GIAC

主办方：msup | 架构 ARCHNOTES

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

微博Kubernetes实践经验分享

彭涛　　新浪微博 架构师

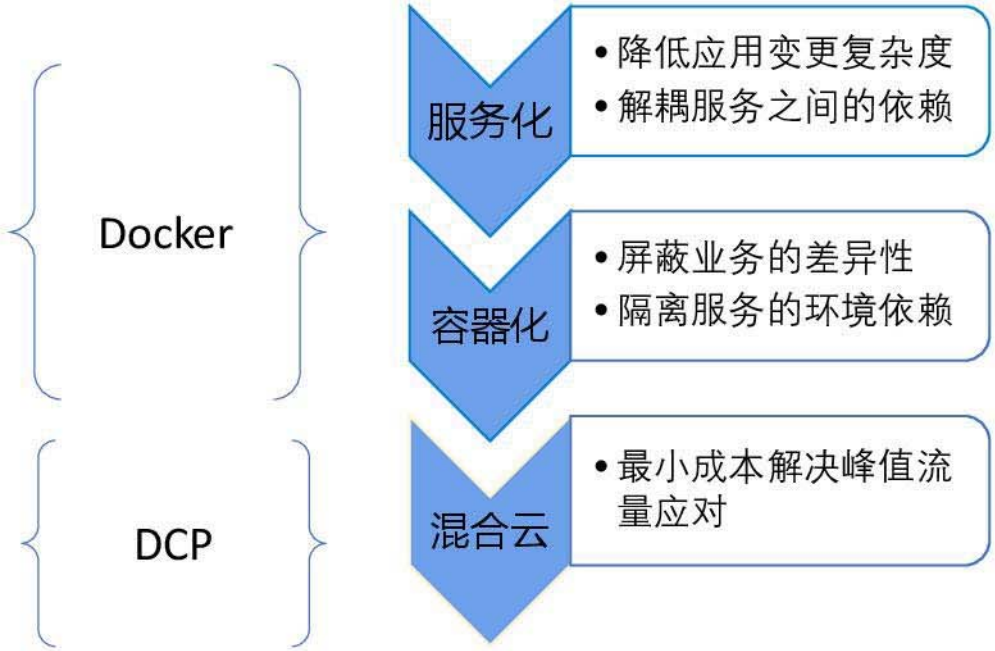
一、大纲

1. 微博做Kubernetes的整体背景
2. 在推进Kubernetes落地遇到的坑
3. 展望和总结

二、背景

- 微博是如何应对挑战的-架构的演进
- 微博当前架构下遇到的问题
- 整体的解决思路

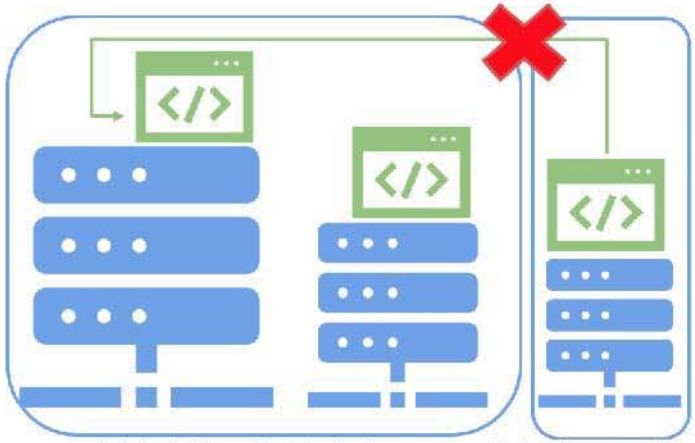
二、微博基础架构的演进


Docker
DCP
服务化
• 降低应用变更复杂度
• 解耦服务之间的依赖
容器化
• 屏蔽业务的差异性
• 隔离服务的环境依赖
混合云
• 最小成本解决峰值流量应对

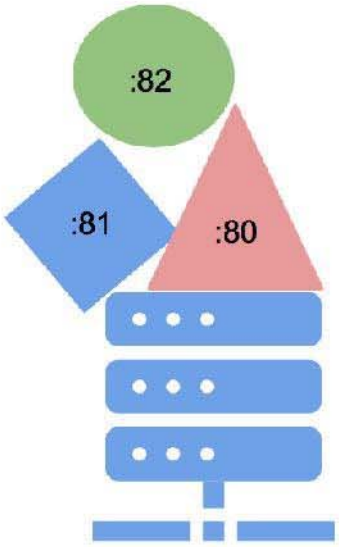
二、微博当前架构下遇到的问题（某女星结婚的流量视图）

- IAAS层的扩容计划是**最少6分钟以上**（三分钟系统自检+三分钟业务容器启动+创建机器排队+获取IP后变更负载均衡…）
- 流量涨幅约50%，而且**在4分钟内**达到顶峰
- IAAS层扩容不能解决该问题，只能降级+扩容并行完成……

