
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

在 Kubernetes 上构建一个精细化和智能化的资源管理系统

曹贺 & 邵伟

字节跳动

日程

- Katalyst 概览
- 应用场景
  - 在离线混部
  - GPU 共享调度
  - 拓扑感知调度
  - 资源效能套件
- 实践效果
- 社区介绍

1 Katalyst 概览

字节跳动的服务类型

微服务

- 实现应用的业务逻辑
- Golang
- 调用链路复杂
- 重 CPU 和 RPC 时延

搜广推服务

- 为 Feed 和搜索提供内容列表的后端服务
- 实时在线推理
- C++
- 追求极致性能，单个服务的资源消耗量大

机器学习和大数据

- 为搜广推离线训练、数据报表提供支撑的数据处理服务、视频转码任务
- 重内存和吞吐

存储服务

- 传统存储、数据库、NoSQL 等
- 有状态应用
- 故障影响面大
- 对资源的稳定性要求高

ByteDance ❤️ Kubernetes

90 万+
节点

1.1 亿+
Pod

支持了数亿用户

600 万+
Deployment

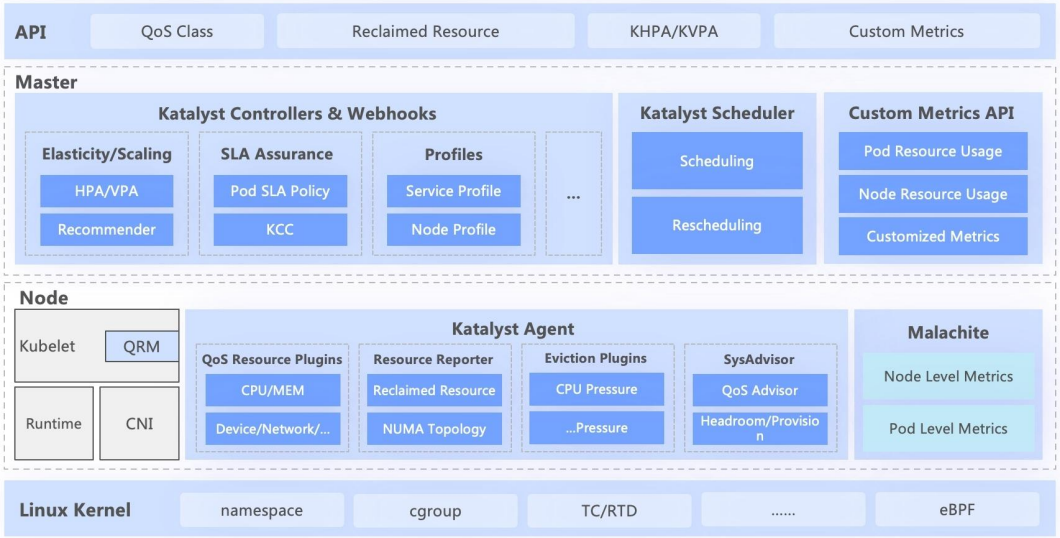
30 万+
离线作业

Katalyst 概览

Katalyst

Katalyst 引申自 “catalyst”，本意为化学反应中的催化剂

旨在为运行在 Kubernetes 上的工作负载提供更加强劲的资源管理能力



https://github.com/kubewharf/katalyst-core

中心组件

- Katalyst Controllers & Webhooks
- Katalyst Scheduler
- Katalyst Custom Metric

单机组件

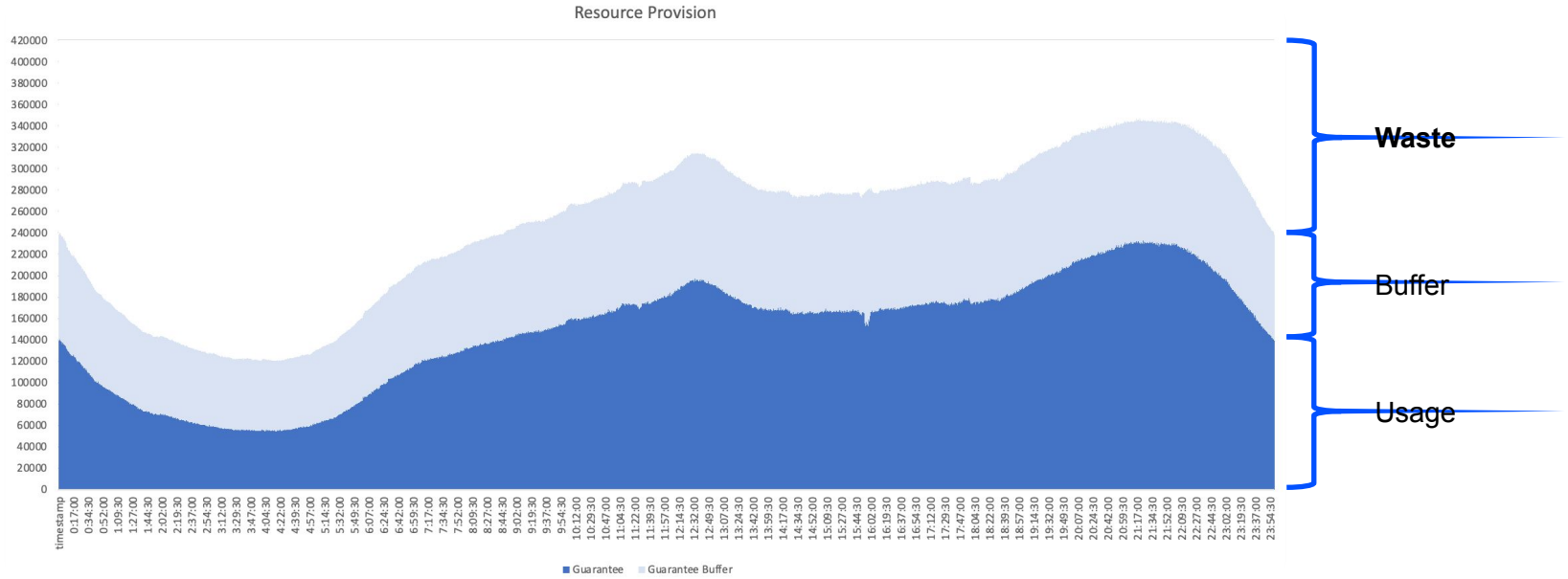
- QoS Resource Manager (QRM)
- Katalyst Agent
  - QRM Plugins
  - SysAdvisor
  - Resource Reporter
  - Eviction Manager
- Malachite

