基于大规模弹性伸缩实现拓扑感知的在离线并池

邵伟 / 字节跳动 高级研发工程师

主讲人

邵 伟

字节跳动 编排调度团队 高级研发工程师

主要负责大规模弹性系统、在离线混合部署等资源优化相关工作

大纲

1 背景介绍

2 在离线弹性体系

3 案例分析

4 未来展望

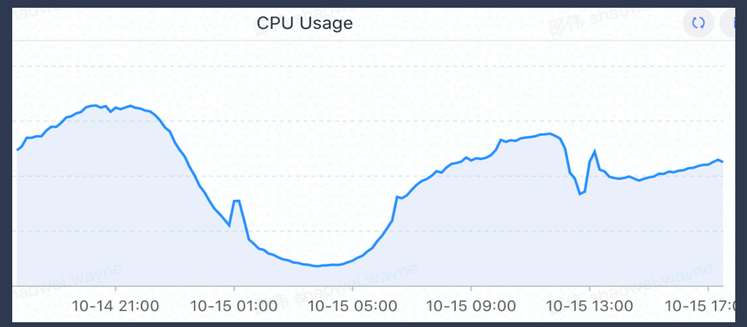
字节云原生化现状：如何实现资源效率提升?

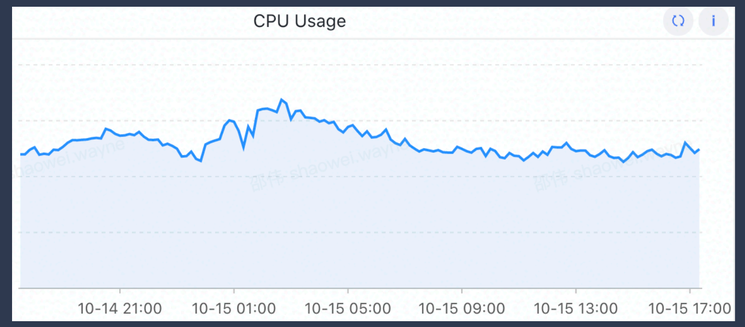
多元化业务体系

- 资源供应优先保证在线稳定
- 在线潮汐明显，离线资源缺乏

在离线并池实现利用率提升

Web 服务和算法服务占据大部分在线资源





资源效率提升场景之一

在线 Web 服务和离线批式作业：业务模型简单，资源模型互补

- Golang 服务
- 重延时，CPU 调度性能关键
- 其他资源需求较少

- Ad-hoc 查询、报表
- 运行时间短，重内存
- 重吞吐，资源可挤压
- 实例可少量杀死

- 多维度资源隔离
- 中心+节点两级管控

资源效率提升场景之二

在线算法服务和离线训练作业：业务模型复杂，资源模型同质

- 推荐、广告、搜索核心服务
- 延迟、效果并重
- 加载模型，重内存
- NUMA 绑定，异构设备支持

- 推荐广告 CTR/CVR、NLP
- 吞吐、效果并重
- 运行时间长，多角色协作
- NUMA 绑定，异构设备支持

- 分时资源复用
- 适配性能需求
- 适配离线编排能力

弹性并池挑战

如何弹

基础弹性能力

- 在线：无状态，可水平扩展
- 离线：协同框架，定制弹性能力

如何用

资源协同感知

- 扩展资源表达和管控
- 在离线统一调度
- 跨集群的资源整合

如何稳

稳定性保证

- 弹性资源分配和回收

大纲

1 背景介绍

2 在离线弹性体系

3 案例分析

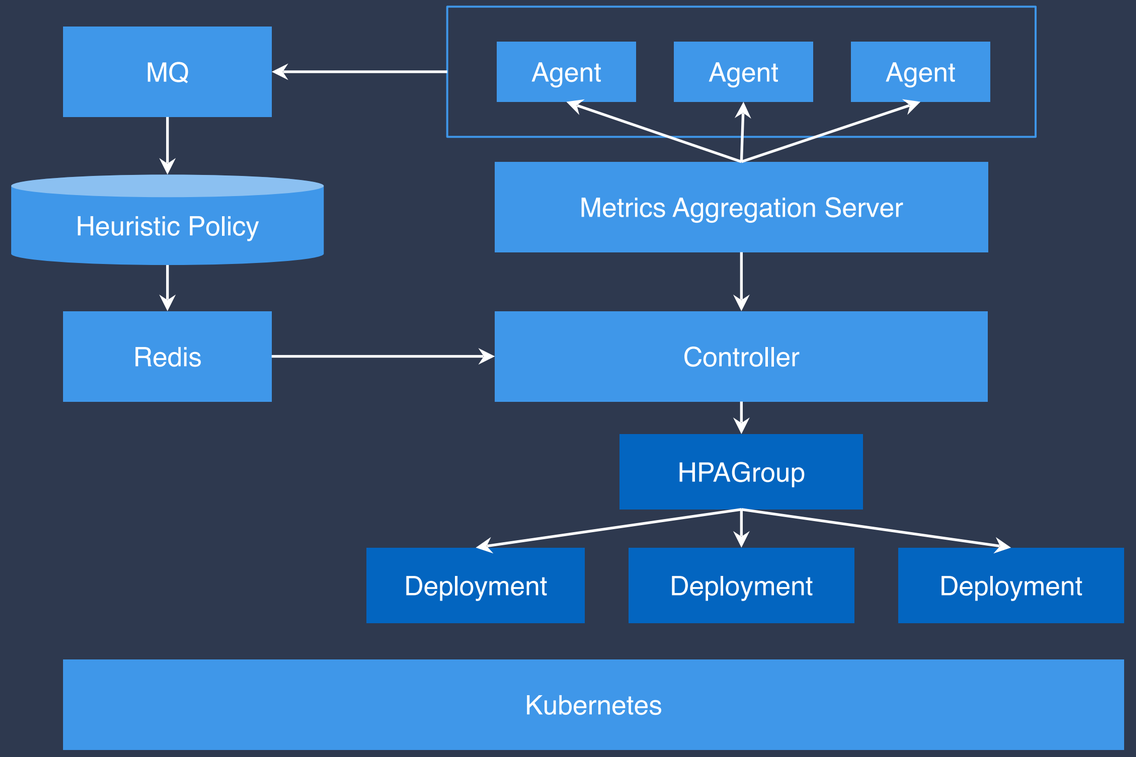
4 未来展望

在线弹性设计：如何保证资源及时归还?

