

同程私有云平台的 弹性计算架构设计与落地实践

彭涛

同程旅行 架构师



大纲

- 出行需求快速增长带来的挑战
- 私有云的弹性计算
- 自动计算服务弹性阈值
- 峰值流量下保障服务稳定性
- 弹性计算降本增效
- 自动漂移解决单机稳定性
- 一些展望和总结

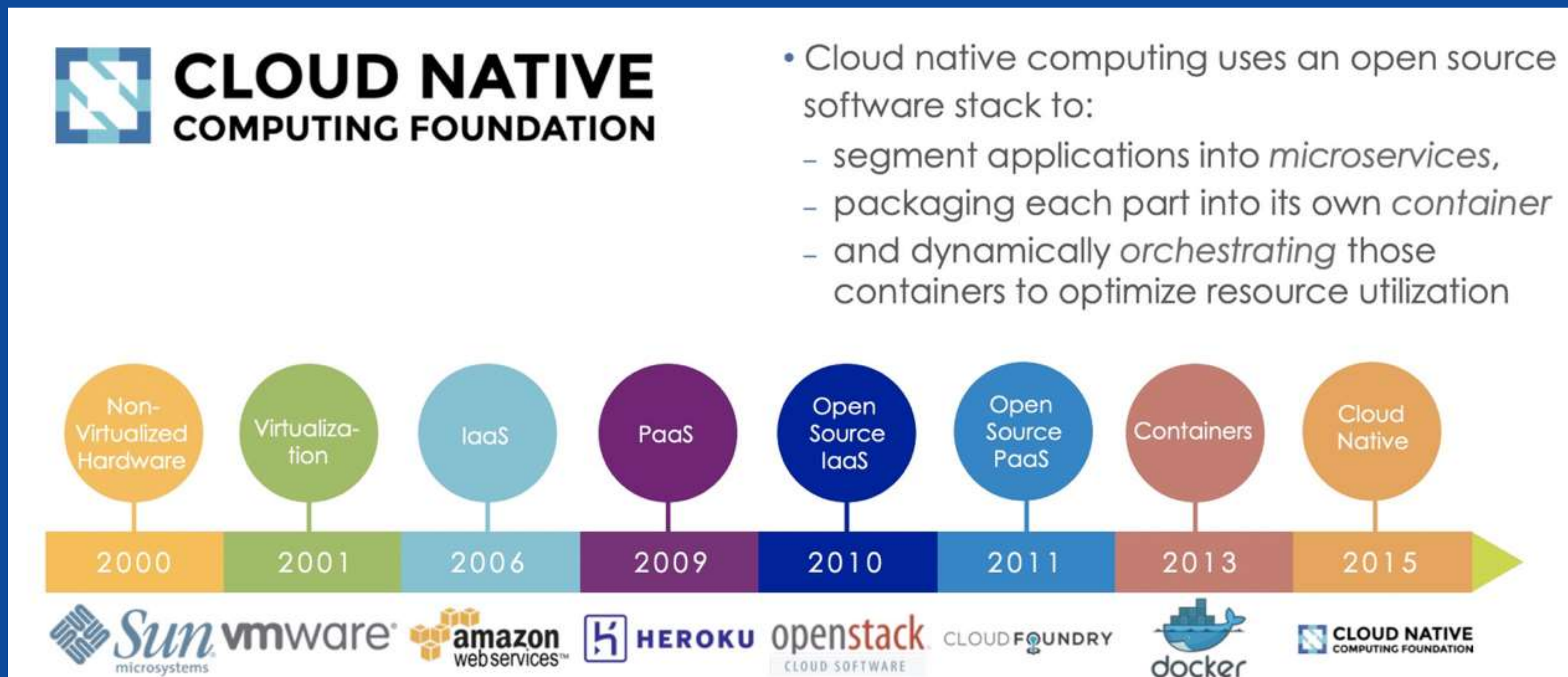
出行需求快速上涨提出的挑战



公司流量上半年多次突破历史最高水位

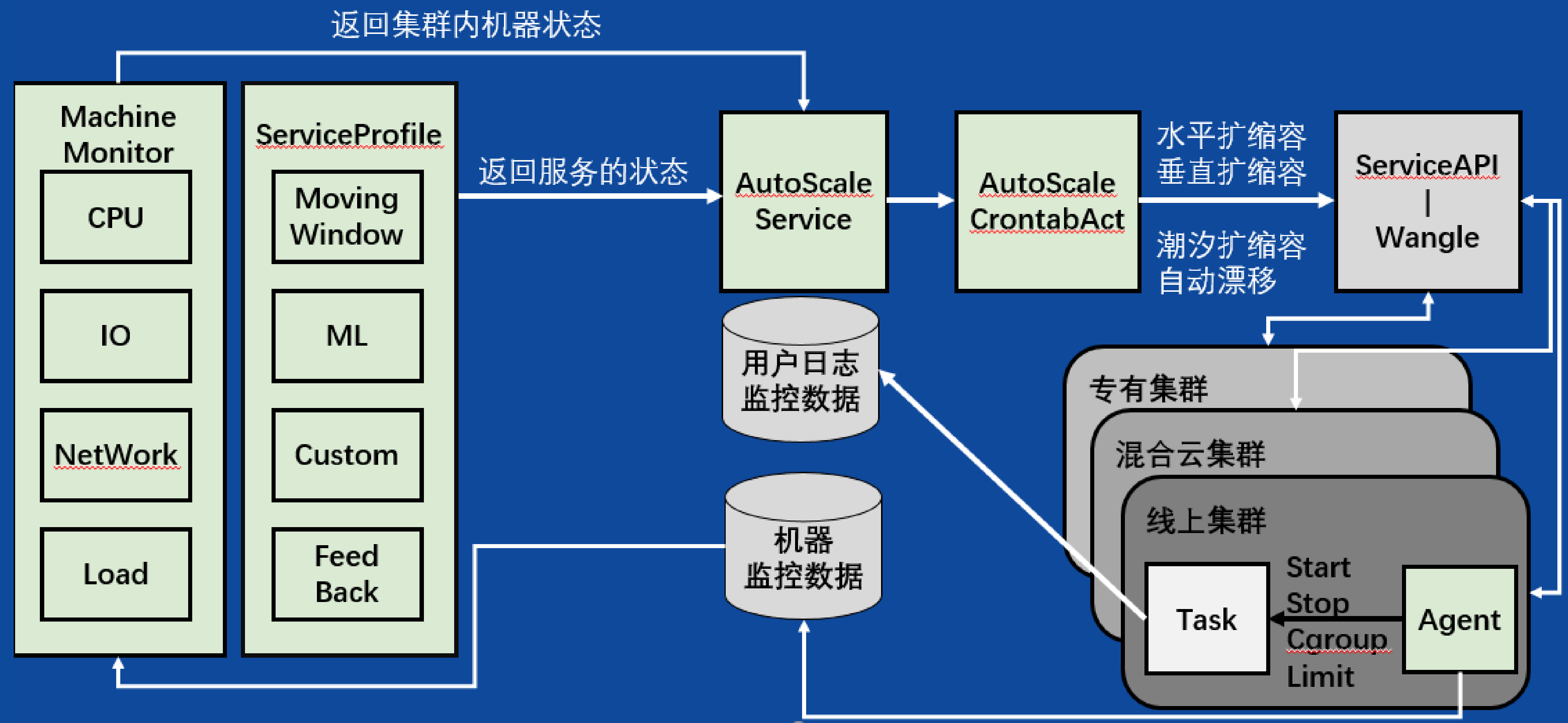


在云原生的道路上任重而道远



目前公司完成了核心业务的容器化，正在向云原生方向进行下一步的演进

弹性计算的整体框架



弹性计算的产品形态



评估服务压力情况的困难点

运维难：
压测成本高，
时效性差



