PR #27491 完整报告

sgl-project/sglang

Fix SWA pool resolution for EAGLE draft workers

合并时间：2026-06-09 02:00

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27491

执行摘要

- 一句话：修复 EAGLE draft worker 的 SWA KV 池解析错误
- 推荐动作：该 PR 值得精读，特别是涉及 TRTLLM、SWA 和 speculative decoding 交互的开发者。设计上清晰区分了 EAGLE 和 FROZEN_KV_MTP 的 SWA 池使用策略。后续若新增 speculative 算法，需注意此处逻辑。

功能与动机

PR 描述指出该提交与 #25103 类似，但被 #26966 错误回退。Issue #25103 修复了 trtllm mha + swa + spec accept length drop，而 #26966 在修复 Gemma 4 MTP 时回退了该修复，导致 EAGLE draft workers 再次使用错误的 SWA 映射。本 PR 重新修复该问题，并确保 EAGLE draft workers 不会错误应用 target allocator 的 SWA 映射（除非是 FROZEN_KV_MTP 路径）。

实现拆解

1. 核心逻辑在 trtllm_mha_backend.py 的 _resolve_swa_kv_pool 方法中重构：优先返回 model_runner.token_to_kv_pool 若它是 SWAKVPool；接着若 is_draft_worker 为 True 且 spec_algorithm 不是 is_frozen_kv_mtp()，则直接返回 None（避免使用分配器的 SWA 映射）；否则回退到 token_to_kv_pool_allocator.get_kvcache() 并检查是否为 SWAKVPool。
2. 在 __init__ 中移除关于 allocator 的注释，改为描述 SWA 池的用途。
3. 新增 test/registered/unit/spec/test_resolve_swa_kv_pool.py 文件，包含 6 个单元测试，覆盖所有分支（活跃池为 SWA、非 SWA 回退、allocator 非 SWA 返回 None、draft worker 非 frozen_kv 返回 None、draft worker frozen_kv_mtp 返回 allocator SWA、非 draft worker 忽略 spec_algorithm）。
4. 修改 dense_attention.py 中的 MockModelRunner，增加 is_draft_worker、spec_algorithm 属性，并让 token_to_kv_pool_allocator 提供 get_kvcache 方法，以支持在注意力框架测试中使用新的 resolve 方法。

关键文件：

- python/sglang/srt/layers/attention/trtllm_mha_backend.py（模块 注意力后端；类别 source；类型 core-logic；符号 _resolve_swa_kv_pool, _maybe_translate_swa, _alloc_swa_page_table, _copy_swa_page_table）：核心修改文件，重构了

_resolve_swa_kv_pool 方法，是修复的关键。

- test/registered/unit/spec/test_resolve_swa_kv_pool.py（模块 测试；类别 test；类型 test-coverage；符号 _mock_runner, TestResolveSwaKvPool, test_active_pool_is_swa_returns_it, test_non_swa_active_pool_falls_through_to_allocator）：新增的单元测试文件，覆盖所有 6 个分支，确保修复可防回归。
- python/sglang/test/kits/attention_unittest/attention_methods/dense_attention.py（模块 测试工具；类别 test；类型 test-coverage；符号 MockModelRunner.init）：修改 MockModelRunner 以支持新属性，使注意力测试能正确实例化，属于测试基础设施配合。

关键符号：TRTLLMHAAttnBackend._resolve_swa_kv_pool, TRTLLMHAAttnBackend._maybe_translate_swa, MockModelRunner.init

关键源码片段

test/registered/unit/spec/test_resolve_swa_kv_pool.py

新增的单元测试文件，覆盖所有 6 个分支，确保修复可防回归。

```
# test/registered/unit/spec/test_resolve_swa_kv_pool.py
def _mock_runner(
    *,
    active_pool=None,
    is_draft_worker=False,
    spec_algorithm=SpeculativeAlgorithm.NONE,
    allocator_kvcache=None,
):
    runner = MagicMock()
    runner.token_to_kv_pool = active_pool
    runner.is_draft_worker = is_draft_worker
    runner.spec_algorithm = spec_algorithm
    runner.token_to_kv_pool_allocator.get_kvcache.return_value = allocator_kvcache
    return runner

class TestResolveSwaKvPool(CustomTestCase):
    def test_active_pool_is_swa_returns_it(self):
        swa = MagicMock(spec=SWAKVPool)
        runner = _mock_runner(active_pool=swa)
        self.assertIs(_resolve(runner), swa)

    def test_non_swa_active_pool_falls_through_to_allocator(self):
        swa = MagicMock(spec=SWAKVPool)
        runner = _mock_runner(active_pool=MagicMock(), allocator_kvcache=swa)
        self.assertIs(_resolve(runner), swa)

    def test_draft_worker_non_frozen_kv_returns_none(self):
        runner = _mock_runner(
            active_pool=MagicMock(),
            is_draft_worker=True,
            spec_algorithm=SpeculativeAlgorithm.EAGLE,
```

```
        allocator_kvcache=MagicMock(spec=SWAKVPool),
    )
    self.assertIsNone(_resolve(runner))
```

评论区精华

- ispobock要求添加单元测试避免回归，作者随后新增了 test_resolve_swa_kv_pool.py。
- gemini-code-assist[bot]建议使用 getattr 安全获取 token_to_kv_pool 以兼容 mock/stub，但作者未采纳，直接使用属性访问，因为所有生产路径和测试 mock 均已保证该属性存在。
- 单元测试覆盖 (testing): 作者新增了 test_resolve_swa_kv_pool.py 覆盖所有分支。
- 防御性编程与 getattr 使用 (style): 作者未采纳，直接使用属性访问，因为所有路径均保证属性存在。

风险与影响

- 风险：核心风险在于修改了 trtllm_mha_backend.py 中的 SWA 池解析逻辑，影响所有使用 TRTLLM MHA 后端 + SWA + speculative decoding 的模型（如 Gemma 4、Blackwell 平台）。若逻辑存在遗漏分支，可能导致 CUDA 非法内存访问或静默错误。新增的单元测试覆盖了 6 个关键路径，但可能缺少一些边缘情况（如 allocator.get_kvcache 返回非 SWA 但为 None 以外的类型）。另外，dense_attention.py 的修改影响注意力测试框架，但改动较小。
- 影响：用户 / 系统：修复了 EAGLE draft worker 中使用 TRTLLM MHA 和 SWA 时的崩溃和错误输出，特别是 Gemma 4 MTP 场景。性能：无负面影响，逻辑分支增加但开销可忽略。团队：需要审阅并确保 CI 通过，特别是 Blackwell 相关测试。
- 风险标记：核心注意力后端变更，依赖 spec_algorithm 分支，回归风险（已由测试缓解）

关联脉络

- PR #25103 [TRTLLM/SWA/Spec] fix trtllm mha + swa + spec accept length drop: 本 PR 是 #25103 类似修复，被 #26966 错误回退。
- PR #26966 [Spec] Fix Gemma 4 MTP with trtllm_mha crash issue: 该 PR 回退了 #25103 的修复，导致本 PR 需要重新修复。