PR #32130 完整报告

sgl-project/sglang

[NPU] [FIX] Fix performance degradation of Qwen3.5-397B-A17B

合并时间：2026-07-23 14:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32130

执行摘要

- 一句话：移除 NPU 热路径 .item() 同步，修复 Qwen3.5-397B 性能退化
- 推荐动作：建议 NPU 推理栈维护者精读：这是 .item() 设备同步在 decode 热路径上代价的具体案例，也展示了 revert 式修复的取舍。值得关注两点：一是在热路径上避免任何 host-device 同步；二是撤除精度保护时应同步提供精度回归测试。更优解法是把判断量移到 host 侧已经具备的标量或改在 device 端带掩码处理，而非简单二选一。

功能与动机

PR body 明确指出：'In ascend_hybrid_linear_attn_backend.py:275 the .item() operation introduces device synchronization and causes service performance degradation.' 即 #31659 的精度保护在 decode 热路径上引入了 .item() 设备同步，拖垮服务性能；commit message 为 'revert #31659'，说明本次选择直接撤回该保护而非异步化重写。

实现拆解

1. 变更入口：唯一改动文件为 python/sglang/srt/hardware_backend/npu/attention/ascend_hybrid_linear_attn_backend.py，落在 update_mamba_state_after_mtp_verify 方法（MTP verify 后回写 mamba 状态）。
2. 移除同步点：删除 all_step0 = (last_steps == 0).all().item() 及 if not all_step0 分支，使 move_intermediate_cache 无条件执行；这一步直接去掉 .item() 触发的 host-device 阻塞。
3. 行为回归：恢复与 CUDA 路径一致的无条件回写 intermediate_ssm 语义，同时丢弃 #31659 对 NPU fused kernel bfloat16 舍入累积漂移的规避。
4. 配套改动：无新增单测或配置；PR body 给出了 NPU 环境变量集与 bench 结果（吞吐 > 900 tokens/s，Median ITL 约 14.8 ms），CI 经 /tag-and-rerun-ci 触发 NPU 用例后由 sglang-npu-bot 合并。

关键文件：

- python/sglang/srt/hardware_backend/npu/attention/ascend_hybrid_linear_attn_backend.py（模块 NPU 注意力；类别 source；类型 core-logic；符号 update_mamba_state_after_mtp_verify, move_intermediate_cache）：本次唯一改动文件，位于 NPU 硬件后端的 hybrid linear attention 实现。PR 移除了 update_mamba_state_after_mtp_verify 中的 .item() 设备同步点，将条件式

move_intermediate_cache 改为无条件执行，修复 Qwen3.5-397B-A17B 在 NPU 上的性能退化，同时移除 #31659 的精度保护。

关键符号：update_mamba_state_after_mtp_verify, move_intermediate_cache

关键源码片段

python/sglang/srt/hardware_backend/npu/attention/ascend_hybrid_linear_attn_backend.py