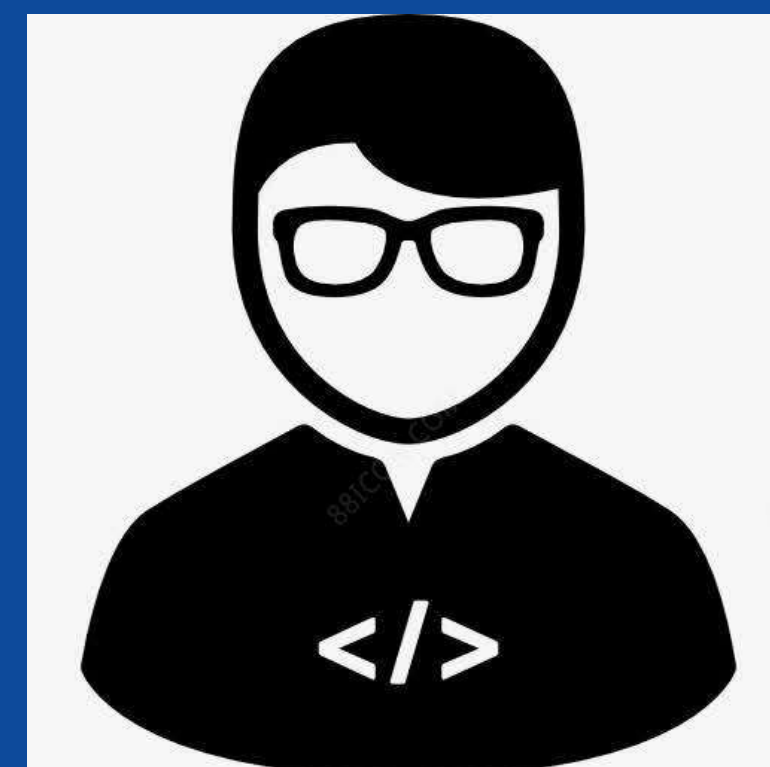
运维压测需要高度侵入代码，但是随着代码的迭代指标随之失真

厂商难：
接入成本高
指标定义难



云厂商只能做简单的测算。或者交给业务自身去判断，接入成本较高

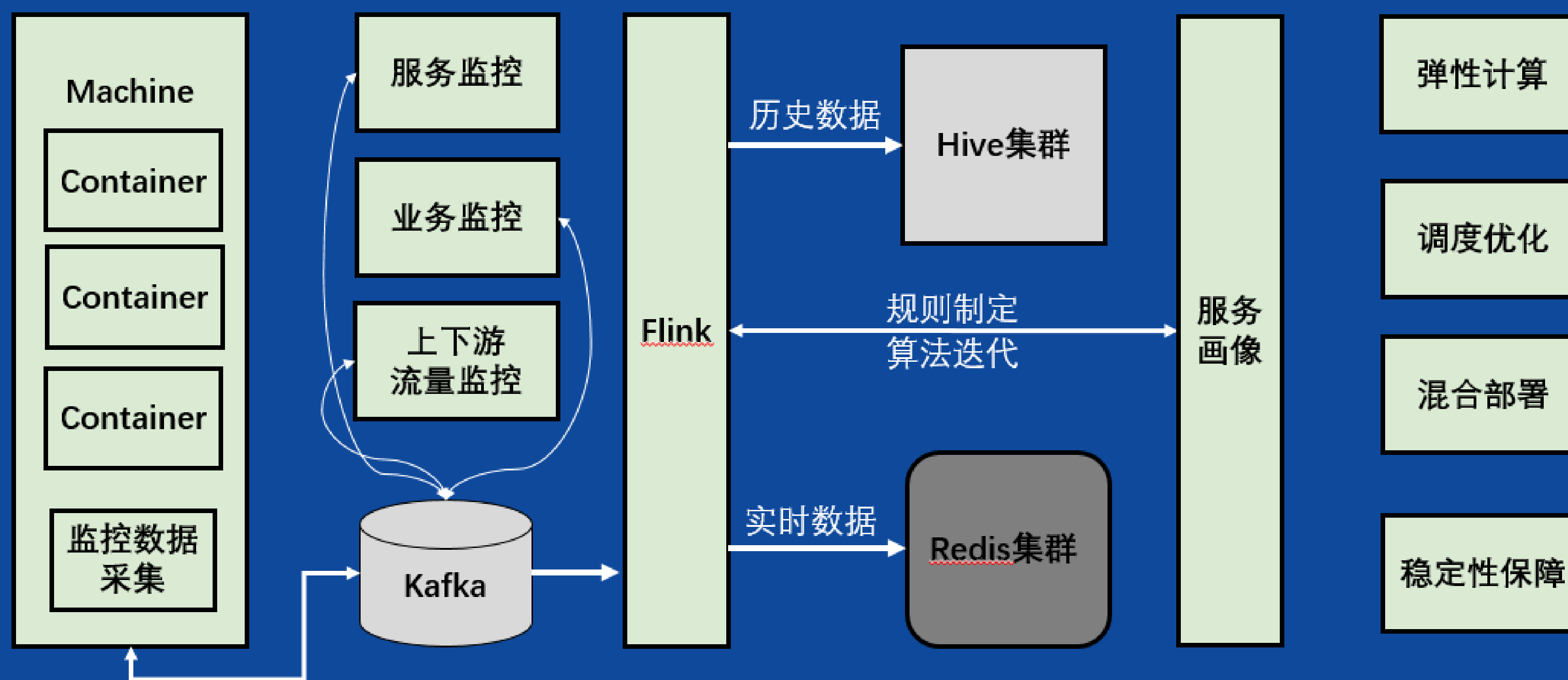
研发难：
操作成本高
影响范围广



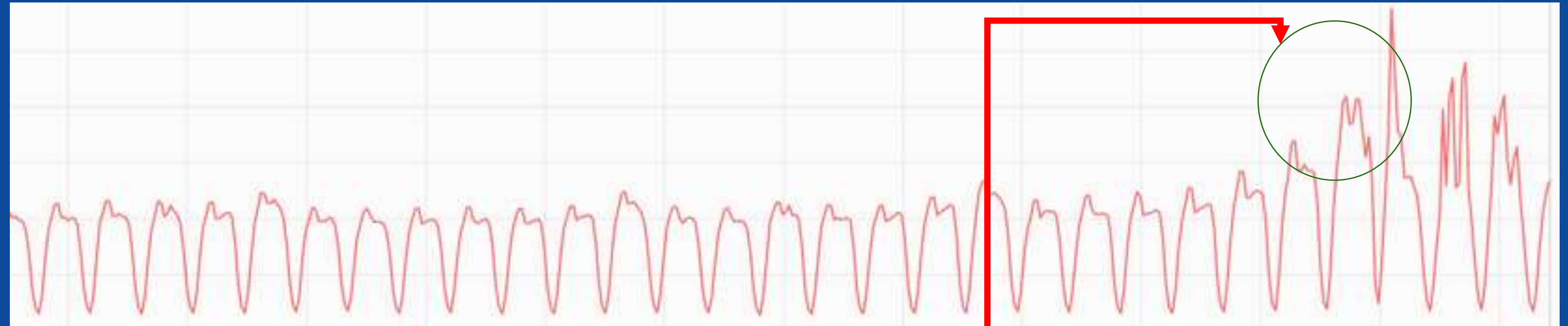
研发评估需要联动上下游一起
操作成本很大

服务画像自动预测应用阈值

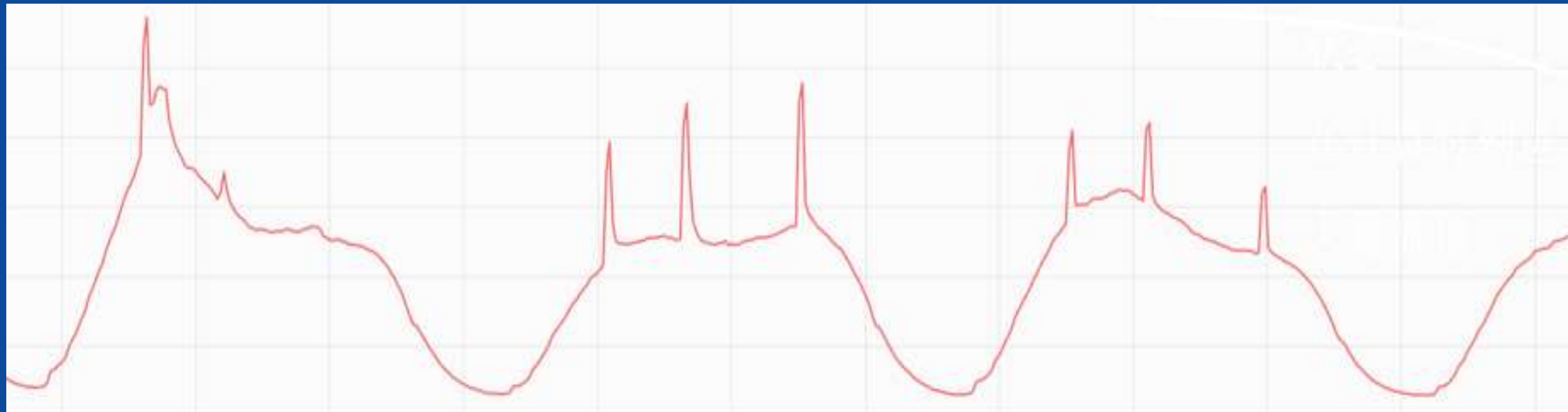
- 1.使用类似Google的Autopilot的时间滑动窗口算法建设的服务画像
- 2.利用时序数据更新和预测应用阈值和压力情况。



