

PR #39565 完整报告

vllm-project/vllm

[Fix][MoRI] Align MoRI-IO message format with P2pNcclConnector and vllm-router

合并时间: 2026-04-23 07:06

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/39565>

执行摘要

- 一句话: 对齐 MoRI-IO 连接器消息格式, 使其与 vllm-router 兼容。
- 推荐动作: 该 PR 值得精读, 特别是地址嵌入和解析的设计决策, 展示了如何通过 `request_id` 传递元数据来简化分布式通信。关注 `parse_moriiio_zmq_address` 和 `get_peer_zmq_from_request_id` 的实现, 以及错误处理策略。

功能与动机

根据 Issue #38692, vLLM router 当前不支持 MoRI KV 连接器, 导致 ROCm 上的 disaggregated serving 体验不如 CUDA。此 PR 旨在对齐 MoRI-IO 与 P2pNcclConnector 的消息格式, 使 MoRI-IO 兼容 vllm-router, 从而提升 ROCm 用户的 parity。

实现拆解

1. 在 `moriiio_common.py` 中新增地址解析函数: 添加 `parse_moriiio_zmq_address` 和 `get_peer_zmq_from_request_id` 函数, 用于解析 ZMQ 地址格式 "host:IP,handshake:PORT,notify:PORT" 并从 `request_id` 中提取 peer 地址。这样, 连接器可以从 `request_id` 中获取连接信息, 无需 router 传递。
2. 重构 toy proxy 服务器注册逻辑: 在 `moriiio_toy_proxy_server.py` 中, 更新 `_listen_for_register` 函数, 使用新的消息格式 (类型为 "P" 或 "D"), 并验证必需字段。移除了旧的 `_append_whole_dict_unique` 函数, 改为直接管理实例列表, 支持实例更新。
3. 修改连接器核心逻辑: 在 `moriiio_connector.py` 的 `update_state_after_alloc` 方法中, 使用 `get_peer_zmq_from_request_id` 和 `parse_moriiio_zmq_address` 来获取 peer 地址, 替代从 `kv_transfer_params` 中读取。同时, 在 `request_finished` 方法中返回简化的 `kv_transfer_params`, 移除冗余字段。
4. 配套调整: 更新了常量如 `PING_INTERVAL` 从 5 秒改为 3 秒, 并添加了错误处理逻辑, 确保服务稳定性。

关键文件:

- `vllm/distributed/kv_transfer/kv_connector/v1/moriiio/moriiio_common.py` (模块 分布式模块; 类别 source; 类型 dependency-wiring; 符号 `parse_moriiio_zmq_address`, `get_peer_zmq_from_request_id`): 新增关键地址解析函数, 定义了 ZMQ 地址格式和从 `request_id` 提取 peer 地址的逻辑, 是消息格式对齐的核心。

- `examples/online_serving/disaggregated_serving/moriiio_toy_proxy_server.py` (模块 示例服务; 类别 `source`; 类型 `dependency-wiring`; 符号 `_listen_for_register`, `start_service_discovery`): 更新了 `toy proxy` 服务器的注册逻辑, 以支持新的消息格式和地址嵌入, 是测试和示例的关键文件。
- `vllm/distributed/kv_transfer/kv_connector/v1/moriiio/moriiio_connector.py` (模块 分布式模块; 类别 `source`; 类型 `core-logic`; 符号 `update_state_after_alloc`, `request_finished`): 修改了连接器核心逻辑, 使用新的地址解析函数替代显式参数传递, 影响 KV 传输流程。

关键符号: `parse_moriiio_zmq_address`, `get_peer_zmq_from_request_id`, `_listen_for_register`, `update_state_after_alloc`

关键源码片段

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

新增关键地址解析函数, 定义了 ZMQ 地址格式和从 `request_id` 提取 `peer` 地址的逻辑, 是消息格式对齐的核心。

```
import regex as re # 新增导入正则模块用于解析

# 正则表达式用于从 request_id 中提取 ZMQ 地址
_PREFILL_ZMQ_RE = re.compile(r"__prefill_addr_(.+?)__decode_addr_")
_DECODE_ZMQ_RE = re.compile(r"__decode_addr_(.+)_[0-9a-f]{32}(?:-.*)?$")
```

```
def parse_moriiio_zmq_address(
    zmq_address: str,
) -> tuple[str, int, int]:
    """解析 MoRI-IO ZMQ 地址为组件。
```

将 `"host:IP,handshake:PORT,notify:PORT"` 解析为 `(host, handshake_port, notify_port)`。

