KUBERNETES COMMUNITY DAYS BEIJING 2025

DRA异构资源精细化管理的生产实践

吉元昊 (@shink), 华为, Apache InLong PMC

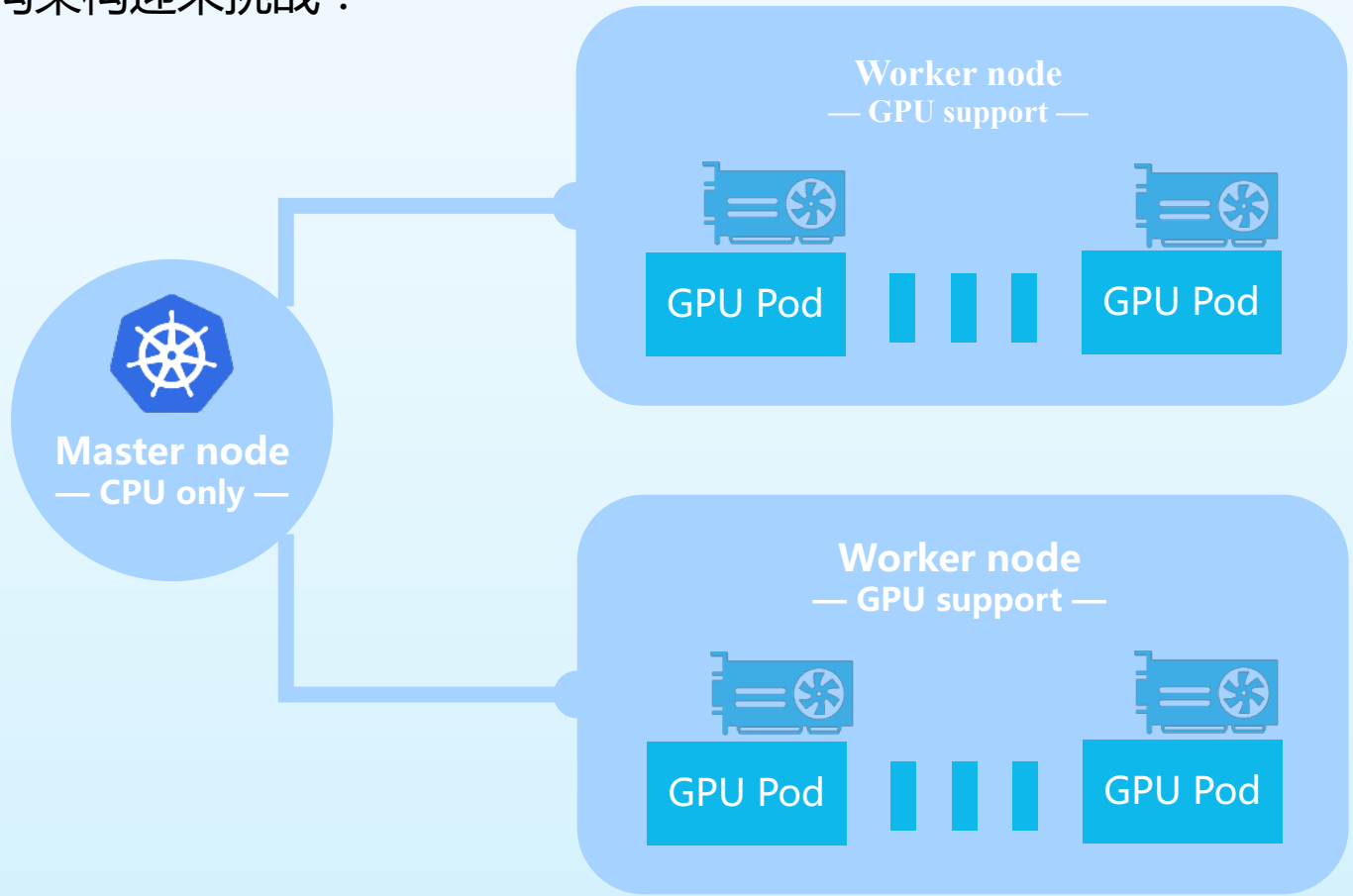
Content

目录

Kubernetes for LLMs

DL训练/推理、HPC 等场景下，K8s 的异构架构迎来挑战：

- 计算
  - 高性能专用加速设备
- 存储
  - 高速本地缓存 + 分布式文件系统
- 网络
  - 高速低延迟网络 / RDMA网络
- 调度
  - 资源争抢 / 亲和性调度