峰值流量的表现形式

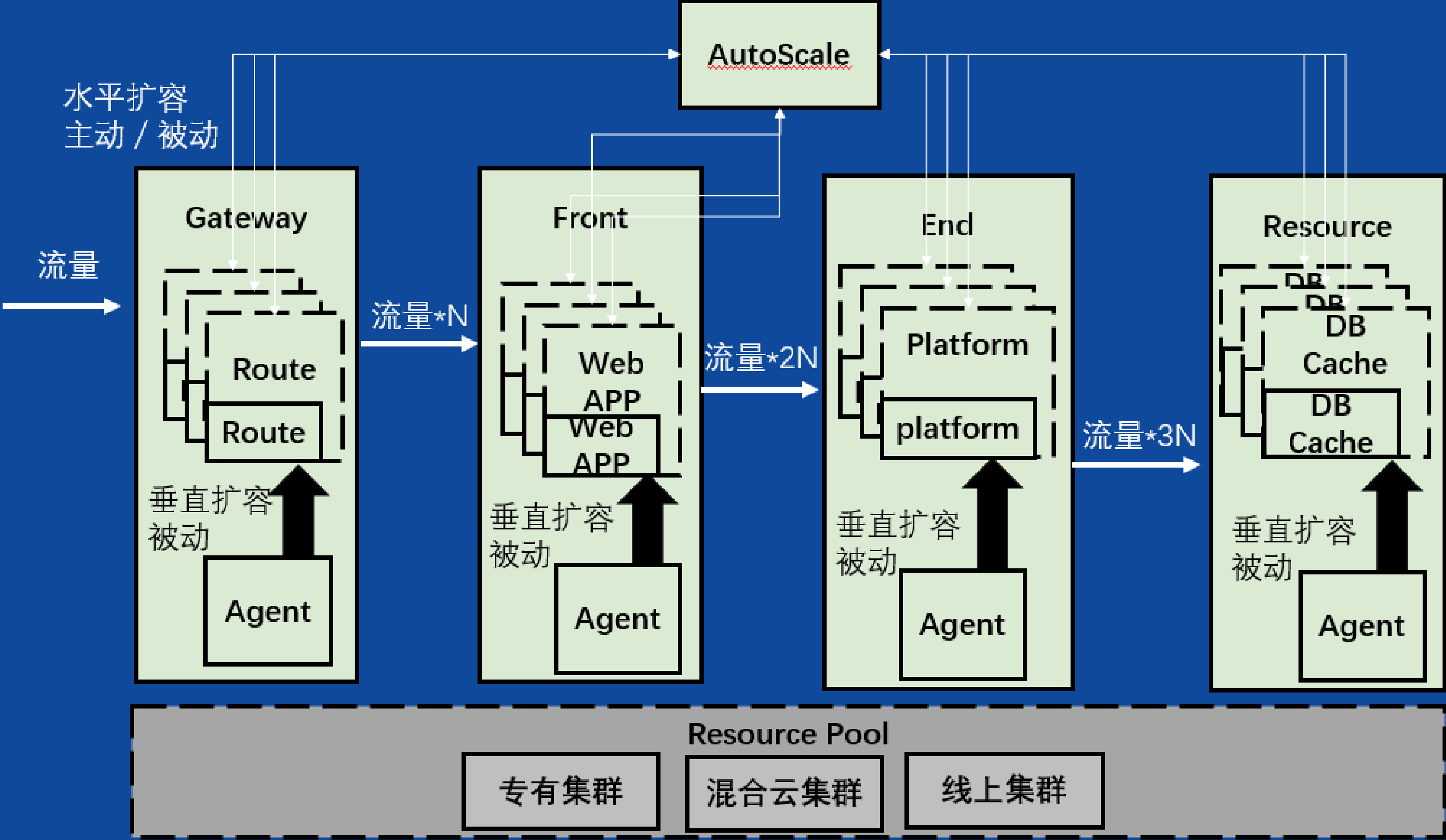


活动流量：活动临时叠加的峰值流量

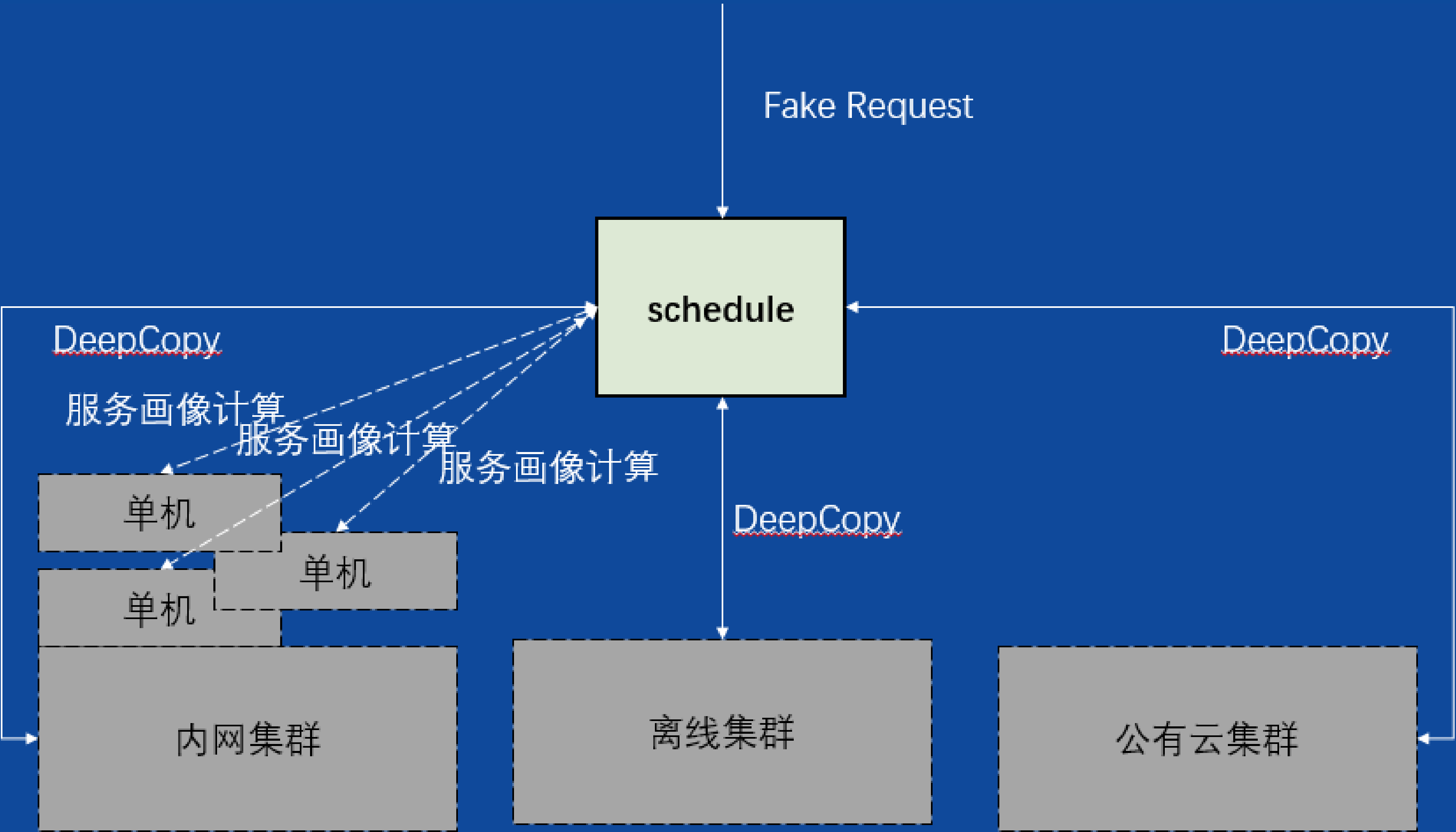


日均流量：
随着业务增长导致的
服务流量快速增长

