GPU 集群
参与实践与生态展望

第一部分介绍参与过的监控和占用治理；第二部分讨论面向团队的 GPU 计算服务生态。

PART 01 · 参与过的工作

监控与资源占用治理

集群 / Pod 可观测性

历史指标、策略窗口与白名单

清理与原因通知

PART 02 · 展望 / 方案设想

统一的 GPU 计算服务生态

开发、训练、数据与应用运行

资源治理、可靠性与成本

自助入口与平台能力建设

参与过的工作

已分配、活跃和高效，是三个问题

长驻开发 Pod 与批任务共享异构设备；进程仍在，不代表 GPU 正在持续工作。

ALLOCATION

设备被谁占用?

资源申请与设备分配

卡型、节点、namespace、任务

ACTIVITY

任务是否有活动?

GPU Utilization、SM Activity

持续窗口，避免看瞬时值

EFFICIENCY

算力是否用得好?

SM Occupancy、带宽、任务进度

需要业务耗时与 profiler 验证

治理目标：减少长期占用与人工巡查。没有真实统计，不报告释放卡数或成本节省比例。

参与过的工作

把监控信号接到定时治理

Python / Kopf 在约定范围内检查历史信号，命中策略后执行 Pod 清理并通知。

01
取证
Prometheus
CPU / GPU / SM

02
判定
历史窗口
样本数与阈值

03
过滤
类别 / 时段
名称白名单

04
清理
Kubernetes API
删除 Pod

05
通知
钉钉消息
原因 + dashboard

600 秒
周期检查
Kopf timer 绑定自定义资源

60 秒
查询步长
样本数达到门槛，所有点低于阈值

现有顺序
删除后通知
未实现预告、延期与释放确认状态机

main.py:83 / :109–110 · prometheus.py:67–74

同一信号，按使用场景应用不同窗口

开发分支：CPU 低活跃 ∩（GPU 低活跃 ∪ SM 低活跃）。Prod 分支不附加 CPU 条件。

代码分支	观察窗口	判定与保护
Prod 启发式分类	60 分钟	GPU 或 SM 持续低活跃
1080 / 2080 开发分支	180 分钟	优先于时段判断
其他开发 · 工作时段	180 分钟	工作日 09:00–20:00
其他开发 · 非工作时段	30 分钟	名称前缀 / 完整名称白名单优先跳过

“Prod”由节点类型和申请卡数推断，并非显式业务等级。阈值单位与注释存在待核对项，不能直接把 0.05 解释为 5%。

集群总览：让分配与活动可被比较

用户提供的 Grafana 配置包含 19 个非 row 面板，支持节点、namespace 与型号观察。

CAPACITY & ACTIVITY

卡数 · 分配 · 活跃

总量与关联 Pod 的设备

型号维度趋势 / 节点分配明细

GPU 活跃与总功率、平均温度

H100 A6000 4090 3090 2080 Ti 1080 Ti

节点视角

驱动 / 卡型 / CPU / 内存 / 时钟

任务视角

下钻 Pod 利用率与显存使用率

口径需一致

设备序列去重、业务 Pod 标签与筛选范围

此页是原始 dashboard 配置的覆盖说明，不是实时截图；型号标签不用于推算集群规模。

Pod 下钻：GPU Utilization 不够解释效率

按设备观察 SM 指标，并把 GPU 与 CPU、内存和网络放到同一时间窗口。

指标	能回答的问题	边界
GPU Utilization	是否存在 GPU 工作	不表示所有 SM 都充分参与
SM Activity	SM 有活跃 warp 的时间比例	等待内存的 warp 也可能算活跃
SM Occupancy	驻留 warp 相对理论容量	高值不直接等于高吞吐
显存 / PCIe / CPU / 网络	容量、传输与主机侧关联	需要阶段计时与对照实验确认瓶颈

监控帮助缩小范围；判定是否可以回收，还需要任务身份、进度与恢复条件。

参与范围：让资源状态可见，让占用可处理

实际实践集中在两层监控视图、周期判定和 Pod 清理通知。

参与内容	已有实现	当前边界
可观测性	集群总览与 Pod 资源下钻	配置可以复核，收益指标待补充
占用判断	CPU / GPU / SM 窗口与白名单	启发式任务分类、量纲需核对
治理动作	删除 Pod 后发送原因和链接	尚无预告、延期与释放确认状态机
实践定位	监控、策略及实现迭代	不等同于已建设完整 GPU 平台