在线服务天然支持水平扩展，关键挑战在于构建**快**、**稳**的弹性系统

在线弹性分层架构

- 离线模型
- 实时数据
- 控制链路
- 调度链路



在线弹性实时性保证

基于历史数据构建资源画像

预测执行扩容行为

自研可扩展内存数据存储

服务建立维度索引，加速查询

实时数据预取，聚合逻辑下发

分片调度 + 乐观并发 bind

镜像 lazy loading 按需拉取

P2P 实现镜像和模型快速分发

非突发流量预先扩容，突发流量分钟级扩容

在线弹性稳定性保证

配额

缩容部分资源仍然占住配额

服务可以随时可回收

数据

系统指标和业务指标协同

完善数据异常 fallback 机制

组件

分布式追踪 Pod 生命周期

完善报警，快速定位组件异常

联邦

集群具备自治能力

联邦提供自动故障切换

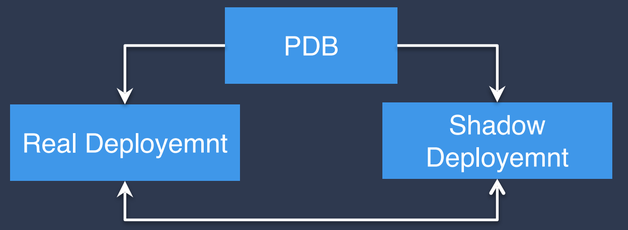
组合

根据调用关系

提供链路组扩缩能力

避免出现服务瓶颈

兜底



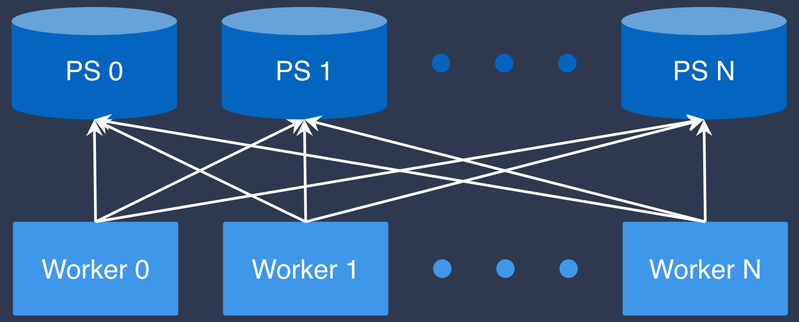
原地进程替换 / 流量摘除

离线分布式训练：PS-Worker 弹性定制

“推广搜” 亿级用户和 item 特征，必须采用 PS-Worker 框架

