

PR #39720 完整报告

sgl-project/sglang

[Fix] Fix GLM5 mHC PP forward

合并时间: 2026-09-16 15:32

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39720>

执行摘要

- 一句话: 修复 GLM5 mHC 层 PP 前向的 residual 崩溃
- 推荐动作: 值得精读, 但重点不在改动量而在设计判断: 这段 5 行的 diff 是理解 GLM5 mHC 如何把残差流折叠进加宽 hidden_states、以及 PP 边界契约如何随配置切换的最佳入口。建议关注三点: ① 为什么选择 "接收侧 + 发送侧同时按 self.config.mhc 分支", 而不是在 PP 传输层统一处理缺省残差; ② 为什么普通层要保留严格 pp_proxy_tensors["residual"] 查找 (快速失败优于静默 None); ③ 单测被删除后, 本项目对这类内部契约修复的回归策略是真实多卡 e2e 而非 CPU 仿真, 写类似 PR 时应提前按此预期准备验证证据。如果团队后续要在更多 PP 配置上支持 mHC, 建议先补一份键集合一致性的契约测试。

功能与动机

PR body 明确指出问题根因: "GLM5 mHC layers carry their residual streams inside the widened hidden state. The pipeline receive buffers therefore contain `hidden_states` without a separate `residual` tensor.", 而 `Glm5NextModel.forward` 假设每个进入的 PP 载荷都带 `residual` 键, 导致非首 stage 直接报 `residual = pp_proxy_tensors["residual"]` `KeyError: 'residual'`。同时发送侧还会把值为 `None` 的 residual 塞回载荷。目标是让模型的 PP 输入输出与 mHC 的 hidden-state-only 契约对齐, 从而让 GLM-5.3 Flash 在测试环境的 PP4 下可运行。

实现拆解

1. 接收侧按配置切换契约 (`Glm5NextModel.forward`, `python/sglang/srt/models/glm5_next.py`): 非首 rank 分支由原来的无条件 `residual = pp_proxy_tensors["residual"]` 改为 `residual = None if self.config.mhc else pp_proxy_tensors["residual"]`。这样 mHC 层不再尝试读取不存在的键, 普通层仍保留严格查找, 一旦上游契约不匹配会立刻抛 `KeyError` 而不是静默降级。
2. 发送侧对齐同一契约: 非最后 rank 的返回由固定双张量 `PPProxyTensors({"hidden_states": ..., "residual": ...})` 改为先判断 `self.config.mhc`, mHC 时只返回 `PPProxyTensors({"hidden_states": hidden_states})`。这一步很关键: 仅修接收侧会让中间 stage 把 `None` 作为 residual 继续往下发, 形成隐式的空残差传播。
3. 未改动的边界保持原样: 逐层前向循环、`_prepare_aux_hidden_state` 中的 `hc_contract(aux_hidden_state, self.config.hc_mult)`、TBO 分段逻辑、末 stage 归一化

(`residual is None` 时只做 `self.norm(hidden_states)`) 以及 PP 传输实现都未触碰, 收敛风险面。

4. 测试配套 (净效果为零) : 首个 commit [392764b1](#) 附带新增

`test/registered/unit/models/test_glm5_next_pipeline.py` (183 行 CPU 回归, 覆盖跨 PP stage 与单 stage 前向); review 中 [Fridge003](#) 要求删除该文件, 第二个 commit [522ed0d4](#) 将其移除。因此最终 PR 只改模型前向, 验证依赖 e2e CI 与 PR body 中的对照实验, 无新增单测。

5. 配置 / 文档 / 部署: 无公共 API、命令行参数、配置项或文档变更; PR body 明确声明 SM120 相关后端依赖 (SGLang #38430 等) 不在本 PR 内, 本 PR 不单独构成 SM120 部署能力。

关键文件:

- `python/sglang/srt/models/glm5_next.py` (模块 模型前向; 类别 source; 类型 data-contract; 符号 `Glm5NextModel.forward`) : 唯一改动文件;
`Glm5NextModel.forward` 的 PP 收发契约是本次修复的核心, 决定了 mHC 层在流水线并行下能否跑通。

关键符号: `Glm5NextModel.forward`

关键源码片段

`python/sglang/srt/models/glm5_next.py`

唯一改动文件; `Glm5NextModel.forward` 的 PP 收发契约是本次修复的核心, 决定了 mHC 层在流水线并行下能否跑通。

```
class Glm5NextModel(nn.Module):
    def forward(
        self,
        input_ids: torch.Tensor,
        positions: torch.Tensor,
        forward_batch: ForwardBatch,
        input_embeds: torch.Tensor = None,
        pp_proxy_tensors: Optional[PPProxyTensors] = None,
    ) -> Union[torch.Tensor, PPProxyTensors]:
        total_num_layers = self.end_layer - self.start_layer
        if self.pp_group.is_first_rank:
            # 首 stage: hidden_states 来自 embedding 或外部 input_embeds,
            # 尚不存在跨 stage 的残差, 因此 residual 恒为 None。
            if input_embeds is None:
                hidden_states = self.embed_tokens(input_ids)
            else:
                hidden_states = input_embeds
            residual = None
        else:
            assert pp_proxy_tensors is not None
            hidden_states = pp_proxy_tensors["hidden_states"]
            # 修复点 1: mHC 把残差流折进被加宽的 hidden_states, PP 载荷里没有独立的 residual;
```

```

# 普通层仍做严格查找，键缺失时立刻抛错以暴露契约不一致，而不是静默用 None 兜底。
residual = None if self.config.mhc else pp_proxy_tensors["residual"]

# ... 中间为逐层前向：mHC 层在层内部维护残差流，普通层继续显式传递 residual，本次未改 ..
.

if not self.pp_group.is_last_rank:
    # 修复点 2：发送侧必须与接收侧使用同一契约，mHC 只发 hidden_states,
    # 避免把 None 当作 residual 塞进载荷污染下游 stage 的字典。
    if self.config.mhc:
        return PPProxyTensors({"hidden_states": hidden_states})
    return PPProxyTensors(
        {
            "hidden_states": hidden_states,
            "residual": residual,
        }
    )
else:
    # 末 stage 归一化：residual 为 None（含 mHC 路径）时只做 norm,
    # 否则走 fused add + norm；该分支逻辑与修复前一致。
    if not forward_batch.forward_mode.is_idle():
        if residual is None:
            hidden_states = self.norm(hidden_states)
        else:
            hidden_states, _ = self.norm(hidden_states, residual)

```

评论区精华

讨论很短但有一个明确结论：reviewer Fridge003 在新增的 CPU 单测文件 [test/registered/unit/models/test_glm5_next_pipeline.py](#) 上直接评论 "Please remove this test"，作者 yilin-void 回复 "Done" (thread id 4023230615)，第二个 commit 即移除了该测试文件。可见维护者不希望为这一内部载荷修复长期维护一份 CPU 侧仿真的 PP 契约测试，转而以真实多卡 e2e 作为回归手段。随后 Fridge003 触发 [/rerun-test test/registered/e2e/models/test_glm53_flash_b200.py test/registered/e2e/models/test_glm53_flash_h200.py](#)，4-gpu-b200 与 8-gpu-h200 两个 workflow 均通过，并最终给出 APPROVED。可见的讨论中没有对实现方案本身提出异议，也没有留下未解决的技术疑虑。

- 是否保留新增的 GLM5 pipeline CPU 单测 (testing)：测试文件在第二个 commit 522ed0d4 ("Remove GLM5 pipeline unit test per review") 中被移除，PR 最终只含模型前向的 5 行改动，回归保护改由多卡 e2e 承担。
- 用真实 GLM-5.3 Flash e2e 验证 PP 修复 (testing)：e2e 通过后 reviewer 给出 APPROVED 并合并，说明维护者接受以真实多卡流水线测试替代被删除的 CPU 单测。

