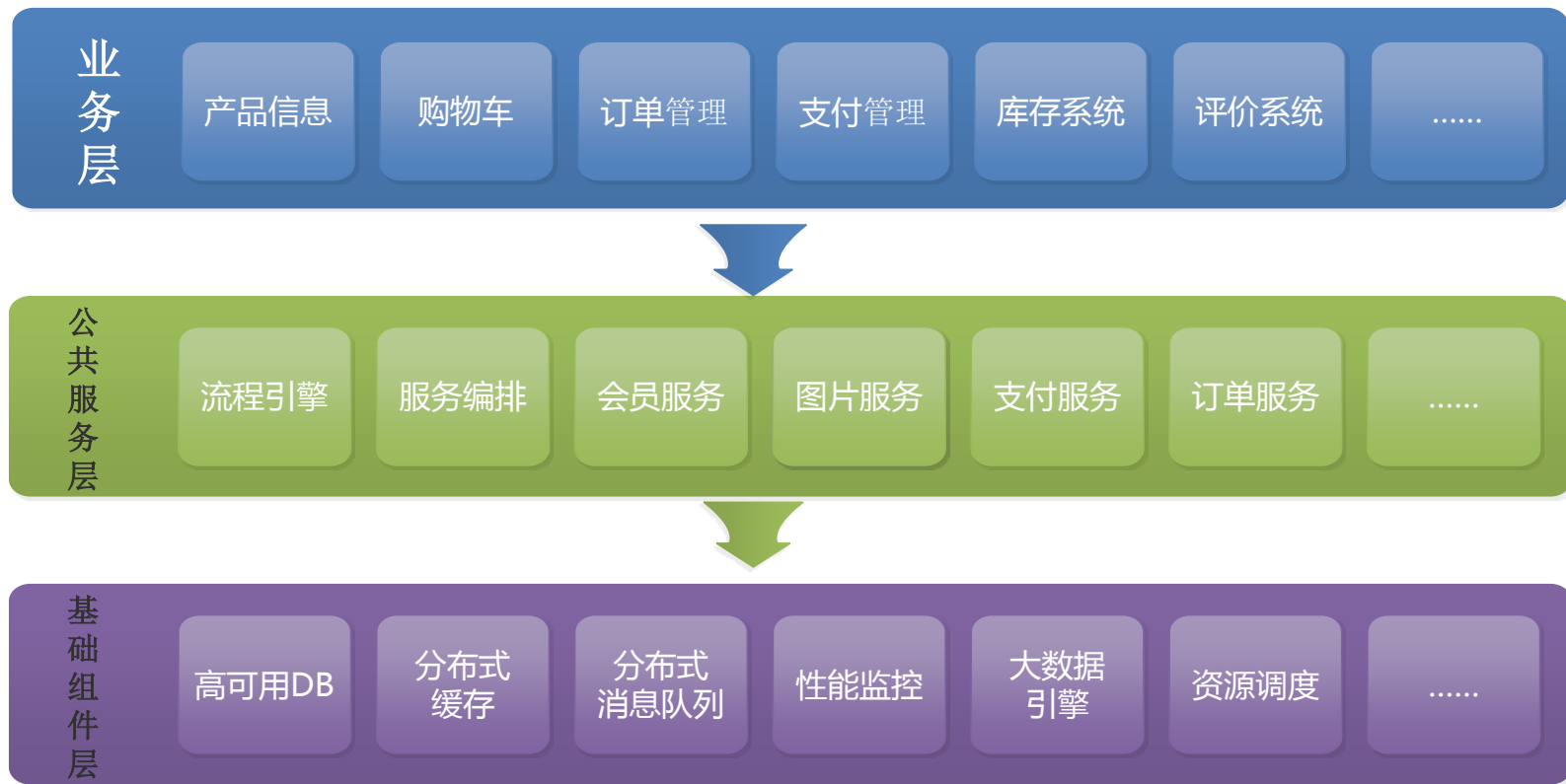
微服务化

无状态

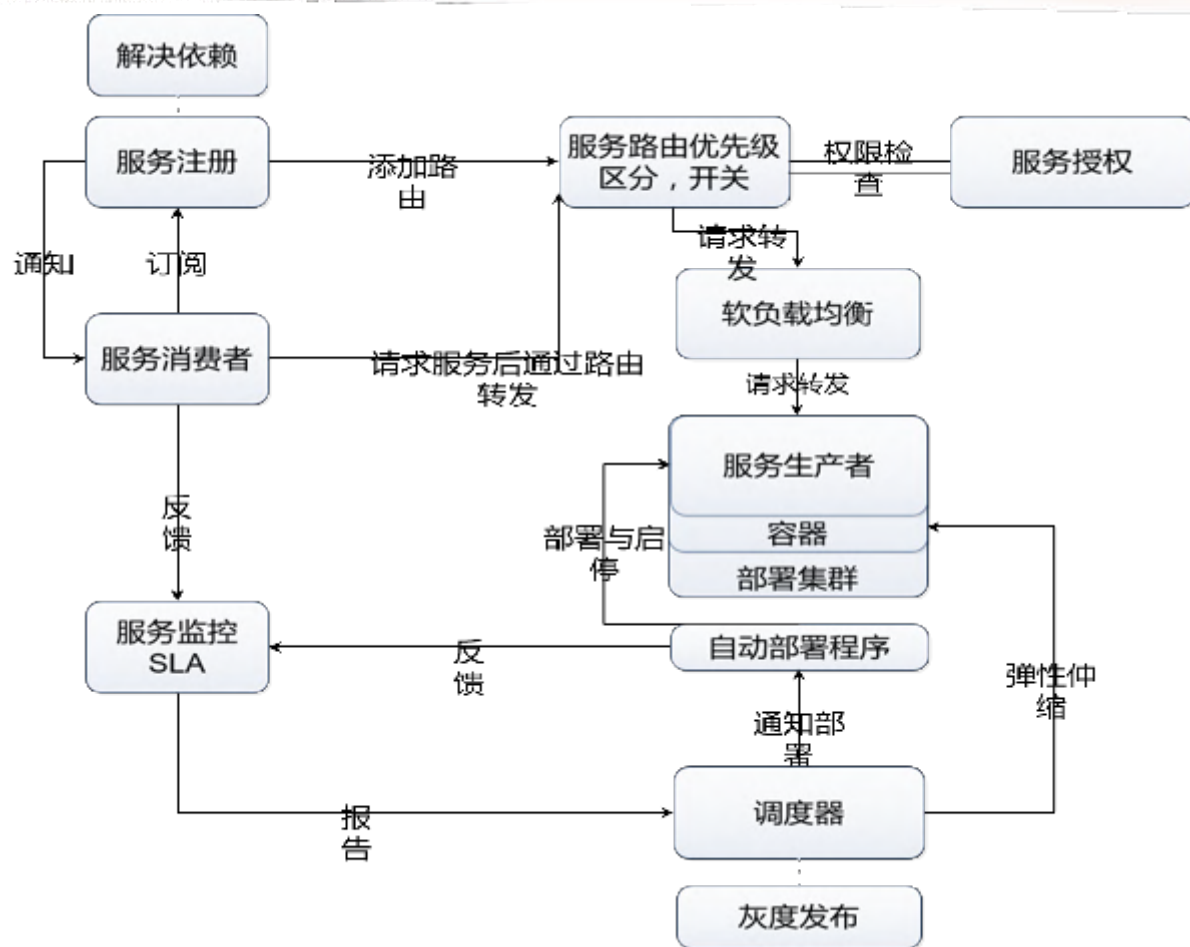
状态分离



电商微服务体系结构



服务治理



对电商云的要求

- 高性能
- 高可用
- 良好伸缩性