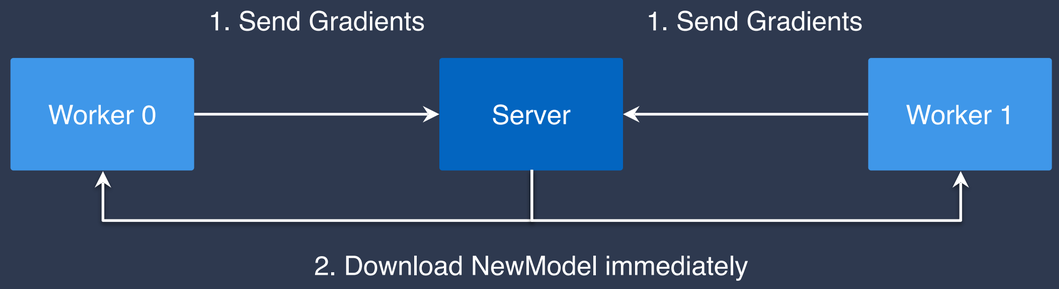
基础介绍

- PS 存储模型参数；Worker 计算更新梯度



负载特性

- PS 实例异常任务失败
- Worker 容忍部分异常；一定程度弹性可加快训练效率
- 性能不均严重导致梯度异常；NUMA 绑定，环境同构



弹性改造

- PS 保证高优资源，灵活 Checkpoint (模型和数据 offset) 和 failover 机制
- Worker 弹性满足 min/max 实例需求

离线分布式训练： Ring AllReduce 弹性定制

CV/NLP 等场景模型参数能单机塞满，使用 Ring AllReduce 实现带宽优化和训练加速

基础介绍

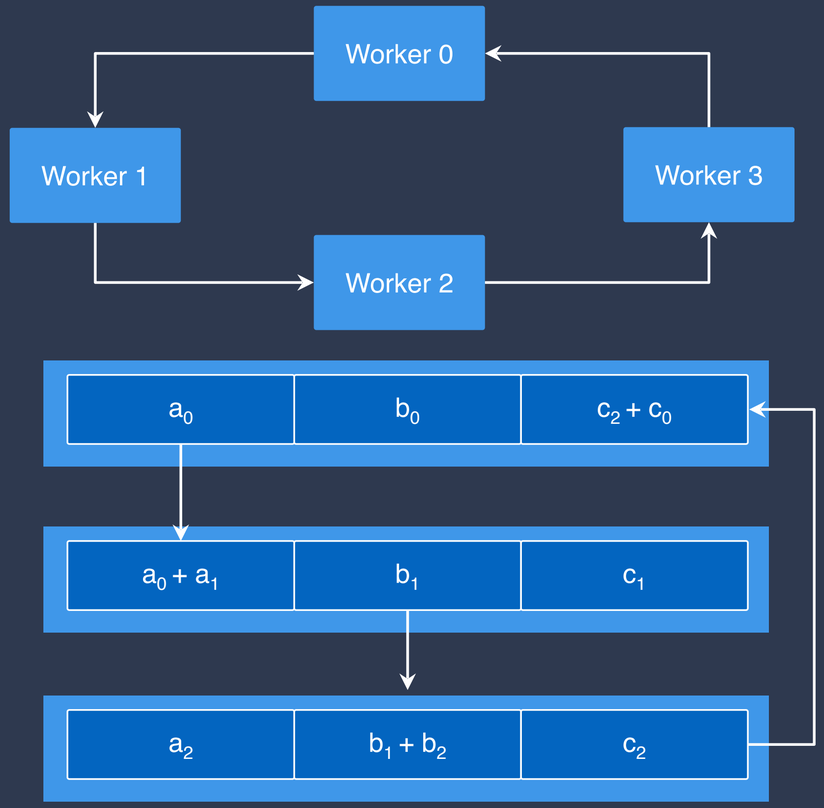
- 无中心节点，Worker 有序，互相交换梯度，2N 次收敛

负载特性

- Worker 天然支持故障容忍和弹性
- 计算提速和 Worker 数线性相关
- 木桶效应；网络带宽大；算力要求高

弹性设计

- 服务发现：维护全局 IP，动态建环
- 状态同步：Worker 维护状态机，发起同步
- 故障恢复：异常 Worker 拉黑，异常梯度恢复
- 数据分割：Worker 数据集正交，扩缩容后重新分片