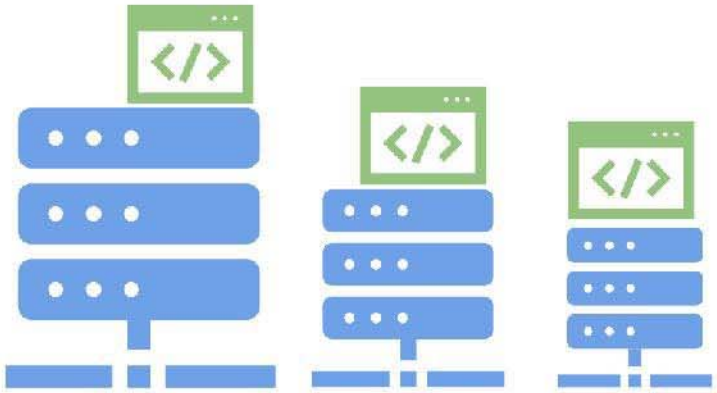
二、微博当前架构下遇到的问题



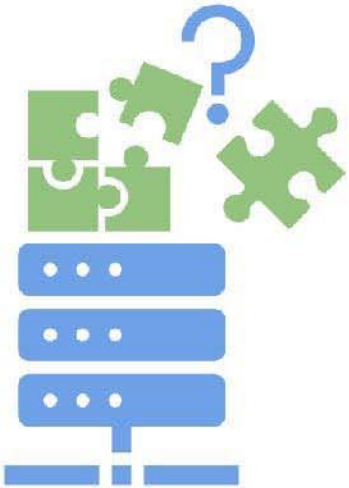
调度策略为服务池独占，不同业务类型无法共享，资源紧张



容器直接使用物理网络，混合部署管理成本高

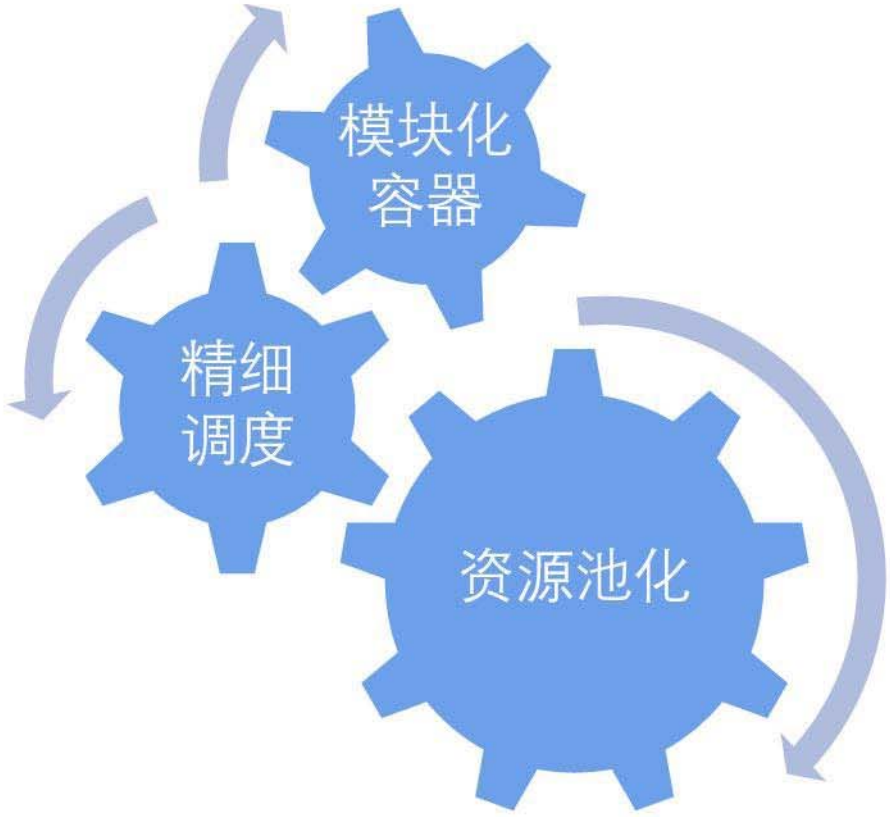


业务容器兼容设备，机器资源无法充分利用

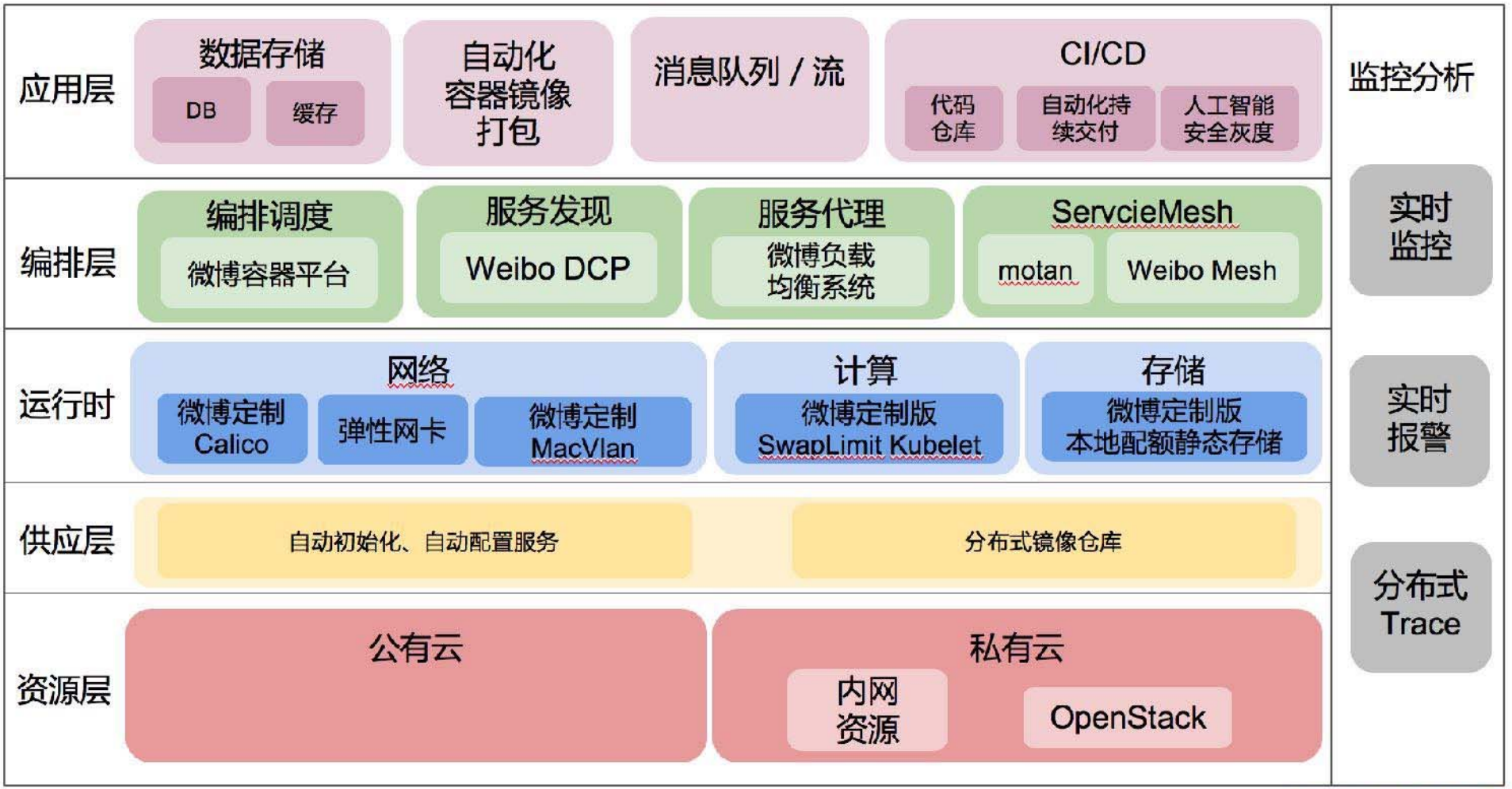


新的基础组件整合难度大

二、整体的解决思路（利用闲散资源完成扩容）



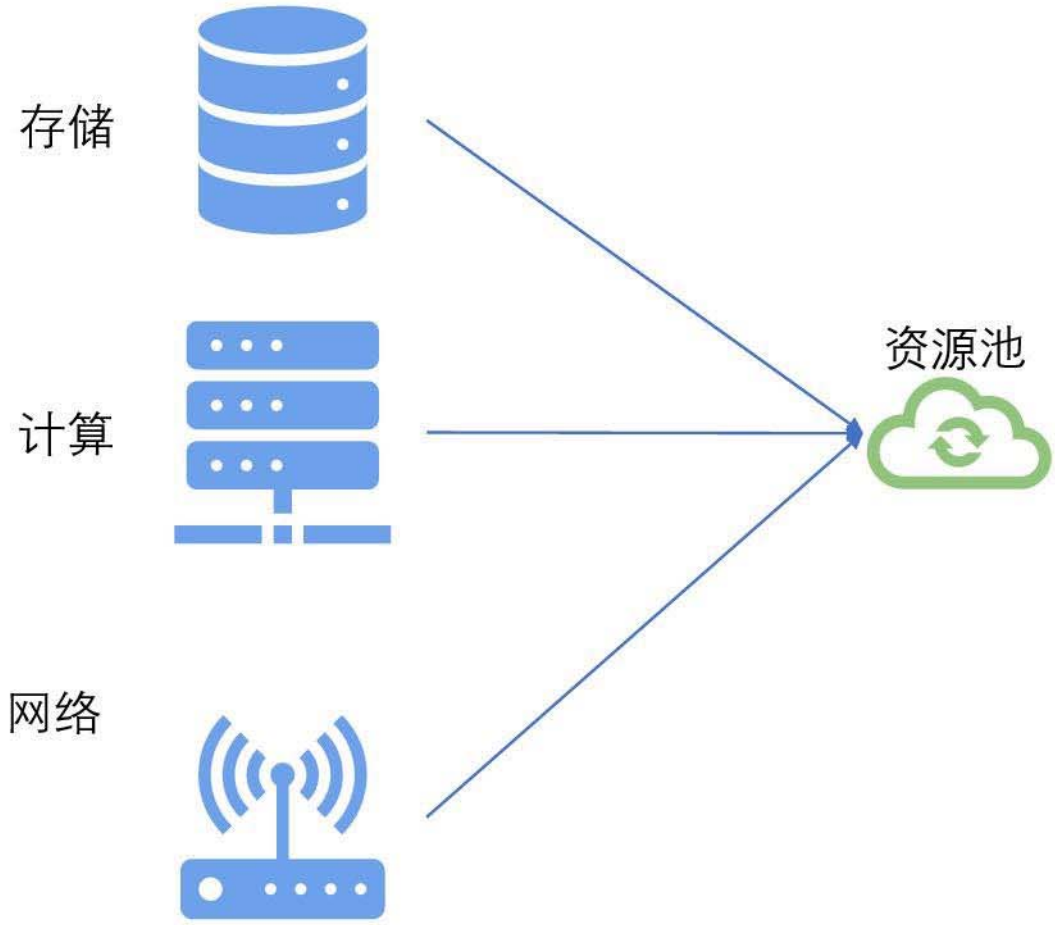
三、微博容器管理平台整体架构



四 、如何基于Kubernetes来解决问题

- 基础建设之网络、计算、存储资源池化
- 精细化调度和IP预分配
- In-place rolling update的滚动发布
- 模块化容器的思维

四、资源池化

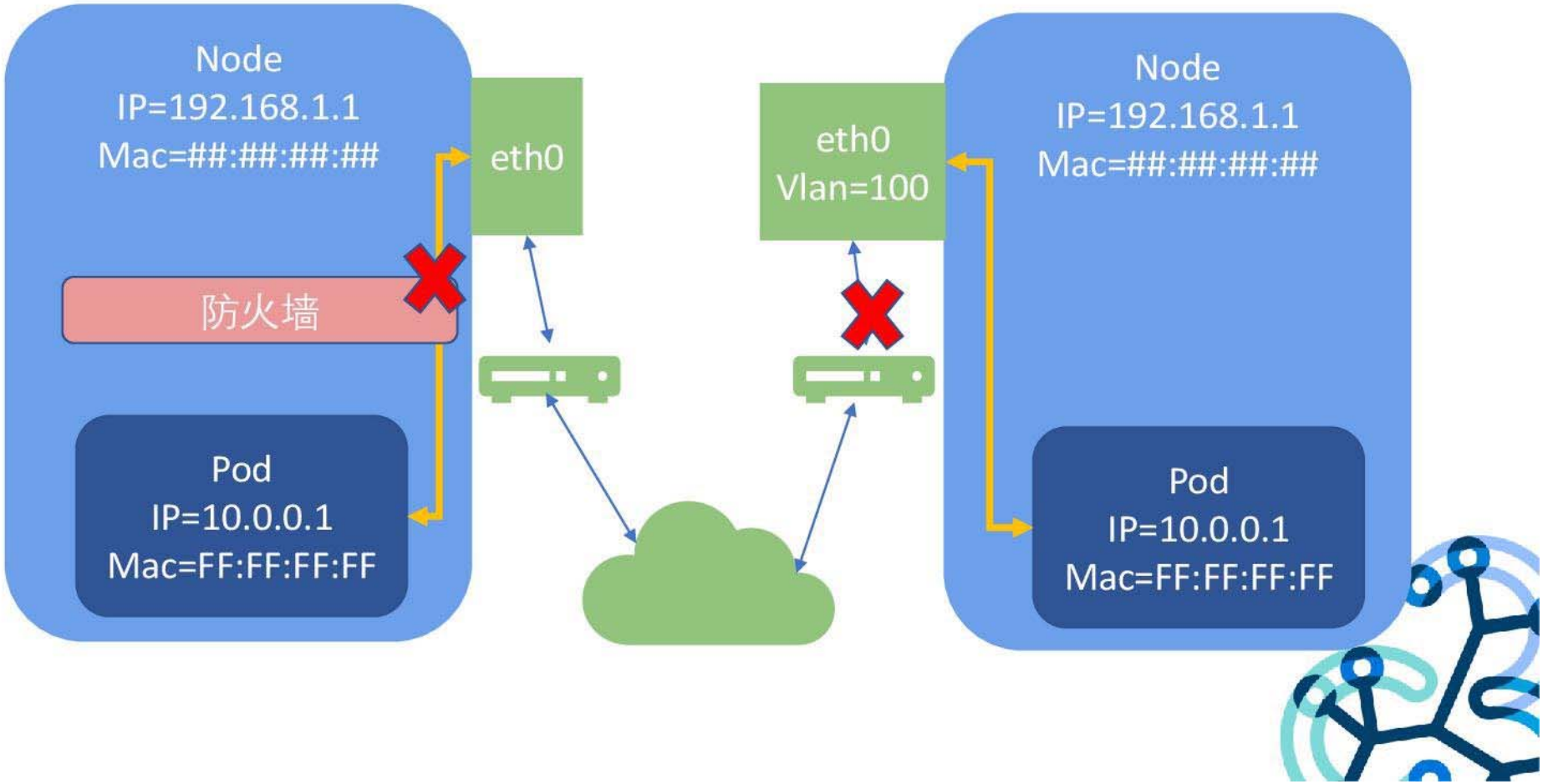


四 、基础建设之网络遇到的问题

公有云网络

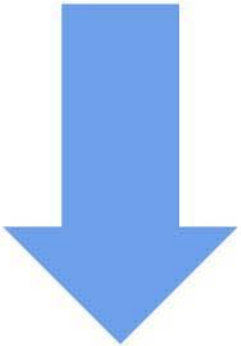
内网网络

虚拟IP会被防火墙拦截



四、公有云网络方案选型

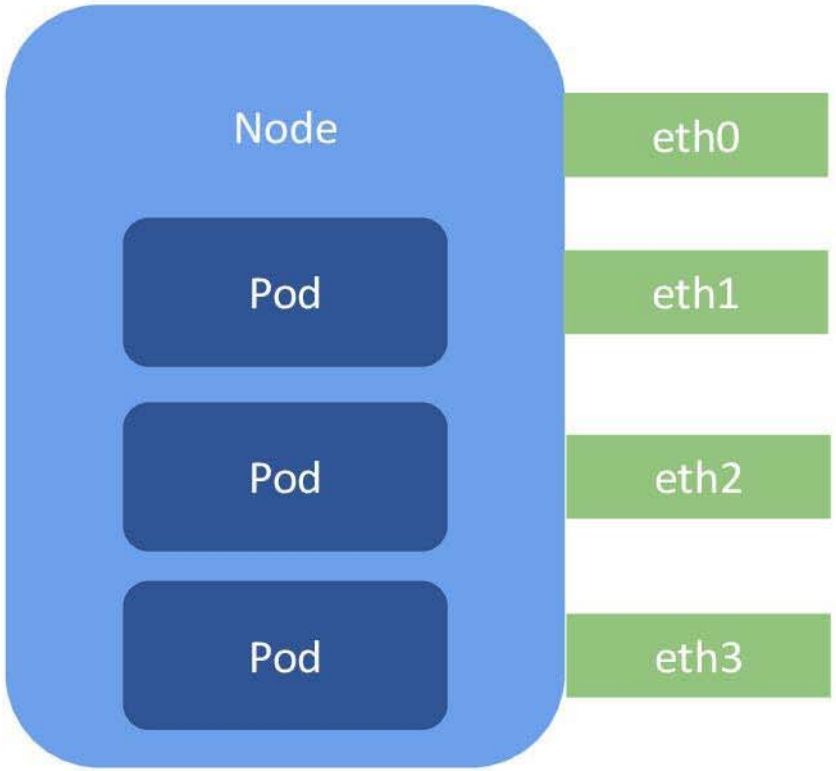
需求	隧道类	BGP类
性能	损耗10%～20%	没有损耗
支持公有云	支持	不支持
结论	不使用	不使用



