

# PR #46116 完整报告

vllm-project/vllm

[Core][KV-transfer] MoRIIO: heterogeneous TP<->DP prefill/decode read routing

合并时间: 2026-07-27 10:10

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/46116>

## 执行摘要

- 一句话: 为 MoRIIO 添加异构 TP/DP 并行路由支持
- 推荐动作: 值得精读。该 PR 展示了在异构并行环境下如何优雅地解决 KV 传输路由问题, 特别是死锁预防和负载均衡的设计。对于了解 MoRIIO 内部机制和分布式推断性能优化的开发者而言, 其中的 `_eager_handshake_all_dp_ranks` 和 `_next_flex_tp_rank` 实现具有较高学习价值。建议关注配套的 RFC #46107 和相关 PR #46115、#47495 以获取完整上下文。

## 功能与动机

参考 RFC #46107, 当前 MoRIIO 假设预填充和解码引擎使用相同的并行度布局。然而, 对于如 DeepSeek-R1 的工作负载, 预填充受延迟限制最佳为 TP, 解码受吞吐限制最佳为 DP/EP。异构并行能够在高并发下实现更高的吞吐量和更低的 TTFT。性能数据显示, 在 512 并发请求时, 2P\_TP8:1D\_DP8EP 配置相比同构 TP8 在吞吐量和 TTFT 上均有显著提升。

## 实现拆解

该 PR 的实现分为三个主要步骤:

1. 修复预填充 DP 排名硬编码: 在 `moriio_connector.py` 中, 修改 `_resolve_read_source` 和 `_moriio_handshake` 函数, 使其使用来自 proxy 传递的 `remote_dp_rank` 字段 (存储于 `ReqMeta.remote_dp_rank`), 而非始终使用 DP0。在 `moriio_common.py` 的 `ReqMeta` 数据类中新增 `remote_dp_rank: int = 0` 字段, 并在 `add_new_req` 中从 `kv_transfer_params` 解析该值。默认值为 0, 保持与原有对称单 DP 行为的字节级兼容。
2. 引入急切前向握手机制: 新增 `_eager_handshake_all_dp_ranks` 方法, 在每次解码前向传播开始时, 对所有可能读取的远程预填充 DP 排名执行一次握手机制。所有本地 TP worker 通过 CPU `all_reduce(MIN)` 屏障同步, 确保所有 worker 在进入前向步骤前完成握手, 防止因懒惰握手机制导致部分 worker 进入前向集合操作而另一部分 worker 仍阻塞在握手 `recv()` 的死锁情况。使用 `_eager_handshaked_engines` 集合避免重复握手。
3. 灵活的预填充 TP 读取 (镜像配置): 新增 `_next_flex_tp_rank` 方法, 在 TP 预填充 + MLA + DP 解码 (即镜像) 配置下, 解码 worker 可以轮询读取预填充 TP 0..N-1 中的任意一个, 而非全部固定读取 TP0。该选择基于严格的确定性轮询算法, 按解码 worker 的 DP 排名偏移, 确保均匀的 NIC 负载分布。轮询行为仅在满足所有条件 (`world_size==1`, `use_mla`, `remote_dp_size==1`, `remote_tp_size>1`) 时启用, 否则回退到原有固定映射。

配套测试：新增 `tests/v1/kv_connector/unit/test_moriio_routing_fairness.py`，轻量级（无需 GPU/ROCm/mori），直接驱动真实的 `_resolve_read_source` 和 `_next_flex_tp_rank` 函数，覆盖 RFC 中所有同构和混合配置，验证硬件公平的预填充读取分布。

关键文件：

- `vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py`（模块 路由逻辑；类别 source；类型 core-logic；符号 `get_engine_name_with_dp_tp`, `_eager_handshake_all_dp_ranks`, `_next_flex_tp_rank`, `_resolve_read_source`）：核心实现文件，包含了所有路由逻辑的变更：`_resolve_read_source` 新增灵活 TP 读取路径，`_eager_handshake_all_dp_ranks` 实现死锁修复，`_next_flex_tp_rank` 实现轮询负载均衡，以及 `get_engine_name_with_dp_tp` 新增 (dp, tp) 会话键。
- `tests/v1/kv_connector/unit/test_moriio_routing_fairness.py`（模块 测试；类别 test；类型 test-coverage；符号 `PDConfig`, `p_dp`, `p_tp`, `d_dp`）：新增的轻量级测试文件，验证路由逻辑的公平性。通过构造 `MoRIIOConnectorWorker` 实例直接调用真实路由函数，覆盖 RFC 中的五种配置，确保对称 TP、灵活 TP、所有者 DP 三种行为均正确，并且轮询确定性、均匀性和交错性得到断言。
- `vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_common.py`（模块 数据模型；类别 source；类型 data-contract）：数据模型变更：在 `ReqMeta` 中添加 `remote_dp_rank` 字段，用于存储当前请求所属的预填充 DP 排名，并修改 `add_new_req` 从参数中获取该字段。默认值 0 确保向后兼容。

关键符号：`get_engine_name_with_dp_tp`, `_eager_handshake_all_dp_ranks`, `_next_flex_tp_rank`, `_resolve_read_source`

