

PR #37818 完整报告

sgl-project/sglang

[Bugfix] Track DFlash Mamba state at checkpoint boundaries

合并时间: 2026-09-12 09:38

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/37818>

执行摘要

- 一句话: 修复 DFlash 跨 Mamba 边界漏检 checkpoint 状态
- 推荐动作: 值得精读, 约 15-20 分钟。这是一个小而经典的 "状态快照与推进时序不一致" 正确性修复: 问题定位精准 (Issue 给出了可运行的极简复现), 修复点集中 (8 行改动), 验证方法扎实 (56 个冻结用例 + 1,024 请求负载测试 + 剩余 mismatch 的诚实归因)。建议重点关注两点: 一是 `to_track_mask` 与 `to_track_ith` 的边界数学 (255→256、254→2、250→3、256→1 四个代表用例), 二是测试被移除的决策——建议后续把 Issue 中的极简复现补回为 CPU 单测, 以固化这条回归防线。

功能与动机

Issue #37817 报告了一个缓存一致性问题: 在 Qwen3.6-35B-A3B 上使用 DFLASH 和 `mamba-radix-cache-strategy=extra_buffer` 时, 若 `verify step` 跨越 256 token 的 checkpoint 边界, "the state at the checkpoint boundary is not copied to the tracking buffer"。根因是 `seq_lens_pre_verify` 克隆自 `batch.seq_lens`, 而 `batch.seq_lens` 在校验前没有推进, 导致 `to_track_mask` 两侧数值相同、边界永不被检测。后果是 "a radix-cache entry whose full-attention KV corresponds to length 256 while its linear-attention state is stale", 造成静默的错误生成。Issue 还给出了极简复现: `pre=255, commit=1` 时 step 0 应保存为 256 处的 checkpoint, 但 `mamba_steps_to_track` 被设为 -1。

实现拆解

整个修复只涉及 `python/sglang/srt/speculative/dflash_worker_v2.py` 一个文件, 变更量为 +6/-2, 分三步完成:

1. 修改函数签名, 新增 post-verify 长度参数: `_update_target_mamba_state_after_verify` 增加 `seq_lens_post_verify: torch.Tensor` 参数。该函数在 TARGET_VERIFY 阶段负责把每个请求最后一个被接受 step 的 Mamba 中间状态提交到跟踪缓冲区。
2. 用 post-verify 长度替换 `batch.seq_lens` 参与边界检测: `to_track_mask` 与 `tracking_point` 原本都基于 `batch.seq_lens` 计算, 但因为 `batch.seq_lens` 在校验前未推进、与 `seq_lens_pre_verify` 相等, 边界检测恒为 False。修改后统一使用 `seq_lens_post_verify`, 使 255 → 256 这类跨边界场景能被正确识别, `to_track_ith = clamp(tracking_point - seq_lens_pre_verify - 1, min=0)` 计算出的 verify step 序号也与 Issue 中的期望行为一致: `pre=254, commit=2` 时跟踪 step 1, `pre=250, commit=3` 时因超出接受区间不跟踪。

3. 调用处提前计算 `new_seq_lens`: 在 `forward_batch_generation` 中, 原本 `new_seq_lens = prefix_lens + commit_lens.to(prefix_lens.dtype)` 在 Mamba 状态提交之后才计算。现在将其提前 (若为 `None` 才计算, 模拟接受路径中 Triton 可能已写入), 作为 `seq_lens_post_verify` 传入, 随后原有兜底计算与 `on_publish` 回调保持不变, 不改变外部语义。
4. 测试与验证配套: PR body 声称新增了 CPU 回归测试 (覆盖 int32/int64 长度、64/128/256 跟踪间隔、跨边界与不跨边界请求), 但该测试在最终合入前被移除 (见提交 2b54249), 理由为现有 Mamba-hybrid DFlash CI 测试已提供覆盖, 因此最终合入状态仅有源码改动。GPU 侧验证充分: 56 个冻结用例在 `unpatched` 下 24 个不一致、`patched` 下仅剩 2 个 (均为 log-prob margin 0.125 的边界 -1 低置信翻转, 且不经过程本 PR 修改路径); 另在 interval 128/256、并发 1/4/8/16 下完成 1,024/1,024 请求负载测试。

关键文件:

- `python/sglang/srt/speculative/dflash_worker_v2.py` (模块 投机解码; 类别 `source`; 类型 `core-logic`; 符号 `_update_target_mamba_state_after_verify, forward_batch_generation`): 唯一变更文件, 承载全部修复逻辑: 为 `_update_target_mamba_state_after_verify` 新增 `seq_lens_post_verify` 参数, 将 `checkpoint` 跨越检测与跟踪点计算从未推进的 `batch.seq_lens` 切换到验证后的提交长度, 并在 `forward_batch_generation` 中提前计算 `new_seq_lens` 传入。这是 DFlash 与 DFlash2 共用的核心投机解码 worker 路径。