在离线资源协同感知：单集群统一调度

在离线并池的前提是构建统一的单集群调度和管控体系

自定义 CRD 扩展资源表达

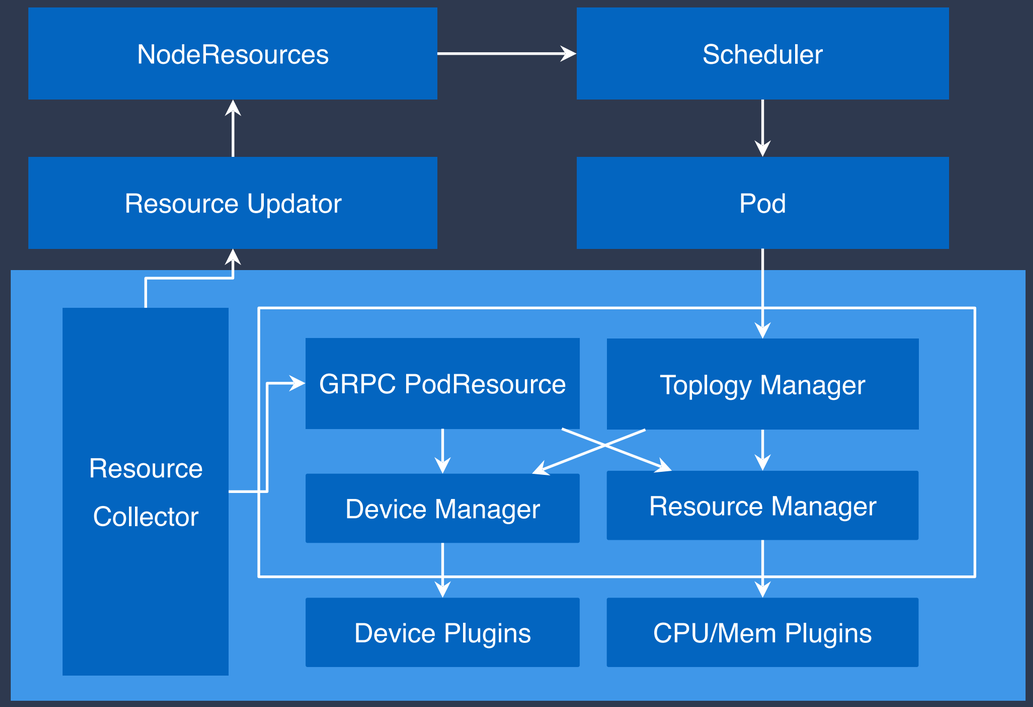
- 在离线均以 Kubernetes 为基座

微拓扑感知调度

- 支持 Gang/Same 等丰富调度语义
- 基于 Scheduler Framework 扩展
  - 自研 predicate 选择符合微拓扑的节点
  - 自研 priority 堆叠 NUMA 资源分配

