PR #26981 完整报告

sgl-project/sglang

Revert "Support spec v2 tree drafting (eagle topk>1) with page_size==1"

合并时间：2026-06-02 08:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26981

执行摘要

- 一句话：回退 spec v2 topk>1 树形 draft 支持，修复默认行为破坏
- 推荐动作：阅读者应关注此 revert 背后的测试覆盖率不足问题：原始 PR #26866 的 CI 状态为 ⚠️，仍被合并，导致默认行为破坏。建议加强 speculative 模块的自动化测试，特别是 topk>1 与 spec v2 的组合场景。回退本身逻辑清晰，值得参考的是 fill_bonus_tokens stride 参数的修正——用 accept_index.shape[1] 而非 speculative_num_draft_tokens 是导致 topk>1 错误的根本原因之一。

功能与动机

PR 作者在 body 中指出 #26866 'breaks the default behaviors when we use topk > 1 under the default spec v2 settings'，因此直接回退以恢复 main 分支稳定性。

实现拆解

步骤 1：核心逻辑回退（eagle_worker_v2.py）

- 删除 _finalize_accepted_tree_path 和 _compact_accepted_to_front 方法，这两者用于 topk>1 时压缩 accepted 路径到每个请求块前端。
- 将 fill_bonus_tokens 的 stride 参数从 accept_index.shape[1] 改回 self.speculative_num_draft_tokens，修复 topk>1 时 bonus token 读取越界。
- 移除 verify 中 self.topk > 1 条件下的路径压缩调用。

步骤 2：配置入口调整（arg_groups/speculative_hook.py）

- _handle_eagle_family 中移除 page_size > 1 的额外条件，当 topk > 1 时直接禁用 overlap schedule（强制 spec v1），回退原因表述从 'spec v2 topk > 1 currently requires page_size == 1' 变更为 'spec v2 currently only supports topk = 1'。

步骤 3：Triton kernel 与多层级 worker 适配（eagle_info_v2.py, multi_layer_eagle_worker_v2.py）

- fill_bonus_tokens Triton kernel 的参数名从 accept_stride 改为 num_draft_tokens，意义更明确。
- multi_layer_eagle_worker_v2.py 中的调用同步回退。

步骤4：测试用例移除（test_spec_eagle_stress.py,test_spec_eagle_topk.py）

- 删除 TestEagle3Topk16V2Retract 和 TestEagle3Topk16SpecV2 两个测试类，它们专门用于验证 spec v2 下的 topk>1 路径。
- 下调测试预估时间。

关键文件：

- python/sglang/srt/speculative/eagle_worker_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 _finalize_accepted_tree_path, _compact_accepted_to_front, verify, move_accepted_tokens_to_target_kvcache）：核心 speculative worker，包含 verify、bonus token 处理及 accepted 路径压缩；revert 删除了 topk>1 压缩路径的两个方法，修正了 fill_bonus_tokens 的 stride 参数
- python/sglang/srt/arg_groups/speculative_hook.py（模块 参数配置；类别 source；类型 core-logic；符号 _handle_eagle_family）：配置入口，控制 spec v1/v2 选择；revert 后 unconditional 回退 topk>1 到 spec v1
- python/sglang/srt/speculative/eagle_info_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 fill_bonus_tokens）：包含 fill_bonus_tokens Triton kernel；参数名从 accept_stride 改为 num_draft_tokens
- python/sglang/srt/speculative/multi_layer_eagle_worker_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 verify）：多层级 worker 中 fill_bonus_tokens 调用同步回退
- test/registered/spec/eagle/test_spec_eagle_stress.py（模块 测试；类别 test；类型 test-coverage；符号 TestEagle3Topk16V2Retract）：删除 TestEagle3Topk16V2Retract 测试类，该测试专门验证 v2 下 topk>1 retract
- test/registered/spec/eagle/test_spec_eagle_topk.py（模块 测试；类别 test；类型 test-coverage；符号 TestEagle3Topk16SpecV2）：删除 TestEagle3Topk16SpecV2 测试类，该测试验证 v2 下 topk=16 树形 draft