- 方便地拓展性
- 安全保证

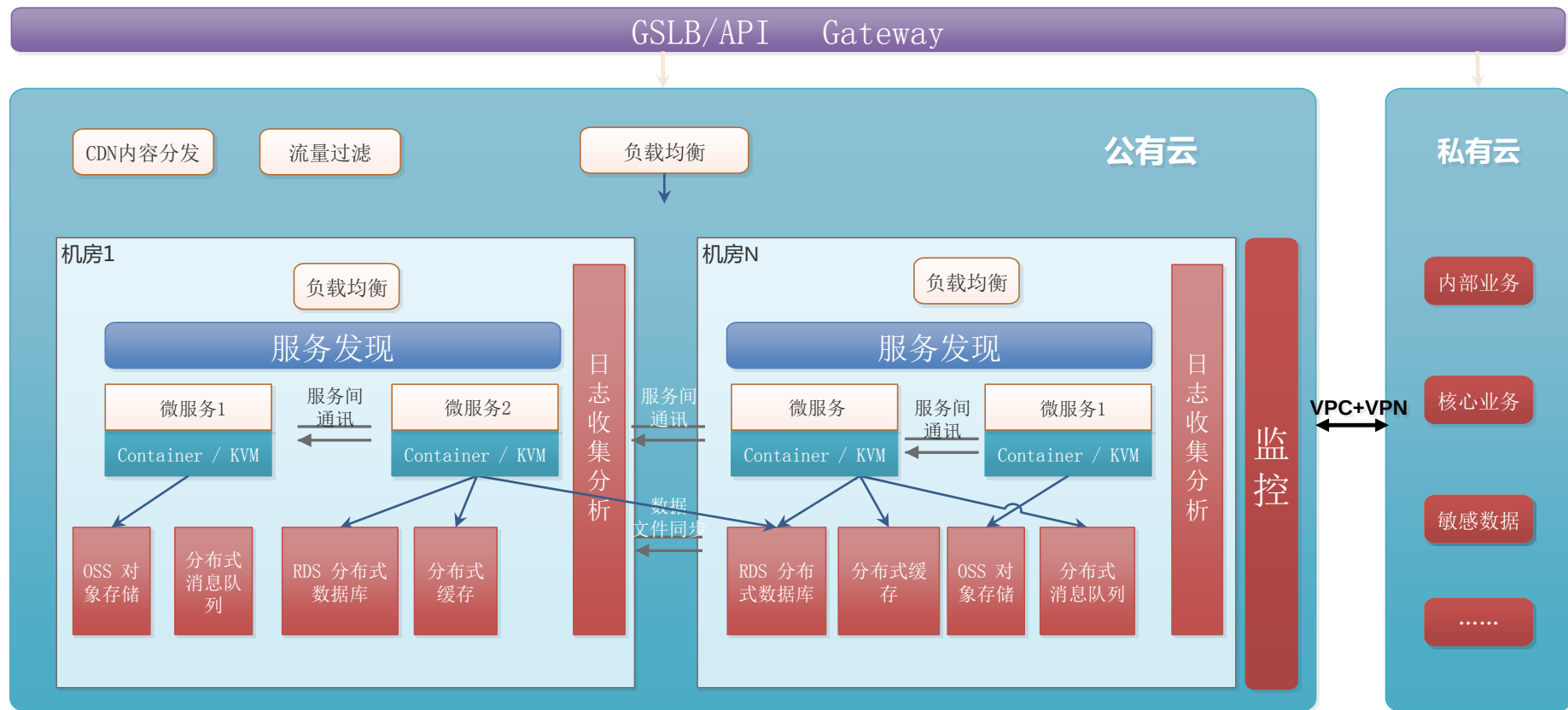


乐视电商云支持的部署模式

- 公有云
- 私有云
- 混合云



混合云结构示意图



平台架构



为何要基于容器技术



应用开发人员

- 版本质量更高
