PR #24318 完整报告

sgl-project/sglang

Fix swa chunk req deferred

合并时间：2026-05-04 14:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24318

执行摘要

- 一句话：修复 SWA chunked request 在 KV 缓存回缩时的 stash 误触发
- 推荐动作：值得精读。该 PR 展示了如何通过一个简单的门控标志位解决状态损坏问题，同时配套了高质量的回归测试（包括辅助函数和清晰的场景划分）。建议关注 _chunked_req_scheduled_last_iter 的设置时机和后续维护。

功能与动机

Issue #24252 报告在配置 --swa-full-tokens-ratio 0.1 且 KV 缓存回缩时，服务器在 extend 路径中因空 micro-batch 崩溃，root cause 是 stash_chunked_request 在 chunked_req 未实际调度时错误修改了 prefix_indices。本 PR 从 #23882 挑选补丁精准修复该问题。

实现拆解

1. 在 Scheduler.init_chunked_prefill 中新增布尔标志位 _chunked_req_scheduled_last_iter，初始化为 False，并添加详细注释说明用途。
2. 在 _get_new_batch_prefill_raw 中，当调用 add_chunked_req 后，检查 self.chunked_req 是否在 adder.can_run_list 中，从而设置标志位；对于 new_chunked_req 直接设为 True（必然被调度）。
3. 在 get_next_batch_to_run 中，将原来的无条件 stash_chunked_request 改为仅在标志位为 True 时执行，避免对 deferred chunked_req 误操作。
4. 新增测试文件 test_scheduler_chunked_req_gate.py，通过构造模拟的 Scheduler、ChunkCache 和 Req，验证 deferred 场景下 prefix_indices 不变，scheduled 场景下正常推进。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 init_chunked_prefill, get_next_batch_to_run, _get_new_batch_prefill_raw）：核心调度器，新增门控标志位避免 stash 误触发
- test/registered/unit/managers/test_scheduler_chunked_req_gate.py（模块 测试；类别 test；类型 test-coverage；符号 _make_req, _make_req_to_token_pool, _make_chunk_cache, _scheduler_for_get_next_batch）：新增回归测试，验证 stash gate 在 deferred 和 scheduled 场景下的行为

关键符号：init_chunked_prefill, get_next_batch_to_run, _get_new_batch_prefill_raw, stash_chunked_request, test_deferred_chunked_req_keeps_real_prefix_indices, test_scheduled_chunked_req_advances_prefix_indices_via_real_stash

关键源码片段

python/sglang/srt/managers/scheduler.py

核心调度器，新增门控标志位避免 stash 误触发

```
# --- init_chunked_prefill 中新增标志位 ---
def init_chunked_prefill(self):
    # ... 前面代码 ...
    self.chunked_req = None
    # 记录当前 chunked_req 是否在上一次迭代中被实际调度到 can_run_list
    # 用于在 get_next_batch_to_run 中判断是否需要 stash_chunked_request
    # 如果 add_chunked_req 因 hybrid SWA 压力 early return，则 req_pool_idx
    # 已被释放且 fill_ids 被 init_next_round_input 重置，此时 stash 会导致
    # double-free 并损坏 prefix_indices
    self._chunked_req_scheduled_last_iter = False
    # ...

# --- get_next_batch_to_run 中条件 stash ---
def get_next_batch_to_run(self) -> Optional[ScheduleBatch]:
    # ...
    if self.chunked_req is not None:
        chunked_req_to_exclude.add(self.chunked_req)
        # 只有上次实际调度过的 chunked_req 才需要 stash
        if self._chunked_req_scheduled_last_iter:
            self.stash_chunked_request(self.chunked_req)
    # ...

# --- _get_new_batch_prefill_raw 中设置标志位 ---
def _get_new_batch_prefill_raw(self, ...):
    # ...
    if self.chunked_req is not None:
        self.chunked_req.init_next_round_input()
        self.chunked_req = adder.add_chunked_req(self.chunked_req)
        self._chunked_req_scheduled_last_iter = (
            self.chunked_req in adder.can_run_list
        )
    else:
        self._chunked_req_scheduled_last_iter = False
    # ...
    # 对于 new_chunked_req（新创建的分块请求），它一定会被加入 can_run_list
    if ...:
        self.chunked_req = adder.new_chunked_req
        self._chunked_req_scheduled_last_iter = True
    # ...
```

