PR #26866 完整报告

sgl-project/sglang

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

合并时间：2026-06-02 06:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26866

执行摘要

- 一句话：支持 page_size=1 时 spec v2 树形推导 (topk>1)
- 推荐动作：值得精读，尤其是 _finalize_accepted_tree_path 和 _compact_accepted_to_front 的实现，以及条件判断中 page_size 的约束决策。对于使用 speculative decoding 的团队，此 PR 修复了多个隐蔽 bug，建议尽快合入。review 中指出的 routed_experts_output 问题需在后续 PR 中跟进。

功能与动机

此前 spec v2 (overlap scheduler) 只支持 topk=1，topk>1 时会自动禁用 overlap 并回退到 spec v1，导致推理吞吐降低。通过限制 page_size==1（不涉及 partial-page duplication）即可安全启用 topk>1 树形推导，同时修复 v2 路径中已有的多个 bug，使顶层设计工作。

实现拆解

1. 条件修改（speculative_hook.py）：在 _handle_eagle_family 中新增 page_size > 1 检查，只有当 page_size>1 且 topk>1 时才强制 fallback 到 v1，page_size==1 时保留 overlap。
2. 核心逻辑扩展（eagle_worker_v2.py）：新增 _finalize_accepted_tree_path 方法，在 verify 后对 topk>1 的情况执行 accepted-path compaction；内部包含 move_accepted_tokens_to_target_kvcache 和 _compact_accepted_to_front。
3. Bug 修复：
   - bonus token stride 从 speculative_num_draft_tokens 改为 accept_index.shape[1]，避免树形结构下多读。
   - move_accepted_tokens_to_target_kvcache 中的 size 从 bs * num_draft_tokens 改为 bs * accept_index.shape[1]，消除 OOB。
   - 同步修改 fill_bonus_tokens kernel 参数名和语义。
4. 多层级 worker：multi_layer_eagle_worker_v2.py 中同步 bonus token stride 修复。
5. 测试新增：在 test_spec_eagle_topk.py 中添加 TestEagle3Topk16SpecV2（正确性 + logprob 校验），在 test_spec_eagle_stress.py 中添加 TestEagle3Topk16V2Retract（压力 retract 场景），均使用 page_size=1 默认配置。

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）：核心实现文件，新增 accepted-path compaction 逻辑并修复多个 bug。
- python/sglang/srt/arg_groups/speculative_hook.py（模块 推测解码；类别 source；类型 core-logic）：路由决策点，修改允许 page_size==1 时 topk>1 走 spec v2。
- test/registered/spec/eagle/test_spec_eagle_topk.py（模块 推测解码；类别 test；类型 test-coverage；符号 TestEagle3Topk16SpecV2）：新增 EAGLE3 topk=16 在 spec v2 下的正确性测试。

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

关键源码片段

python/sglang/srt/speculative/eagle_worker_v2.py

核心实现文件，新增 accepted-path compaction 逻辑并修复多个 bug。

```
def verify(self, batch: ScheduleBatch):
    # ... 采样逻辑 ...
    if not batch.forward_mode.is_idle():
        accept_tokens = predict[accept_index]
        bonus_tokens = torch.empty_like(accept_lens, dtype=torch.int32)
        # 关键修复：使用 accept_index.shape[1]（即 spec_steps+1）作为每请求宽度
        # 而不是 speculative_num_draft_tokens，后者在 tree drafting 时偏大
        fill_bonus_tokens[(bs,)](
            accept_tokens,
            accept_lens,
            bonus_tokens,
            accept_index.shape[1],
        )
    else:
        bonus_tokens = torch.empty((0,), device=self.device, dtype=torch.int32)

    # ... logprob 计算 ...

    # 新增：tree drafting (topk>1) 时 compact 已接受路径
    if not batch.forward_mode.is_idle() and self.topk > 1:
        predict = self._finalize_accepted_tree_path(
            batch, accept_index, accept_lens, predict, logits_output, bs
        )

    return GenerationBatchResult(
        # ...
        next_token_ids=predict,
        # ...
    )

def _finalize_accepted_tree_path(
```

```
    self,
    batch: ScheduleBatch,
    accept_index: torch.Tensor,
    accept_lens: torch.Tensor,
    predict: torch.Tensor,
    logits_output,
    bs: int,
) -> torch.Tensor:
    """
    将树形验证后的 accepted path 连续化（compact）到每个请求块的前端。
    KV 槽位通过 move_accepted_tokens_to_target_kvcache 搬移，
    predict 和 hidden_states 通过 _compact_accepted_to_front 重排。
    """
    # accept_lens 包含 bonus token，减 1 得到有效的 accepted draft 数量
    self.move_accepted_tokens_to_target_kvcache(
        batch, accept_index, accept_lens - 1
    )
    predict = self._compact_accepted_to_front(predict, accept_index, bs)
    if logits_output.hidden_states is not None:
        logits_output.hidden_states = self._compact_accepted_to_front(
            logits_output.hidden_states, accept_index, bs
        )
    return predict


def _compact_accepted_to_front(
    self, x: torch.Tensor, accept_index: torch.Tensor, bs: int
) -> torch.Tensor:
    """
    将 x (形状 [bs * num_draft_tokens, ...]) 中 accept_index 指向的 token
    收集到每个请求的前 accept_index.shape[1] 个位置。
    """
    ai_size = bs * accept_index.shape[1]
    x_flat = x.view(ai_size, -1)
    # accept_index 展平后作为索引，只取有效元素
    idx = accept_index.view(-1)
    out = x_flat[idx]
    return out.view(bs, accept_index.shape[1], *x.shape[1:])


def move_accepted_tokens_to_target_kvcache(
    self,
    batch: ScheduleBatch,
    accept_index: torch.Tensor,
    num_correct_drafts: torch.Tensor,
):
    """
    将 accepted token 的 KV cache 从 draft 位置搬移到目标 cache 的连续位置。
    """
```

```
    bs = len(batch.seq_lens)
    # 修复：size 应为 accept_index 的元素数，而非 bs * num_draft_tokens
    size = bs * accept_index.shape[1] # 安全：每个请求最多 spec_steps+1 个位置
    # ... 剩余搬移逻辑 ...
```