- 应用可扩展性更强
- 应用隔离效果更好



IT 架构师

- 横向扩展速度更快
- 测试周期更短
- 部署错误更少



IT 运营人员

- 版本质量更高
- 更高效地替换生产环境中的全部虚拟机
- 应用管理更便捷



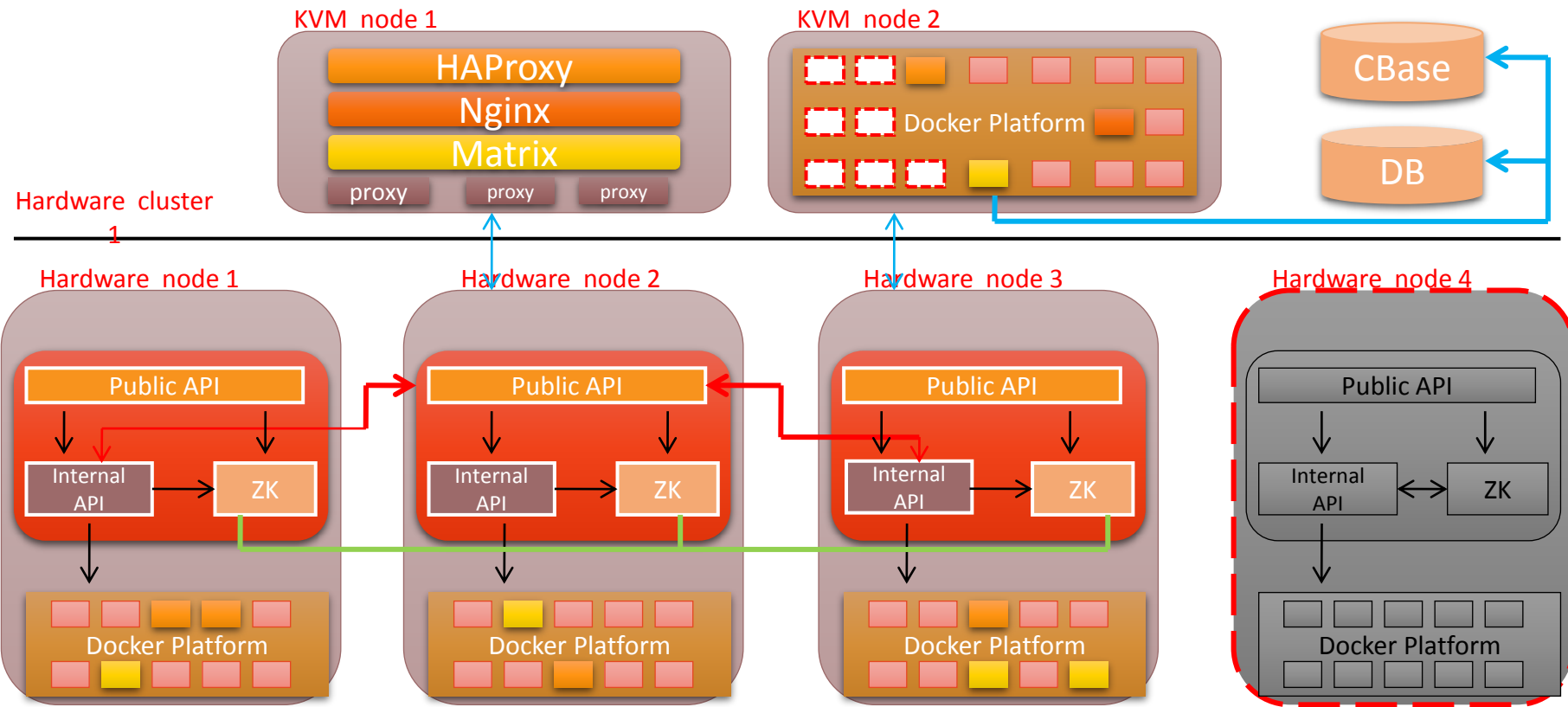
BeeHive & Matrix

- Matrix系统，负责用户体系与资源及组件关系维护、跨域资源的全球调度。支持自助化组件申请，服务监控报警，计费，服务编排等服务。
- Beehive 系统，负责在IaaS层之上进行组件服务的调度管理。支持容器资源的构建、集群管理、容器管理等服务。