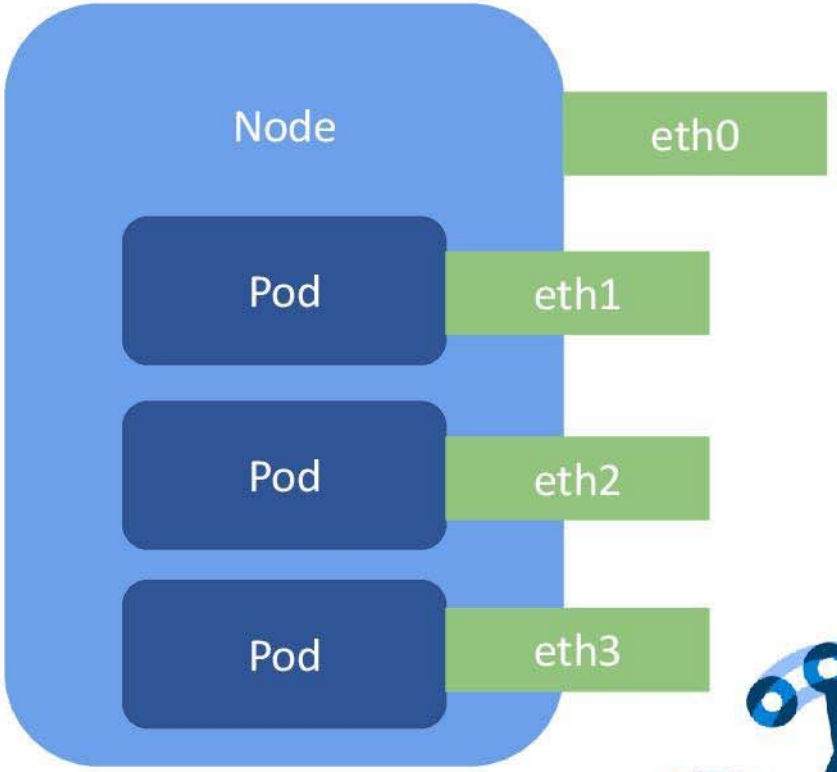
弹性网卡+CNI-Host-Device

四 、公有云网络方案

弹性网卡负责虚拟出
多块带可用IP的虚拟网卡

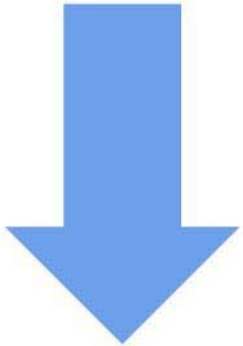


Host-Device 负责把虚拟网卡
插入到容器里面



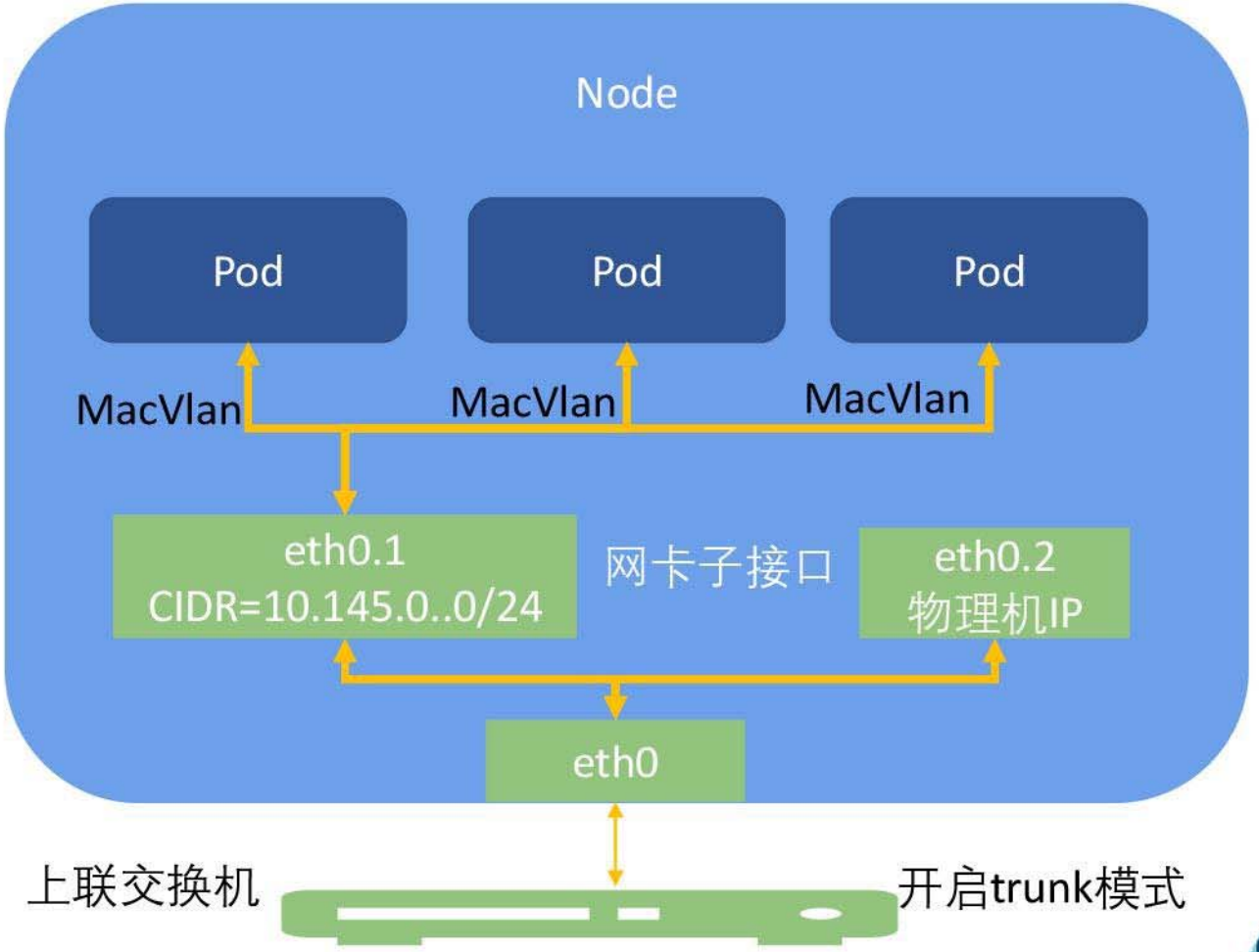
四、内网网络方案选型

需求	隧道类	BGP类
性能	损耗10% ~ 20%	会使用IPTables做ACL, 容易丢包
结论	不使用	需要做二次开发后使用

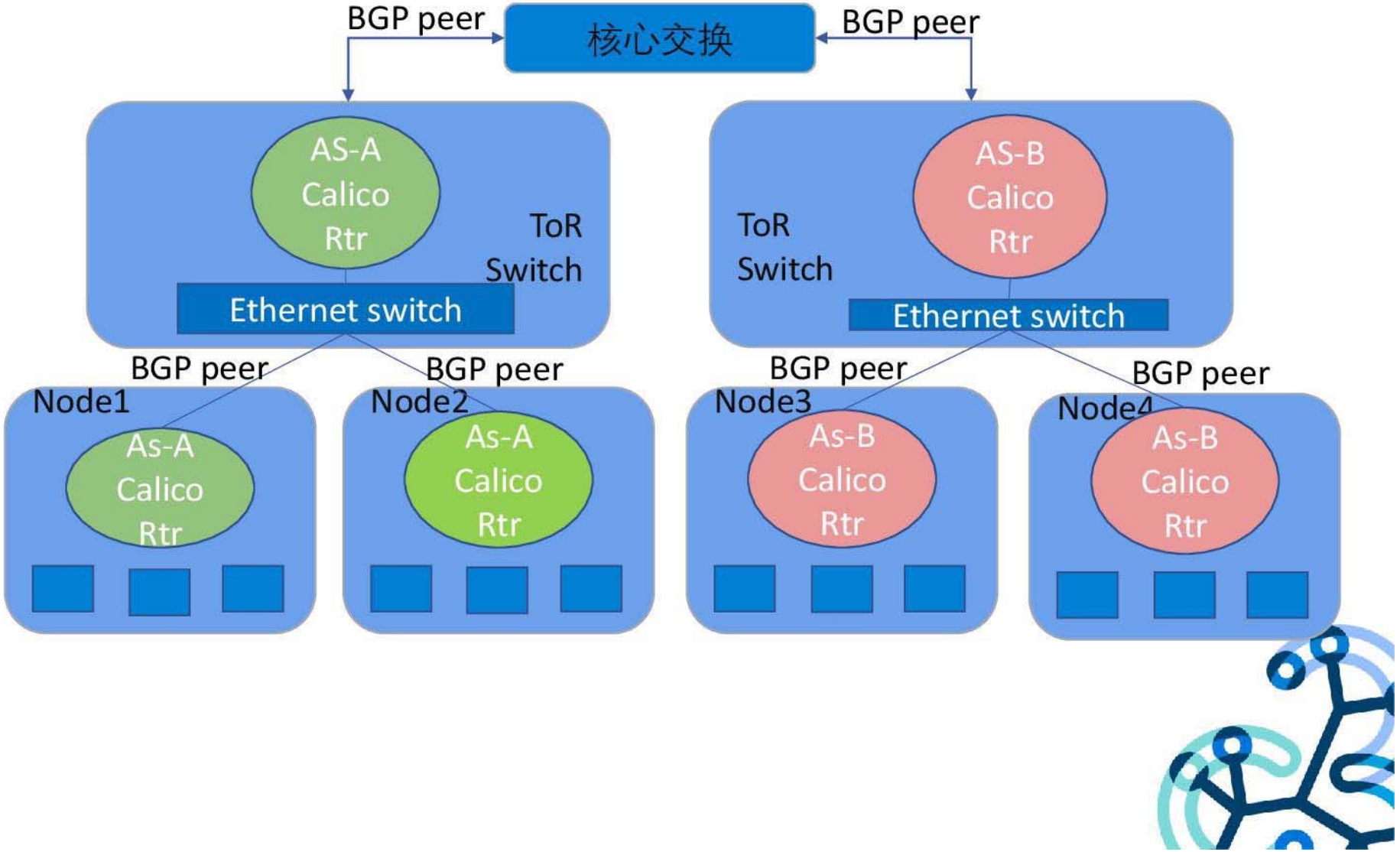


Vlan+MacVlan

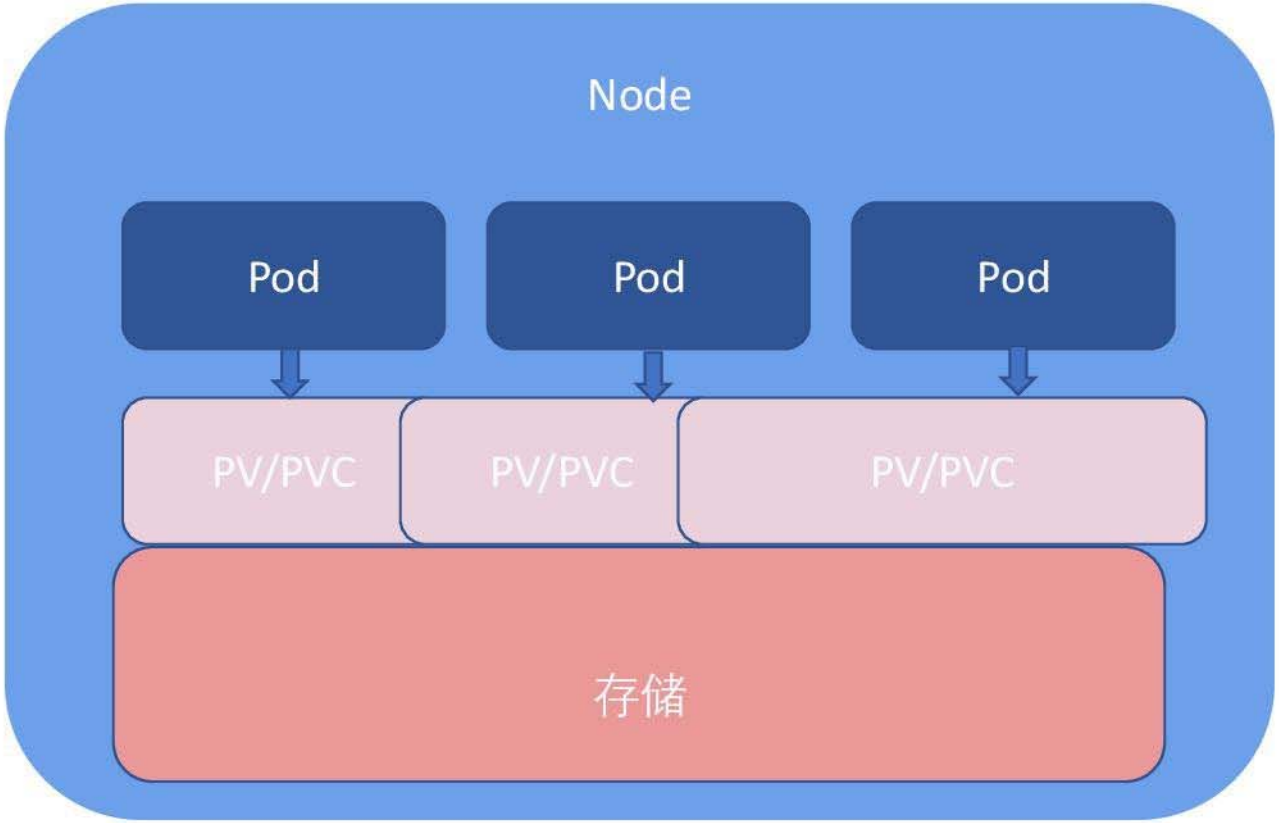
四 、内网网络方案


Node
Pod
Pod
Pod
MacVlan
MacVlan
MacVlan
eth0.1
CIDR=10.145.0..0/24
网卡子接口
eth0.2
物理机IP
eth0
上联交换机
开启trunk模式

