PR #24217 完整报告

sgl-project/sglang

fix: STANDALONE spec-decode hidden-size mismatch crash

合并时间：2026-05-11 08:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24217

执行摘要

- 一句话：修复目标 / 草稿隐藏层大小不匹配时的推测解码崩溃
- 推荐动作：值得精读，特别是对推测解码内部形状管理的分析。展示了如何诊断和修复一个 phase 边界上的形状问题。建议关注 *decode* 与 *extend* 阶段不同形状处理的决策逻辑，以及未来可能的清除计划。

功能与动机

源于 Issue #24215。在 STANDALONE 推测解码下，当目标模型和草稿模型的隐藏层大小不同时，`EagleDraftInput.merge_batch` 在拼接两个张量时抛出 `RuntimeError: Sizes of tensors must match except in dimension 0. Expected size 4096 but got size 2048`。根因是空闲占位符使用了草稿模型的 `spec_hidden_size` 而非目标模型的，导致形状不匹配。

实现拆解

1. 诊断问题：确定两个地点错误地使用了草稿模型配置：*eagle_worker.py* 中的 *forward_draft_extend_after_decode* 空闲回退，以及 *eagle_draft_extend_cuda_graph_runner.py* 中的 CUDA graph 缓冲区创建。
2. 修复空闲回退：在 *eagle_worker.py* 中，从 self.model_config 改为 self.target_worker.model_runner.model_config 获取 hidden_size 和 dtype。
3. 修复 CUDA graph 缓冲区：在 *eagle_draft_extend_cuda_graph_runner.py* 中，类似地改用 target config。
4. 保留 *decode* 阶段形状：*decode* 阶段保持 draft 形状，因为调度器期望此形状；如果翻转会导致崩溃。
5. 测试：添加了 stage-b 测试 test_standalone_hidden_size_regression.py 直接验证该路径。

关键文件：

- python/sglang/srt/speculative/*eagle_worker.py*（模块 推测解码；类别 source；类型 *core-logic*；符号 forward_draft_extend_after_decode）：核心修复位置：修改了 *forward_draft_extend_after_decode* 方法中的空闲回退部分，使用目标模型配置创建占位符
- python/sglang/srt/speculative/*eagle_draft_extend_cuda_graph_runner.py*（模块 推测解码；类别 source；类型 *core-logic*）：CUDA graph 缓冲区创建时使用目标模型配置，确保隐藏状态缓冲区形状正确

关键符号：eagle_worker.EAGLEWorker.forward_draft_extend_after_decode，eagle_draft_extend_cuda_graph_runner.EagleDraftExtendCudaGraphRunner.init

关键源码片段

python/sglang/srt/speculative/eagle_worker.py

核心修复位置：修改了 forward_draft_extend_after_decode 方法中的空闲回退部分，使用目标模型配置创建占位符

```
    def forward_draft_extend_after_decode(
        self, batch: ScheduleBatch
    ) -> EagleDraftInput:
        draft_extend_input: EagleDraftExtendInput = batch.spec_info
        # ... 前面的备份代码 ...
        if not input_is_idle and draft_extend_input.input_ids.shape[0] == 0:
            # 所有请求已完成验证，切换到空闲 ExtendInput
            batch = batch.copy()
            batch.prepare_for_idle()
            # fix: 使用目标模型配置而不是草稿模型配置
            target_cfg = self.target_worker.model_runner.model_config
            hidden_size = (
                target_cfg.hidden_size * 3
                if self.speculative_algorithm.is_eagle3()
                and self.eagle_use_aux_hidden_state
                else target_cfg.spec_hidden_size
            )
            draft_extend_input = EagleDraftExtendInput.create_idle_input(
                device=self.device,
                hidden_size=hidden_size,
                dtype=target_cfg.dtype,
                capture_hidden_mode=CaptureHiddenMode.LAST,
            )
            batch.spec_info = draft_extend_input
        # ... 后续逻辑 ...
```

python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py

CUDA graph 缓冲区创建时使用目标模型配置，确保隐藏状态缓冲区形状正确

```
            if (
                self.eagle_worker.speculative_algorithm.is_eagle3()
                and self.eagle_worker.eagle_use_aux_hidden_state
            ):
                # EAGLE3 分支：使用目标 hidden_size*3 或 target_hidden_size*3
                # ...
            else:
                # 非 EAGLE3：hidden_states 携带目标模型输出，使用目标配置
                target_cfg = self.eagle_worker.target_worker.model_runner.model_config
                hidden_states = torch.zeros(
                    (
```

```
            self.max_num_token,
            target_cfg.spec_hidden_size, # 原来用 self.model_runner.model_config.spec_
            hidden_size
        ),
        dtype=target_cfg.dtype, # 原来用 self.model_runner.dtype
    )
```

评论区精华

Reviewer kpham-sgl 在测试文件上要求更强测试（*Need a stronger test*），作者 brian030128 随后添加了 stage-b GPU 回归测试直接触发崩溃路径。gemini-code-assist[bot] 的自动 code review 确认变更正确。PR body 中作者详细分析了 decode 和 extend 阶段不同形状的设计权衡，并指出未来任务 #21058 将彻底消除 hidden_states 的携带。

- 增加回归测试 (testing): 添加了 stage-b 测试，并移除了原来的单元测试。PR 合并前 CI 全部通过。

风险与影响

- 风险：风险较低。修复针对两个特定地点，且已有 capture_for_decode 归一化作为保护。需注意：空闲回退仅在 enable_dp_attention=True 时可达；CUDA graph 缓冲区形状不匹配先前会静默跳过 copy_，可能隐藏问题。未来 PR #21434 将完全消除 STANDALONE 路径中的 hidden_states，届时这些修复代码会成为死代码。
- 影响：影响范围仅限使用了不同隐藏层大小的目标 / 草稿模型对的 STANDALONE 推测解码用户。修复后这些用户不再遇到崩溃。对其他配置和模型无影响。团队在推测解码代码的健壮性和可维护性方面获得提升。
- 风险标记：核心路径变更，特殊配置触发，测试覆盖回归路径

关联脉络

- 暂无明显关联 PR