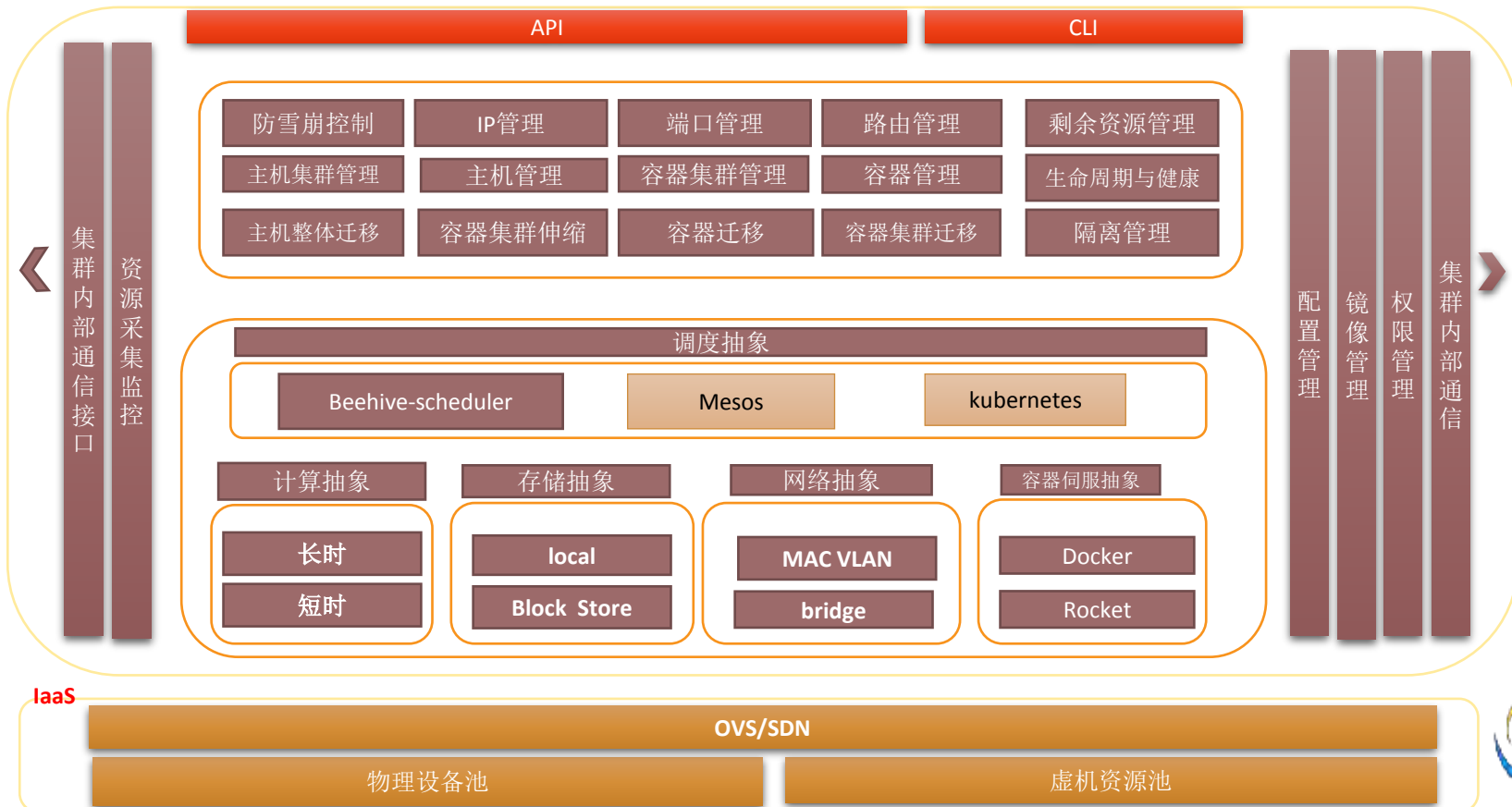


Matrix & Beehive协同部署示意图

KVM cluster 1



Beehive内部结构



抢购案例介绍

独立
部署

秒杀业务独立部署，与原业务隔离

抗压
部署

秒杀按钮相关业务链抗压部署

弹性
伸缩

应用按需取用资源部署节点数、弹性伸缩

流量
过滤

异常流量过滤，频次限制

页面
静态化

秒杀页面尽量 使用静态页面



场景举例及应对方法

- **场景一：** 用户不断刷新页面获取商品信息造成大量访问请求，会占用大量的带宽资源。

解决办法： 页面尽量静态化，压缩页面内容，优化HTTP头，可以使用乐视CDN做静态加速。



场景举例及应对方法

- **场景二：** CDN解决了静态资源的问题，但仍有大量有效或恶意的请求被转发至服务端，服务器压力很大。

解决办法： 乐视云防护可进行流量过滤，把非正常流量、无效流量在到达服务端之前清洗掉，可以屏蔽绝大部分DDoS，并接入WAF程序对HTTP请求进行分析，可以有效拦截SQL注入、XSS跨站、获取敏感信息、漏洞利用等常见攻击行为。