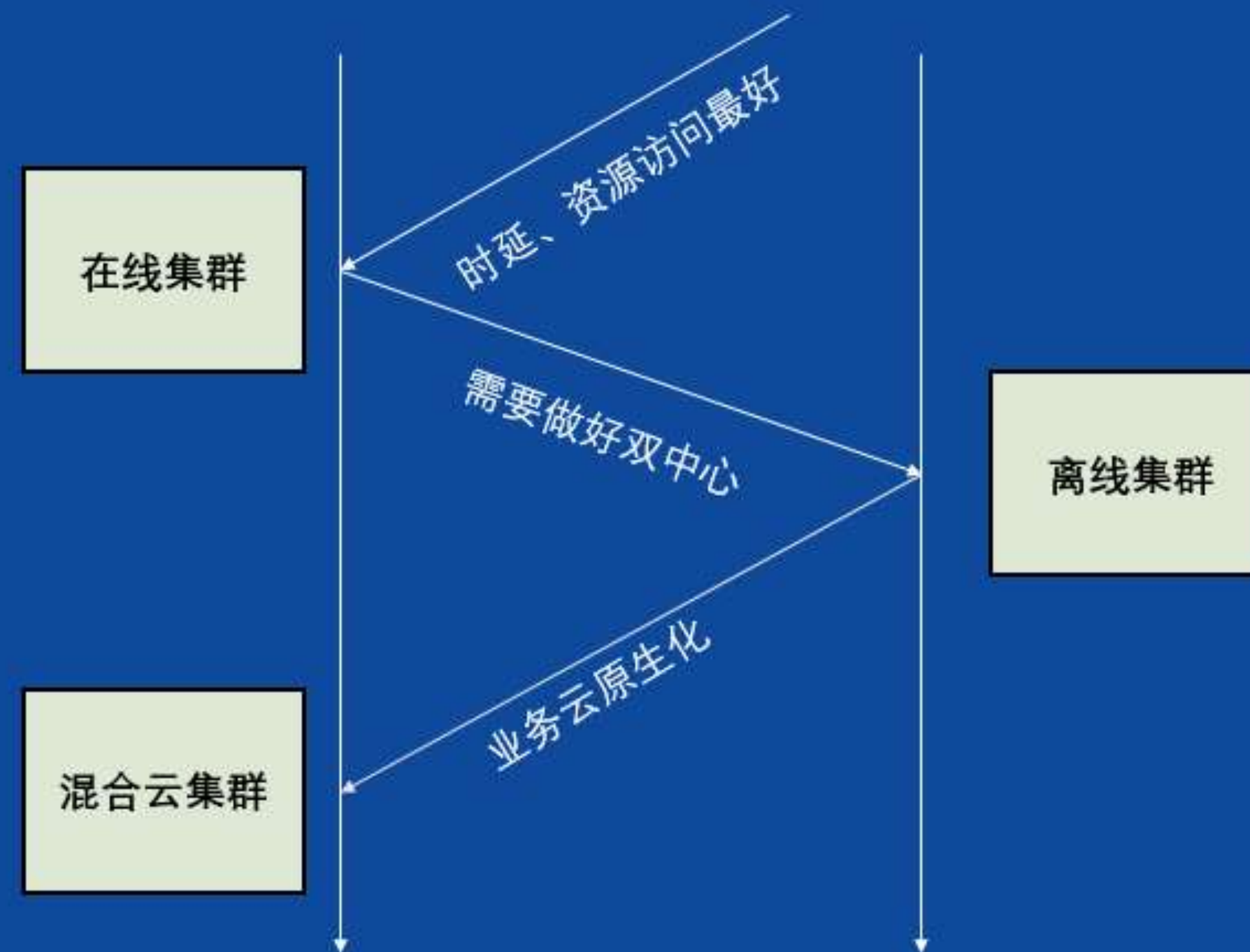
如何在峰值流量下保障服务稳定性



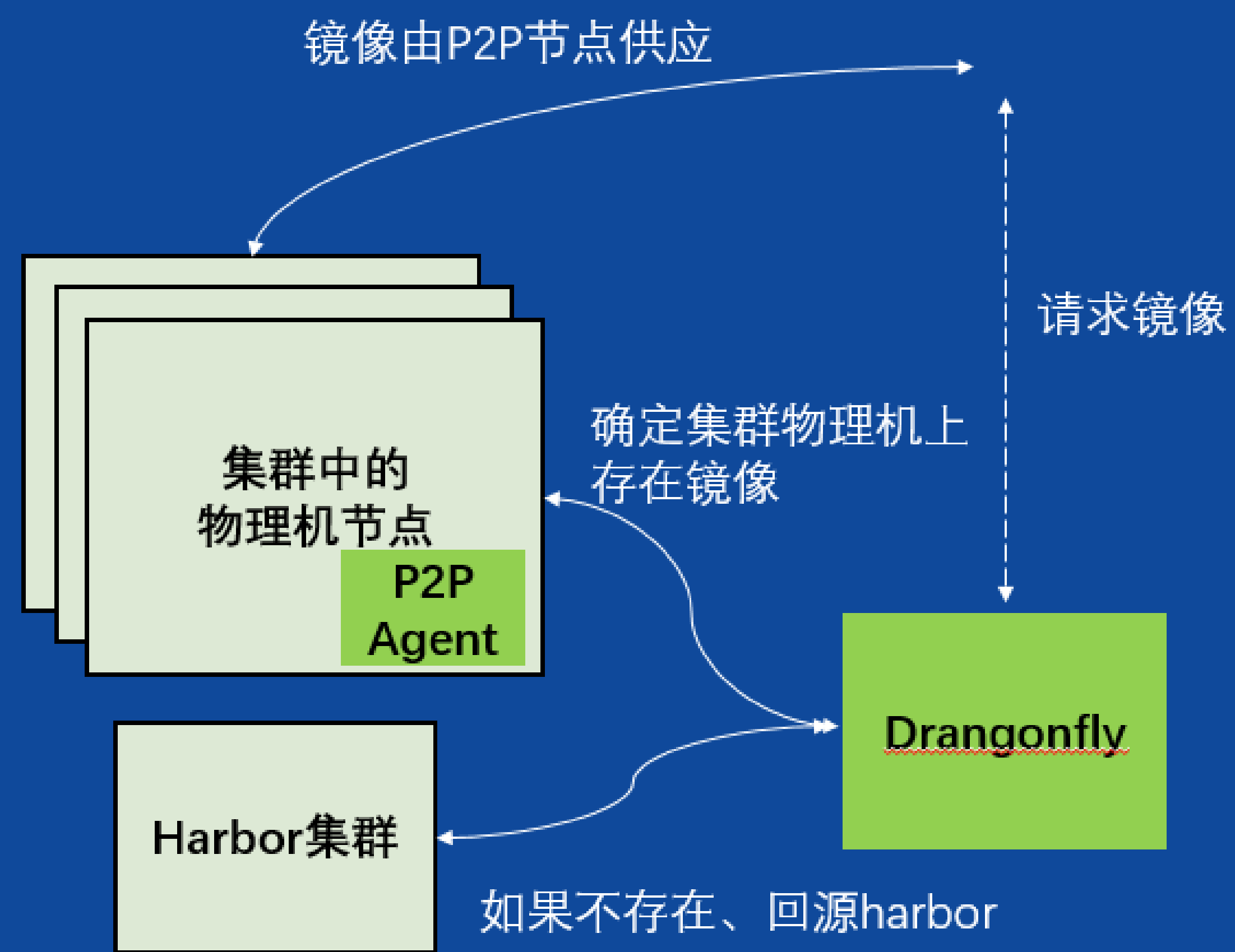
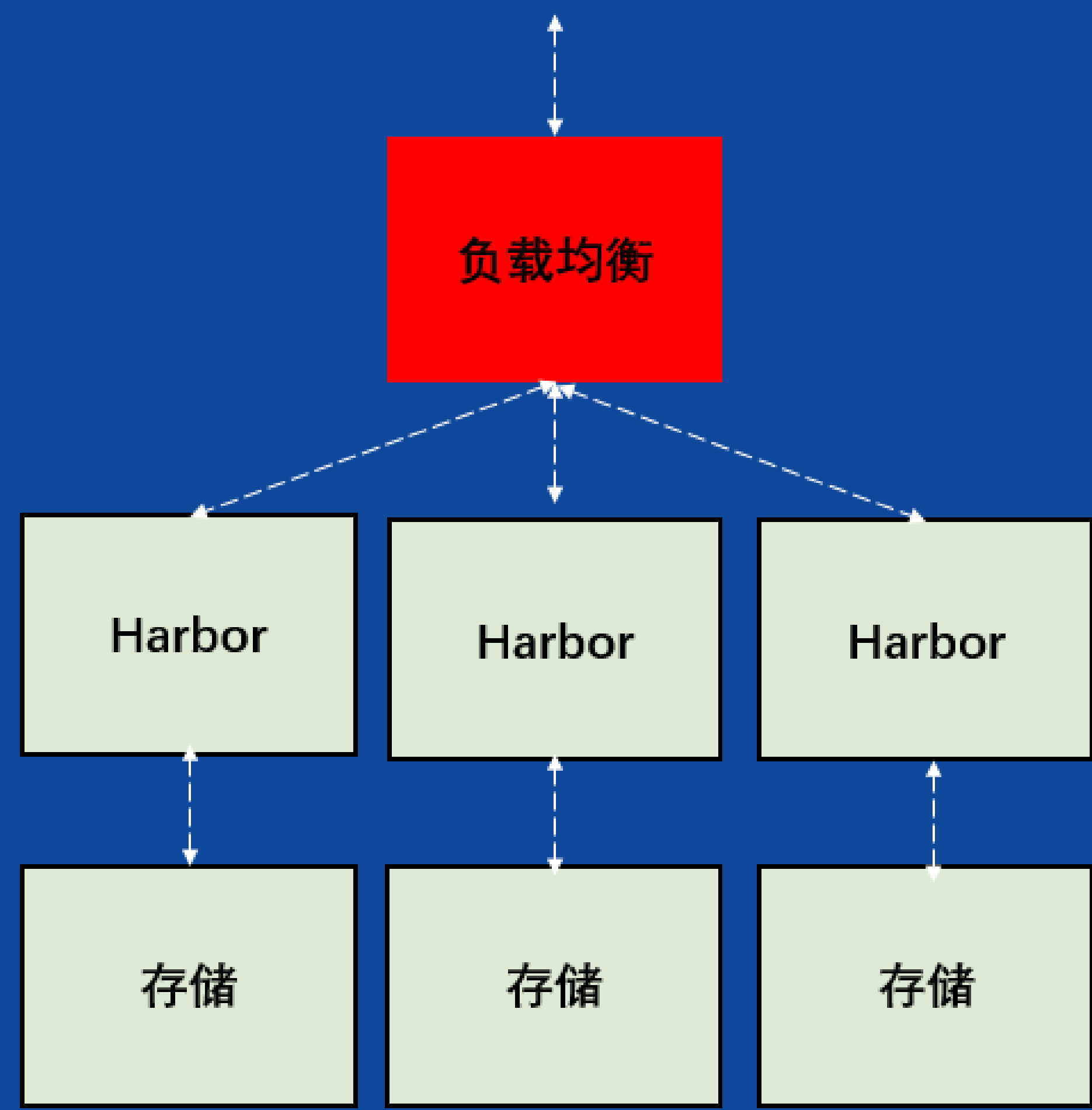
混合云架构下的库存管理



