

PR #46332 完整报告

vllm-project/vllm

[ROCm][P/D] Support MoRIIO heterogeneous TP fan-in

合并时间: 2026-06-23 18:33

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/46332>

执行摘要

- 一句话: 支持 MoRIIO 异构 TP 的 ACK fan-in 与 rank 映射
- 推荐动作: 值得精读, 特别是 `get_moriiio_remote_tp_rank` 的映射策略和 ACK fan-in 的设计。该设计为不同推理阶段的 TP 异构提供了可扩展的框架。测试覆盖全面, 建议参考测试用例理解边界条件。

功能与动机

在分布式推理中, `prefill` 和 `decode` 阶段对计算资源的需求不同, 使用不同的 TP 大小可以提高资源利用率。该 PR 在 #46290 的基础上, 扩展 MoRIIO 以支持异构 TP 配置, 使 `prefill` 节点和 `decode` 节点可以采用不同的 TP 大小进行 KV 传输。

实现拆解

1. 定义结构化 ACK 类型: 在 `moriiio_common.py` 中添加 `MoRIIOTransferAck` `NamedTuple`, 包含 `transfer_id` 和 `consumer_tp_size`, 替代原来纯字符串传输。
2. 实现远程 TP rank 映射: 在 `moriiio_connector.py` 中添加 `get_moriiio_remote_tp_rank` 函数, 根据本地 TP rank、本地 TP 大小和远程 TP 大小, 通过整除或乘法映射到远程 TP rank, 确保通讯时使用正确的端口。
3. 添加 KV head 复制验证守卫: `validate_moriiio_heterogeneous_tp_kv_heads` 函数检查是否满足复制 KV head 的要求 (或 MLA), 若不满足则抛出 `NotImplementedError`。
4. 实现 ACK fan-in 计数: `get_moriiio_expected_ack_count` 计算 `producer` 端期望的 ACK 数量, `resolve_moriiio_transfer_ack` 在接收到 ACK 时计数, 达到预期后标记传输完成。
5. 修改消息发送和接收: 在 `moriiio_engine.py` 中, `send_notify` 增加 `message_fields` 参数以携带 `consumer_tp_size`; `_handle_release_message` 解析 `consumer_tp_size` 并构建 `MoRIIOTransferAck`; `_mark_request_done` 和 `_handle_completion_message` 改为使用 `MoRIIOTransferAck`。
6. 补充单元测试: 新增 `test_moriiio_tp_ack.py` 覆盖 rank 映射、head 守卫、ACK 计数、反向兼容等场景; 调整已有测试用例适配新接口。

关键文件:

- `vllm/distributed/kv_transfer/kv_connector/v1/moriiio/moriiio_connector.py` (模块 连接层; 类别 `source`; 类型 `core-logic`; 符号 `get_moriiio_remote_tp_rank`, `validate_moriiio_heterogeneous_tp_kv_heads`, `get_moriiio_expected_ack_count`,

resolve_moriio_transfer_ack)：核心文件，新增远程 TP rank 映射、KV head 验证守卫、ACK fan-in 计数和 ACK 解析函数，是所有新功能的主入口。