test/registered/unit/managers/test_scheduler_chunked_req_gate.py

新增回归测试，验证 stash gate 在 deferred 和 scheduled 场景下的行为

```
"""Regression tests for the SWA chunked-req stash gate (#24252)."""
import torch
from sglang.srt.managers.schedule_batch import Req
from sglang.srt.managers.scheduler import Scheduler
from sglang.srt.mem_cache.chunk_cache import ChunkCache

def _make_req(req_pool_idx, fill_ids, prefix_indices, extend_input_len) -> Req:
    """创建一个最小 Req 对象，仅填充测试需要的字段。"""
    req = Req.__new__(Req)
    req.rid = "test-req"
    req.origin_input_ids = list(fill_ids)
    req.output_ids = []
    req.fill_ids = list(fill_ids)
    req.prefix_indices = prefix_indices
    req.req_pool_idx = req_pool_idx
    req.extend_input_len = extend_input_len
    req.is_chunked = 0
    req.host_hit_length = 0
    req.cache_protected_len = 0
    req.skip_radix_cache_insert = False
    req.last_node = None
    req.swa_uuid_for_lock = None
    req.session = None
    req.return_logprob = False
    req.logprob_start_len = -1
    req.positional_embed_overrides = None
    req.extra_key = None
    req.mamba_pool_idx = None
    req.sampling_params = SimpleNamespace(max_new_tokens=128, ignore_eos=False)
    return req

def _make_req_to_token_pool(num_slots, max_context):
    # 每个槽位设置可识别的指纹，用于检测 prefix_indices 是否被篡改
    pool = SimpleNamespace()
    pool.req_to_token = (
        torch.arange(max_context, dtype=torch.int32).unsqueeze(0).repeat(num_slots, 1)
        + torch.arange(num_slots, dtype=torch.int32).unsqueeze(1) * 1000
    )
    return pool

class TestStashGatePreservesPrefixIndices(CustomTestCase):
    POOL_IDX = 4
    INITIAL_PREFIX_LEN = 8
    POST_RESET_FILL_LEN = 32
    NUM_SLOTS = 8
    MAX_CONTEXT = 64
```

```
    def _build(self, flag: bool):
        pool = _make_req_to_token_pool(self.NUM_SLOTS, self.MAX_CONTEXT)
        cache = _make_chunk_cache(pool)
        initial_prefix = pool.req_to_token[self.POOL_IDX, :self.INITIAL_PREFIX_LEN].to(
            dtype=torch.int64, copy=True
        )
        req = _make_req(
            req_pool_idx=self.POOL_IDX,
            fill_ids=list(range(self.POST_RESET_FILL_LEN)),
            prefix_indices=initial_prefix,
            extend_input_len=0,
        )
        s = _scheduler_for_get_next_batch(tree_cache=cache, chunked_req=req)
        s._chunked_req_scheduled_last_iter = flag
        return s, req, initial_prefix, pool

    def test_deferred_chunked_req_keeps_real_prefix_indices(self):
        # bug 场景：deferred chunked_req 在 stash 时不应修改 prefix_indices
        s, req, initial_prefix, _ = self._build(flag=False)
        Scheduler.get_next_batch_to_run(s)
        self.assertEqual(req.prefix_indices.shape[0], self.INITIAL_PREFIX_LEN)
        self.assertTrue(torch.equal(req.prefix_indices, initial_prefix))

    def test_scheduled_chunked_req_advances_prefix_indices_via_real_stash(self):
        # 对称验证：scheduled 场景下 stash 正常推进 prefix_indices
        s, req, initial_prefix, _ = self._build(flag=True)
        Scheduler.get_next_batch_to_run(s)
        # 期待 prefix_indices 被扩展到 POST_RESET_FILL_LEN
        self.assertEqual(req.prefix_indices.shape[0], self.POST_RESET_FILL_LEN)
```

评论区精华

Review 中没有实质技术争论。Gemini Code Assist 自动审查后表示无反馈；Ratish1 直接批准。因此无核心争议点。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。主要风险是标志位维护可能遗漏某些调用路径（如混合 chunk 场景），但测试覆盖了两种关键场景。后续若有代码修改需同步更新标志位设置。
- 影响：影响使用 hybrid SWA 和 chunked prefill 的用户，尤其是设置了 --swa-full-tokens-ratio 非零值的场景。修复后服务器在 KV 缓存回缩时不再因 stash 误触发而崩溃，稳定性提升。对未触发回缩或未使用 SWA 的场景无影响。
- 风险标记：核心路径变更，新增状态标志，测试覆盖新增

关联脉络

- PR #23882 Fix swa chunk req deferred: 本 PR 从该 PR 挑选（cherry-pick）补丁，修复相同的 root cause。
- PR #24252 [Bug] Empty micro-batch produced after KV-pool retraction crashes ...: 关联 Issue，本 PR 修复了该 Issue 报告的崩溃问题。