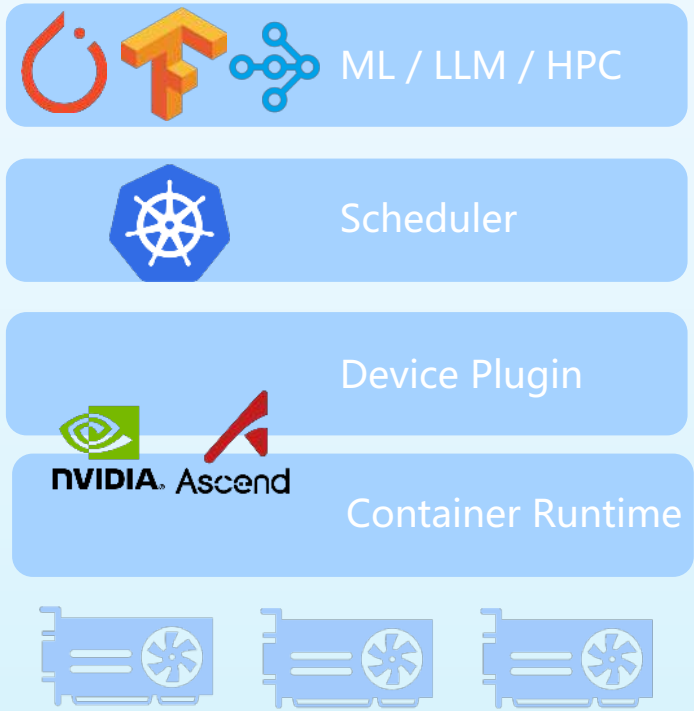


Device Plugin

- Introduced in 1.8，GA in 1.26
- 独立、可插拔的外部组件，由各设备厂商实现
- Device Plugin 负责设备发现、设备分配、健康状态上报
- Runtime 负责环境配置、设备挂载等底层功能支持