## 关键源码片段

### `tests/v1/kv_connector/unit/test_moriio_routing_fairness.py`

新增的轻量级测试文件，验证路由逻辑的公平性。通过构造 `MoRIIOConnectorWorker` 实例直接调用真实路由函数，覆盖 RFC 中的五种配置，确保对称 TP、灵活 TP、所有者 DP 三种行为均正确，并且轮询确定性、均匀性和交错性得到断言。

```
# 轻量级单元测试核心：构造虚拟 worker 并调用真实路由函数
# 模拟 proxy 的公平请求流，验证连接器不会引入瓶颈
```

```
# PDConfig 数据类描述预填充 / 解码模式的维度
```

```
@dataclass(frozen=True)
```

```
class PDConfig:
```

```
    name: str
```

```
    p_mode: str # "TP8" 或 "DP8EP"
```

```
    d_mode: str
```

```
    @property
```

```
    def p_dp(self) -> int: # 预填充 DP 大小
```

```
        return MODE_DIMS[self.p_mode][0]
```

```
    @property
```

```
    def p_tp(self) -> int: # 预填充 TP 大小
```

```

        return MODE_DIMS[self.p_mode][1]
    @property
    def d_dp(self) -> int: # 解码 DP 大小
        return MODE_DIMS[self.d_mode][0]
    @property
    def d_tp(self) -> int: # 解码 TP 大小
        return MODE_DIMS[self.d_mode][1]
    @property
    def n_prefill_gpus(self) -> int:
        return self.p_dp * self.p_tp

# 覆盖 RFC 中的五种 P/D 配置
CONFIGS = [
    PDConfig("1P_TP8:1D_TP8", "TP8", "TP8"),
    PDConfig("2P_TP8:1D_DP8EP", "TP8", "DP8EP"),
    PDConfig("2P_TP8:2D_TP8", "TP8", "TP8"),
    PDConfig("2P_DP8EP:3D_DP8EP", "DP8EP", "DP8EP"),
    PDConfig("2P_DP8EP:4D_TP8", "DP8EP", "TP8"),
]

# 核心测试函数：根据配置生成公平请求流，收集每个预填充 GPU 的命中计数
def prefill_target_multiset(cfg: PDConfig, rounds: int) -> Counter:
    workers = build_decode_workers(cfg)
    hits: Counter = Counter()
    for _ in range(rounds):
        for owner_dp in range(cfg.p_dp): # 模拟 proxy 公平分配
            for worker in workers:
                meta = make_meta(
                    p_tp=cfg.p_tp, p_dp=cfg.p_dp, remote_dp_rank=owner_dp
                )
                chosen_tp, _flexible = worker._resolve_read_source(meta)
                # 计算对应的全局 GPU 索引
                hits[(owner_dp, chosen_tp)] += 1
    return hits

```

## 评论区精华

由于该 PR 的 review 评论为空，主要讨论发生在 PR 常规评论中。PR 作者 edwinlim0919 多次请求 review，并提供了在 MI300X 上使用 DeepSeek-R1 模型的端到端验证结果，包括准确率 (gsm8k strict-match 0.9613) 和吞吐量数据。reviewer tanpinsiang 在最后一个评论中提供了独立的测试验证，使用 MiniMax-M3-MXFP8 模型在 TP8 同构配置下进行回归测试，确认 73 个单元测试全部通过，且 router/prefill/decode 健康检查均正常。没有发现设计争议或未解决的问题。

- 测试验证结果 (testing): 验证了异构路由逻辑在回归测试中无退化。

## 风险与影响

- 风险:

1. 兼容性风险：新增的 `remote_dp_rank` 字段默认值为 0，且所有新逻辑由条件门控（`flexible_gate`），对称 TP 和所有者 DP 行为与之前字节级一致，无需担心回归。
2. 死锁风险：急切握手机制使用了 CPU 同步屏障，若某个 worker 的握手失败（如网络分区），所有 worker 将统一失败快速，避免长时间挂起。但增加了前向传播前的延迟，不过由于握手仅执行一次，后续命中缓存，对持续运行的影响可忽略。
3. 性能风险：轮询负载均衡引入了确定性计算开销，但相对于 RDMA 传输延迟极低。
4. 测试覆盖：轻量级单元测试覆盖了所有路由逻辑，但缺乏集成测试验证实际硬件上的端到端行为（已在作者独立测试中覆盖）。- 影响：用户影响：对于使用 MoRADIO 进行 KV 传输并希望采用异构预填充 / 解码并行度的用户，该 PR 提供了关键支持，允许配置如 TP8 预填充 + DP8EP 解码的组合。对于不改变现有同构配置的用户，行为完全不变。系统影响：需要对路由器和 proxy 进行相应修改（记录在关联 PR #46115 和 #47495 中）以传递正确的 `remote_dp_rank`。该 PR 仅处理连接器（decode 端）的读取路由，proxy 端的选择不在本 PR 范围内。团队影响：需要维护两个互相依赖的相关 PR。- 风险标记：核心路径变更，兼容性风险，死锁预防机制，依赖关联 PR

## 关联脉络

- PR #46107 [RFC]: Heterogeneous TP<->DP parallelism across the P/D disaggregation boundary (MoRADIO): 本 PR 是 RFC #46107 的实现之一，提供了核心路由变更。
- PR #46115 [Bugfix][KV-transfer] MoRADIO proxy fix for heterogeneous DP: 关联的 proxy 修改，负责传递正确的 `remote_dp_rank`，与本 PR 的路由变更配合使用。
- PR #47495 [Bugfix][KV-transfer] MoRADIO backpressure for heterogeneous DP: 关联的反压机制 PR，确保异构 DP 下 KV 传输的稳定性。