

PR #29654 完整报告

sgl-project/sglang

[spec] Fix index_share_for_mtp_iteration being a no-op in EAGLE MTP draft

合并时间: 2026-06-30 05:52

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29654>

执行摘要

- 一句话: 修复 EAGLE MTP 草稿步中 indexer 重用失效 bug
- 推荐动作: 该 PR 修复了一个隐蔽但性能影响显著的问题, 核心设计 (将跨步状态移动到引用共享的 spec_info 对象) 值得精读。对于维护 EAGLE 草稿或类似跨步共享状态的场景具有参考价值。建议合并。

功能与动机

`index_share_for_mtp_iteration` (#28192) 旨在仅在首个 MTP 草稿步运行 DSA indexer 并在后续步复用其 top-k, 但实际从未生效——每个草稿步都重新计算 indexer, 因为写回丢失在复制的 ForwardBatch 上。短上下文时 indexer 选择所有 token, 恢复与重计算结果一致, 因此问题未被注意到; 长上下文时差异显著, 性能开销大。

实现拆解

1. 数据载体迁移: 在 `python/sglang/srt/speculative/eagle_info.py` 的 `EagleDraftInput` 数据类中新增 `mtp_topk_indices: Optional[torch.Tensor]` 字段, 该字段通过引用在草稿步间共享。
2. 移除旧字段: 在 `python/sglang/srt/model_executor/forward_batch_info.py` 中删除 `ForwardBatch.topk_indices` 字段, 保留 `reuse_mtp_topk_indices` 作为门控标志。
3. 读 / 写目标重定向: 在 `python/sglang/srt/models/deepseek_nextn.py` 的 `forward` 方法中, 将 `prev_topk_indices` 的来源和 `topk_indices` 写回目标从 `forward_batch.topk_indices` 改为 `forward_batch.spec_info.mtp_topk_indices`。
4. 初始化与清理: 在 `python/sglang/srt/speculative/eagle_worker_v2.py` 的 `draft_forward` 方法中, 将初始化 `forward_batch.topk_indices = None` 改为 `spec_info.mtp_topk_indices = None`, 循环后的清理也相应调整。
5. 测试: PR 描述中提供了 GLM-5.2-FP8 在 4xB300 上的手动验证结果, 但未包含自动化测试文件变更。

关键文件:

- `python/sglang/srt/speculative/eagle_info.py` (模块 推测解码; 类别 source; 类型 core-logic; 符号 `EagleDraftInput`): 新增 `mtp_topk_indices` 字段到 `EagleDraftInput` 数据类, 实现跨步引用共享。

- python/sglang/srt/models/deepseek_nextn.py (模块 模型层; 类别 source; 类型 data-contract; 符号 DeepseekV3ForCausalLMNextN.forward) : 重定向 prev_topk_indices 来源和 topk_indices 写回目标到 spec_info.mtp_topk_indices。
- python/sglang/srt/speculative/eagle_worker_v2.py (模块 推测解码; 类别 source; 类型 core-logic; 符号 EagleWorkerV2.draft_forward) : 调整初始化与清理代码, 使用 spec_info.mtp_topk_indices 替代 forward_batch.topk_indices。
- python/sglang/srt/model_executor/forward_batch_info.py (模块 数据契约; 类别 source; 类型 data-contract; 符号 ForwardBatch) : 删除废弃的 topk_indices 字段, 只保留 reuse_mtp_topk_indices 门控标志。

关键符号: EagleDraftInput.init, DeepseekV3ForCausalLMNextN.forward, EagleWorkerV2.draft_forward