四 、内网BGP方案

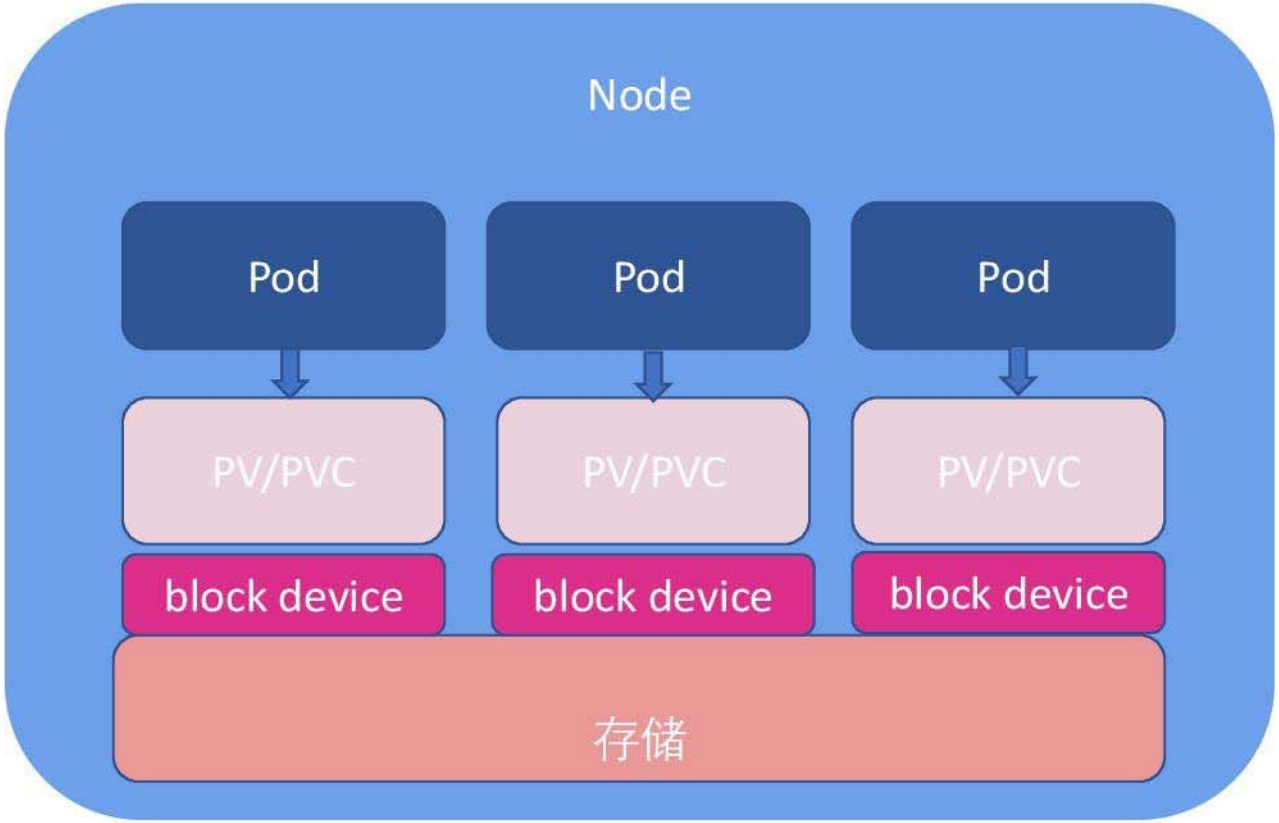

BGP peer
核心交换
BGP peer
AS-A
Calico
Rtr
ToR
Switch
Ethernet switch
AS-B
Calico
Rtr
ToR
Switch
Ethernet switch
BGP peer
BGP peer
BGP peer
BGP peer
Node1
As-A
Calico
Rtr
Node2
As-A
Calico
Rtr
Node3
As-B
Calico
Rtr
Node4
As-B
Calico
Rtr