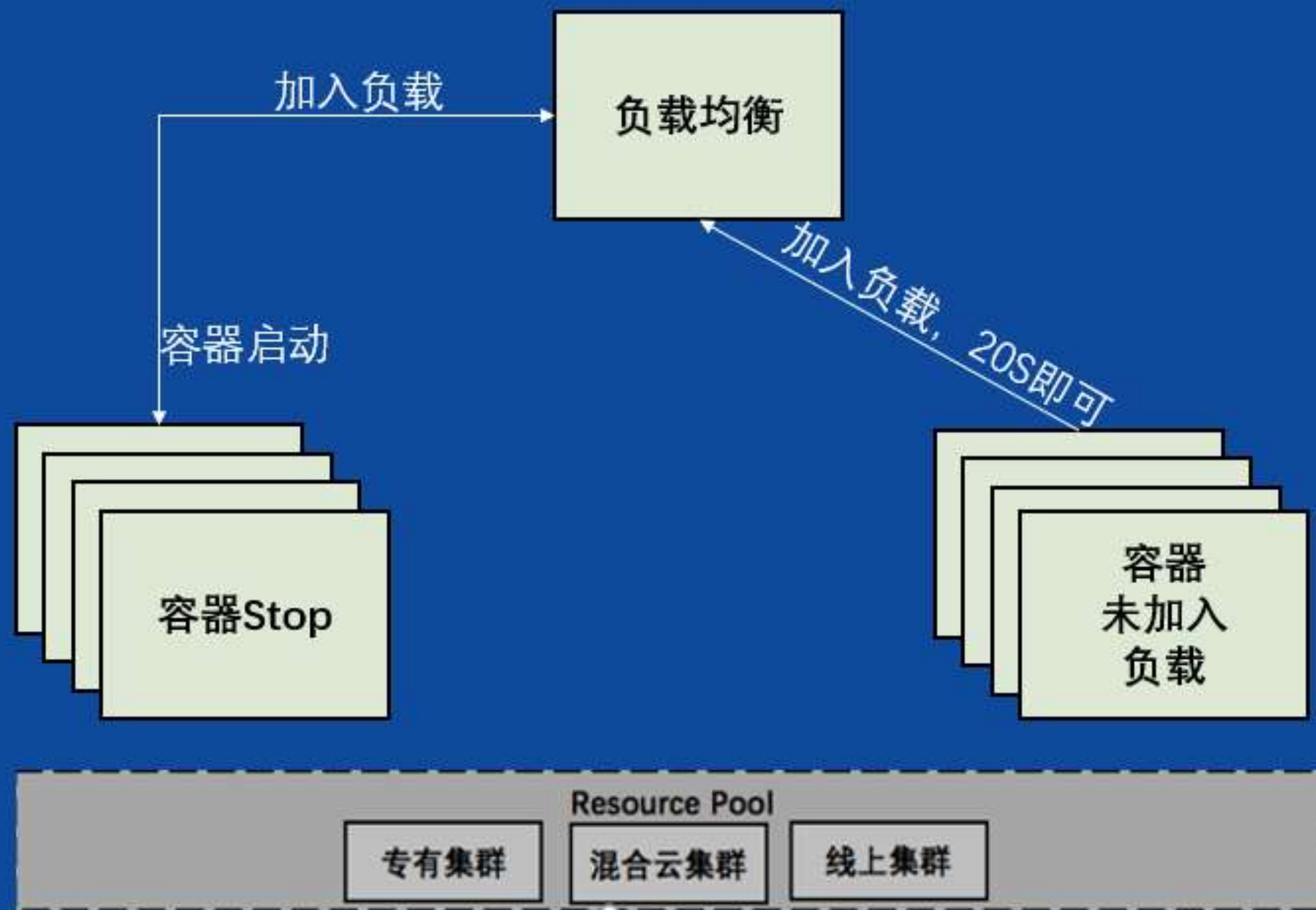
混合云架构下的扩容顺序管理



大规模扩容带来的压力

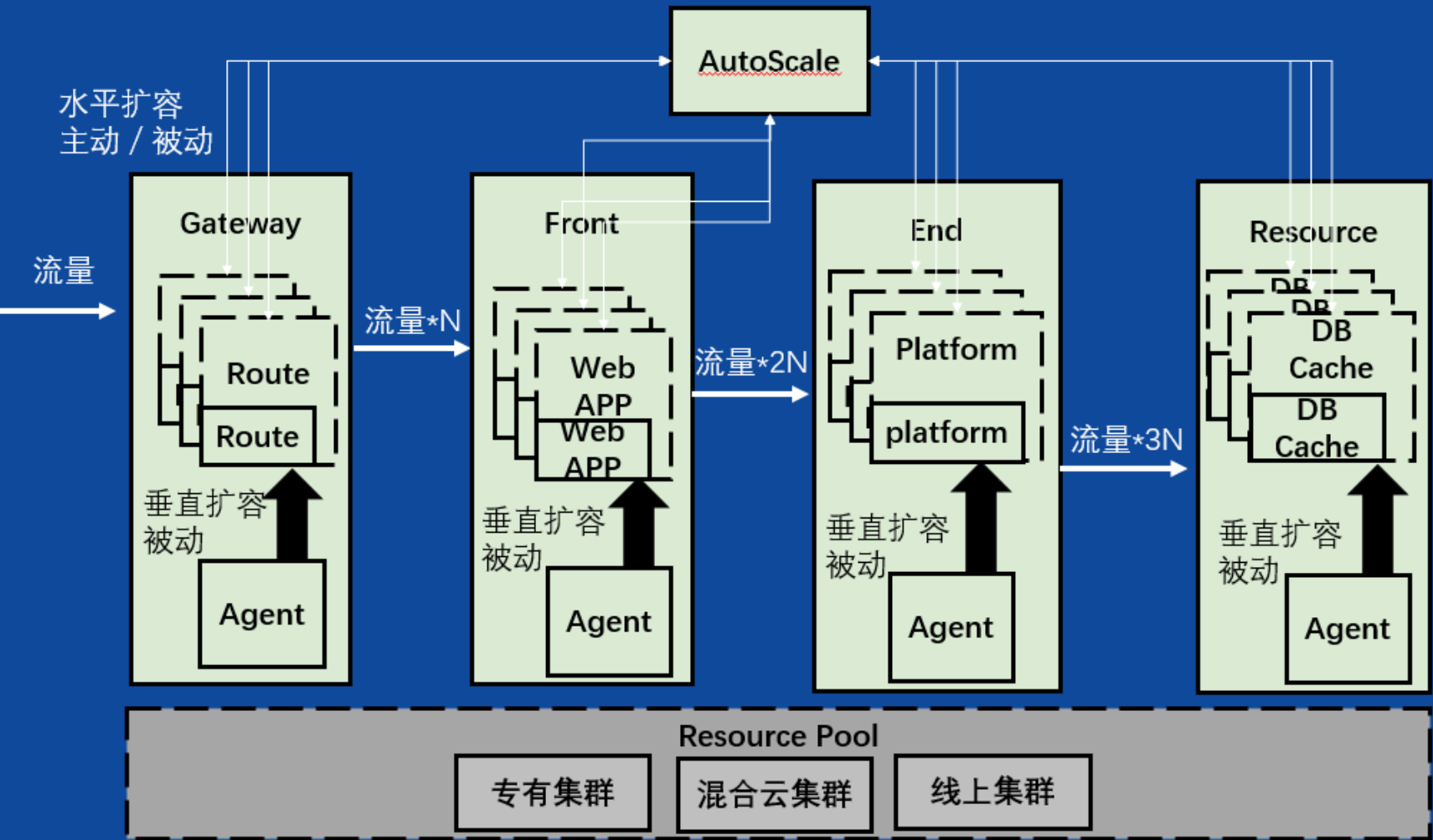


弹性扩缩容能够做到多快？



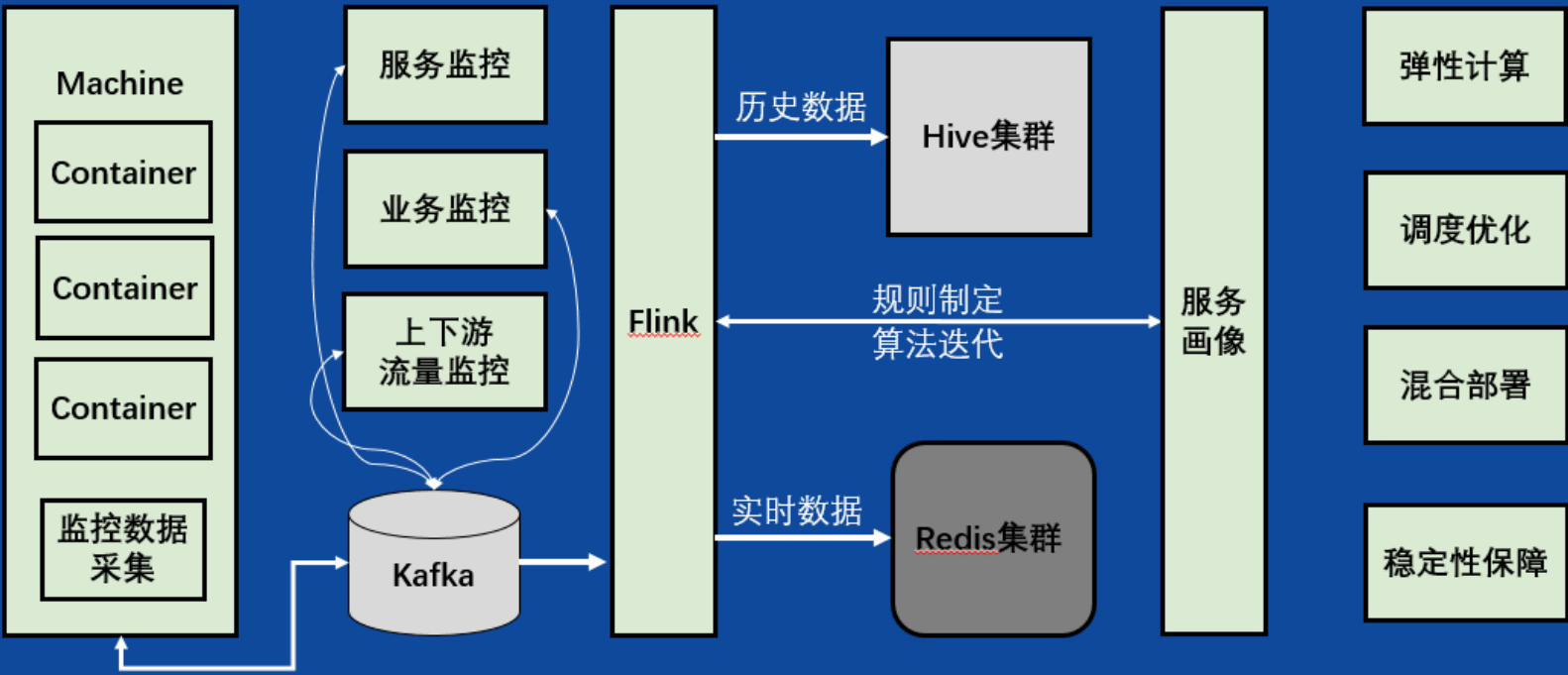
弹性计算如何节省成本

降低采购成本



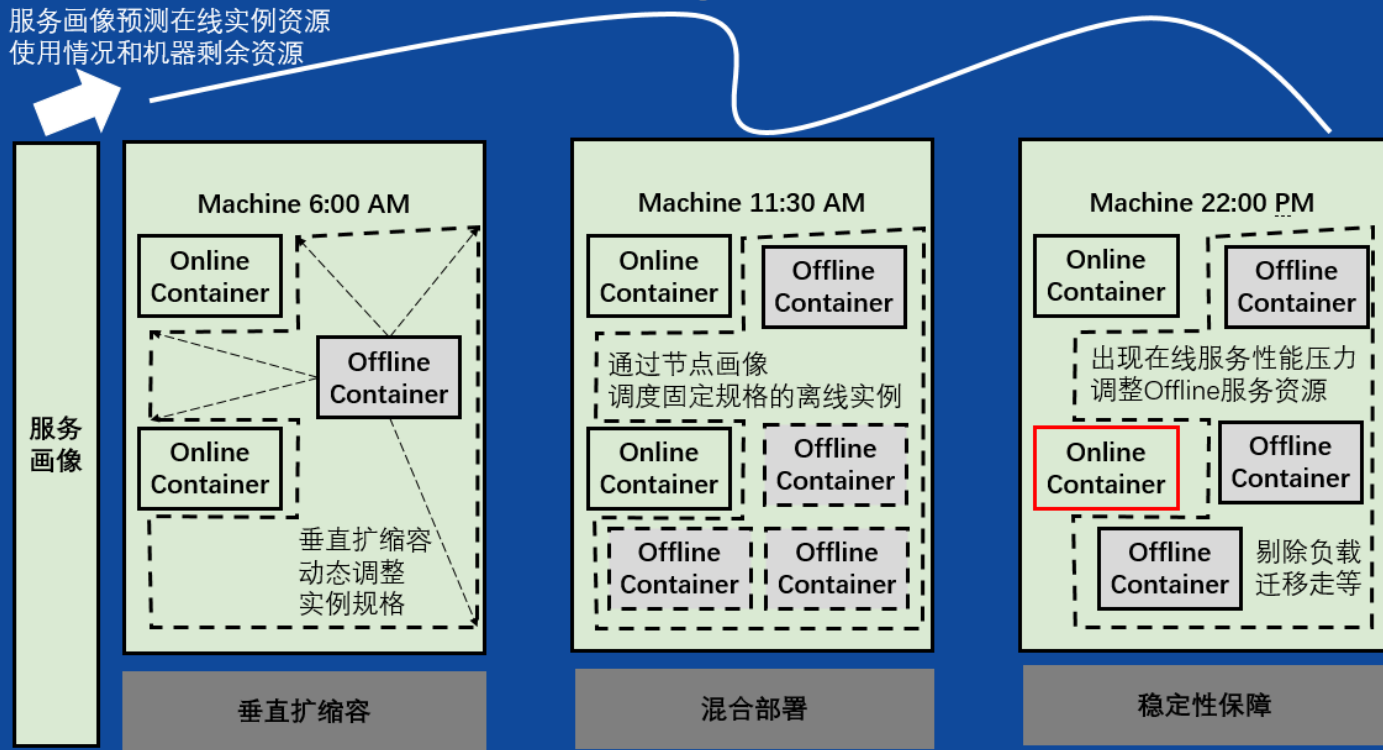
减少应对峰值流量所需的机器采购节省成本

合理资源应对流量



缩容线上服务实例，减少无用资源的浪费

盘活资源利用率



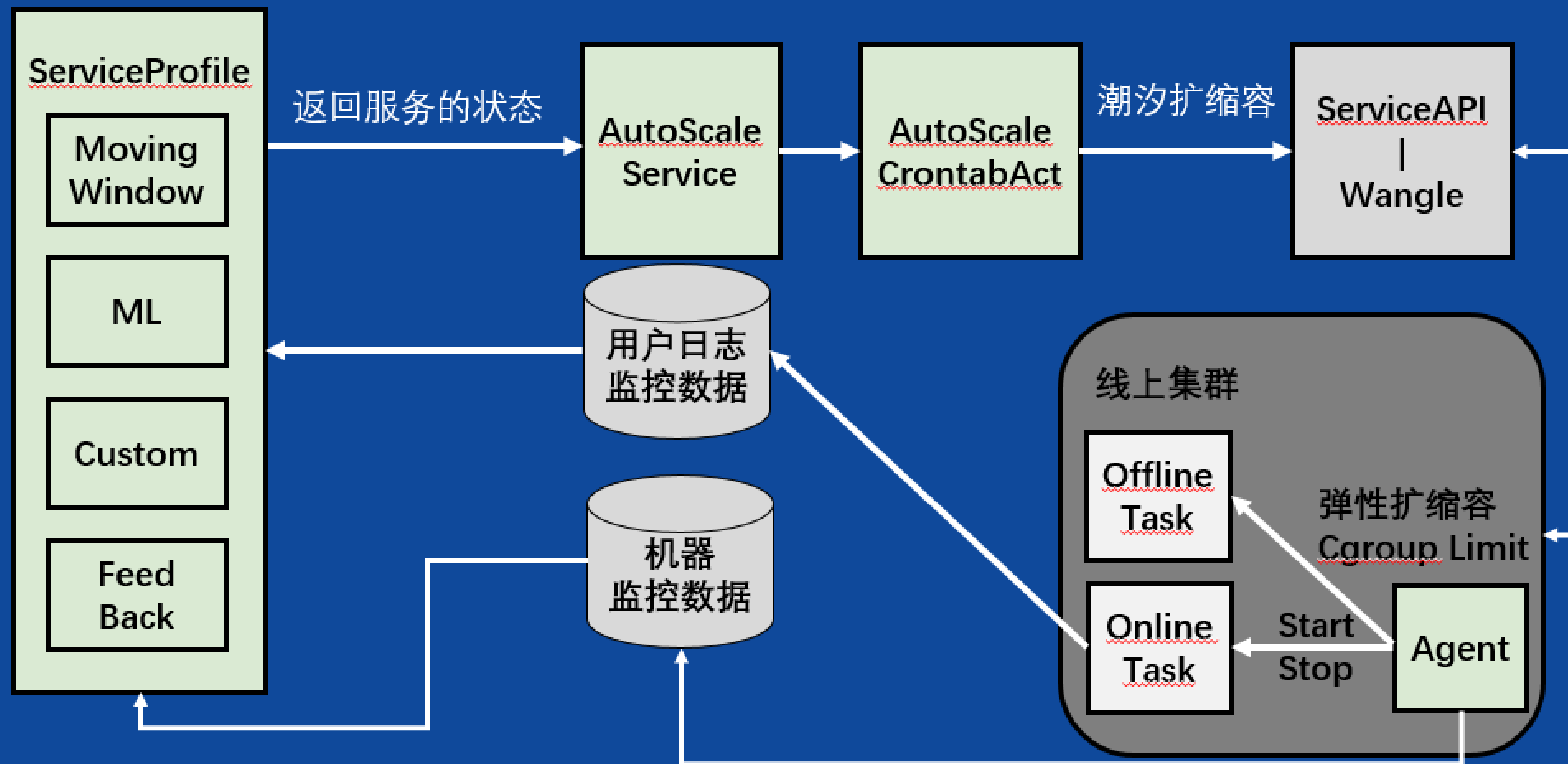
进行离在线混合部署

资源利用率目前的卡点



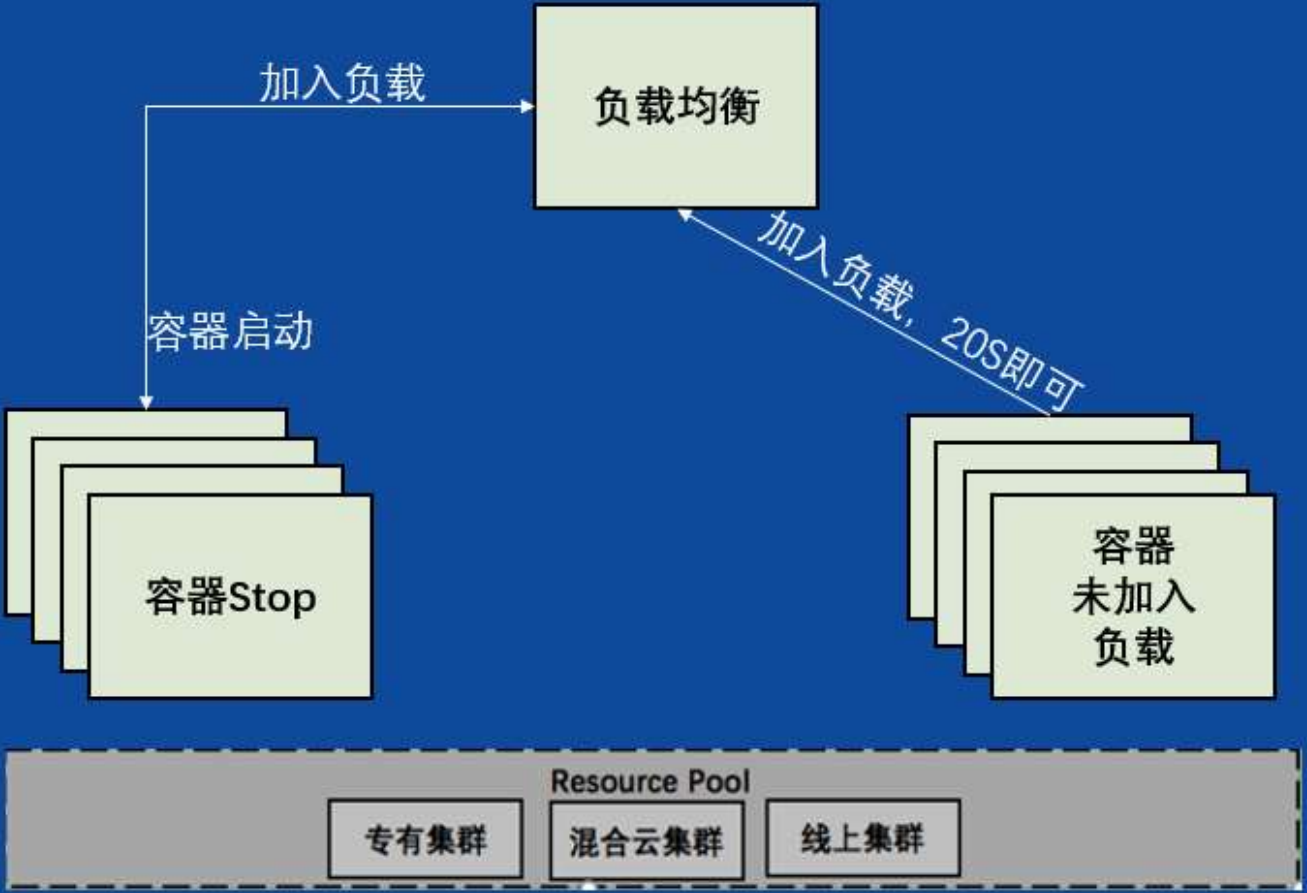