微拓扑感知管控

- 扩展 Resource Manager
  - 自研 plugin 实现基于角色的分配策略



在离线资源协同感知： 跨集群资源整合

离线作业独立集群部署，需要跨集群感知到在线资源

基于 virtual kubelet 实现跨集群资源整合

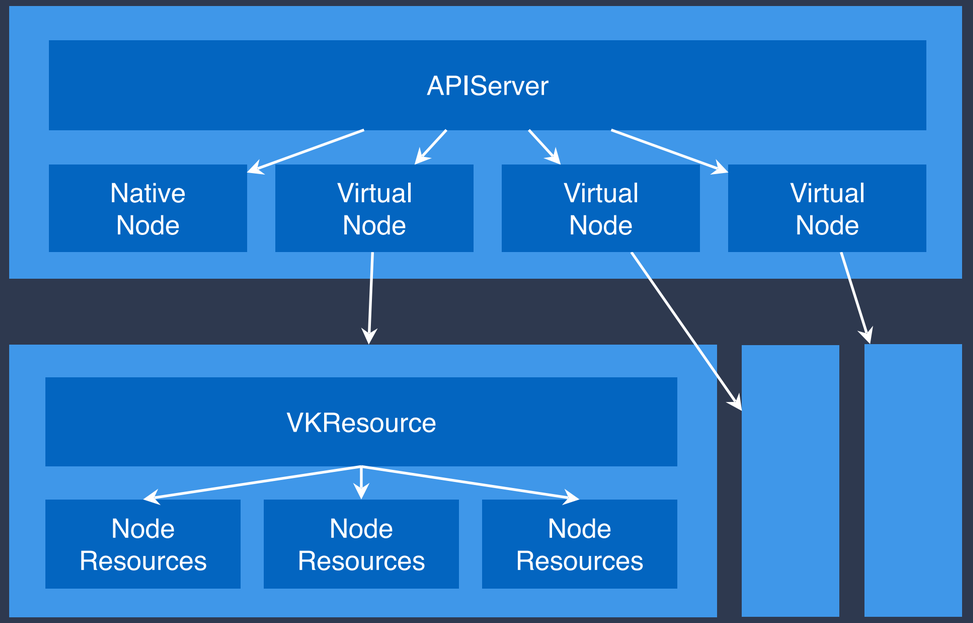
- 提供租户独占集群，方便自定义编排逻辑
- 提供 K8S Native 资源接入体验

两种用法

- native 资源兜底，弹性资源加速
- 弹性资源一站式供应

新的挑战

- 资源汇总，丧失微拓扑语义，跨集群 Gang/Same 调度困难
- 相同 Job 的 Pod 只会同步到相同后端
  - Gang 在前端保证，Same 和微拓扑在后端保证



