PR #23174 完整报告

sgl-project/sglang

Fix hybrid swa chunked prefill oom

合并时间：2026-04-21 10:46

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23174

执行摘要

- 一句话：修复混合 SWA 分块预填充的内存溢出问题，确保预留页面空间并正确处理资源不足时的延迟处理。
- 推荐动作：该 PR 值得精读，特别是 add_chunked_req 方法中的设计决策：如何平衡资源分配与 OOM 风险，以及延迟处理与回退策略的选择。关注预留页面空间的实现细节和测试用例的设计，这些展示了在资源受限场景下的稳健性处理模式。

功能与动机

根据 PR body 描述，动机是解决混合 SWA 路径中分块预填充的内存溢出问题。具体来说，PrefillAdder.add_chunked_req 在混合 SWA 路径中计算剩余令牌时，未考虑 alloc_extend 需要为每个请求预留 extend_num_tokens + page_size 的空间，导致当 rem_swa_tokens == extend_num_tokens 时，调度器分配全部 SWA 预算，但 alloc_extend 无法获取额外页面而 OOM。此外，当 _rem_tokens <= 0 时，原有逻辑无条件回退到 rem_chunk_tokens，绕过 SWA 预算，在持续压力下同样可能触发 OOM。修复后，请求将在资源不足时延迟到下一轮处理。

实现拆解

1. 核心逻辑修复：修改 python/sglang/srt/managers/schedule_policy.py 中的 add_chunked_req 方法。在混合 SWA 模式下，计算 _rem_tokens 时从 int(self.rem_swa_tokens) 中减去 self.page_size，为 alloc_extend 预留页面空间。当 _rem_tokens <= 0 时，如果是混合 SWA 模式，则直接返回请求对象以延迟处理，而不是回退到 rem_chunk_tokens。
2. 测试配套：在 test/registered/unit/managers/test_prefill_adder.py 中新增两个测试函数。test_add_chunked_req_hybrid_swa_reserves_page_for_alloc_extend 验证预留页面逻辑，确保 extend_input_len 不超过 rem_swa_tokens - page_size。test_add_chunked_req_hybrid_swa_defers_when_swa_below_page 验证当 SWA 预算低于页面大小时，请求被延迟而非回退。同时添加辅助函数 _build_hybrid_swa_chunked_req 以简化测试构建。

关键文件：

- python/sglang/srt/managers/schedule_policy.py（模块 调度策略；类别 source；类型 core-logic；符号 add_chunked_req）：修复了混合 SWA 分块预填充的内存溢出问题，是

核心逻辑变更所在。

- test/registered/unit/managers/test_prefill_adder.py（模块 预填充测试；类别 test；类型 test-coverage；符号 _build_hybrid_swa_chunked_req, test_add_chunked_req_hybrid_swa_reserves_page_for_alloc_extend, test_add_chunked_req_hybrid_swa_defers_when_swa_below_page）：新增测试用例验证修复逻辑，确保预留页面和延迟处理行为正确。

关键符号：add_chunked_req

关键源码片段

python/sglang/srt/managers/schedule_policy.py

修复了混合 SWA 分块预填充的内存溢出问题，是核心逻辑变更所在。

```
def add_chunked_req(self, req: Req):
    if self.dllm_config is not None:
        _rem_tokens = self._get_dllm_remain_tokens()
    else:
        _rem_tokens = min(self.rem_chunk_tokens, int(self.rem_total_tokens))
        if self.is_hybrid_swa:
            # alloc_extend needs extend_num_tokens + page_size per request,
            # so reserve one page here to avoid OOM
            _rem_tokens = min(
                _rem_tokens, int(self.rem_swa_tokens) - self.page_size
            )
        # The chunked_req must be added to the list; otherwise, it will cause a memory leak.
        # Therefore, in certain cases where _rem_tokens <= 0, it should be replaced with rem_
        chunk_tokens.
        if _rem_tokens <= 0:
            if self.is_hybrid_swa:
                return req # 混合 SWA 模式下资源不足，延迟请求而非回退
            _rem_tokens = self.rem_chunk_tokens # 非混合 SWA 模式回退到分块令牌预算

    truncated = req.extend_input_len > _rem_tokens
    req.set_extend_input_len(min(req.extend_input_len, _rem_tokens))
    req.fill_ids = req.fill_ids[: len(req.prefix_indices) + req.extend_input_len]
    self.can_run_list.append(req)
    self._update_prefill_budget(
        0,
        req.extend_input_len,
        (
            min(req.sampling_params.max_new_tokens, CLIP_MAX_NEW_TOKENS)
            if not truncated
            else 0
        ),
    )

    # Return if chunked prefill not finished
```

```
        return req if truncated else None
```

