PR #35727 完整报告

sgl-project/sglang

[NPU][Bugfix] Fix OOB gather in decode KV allocation when free pool is tight

合并时间：2026-08-27 19:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35727

执行摘要

- 一句话：修复 NPU decode 分配器在空闲页池紧张时的越界 gather
- 推荐动作：值得快速阅读，理解 NPU 分配器的边界条件处理。建议后续补充针对该边界情况的单元测试，以巩固修复效果。

功能与动机

PR 正文明确指出，当空闲页池恰好紧密时（len(self.free_pages) == num_new_pages），start_new_pages 的最后一个元素会等于 num_new_pages，导致 self.free_pages[start_new_pages] 索引越界。该问题仅在池恰好用尽时触发，并在 CI 的 test_npu_schedule_conservativeness.py 中导致 AI Core 错误和服务器崩溃。

实现拆解

1. 在 python/sglang/srt/hardware_backend/npu/allocator_npu.py 的 alloc_decode 方法中，在计算 out_indices 之前，对 start_new_pages 应用 clamp，限制其最大值不超过 num_new_pages - 1。
2. 由于 num_new_pages == 0 的分支提前返回，clamp 操作不会出现上限为 -1 的情况。
3. 该 clamp 不影响实际结果，因为被 clamp 的行 need_new_pages == 0，在后续计算中会被乘以 0 而屏蔽。
4. 对每个 decode batch 仅增加一次元素级 clamp 操作，性能开销可忽略。

关键文件：

- python/sglang/srt/hardware_backend/npu/allocator_npu.py（模块 内存分配；类别 source；类型 core-logic；符号 alloc_decode）：这是唯一的变更文件，修复了 NPU decode KV 分配器中的越界索引问题。

关键符号：alloc_decode

关键源码片段

python/sglang/srt/hardware_backend/npu/allocator_npu.py

这是唯一的变更文件，修复了 NPU decode KV 分配器中的越界索引问题。

```
# python/sglang/srt/hardware_backend/npu/allocator_npu.py
# 修复空闲页池恰好用尽时的越界 gather
```

```
def alloc_decode(
    self,
    seq_lens: torch.Tensor,
    seq_lens_cpu: torch.Tensor,
    last_loc: torch.Tensor,
):
    # ...（前置检查和计算 num_new_pages 等）

    need_new_pages = (seq_lens % self.page_size == 1).int()
    end_new_pages = torch.cumsum(need_new_pages, 0)
    # start_new_pages 是 free_pages 的前缀索引，范围应为 [0, num_new_pages-1]
    start_new_pages = end_new_pages - need_new_pages
    if num_new_pages == 0:
        out_indices = last_loc + 1
    else:
        # 关键修复：clamp 防止 start_new_pages 等于 num_new_pages 时越界
        # 被 clamp 的行 need_new_pages == 0，后续会被 * 0 屏蔽，不影响结果
        start_new_pages = start_new_pages.clamp(max=num_new_pages - 1)
        out_indices = (last_loc + 1) * (1 - need_new_pages) + self.free_pages[
            start_new_pages
        ] * self.page_size * need_new_pages

    # ...（debug 断言和更新 free_pages）
    return out_indices.int()
```

评论区精华

该 PR 仅由机器人审查并批准，无人工评论。PR 正文中详细解释了 clamp 的安全性和正确性依据。

- 暂无高价值评论线程

风险与影响

- 风险：该变更仅增加了一行 clamp 操作，逻辑简单，风险较低。但未添加针对该边界情况的单元测试，可能在未来重构中再次引入回归。此外，该修复仅针对 NPU 后端，其他后端的类似逻辑是否存在相同问题值得检查。
- 影响：影响范围集中在 NPU 后端的 KV 缓存分配路径，解决了在特定内存紧张场景下的崩溃问题。对正常路径无性能影响，提高了 NPU 后端的稳定性。
- 风险标记：缺少测试覆盖

关联脉络

- PR #36739 [misc] Fold the allocator free-group flag into free_group: 同样涉及 NPU 分配器（allocator_npu.py），可能影响 free_pages 管理逻辑。
- PR #36640 [NPU] [bugfix] Fix import of ggml_moe_a8_vec and Fix NPU MLA HiCache backup accessing missing data_ptrs: 同为 NPU 相关 bugfix，关注 NPU 内存与缓存路

径的稳定性。