

PR #47764 完整报告

vllm-project/vllm

[ROCm][KVConnector][MoRI-IO] Fix WRITE-mode remote-TP rank collapse (#46332 follow-up)

合并时间: 2026-07-28 09:26

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47764>

执行摘要

- 一句话: 修复 WRITE 模式远程 TP 秩折叠问题
- 推荐动作: 建议精读 `moriiio_common.py` 中 `add_new_req` 方法对 `remote_tp_size` 的降级处理模式, 该模式在处理分布式配置键的多版本兼容性方面具有参考价值。

功能与动机

46332 引入了异构 TP 支持, WRITE 模式下 prefill 需要查询 decode 的 `remote_tp_size` 以计算正确的目标秩。但代理在 `kv_transfer_params` 中设置的是 `remote_tp_size` 而非 `tp_size`, 而旧代码读取 `tp_size` (未设置), 默认值为 1, 导致 8 个 prefill TP 秩全部收敛到 decode 秩 0, 仅 1/8 的 decode KV 缓存被写入, 精度接近零。

实现拆解

1. 修复配置键读取 (`moriiio_common.py add_new_req` 方法): 将 `kv_transfer_params.get("tp_size", 1)` 替换为 `int(kv_transfer_params.get("remote_tp_size") or kv_transfer_params.get("tp_size") or 0)`; 优先读取 `remote_tp_size`, 若不存在则降级读取 `tp_size`, 两者均不存在时直接返回 0 作为 sentinel 值。
2. 保护同构 TP 回退 (`moriiio_connector.py _remote_tp_rank` 方法): 在调用 `get_moriiio_remote_tp_rank` 之前, 判断 `remote_tp_size == 0` (即上游未提供该值) 时, 使用本地 `world_size` 替代, 确保同构 TP 场景下秩映射按预先分配的方式正确执行, 避免收敛至 0。
3. 增强 `_compute_block_transfer_offsets` 回退逻辑 (`moriiio_connector.py`): 在 `validate_moriiio_heterogeneous_tp_kv_heads` 调用中, 将 `remote_tp_size` 判断从 `if remote_tp_size is not None else self.world_size` 修改为 `if remote_tp_size and remote_tp_size > 0 else self.world_size`, 同时覆盖 `None` 和 0 两种缺失情况。

关键文件:

- `vllm/distributed/kv_transfer/kv_connector/v1/moriiio/moriiio_common.py` (模块 KV 连接器; 类别 source; 类型 core-logic; 符号 `add_new_req`): 修复配置键读取, 从 `tp_size`

改为优先读取 `remote_tp_size`，并设置 `or 0` 作为 sentinel；这是导致秩崩溃的根本原因。

- `vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py`（模块 KV 连接器；类别 source；类型 core-logic；符号 `_remote_tp_rank`，`_compute_block_transfer_offsets`）：修复 `_remote_tp_rank` 和 `_compute_block_transfer_offsets` 两个方法中的 sentinel 处理，将 0 视为同构 TP 回退，避免秩折叠。

关键符号：`add_new_req`，`_remote_tp_rank`，`_compute_block_transfer_offsets`

关键源码片段

`vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_common.py`

修复配置键读取，从 `tp_size` 改为优先读取 `remote_tp_size`，并设置 `or 0` 作为 sentinel；这是导致秩崩溃的根本原因。

```
# vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_common.py
# 在 add_new_req 方法中，构造 ReqMeta 时修正 tp_size 的来源
# 原代码：tp_size=kv_transfer_params.get("tp_size", 1)
# 代理（proxy）实际将远端 TP 度写入 "remote_tp_size"，而非 "tp_size"
# 旧字段在 WRITE 预填充请求中未设置，导致默认值 1，所有秩折叠至 decode rank 0
```

```
_req = ReqMeta(
    # ... 其他字段保持不变 ...
    # Remote peer TP degree (used as remote_tp_size downstream). The
    # proxy advertises it under "remote_tp_size"; #46332 read "tp_size"
    # which is absent on WRITE producer requests -> defaulted to 1 ->
    # rank collapse. Read the right key; 0 == unknown (== homogeneous).
    tp_size=int(
        kv_transfer_params.get("remote_tp_size") # 优先：代理提供的正确键
        or kv_transfer_params.get("tp_size") # 降级：旧配置兼容
        or 0 # 兜底：标记为未知
    ),
    # ... 其他字段 ...
)
```

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

修复 `_remote_tp_rank` 和 `_compute_block_transfer_offsets` 两个方法中的 sentinel 处理，将 0 视为同构 TP 回退，避免秩折叠。

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

```
class MoRIIOConnector:
    def _remote_tp_rank(self, remote_tp_size: int) -> int:
        # 0/unknown remote TP == homogeneous (avoids collapsing all ranks to 0).
        # moriio_common.py 已将缺失值转化为 0，因此此处只需判断 == 0 即可
        if remote_tp_size == 0:
            remote_tp_size = self.world_size
        return get_moriio_remote_tp_rank(self.tp_rank, self.world_size, remote_tp_size)
```

```

def _compute_block_transfer_offsets(
    self,
    # ... 其他参数 ...
    remote_tp_size: int | None = None,
) -> tuple[list[int], list[int], list[int]]:
    # ... 方法体 ...
    validate_moriiio_heterogeneous_tp_kv_heads(
        local_tp_size=self.world_size,
        remote_tp_size=(
            remote_tp_size
            if remote_tp_size and remote_tp_size > 0 # 同时排除 None 和 0
            else self.world_size
        ),
        total_num_kv_heads=self.model_config.get_total_num_kv_heads(),
        is_mla=self._is_mla_cache_layer(layer_name),
    )
    # ... 后续逻辑 ...

```

评论区精华

关于 `<= 0` 还是 `== 0` 的 sentinel 检查：lcskrishna 最初建议使用 `if remote_tp_size is None`，但 avininjamay8 指出 `moriiio_common.py` 已将值强制转换为 `int`，`None` 已转化为 `0`。lcskrishna 进而建议使用 `== 0` 替代 `<= 0`，以避免静默掩盖负数值的上游 bug。avininjamay8 采纳该建议，在第二个提交中将 `<= 0` 改为 `== 0`。

- `remote_tp_size` sentinel 检查：`<= 0` vs `== 0` (correctness): 采用 `== 0`。

风险与影响

- 风险：
 1. 回归风险（低）：两个文件仅核心路径的 2 处关键逻辑更改，且通过同构 / 异构 TP 多组测试验证。
 2. 性能影响（无）：仅修复秩映射，不含额外开销。
 3. 兼容性风险（低）：新增 `remote_tp_size` 键读取，旧配置仅含 `tp_size`，or 链保证向下兼容。
 4. 可维护性风险（极低）：通过显式 `int` 强制和 sentinel `0` 统一处理缺失键场景，链路清晰。
 - 影响：用户影响（高）：MoRIIO WRITE 模式在 1P1D TP8 同构和异构场景（如 prefill TP4 / decode TP8）下 KV 传输精度从 0% 恢复至正常水平（GSM8K ~0.94~0.95）。系统影响（无）：不变更任何接口或协议，仅修正内部参数传递。团队影响（低）：约 15 行变更，代码可追溯。
 - 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #46332 Add remote tp size (heterogeneous tp transfer) for MoRIIO connector (#46332): 本 PR 是 #46332 的 follow-up，修复该 PR 引入的 WRITE 模式下 TP 秩折叠 bug。