四 、基础建设之存储遇到的问题



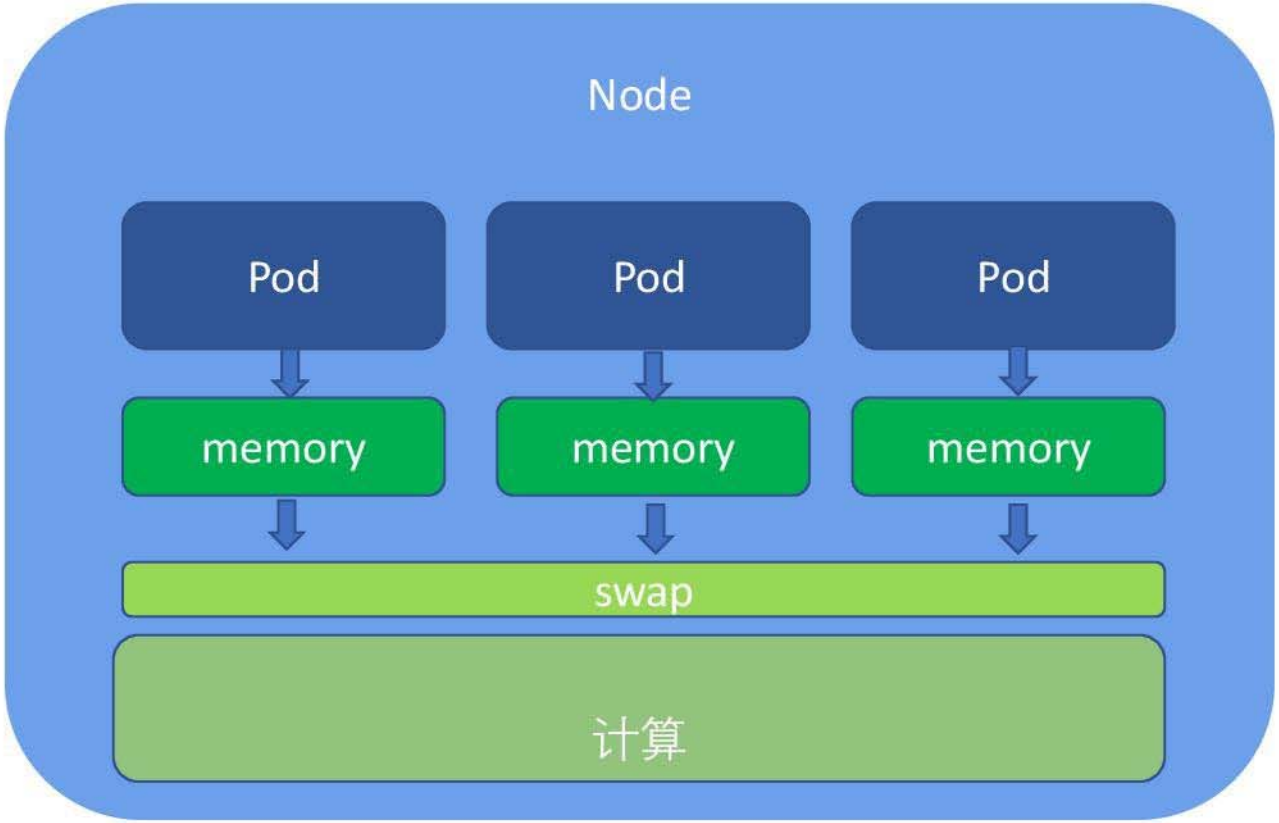
不提供有配额限制的本地静态存储

四 、静态存储方案



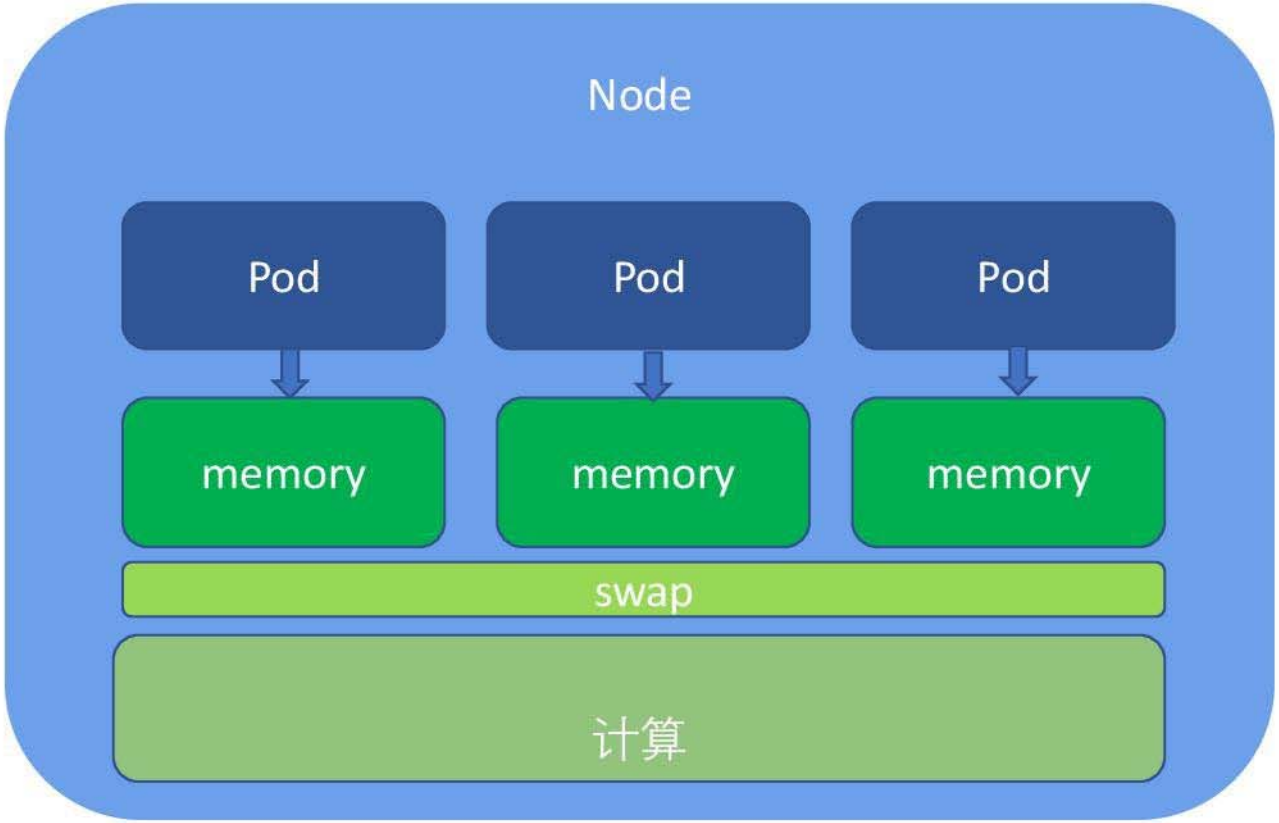
通过块设备实现本地静态存储

四、基础建设之计算遇到的问题



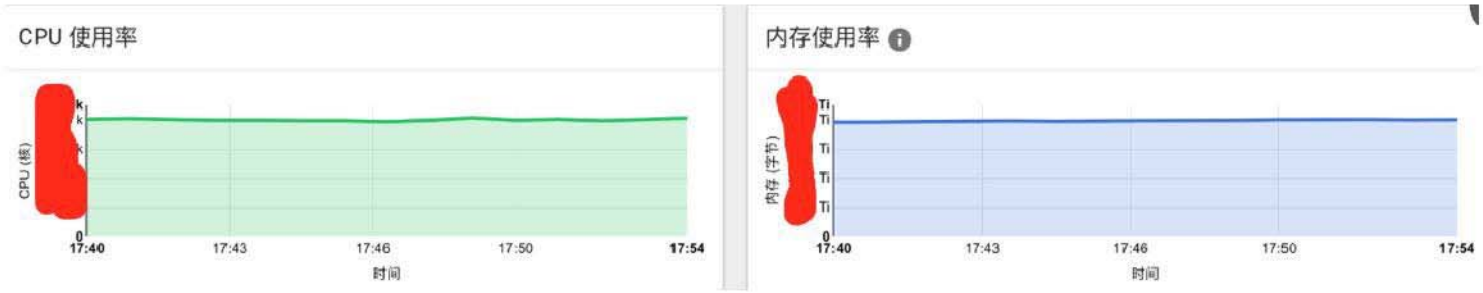
没有限制Pod访问物理机swap

四、基础建设之计算遇到的问题



Kubernetes通过cgroups实现的内存限制，可以加以修改

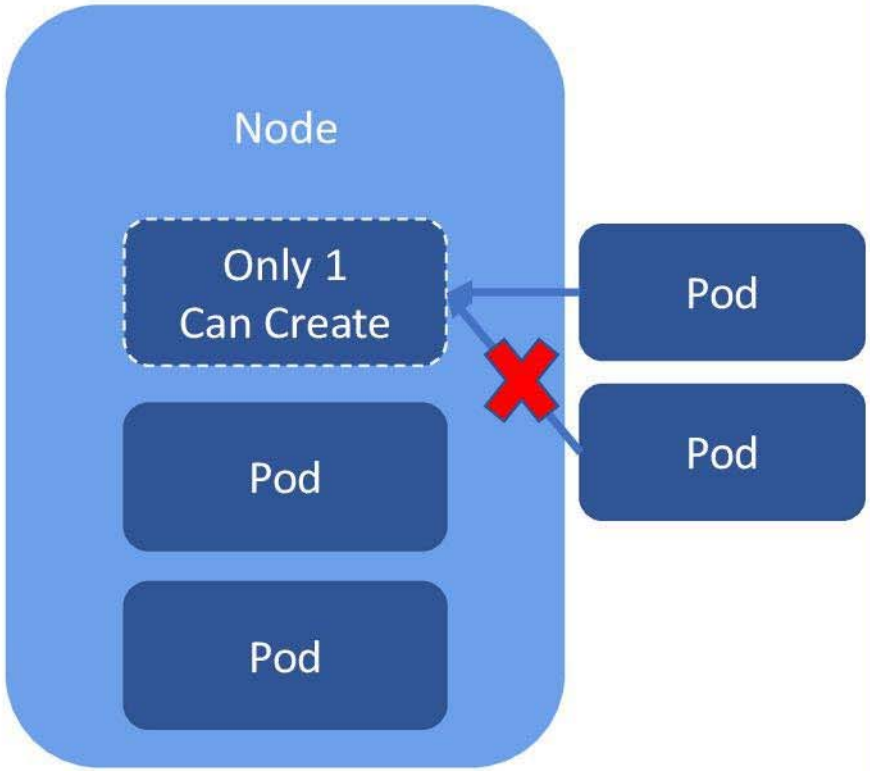
四 、资源池化的总结和归纳



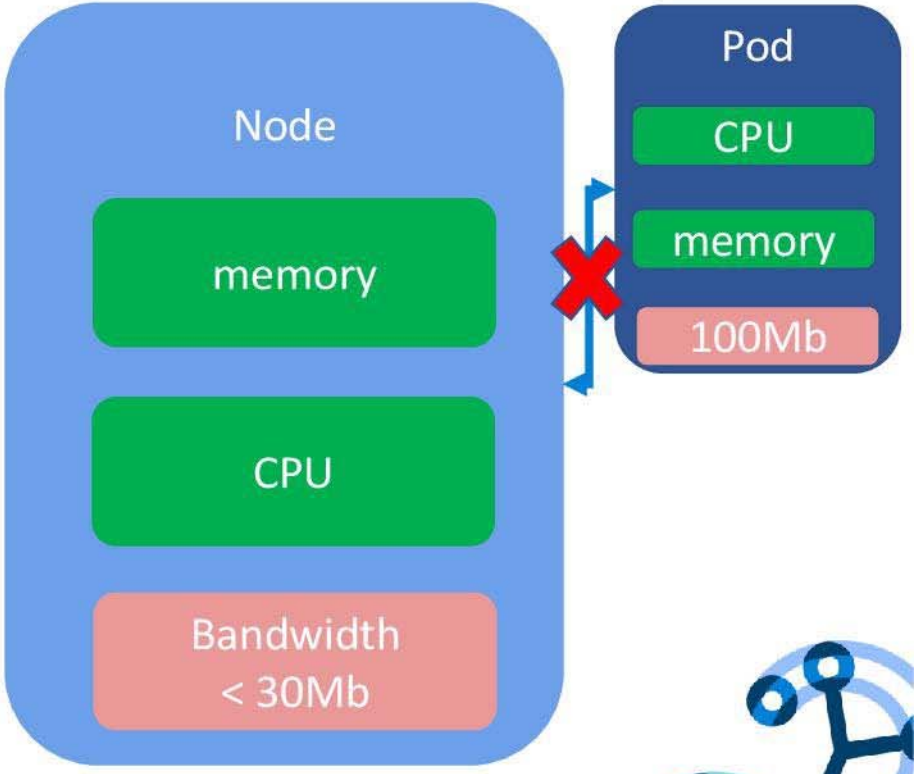
在不增加成本的情况下通过Kubernetes提供了资源核心服务池30% ~ 50%算力的计算资源