2 应用场景

2.1 在离线混部

资源规划的挑战

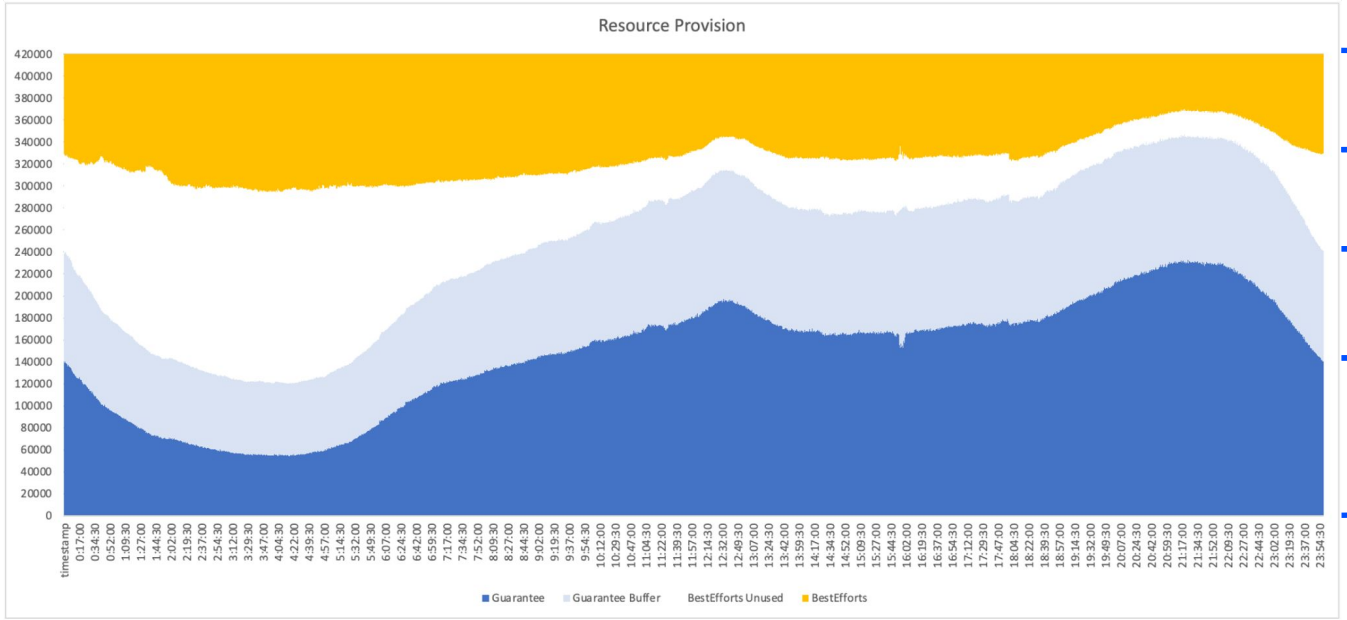
- 在线业务的资源利用率呈现潮汐现象，夜间的资源利用率非常低
- 用户倾向于过度申请资源以保证服务的稳定性，造成资源浪费



在离线混部

在线业务和离线作业对资源的使用模式天然是互补的:

- 在线业务重 CPU 和 RPC 时延
- 离线作业重内存和吞吐



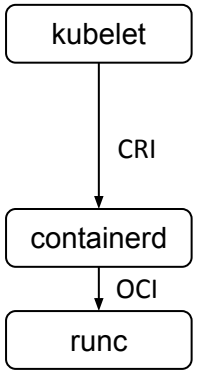
扩展的 QoS 级别

- **4 种扩展的 QoS 级别**
  - 表达了服务对资源质量的要求
  - 以 CPU 为主导的维度来命名
- **更多 QoS Enhancement**
  - NUMA 绑定
  - NUMA 独占
  - 流量分级
  - …

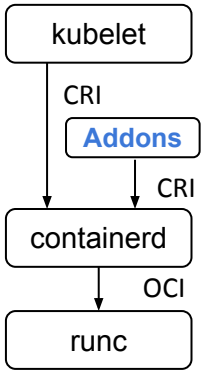
QoS Class	特性	适合的负载类型	与 K8s QoS 的映射关系
dedicated_cores	• 独占核心，不与其他 负载共享 • 支持 NUMA 绑定，提供更极致的性能体 验	对延迟极度敏感的业务，比如搜索、广告、推荐等	Guaranteed
shared_cores	• 共享 CPU 池 • 支持根据业务场景进一步切分 CPU 池	可以容忍一定的 CPU 限流或者干扰的业务，比如微服务	Guaranteed/ Burstable
reclaimed_cores	• 超售资源 • 资源质量相对无保障，甚至可能被驱逐	对延迟不敏感、更在乎吞吐的业务，比如离线训练和批处理作业	BestEffort
system_cores	• 预留核心 • 保障系统组件的稳定性	关键的系统组件	Burstable

