

PR #54249 完整报告

vllm-project/vllm

[ROCm][CI] Fix test_ray_v2_executor

合并时间: 2026-08-29 06:11

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/54249>

执行摘要

- 一句话: 修复 ROCm/XPU 下 Ray 初始化 GPU 数量误设问题。
- 推荐动作: 值得快速阅读, 理解 Ray 初始化时 GPU 资源声明的机制。改动简单, 但涉及分布式执行的关键初始化路径, 对 ROCm/XPU 用户有益。

功能与动机

该问题源于 `test_ray_v2_executor` 中的 `test_colocated_tcpstore_ports_differ` 测试, 该测试在同一主机上创建两个并置的 `RayExecutorV2` 引擎。在 ROCm/XPU 平台上, `initialize_ray_cluster` 函数会尝试连接现有 Ray 实例, 若无则启动新实例, 并使用 `parallel_config.world_size` 作为 `num_gpus` 参数。由于每个引擎的 `world_size` 为 1, Ray 会错误地认为节点仅有 1 块 GPU, 导致第一个引擎占用唯一广告 GPU, 第二个引擎因找不到空闲 GPU 而永久等待。

实现拆解

1. 变更入口: 修改 `vllm/v1/executor/ray_utils.py` 中的 `initialize_ray_cluster` 函数。
2. 核心逻辑调整: 将启动新 Ray 实例时的 `num_gpus` 参数从 `parallel_config.world_size` 改为 `current_platform.device_count()`。这样 Ray 能识别节点实际拥有的 GPU 数量, 而不是每个引擎所需的 GPU 数。
3. 影响: 此改动仅影响 ROCm/XPU 平台启动新 Ray 实例时的 GPU 资源声明, 不影响 CUDA 平台 (其走 `else` 分支) 和已有 Ray 实例的连接。
4. 测试配套: 本 PR 未添加新的测试文件, 依赖现有的 `test_ray_v2_executor` 测试验证。

关键文件:

- `vllm/v1/executor/ray_utils.py` (模块 执行器; 类别 `source`; 类型 `core-logic`; 符号 `initialize_ray_cluster`): 修复 Ray 初始化时 GPU 数量设置错误的根本原因

关键符号: `initialize_ray_cluster`

关键源码片段

`vllm/v1/executor/ray_utils.py`

修复 Ray 初始化时 GPU 数量设置错误的根本原因

```
# vllm/v1/executor/ray_utils.py ( 片段 )
```

```
# 在 ROCm/XPU 平台，若没有现成 Ray 实例，则启动新实例。
# 原代码使用 world_size 作为 num_gpus，在单卡引擎场景下会导致 Ray 认为节点仅有 1 块 GPU。
# 修改为使用 device_count() 以声明节点实际 GPU 数，避免资源不足导致的等待。
ray.init(
    address=ray_address,
    num_gpus=current_platform.device_count(), # 关键修复
    runtime_env=parallel_config.ray_runtime_env,
)
```

评论区精华

无实质性技术讨论。Review 中由 AndreasKaratzas 批准，Claude bot 指出 fork PR 未自动审查。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。改动仅影响 ROCm/XPU 平台新启动 Ray 实例时的 num_gpus 参数。潜在风险是：如果节点 GPU 数量小于 world_size，Ray 可能报告错误，但现有代码已对 CUDA 平台有预检查，此改动未引入类似检查，可能导致不明确错误。
- 影响：影响范围限于 ROCm/XPU 平台的分布式执行路径。解决了测试挂起问题，提高了 CI 稳定性，也避免了在多 GPU 节点上运行单卡引擎时 Ray 资源利用不充分的问题。
- 风险标记：分布式初始化修复，测试依赖现有用例

关联脉络

- PR #50969 [BugFix] Bind RayExecutorV2 TCPStore before publishing its port: 同一文件 (ray_utils.py) 及 RayExecutorV2 初始化流程，相关分布式稳定性修复。