四、Kubernetes的调度存在哪些问题

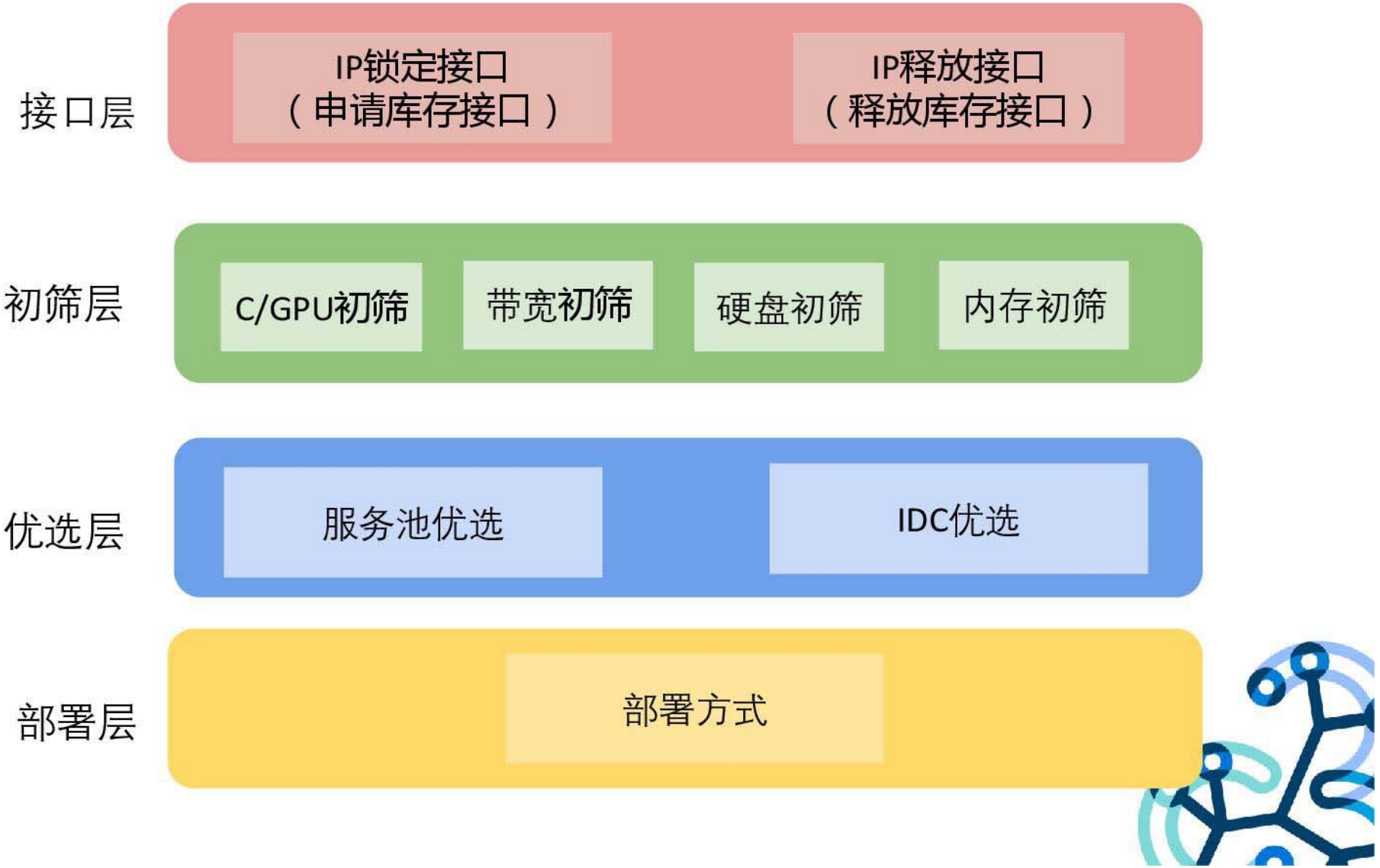
无法给出库存



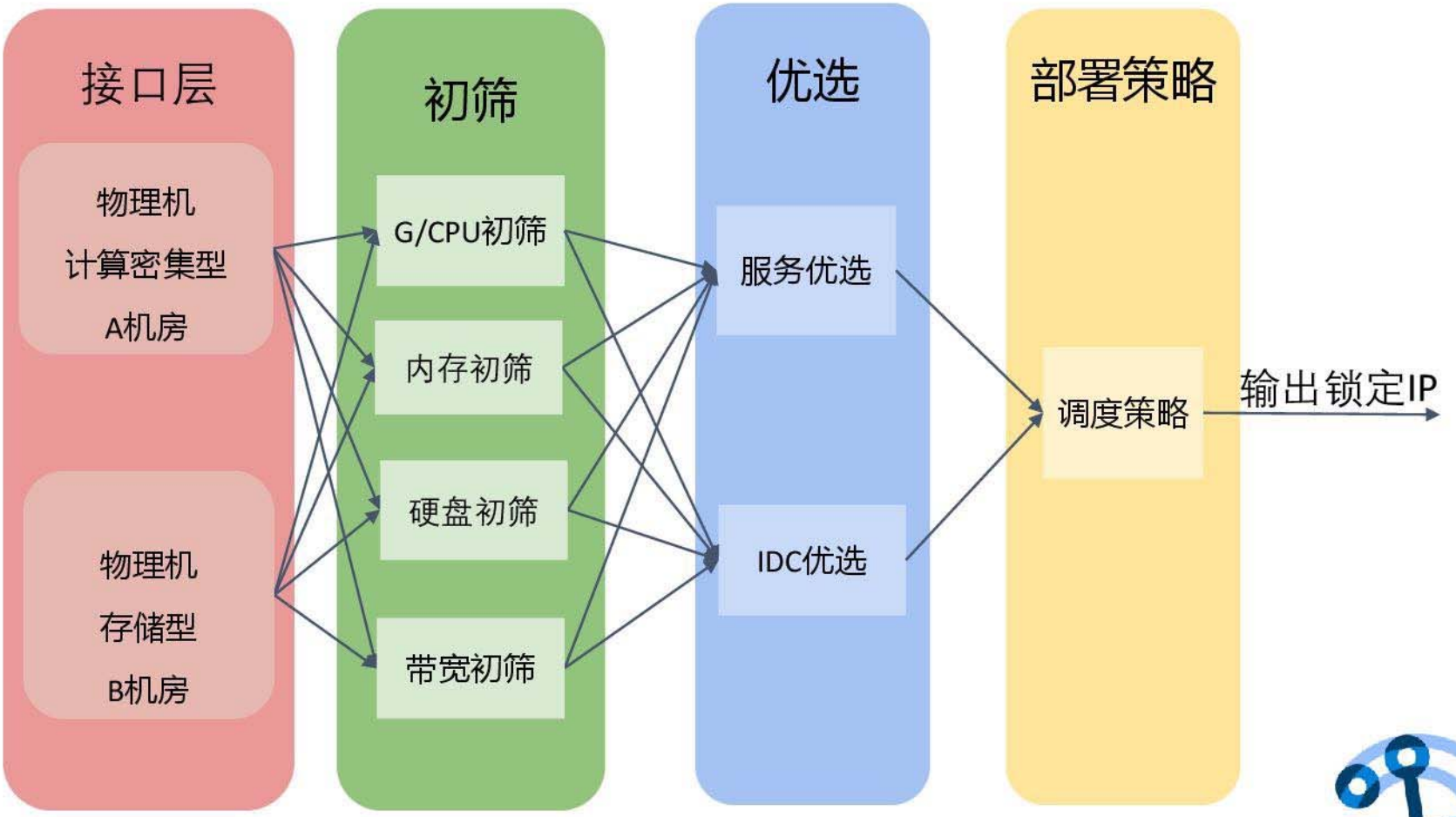
调度策略筛选维度少



四、微博容器平台调度架构



四 、微博容器精细化调度

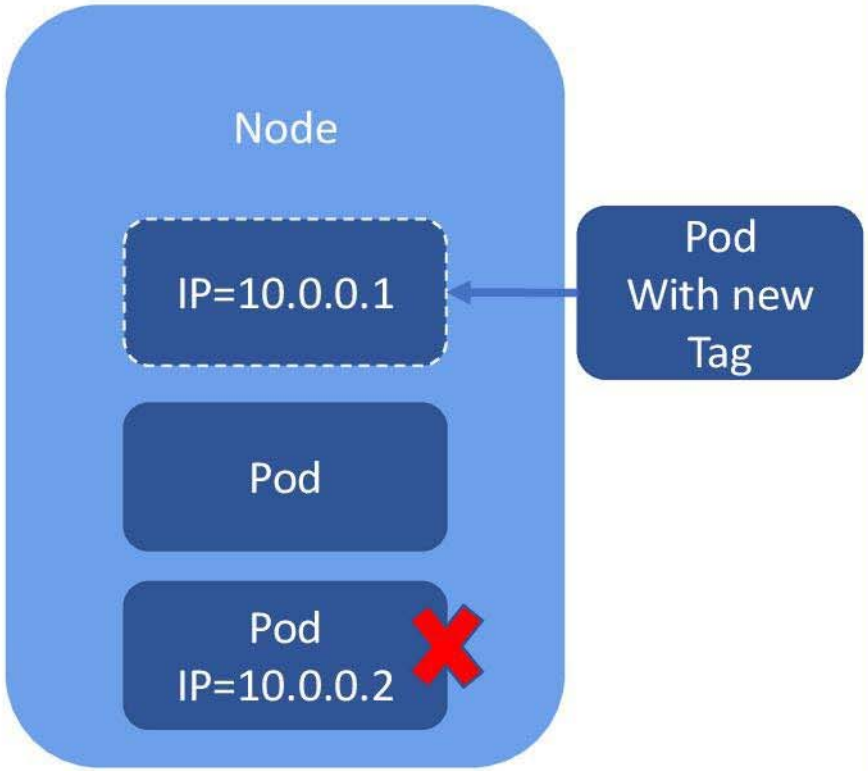

接口层
物理机
计算密集型
A机房
物理机
存储型
B机房
初筛
G/CPU初筛
内存初筛
硬盘初筛
带宽初筛
优选
服务优选
IDC优选
部署策略
调度策略
输出锁定IP

四、微博容器精细化调度和IP预分配的优势

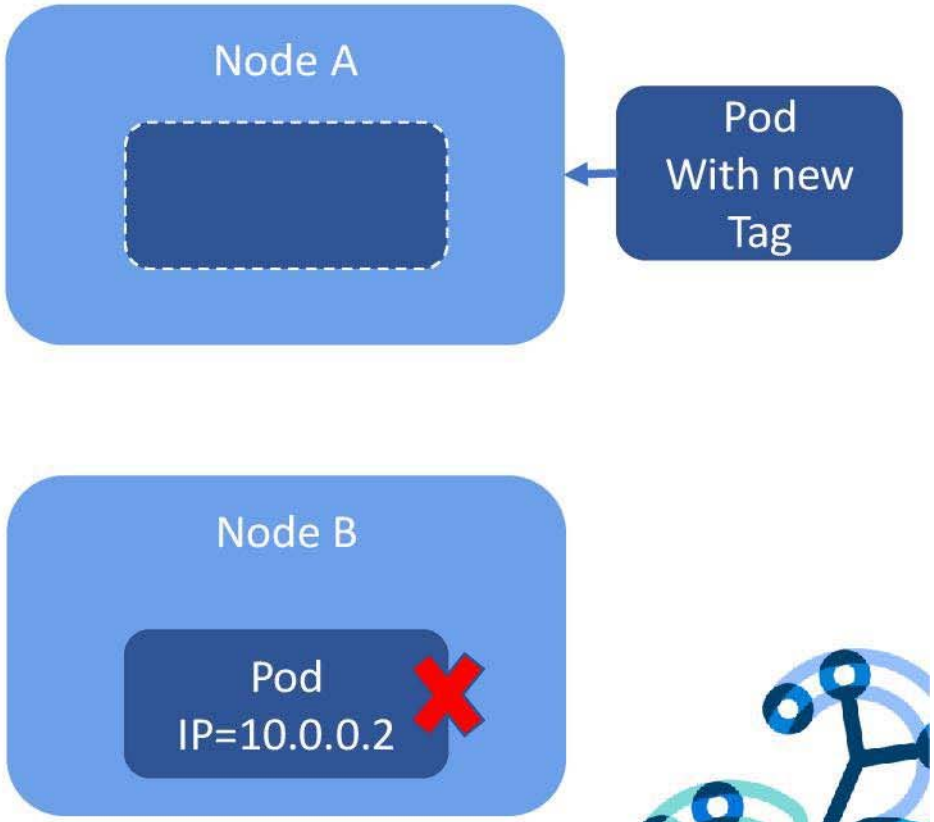
需求	IAAS层弹性扩容	微博容器平台扩容
耗时	1分钟创建机器+3分钟开机+3分钟服务启动完成+30秒7层负载变更=6分30秒	只需要3分钟
结论	应对峰值流量有6分多钟的窗口期	耗时只有IAAS方案的一半