插件化的资源管理

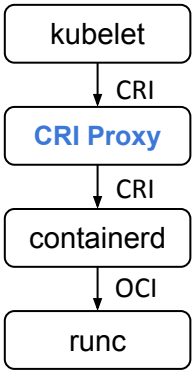
原生 K8s



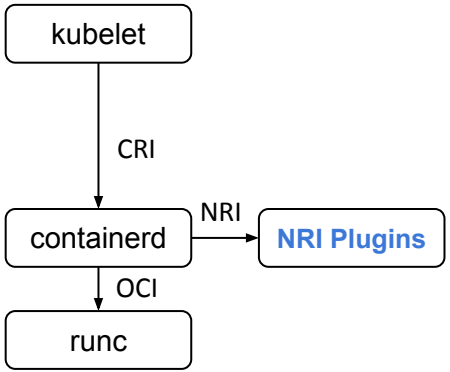
旁路异步更新



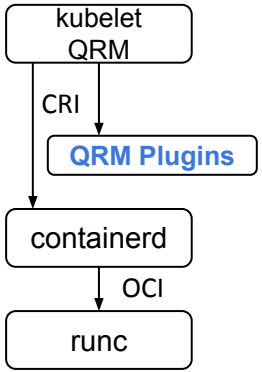
劫持 CRI 请求



NRI

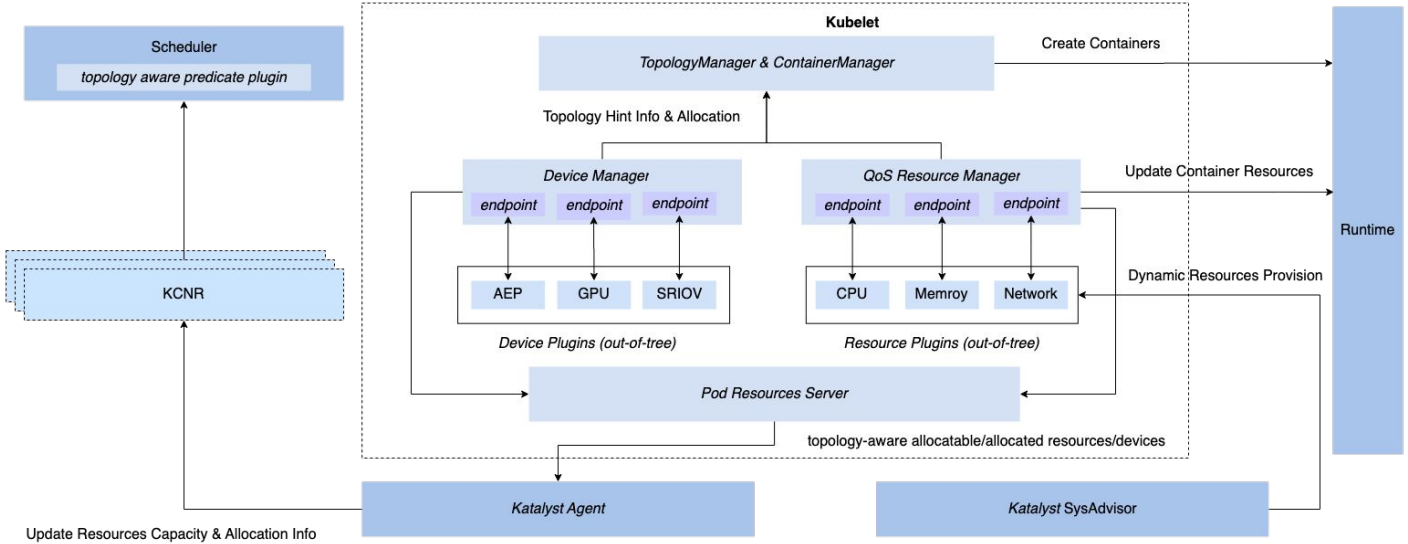


Kubelet 侧 Hook



寻找扩展资源管理策略的最佳 Hook 点

QoS Resource Manager

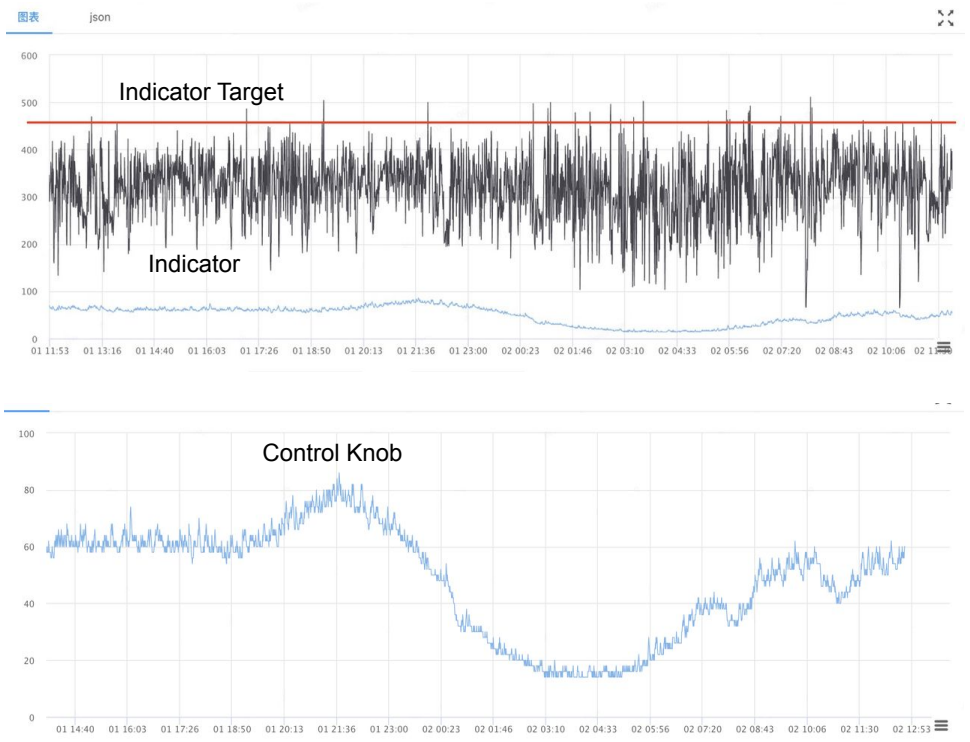


- **QoS Resource Manager**
  - 为资源管理插件提供注册机制
  - 作为 Hint Provider 注册到 Topology Manager
  - 周期性调用运行时，更新为容器分配的资源
- **QoS Resource Plugin**
  - 定制对容器的资源分配策略

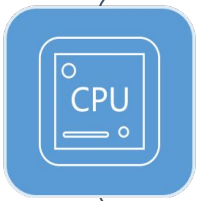
服务画像与资源预测

基于负反馈的 PID 控制算法

- 分析服务的业务指标和系统指标之间的关系
- 持续调整 Control Knob 使 Indicator 当前的值不断逼近 “甜点位”



