PR #27923 完整报告

sgl-project/sglang

Fix MambaPool.clear_slots OOM by replacing expand-based tensor allocation with scalar zeroing

合并时间：2026-07-02 09:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27923

执行摘要

- 一句话：修复 MambaPool.clear_slots NPU OOM
- 推荐动作：值得精读，展示了一个典型的针对硬件特性的内存优化案例。尤其适合关注跨平台 GPU 编程或显存敏感型推理服务的开发者学习。

功能与动机

NPU 上部署 Qwen3.5-35B-A3B 服务时出现 OOM，根因是 MambaPool.clear_slots 中使用 torch.zeros(1).expand(...) 创建了巨大的临时张量，在 NPU 上额外分配大量显存。

实现拆解

1. 修改核心函数：在 python/sglang/srt/mem_cache/memory_pool.py 的 clear_slots 方法中，根据 _is_npu 标志进行条件分支。
2. 保留原 CUDA 路径：if not _is_npu 分支沿用原来的 torch.zeros(1).expand(...) + 高级索引赋值逻辑，确保 CUDA 行为不变。
3. 新增 NPU 优化路径：else 分支直接使用 t[:, indices] = 0 逐标量清零，避免创建临时扩展张量，显著降低内存占用。

关键文件：

- python/sglang/srt/mem_cache/memory_pool.py（模块 内存池；类别 source；类型 core-logic；符号 clear_slots）：核心变更文件，修复 MambaPool.clear_slots 的 OOM 问题

关键符号：clear_slots

关键源码片段

python/sglang/srt/mem_cache/memory_pool.py

核心变更文件，修复 MambaPool.clear_slots 的 OOM 问题

```
def clear_slots(self, indices: torch.Tensor):
    """Zero out mamba state at the given pool indices. Must run on forward stream."""
    # NPU 上 torch.zeros(1).expand(...) 会分配大块显存导致 OOM
    # 因此对 NPU 使用标量赋值，避免分配扩展张量
    if not _is_npu:
```

```
            need_size = len(indices)
            for i in range(len(self.mamba_cache.conv)):
                t = self.mamba_cache.conv[i]
                # 创建 (1, need_size, ...) 的零张量并扩展
                z = torch.zeros(1, dtype=t.dtype, device=t.device).expand(
                    t.shape[0], need_size, *t.shape[2:]
                )
                t[:, indices] = z
            t = self.mamba_cache.temporal
            z = torch.zeros(1, dtype=t.dtype, device=t.device).expand(
                t.shape[0], need_size, *t.shape[2:]
            )
            t[:, indices] = z
        else:
            # NPU 路径：直接标量清零，无需分配临时张量
            for i in range(len(self.mamba_cache.conv)):
                t = self.mamba_cache.conv[i]
                t[:, indices] = 0
            t = self.mamba_cache.temporal
            t[:, indices] = 0
```

评论区精华

gemini-code-assist[bot] 最初建议先跳过 _execute_deferred_mamba_cow_and_clear 整体，但随后指出这种做法会破坏两个方面：

- 槽位复用时的脏状态：不清除会导致新请求读到旧状态，产生错误输出。
- Copy-on-Write 操作：跳过会导致共享前缀缓存被直接覆盖，破坏正确性。 最终提交采取了更精确的优化方案：仅替换 clear_slots 内部的张量分配方式，不改变控制流。
- Skipping entire clear/cow logic vs. optimizing clear_slots (correctness): 采纳了更精确的优化：仅修改 clear_slots 内部实现，保留原控制流。

风险与影响

- 风险：低风险。变更仅在 clear_slots 内部，且语义等价（标量清零 vs 扩展张量赋值）。NPU 路径已单独分支，不会影响 CUDA 行为。但缺少针对 t[:, indices] = 0 的逐元素性能基准测试，可能在某些极端形状下触发不同 kernel 路径。
- 影响：用户影响：修复 NPU 上 Mamba 模型（如 Qwen3.5-35B-A3B）的 OOM 问题，使其能够正常部署。 系统影响：改变仅在 NPU 路径生效，CUDA 侧无变化。 团队影响：减少了 NPU 与 CUDA 之间的行为差异，提高可维护性。
- 风险标记：NPU 专用路径 ，缺少测试覆盖

关联脉络

- PR #29678 feat(mem_cache): unified memory pool for hybrid Mamba / SWA models: 同一内存池模块的近期重构，引入 Mamba 缓存相关逻辑