流量呈现波峰波谷的形式，但是承载流量的在线实例数量一直不变

进入到资源使用优化深水区-ScaleZero



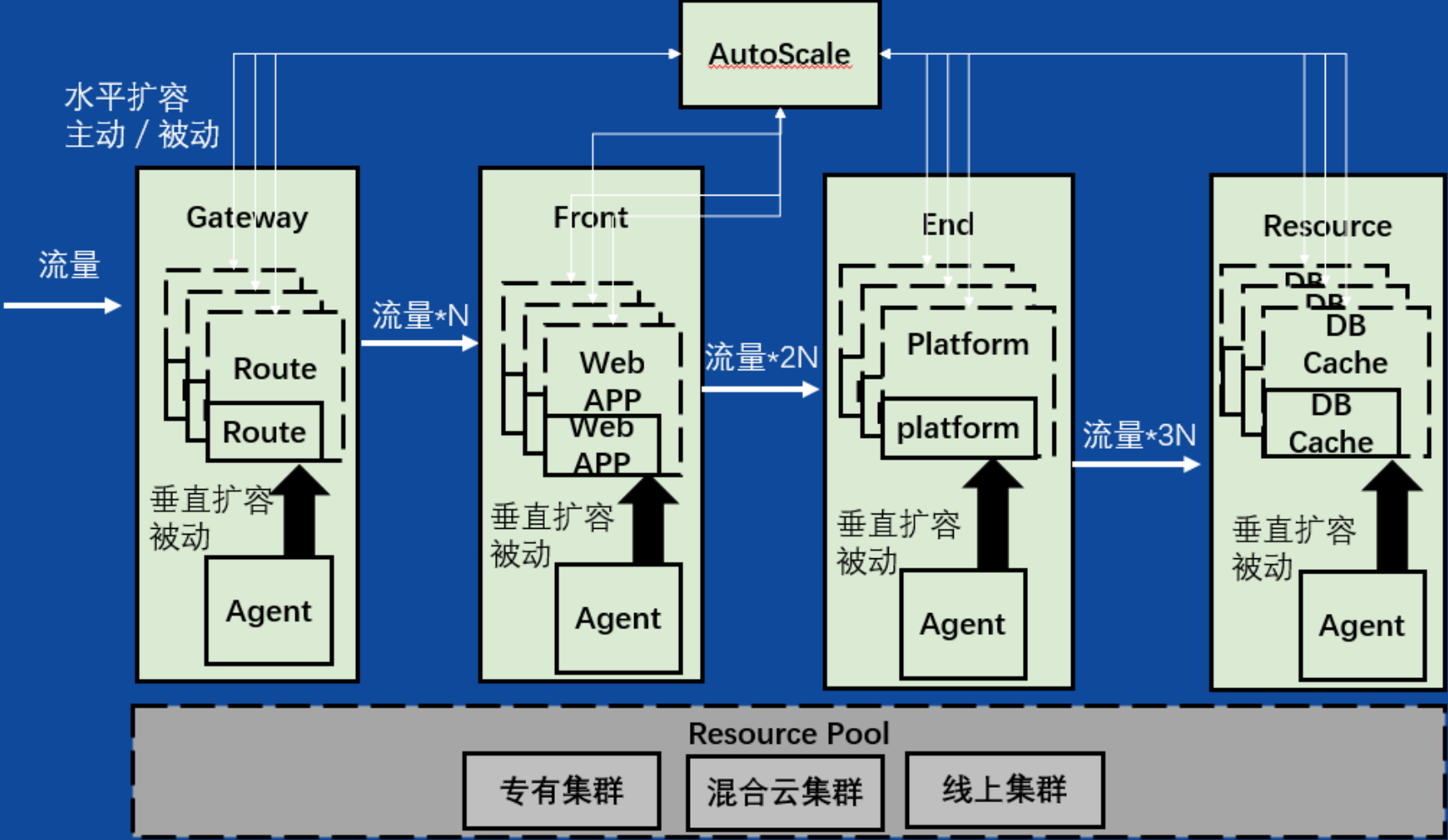
ScaleZero的实现考量

规避慢启动



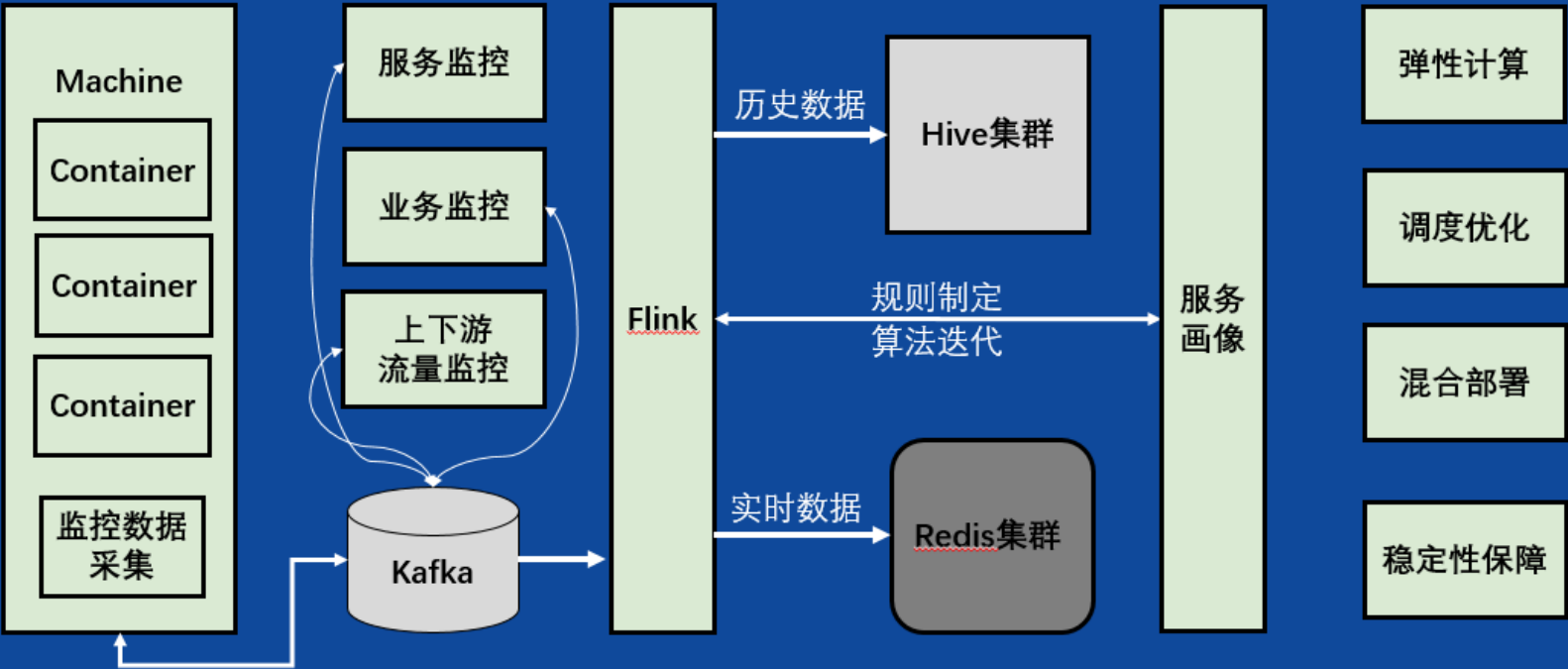
Scale逻辑上将在
在线实例“缩容”
实际上只是
剔除负载或者Stop掉

流量扛得住



搭配
垂直扩缩容、水平多集
群扩容、链式扩容
解决

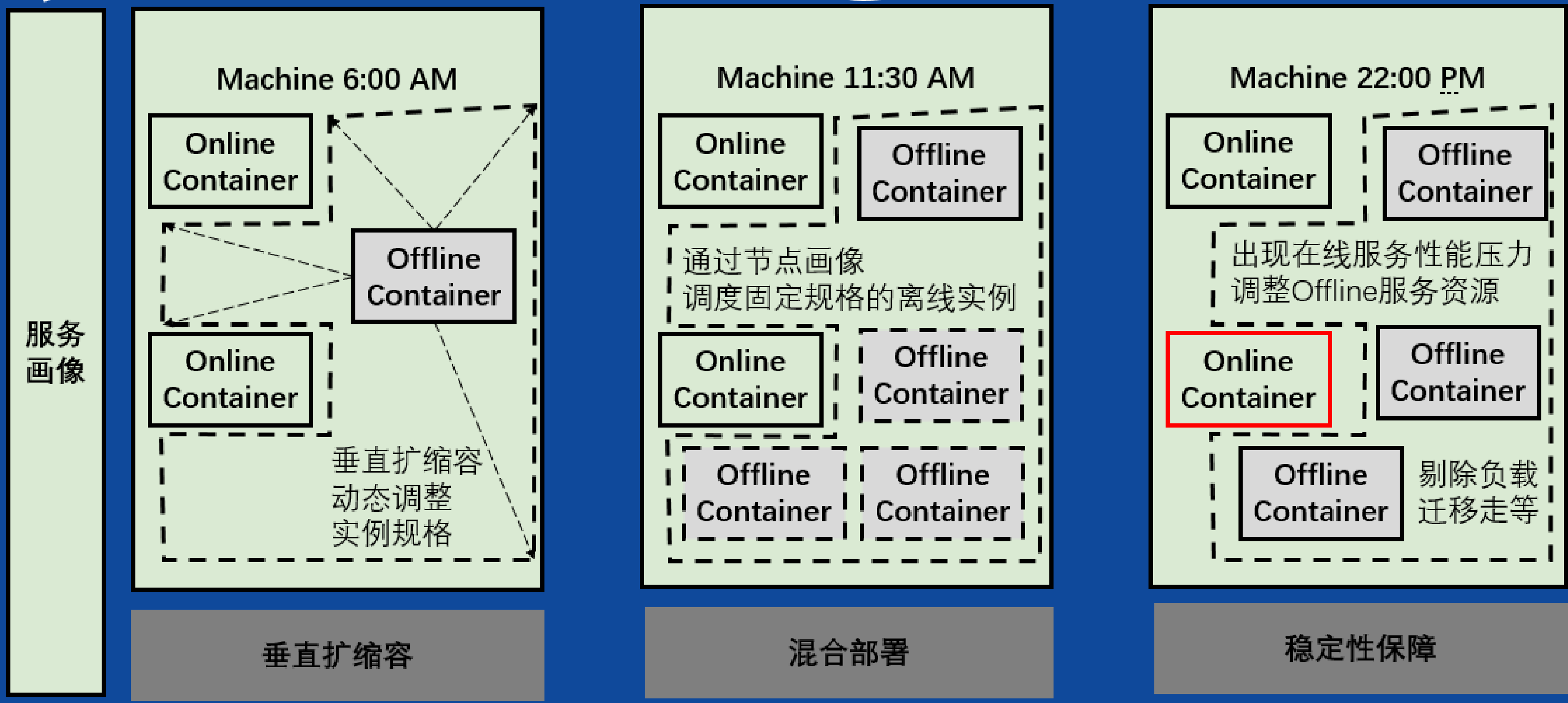
服务稳得很



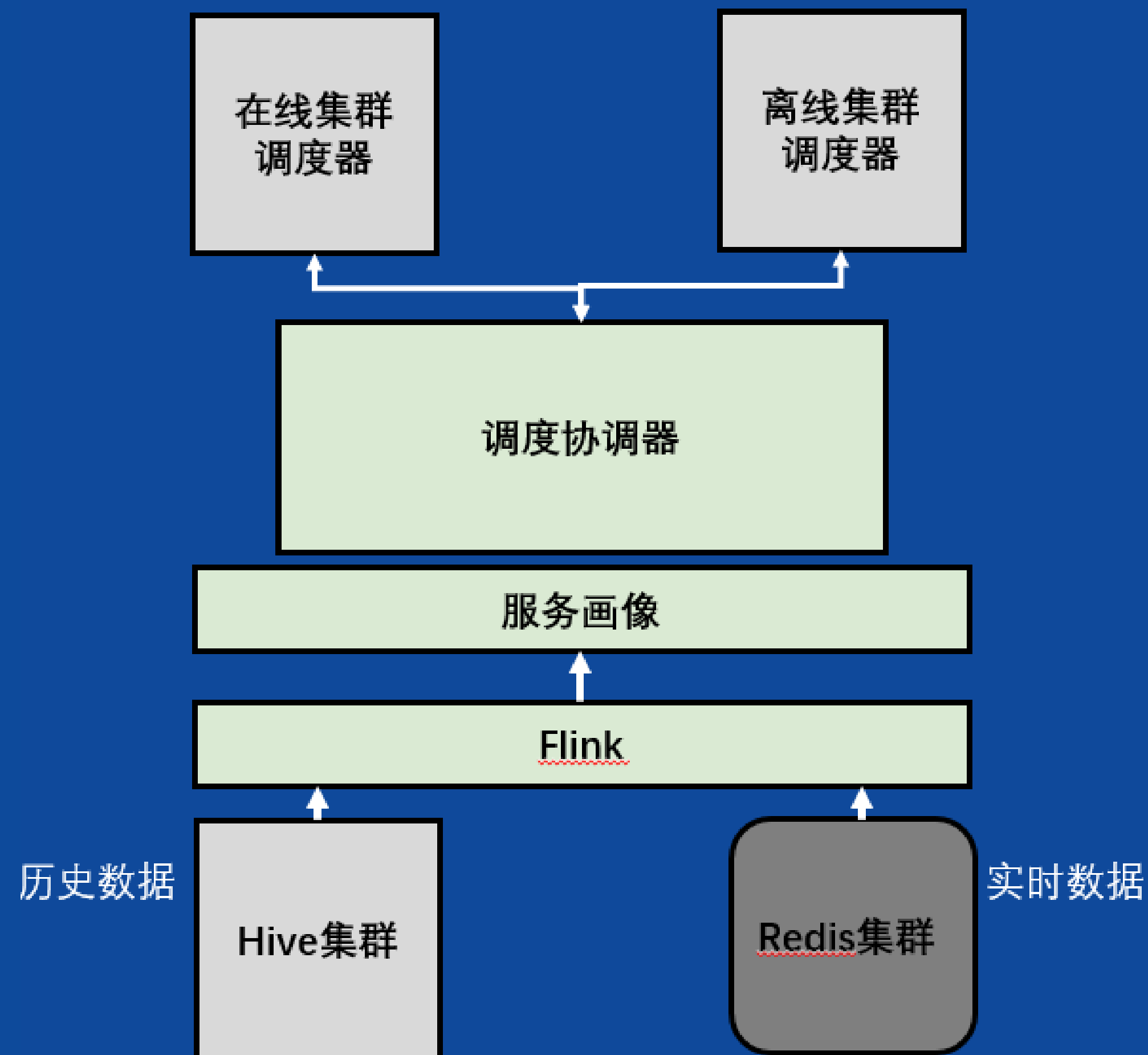
建立以业务SLO为基础的
的服务画像体系，对于
服务的承载能力做准确
评估

进一步节省成本-实时在离线混合部署