- tests/v1/kv_connector/unit/test_moriio_tp_ack.py (模块 单元测试；类别 test；类型 test-coverage；符号 test_remote_tp_rank_same_tp_maps_to_self, test_remote_tp_rank_p4_d8_floor_maps_decode_to_prefill, test_remote_tp_rank_p8_d4_maps_to_first_prefill_rank_per_pair, test_remote_tp_rank_invalid_non_multiple_tp_raises)：新测试文件，全面覆盖 rank 映射、head 守卫、ACK 计数和向后兼容场景，共 310 行，是验证新功能正确性的关键。
- vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_common.py (模块 公共定义；类别 source；类型 data-contract；符号 MoRIIOTransferAck)：新增 MoRIIOTransferAck NamedTuple 类型，为结构化 ACK 提供数据类型基础，影响所有消息处理路径。
- vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_engine.py (模块 引擎层；类别 source；类型 core-logic；符号 _mark_request_done, _handle_release_message, _handle_completion_message, send_notify)：修改完成通知路径，使用 MoRIIOTransferAck 包装消息，并调整 send_notify 以传递额外字段，是消息收发的核心引擎。
- tests/v1/kv_connector/unit/test_moriio_kv_layout.py (模块 布局测试；类别 test；类型 test-coverage；符号 send_notify)：适配新接口，验证 send_notify 扩展和 MoRIIOTransferAck 在消息路由中的行为。
- tests/v1/kv_connector/unit/test_moriio_connector.py (模块 连接器测试；类别 test；类型 test-coverage；符号 send_notify)：适配 send_notify 接口变化，确保 connector 测试通过。

关键符号：get_moriio_remote_tp_rank, validate_moriio_heterogeneous_tp_kv_heads, get_moriio_expected_ack_count, resolve_moriio_transfer_ack, MoRIIOTransferAck, _handle_release_message, send_notify, _mark_request_done

关键源码片段

vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py

核心文件，新增远程 TP rank 映射、KV head 验证守卫、ACK fan-in 计数和 ACK 解析函数，是所有新功能的主入口。

```
# MoRIIO 异构 TP 远程 rank 映射函数
# 当本地 TP 大小与远程 TP 大小不同时，将本地 TP rank 映射到远程 TP rank
# 支持两种场景：本地 TP 较小（prefill 到 decode）和本地 TP 较大（decode 到 prefill）
def get_moriio_remote_tp_rank(
    local_tp_rank: int, local_tp_size: int, remote_tp_size: int
) -> int:
    if local_tp_size <= 0 or remote_tp_size <= 0:
        raise ValueError("TP sizes must be positive")
    if local_tp_rank < 0 or local_tp_rank >= local_tp_size:
        raise ValueError(
            f"local_tp_rank {local_tp_rank} must be in [0, {local_tp_size})"
```

```

    )
    # 同构 TP: 直接返回本地 rank
    if remote_tp_size == local_tp_size:
        return local_tp_rank
    # 远程 TP 较大: 每个本地 rank 负责多个远程 rank, 均匀扩展
    if remote_tp_size > local_tp_size:
        if remote_tp_size % local_tp_size != 0:
            raise ValueError(
                f"remote tp_size {remote_tp_size} must be a multiple of local "
                f"tp_size {local_tp_size} for heterogeneous-TP P/D"
            )
        # 例: P4 -> D8, 本地 rank 0 -> 远程 rank 0, rank 1 -> 远程 rank 2
        return local_tp_rank * (remote_tp_size // local_tp_size)
    # 远程 TP 较小: 多个本地 rank 映射到同一个远程 rank
    if local_tp_size % remote_tp_size != 0:
        raise ValueError(
            f"local tp_size {local_tp_size} must be a multiple of remote "
            f"tp_size {remote_tp_size} for heterogeneous-TP P/D"
        )
    # 例: D8 -> P4, 本地 rank 0/1 -> 远程 rank 0, rank 2/3 -> 远程 rank 1
    return local_tp_rank // (local_tp_size // remote_tp_size)