弹性资源供应带来使用稳定性问题

在资源弹性供应的前提下，实现其合理的分配和回收

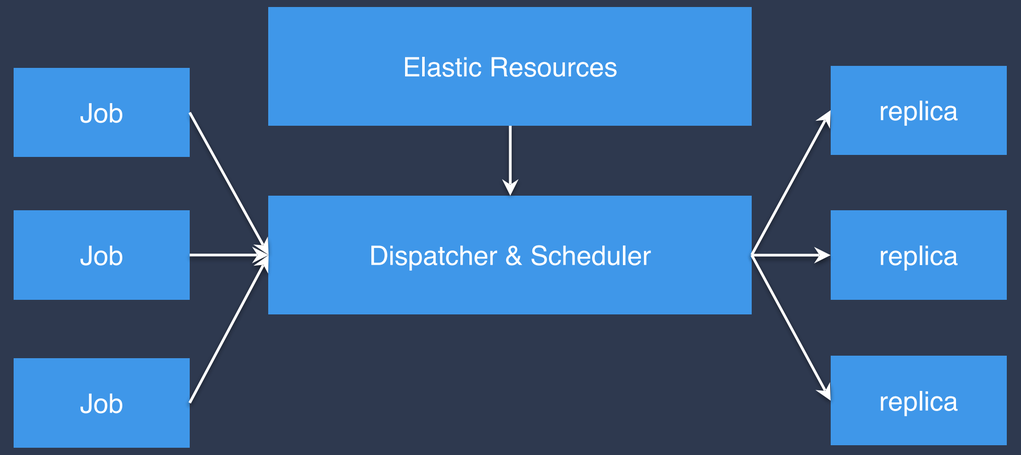
☐ 资源供应不稳定

- 资源总量、资源环境、供应时间无法精确预知
- 在线抖动随时产生资源回收

☐ 单个作业内支持 min/max 语义

- 资源描述并非确定值
- 训练速度和满足更多作业间寻找平衡

☐ 多个作业间可能出现资源死锁



基于 K8s Priority 体系适配资源弹性

资源供应

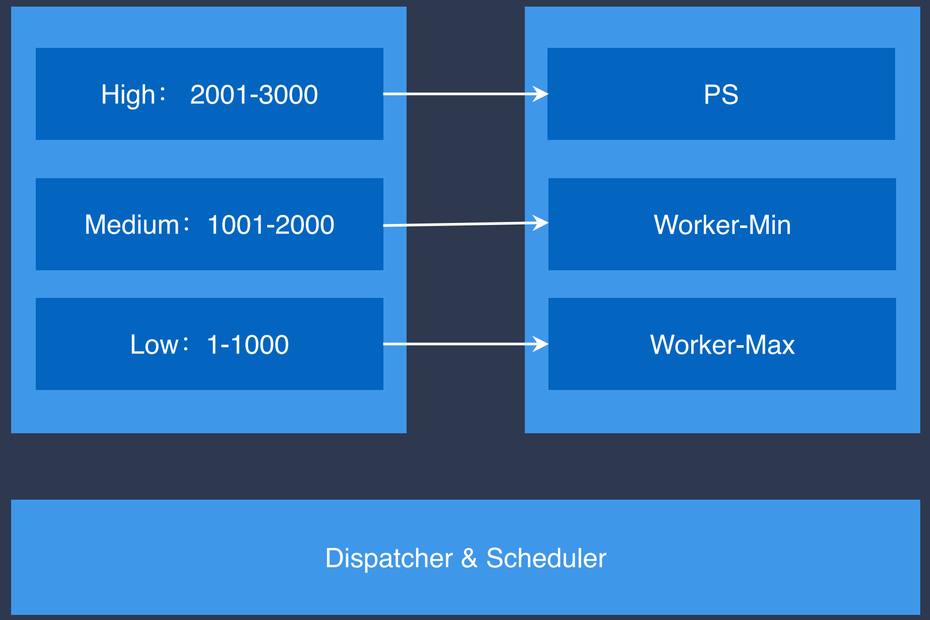
- deletion cost 定制缩容策略
  - 尽可能缩容整机、整 socket
  - 尽可能缩容同 Pod、同质 GPU

资源分配与调度

- 优先满足单作业资源分配
- Gang 语义超时自动回滚调度状态

资源回收

- 基于优先级实现资源回收
- Job 资源回收后通知离线执行 checkpoint
- 在线 > high



弹性并池体系总结

如何弹

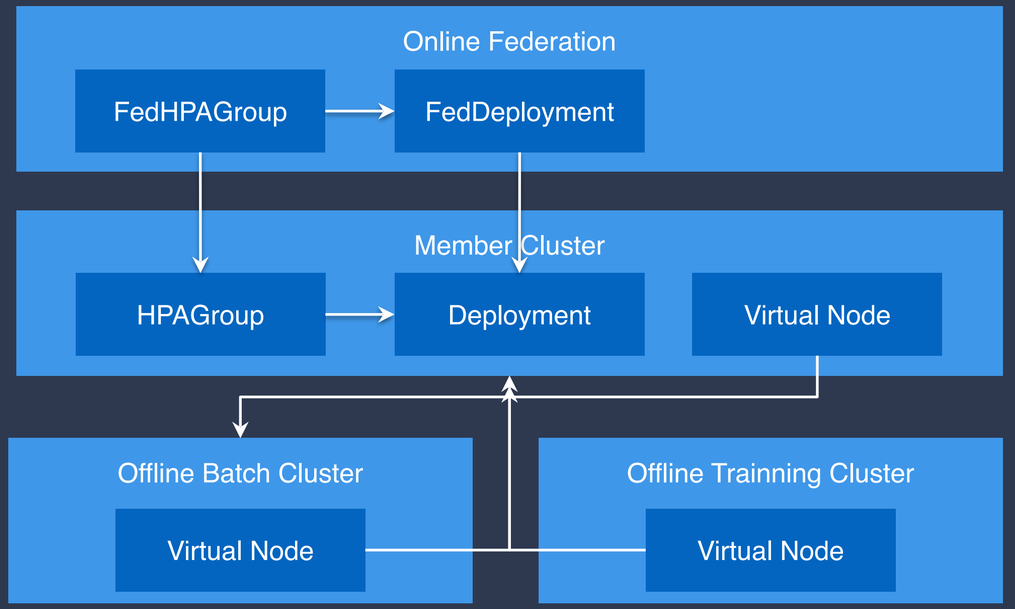
- 在线：天然支持扩展，构建**快**、**稳**的弹性系统
- 离线：协同框架定制弹性能力