【解决的问题】

1. 像 CPU/Mem 资源一样，使 K8s 能感知到 GPU 资源
2. 适配不同的硬件



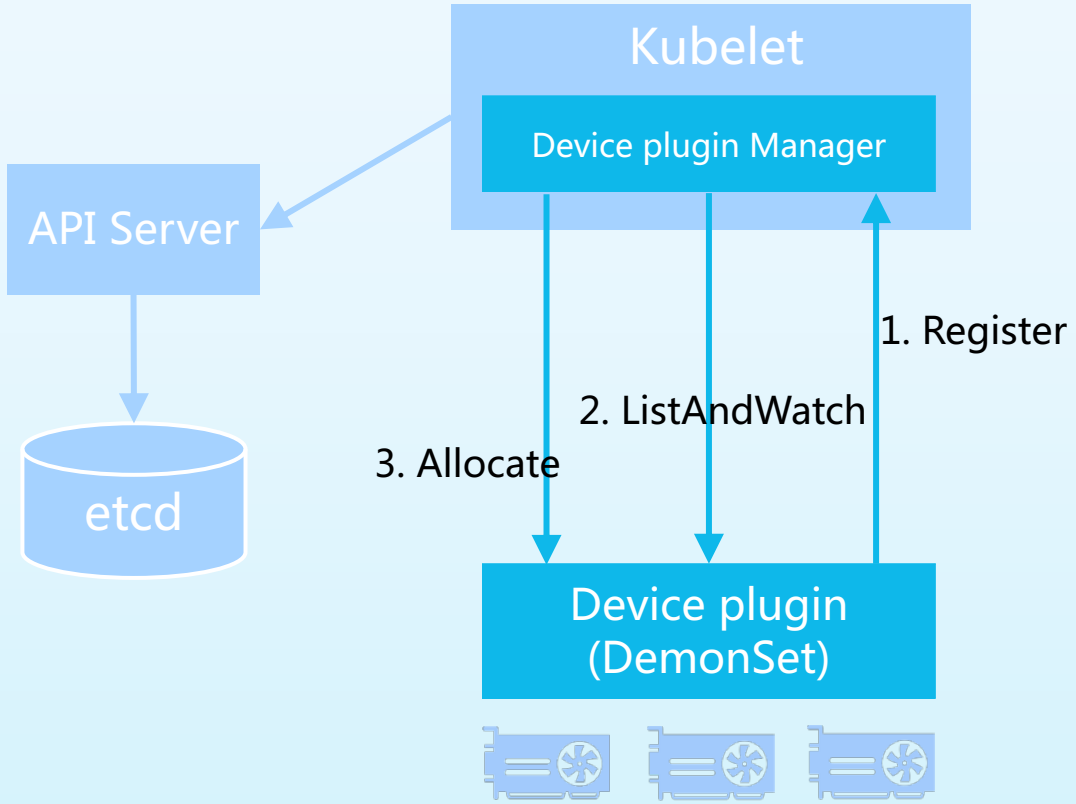
Device Plugin

【流程】

1. DP 作为 DaemonSet 方式启动
2. DP 调用 Kubelet 的 *Register()* 接口进行注册
3. Kubelet 调用 DP 的 *ListAndWatch()* 查询设备资源
4. Kubelet 调用 DP 的 *Allocate()* 分配设备资源，设置环境变量： *ASCEND_VISIBLE_DEVICES*