多维度的资源隔离机制



CPUSet
Intel RDT
SCHED_IDLE

Per-memcg 异步内存回收
NUMA 绑定
用户态的 Advisor

在线业务的日志 IO 异步化
在离线分盘
IOCost

多网卡实现硬隔离
流量分级
通过 eBPF + EDT 限制带宽

隔离是手段而不是目的，根据业务场景寻找最合适的方式

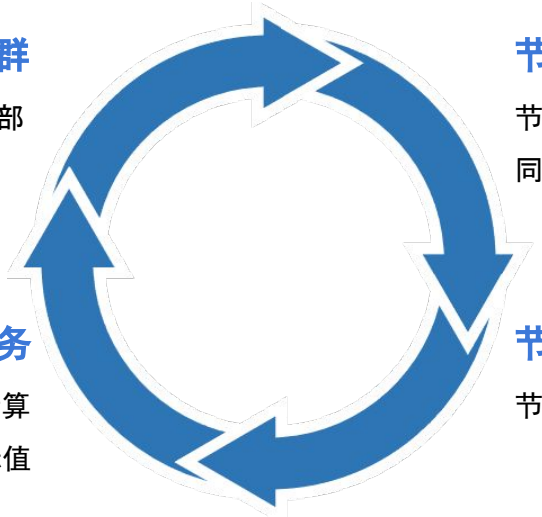
多层级的动态配置管理

集群

全局默认配置，比如是否开启混部

节点池

节点池级别的配置，比如为 CPU 和 GPU 节点池配置不同的资源管理策略



服务

服务级别的差异化 SLO 配置，比如通过服务画像计算出的 Indicator 的目标值

节点

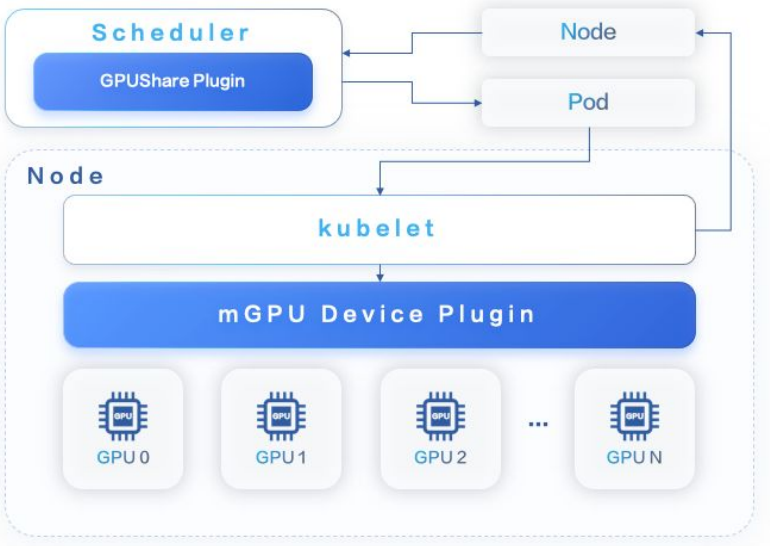
节点级别的配置，比如为了调试问题而做的临时配置

2.2 GPU 共享调度

GPU 共享调度

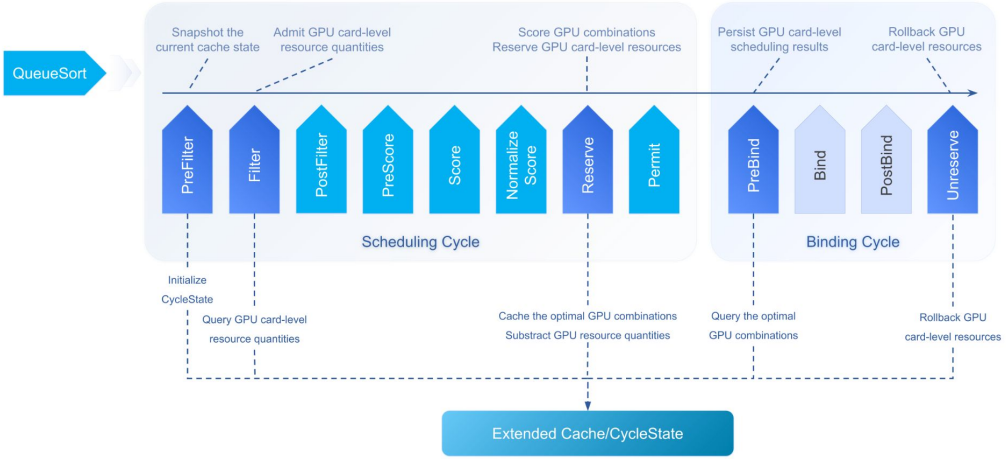
- K8s 原生只支持容器申请整数个 GPU，在 AI 推理场景下会浪费大量昂贵的 GPU 资源
- mGPU 支持 1% 算力粒度和 1 MiB 显存粒度的容器调度