请求预处理，通过业务逻辑限制尽早拦截无需继续处理的请求；透传的流量将作用在提前扩容的资源上。



场景举例及应对方法

- **场景三**：大量要处理的请求导致数据库压力巨大，频繁的操作导致数据库性能低下甚至宕机。

解决办法：针对业务设计高效的缓存策略，热数据利用OCS进行缓存，需全局一致性的数据存储在高可用RDS集群，集群中一半以下的节点宕机都无需担心。



场景举例及应对方法

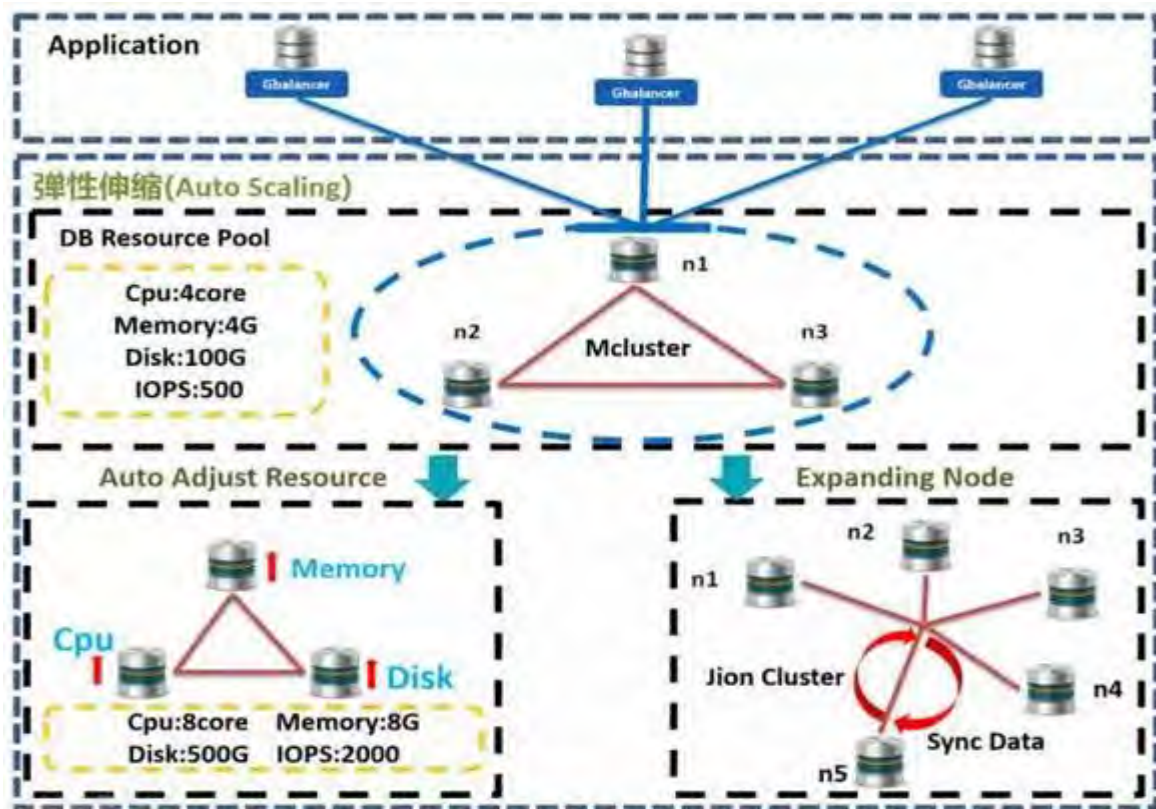
- **场景四：** 高并发下数据不一致性会导致超卖等情况。

解决办法： 通过上面请求预处理后已经大大减少了订单的创建和超卖的可能，但超卖还是可能存在的。首先，利用消息队列来缓冲请求，超过设定阈值的请求不处理，直接返回抢购结束。

其次，利用分布式消息队列来同步商品库存量，操作库存时临时加锁，将库存减扣和订单更新等必要操作封装成事务。



高可用RDS介绍



采坑经验

- 容器CPU资源的隔离性
- 减少分布式锁的应用
- 容器并发创建效率
- 大规模容器节点的管理salt的不稳定性
- 不同类型应用（CPU/IO）即时部署时的环境选择





Thanks

高效运维社区
开放运维联盟

荣誉出品



GOPS2016
Beijing

DevOpsDays 即将首次登陆中国



DevOps 之父 Patrick Debois 与您相约

DevOpsDays 北京站 2017年3月18日



门票早鸟价仅限前100名，请从速哟

<http://2017-beijing.devopsdayschina.org/>



GOPS2016
Beijing



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好





Thanks

高效运维社区
开放运维联盟

荣誉出品