四、Kubernetes滚动发布有什么问题

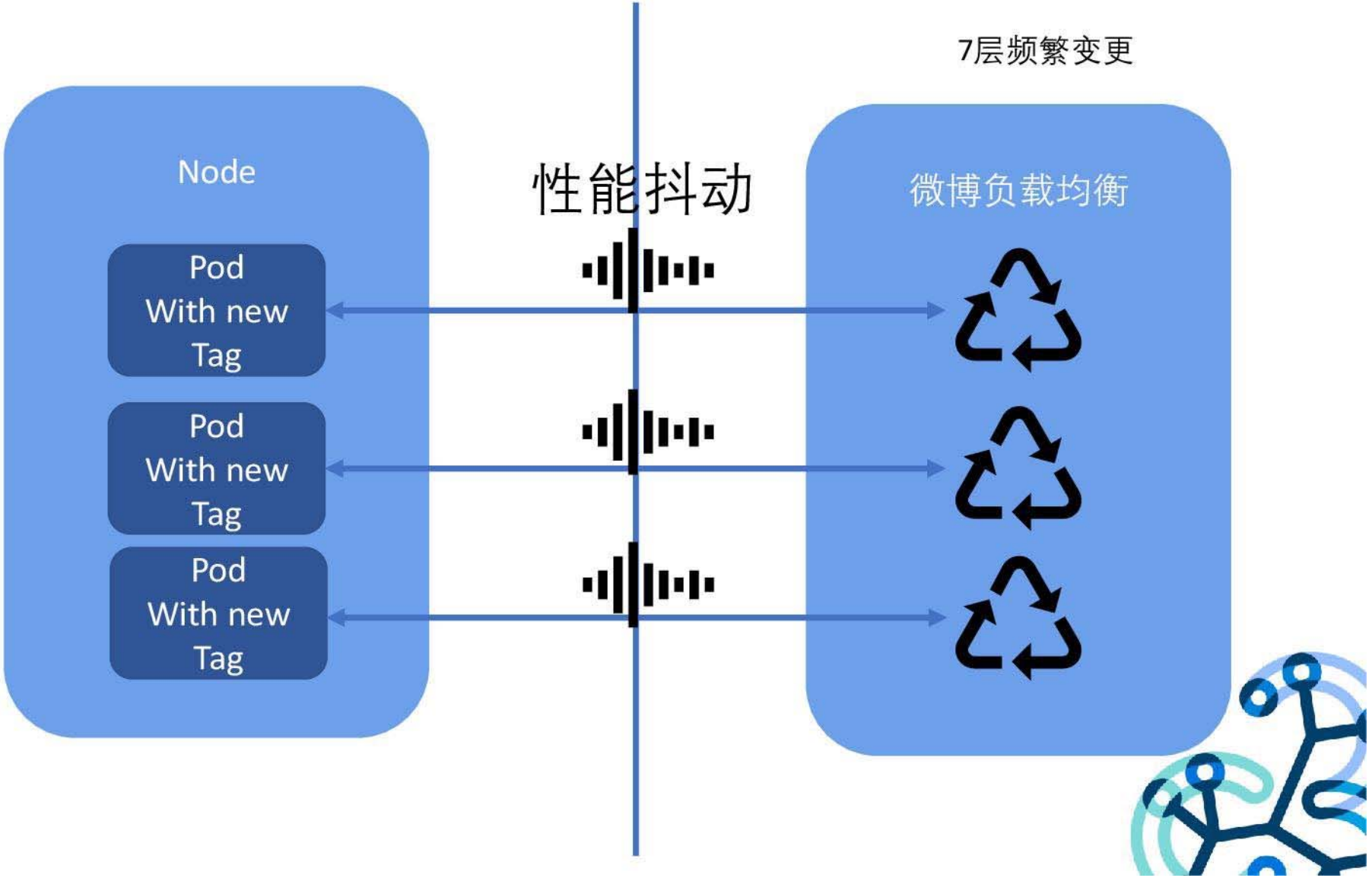
IP无法固定



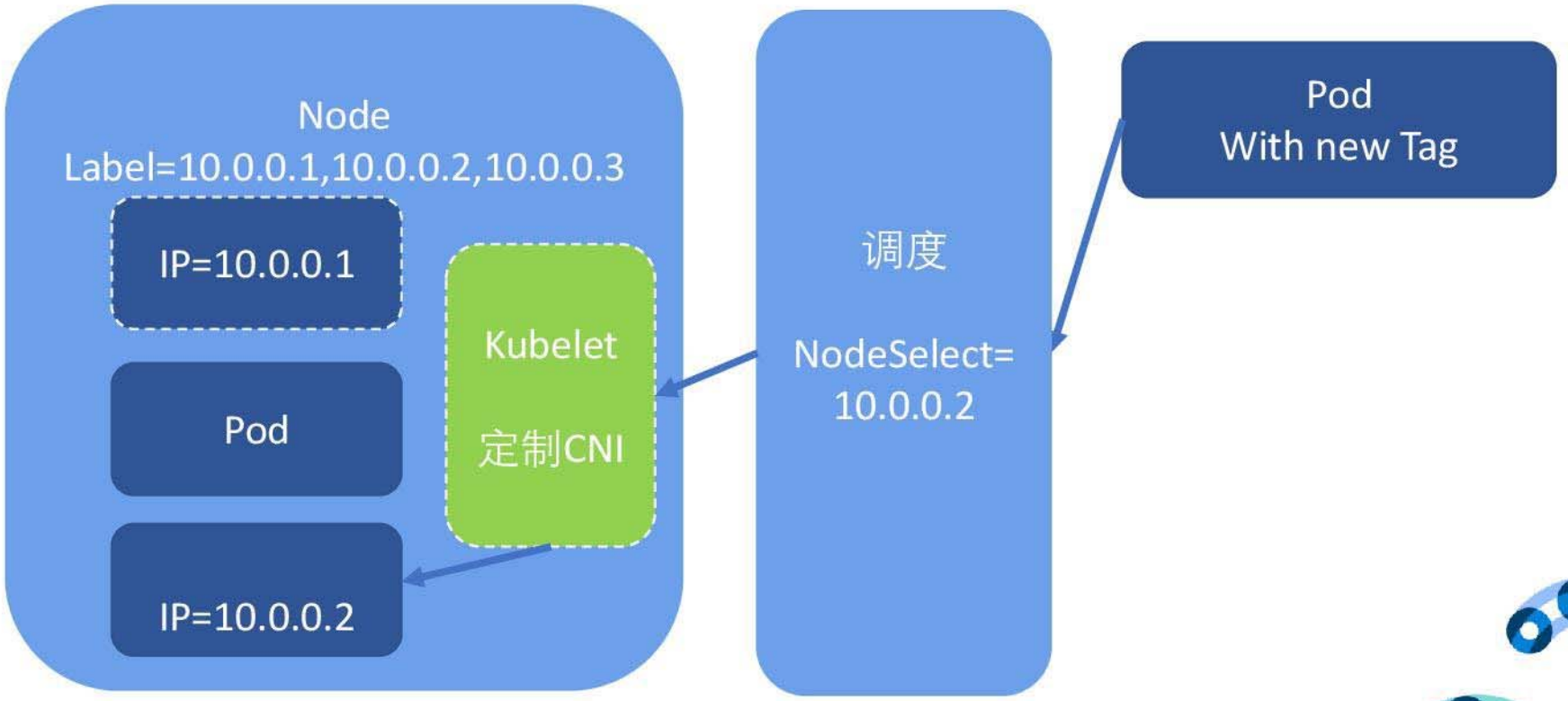
不支持原地升级



四、Kubernetes滚动发布有什么问题



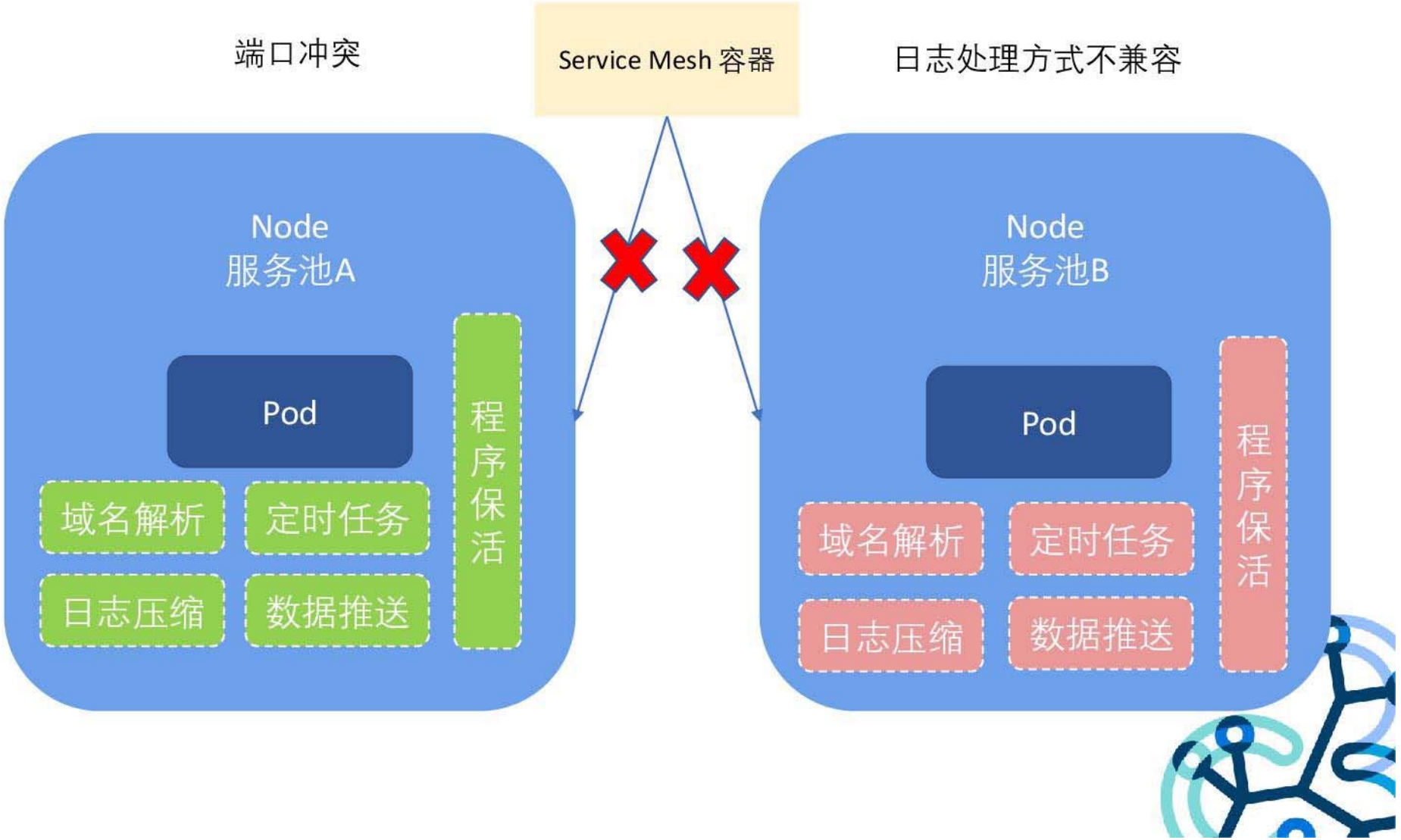
四 、微博容器平台支持In-place rolling update



四、In-place rolling update的优劣

需求	In-place rolling update	Kubernetes原生
服务是否有损	有损（步长数）	无损
变更负载均衡次数	0	副本数/步长
本地日志存储	不影响	有影响
结论	微博采用这种	不采用