每个键值对在第一个冒号处分割, 以正确处理 IPv6 地址。
如果缺少 `host`、`handshake` 或 `notify` 键, 或端口值非数字, 则抛出 `ValueError`。

```
"""
parts: dict[str, str] = {}
for segment in zmq_address.split(","):
    key, _, val = segment.partition(":") # 使用 partition 确保只分割第一个冒号
    parts[key.strip()] = val.strip()
try:
    host = parts["host"]
    handshake_port = int(parts["handshake"])
    notify_port = int(parts["notify"])
except (KeyError, ValueError) as e:
    raise ValueError(
        f"Malformed zmq_address {zmq_address!r}: expected "
        f"'host:IP,handshake:PORT,notify:PORT' format"
    ) from e # 抛出错误而非静默回退, 确保数据正确性
return host, handshake_port, notify_port
```



```

    ) # 记录错误并跳过, 避免崩溃
    continue
# 构建实例信息, 从 http_address 派生 request_address
instance = {
    "role": role,
    "request_address": f"http://{data['http_address']}/v1",
    "http_address": data["http_address"],
    "zmq_address": data["zmq_address"], # ZMQ 地址将嵌入 request_id
    "dp_size": data["dp_size"],
    "tp_size": data["tp_size"],
    "transfer_mode": data["transfer_mode"],
}
global TRANSFER_TYPE
transfer_mode = instance["transfer_mode"]
target_list = prefill_instances if role == "P" else decode_instances
with _list_lock:
    if TRANSFER_TYPE is None:
        TRANSFER_TYPE = transfer_mode
        logger.info("SET TRANSFER TYPE TO %s", TRANSFER_TYPE)
    elif transfer_mode != TRANSFER_TYPE:
        logger.error(
            "Mismatched transfer mode: expected %s, got %s; skipping registration of
            %s",
            TRANSFER_TYPE,
            transfer_mode,
            data["http_address"],
        ) # 记录错误并跳过, 而不是抛出异常
        continue
# 检查现有实例并更新, 以处理重启
existing_idx = next(
    (idx for idx, i in enumerate(target_list) if i.get("http_address") == data["http_
    address"]),
    None,
)
if existing_idx is not None:
    target_list[existing_idx] = instance # 更新现有条目
    logger.info("Updated existing %s instance: %s", "Prefill" if role == "P" else
    "Decode", instance)
else:
    target_list.append(instance) # 添加新条目
    logger.info("Registered %s instance: %s", "Prefill" if role == "P" else "Decode",
    instance)
else:
    logger.warning("Received message with unrecognized type %r; ignoring", data.
    get("type")) # 记录警告

```

评论区精华

- 服务线程终止风险: gemini-code-assist[bot] 指出在 toy proxy 中, 抛出 ValueError 可能终止后台线程, 建议改为日志记录和跳过。simondanielsson 同意, 但指出问题在 PR 前已存在, 后续提交中已修复。
- 地址解析错误处理: 对于 malformed ZMQ 地址, gemini-code-assist[bot] 建议回退到默认值, 但 simondanielsson 认为应抛出错误以避免静默连接错误, 决策是保持抛出错误。
- transfer_id 缺失风险: 评论指出 WRITE 模式下缺少 transfer_id 可能导致 KeyError, 已通过提交修复。
- 注册逻辑刷新: 建议更新注册逻辑以处理实例重启, 已通过提交实现刷新现有条目。
 - toy proxy 中 ValueError 导致服务线程终止的风险 (correctness): 后续提交中已改为日志记录和跳过, 避免服务崩溃。
 - malformed ZMQ 地址解析错误处理策略 (design): 决策是抛出 ValueError, 避免静默连接错误。
 - WRITE 模式下 transfer_id 缺失的 KeyError 风险 (correctness): 已通过提交修复, 确保 transfer_id 被传递。
 - 注册逻辑应刷新现有条目以处理实例重启 (design): 已通过提交实现, 在 `_listen_for_register` 中添加了检查并更新现有实例的逻辑。

风险与影响

- 风险:
 - 解析错误风险: 如果 request_id 格式不正确或 ZMQ 地址 malformed, `parse_morrio_zmq_address` 或 `get_peer_zmq_from_request_id` 可能抛出 ValueError, 导致请求失败或引擎崩溃。
 - 服务稳定性: toy proxy 中的错误处理不当可能影响服务可用性, 但已通过日志记录和跳过无效注册改进。
 - 兼容性依赖: 新格式需要 vllm-router 端配合更新, PR body 提到需要 router 端的两个 PR (#138 和 #114), 否则可能无法完全工作。
- 影响:
 - 用户影响: ROCm 用户现在可以使用 vllm-router 与 MoRI-IO 连接器进行 disaggregated serving, 提升体验并与 CUDA 环境对齐。
 - 系统影响: MoRI-IO 连接器现在与 P2pNcclConnector 的消息格式对齐, 简化了分布式 KV 传输的逻辑, 减少了 router 的负担。
 - 团队影响: 需要协调 router 端的更新, 确保整体兼容性; 代码变更涉及核心分布式模块, 可能影响后续开发。
 - 风险标记: 解析错误风险, 服务线程终止, 兼容性依赖

关联脉络

- PR #40597 [Bugfix][CI] Fix v1/kv_connector/unit/test_nixl_connector_hma.py::test_fewer_blocks_with_hma: 同属 kv-connector 模块的测试修复, 涉及分布式 KV 传输的稳定性。