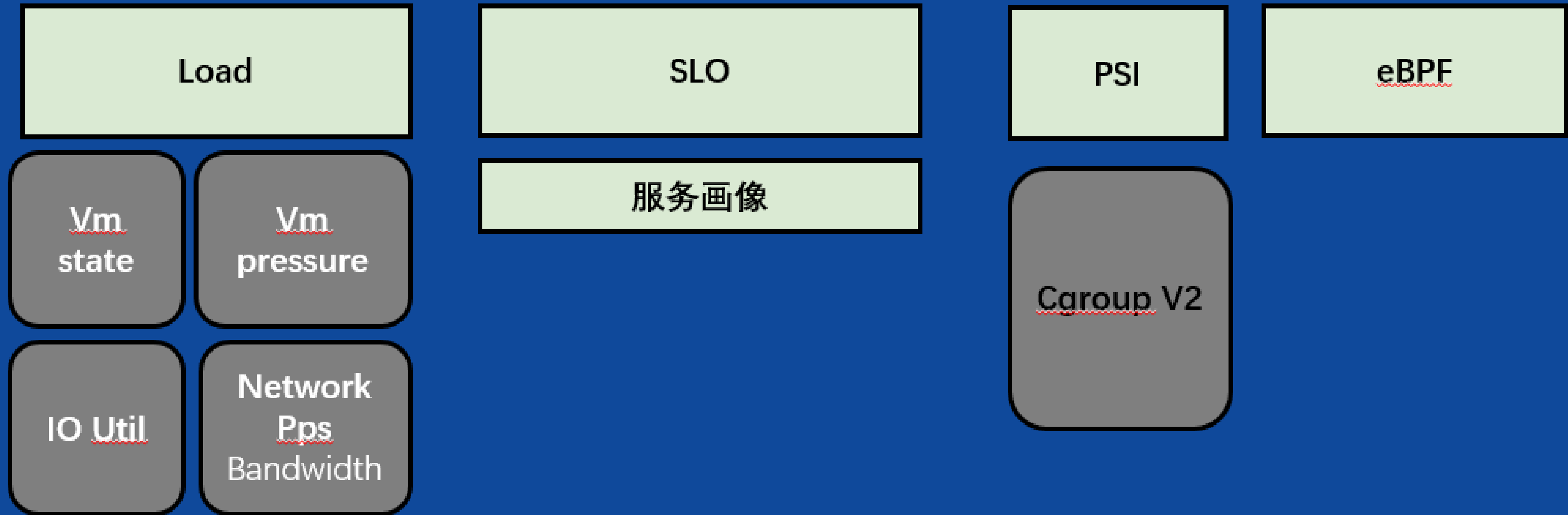
服务画像预测在线实例资源
使用情况和机器剩余资源



实时离在线混合部署的调度架构设计



实时离在线混合压力感知处理

















传统的Load、IOUtil
等用于物理机稳定性
保障的保底驱逐依据

基于业务的服务
画像结合实时数
据判断服务压力

结合CgroupV2的PSI,
新型的eBPF的hook
探索获取更详细的压力信息

集群规模庞大-服务稳定性问题开始凸显

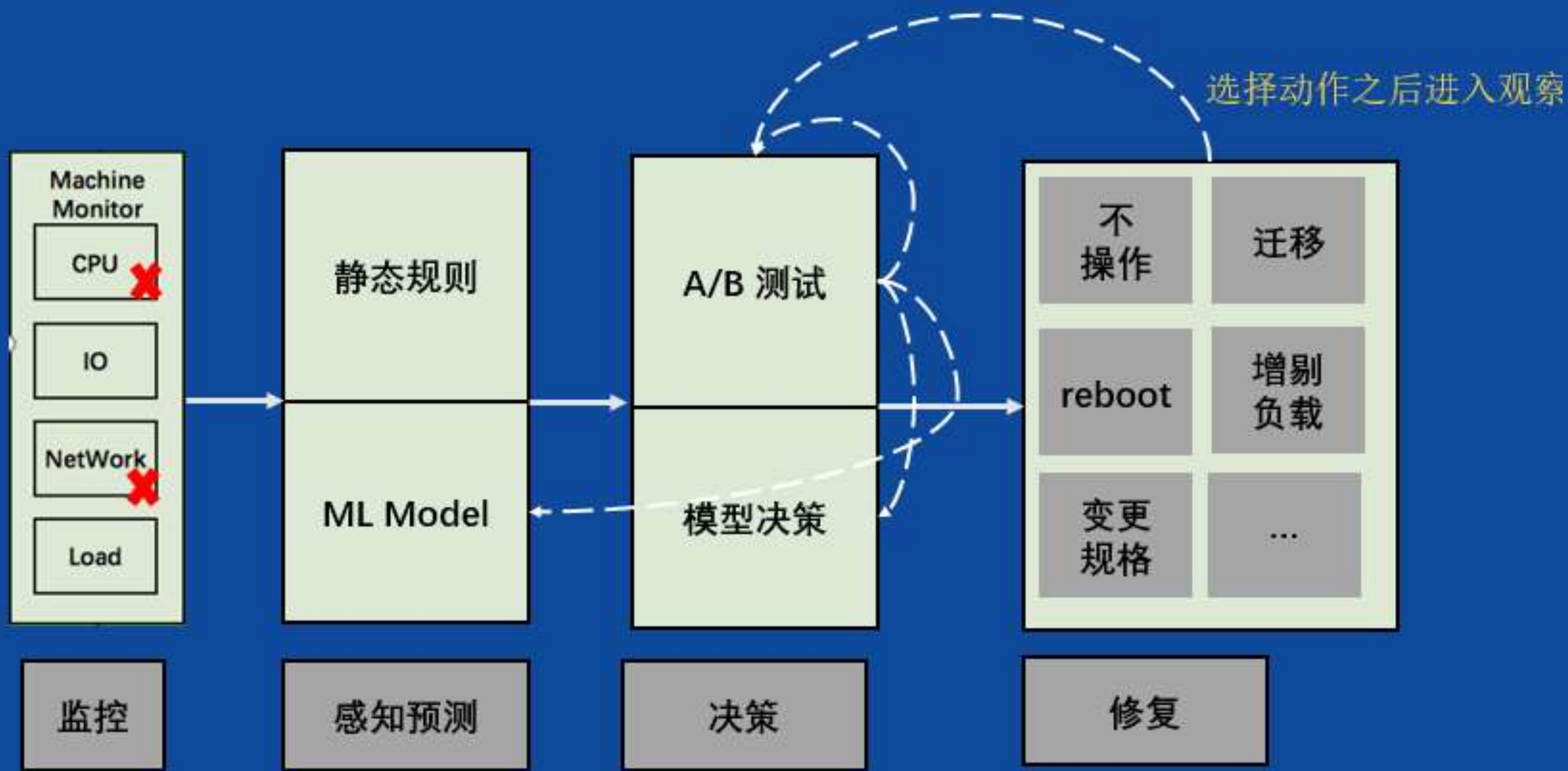
名称	主机名	IP	指标	策略模板	发送	维护
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	load.5min	oce...	0	正...
	cpu.busy	...	0	正...
