

PR #27966 完整报告

sgl-project/sglang

[Spec] Dedup post-verify mamba state commit into shared spec_utils helpers

合并时间: 2026-06-12 09:33

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27966>

执行摘要

- 一句话: 合并 EAGLE/NGRAM 的 Mamba 状态提交逻辑到 spec_utils
- 推荐动作: 推荐 spec 解码相关开发者精读, 尤其是 AST 对比验证方法和模块化提取策略值得借鉴。其余开发者可了解架构演进方向。

功能与动机

PR #26351 修复了 NGRAM 推测解码后 Mamba 状态未提交的 bug, 但修复逻辑在 EAGLE v2 和 NGRAM 中存在两处重复, 增加维护成本。本 PR 通过提取公共 helper, 消除重复并统一行为, 同时扩展对 Kimi-Linear 等模型的支持。

实现拆解

1. 在 spec_utils.py 中新增 prepare_mamba_track_for_verify 函数, 封装 verify 前刷新 mamba track indices 的逻辑 (清空 mask/seqLens)。
2. 在 spec_utils.py 中新增 commit_mamba_states_after_verify 函数, 封装 verify 后提交各请求最后一个接受步骤的 Mamba 状态的逻辑, 并处理跨区间跟踪。
3. 从 eagle_worker_v2.py 删除 _mamba_verify_update 方法, 调用 commit_mamba_states_after_verify; 从 ngram_worker.py 删除 _mamba_verify_update 和内联 prepare 代码, 改用两个新函数。
4. 在 eagle_info_v2.py 中, 将 prepare_for_v2_verify 中原有的内联块替换为 prepare_mamba_track_for_verify 调用。
5. 引入行为变化: gate 条件改为 mambaish_config is not None, 扩展覆盖范围; EAGLE 路径增加了对 attn_backend 是否有 update_mamba_state_after_mtp_verify 方法的防御性检查。

关键文件:

- python/sglang/srt/speculative/spec_utils.py (模块 推测解码工具; 类别 source; 类型 core-logic; 符号 prepare_mamba_track_for_verify, commit_mamba_states_after_verify): 核心提取位置, 新增两个公共函数 prepare_mamba_track_for_verify 和 commit_mamba_states_after_verify, 被 EAGLE 和 NGRAM 统一调用。
- python/sglang/srt/speculative/eagle_worker_v2.py (模块 EAGLE 推理核; 类别 source; 类型 core-logic; 符号 _mamba_verify_update, verify): 删除原有的

_mamba_verify_update 方法，改为调用 spec_utils 中的公共函数，简化验证流程。

- python/sglang/srt/speculative/ngram_worker.py (模块 NGRAM 推测核；类别 source；类型 core-logic；符号 _mamba_verify_update, _prepare_for_speculative_decoding)：删除内联的 mamba 状态更新逻辑，统一使用 spec_utils 函数，同时删除无用的导入。
- python/sglang/srt/speculative/eagle_info_v2.py (模块 推测信息层；类别 source；类型 dependency-wiring)：将 prepare_for_v2_verify 中的内联代码替换为 prepare_mamba_track_for_verify 调用，减少重复。

关键符号：prepare_mamba_track_for_verify, commit_mamba_states_after_verify, _mamba_verify_update

关键源码片段

python/sglang/srt/speculative/spec_utils.py

核心提取位置，新增两个公共函数 prepare_mamba_track_for_verify 和 commit_mamba_states_after_verify，被 EAGLE 和 NGRAM 统一调用。

```
def prepare_mamba_track_for_verify(batch: ScheduleBatch) -> None:
    # 在 TARGET_VERIFY 前从 reqs 重建 mamba track indices
    # 清空 mask 避免在 verify 中触发跟踪 (跟踪改由 commit_mamba_states_after_verify 负责)
    if not get_global_server_args().enable_mamba_extra_buffer():
        return
    set_mamba_track_indices_from_reqs(batch)
    batch.mamba_track_mask = None
    batch.mamba_track_seqlens = None

def commit_mamba_states_after_verify(
    target_worker: TpModelWorker,
    batch: ScheduleBatch,
    accept_lens: torch.Tensor,
    accept_index: torch.Tensor,
    draft_token_num: int,
) -> None:
    """提交每个请求最后接受步骤的 Mamba 状态后验证."""
    model_runner = target_worker.model_runner
    if model_runner.mambaish_config is None:
        return
    attn_backend = model_runner.attn_backend
    if not hasattr(attn_backend, "update_mamba_state_after_mtp_verify"):
        return
    bs = accept_lens.shape[0]
    if not batch.forward_mode.is_idle() and accept_index.numel() > 0:
        # 计算每个请求的 accept_indices 偏移
        accept_indices_offset = torch.arange(
            0,
            bs * draft_token_num,
            step=draft_token_num,
```

```

        dtype=accept_lens.dtype,
        device=accept_lens.device,
    )
    req_idx = torch.arange(bs, dtype=torch.int64, device=accept_lens.device)
    # 最后一个接受步骤在树中的索引
    last_correct_step_indices = (
        accept_index[req_idx, (accept_lens - 1).to(torch.int64)]
        - accept_indices_offset
    )
    # 处理跨区间跟踪
    if batch.mamba_track_indices is not None:
        seq_lens_pre_verify = batch.seq_lens
        seq_lens_post_verify = batch.seq_lens + accept_lens
        mamba_track_interval = get_global_server_args().mamba_track_interval
        to_track_mask = (
            seq_lens_pre_verify // mamba_track_interval
            != seq_lens_post_verify // mamba_track_interval
        )
        tracking_point = (
            seq_lens_post_verify // mamba_track_interval * mamba_track_interval
        )
        to_track_ith = torch.clamp(
            tracking_point - seq_lens_pre_verify - 1, min=0
        ).to(torch.int64)
        candidate_track_steps = (
            accept_index[req_idx, to_track_ith] - accept_indices_offset
        )
        mamba_steps_to_track = torch.where(
            to_track_mask,
            candidate_track_steps,
            torch.full_like(candidate_track_steps, -1),
        )
    else:
        mamba_steps_to_track = None

    attn_backend.update_mamba_state_after_mtp_verify(
        last_correct_step_indices=last_correct_step_indices,
        mamba_track_indices=batch.mamba_track_indices,
        mamba_steps_to_track=mamba_steps_to_track,
        model_runner=model_runner,
    )

```

评论区精华

无 review 评论。PR body 详细说明了机械变换的 AST 验证方法和有意行为变化。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低：提取逻辑经过 AST 对比验证，确保与原始代码一致。注意行为变化（`mambaish_config`）可能使之前未覆盖的模型（如 Kimi-Linear）开始执行 Mamba 状态提交，但该提交在不受支持的 backend 上被 `hasattr` 守卫安全跳过。未包含 DFLASH worker 的变体，等待后续统一。
- 影响：对用户无功能影响，对开发者降低维护成本，为未来统一验证管线打下基础。主要在 `speculative-decoding` 子模块内，不影响其他系统。
- 风险标记：核心路径（推测解码）变更，模型行为影响（条件放宽），缺少 DFLASH 变体统一

关联脉络

- PR #26351 [bugfix] commit Mamba states after NGRAM target verify: 本 PR 堆叠在该修复之上，提取其公共逻辑。