```

tests/v1/kv_connector/unit/test_moriio_tp_ack.py

新测试文件, 全面覆盖 rank 映射、head 守卫、ACK 计数和向后兼容场景, 共 310 行, 是验证新功能正确性的关键。

```

# 测试远程 TP rank 映射: 同构 TP 时 rank 不变
def test_remote_tp_rank_same_tp_maps_to_self():
    assert [get_moriio_remote_tp_rank(rank, 4, 4) for rank in range(4)] == [0, 1, 2, 3]

# 测试 P4/D8 映射: 本地 TP=8 远程 TP=4, 每两个本地 rank 对应一个远程 rank
def test_remote_tp_rank_p4_d8_floor_maps_decode_to_prefill():
    assert [get_moriio_remote_tp_rank(rank, 8, 4) for rank in range(8)] == [0, 0, 1, 1, 2, 2, 3, 3]

# 测试 P8/D4 映射: 本地 TP=4 远程 TP=8, 每个本地 rank 映射到远程的偶数 rank
def test_remote_tp_rank_p8_d4_maps_to_first_prefill_rank_per_pair():
    assert [get_moriio_remote_tp_rank(rank, 4, 8) for rank in range(4)] == [0, 2, 4, 6]

# 测试非倍数配置抛出异常
@pytest.mark.parametrize(
    ("local_tp_rank", "local_tp_size", "remote_tp_size"),
    [(0, 6, 4), (0, 4, 6)],
)
def test_remote_tp_rank_invalid_non_multiple_tp_raises(
    local_tp_rank: int, local_tp_size: int, remote_tp_size: int
):
    with pytest.raises(ValueError, match="multiple"):
        get_moriio_remote_tp_rank(local_tp_rank, local_tp_size, remote_tp_size)

```

vllm/distributed/kv_transfer/kv_connector/v1/moriiio/moriiio_common.py

新增 `MoRIIOTransferAck` `NamedTuple` 类型，为结构化 ACK 提供数据类型基础，影响所有消息处理路径。

```
# 结构化 ACK 消息类型
# transfer_id: 传输标识符
# consumer_tp_size: 消费者端的 TP 大小，用于 producer 端计算期望的 ACK 数量
class MoRIIOTransferAck(NamedTuple):
    transfer_id: TransferId
    consumer_tp_size: int = 1 # 默认 1 表示同构或消费者 TP 较小
```

评论区精华

本 PR 没有 Review 讨论，仅由 `tjtanaa` 批准。`tjtanaa` 在两台 MI355X 节点上进行了手动验证并留下了详细的部署命令记录，确保功能正确。

- 验证与部署记录 (testing): 验证通过，PR 获得批准。

风险与影响

- 风险:
 1. 映射假设倍数关系: `get_moriiio_remote_tp_rank` 要求 `local_tp_size` 和 `remote_tp_size` 满足倍数关系，若用户配置不满足则会在初始化时抛出 `ValueError`，但若在运行时动态变化则可能存在未覆盖的校验。
 2. KV head 复制限制: `validate_moriiio_heterogeneous_tp_kv_heads` 仅支持复制 KV head 或 MLA，若模型使用 split KV head (如某些 GQA 配置) 且 TP 数大于 head 数时会报错，但该限制可能不透明，用户需要显式了解。
 3. ACK 计数依赖于 `consumer_tp_size` 正确传递: 若旧版本 producer 发送无 `consumer_tp_size` 的 `release` 消息，则当作单个 ACK 处理，但若配置为异构 TP 且消费者 TP 更大，则可能导致计数不足从而遗漏 ACK。
 4. 向后兼容性: 接受纯文本字符串 ACK 作为单个 ACK，但 producer 端如果同时发送了结构化 ACK 与纯文本 ACK 的混合，可能造成重复计数或状态混乱 (已通过去重逻辑缓解)。- 影响: 该 PR 仅影响使用 MoRIIO 进行 KV 传输并启用异构 TP 的用户。对于同构 TP 用户无功能变化，保持向后兼容。系统方面，增加了一个 `NamedTuple` 和几个全局函数，对性能影响可以忽略。团队需要确保在 ROCm 环境下正确配置 MoRIIO 以及理解映射规则。- 风险标记: 异构 TP 倍数限制，KV head 复制要求，旧版消息向后兼容，ACK 计数依赖 `consumer_tp_size` 正确传递

关联脉络

- PR #46290 [ROCm][P/D] Fix MoRIIO WRITE mode for mixed KV layouts: 本 PR 构建在 #46290 的 typed control-message 支持之上，扩展支持异构 TP fan-in。