	net.out.percent/iface=e...	...	0	正...
	net.out.percent/iface=b...	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	cpu.busy	...	0	正...
	mem.memfree.percent	...	0	正...

容器平台的服务稳定性评量标准

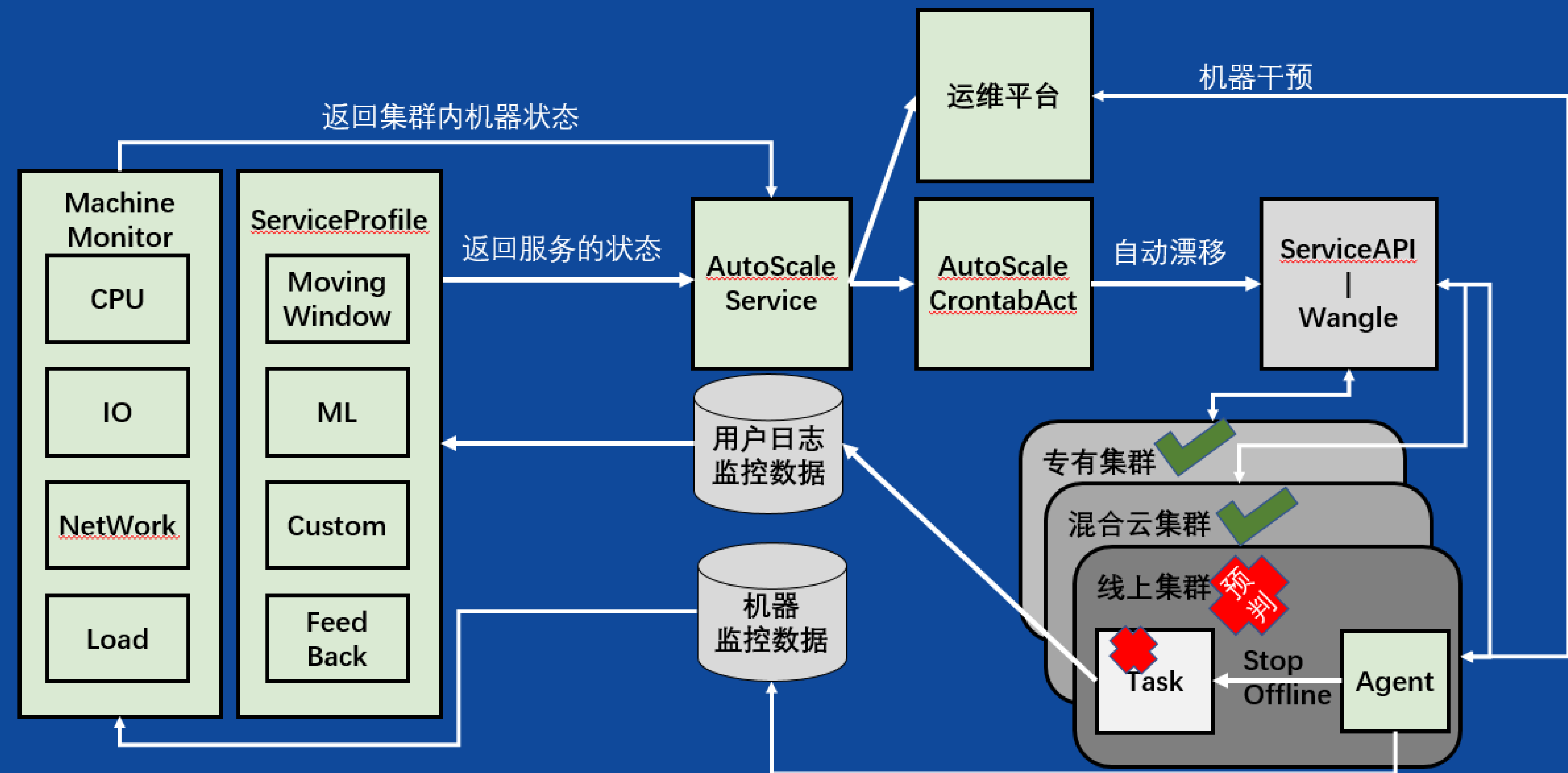


服务中断率：
部署在平台的服务因非代码问题导致的服务中断时长/
原本应该正常服务的时长

不能忽略-稳定性保障



服务自动漂移Workflow



一些展望与总结

eBPF

底层资源
细粒度管理

干扰
避让

Cgroup
V2

存算
分离

容器
运行时

有状态容器
弹性扩缩容

预测算法

THANKS

—
Global
Architect Summit

