PR #25029 完整报告

sgl-project/sglang

[Spec] Mamba scatter cleanup; fix multi-layer positional bug; dflash naming

合并时间：2026-05-12 11:36

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25029

执行摘要

- 一句话：清理 Mamba 散射逻辑并修复多层位置 bug
- 推荐动作：值得阅读以了解代码重构和命名规范化实践。设计决策包括统一参数名称、简化计算步骤、保留向后兼容。

功能与动机

PR body 说明 this is a follow-up cleanup after #25014，旨在保持命名一致性并修复遗留的 bug。

实现拆解

1. 统一参数命名：将 hybrid_linear_attn_backend.py 和 ascend_hybrid_linear_attn_backend.py 中 update_mamba_state_after_mtp_verify 的参数 accept_steps 重命名为 last_correct_step_indices，并更新所有调用方。
2. 简化 eagle_worker._mamba_verify_update：移除临时变量 num_accept_tokens，改用 num_correct_drafts + 1 直接计算累积和，消除冗余。
3. 修复 multi_layer_eagle_worker：将 max_relative_indices_per_req 改为 last_correct_step_indices，并修正调用 update_mamba_state_after_mtp_verify 时的位置参数顺序（原名 accept_steps 已被前一步更改）。
4. dflash 内部重命名：在 dflash_utils.py 和 dflash_worker.py 中，将 accept_len 改为 correct_len，返回值及注释保持一致。
5. 更新注释：所有相关文件中的内联注释和 docstring 跟随新命名更新，提高可读性。

关键文件：

- python/sglang/srt/speculative/eagle_worker.py（模块 投机解码；类别 source；类型 core-logic；符号 _mamba_verify_update）：核心投机解码 worker，修复 _mamba_verify_update 中计算逻辑和命名一致性。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 投机解码；类别 source；类型 core-logic；符号 _mamba_verify_update, move_accepted_tokens_to_target_kvcache）：投机解码 v2 worker，同样需要 Mamba 状态更新修复，并改进 move_accepted_tokens 注释。

- python/sglang/srt/speculative/dflash_utils.py（模块 投机解码；类别 source；类型 core-logic；符号 compute_dflash_correct_drafts_and_bonus, compute_dflash_sampling_correct_drafts_and_bonus）：DFlash 模块，统一命名 accept_len -> correct_len，避免语义混淆。
- python/sglang/srt/layers/attention/hybrid_linear_attn_backend.py（模块 注意力后端；类别 source；类型 core-logic；符号 update_mamba_state_after_mtp_verify）：注意力后端接口调整，重命名参数 accept_steps -> last_correct_step_indices，与调用端对齐。
- python/sglang/srt/speculative/multi_layer_eagle_worker.py（模块 投机解码；类别 source；类型 core-logic；符号 verify）：修复 multi-layer 位置参数传递 bug，统一变量名为 last_correct_step_indices。

关键符号：_mamba_verify_update, update_mamba_state_after_mtp_verify, verify, move_accepted_tokens_to_target_kvcache, compute_dflash_correct_drafts_and_bonus, compute_dflash_sampling_correct_drafts_and_bonus

关键源码片段

python/sglang/srt/speculative/eagle_worker.py

核心投机解码 worker，修复 _mamba_verify_update 中计算逻辑和命名一致性。

```
def _mamba_verify_update(
    self,
    batch: ScheduleBatch,
    res: EagleVerifyOutput,
    logits_output: LogitsProcessorOutput,
    spec_info: EagleVerifyInput,
    seq_lens_pre_verify: torch.Tensor,
):
    # Under DP attention, some ranks can be IDLE during target verify
    if batch.forward_mode.is_idle():
        return

    # 直接使用 num_correct_drafts 而非 num_accept_tokens
    num_correct_drafts = torch.tensor(
        res.num_correct_drafts_per_req_cpu,
        device=logits_output.hidden_states.device,
        dtype=torch.int64,
    )
    # 加 1 以包含 bonus token
    cumulative_num_accept_tokens = torch.cumsum(num_correct_drafts + 1, dim=0)

    ... # 计算 accepted_indices_start 和 offset

    if spec_info.topk > 1 and res.accepted_indices.shape[0] > 0:
        # topk > 1 时通过 accepted_indices 计算最后正确步数
        last_correct_step_indices = (
            res.accepted_indices[cumulative_num_accept_tokens - 1]
```

```
            - accepted_indices_offset
        )
    else:
        # topk == 1 时直接用 num_correct_drafts
        last_correct_step_indices = num_correct_drafts

    ... # 处理 track indices

    # 调用 backend 更新，传递 last_correct_step_indices
    self.target_worker.model_runner.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=self.target_worker.model_runner.model,
    )
```

python/sglang/srt/speculative/eagle_worker_v2.py

投机解码 v2 worker，同样需要 Mamba 状态更新修复，并改进 move_accepted_tokens 注释。

```
def _mamba_verify_update(
    self,
    batch: ModelWorkerBatch,
    verify_input: EagleVerifyInput,
    accept_lens: torch.Tensor,
    accept_index: torch.Tensor,
    bs: int,
):
    # `accept_lens` already includes the bonus token (drafts + 1 per req).
    if not batch.forward_mode.is_idle() and accept_index.numel() > 0:
        if verify_input.topk != 1:
            raise ValueError("Spec v2 currently only supports topk = 1.")

        # 重命名 accept_steps -> last_correct_step_indices
        last_correct_step_indices = accept_lens - 1

        ... # mamba_track_indices 处理

        self.target_worker.model_runner.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=self.target_worker.model_runner.model,
        )
```

python/sglang/srt/speculative/dflash_utils.py

DFlash 模块，统一命名 accept_len -> correct_len，避免语义混淆。

```
def compute_dflash_correct_drafts_and_bonus(
    *,
```

```
    candidates: torch.Tensor,
    target_predict: torch.Tensor,
) -> Tuple[torch.Tensor, torch.Tensor]:
    """
    Returns:
        correct_len: int32 tensor [bs], number of accepted *draft* tokens (excluding current
        token and bonus token).
        bonus: int64 tensor [bs], the target-predicted token at index correct_len.
    """
    ...
    matches = candidates[:, 1:] == target_predict[:, :-1]
    # 重命名 local variable
    correct_len = matches.to(torch.int32).cumprod(dim=1).sum(dim=1)
    bonus = target_predict[torch.arange(bs, device=target_predict.device), correct_len]
    return correct_len, bonus.to(torch.int64)
```

评论区精华

PR 本身没有 review 评论；issue 评论中作者调用了 /rerun-test 重新运行相关测试（test_mamba_state_scatter_triton.py, test_eagle_infer_a.py 等），且所有测试均通过。

- 测试重新运行 (testing): 测试通过

风险与影响

- 风险：风险较低，主要是语义等价的重构。但需注意 update_mamba_state_after_mtp_verify 的参数传递方式（多处以位置参数调用），确保参数名称和顺序匹配新接口。修复的 multi-layer 位置 bug 可能只在特定配置下触发。
- 影响：影响投机解码（Eagle、DFlash）中 Mamba 状态更新逻辑。对用户无影响，但提高了代码可读性和维护性。修复的 multi-layer 位置 bug 在混合 GDN 模型场景下可能避免错误的状态更新。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #25014 [Spec] Internal rename per N2 v2 naming rule: 本 PR 堆叠在 #25014 之上，是后续清理工作。