

PR #34238 完整报告

sgl-project/sglang

[AMD] Broadcast the EAGLE greedy verify decision across TP ranks on ROCm

合并时间: 2026-08-15 13:37

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34238>

执行摘要

- 一句话: ROCm 上广播 EAGLE verify 决策, 修复 TP 死锁
- 推荐动作: 值得精读。该 PR 虽小, 但揭示了分布式投机解码中一个隐蔽的跨 rank 一致性陷阱, 并展示了如何通过提升同步点作用域来覆盖所有执行路径。对理解 ROCm/CUDA 平台差异导致的死锁原因、以及 TP 集合通信前需要保证状态一致的原则很有价值。建议关注后续是否补充自动化测试。

功能与动机

PR body 明确指出: On ROCm with TP > 1, EAGLE speculative decoding can intermittently deadlock. Observed on 4x MI355X (gfx950), GLM-4.7-Flash, TP4, with 2 deadlocks in 5 runs。根因是 `_is_hip` 使 `eagle_sample()` 总走 greedy verify 路径, 而该路径在不同 rank 上可能收到不同 draft 候选; 采样路径原本已从 rank 0 广播结果, greedy 路径却没有, 导致 rank 间接受 draft 数不一致, 最终在 `seq_lens` / batch 组成上发散而死锁。

实现拆解

1. 问题定位: 在 `python/sglang/srt/speculative/eagle_utils.py` 的 `eagle_sample()` 中, ROCm 因 `_is_hip` 恒真而总走 greedy verify 分支, 该分支缺少采样路径已有的跨 rank 广播。
2. 核心变更: 将原本位于采样路径内部的 `tp_group.broadcast()` 块上移, 使其位于 `sampling_fn()` 调用之后、`SIMULATE_ACC_LEN` 处理之前, 从而同时覆盖 greedy verify 与采样两条路径。
3. 同步对象: 从 rank 0 广播 `predict`、`accept_index`、`num_correct_drafts` 三个可变张量, 保证所有 TP rank 对本次 verify 的接受结果一致。
4. 并行组选择: 保留 `is_dp_attention_enabled()` 分支, DP attention 场景下使用 `attn_tp_group`, 否则使用 `get_tp_group()`, 确保广播作用于正确的进程组。
5. 验证与配套: 真机 4x MI355X、GLM-4.7-Flash、TP4 验证, 死锁率从 5 次中 2 次降到 6 次中 0 次; 未新增自动化单元测试, 注释同步更新以解释原理。

关键文件:

- `python/sglang/srt/speculative/eagle_utils.py` (模块 投机解码; 类别 source; 类型 core-logic; 符号 `eagle_sample`): 核心修复文件: 将 EAGLE verify 决策广播从采样路径提升到公共路径, 覆盖 ROCm greedy verify 分支, 修复 TP rank 间 `seq_lens` 发散导致的

死锁。

关键符号: eagle_sample

关键源码片段

python/sglang/srt/speculative/eagle_utils.py

核心修复文件: 将 EAGLE verify 决策广播从采样路径提升到公共路径, 覆盖 ROCm greedy verify 分支, 修复 TP rank 间 seq_lens 发散导致的死锁。

eagle_sample() 内部, verify 采样完成后、SIMULATE_ACC_LEN 处理前

```
sampling_fn(
    predicts=predict, # mutable: 各位置的预测 token
    accept_index=accept_index, # mutable: 接受的 draft 位置索引
    accept_token_num=num_correct_drafts, # mutable: 接受的 draft 数量
    candidates=candidates,
    retrieve_index=verify_input.retrieve_index,
    retrieve_next_token=verify_input.retrieve_next_token,
    retrieve_next_sibling=verify_input.retrieve_next_sibling,
    uniform_samples=coins,
    uniform_samples_for_final_sampling=coins_for_final_sampling,
    target_probs=target_probs,
    draft_probs=draft_probs,
    threshold_single=get_spec().speculative_accept_threshold_single,
    threshold_acc=get_spec().speculative_accept_threshold_acc,
    deterministic=True,
)

# 跨 TP rank 同步 verify 决策:
# 各 rank 送入本函数的张量存在微小差异 (如 AMD all-reduce 非逐位一致、
# softmax/top_k 浮点不确定), 使不同 rank 接受不同数量的 draft;
# 这会导致提交的 seq_lens 与 batch 组合发散, 下一次 TP 集合通信即死锁。
# 故从 rank 0 广播三个关键张量, 保证各 rank 决策一致。
tp_group = (
    get_parallel().attn_tp_group if is_dp_attention_enabled() else get_tp_group()
)
if tp_group.world_size > 1:
    tp_group.broadcast(predict, src=0)
    tp_group.broadcast(accept_index, src=0)
    tp_group.broadcast(num_correct_drafts, src=0)
```

评论区精华

HaiShaw 在 review 中建议将广播块上移以覆盖所有路径:

Can we move this block under L736, or simply move block L824-L836 out (indent -> left) to cover all cases?

作者回应:

Done. Moved this block one level up to cover both paths, and updated the comment accordingly.

该建议避免了仅对 ROCm greedy 分支打补丁的窄修复，从结构上保证了未来新增分支也能自动获得跨 rank 一致性，最终被采纳并获 approve。

- 广播块位置是否应覆盖所有路径 (design): 接受建议，广播块提升到采样与 greedy 两个路径共用的位置，结构上更稳健。

风险与影响

- 风险：
 - 性能风险：eagle_sample() 每次调用将额外执行 3 次进程组广播 (predict、accept_index、num_correct_drafts)，在 TP 规模较大时增加同步开销。但该广播发生在 speculative verify 后、下一次集合通信前，原本就是同步点，实际额外开销有限。
 - 正确性风险：广播以 rank 0 为准，若 rank 0 本身决策异常（如显存错误）会被放大到全体 rank，但本场景 rank 0 无特殊偏差，风险可接受。
 - 兼容性风险：广播现在对 CUDA 的 greedy 路径也生效（原本仅采样路径），可能引入微小额外同步，但不会改变 CUDA 行为（greedy 路径在 CUDA 上本身也需一致性）。
 - 测试风险：未添加自动化单测，回归依赖真机验证；后续若 AMD all-reduce 行为变化或并行组结构调整，可能重新引入同类问题。
- 影响：
 - 用户影响：AMD ROCm 上启用 TP>1 且使用 EAGLE/MTP 投机解码的用户是直接受益者，间歇性死锁被消除；CUDA 用户功能不变，仅有轻微广播开销。
 - 系统影响：涉及 eagle_utils.py 中投机解码核心路径，所有使用 EAGLE 后端的部署都会执行该广播逻辑（TP>1 时），属于调度热路径上的行为调整。
 - 团队影响：为后续修复其他平台（如 NPU）的投机一致性提供了参考模式；统一广播位置降低了维护复杂度。
 - 风险标记：核心路径变更，缺少测试覆盖，平台相关（ROCm）

关联脉络

- PR #34844 [Spec] Support MegaMoE for DSpark under dp attention: 同属 speculative decoding 在并行组（DP attention / TP）下的适配，该 PR 也涉及 is_dp_attention_enabled() 相关的组选择逻辑，与本 PR 的广播组选择相互关联。