整体架构

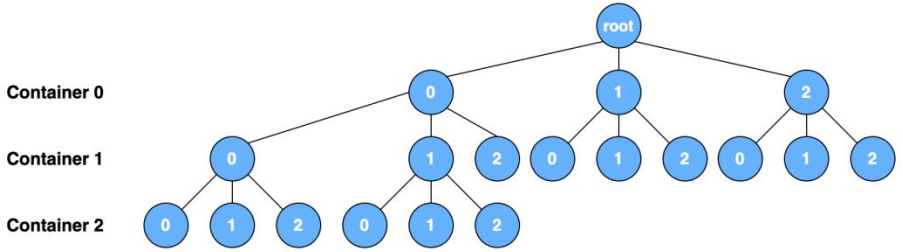


即将到来的相关分享：https://sched.co/1Rj4O

调度插件



调度算法

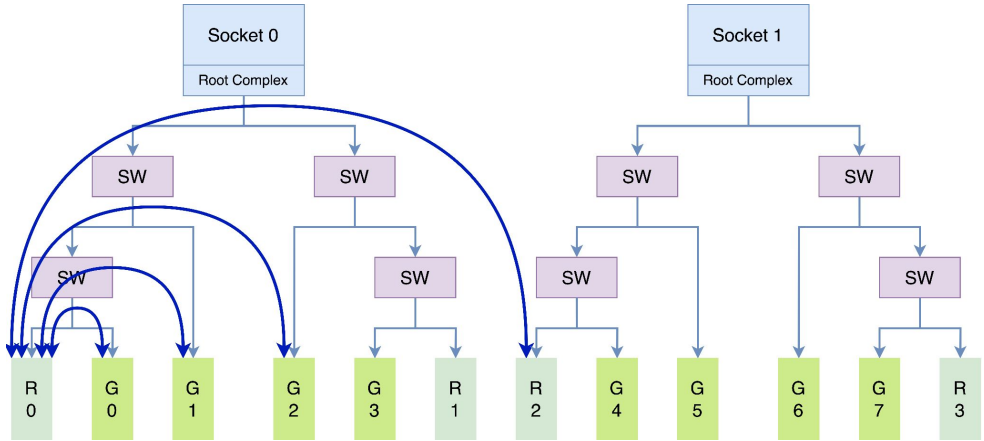


2.3 拓扑感知调度

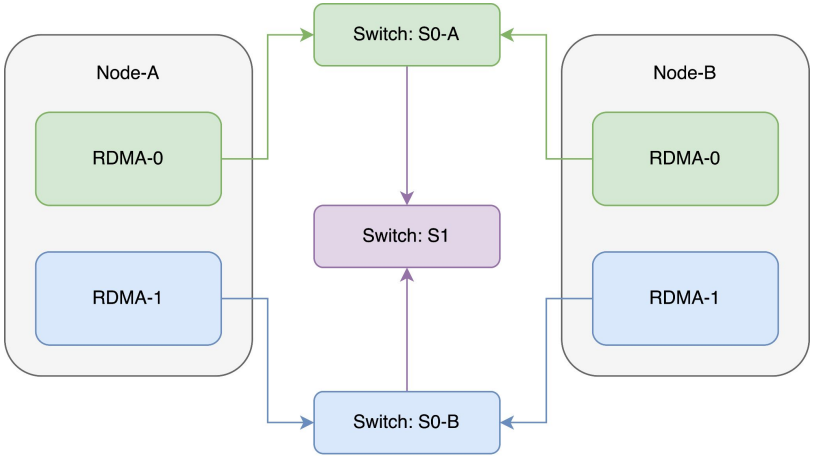
拓扑感知调度

- K8s 原生调度器不感知节点的微拓扑，可能导致大量的 Admit 失败
- K8s 原生的拓扑亲和策略只考虑了 NUMA 拓扑

GPU 和 RDMA 在 PCIe Switch 级别的亲和



RDMA 之间在交换机级别的亲和



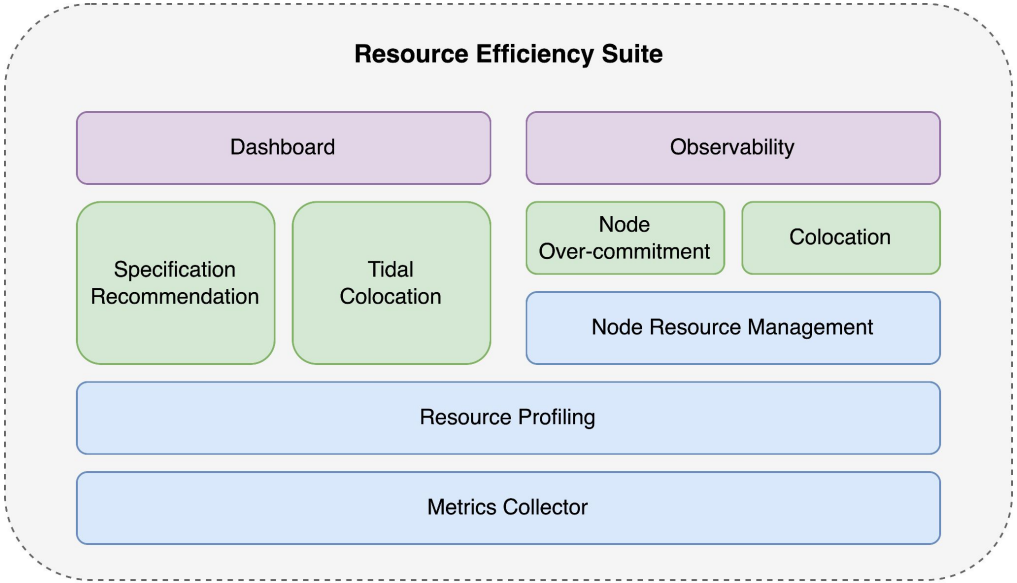
即将到来的相关分享：https://sched.co/1Rj4O

2.4 资源效能套件

资源效能套件

对于云的用户来说，落地混部的门槛比较高

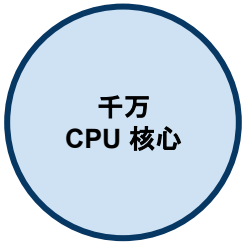
- **规格推荐**
  - 重建更新
  - 原地更新
- **潮汐混部**
  - HPA/CronHPA/智能 HPA
  - 节点池管理
- **节点资源超分**
  - 在用户无感知的情况下，允 许调度器调度更多 Pod
  - 干扰检测与缓解
  - 长短周期结合的资源预测算法



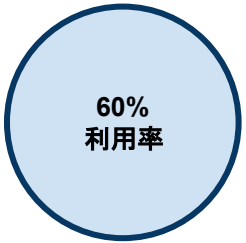
3 实践效果

实践效果

部署了超过 90 万节点



管理了千万级别的 CPU 核心



将天级资源利用率从 23% 提升到 60%

4 社区介绍

里程碑

版本	状态	日期	核心特性
0.1	已发布	2023.02.27	• 在离线混部 (MVP 版本)