如何用

- 单集群内协同资源感知，一体化调度
- 跨集群间资源整合，适配调度语义

如何稳

- 缩容策略匹配资源同质需求
- 以优先级为基础，实现灵活资源分配和回收



大纲

1 背景介绍

2 在离线弹性体系

3 案例分析

4 未来展望

Case 1： Ring AllReduce 使用弹性资源

NLP 基于 horovod + et-operator 实现 RingAllReduce 框架弹性

框架说明

- 中心 Launcher：构建通讯环、检查状态、处理异常
- 稳定资源：弹性资源 = 1：7

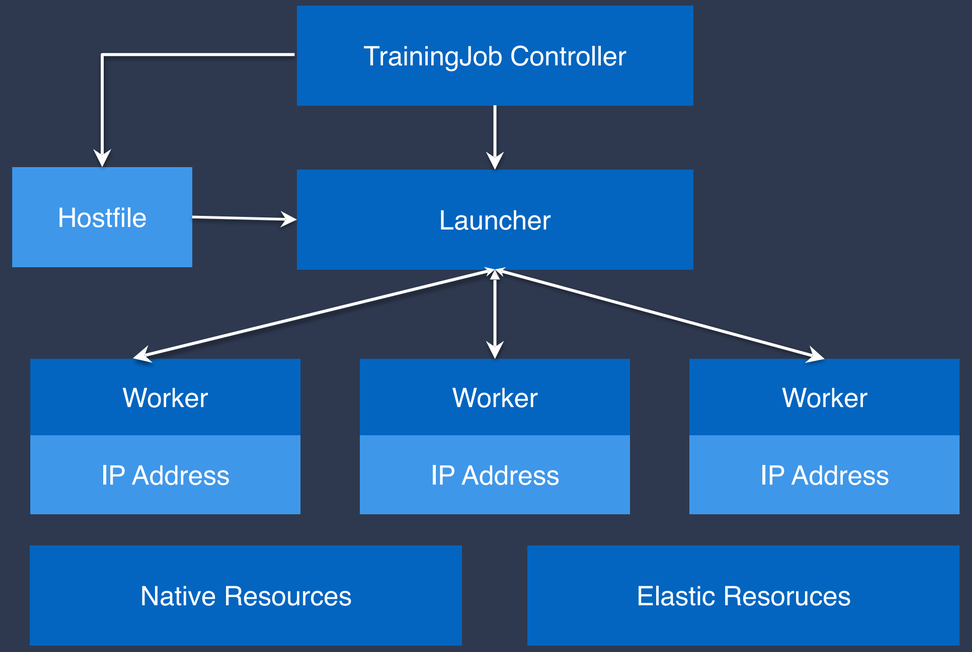
稳定资源

- Launcher 崩溃训练失败
- (0, min」Worker 部署在稳定资源，保证稳定

弹性资源

- (min, max」Worker 部署在弹性资源，加速训练
- 同质 V100 GPU 、同 Pod 调度、RDMA 加速
- 在线推理：模型较大，扩缩即整机 8卡/4卡 GPU