关键符号：_finalize_accepted_tree_path, _compact_accepted_to_front, fill_bonus_tokens, move_accepted_tokens_to_target_kvcache, _handle_eagle_family, verify

关键源码片段

python/sglang/srt/speculative/eagle_worker_v2.py

核心 speculative worker，包含 verify、bonus token 处理及 accepted 路径压缩；revert 删除了 topk>1 压缩路径的两个方法，修正了 fill_bonus_tokens 的 stride 参数

```
# 在 verify 方法中，fill_bonus_tokens 的 stride 参数从 accept_index.shape[1] 改回
# speculative_num_draft_tokens，因为对于 topk>1，accept_index.shape[1] 不等于
# num_draft_tokens，导致 bonus token 读取越界。
if not batch.forward_mode.is_idle():
    accept_tokens = predict[accept_index]
    bonus_tokens = torch.empty_like(accept_lens, dtype=torch.int32)
    fill_bonus_tokens[(bs,)](
        accept_tokens,
        accept_lens,
```

```
            bonus_tokens,
            self.speculative_num_draft_tokens,
        )
    else:
        bonus_tokens = torch.empty((0,), device=self.device, dtype=torch.int32)

    # 移除了 topk>1 时的 accepted tree path 压缩逻辑，
    # 因为该逻辑在 spec v2 中尚不稳定，且破坏默认行为。
    # 原代码（已删除）：
    # if not batch.forward_mode.is_idle() and self.topk > 1:
    # predict = self._finalize_accepted_tree_path(...)

    next_draft_input = EagleDraftInput(bonus_tokens=bonus_tokens)
    return GenerationBatchResult(
        ...
    )
```

python/sglang/srt/arg_groups/speculative_hook.py

配置入口，控制 spec v1/v2 选择；revert 后 unconditional 回退 topk>1 到 spec v1

```
    # 回退后，只要 topk>1 就禁用 overlap schedule（强制 spec v1），
    # 不再检查 page_size。
    def _handle_eagle_family(server_args: "ServerArgs") -> None:
        ...
        spec_v1_reason = None
        if (
            server_args.speculative_eagle_topk is not None
            and server_args.speculative_eagle_topk > 1
            and not server_args.disable_overlap_schedule
        ):
            # 原 PR #26866 曾允许 page_size==1 时走 v2，现在全面回退
            server_args.disable_overlap_schedule = True
            spec_v1_reason = "spec v2 currently only supports topk = 1"
        elif (
            not envs.SGLANG_ENABLE_SPEC_V2.get()
            and not server_args.disable_overlap_schedule
        ):
            server_args.disable_overlap_schedule = True
            spec_v1_reason = "SGLANG_ENABLE_SPEC_V2=False"
        ...
```

评论区精华

无 review 讨论；PR 作者直接回退并合并。

- 暂无高价值评论线程

风险与影响

- 风险：低风险：Revert 是直接回退已合并的变更，功能回到 #26866 之前的状态。主要风险在于未来再次引入类似功能时需要更充分的验证，但本次 revert 本身不会引入新问题。
- 影响：对用户：回退后，当 topk>1 且使用 spec v2 (overlap) 时，行为会回退到 spec v1，这修复了 #26866 导致的不稳定。对于依赖 #26866 功能（page_size==1 时 topk>1 走 v2）的用户，此功能被移除，需要等待更稳定的实现。对系统：减少了一条存在缺陷的逻辑路径，整体稳定性提升。
- 风险标记：回退功能，默认行为破坏修复，兼容性影响

关联脉络

- PR #26866 Support spec v2 tree drafting (eagle topk>1) with page_size==1: 本 PR 为 #26866 的完全回退，因其破坏了默认 spec v2 下 topk>1 的行为