PR #1934 完整报告

THUDM/slime

Add GPU placement validation before starting rollout engines

合并时间：2026-05-25 15:39

原文链接：http://prhub.com.cn/THUDM/slime/pull/1934

执行摘要

- 一句话：新增 GPU 放置位置验证，修复启动时 IndexError
- 推荐动作：建议快速合入并同步关联 Issue #1896 关闭。该 PR 设计简洁，验证逻辑独立可测试，是良好的防御性编程例证，值得阅读。

功能与动机

修复 Issue #1896：多角色 megatron 配置重构（PR #1866）后，critic_only 测试因 placement group GPU 数量与 sglang 引擎配置不一致，在 start_engines 时触发 IndexError: list index out of range。

实现拆解

1. 新增验证函数：在 slime/ray/rollout_validation.py 创建 validate_server_group_gpu_indices()，接受 placement group 的 gpu_offset、num_gpu_per_engine、num_engines 等参数，计算所需 GPU 槽位并检查是否溢出可用 GPU 列表。
2. 注入调用点：在 slime/ray/rollout.py 的 ServerGroup.start_engines() 中，于索引 reordered_gpu_ids 之前调用验证函数，若配置不合法立即抛出 ValueError，附带工作类型、各配置项和修复提示。
3. 单元测试覆盖：新增 tests/test_rollout_validation.py，包含三个测试用例：有效配置不抛出异常、空引擎组跳过验证、无效配置验证错误消息包含所有关键字段。

关键文件：

- slime/ray/rollout_validation.py（模块 rollout；类别 source；类型 core-logic；符号 validate_server_group_gpu_indices）：新增验证函数的入口点，是本次变更的核心。
- slime/ray/rollout.py（模块 rollout；类别 source；类型 dependency-wiring）：修改了引擎启动入口，引入验证调用，是变更触发点。
- tests/test_rollout_validation.py（模块 测试；类别 test；类型 test-coverage；符号 test_validate_server_group_gpu_indices_accepts_valid_config, test_validate_server_group_gpu_indices_allows_empty_group, test_validate_server_group_gpu_indices_reports_config_context）：新增单元测试，覆盖验证函数的三个场景。

关键符号：validate_server_group_gpu_indices

关键源码片段

slime/ray/rollout_validation.py

新增验证函数的入口点，是本次变更的核心。

```
def validate_server_group_gpu_indices(
    *,
    worker_type: str,
    gpu_offset: int,
    num_gpus_per_engine: int,
    num_gpu_per_engine: int, # 单节点上每个引擎实际使用的 GPU 数
    num_engines: int,
    num_available_gpus: int, # placement group 中可用的 GPU ID 数量
    rollout_num_gpus: int,
    rollout_num_gpus_per_engine: int,
) -> None:
    # 空引擎组直接通过（例如 placeholder 类型）
    if num_engines == 0:
        return

    # 计算所需 GPU 槽位：偏移量 + 引擎数 × 每引擎 GPU 数
    required_gpu_slots = gpu_offset + num_engines * num_gpu_per_engine

    # 当所有参数合法且不超出可用 GPU 范围时，检查通过
    if gpu_offset >= 0 and num_gpu_per_engine > 0 and required_gpu_slots <= num_available_
    gpus:
        return

    # 否则抛出 ValueError，包含所有配置上下文以方便调试
    raise ValueError(
        "Invalid rollout server group GPU placement: "
        f"worker_type={worker_type}, "
        f"gpu_offset={gpu_offset}, "
        f"num_gpus_per_engine={num_gpus_per_engine}, "
        f"num_gpu_per_engine_on_node={num_gpu_per_engine}, "
        f"num_engines={num_engines}, "
        f"required_gpu_slots={required_gpu_slots}, "
        f"len(reordered_gpu_ids)={num_available_gpus}, "
        f"rollout_num_gpus={rollout_num_gpus}, "
        f"rollout_num_gpus_per_engine={rollout_num_gpus_per_engine}. "
        "Please align --rollout-num-gpus, --rollout-num-gpus-per-engine, "
        "and --sglang-config server_groups."
    )
```

tests/test_rollout_validation.py

新增单元测试，覆盖验证函数的三个场景。

```
import pytest
from slime.ray.rollout_validation import validate_server_group_gpu_indices

@pytest.mark.unit
def test_validate_server_group_gpu_indices_accepts_valid_config():
    # 有效配置：offset=2, 2 engines, 每引擎 1 GPU, 共 4 GPU，不越界
    validate_server_group_gpu_indices(
        worker_type="regular",
        gpu_offset=2,
        num_gpus_per_engine=1,
        num_gpu_per_engine=1,
        num_engines=2,
        num_available_gpus=4,
        rollout_num_gpus=4,
        rollout_num_gpus_per_engine=1,
    )

@pytest.mark.unit
def test_validate_server_group_gpu_indices_allows_empty_group():
    # 空引擎组（num_engines=0）应直接通过
    validate_server_group_gpu_indices(
        worker_type="placeholder",
        gpu_offset=4,
        num_gpus_per_engine=1,
        num_gpu_per_engine=1,
        num_engines=0,
        num_available_gpus=4,
        rollout_num_gpus=4,
        rollout_num_gpus_per_engine=1,
    )

@pytest.mark.unit
def test_validate_server_group_gpu_indices_reports_config_context():
    # 无效配置：offset=3, 1 engine, 每引擎 2 GPU, 共 4 GPU，需要 5 个槽位 > 4
    with pytest.raises(ValueError) as exc_info:
        validate_server_group_gpu_indices(
            worker_type="regular",
            gpu_offset=3,
            num_gpus_per_engine=2,
            num_gpu_per_engine=2,
            num_engines=1,
            num_available_gpus=4,
            rollout_num_gpus=4,
            rollout_num_gpus_per_engine=2,
        )
    # 验证错误消息中包含所有关键字段
    message = str(exc_info.value)
    assert "worker_type=regular" in message
    assert "gpu_offset=3" in message
```

```
assert "num_gpus_per_engine=2" in message
assert "num_engines=1" in message
assert "required_gpu_slots=5" in message
assert "len(reordered_gpu_ids)=4" in message
assert "rollout_num_gpus=4" in message
assert "rollout_num_gpus_per_engine=2" in message
```

评论区精华

未产生 review 评论讨论。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。仅添加前置检查，不改变引擎启动流程。但需注意当 num_gpu_per_engine 计算为 min(self.num_gpus_per_engine, args.num_gpus_per_node) 时，验证使用裁剪后的值而非原始值，若与后续实际索引逻辑不一致可能导致误报。
- 影响：对用户：配置错误时更早获得清晰错误信息，提升调试效率。对系统：无性能影响，检查计算量极小。对团队：减少此类回归的排查成本。
- 风险标记：回归修复，核心路径变更，配置耦合风险

关联脉络

- PR #1866 Rename critic config to megatron config: 本次修复的 bug 正是由该 PR 引入，多角色 megatron 配置导致 sglang 引擎 GPU 计数偏移。
- PR #1896 [Bug] test_qwen2.5_0.5B_ppo_critic_only_short.py fails with IndexError: 该 issue 报告了 bug 现象及根因分析，本 PR 直接修复该 issue。