关键符号: `_update_target_mamba_state_after_verify, forward_batch_generation`

关键源码片段

`python/sglang/srt/speculative/dflash_worker_v2.py`

唯一变更文件, 承载全部修复逻辑: 为 `_update_target_mamba_state_after_verify` 新增 `seq_lens_post_verify` 参数, 将 `checkpoint` 跨越检测与跟踪点计算从未推进的 `batch.seq_lens` 切换到验证后的提交长度, 并在 `forward_batch_generation` 中提前计算 `new_seq_lens` 传入。这是 DFlash 与 DFlash2 共用的核心投机解码 worker 路径。

核心修复: `checkpoint` 跨越检测改用 `post-verify` 长度

```
def _update_target_mamba_state_after_verify(
    self,
    *,
    batch: ScheduleBatch,
    seq_lens_pre_verify: torch.Tensor,
    seq_lens_post_verify: torch.Tensor,
    commit_lens: torch.Tensor,
) -> None:
    """Commit Mamba intermediate states for accepted verify steps.
```

校验阶段 Mamba 内核以 ``disable_state_update=True`` 运行, 仅缓存逐 `step` 的中间状态; 接受后需要把每个请求最后一个被接受 `step` 的状态提交到跟踪缓冲区。

```
"""
```

```

if not self._need_mamba_verify_commit:
    return
attn_backend = self.target_worker.model_runner.attn_backend

last_correct_step_indices = commit_lens.to(torch.int64) - 1
mamba_steps_to_track = None

if batch.mamba_track_indices is not None:
    mamba_track_interval = mamba_track_grid(batch.tree_cache.page_size)
    # 核心修复：原实现用 batch.seq_lens 与 seq_lens_pre_verify 比较，
    # 但校验前 batch.seq_lens 尚未推进，两侧数值相同，跨边界检测恒为 False，
    # 导致长度 255 -> 256 时 step 0 未被保存为 checkpoint；
    # 现在改用校验后的提交长度 seq_lens_post_verify 参与比较。
    to_track_mask = (
        seq_lens_pre_verify // mamba_track_interval
        != seq_lens_post_verify // mamba_track_interval
    )
    # 跟踪点取 post-verify 长度所在边界的整数倍，例如 256、512……
    tracking_point = (
        seq_lens_post_verify // mamba_track_interval * mamba_track_interval
    )
    # 计算需保存的 verify step 序号：边界相对 pre-verify 位置的偏移减 1。
    # 例如 pre=254、post=256 时 ith=1，表示第 1 个 verify step 跨越到 256。
    to_track_ith = torch.clamp(tracking_point - seq_lens_pre_verify - 1, min=0)
    # 边界必须落在被接受区间内才允许跟踪：
    # 例如 pre=250、commit=3 时 ith=5 超出接受范围，不应保存 checkpoint。
    can_track_mask = to_track_mask & (
        to_track_ith < commit_lens.to(torch.int64)
    )
    mamba_steps_to_track = torch.where(
        can_track_mask,
        to_track_ith.to(torch.int64),
        torch.full_like(to_track_ith, -1, dtype=torch.int64),
    )

model_runner = self.target_worker.model_runner
if hasattr(attn_backend, "update_mamba_state_after_mtp_verify"):
    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=model_runner.model,
        req_pool_indices=batch.req_pool_indices[: commit_lens.shape[0]],
    )

```

调用处：提前计算 post-verify 长度并传入

```

if self._need_mamba_verify_commit:
    assert seq_lens_pre_verify is not None
    # 提前计算提交后的序列长度并传给 checkpoint 检测；

```

```

# 模拟接受路径中 Triton 内核可能已写入 new_seq_lens, 此时直接复用。
if new_seq_lens is None:
    new_seq_lens = prefix_lens + commit_lens.to(prefix_lens.dtype)
self._update_target_mamba_state_after_verify(
    batch=batch,
    seq_lens_pre_verify=seq_lens_pre_verify,
    # 传入 post-verify 长度, 使 255 -> 256 这类边界可被正确检测。
    seq_lens_post_verify=new_seq_lens,
    commit_lens=commit_lens,
)

if new_seq_lens is None:
    new_seq_lens = prefix_lens + commit_lens.to(prefix_lens.dtype)
if on_publish is not None:
    on_publish(new_seq_lens)

```

评论区精华

本 PR 没有 review 评论 (review_comments_count=0), kpham-sgl 直接 APPROVED。有价值的讨论集中在 Issue 评论与提交历史中:

- CI rebase 与 rerun: github-actions 提示 "A major update has landed on main. Your PR is diverged relative to required base commit", 首次 rerun 未分发; 分支合并 main 后 (提交 3f752d3), Fridge003 再次发起 [/rerun-test](#), 覆盖 [test_dflash_extra_buffer_lazy.py](#)、[test_dflash_overlap_hostsync.py](#) 及 GDN/lightning 注意力等测试, 在 [4-gpu-b200](#) 与 [1-gpu-h100](#) 上全部通过。
- 回归测试被移除: 提交 2b54249 删除了最初加入的 CPU 回归测试, commit message 说明 "Coverage comes from the existing Mamba-hybrid DFlash CI tests instead"。该决策未在 review 评论中展开, 属于合入前的维护性取舍, 但也削弱了针对本 bug 的定向回归防护。
- 剩余 mismatch 的诚实归因: PR body 明确说明 patched 后剩余的 2/56 不一致来自同一个边界 -1 用例, 其 cold-path top-1 与 top-2 log-prob margin 仅 0.125、低于 0.5 稳定阈值, 且该用例不跨边界、不经过本 PR 修改的 checkpoint-crossing 路径, 因此不构成回归证据。
 - DFlash 相关测试 rerun 验证 (testing): 补测通过, 验证修复未破坏 DFlash/Mamba 相关路径; AMD ROCm 槽位未见重跑结果。
 - 回归单测在合入前被移除 (testing): 测试被移除, 最终 PR 仅含源码改动; 依赖现有 DFlash CI 测试提供间接覆盖。

风险与影响

- 风险:
 1. 回归测试缺失: 定向 CPU 回归测试在合入前被移除, Issue #37817 中给出的极简复现 (几行 torch 断言) 未固化为自动化测试。若未来有人重构 `_update_target_mamba_state_after_verify`, 同类 bug 可能再次静默引入。

2. 影响 DFlash 与 DFlash2 双路径: DFlashWorkerV2 是两条路径共用的 worker, 本次修改同时改变两者行为。虽然 GPU 验证覆盖了 Qwen3.6-35B-A3B, 但其他 Mamba 混合模型 (如 Nemotron Lightning 系列, 见 rerun 测试中的 TestNvidiaNemotron35LightningNVFP4DFlash) 建议保持 CI 关注。
3. dtype 隐式类型提升: to_track_ith 的计算涉及 tracking_point - seq_lens_pre_verify - 1, seq_lens_post_verify 与 seq_lens_pre_verify 的 dtype (int32/int64) 可能不同, torch 会发生类型提升; PR 声称测试覆盖了 int32/int64 组合, 但测试未保留, 生产行为依赖张量 dtype 的具体来源。
4. 时序微调: new_seq_lens 的计算被提前到 Mamba 状态提交之前。虽然表达式与后续兜底完全一致、幂等, 但若未来在该区间插入依赖执行顺序的逻辑, 需注意此处的提前计算。
5. CI 状态: PR body 中 Base/Extra/AMD ROCm 三个 CI 槽位显示失败状态, 与 rebase 要求相关; 重跑指定测试集全部通过, 但 AMD ROCm 10 槽位未看到重跑结果, 存在平台覆盖盲区。- 影响: 用户影响: 修复了 Mamba 混合模型 (典型如 Qwen3.6-35B-A3B) 在 DFLASH/DFlash2 + extra_buffer radix cache 策略下 checkpoint 状态漏存导致的静默错误输出。此类错误不报错、不崩溃, 仅产生错误 token, 最难排查, 修复价值高。

系统影响: 消除了 "full-attention KV cache 对应长度 256 而 linear-attention state 停留在旧位置" 的状态漂移, 避免 radix cache 被污染后长期复用错误状态。改动仅替换用于 checkpoint 记账的长度张量, 不引入额外 kernel 或通信开销, 性能无影响。

团队影响: DFlashWorkerV2 是 DFlash/DFlash2 共用核心路径, 后续维护者需要理解 seq_lens_post_verify 的语义 (必须取验证后提交长度, 而非调度器尚未推进的 batch.seq_lens), 本 PR 确立的修复模式对同类 speculative decoding 缓存记账问题有参考价值。

- 风险标记: 核心投机解码路径变更, 回归测试被移除, 依赖现有 CI 间接覆盖, 缓存一致性敏感

关联脉络

- PR #37069 feat: support TP>1 Domino rollout for DFlash V2: 同文件 dflash_worker_v2.py 的后续功能演进, 扩展 DFlash V2 在 TP>1 下的 Domino 回滚路径, 与本 PR 的 verify 提交与 Mamba 状态记账逻辑位于同一 worker 内。
- PR #37709 [PD] Transfer the DCP-replicated DSPARK draft KV in DCP1->DCP-N layouts: 同为 speculative decoding 场景下的 KV 一致性修复, 与本次 full-attention KV 与 linear-attention state 位置漂移问题属于同一大类缓存一致性问题。
- PR #38756 [AMD] aiter: resolve SWA KV pool for draft workers + guard paged decode: 同为 draft worker 路径的 KV 池解析与守卫修复, 与本 PR 的 Mamba 状态跟踪同属 speculative decoding worker 状态记账范畴。