

PR #2012 完整报告

radixark/miles

router: enable dp-aware routing under dp-attention

合并时间: 2026-07-31 14:04

原文链接: <http://prhub.com.cn/radixark/miles/pull/2012>

执行摘要

- 一句话: 开启 dp-attention 时自动启用 dp-aware 路由
- 推荐动作: PR 虽然只有 3 行源码 + 28 行测试, 但值得精读: PR body 提供了完整的生产故障分析 (GLM-5.2 744B 上 dp rank 排队与超时数据), 展示了如何把线上指标转化为一行默认行为修改; 默认开启而非新增开关的设计取舍也值得借鉴。建议重点看 `validate_args` 的参数编排位置与测试中手动补齐 `RouterArgs` 字段的做法。

功能与动机

PR body 指出 `--enable-dp-attention` 会把 engine 拆成 dp rank, 但 router 只知道 engine, 负载均衡停在 engine 边界, `sglang` 内部 `dispatch` 在长粘性 session 下会把请求集中到单个 rank。实测 GLM-5.2 744B (16x GB300、8 engines x dp 4、session server 使用 `manual + min_load`) 中, `dp_rank=2` 排队 11 个请求而其余 3 个 rank 完全空闲, 2232 个 rollout episode 失败里 2194 个是超时, step time 约为 TP-only 配置的 2 倍。作者因此决定开启 `sgl-router` 已有的 `--router-dp-aware`, 并认为该 flag 没有 dp-attention 时无意义、dp-attention 部署都需要它, 所以直接设为默认行为而不是新增开关。

实现拆解

1. 变更入口: 在 `miles/backends/sglang_utils/arguments.py` 的 `validate_args` 中, 紧跟 `sglang_dp_size > 1` 的一致性断言之后, 新增 `if args.sglang_enable_dp_attention:`
`args.router_dp_aware = True`。这里不是 `add_sglang_arguments` 的参数注册处, 因为 `router_dp_aware` 由 `RouterArgs.add_cli_args(..., use_router_prefix=True)` 注册, `validate_args` 负责把所有来源的参数统一整理成最终生效值, 放在这里可以保证 CLI、YAML 配置、默认值三种来源都吃到该规则。
2. 测试配套: 新增 `tests/fast/backends/sglang_utils/test_router_dp_aware.py`, 先注册到 `stage-a-cpu CPU CI`, 再用 `_args` 辅助函数组装参数: 调用 `add_sglang_arguments` 建 `parser`、手动补上 `RouterArgs` 注册的 `router_assignment_mode` 与 `router_dp_aware` 默认值、最后执行 `validate_args`。两个用例分别验证开启 dp-attention 后 `router_dp_aware` 变为 `True`, 以及未开启时保持 `False`。
3. 设计取舍: 没有新增 CLI 开关, 而是让 dp-attention 自动带出 dp-aware routing。这样避免用户漏配导致负载倾斜复现, 同时因为该 flag 对非 dp-attention 部署无意义, 不会被误伤。测试文件同时覆盖了反例 (不开启 dp-attention 不改变值), 防止将来有人把默认值逻辑改坏。

关键文件：

- miles/backends/sglang_utils/arguments.py（模块 参数校验；类别 source；类型 core-logic；符号 validate_args）：核心变更文件：validate_args 在 dp-attention 开启时强制 router_dp_aware=True，是负载均衡粒度调整的落点。
- tests/fast/backends/sglang_utils/test_router_dp_aware.py（模块 路由测试；类别 test；类型 test-coverage；符号 _args, test_dp_attention_turns_on_router_dp_aware, test_router_dp_aware_untouched_without_dp_attention）：新增测试覆盖 dp-attention 开启 / 关闭两种路径，注册到 stage-a-cpu CI，防止默认行为回归。

关键符号：validate_args, _args, test_dp_attention_turns_on_router_dp_aware, test_router_dp_aware_untouched_without_dp_attention

关键源码片段

miles/backends/sglang_utils/arguments.py

核心变更文件：validate_args 在 dp-attention 开启时强制 router_dp_aware=True，是负载均衡粒度调整的落点。

```
def validate_args(args):
    # 让 sglang tp size 默认等于 rollout 每 engine 的 GPU 数
    args.sglang_tp_size = args.rollout_num_gpus_per_engine

    if args.true_on_policy_mode:
        args.sglang_enable_deterministic_inference = True

    if getattr(args, 'recompute_logprobs_via_prefill', False):
        args.sglang_enable_prefill_only_deterministic_inference = True
        args.sglang_enable_deterministic_inference = True

    # dp-attention 必须与 dp size 一起使用，这里做一致性校验
    if args.sglang_dp_size > 1:
        assert args.sglang_enable_dp_attention

    # 核心变更：开启 dp-attention 时，让 router 感知 dp rank,
    # 否则 router 的负载均衡只到 engine 粒度，sglang 内部 dispatch
    # 会把长 session 请求集中到某个 rank，造成排队与超时
    if args.sglang_enable_dp_attention:
        args.router_dp_aware = True

    # session server 下默认 manual + min_load 路由，避免随机分配
    if args.sglang_router_policy is None and args.use_session_server:
        args.sglang_router_policy = 'manual'
        if args.router_assignment_mode == 'random':
            args.router_assignment_mode = 'min_load'

    # IPv6 地址需要包装后 router 端才能正确解析
    if getattr(args, 'sglang_router_ip', None):
        args.sglang_router_ip = _wrap_ipv6(args.sglang_router_ip)
```