test/registered/unit/managers/test_prefill_adder.py

新增测试用例验证修复逻辑，确保预留页面和延迟处理行为正确。

```
    def test_add_chunked_req_hybrid_swa_reserves_page_for_alloc_extend(self):
        # alloc_extend needs extend_num_tokens + page_size per request. If the
        # scheduler hands out all of rem_swa_tokens, alloc_extend cannot get its
        # extra page and OOMs. With the fix, extend_input_len must cap at
        # rem_swa_tokens - page_size so the page is reserved.
        PAGE_SIZE = 64
        REM_SWA = 100
        adder, req = self._build_hybrid_swa_chunked_req(
            page_size=PAGE_SIZE, rem_swa=REM_SWA
        )

        result = adder.add_chunked_req(req)

        self.assertIs(result, req) # truncated → chunked prefill continues
        req.set_extend_input_len.assert_called_once()
        new_len = req.set_extend_input_len.call_args.args[0]
        self.assertLessEqual(new_len + PAGE_SIZE, REM_SWA) # 验证预留页面空间
        self.assertEqual(new_len, REM_SWA - PAGE_SIZE) # 验证计算正确

    def test_add_chunked_req_hybrid_swa_defers_when_swa_below_page(self):
        # When rem_swa_tokens <= page_size there is no room to serve even the
        # reservation, so the chunked req must be deferred (returned unchanged)
        # instead of falling back to rem_chunk_tokens and bypassing SWA budget.
        PAGE_SIZE = 64
        adder, req = self._build_hybrid_swa_chunked_req(
            page_size=PAGE_SIZE, rem_swa=PAGE_SIZE
        )
        original_len = req.extend_input_len

        result = adder.add_chunked_req(req)

        self.assertIs(result, req) # 请求被延迟，未修改
        req.set_extend_input_len.assert_not_called() # 验证未调用设置长度
        self.assertEqual(req.extend_input_len, original_len) # 长度保持不变
        self.assertEqual(len(adder.can_run_list), 0) # 未加入运行列表
```

评论区精华

review 中仅有 gemini-code-assist[bot] 的评论，总结了 PR 的修改内容和目的，指出修改了 add_chunked_req 方法以防止 OOM，并添加了单元测试验证预留逻辑和延迟行为。评论者表示没有反馈，说明变更已被接受。

- 修复混合 SWA 分块预填充 OOM 的代码审查 (correctness): 变更被接受，没有进一步反馈。

风险与影响

- 风险：技术风险较低，但需注意：
  - 回归风险：修改了 add_chunked_req 的核心逻辑，可能影响其他非混合 SWA 路径或 DLLM 配置下的行为。但变更范围小，且新增测试覆盖了混合 SWA 场景，降低了风险。
  - 性能影响：预留页面空间可能略微减少单次可处理的令牌数，但避免了 OOM 导致的崩溃，整体更稳定。
  - 兼容性：仅影响混合 SWA 模式，对其他模式无影响。
  - 影响：对系统的影响是修复了内存溢出问题，提升了混合 SWA 模式下分块预填充的稳定性。用户将不再遇到因 OOM 导致的服务器崩溃，特别是在高负载或资源紧张时。对团队而言，新增的测试用例增强了相关逻辑的验证，有助于防止未来回归。影响范围限于调度器的预填充处理模块，不涉及其他子系统。
  - 风险标记：核心路径变更，资源分配逻辑调整

关联脉络

- PR #23252 [Fix] Solve the error lead by _commit_transfer_to_req() when using IntraNode NVLink in PD disaggregation: 同样涉及调度和内存管理相关的 bugfix，关注资源分配和错误处理。
- PR #23138 fix: reset empty prefill batch fullness: 涉及调度器中的预填充处理，修复了批次满载标志重置问题，与本 PR 同属调度模块的 bugfix。
- PR #22894 fix(hicache): emit KV events for L2 host cache insertions: 涉及缓存和内存管理，与本 PR 的 KV 缓存和资源分配主题相关。