【不足】

1. 设备共享（DP 仅支持资源独占）
2. 复杂条件的设备筛选（DP 仅支持描述用量）
3. 亲和性调度（DP 不支持描述设备间的拓扑关系）

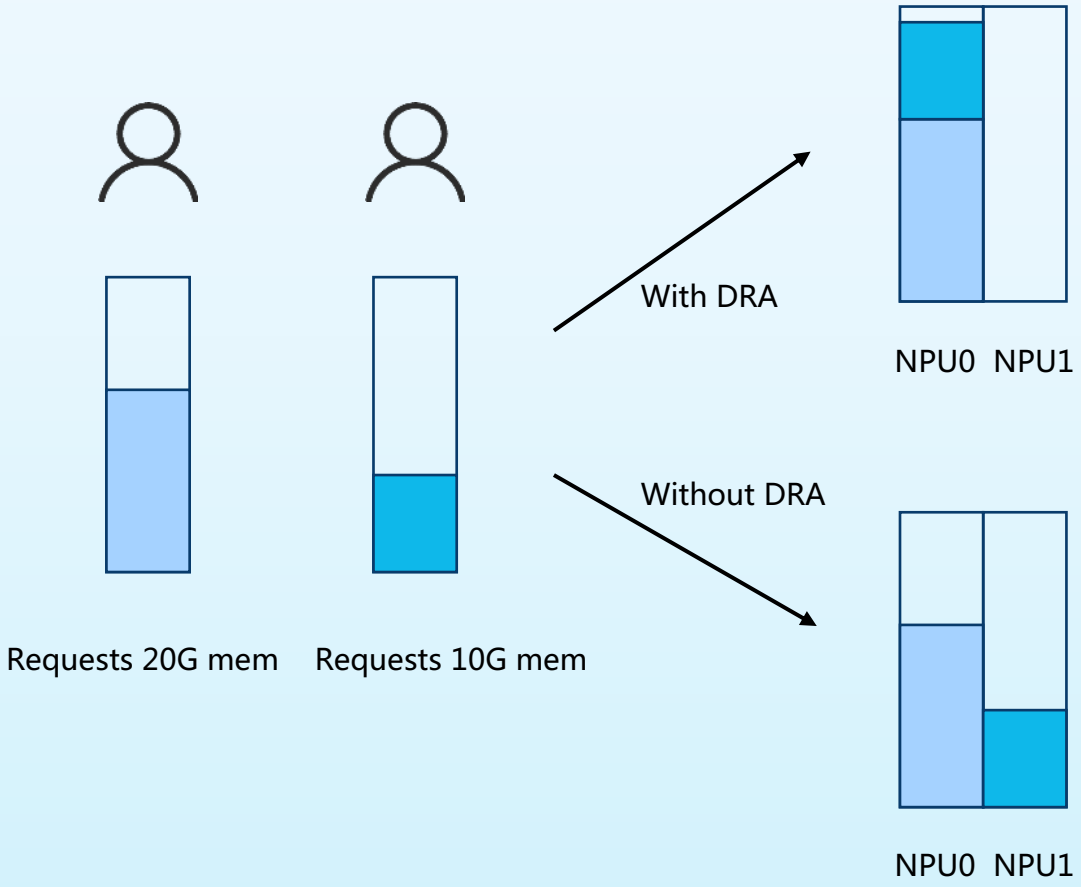


Dynamic Resource Allocation

- 资源请求的新方式:
  - Alpha since 1.26
  - Major API changes in 1.31
  - Beta in 1.32
  - GA in ??
  - Disabled by default