tests/fast/backends/sglang_utils/test_router_dp_aware.py

新增测试覆盖 dp-attention 开启 / 关闭两种路径，注册到 stage-a-cpu CI，防止默认行为回归。

```
from tests.ci.ci_register import register_cpu_ci

# 注册到 stage-a-cpu CPU CI，预估耗时 20 秒
register_cpu_ci(est_time=20, suite='stage-a-cpu', labels=[])

import argparse

from miles.backends.sglang_utils.arguments import add_sglang_arguments, validate_args

def _args(argv):
    # add_sglang_arguments 不包含 RouterArgs 的 CLI，因此这里手动补上默认值
    parser = add_sglang_arguments(argparse.ArgumentParser())
    args = parser.parse_args(argv)
    args.rollout_num_gpus_per_engine = 4
    args.true_on_policy_mode = False
    args.use_session_server = False
    # 以下两个字段由 RouterArgs.add_cli_args 注册，测试里直接设置
    args.router_assignment_mode = 'random'
    args.router_dp_aware = False
    validate_args(args)
    return args

def test_dp_attention_turns_on_router_dp_aware():
    # 开启 dp-attention 后，validate_args 应自动开启 dp-aware 路由
    assert _args(['--sglang-enable-dp-attention', '--sglang-dp-size', '4']).router_dp_aware is True

def test_router_dp_aware_untouched_without_dp_attention():
    # 未开启 dp-attention 时，router_dp_aware 保持默认 False
    assert _args([]).router_dp_aware is False
```

评论区精华

审阅人 Zhichenzzz 给出 APPROVED，唯一意见是“lgtn! just fix the failed ci args”，提示某个 CI 参数需要修复，但没有在代码行内展开讨论。结合测试实现，这最可能指向 _args 中需要手动补齐 RouterArgs 注册的字段；最终提交已包含完整补齐并通过 CI，未发现遗留技术争议。Issue 评论区只有 Gemini Code Assist 的服务下线通知，与本次变更无关。

- CI 参数失败提示 (testing): PR 已合并，最终提交包含完整的参数补齐实现，CI 相关要求已满足；没有遗留的公开技术争论。

风险与影响

- 风险：

1. 默认行为变更：所有启用 `--sglang-enable-dp-attention` 的部署都会强制 `router_dp_aware=True`，没有提供覆盖开关，未来若出现依赖 engine 级路由的场景将无法关闭，当前可接受但建议文档注明。
2. 断言联动：`validate_args` 已有 `sglang_dp_size > 1` 时必须开启 `dp-attention` 的断言，因此 `dp_size>1` 必然触发新行为；`dp_size==1` 但开启 `dp-attention` 时也会触发，不过单 rank 场景影响有限。
3. 测试覆盖边界：新增测试只到参数层，不验证 `sgl-router` 真正按 rank 注册 worker 的行为，若 `RouterArgs.from_cli_args` 转发链路变化，回归无法被这两个用例捕获。
4. 兼容性：无 schema、CLI、部署配置变更，非 `dp-attention` 路径完全不受影响。
 - 影响：对部署系统：启用 `dp-attention` 的 RL 训练与推理任务将获得 rank 粒度负载均衡，直接缓解长 session 请求集中到单一 rank 导致的排队、超时与 2 倍 step time 问题。
 - 对非 `dp-attention` 部署无影响。对团队：这是无配置默认行为，运维无需学习新参数；新增 `stage-a-cpu CPU CI` 用例保证该默认行为不回归。整体影响面集中在 `sglang` 后端参数校验路径与 session server 路由场景。
 - 风险标记：默认行为变更，无覆盖开关，测试仅到参数层

关联脉络

- PR #2028 session: collect speculative-decoding counters: 同仓库近期唯一可对照 PR，同属 session/rollout 路径的稳定性优化：2028 补齐 session 路径的 spec 计数器，本 PR 修 session server 下 dp 路由倾斜；文件不重叠，但都在完善 agentic RL rollout 的生产可观测性与可用性。