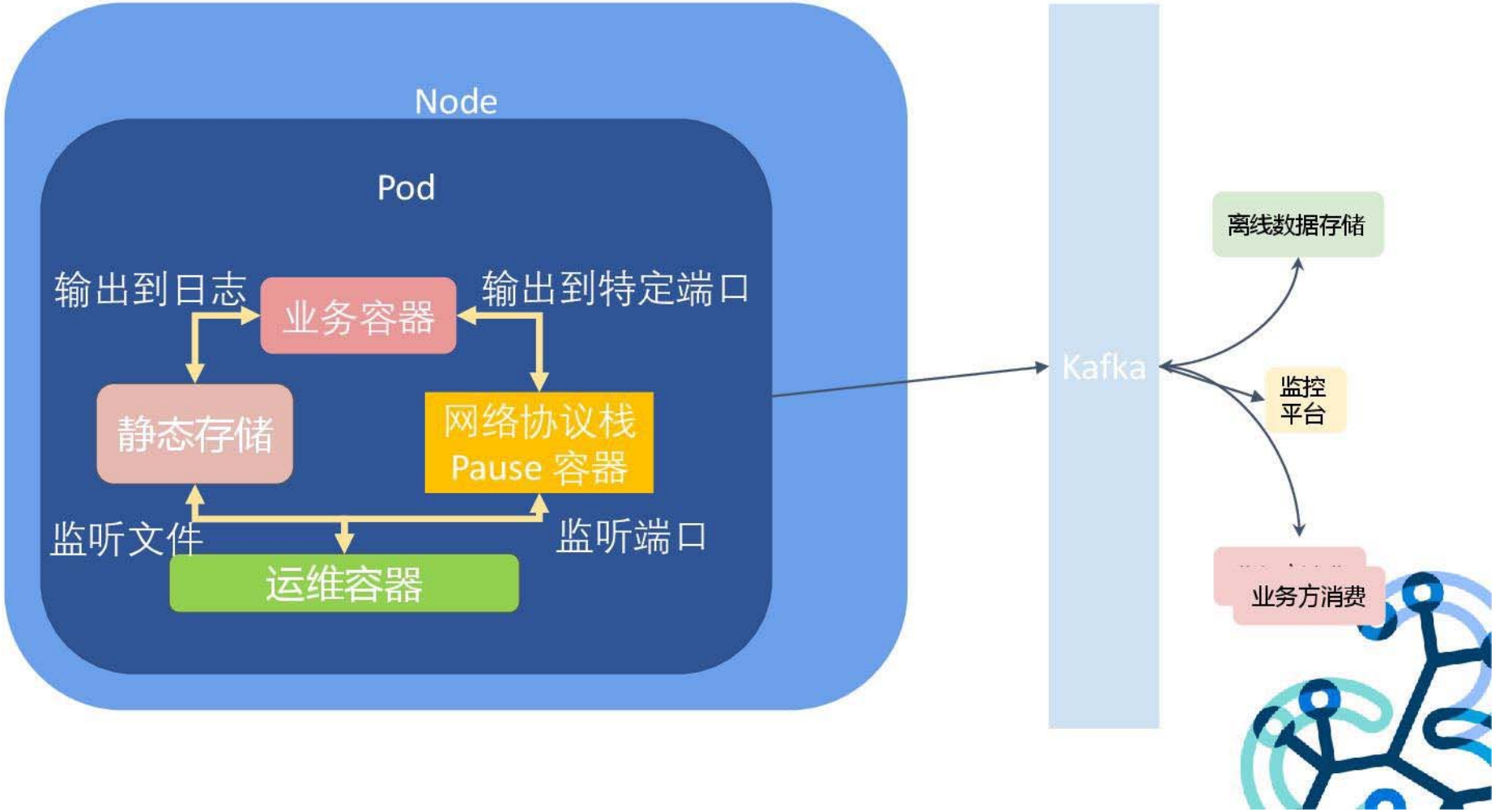
四、新业务容器整合难度大



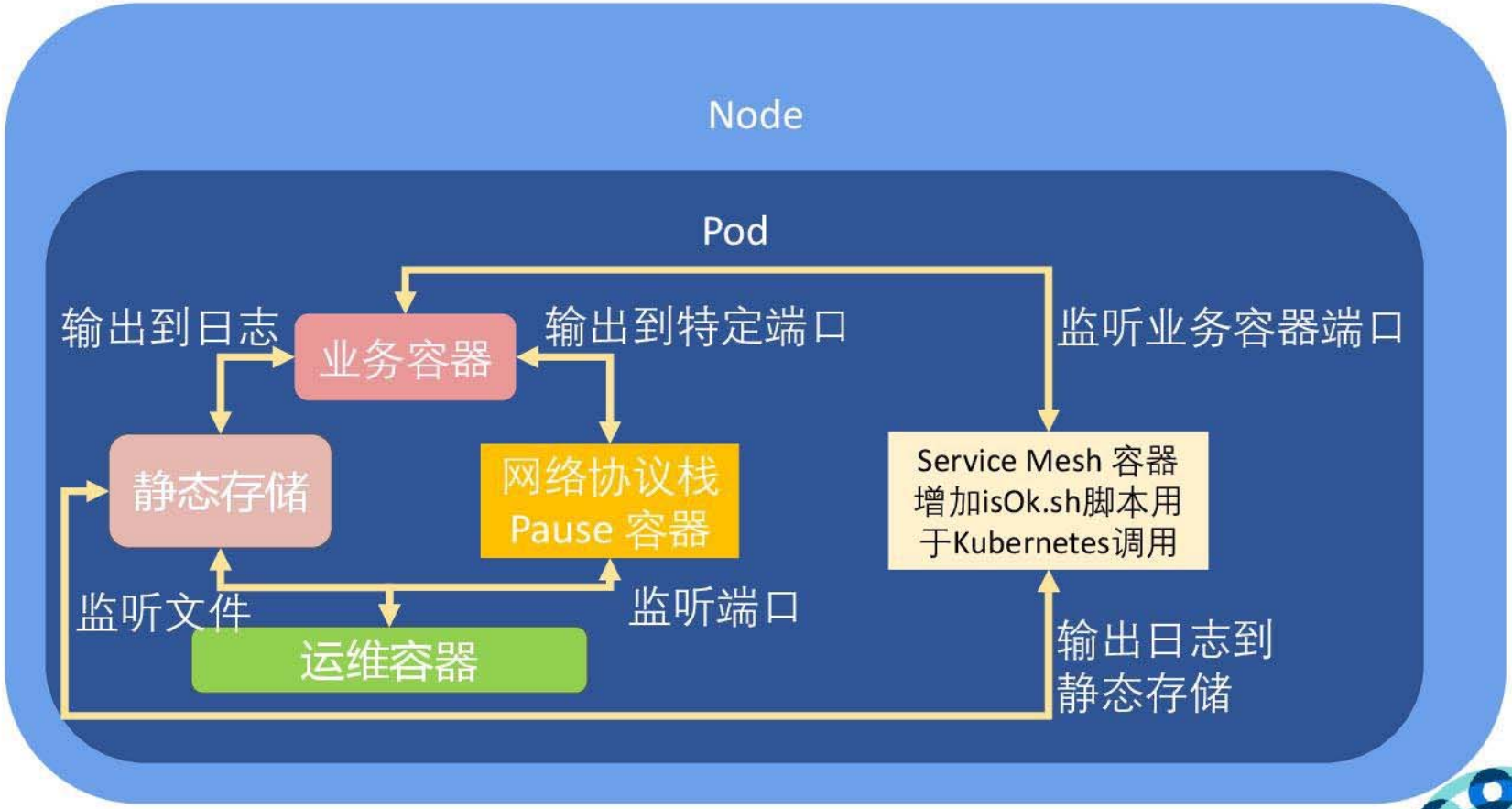
四、在容器时代的运维应该是什么样的？

1. 模块化
2. 微型化
3. 即插即用
4. 通过共享存储来工作
5. 通过共享网络来通信
6. 具有自运维能力

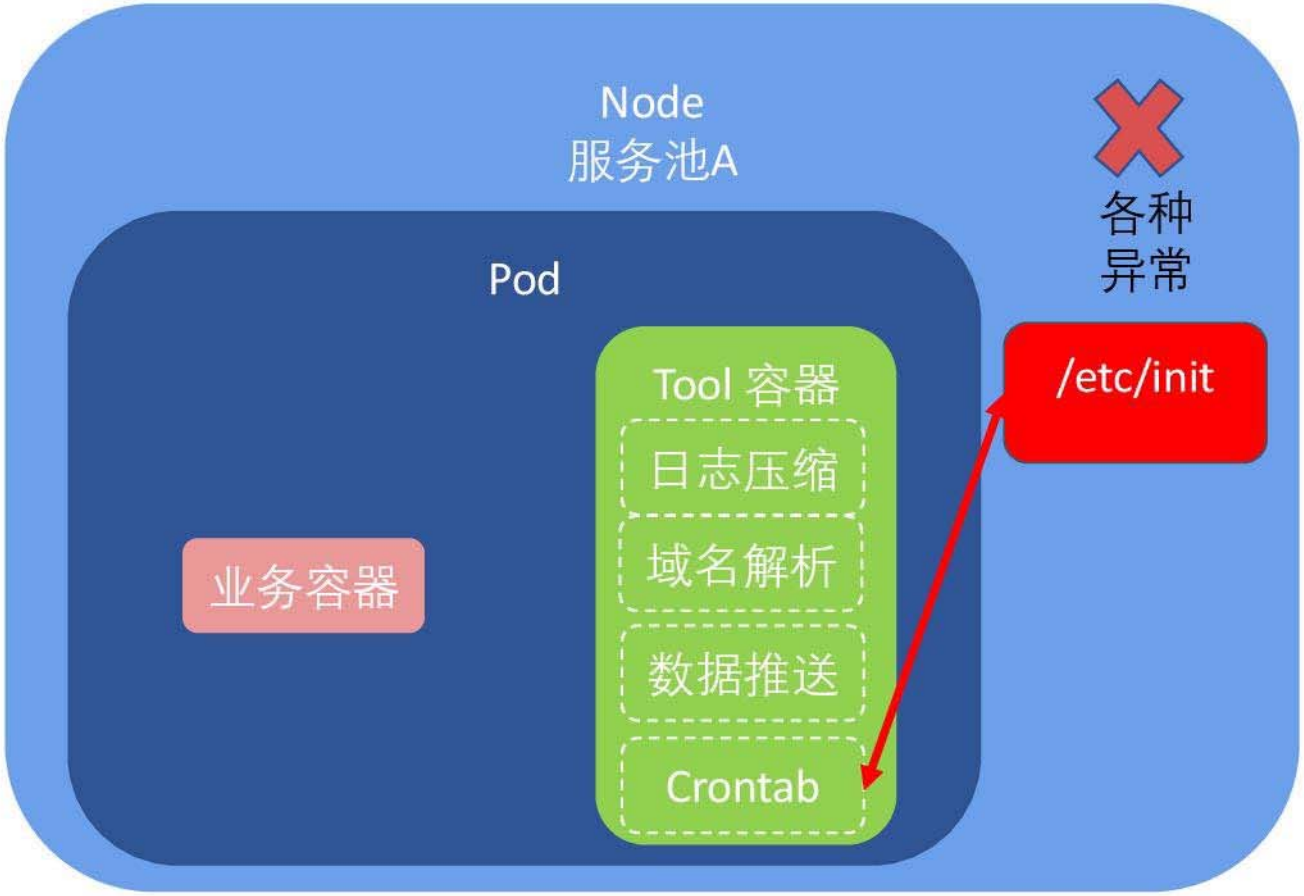
四、利用Kubernetes的Pause模型整合容器完成日志推送



四、利用KubernetesPause模型ServiceMesh容器接入



四、不要使用系统的crontab去在容器里面定时任务

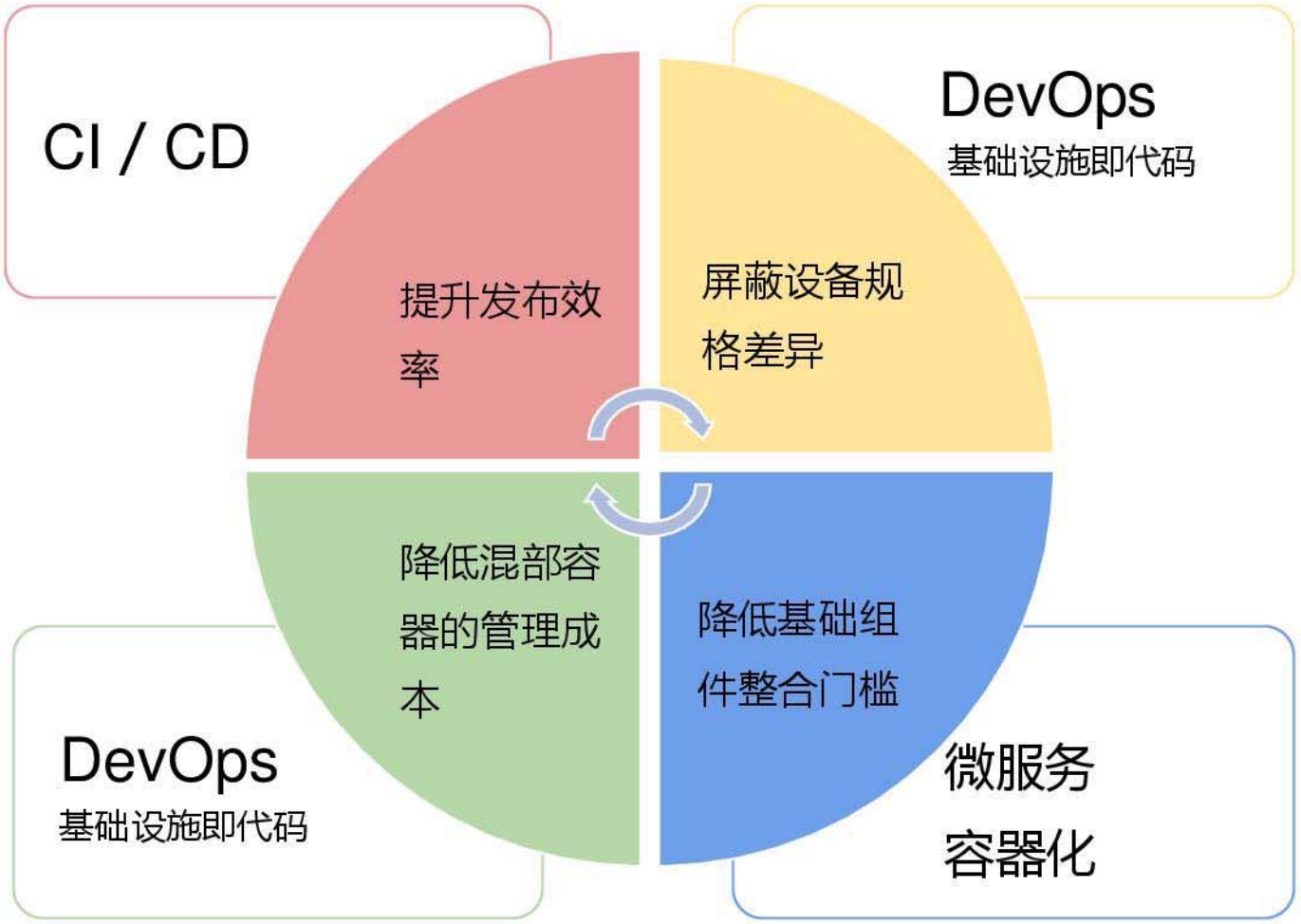


五 、总结

- 后续的演进
- QA

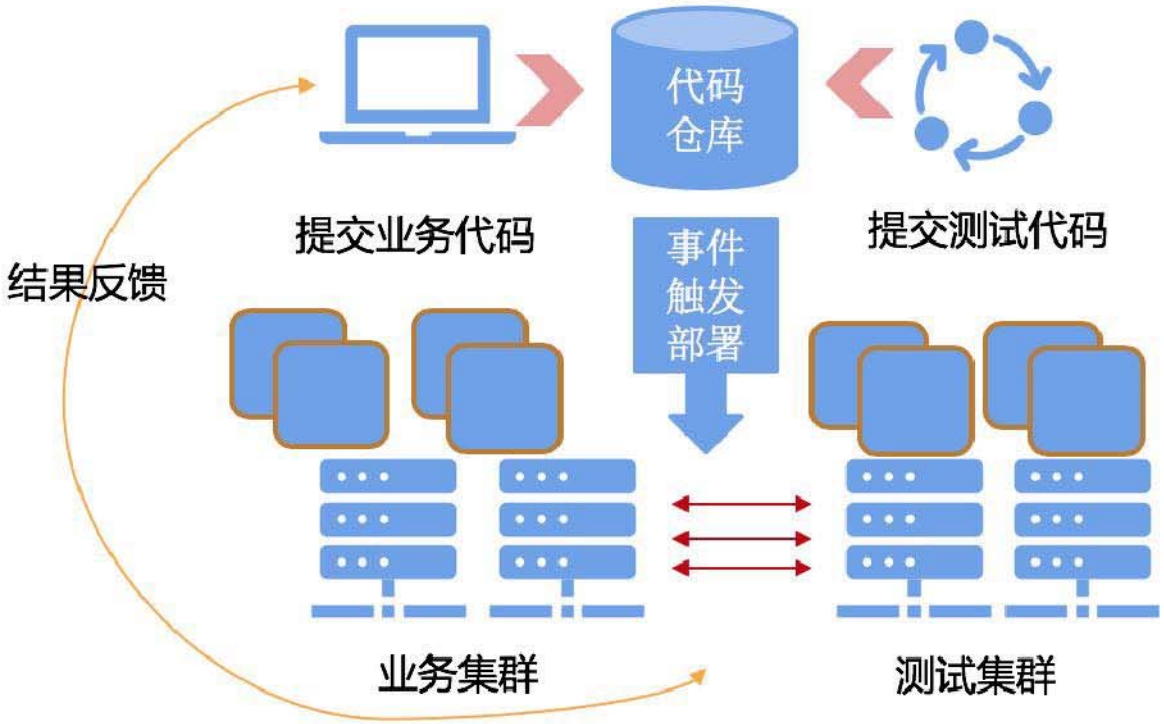
五 、后续的演进

- 向云原生的方式改进我们的服务



五 、后续的演进

- CI / CD



五 、后续的演进

- 人工智能的灰度发布生成指标指导自动化扩容