PR #25839 完整报告

sgl-project/sglang

[NPU] Support chunk prefill for Qwen3.5/Qwen3.6 models

合并时间：2026-05-21 14:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25839

执行摘要

- 一句话：修复 NPU 上 Qwen3.5/3.6 分块预填充准确性问题
- 推荐动作：建议合并。修复明确、改动量小，且附有精度对比证据。但缺乏对应的单元测试，未来建议补充 NPU 上的分块预填充测试。

功能与动机

PR body 明确指出 Qwen3.5/3.6 模型不支持分块预填充（chunked prefill），存在准确性问题。经分析发现跨分块传递的 ssm_states 形状错误。

实现拆解

变更包含两处改动：

1. 在 ascend_gdn_backend.py 的 forward_extend 函数中，移除针对投机解码的特殊转置分支（原第 353-360 行），统一使用 last_recurrent_state = last_recurrent_state.to(ssm_states.dtype, copy=False) 直接赋值，不再区分 spec_algorithm 类型。
2. 在 memory_pool.py 的 __init__ 函数中，当 speculative_num_draft_tokens 非空且后端为 NPU 时，对 temporal_state 进行转置（transpose(-1, -2)），并同步更新 temporal_state_shape，确保中间 SSM 状态缓存张量形状与 NPU 内核期望一致。

关键文件：

- python/sglang/srt/hardware_backend/npu/attention/ascend_gdn_backend.py（模块 注意力后端；类别 source；类型 core-logic；符号 forward_extend）：移除投机解码分支的 ssm_states 转置，统一赋值路径。
- python/sglang/srt/mem_cache/memory_pool.py（模块 内存池；类别 source；类型 core-logic；符号 init）：为 NPU 添加 temporal_state 转置以对齐中间状态形状。

关键符号：forward_extend, init

关键源码片段

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

移除投机解码分支的 ssm_states 转置，统一赋值路径。

```
def forward_extend(...):
    # ... 前向逻辑 ...
    core_attn_out, last_recurrent_state, h = self.kernel_dispatcher.extend(...)
    if last_recurrent_state is not None:
        last_recurrent_state = last_recurrent_state.to(ssm_states.dtype, copy=False)
    # 原逻辑：根据 spec_algorithm 分支转置或直接赋值
    # 现统一直接赋值，消除分歧
    ssm_states[cache_indices] = last_recurrent_state
    if h is not None:
        self._track_mamba_state_extend(forward_batch, h, ssm_states, forward_metadata)
    return core_attn_out
```

python/sglang/srt/mem_cache/memory_pool.py

为 NPU 添加 temporal_state 转置以对齐中间状态形状。

```
temporal_state = torch.zeros(
    size=(num_mamba_layers, size + 1) + temporal_state_shape,
    dtype=ssm_dtype, device=device,
)
if speculative_num_draft_tokens is not None:
    if _is_npu:
        # NPU 内核需要特定维度顺序，转置最后两维
        temporal_state = temporal_state.transpose(-1, -2)
        # 同步更新 shape 以便后续状态缓存创建
        temporal_state_shape = (
            *temporal_state_shape[:-2],
            temporal_state_shape[-1],
            temporal_state_shape[-2],
        )
    intermediate_ssm_state_cache = torch.zeros(
        size=(
            num_mamba_layers, spec_state_size + 1,
            speculative_num_draft_tokens,
            *temporal_state_shape,
        ),
        dtype=ssm_dtype, device="cuda",
    )
```

评论区精华

无实质 review 讨论。仅有 sglang-npu-bot 自动批准。CI extra 曾失败，后经 /rerun-failed-ci 和 /tag-and-rerun-ci extra 重试。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。

- 移除投机解码分支的转置逻辑：该分支仅在 forward_batch.spec_algorithm.is_none() 为 False 时执行，即投机解码场景。变更后统一走不带转置的路径，可能影响 NPU 上投机解码的正确性，但 PR 未提供相关测试。
- NPU 分支转置：增加 NPU 特有逻辑，仅影响 NPU + 投机解码场景，但同样缺乏测试覆盖。
- 影响：影响范围限于 NPU 硬件后端。修复 Qwen3.5/3.6 分块预填充的精度问题（附有 ceval 指标对比图，修复后精度恢复正常）。用户侧：NPU 用户使用上述模型时，分块预填充功能可用且准确。团队侧：改动极简，影响面窄。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR