PR #23929 完整报告

sgl-project/sglang

[AMD] Support sdma path for moriep

合并时间：2026-04-30 14:57

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23929

执行摘要

- 一句话：为 AMD MoRI EP 添加 SDMA 路径支持
- 推荐动作：该 PR 提供了明确的硬件加速路径，设计简洁，值得 AMD 相关开发者关注。建议合并后补充单元测试覆盖 SDMA 路径的 dispatch/combine 逻辑，并考虑增加版本检测以增强鲁棒性。

功能与动机

AMD 平台需要利用 SDMA（System DMA）的硬件能力降低 MoE token 调度延迟。通过将融合的 dispatch/combine 拆分为 send 和 recv 两个阶段，可以在通信和计算之间实现更细粒度的重叠，提升低延迟场景下的吞吐量。

实现拆解

1. 在 init_mori_op 函数中添加 enable_sdma 参数：允许调用方控制是否启用 SDMA 路径，该参数默认 False，通过 LRU 缓存保持线程安全。
2. 在 _MoriEPDispatcherImplBase.__init__ 中读取环境变量：使用 get_bool_env_var("MORI_ENABLE_SDMA", "false") 初始化 self.enable_sdma，便于运行态切换。
3. 修改低延迟模式判断逻辑：将 async_mode 的条件从仅检查 deepep_mode.enable_low_latency() 改为其与 enable_sdma 的或运算，使得 SDMA 启用时自动选用 EpMode.LOW_LATENCY 配置。
4. 在 _dispatch_core 和 _combine_core 中根据 enable_sdma 选择不同 API：启用 SDMA 时，使用 dispatch_send + dispatch_recv 和 combine_send + combine_recv 代替原有的融合 dispatch 和 combine，保持参数接口一致。

关键文件：

- python/sglang/srt/layers/moe/token_dispatcher/moriep.py（模块 调度器；类别 source；类型 core-logic；符号 init_mori_op, _MoriEPDispatcherImplBase.init, _MoriEPDispatcherImplBase.mori_op, _dispatch_core）：核心变更文件，包含 SDMA 初始化、配置和调度逻辑的全部修改。

关键符号：init_mori_op, _MoriEPDispatcherImplBase.init, _MoriEPDispatcherImplBase.mori_op, _dispatch_core, _combine_core

关键源码片段

python/sglang/srt/layers/moe/token_dispatcher/moriep.py

核心变更文件，包含 SDMA 初始化、配置和调度逻辑的全部修改。

```
# python/sglang/srt/layers/moe/token_dispatcher/moriep.py
# 在 init_mori_op 函数中新增 enable_sdma 参数
@lru_cache(maxsize=4)
def init_mori_op(
    group,
    router_topk,
    num_experts,
    num_local_experts,
    hidden_size,
    params_dtype,
    num_max_dispatch_tokens_per_rank,
    deepep_mode,
    instance_id=0,
    fp8_dispatch=False,
    fp4_dispatch=False,
    enable_sdma=False, # 新增参数，用于控制是否启用 SDMA 路径
):
    ...
    # 修改 async_mode 判断逻辑：deepep_mode 或 enable_sdma 任一为真则进入低延迟模式
    async_mode = deepep_mode.enable_low_latency() or enable_sdma
    if async_mode:
        mode = EpMode.LOW_LATENCY
    ...

# 在 _MoriEPDispatcherImplBase.__init__ 中读取环境变量
class _MoriEPDispatcherImplBase:
    def __init__(self, group, router_topk, permute_fusion, num_experts,
                 num_local_experts, hidden_size, params_dtype, deepep_mode,
                 instance_id=0):
        ...
        # 通过环境变量 MORI_ENABLE_SDMA 控制 SDMA 开关，默认关闭
        self.enable_sdma = get_bool_env_var("MORI_ENABLE_SDMA", "false")
        ...

    @property
    def mori_op(self):
        if self._mori_op is None:
            ...
            # 将 enable_sdma 传入 init_mori_op
            self._mori_op = init_mori_op(
                self.group,
                self.router_topk,
                ...
                self.enable_sdma, # 新增参数
```

```
            )
        return self._mori_op
```

评论区精华

review 中 gemini-code-assist[bot] 提出了代码重构建议：将 dispatch/combine 中的 if/else 分支统一为函数引用赋值的模式，减少代码重复。最终版本采纳了建议，使用了 `dispatch_fn = (self.mori_op.dispatch_send if self.enable_sdma else self.mori_op.dispatch)` 的形式。HaiShaw 审阅后批准了 PR，未提出其他疑虑。

- dispatch/combine 中代码重复重构 (design): 采纳建议，最终版本使用了 dispatch_fn = (self.mori_op.dispatch_send if self.enable_sdma else self.mori_op.dispatch) 等简洁写法。

风险与影响

- 风险：
  1. 环境变量依赖风险：SDMA 路径完全由 MORI_ENABLE_SDMA 环境变量控制，若用户误设置或未安装相应硬件，可能导致运行时错误。当前代码未提供回退机制，且缺乏明确的错误提示。
  2. 低延迟模式副作用：当 enable_sdma=True 时，async_mode 被强制启用，将 mode 设置为 EpMode.LOW_LATENCY。这可能覆盖用户通过 deepep_mode 指定的其他模式，影响非 SDMA 场景的行为。
  3. 隐式依赖：需要用户自行确保 mori 库支持 SDMA 相关 API（dispatch_send 等），若库版本不匹配会引发 AttributeError。 - 影响：影响范围：仅影响使用 MoRI EP 且在 AMD 平台上启用了 SDMA 的用户。默认行为不改变，兼容性良好。性能影响：SDMA 拆分 send/recv 有望降低延迟，但具体提升幅度需进一步基准测试验证。团队影响：变更规模小（+21/-5），仅修改一个文件，无需额外测试或文档更新。
- 风险标记：依赖环境变量控制功能切换，缺少 SDMA 路径的单元测试覆盖

关联脉络

- 暂无明显关联 PR