> 下面进入生态展望：把 GPU 作为团队可持续使用的计算服务。

展望 · 方案设想

生态全景：打造团队可持续使用的 GPU 计算服务

让用户容易获得可复现的计算环境，让团队公平使用资源，让平台能可靠运营。

01 / SUPPLY

资源供给

设备库存、卡型与拓扑
驱动、网络、存储和容量

02 / DEVELOP

开发与实验

工作空间、环境模板、调试
实验记录、协作与可复现性

03 / EXECUTE

任务与应用运行

批处理、分布式训练、推理
排队、恢复、服务发布

04 / DATA

数据与产物

数据版本、缓存、checkpoint
模型、评估结果与保留策略

05 / GOVERN

资源与成本治理

身份、配额、公平性、租约
审计、归属与成本核算

06 / OPERATE

可靠性与运维

监控、告警、诊断、升级
SLO、备份恢复与值班

共同基础：项目 / 用户身份 · 统一任务标识 · 版本与配置 · 策略 API · 可追溯事件

展望 · 方案设想

用户旅程：从一次开发实验到稳定运行的应用

统一项目与任务上下文，把环境、数据、资源和产物接起来。

01
准备
项目 / 配额
镜像 / 数据

02
开发
IDE / Notebook
调试 / profiling

03
执行
训练 / 批处理
排队 / 重试

04
交付
评估 / 模型版本
批推理 / 服务

05
运营
告警 / 升级
资源归还 / 归档

开发者
容易开始、容易复现
环境模板与示例任务
输入、代码和镜像版本可追踪

任务负责人
过程可见、失败可恢复
排队解释、日志与进度
checkpoint、取消与恢复契约

团队管理者
公平使用、预算有据
项目配额、优先级与用量归属
服务质量和资源需求可衡量

展望 · 方案设想

生态组合：成熟能力复用，平台把契约接好

候选组件按场景选择，核心建设是身份、任务、环境、策略与证据的统一。

能力层	可评估的生态组件	平台需要补齐的连接
设备与集群	Kubernetes · GPU Operator / device plugin	驱动兼容矩阵、拓扑、健康与生命周期
准入与运行	Kueue 或 Volcano；Ray / 训练 operator	配额与优先级、Job 状态、取消和恢复
开发与产物	JupyterHub / 远程 IDE · MLflow · 镜像/对象存储	模板、数据与模型版本、评估和保留
推理与应用	按需评估 KServe / Triton	部署契约、扩缩容、发布与回滚
观察与交付	DCGM · Prometheus/Grafana · GitOps/CI	关联任务和资源，诊断、升级及审计

组件为候选，不是已部署清单；Kueue 侧重作业准入，Volcano 提供批调度能力，按现有环境选择集成路线。

展望 · 方案设想

治理与运维：公平、高效、可靠，要分别衡量

既管理用户和任务，也管理设备健康、服务质量与变更风险。

领域	要建立的能力	关注的结果
公平分配	项目配额、优先级、卡型/拓扑队列、抢占约定	不同任务等级的等待时间与饥饿情况
运行效率	任务性能诊断、数据缓存、容量规划、用量归属	吞吐/延迟、有效使用、按型号成本
可靠性	GPU/节点健康、SLO、告警分级、值班与 runbook	启动成功率、任务故障、MTTR
升级与恢复	驱动/框架兼容、维护排空、灰度、备份演练	变更失败率、恢复结果、RPO/RTO
权限与生命周期	SSO/RBAC、数据隔离、密钥、审计、保留策略	权限可追踪、产物可恢复、例外有期限

训练、交互开发与在线推理采用不同服务目标。释放 GPU-hours、业务吞吐与真实成本收益分开统计。

一个平台，多种入口：用户与运维各取所需

统一身份、环境、任务、资源、诊断与策略 API；CLI、Web、Bot 使用相同的权限和状态。

CLI / SDK

开发与自动化

创建环境、校验与提交任务

查询配额、排队、日志和产物

集成 CI；诊断支持 JSON 输出

WEB

发现与协作

资源目录、环境模板、任务视图

实验 / 模型产物与团队用量

延期、审批、申诉和运维看板

BOT

解释与辅助

“如何开始这个训练任务？”

“为什么排队或运行变慢？”

引用文档与证据，操作走策略 API

建设先从只读 CLI 与已有 Grafana 集成开始，再按需求增加任务入口、门户和 bot。

展望 · 方案设想

Agent：围绕任务调查问题，帮助用户推进下一步

连接任务、Argo、Pod、GPU 与运行证据，让用户直接描述目标。

FIRST / DIAGNOSE

任务诊断与治理解释

为什么排队、失败或低活跃?

