

PR #2057 完整报告

THUDM/slime

Allow zero-GPU rollout router startup

合并时间: 2026-06-12 12:10

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2057>

执行摘要

- 一句话: 允许零 GPU 启动仅路由的 rollout 服务
- 推荐动作: 此 PR 值得关注, 尤其是对于计划使用自定义 rollout 引擎的团队, 零 GPU 模式提供了灵活部署选项。设计简洁, 测试覆盖良好, 建议在相关文档中强调零 GPU 模式的外部引擎依赖。

功能与动机

PR body 指出需要将 `--rollout-num-gpus 0` 作为一种显式的仅路由 rollout 模式, 包括在 `colocate` 模式下。允许构建一个空的默认 SGLang 部署, 使得自定义 rollout 函数仍可使用路由参数, 而不启动本地 SGLang 服务器。当无可更新的 rollout 引擎运行时跳过权重同步。

实现拆解

1. 参数定义与验证 (`slime/utils/arguments.py`): 修改 `--rollout-num-gpus` 的帮助文本, 明确 0 表示仅路由模式; 在 `slime_validate_args` 中调整 `debug_rollout_only` 和 `colocate` 分支: 当 `rollout_num_gpus == 0` 时不再自动覆盖为 actor 大小, 而是保留 0 并置零 actor 节点数; `colocate` 下当值为 0 时仅记录日志。
2. 配置解析 (`slime/ray/rollout.py`): 在 `_resolve_sglang_config` 中增加提前返回, `rollout_num_gpus == 0` 时返回一个空 `SglangConfig` (含无 `server_groups` 的默认模型), 避免引擎启动。
3. 权重更新跳过 (`slime/backends/megatron_utils/actor.py`): `update_weights` 方法在获取引擎列表后检查是否为空且无需重连, 若是则跳过权重更新并记录日志。
4. Placement group 布局 (`slime/ray/placement_group.py`): `colocate` 模式下使用 `max(actor_num_gpus, args.rollout_num_gpus)` 确保当 `rollout_num_gpus=0` 时仍分配 actor 所需 GPU。
5. 测试覆盖: 在 `test_megatron_argument_validation.py` 新增 `make_slime_validate_args` 辅助和三个测试用例验证零 GPU 在 `colocate`/ 非 `colocate` 下的保留; 在 `test_sglang_config.py` 新增 `TestZeroGpuRolloutConfig` 类测试空配置解析和启动行为; 在 `test_placement_group.py` 适配布局变更。
6. 文档更新: 四个 `get_started` 文档 (中英文 `quick_start` 和 `usage`) 更新描述, 提及零 GPU 选项。

关键文件:

- slime/utils/arguments.py (模块 参数配置; 类别 source; 类型 core-logic) : 定义了 --rollout-num-gpus 参数的帮助文本, 并修改了 slime_validate_args 函数以正确处理零 GPU 模式, 包括在 colocate 和 debug_rollout_only 逻辑中。
- slime/ray/rollout.py (模块 路由引擎; 类别 source; 类型 core-logic) : 在 _resolve_sglang_config 中添加提前返回, 当 rollout_num_gpus=0 时返回空配置, 避免启动 SGLang 引擎。
- slime/backends/megatron_utils/actor.py (模块 权重更新; 类别 source; 类型 core-logic) : update_weights 方法增加跳过逻辑, 当无可更新的 rollout 引擎时直接返回。
- slime/ray/placement_group.py (模块 资源调度; 类别 source; 类型 core-logic) : colocate 模式下使用 max(actor, rollout) 防止零 GPU 时返回负数。
- tests/test_megatron_argument_validation.py (模块 参数验证测试; 类别 test; 类型 test-coverage; 符号 make_slime_validate_args, test_slime_validate_args_preserves_zero_rollout_gpus_under_colocate, test_slime_validate_args_preserves_larger_rollout_gpus_under_colocate, test_slime_validate_args_preserves_zero_rollout_gpus_without_colocate) : 新增 make_slime_validate_args 辅助函数和三个测试用例, 验证零 GPU 参数在 colocate 和非 colocate 下被保留。
- tests/utils/test_sglang_config.py (模块 配置解析测试; 类别 test; 类型 test-coverage; 符号 test_config_allows_model_with_no_server_groups, TestZeroGpuRolloutConfig, test_resolve_default_zero_gpu_config_has_no_server_groups, test_zero_gpu_config_takes_precedence_over_preload_num_servers) : 新增 TestZeroGpuRolloutConfig 测试类, 覆盖配置解析和启动行为。
- tests/test_placement_group.py (模块 资源组测试; 类别 test; 类型 test-coverage) : 调整测试以适配 colocate 模式下的 max 逻辑。