资源收益：日常平均 800 卡 V100 GPU



Case 2：PS-Worker 使用弹性资源

CTV/CVR 采用自研 PS-Worker 框架异步训练

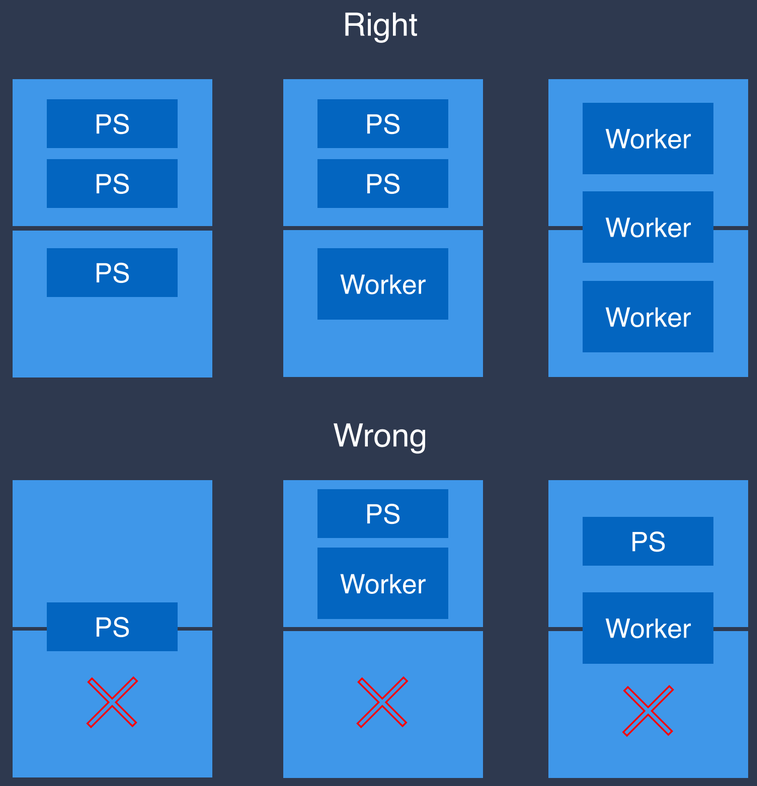
□ 框架说明

- 资源紧张，全量弹性，在线并池共用 CPU/GPU
  - PS on CPU ，同构同 Pod，与 Worker 不共享 NUMA
  - Sparse on CPU，可共享 NUMA
  - Dense on GPU，同构 T4 GPU

□ 弹性要点

- 扩缩即 NUMA：在离线共享部分单机资源
  - Memory Migrate、双网卡、流量打标、OOM Score
- 碎片接入 spark/video transcoding 填充

□ 资源收益：夜间平均 300w 核心 * 7h



Case 3：在线服务使用弹性资源

在线服务弹性资源接入

□ 接入场景

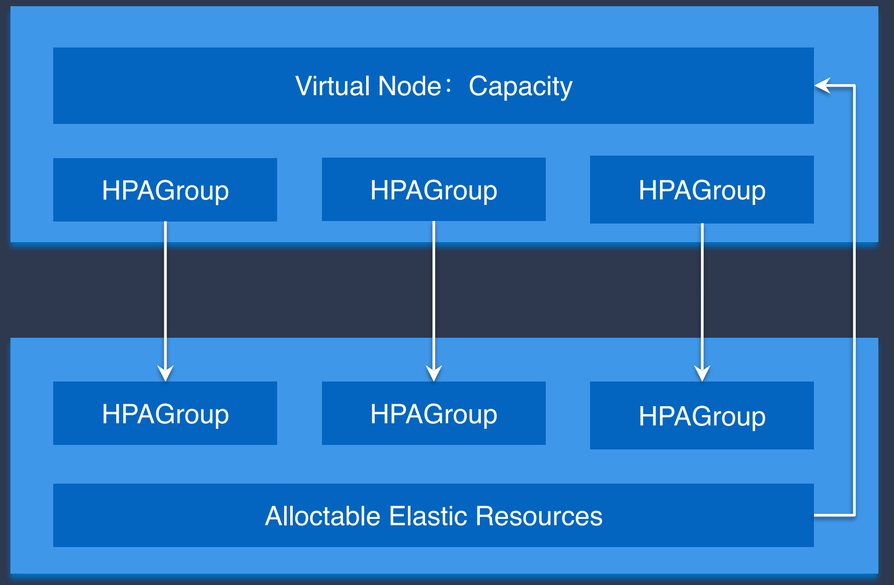
- 核心服务常态降级，晚高峰更多资源正向效果收益
- 促销、节日等活动临时弹性资源供应
- 直播等异常流量增长情况

□ 弹性要点

- CronHPA + 分时 Quota 预先分配配额
- UtilHPA 支持临时突破配额上限
  - 低优离线容量水位管理和报警机制

□ 资源收益

- **日常峰值 40w 核心**
- **春节供应 250w 核心**



大纲

1 背景介绍

2 在离线弹性体系

3 案例分析

4 **未来展望**

后续展望

规模化 + 产品化 + 精细化

- 在线服务弹性规模和时间扩展
- 离线作业云原生和弹性化改造
- 在离线统一联邦

- 弹性容器产品能力完善
- 可视化资源视图和容量统计

- 精细化调度，减少资源碎片
- 集群维度资源切分到全面整合

欢迎交流

THANKS

软件正在改变世界

SOFTWARE IS CHANGING THE WORLD