本次唯一改动文件，位于 NPU 硬件后端的 hybrid linear attention 实现。PR 移除了 update_mamba_state_after_mtp_verify 中的 .item() 设备同步点，将条件式 move_intermediate_cache 改为无条件执行，修复 Qwen3.5-397B-A17B 在 NPU 上的性能退化，同时移除 #31659 的精度保护。

```
# update_mamba_state_after_mtp_verify：MTP verify 后回写 mamba 状态，
# 位于 NPU 专属的 ascend_hybrid_linear_attn_backend.py。
def update_mamba_state_after_mtp_verify(
    self,
    last_correct_step_indices,
    mamba_track_indices,
    mamba_steps_to_track,
):
    request_number = last_correct_step_indices.shape[0]

    # 取出本次请求对应的 mamba cache 槽位索引与各层缓存
    state_indices_tensor = (
        self.linear_attn_backend.forward_metadata.mamba_cache_indices[:request_number]
    )
    mamba_caches = (
        self.linear_attn_backend.req_to_token_pool.get_speculative_mamba2_params_all_layers()
    )
    conv_states = mamba_caches.conv[0]
    ssm_states = mamba_caches.temporal
    intermediate_state_cache = mamba_caches.intermediate_ssm
    dst_indices_tensor = state_indices_tensor.to(torch.int64) # [N] 目标槽位
    src_indices_tensor = torch.arange( # [N] 源槽位
        dst_indices_tensor.shape[0],
        device=dst_indices_tensor.device,
        dtype=torch.int64,
    )
    last_steps = last_correct_step_indices.to(torch.int64) # [N] 各请求接受步数

    # PR #32130 核心变化：删除原 all_step0 = (last_steps == 0).all().item() 分支。
    # .item() 会触发 host-device 同步，拖累 Qwen3.5-397B-A17B 吞吐；
    # 现无条件回写 intermediate_ssm，保持与 CUDA 路径一致的行为。
    move_intermediate_cache(
        ssm_states,
        intermediate_state_cache,
        dst_indices_tensor,
```

```
        src_indices_tensor,
        last_steps,
    )

    draft_token_num = intermediate_state_cache.shape[2]

    # 需要跟踪的步骤（mamba_track_indices 非空时）单独回写中间状态
    if mamba_track_indices is not None:
        assert mamba_steps_to_track is not None
        mamba_track_indices = mamba_track_indices.to(torch.int64)
        mamba_steps_to_track = mamba_steps_to_track.to(torch.int64)
        move_intermediate_cache(
            ssm_states,
            intermediate_state_cache,
            mamba_track_indices,
            src_indices_tensor,
            mamba_steps_to_track,
        )
        track_mask = mamba_steps_to_track >= 0
        # NPU 不保留逐 step 的 conv 中间态，回滚前先记录 verify 窗口的 conv 状态
        track_indices = mamba_track_indices[track_mask]
        if track_indices.numel() > 0:
            conv_states[:, track_indices] = conv_states[:, dst_indices_tensor[track_mask]]

    # 回滚 conv 状态到 verify 前的槽位
    if dst_indices_tensor.numel() > 0:
        conv_state_rollback(
            conv_states,
            dst_indices_tensor,
            last_steps,
            draft_token_num,
        )
```

评论区精华

review 评论为空，没有实质技术交锋；Issue 评论中 sglang-npu-bot 明确要求 'Only modify the NPU-related parts. After passing the NPU test cases, merge it.'，即只允许改动 NPU 相关部分并通过 NPU 测试后合并，该条件已满足。被移除保护逻辑背后的精度担忧（bfloat16 舍入累积）只在代码注释中体现，讨论区未展开，也未留下精度回归的验证记录。

- NPU 合并范围与 CI 要求 (other): PR 仅改动 NPU 后端文件，经 /tag-and-rerun-ci 触发 NPU 测试后由 sglang-npu-bot 合并。

风险与影响

- 风险：
  1. 精度回归风险（高）：原注释指出 NPU 上 fused kernel 的 bfloat16 舍入差异会在数千 decode 步中累积并漂移为垃圾输出；本 PR 在无精度测试的情况下移除了该保护，

accept_lens == 1 时也将执行 move_intermediate_cache。

2. 性能收益不确定：虽然移除了 .item() 同步，但 move_intermediate_cache 现在总是执行；若该拷贝在部分 batch 下开销大于同步，性能可能回落。bench 仅覆盖 Qwen3.5-397B-A17B + random 长输入一种负载。
3. 影响面：仅影响 NPU + ascend_hybrid_linear_attn_backend（hybrid Mamba + MTP 推测解码），CUDA/ROCm 路径不受影响。 - 影响：影响范围较窄但直接：NPU 上以 Qwen3.5-397B-A17B（w4a8、tp16、dp4、NEXTN）配置运行的用户可看到吞吐回升至 900+ tokens/s、Median ITL 约 14.8 ms；代码层面减少一个 decode 热路径同步点，为 NPU 后端后续消除同步提供了范例。对团队而言，需要补充 MTP 精度基线测试，否则该文件的 ' 性能 - 精度跷跷板 ' 可能再次出现。 - 风险标记：核心推理热路径，缺少精度测试覆盖，精度回归风险，revert 式修复

关联脉络

- PR #31659 （被本 PR revert，原标题未知）：本 PR 的 commit message 即为 'revert #31659'，两者围绕同一函数的设备同步与精度取舍，head 分支名 br_revert_prefix 也指向该回退。
- PR #33102 [gdn] fused replayssm ring write into flashinfer gdn mtp verify kernel: 同属 speculative MTP verify 后 mamba 状态更新路径的 kernel 优化，反映了该功能线持续的演进方向，本 PR 是其中针对 NPU 同步开销的回退修正。