关键符号: slime.utils.arguments.slime_validate_args, slime.ray.rollout._resolve_sglang_config, slime.backends.megatron_utils.actor.MegatronActor.update_weights, slime.ray.placement_group._get_placement_group_layout

关键源码片段

slime/utils/arguments.py

定义了 --rollout-num-gpus 参数的帮助文本, 并修改了 slime_validate_args 函数以正确处理零 GPU 模式, 包括在 colocate 和 debug_rollout_only 逻辑中。

slime/utils/arguments.py - slime_validate_args 中零 GPU 相关分支

```
if args.debug_rollout_only:
    # debug_rollout_only 模式下, 仅在未显式设置 rollout_num_gpus 时自动计算
    if args.colocate and args.rollout_num_gpus is None:
        args.rollout_num_gpus = args.actor_num_gpus_per_node * args.actor_num_nodes
    elif args.rollout_num_gpus == 0:
        # 显式零 GPU: 置零 actor 节点, 避免后续创建引擎
        args.actor_num_gpus_per_node = 0
        args.actor_num_nodes = 0
```

```

else:
    args.actor_num_gpus_per_node = min(8, args.rollout_num_gpus)
    args.actor_num_nodes = args.rollout_num_gpus // args.actor_num_gpus_per_node
args.colocate = False
args.offload_train = args.offload_rollout = False

# ... 后续代码

if args.colocate:
    if args.offload_train is None:
        args.offload_train = True
    if args.offload_rollout is None:
        args.offload_rollout = True
    # 零 GPU 模式：保留 0 并记录日志，不自动覆盖
    if args.rollout_num_gpus is None:
        args.rollout_num_gpus = args.actor_num_gpus_per_node * args.actor_num_nodes
    elif args.rollout_num_gpus == 0:
        logger.info("rollout_num_gpus is 0 under colocate; no local SGLang engines will be
        launched.")

```

slime/ray/rollout.py

在 `_resolve_sglang_config` 中添加提前返回，当 `rollout_num_gpus=0` 时返回空配置，避免启动 SGLang 引擎。

slime/ray/rollout.py - `_resolve_sglang_config` 零 GPU 分支

```

def _resolve_sglang_config(args) -> SglangConfig:
    # 零 GPU 模式：直接返回空配置，避免底层创建任何 SGLang 服务器
    if args.rollout_num_gpus == 0:
        logger.info("Building empty SglangConfig for zero-GPU rollout mode.")
        return SglangConfig(models=[ModelConfig(name="default", server_groups=[])])

    if args.prefill_num_servers is not None:
        return SglangConfig.from_prefill_num_servers(args)
    # ... 原有逻辑

```

评论区精华

此 PR 无 review 讨论，由作者直接合并。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 零 GPU 配置误用：用户设置 `--rollout-num-gpus 0` 后若未提供外部引擎（`--rollout-external-engine-addr`s 或自定义 rollout 函数），则推理服务不可用，可能导致运行时错误。

- 权重同步跳过风险: actor.py 中跳过权重更新时仅记录日志, 用户可能误以为权重已同步, 需确保文档明确说明。
- 配置优先级冲突: rollout.py 中零 GPU 检测优先于 prefill_num_servers 检查, 可能改变原有默认行为, 需通过测试验证。
- Placement group 资源计算: max(actor, rollout) 当 rollout=0 时分配 actor 所需 GPU, 与无 colocate 行为一致, 风险可控。
- 影响:
 - 用户: 可以使用零 GPU 部署路由器, 适合仅使用自定义 rollout 函数的场景, 降低资源需求, 但需自己管理推理引擎。
 - 系统: 新代码路径经过充分测试, 风险较低。
 - 团队: 维护新的逻辑分支, 需要确保与其他配置 (如 prefill_num_servers、colocate) 协调, 未来扩展时注意退路。
 - 风险标记: 零 GPU 配置误用, 权重同步跳过, 配置优先级变更

关联脉络

- PR #2050 Set RAY_USE_UVLOOP=0 for Ray actors: 同为 rollout 相关改进, 修改了 rollout.py 和 placement_group.py。
- PR #2027 Only upload per sample stats to wandb: 修改了 rollout.py 和 wandb 相关, 涉及 rollout 配置调整。