关联事件、日志与 DCGM

给出证据与下一步检查

NEXT / PREPARE

任务提交助手

解析模型、数据与 baseline

选择模板、校验参数和资源

生成提交预览，确认后执行

LATER / EXPERIMENT

性能优化实验助手

固定负载、建立基线

在预算内选择对照实验

验证吞吐、显存与结果质量

> 先回答“为什么”和“接下来怎么做”，逐步开放经过授权的操作。

已有监控与 Argo 平台是接入起点；这些 Agent 能力均属于建设设想。

展望 · 方案设想

Skill：把团队经验沉淀成可复用的工作方法

Agent 选择调查路径，Skill 提供方法，平台工具提供证据与受控执行。

层次	负责什么	具体内容
Agent	理解目标、选方法、逐步验证	确定 Job、提出假设、选择下一项检查
Skill	适用条件、检查顺序与判断边界	指标语义、反证、停止条件、输出模板
环境适配	把通用方法接入实际集群	标签与 UID 映射、数据源、模板和 runbook
平台 API	数据访问、动作与权限控制	授权、配额、审批、幂等、审计与操作验证

Skill 包可包含 SKILL.md、参考资料、查询、采集脚本与案例。维护负责人、版本和回放评测；权限由平台执行。

展望 · 方案设想

首个 MVP：解释一次低活跃候选

从已有占用治理切入，先做只读解释，再讨论延期、恢复和处置。

01
定位
Job / Pod UID
owner / controller

02
取证
规则版本 / 窗口
CPU / GPU / SM

03
核对
单位 / 新鲜度
任务阶段与例外

04
解释
命中依据 / 反证
事实与假设分开

05
建议
补充检查 / 延期
预览进入策略 API

方法
explain-idle-candidate
将已有规则与指标语义
组织成第一项只读 Skill

验证
先回放，再实时
闲置、checkpoint、缺数据
单位不一致、同名 Pod 重建

结果
解释准确，下一步明确
证据引用正确，缺信息能停止
减少用户与运维手工查询

当前 monitor 主要在删除后通知；先通过离线案例或 dry-run 补齐候选证据快照，再接实时解释。

展望 · 方案设想

以资源占用治理为切入口，验证平台契约

低活跃是调查线索；能否结束任务，还取决于任务类型、保护条件与用户约定。

01
候选
指标窗口
数据质量检查

02
预告
理由与证据
owner 可见

03
协商
租约 / 延期
保护与申诉

04
处置
动作前重查
按任务结束

05
确认
设备释放验证
审计与反馈

保护
缺数据不扩大动作
身份不明、信号陈旧或 checkpoint 中，暂停处置

对象
按 workload 管生命周期
区别裸 Pod、Job、Deployment 与交互会话

控制
先 dry-run 再启用
对照人工判断，小范围可恢复任务优先

展望 · 方案设想

建设路径：先打通最小用户闭环，再扩展生态

第一步服务真实任务：从准备环境到运行、诊断、保存结果和归还资源。

优先级	交付内容	验证门槛
P0 · 共同基础	身份/库存/版本/指标口径，需求与责任边界	知道谁、用什么、为何等待；建立基线
P1 · 用户闭环	环境/Job 契约、CLI 与只读诊断 Skill	代表任务能启动、解释问题并取回结果
P2 · 治理运营	队列/配额、租约、受控回收、升级与恢复	公平性、误回收、SLO 与演练可验证
P3 · 按需扩展	模型/推理集成、受控实验、Web/Bot	按用户收益与维护成本选择，不全量铺开

> 目标：用户容易使用，团队公平共享，平台可靠运营。

参与过的工作：监控与占用治理。以上完整生态为展望，建设节奏由需求、基线与验证结果决定。