PR #47466 完整报告

vllm-project/vllm

[Bugfix] Fix PD disagg + MTP correctness for Qwen3.5(GDN)

合并时间：2026-07-07 21:51

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47466

执行摘要

- 一句话：修复 PD 分离下 GDN 状态被错误覆盖的准确率 bug
- 推荐动作：值得精读。该 PR 展示了如何在复杂的推测解码 + PD 分离场景下识别并修复状态机错误。两种修复方案的权衡（调度器 vs. kernel）值得学习。同时展示了通过收紧判断条件而非禁用优化来解决问题的方法。

功能与动机

Qwen3.5-0.8B 在 PD 分离 + MTP 推测解码 + 默认 cudagraph 场景下，当并发大于 1 时 GSM8K 准确率从约 0.32 跌至约 0.15-0.19。根因分析显示，decode worker 在收到新移交的请求时，prompt 尾 token xN 尚未完成前向，但调度器错误地将其标记为 decode 步骤，导致输入组装 kernel 将上一个采样 token 覆盖到 xN 位置，从而使 GDN 递归状态从错误 token 开始演化。详见 PR body。

实现拆解

1. 调度器中 decode 判断条件修正（gpu_model_runner.py_prepare_inputs）：将判断请求是否为 decode 步骤的条件从 num_computed_tokens >= num_prompt_tokens 改为 num_scheduled_tokens[req_idx] == draft_len + 1，确保仅当该步骤恰好只有一个非推测 token（即真正的 decode 步骤）时才标记为 spec-decode 行。
2. 输入组装 kernel 保护（input_batch.py_combine_sampled_and_draft_tokens_kernel）：增加 first_logit_seq_pos >= prefill_len 条件，使得 prompt 尾 token 对应的 logit 位置不会被 last_sampled_token 覆盖，保留正确的输入 token。
3. 注意力元数据对齐（mamba_hybrid.pyprepare_attn_metadata）：将 spec-decode 行的判断从基于 is_prefilling_np 改为基于 num_scheduled_tokens == num_draft_tokens_per_req + 1，与调度器条件对齐。
4. 移除冗余状态播种（states.pyadd_request）：删除对 last_sampled_tokens 的播种逻辑（之前用于 PD 场景设置第一个 decode 输入），因为 kernel 保护已足够。
5. 测试配套：修改 run_accuracy_test.sh 和 config_sweep_accuracy_test.sh，添加 ENFORCE_EAGER 变量和 MTP+PD 配置变体，覆盖 cudagraph 场景的回归测试。

关键文件：

- vllm/v1/worker/gpu/input_batch.py（模块 输入批处理；类别 source；类型 core-logic；符号 _combine_sampled_and_draft_tokens_kernel,

combine_sampled_and_draft_tokens）：修改了输入组装 kernel，添加条件保护 prompt尾 token 不被 last_sampled_token 覆盖，是核心修复之一。

- vllm/v1/worker/gpu_model_runner.py（模块 模型执行器；类别 source；类型 core-logic；符号 _prepare_inputs）：修改了调度器条件，从基于 num_computed_tokens 改为基于 num_scheduled_tokens，确保只有真正的 decode 步骤才标记为 spec-decode。
- vllm/v1/worker/gpu/model_states/mamba_hybrid.py（模块 注意力元数据；类别 source；类型 data-contract；符号 prepare_attn_metadata）：注意力元数据中 spec-decode mask 的判断条件与调度器对齐，确保只有真正的 decode 行被标记。
- vllm/v1/worker/gpu/states.py（模块 状态管理；类别 source；类型 core-logic；符号 add_request）：移除了 last_sampled_tokens 的播种逻辑，简化状态初始化并避免潜在冲突。
- tests/v1/kv_connector/nixl_integration/run_accuracy_test.sh（模块 集成测试；类别 test；类型 test-coverage）：测试脚本支持 ENFORCE_EAGER 变量以覆盖 cudagraph 场景的准确率测试。
- tests/v1/kv_connector/nixl_integration/config_sweep_accuracy_test.sh（模块 配置扫描；类别 test；类型 test-coverage）：添加了 MTP+PD 的配置变体，确保回归测试覆盖此场景。

关键符号：_combine_sampled_and_draft_tokens_kernel, _prepare_inputs, prepare_attn_metadata, add_request