python/sglang/srt/arg_groups/speculative_hook.py

路由决策点，修改允许 page_size==1 时 topk>1 走 spec v2。

```
def _handle_eagle_family(server_args: "ServerArgs") -> None:
    # ... 其他逻辑 ...
    spec_v1_reason = None
    if (
        server_args.speculative_eagle_topk is not None
        and server_args.speculative_eagle_topk > 1
        and server_args.page_size > 1 # 新增：仅当 page_size>1 时 fallback
        and not server_args.disable_overlap_schedule
    ):
        # Spec v2 tree drafting 支持 page_size==1 时 topk>1.
        # page_size>1 的 draft KV 分配 (partial-page duplication) 尚未移植到 v2,
        # 因此仅在此场景 fallback 到 v1.
        server_args.disable_overlap_schedule = True
        spec_v1_reason = "spec v2 topk > 1 currently requires page_size == 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"

    if server_args.disable_overlap_schedule:
        logger.warning(
            "Spec v1 is used for eagle/eagle3/standalone speculative decoding because %s.",
            spec_v1_reason or "overlap schedule is disabled",
        )
    else:
        logger.warning(
            "Spec v2 is enabled by default for eagle/eagle3/standalone speculative decoding."
        )
```

评论区精华

Review (gemini-code-assist) 指出两个要点：

- 优化机会：move_accepted_tokens_to_target_kvcache 中临时张量 tgt_cache_loc 和 accepted_out_cache_loc 可按 ai_size=bs*(spec_steps+1) 分配而非 size，减少显存占用。
- 潜在正确性问题：当请求 return_routed_experts=True 或 return_indexer_topk=True 时，forward_batch_output.routed_experts_output 和 indexer_topk_output 未随 predicted token 一起 compact，后续下游可能出现数据错位。

- VRAM 分配优化：临时张量 size 可缩小 (performance): 未在本次 PR 中采纳，待后续优化。
- routed_experts_output 和 indexer_topk_output 未被 compact (correctness): 未在本次 PR 中修复，需后续 PR 跟进。

风险与影响

- 风险：
  1. routed_experts_output / indexer_topk_output 未 compact（已由 review 指出）：若用户显式请求这些输出且启用 topk>1，返回的数据将被污染，目前代码未处理。
  2. page_size>1 时仍强制 v1：部分用户可能因默认 page_size>1 而无法受益于此 PR，需手动调整 page_size=1。
  3. compaction 覆盖不全：hidden_states 在 logits_output 中 compact 了，但其他可能存在的 per-node 张量（如 mamba states 等）尚未确认。
  4. 测试局限：新增测试只覆盖 EAGLE3 topk=16，未覆盖 EAGLE/Llama-2 topk=8 等模型。 - 影响：对用户：在 page_size=1 时可获得 spec v2 (overlap) 的 topk>1 能力，提升解码吞吐；之前因 bug 可能偶发的 OOB 崩溃或错误推理自此修复。对系统：修复了核心 speculative 路径的边界错误，降低稳定性风险。对团队：新增的 compaction 机制为后续支持 page_size>1 的树形 draft KV 分配奠定了架构基础。 - 风险标记：routed_experts_output 未 compact，page_size>1 用户仍需手动调整，测试覆盖局限（仅 EAGLE3 topk16）

关联脉络

- PR #26981 Revert "Support spec v2 tree drafting (eagle topk>1) with page_size==1": 该 revert 正是回退了本 PR 的改动，因为发现默认行为被破坏（可能与 page_size>1 默认值冲突）。与本 PR 直接相关。
- PR #26424 [Perf][Spec Decoding] Skip cat/topk/sort/gather in draft_forward for topk=1: 同为 spec v2 性能优化，本 PR 扩展 topk>1 支持后需确保快速路径不会误用。
- PR #25940 [SPEC] feat: add adaptive speculative decoding metrics: 新增的指标可能与 topk>1 场景相关，需要验证兼容性。