风险与影响

- 风险：

1. 缺少自动化回归覆盖：单测被删除后，PR 的唯一保护是 GLM-5.3 Flash 的多卡 e2e (`test_glm53_flash_b200.py` / `test_glm53_flash_h200.py`) 与一次性人工对照实验；
`self.config.mhc` 分支逻辑没有任何单测锁定，若后续重构把 `residual` 键重新加回或改动分支条件，CPU CI 无法发现。
1. 契约只覆盖收发两端：改动仅作用于 `Glm5NextModel.forward` 的入 / 出载荷，中间 stage 的 `PPProxyTensors` 键集合在 `mHC` 下变成单键字典，与普通层的双键字典不一致。
若下游或旁路逻辑（如余下各 PP rank 的 `pp_proxy_tensors` 消费点、MTP / dflash capture 相关的 aux hidden states 收集）隐含依赖键集合一致，可能在其他组合下暴露问题——本 PR 材料中无法确认这些路径，需要额外核对。
2. 配置属性耦合：分支判定依赖 `self.config.mhc`，与 `_prepare_aux_hidden_state` 中的 `self.config.mhc` / `self.config.hc_mult` 形成多点耦合；若某份模型配置缺失 `mhc` 字段，会在 `forward` 热路径上直接抛 `AttributeError`，而这里没有默认值兜底。
3. 验证强度有限：PR body 自述这是 "small-sample end-to-end regression check, not a full GSM8K accuracy evaluation or a claim of token-by-token equivalence"，只验证了 50 例子集与 PP4/TP4 结果一致，无法排除长序列、不同 PP 划分 (PP2/PP8) 或投机解码路径下的偏差。 - 影响：
 1. 对用户：直接解锁 GLM-5.3 Flash 在 TP=1, PP=4 (多卡显存受限、需要 PP 切分) 场景下的可用性，此前该组合会在非首 stage 直接崩溃；对已用 TP4 的用户无行为变化。
4. 对系统：改动位于模型前向主路径的 PP 边界，影响面限于 `mHC` 配置 (`config.mhc` 为真) 的 GLM5 模型；普通层路径语义完全不变，因此对其它模型、其它并行组合零影响，回归半径很小。
5. 对团队：本 PR 展示了 "模型侧契约需要跟随配置切换" 的修法，属于可复用的模式；同时暴露出 `mHC` 特性与 PP 组合的覆盖仍偏薄，后续若继续扩 PP 支持需要补更系统的契约测试。测试被 reviewer 主动裁掉，也意味着维护成本与 CI 覆盖之间的取舍由 reviewer 承担，作者后续改动需注意多卡验证责任。 - 风险标记：缺少单测覆盖，核心前向路径变更，`mHC` 契约仅覆盖收发两端，小样本精度验证

关联脉络

- PR #38913 [Kernel] Add H20 block-FP8 MoE configs for GLM-5.3-Flash EP4/EP8: 同属 GLM-5.3 Flash 这条模型支持线：该 PR 补的是 H20 上的 MoE 调优配置，本 PR 补的是 PP4 下的前向正确性，两者共同决定 GLM-5.3 Flash 在不同硬件与并行组合上的可用性。
- PR #39547 [AMD][bugfix] Fix DSV4 MTP crash: 同样是 "用配置 / 标志门控以修复模型侧崩溃" 的修法 (该 PR 用 `is_dspark` 门控 D2H 优化，本 PR 用 `config.mhc` 门控 PP 载荷契约)，可作为同类小改动修复思路的对照。
- PR #39561 [NPU] [bugfix] Fix undefined require_attn_tp_allgather: 同为流水线并行相关路径的启动期 bugfix (涉及 `scheduler_pp_mixin.py`)，说明 PP 组合路径是近期缺陷高发区，本 PR 是同一方向上的模型侧补充。