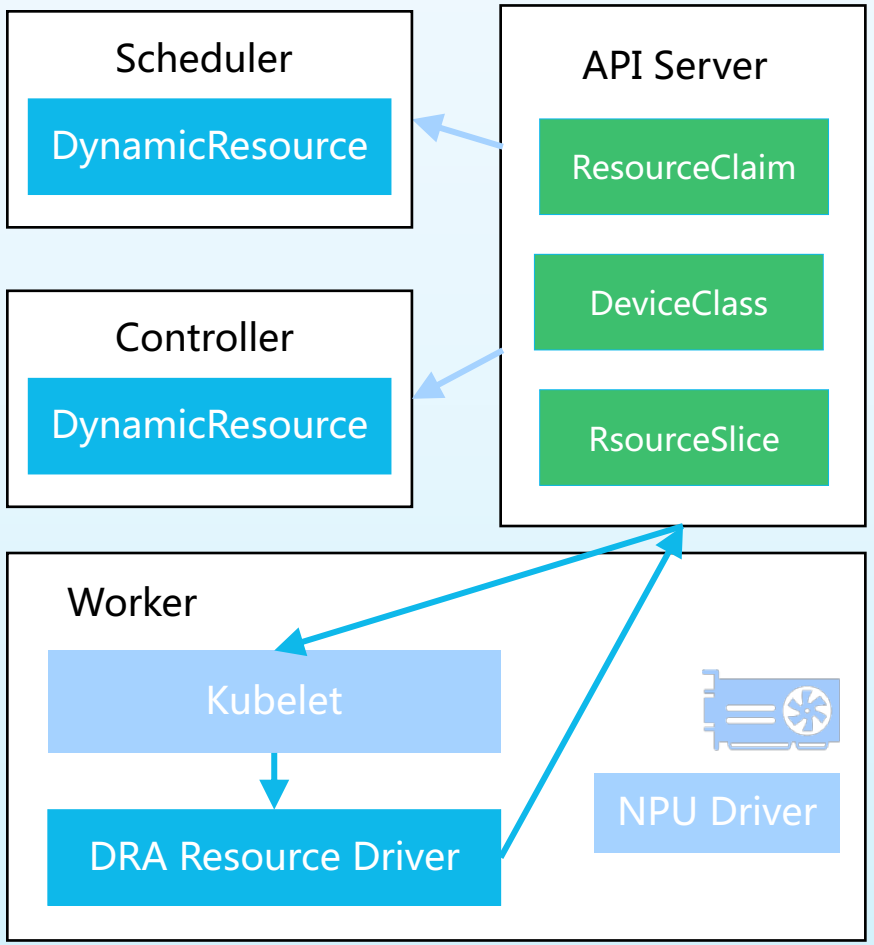
【优势】

- 弥补 Device Plugin 无法应对复杂需求描述的不足
- 提供多元化的标签与选择功能，辅助 Scheduler 决策



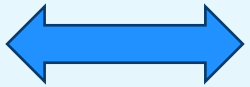
Dynamic Resource Allocation

- DRA Resource Driver:
  - 向 kubelet 注册资源账本信息
  - 从 ResourceSlice 对象中检索可用资源，同时跟踪已分配给 ResourceClaim 的资源，然后从剩余的资源中进行选择
  - 在调度器完成调度决策后，DRA Driver 负责完成设备挂载/初始化工作
- 资源：
  - DeviceClass: 描述设备参数，支持 CEL 表达式
  - ResourceClaim：描述请求资源，支持 CEL 表达式
  - ResourceSlice：描述集群中可用资源片段



DP vs DRA

```
kind: Pod
spec:
  containers:
  - name: container0
    resources:
      requests:
        huawei.com/Ascend310P: 2
```



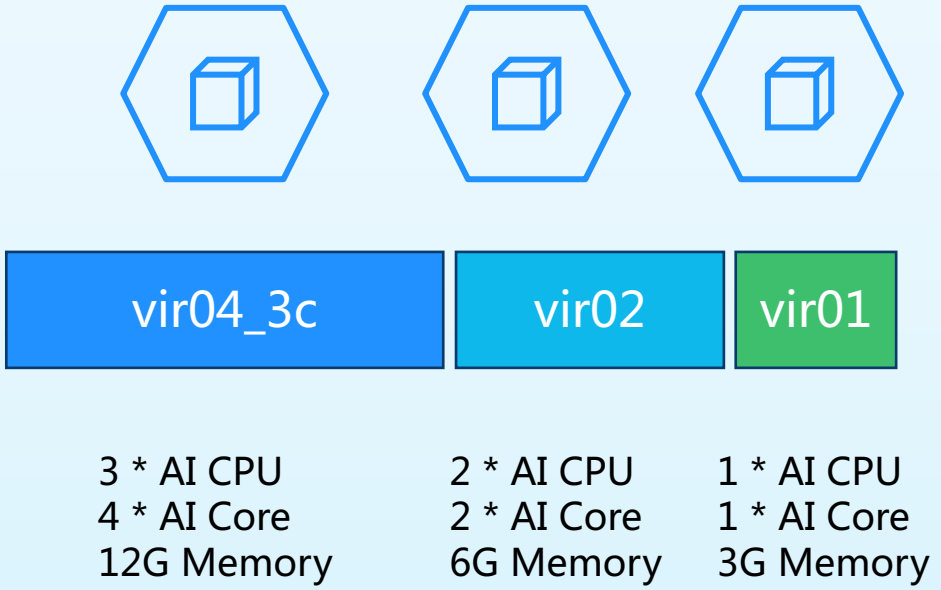
```
apiVersion: resource.k8s.io/v1beta1
kind: DeviceClass
name: huawei.com/ascend-310p-1/4
spec.selectors:
  - cel:
      expression: device.model == "Ascend310P”
---
apiVersion: resource.k8s.io/v1beta1
kind: ResourceClaimTemplate
metadata:
  name: 310p-1/4-template
spec.spec.devices:
  requests:
  - deviceClassName: "huawei.com/ascend-310p-1/4"
    selectors:
      - cel:
          expression: |-
            device.attributes["model"] == "Ascend310P”
---
apiVersion: resource.k8s.io/v1beta1
kind: Pod
spec:
  containers:
  - image: ascendai/cann:8.0.0
    resources:
      claims:
      - name: 310p-1/4
  resourceClaims:
  - name: 310p-1/4
    resourceClaimTemplateName: 310p-1/4-template
```