关键源码片段

python/sglang/srt/speculative/eagle_info.py

新增 mtp_topk_indices 字段到 EagleDraftInput 数据类, 实现跨步引用共享。

```
# 来自 python/sglang/srt/speculative/eagle_info.py
# EagleDraftInput 数据类, 通过 spec_info 引用在草稿步间共享
@dataclass
class EagleDraftInput(SpecInput):
    topk_p: torch.Tensor = None
    topk_index: torch.Tensor = None
    draft_probs: torch.Tensor = None
    hidden_states: Optional[torch.Tensor] = None
    capture_hidden_mode: CaptureHiddenMode = CaptureHiddenMode.FULL

    # 新增字段: 跨步存活的 topk_indices
    # spec_info 通过引用共享, 写回不会因 ForwardBatch 复制而丢失
    mtp_topk_indices: Optional[torch.Tensor] = None

    bonus_tokens: torch.Tensor = None
    kv_indptr: torch.Tensor = None
    kv_indices: torch.Tensor = None
    num_tokens_per_req: int = -1
    num_tokens_for_logprob_per_req: int = -1
    future_indices: Optional[torch.Tensor] = None
```

python/sglang/srt/models/deepseek_nextn.py

重定向 prev_topk_indices 来源和 topk_indices 写回目标到 spec_info.mtp_topk_indices。

```
# 来自 python/sglang/srt/models/deepseek_nextn.py
# 在 decoder forward 中传递和写回 topk_indices
with get_global_expert_distribution_recorder().disable_this_region():
    hidden_states, residual, topk_indices = self.decoder(
        positions,
        hidden_states,
```

```

forward_batch,
residual,
zero_allocator,
prev_topk_indices=(
    # 从 spec_info 读取, 而非 forward_batch
    forward_batch.spec_info.mtp_topk_indices
    if forward_batch.reuse_mtp_topk_indices
    else None
),
)
)
if forward_batch.reuse_mtp_topk_indices:
    # 写回到 spec_info, 确保跨步保留
    forward_batch.spec_info.mtp_topk_indices = topk_indices

```

python/sglang/srt/speculative/eagle_worker_v2.py

调整初始化与清理代码, 使用 `spec_info.mtp_topk_indices` 替代 `forward_batch.topk_indices`。

```

# 来自 python/sglang/srt/speculative/eagle_worker_v2.py
# draft_forward 方法中的初始化与清理
if self.index_share_for_mtp_iteration:
    forward_batch.reuse_mtp_topk_indices = True
    # 初始化 mtp_topk_indices 在 spec_info 上
    spec_info.mtp_topk_indices = None

for i in range(self.speculative_num_steps):
    # ... 草稿循环 ...
    if self.index_share_for_mtp_iteration:
        # 清理
        spec_info.mtp_topk_indices = None
        forward_batch.reuse_mtp_topk_indices = False

```

评论区精华

机器人审查者 `gemini-code-assist[bot]` 建议使用 `getattr` 防御性访问

`forward_batch.spec_info.mtp_index_topk` 并在写回前检查 `None`, 以防 `spec_info` 为 `None` 或缺少属性导致 `AttributeError`。但此建议未被采纳, 因为 `reuse_mtp_topk_indices` 门控确保只有启用时才进入该路径, 且 `spec_info` 在草稿步期间始终有效。实际 PR 中字段名也为 `mtp_topk_indices` 而非 `mtp_index_topk`。

- 防御性编程: 使用 `getattr` 并检查 `None` (design): 未采纳。因为 `reuse_mtp_topk_indices` 门控确保只有启用时才进入该路径, 且 `spec_info` 在草稿步期间始终有效。实际字段名为 `mtp_topk_indices`。

风险与影响

- 风险: 低风险。仅在 EAGLE MTP 草稿的 `index_share_for_mtp_iteration` 特性启用时影响逻辑; 不启用时行为完全不变。变更涉及数据契约 (字段迁移), 但已在 4 个文件中一致更新, 无残留引用。缺少自动化测试覆盖, 但手动验证了加速效果和输出等价性。

- 影响：影响范围限于使用 EAGLE MTP 草稿且启用 `index_share_for_mtp_iteration` 的 DeepSeek 模型。长上下文（~500k ISL）场景下每个草稿步加速约 1.88 倍（4.167 ms → 2.215 ms），短上下文（~80k）加速较小。接受长度无影响（4.64 vs 4.65），草案质量保持。
- 风险标记：缺少测试覆盖

关联脉络

- PR #28192 related to `index_share_for_mtp_iteration` feature: 本 PR 修复了 #28192 引入的 `index_share_for_mtp_iteration` 特性中的 bug。