0.2	已发布	2023.06.13	• 支持 dedicated_cores with numa_binding (节点侧) • 支持在分配网卡时考虑 NUMA 亲和 • 流量分级 • 基于 RSS 超用情况的驱逐
0.3	已发布	2023.08.08	• 动态配置 • 基于 PID 控制算法的服务画像 • 用户态内存管理 (Drop Cache、内存迁移、离线大框等)
0.4	待发布	2023 年 9 月底 (预计)	• 拓扑感知调度 • 规格推荐 • 潮汐混部 • 节点资源超分 • IOCost
…			• 支持 dedicated_cores with numa_binding (调度器侧) • OOM 优先级 • …

联系方式

- **社区双周会议**
  - 周四 19:30 (北京时间)
  - 会议记录与日程
- **Slack**
  - kubewharf.slack.com
  - Channel: katalyst
- **社区飞书群**

- **曹贺**
  - Email: caohe.ch@bytedance.com
  - GitHub: @caohe
- **邵伟**
  - Email: shaowei.wayne@bytedance.com
  - GitHub: @waynepeking348
- **后续即将到来的分享**
  - https://sched.co/1Rj4O
  - https://sched.co/1Rj3f

GitHub 仓库：https://github.com/kubewharf/katalyst-core


KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
Thank you!