KCD GROWING CLOUD NATIVE TOGETHER BEIJING

DRA on Ascend

vNPU 虚拟化：

- 通过资源虚拟化的方式、按照虚拟化实例模板，将物理 NPU 切分成若干份 vNPU
- 虚拟化方式：硬件虚拟化 / 软件虚拟化
- DRA on Ascend 实现：
  - 静态虚拟化：事先基于模板切分 NPU，并创建对应的 DeviceClass
  - 动态虚拟化：根据 RCT 中声明的模板切分 NPU，上报的 ResourceSlice 中声明所有分配方案的规格

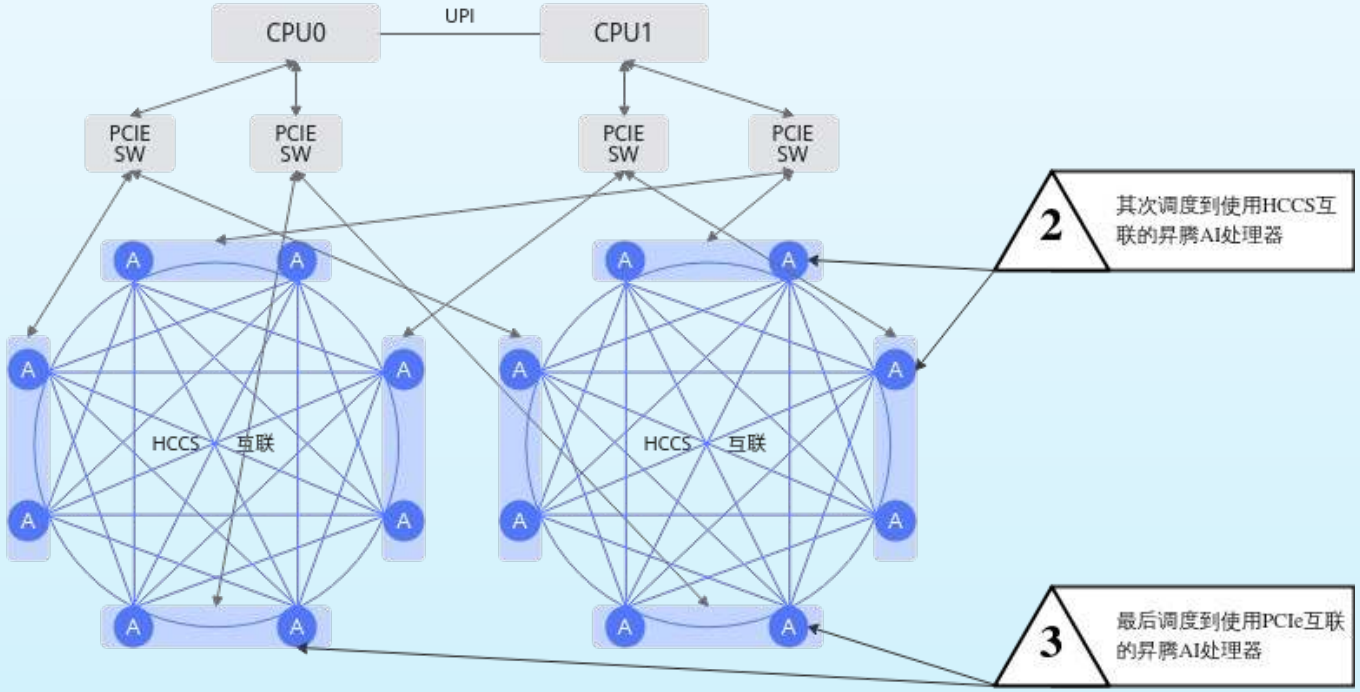


DRA on Ascend

NPU 调度：

- 类似 Volcano + DP 的效果
- 优先使用同一张卡
- 其次调度到使用 HCCS 互联的节点
- 最后调度到使用 PCIe 互联的节点

例1 A300I DUO推理卡



例2 Atlas 200T A2 Box16

Thanks.