关键源码片段

vllm/v1/worker/gpu/input_batch.py

修改了输入组装 kernel，添加条件保护 prompt 尾 token 不被 last_sampled_token 覆盖，是核心修复之一。

```
# head 版本：_combine_sampled_and_draft_tokens_kernel
# 关键变更：增加 first_logit_seq_pos 和条件保护
seq_len = tl.load(seq_lens_ptr + batch_idx)
prefill_len = tl.load(prefill_len_ptr + req_state_idx)
if seq_len <= prefill_len:
    # Handling prefill tokens. No sampled or draft tokens.
    return

# Keep prompt-tail slots intact; only rewrite generated-token slots.
first_logit_seq_pos = seq_len - num_logits
if NUM_NEW_SAMPLED_TOKENS > 0 and first_logit_seq_pos >= prefill_len:
    # Write the last sampled token ID to input_ids.
    last_token_id = tl.load(last_sampled_tokens_ptr + req_state_idx)
    tl.store(input_ids_ptr + logits_start, last_token_id)
```

vllm/v1/worker/gpu_model_runner.py

修改了调度器条件，从基于 num_computed_tokens 改为基于 num_scheduled_tokens，确保只有真正的 decode 步骤才标记为 spec-decode。

```
# head 版本：_prepare_inputs 中 spec-decode 条件段
# 原条件：num_computed_tokens >= num_prompt_tokens
# 新条件：num_scheduled_tokens[req_idx] == draft_len + 1
for req_id, draft_token_ids in scheduler_output.scheduled_spec_decode_tokens.items():
    req_idx = self.input_batch.req_id_to_index[req_id]
    draft_len = len(draft_token_ids)
    num_draft_tokens[req_idx] = draft_len
    # 关键修复：仅当该步骤恰好只调度了 1 个非推测 token 时视为 decode
    if num_scheduled_tokens[req_idx] == draft_len + 1:
        num_decode_draft_tokens[req_idx] = draft_len
```

vllm/v1/worker/gpu/model_states/mamba_hybrid.py

注意力元数据中 spec-decode mask 的判断条件与调度器对齐，确保只有真正的 decode 行被标记。

```
# head 版本：prepare_attn_metadata 中 spec_decode_mask 段
num_draft_tokens_per_req = input_batch.num_draft_tokens_per_req
if num_draft_tokens_per_req is not None:
    # A row is a spec-decode row only when its whole prompt is already
    # computed, i.e. exactly one non-draft (decode) token is scheduled.
    is_decode = (
        input_batch.num_scheduled_tokens == num_draft_tokens_per_req + 1
    )
    spec_decode_mask = (num_draft_tokens_per_req > 0) & is_decode
    num_decode_draft_tokens_np[: input_batch.num_reqs] = np.where(
        spec_decode_mask, num_draft_tokens_per_req, -1
    )
```

评论区精华

1. 与 #45237 的兼容性：作者 andakai 询问 qianlihuang 在调度器条件中增加 num_computed_tokens >= request.num_prompt_tokens 是否会影响之前的 PR（#45237）。讨论后决定采用保持调度器优化、收紧 kernel 的替代方案。
2. MRV2 SSM 等效修复：njhill 指出需要相同修复用于 MRV2 SSM spec decode 条件，并移除了冗余的 last_sampled_tokens 播种。他和 andakai 确认后合入该改进。

- 与 PR #45237 兼容性讨论 (question): 讨论后决定采用替代方案：保持调度器优化，收紧 kernel 条件，避免影响原 PR。
- MRV2 SSM 等效修复与冗余代码移除 (correctness): njhill 提交了补充 commit，作者确认后合并。

风险与影响

- 风险：风险较低。修复只影响 PD 分离 + MTP 推测解码 + GDN 状态模型（如 Qwen3.5）的边界条件。正常解码和纯 prefill 路径不受影响。但需关注调度器条件从 num_computed_tokens 切换到 num_scheduled_tokens 是否会影响其他推测解码方法（如 Eagle/Medusa）；已通过 njhill 补充 MRV2 SSM 的相关修复，但未覆盖所有模型。测试配置已增强，但仍需持续监控。

- 影响：对使用 PD 分离 + MTP + GDN 模型的用户：准确率大幅提升，回归正常。其他用户无影响。测试脚本也相应增强，支持 cudagraph 场景的准确率测试。团队需要确保未来的调度器变更不破坏此边界条件。
- 风险标记：调度器条件重构，MTP 路径修正，需要验证其他推测解码方法

关联脉络

- PR #45237 [Bugfix] Fix PD disagg + spec decode combined scheduler path: 该 PR 引入了原始的调度器 MTP padding 优化，本 PR 与之紧密相关，需要确保兼容性。讨论中明确提到了该 PR。