# PR #26397 完整报告

sgl-project/sglang

Reland "[perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1 (#26235)"

合并时间：2026-05-27 05:14

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26397

## 执行摘要

- 一句话：重做 EAGLE 草稿 topk==1 softmax 跳过优化并修复 AMD 回归
- 推荐动作：该 PR 是一次典型的重做优化并修复平台兼容性的案例，值得关注其平台门控的设计模式。虽然改动小，但涉及回退与重做决策，以及通过代码注释记录回归原因，具有良好的可维护性。建议精读以了解投机解码性能优化与平台差异处理。

## 功能与动机

EAGLE 投机解码在 topk == 1 时，草稿树退化为单路径，下游不再使用 topk_p 概率值，因此可以跳过高开销的 softmax 和 topk 重排序，直接取 logits 的 argmax 以提升性能。之前版本（PR #26235）已实现该优化，但引发了 DeepSeek-V3.2 MTP 在 ROCm 7.2 mi35x 上的严重精度回归（GSM8K 准确率从 0.975 跌至 0.035）。回归分析确认根因为 argmax 在 FP8 logits 上的断点行为与 softmax+max 路径不同。当前 PR 通过平台门控（CUDA 使用 argmax，ROCm 保持原 softmax 路径）在获得性能收益的同时避免回归。

## 实现拆解

1. 在 *eagle_worker_v2.py* 中添加平台门控的 argmax 分支：在 draft_forward() 和 _draft_extend_for_decode() 两个方法中，计算 topk_index/topk_p 时，若 self.topk == 1 and not _is_hip 为 True，则使用 torch.argmax 替代 torch.softmax + fast_topk，并将 topk_p 设置为全 1。否则维持原逻辑。
2. 在 *eagle_draft_extend_cuda_graph_runner.py* 中添加相同门控：在 run_once() 函数（CUDA Graph 捕获的回调）中，应用相同条件分支，保证 CUDA Graph 捕获路径也跳过 softmax。同时在该文件顶部增加 from sglang.srt.utils import is_hip 和全局变量 _is_hip = is_hip()。
3. 平台区分的意图：仅 CUDA 启用 argmax 优化，ROCm 上保持 softmax 路径，避免 FP8 下 argmax 断点不同导致的草稿选择错误。
4. 测试验证：PR 讨论中触发了 4-gpu-b200 测试（DeepSeek-V3.2 FP4 MTP）通过；但 AMD nightly 精度测试未自动触发，需人工确认。

关键文件：

- python/sglang/srt/speculative/eagle_worker_v2.py（模块 投机解码；类别 source；类型 core-logic；符号 draft_forward, _draft_extend_for_decode）：核心变更文件：在 draft_forward 和 _draft_extend_for_decode 两个方法中添加了 topk==1 时 argmax 跳过

softmax 的逻辑，并包含平台门控。

- python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py（模块 投机解码；类别 source；类型 core-logic；符号 run_once）：CUDA Graph 捕获路径也需同步优化，并在文件头增加 is_hip 判断和全局变量 _is_hip。

关键符号：draft_forward, _draft_extend_for_decode, run_once

## 关键源码片段

### python/sglang/srt/speculative/eagle_worker_v2.py

核心变更文件：在 draft_forward 和 _draft_extend_for_decode 两个方法中添加了 topk==1 时 argmax 跳过 softmax 的逻辑，并包含平台门控。

```
# file: sglang/srt/speculative/eagle_worker_v2.py
# ... inside draft_forward loop ...
if self.topk == 1 and not _is_hip:
    # topk=1 → 退化单路径树；topk_p 下游未使用，
    # 因此跳过 softmax，直接对 logits 做 argmax。
    # 仅在 CUDA 上启用：ROCm 的 argmax 断点行为与
    # softmax+max 路径在 FP8 logits 上不同，会导致
    # MTP 草稿选择错误（DSV3.2 MTP GSM8K，见 #26358）。
    topk_index = torch.argmax(
        logits_output.next_token_logits, dim=-1, keepdim=True
    )
    topk_p = torch.ones_like(topk_index, dtype=torch.float32)
else:
    probs = torch.softmax(logits_output.next_token_logits, dim=-1)
    topk_p, topk_index = fast_topk(probs, self.topk, dim=-1)
```

### python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py

CUDA Graph 捕获路径也需同步优化，并在文件头增加 is_hip 判断和全局变量 _is_hip。

```
# file: sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py
# 新增导入和全局变量
from sglang.srt.utils import (
    is_hip,
    require_attn_tp_gather,
    require_gathered_buffer,
    require_mlp_sync,
    require_mlp_tp_gather,
)
_is_hip = is_hip()

# ... inside run_once() ...
# ROCm 的 argmax 断点行为与 CUDA 的 softmax+max
# 路径在 FP8 logits 上不同，这会影响 AMD 上 MTP 草稿选择。
# 仅在 CUDA 上使用快速路径。
if self.topk == 1 and not _is_hip:
    ret.topk_index = torch.argmax(
```

```
            ret.next_token_logits, dim=-1, keepdim=True
        )
        ret.topk_p = torch.ones_like(ret.topk_index, dtype=torch.float32)
    else:
        probs = torch.softmax(ret.next_token_logits, dim=-1)
        ret.topk_p, ret.topk_index = fast_topk(probs, self.topk, dim=-1)
```

## 评论区精华

核心讨论围绕 ROCm 平台回归问题展开。在 #26358 回退后，michaelzhang-ai 指出："R108 verify on the revert branch just confirmed the revert recovered MTP from 0.035 → 0.975 on the exact same hardware + aiter pin, so the cause is the topk==1 path itself"。因此当前 PR 通过 _is_hip 门控将 argmax 优化限制在 CUDA 上，解决回归问题。无其他设计争议。

- ROCm 平台回归问题与平台门控的引入 (correctness): 新增 _is_hip 门控，仅在 CUDA 上启用 argmax 优化，ROCm 上保持原 softmax 路径。

## 风险与影响

- 风险：
  1. 回归风险（AMD）：虽然已通过 _is_hip 门控避免 ROCm 上的回归，但若未来 AMD 驱动 / 库更新修复了 argmax 行为，此门控会成为性能优化的障碍。需要持续跟踪。
  2. CUDA 路径验证不足：CUDA 上的 argmax 优化缺少直接测试覆盖（如独立的精度或性能测试），依赖现有集成测试验证。
  3. topk==1 假设：优化假设 topk==1 时 topk_p 下游不被使用，若未来逻辑变化需同步更新此假设。 - 影响：对 CUDA 用户：投机解码草稿生成推理速度提升（跳过大 softmax）；对 ROCm 用户：无功能影响。改动集中在草稿生成阶段，不影响推理结果正确性（CUDA 上 argmax 与 softmax+max 等价）。影响范围限于使用 EAGLE 投机解码且 topk==1 的模型（如 DeepSeek 系列）。 - 风险标记：平台兼容修复，性能优化，缺少测试覆盖

## 关联脉络

- PR #26235 [perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1: 被回退的原始优化 PR，当前 PR 是其重做版本。
- PR #26358 Revert "[perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1 (#26235)": 因 ROCm 回归而回退 #26235，当前 PR 直接 revert 此回退后增加修复。