PR #31747 完整报告

sgl-project/sglang

[AMD] DSv4: bring HIP compress-state pool into the memory_saver KV_CACHE region

合并时间：2026-07-29 17:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31747

执行摘要

- 一句话：修复 AMD DSv4 压缩状态池内存回收问题
- 推荐动作：建议精读。此 PR 展示了如何通过消除平台特定分支、复用统一内存管理基础设施来修复可回收内存泄漏问题，设计干净、验证充分。值得关注的是 TorchMemorySaverAdapter 和 custom_mem_pool 的集成模式，该模式可推广到其他需要被 pause()/resume() 管理的分配场景。

功能与动机

在 gfx950 上，DeepSeek-V4 的压缩状态池 kv_score_buffer 分配在 torch_memory_saver 区域之外，导致 pause() 无法回收此内存，联合 RL 训练期间每张 GPU 约 52 GiB 无法释放。PR body 明确描述："the compress-state pool allocated kv_score_buffer outside the torch_memory_saver region. As a result, pause() could not reclaim this memory during colocated RL training."

实现拆解

1. 统一分配路径：在 python/sglang/srt/mem_cache/deepseek_v4_compress_state.py 的 _alloc_kv_score_buffer 方法中，移除 _is_hip 条件分支。之前 HIP 路径直接使用 torch.empty() 分配，不经过 torch_memory_saver 区域；CUDA 路径则通过 TorchMemorySaverAdapter.region(GPU_MEMORY_TYPE_KV_CACHE) 和自定义内存池分配。现在 HIP 和 CUDA 使用相同的分配逻辑，即都进入 region 上下文，确保分配被 torch_memory_saver 跟踪。
2. 启用内存保存器：在 python/sglang/srt/mem_cache/deepseek_v4_memory_pool.py 的 DeepSeekV4TokenToKVPool.__init__ 中，移除 if _is_hip: self._init_paged_compress_states(False) 分支，改为对所有平台统一调用 self._init_paged_compress_states(enable_memory_saver)，从而使 HIP 平台也能利用 memory_saver 机制。
3. 计算路径无变化：PR 强调仅改变缓冲区的分配和注册方式，压缩器的计算逻辑保持不变。

关键文件：

- python/sglang/srt/mem_cache/deepseek_v4_compress_state.py（模块 压缩状态池；类别 source；类型 core-logic；符号 _alloc_kv_score_buffer）：核心修改文件：移除 _is_hip 分支，将 HIP 的 kv_score_buffer 分配纳入 torch_memory_saver 区域，是实现

内存回收的关键变化。

- python/sglang/srt/mem_cache/deepseek_v4_memory_pool.py（模块 内存池；类别 source；类型 core-logic；符号 DeepSeekV4TokenToKVPool.init）：辅助修改文件：移除 _is_hip 条件分支，统一调用 _init_paged_compress_states(enable_memory_saver)，确保 HIP 平台也启用 memory saver。

关键符号：_alloc_kv_score_buffer, DeepSeekV4TokenToKVPool.init

关键源码片段

python/sglang/srt/mem_cache/deepseek_v4_compress_state.py

核心修改文件：移除 _is_hip 分支，将 HIP 的 kv_score_buffer 分配纳入 torch_memory_saver 区域，是实现内存回收的关键变化。

```
def _alloc_kv_score_buffer(
    self, *, dtype: torch.dtype, device: str, enable_memory_saver: bool
) -> None:
    """Allocate the flat ``(self._size, self.last_dim)`` kv+score buffer
    under the memory-saver / custom-mem-pool context and wrap it in
    :class:`KVAndScore`."""
    # [AMD Fix] Removed the _is_hip branch that used bare torch.empty() outside
    # the memory_saver region. Now both HIP and CUDA use the same allocation
    # path, ensuring the buffer is tracked by torch_memory_saver so that
    # pause() can reclaim it during co-located RL training.
    self.memory_saver_adapter = TorchMemorySaverAdapter.create(
        enable=enable_memory_saver
    )
    self.enable_custom_mem_pool, self.custom_mem_pool, _ = (
        maybe_init_custom_mem_pool(device=device)
    )
    with self.memory_saver_adapter.region(GPU_MEMORY_TYPE_KV_CACHE):
        with (
            torch.cuda.use_mem_pool(self.custom_mem_pool)
            if self.custom_mem_pool
            else nullcontext()
        ):
            self.kv_score_buffer = KVAndScore(
                torch.empty(
                    (self._size, self.last_dim),
                    dtype=dtype,
                    device=device,
                )
            )
```

python/sglang/srt/mem_cache/deepseek_v4_memory_pool.py

辅助修改文件：移除 _is_hip 条件分支，统一调用 _init_paged_compress_states(enable_memory_saver)，确保 HIP 平台也启用 memory saver。

```
# In DeepSeekV4TokenToKVPool.__init__:
```

```
# ... after setting up pools ...
self._init_compressed_layer_mapping()

# [AMD Fix] Previously this was:
# if _is_hip:
# self._init_paged_compress_states(False)
# else:
# self._init_paged_compress_states(enable_memory_saver)
# Now both platforms use the same enable_memory_saver flag.
self._init_paged_compress_states(enable_memory_saver)
```

评论区精华

Review 评论中无实质性讨论。一位 reviewer 评论 "LGTM. Only effect hip code path."，另一位 reviewer 直接批准。说明改动意图明确、影响范围清晰，未引发设计争议。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低，但需关注以下几点：
  - 回归风险：CUDA 路径未改动，回归可能性低；HIP 路径从裸 torch.empty() 切换为 region+ 内存池分配，若 TorchMemorySaverAdapter 或 custom_mem_pool 在特定 HIP 配置下初始化异常，可能导致分配失败或行为异常。
  - 性能风险：新路径可能引入微小的分配开销（如 region 上下文切换），但 PR 验证显示池大小从 2.02 GiB 降至 1.87 GiB，且 resume() 正确重映射，表明性能影响可接受。
  - 兼容性风险：仅影响 AMD HIP 路径，与 CUDA 无关；对其他后端（如 NPU）无影响。
  - 影响：
  - 用户影响：使用 AMD MI355X (gfx950) 运行 DeepSeek-V4 联合 RL 训练的用户将直接受益，GPU 内存占用减少约 52 GiB，训练阶段内存占用从 143 GiB 降至 87 GiB，提升资源利用率和吞吐量。
  - 系统影响：改动涉及压缩状态池的分配路径，属于内存管理核心逻辑，但影响范围限定在 AMD HIP 平台。
  - 团队影响：为 AMD 平台与 CUDA 平台对齐了内存管理行为，降低了后续维护复杂度。
  - 风险标记：AMD 平台特定变更，核心内存分配路径变更

关联脉络

- PR #30256 Add Mooncake tenant id support: 同样涉及 mem_cache 模块的存储后端变更，且与 Mooncake 相关，但本 PR 专注于 AMD HIP 内存回收，关联度较低。
- PR #32620 Eliminate redundant DSA state transfers (Mooncake): 同为 AMD 平台下的内存 / 传输优化，属于同一功能线的性能改进。
- PR #32701 [Perf] Free KV pages by segment in the paged allocator without a device sync: 同为 mem_cache 模块的分配器优化，关注内存回收，但方向不同。