PR #36343 完整报告

sgl-project/sglang

[AMD] Fall back to CPU tensor for decode retraction on ROCm

合并时间：2026-08-27 09:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36343

执行摘要

- 一句话：ROCm 回退 *decode* retraction 后备后端到 CPU tensor
- 推荐动作：该 PR 值得快速了解，因为它是一个重要的平台兼容性修复。应关注后续 #35233 的指针别名修复以及重新启用 host_pool 的时机。

功能与动机

PR #34801 使 HiCache 的 host_pool 成为符合条件的配置的隐式 *decode* retraction 后端。在 MI35x 上，注册的主机内存 CPU 和 GPU 虚拟地址可能不同，而当前 main 分支缺少 #35233 跟踪的设备可访问指针别名处理。因此，大规模 host-to-device retraction 恢复可能导致 GPU 进程出错。尽管底层 HiCache 内核有 HIP 覆盖，但基于 host_pool 的 PD *decode* retraction 路径在 MI35X 上尚不可靠或未端到端验证。

实现拆解

1. 在 python/sglang/srt/mem_cache/kv_cache_builder.py 中，导入 is_hip 工具函数。
2. 在 resolve_decode_retraction_backup 函数中，当选择 host_pool 后端时，增加 and not is_hip() 条件，使得在 HIP 平台上自动选择回退到 cpu_tensor。
3. 保留了非 ROCm 的默认行为和显式指定 host_pool 的选择。
4. 该变更仅影响自动选择逻辑，不影响显式配置。
5. 未添加新的测试文件，但删除了一个相关单元测试（提交信息表明）。

关键文件：

- python/sglang/srt/mem_cache/kv_cache_builder.py（模块 内存缓存；类别 source；类型 core-logic；符号 resolve_decode_retraction_backup, is_hip）：修改核心逻辑，通过 is_hip() 条件使 ROCm 平台回退到 cpu_tensor。

关键符号：resolve_decode_retraction_backup

关键源码片段

python/sglang/srt/mem_cache/kv_cache_builder.py

修改核心逻辑，通过 is_hip() 条件使 ROCm 平台回退到 cpu_tensor。

```
# python/sglang/srt/mem_cache/kv_cache_builder.py
```

```
# 新增导入
from sglang.srt.utils import is_hip

# ...

def resolve_decode_retraction_backup(*, tp_worker: BaseTpWorker) -> str:
    """决策 decode retraction 的后备后端，并写入运行时上下文。"""
    disagg = get_disagg()
    memory = get_memory()
    fields = {}

    backend = disagg.disaggregation_decode_retraction_backup
    if backend is None:
        kv_cache = tp_worker.get_memory_pool()[1].get_kvcache()
        full_tokens_per_layer = (
            tp_worker.get_tokens_per_layer_info()[0]
            if tp_worker.is_hybrid_swa
            else None
        )
        # host-pool retraction 只传输 full 和 sliding-window 组件，
        # 因此有 recurrent state 的模型保持在 cpu_tensor。
        supports_host_pool = not uses_ssm_state(
            tp_worker.model_runner.model_config
        ) and (
            isinstance(kv_cache, MHATokenToKVPool)
            or (isinstance(kv_cache, SWAKVPool) and full_tokens_per_layer > 0)
        )
        schedule = get_schedule()
        priority_preemption = (
            schedule.enable_priority_scheduling
            and not schedule.disable_priority_preemption
        )
        # ROCm 平台：大型 retraction 恢复可能使 GPU 进程崩溃，
        # 因此在 HIP 上强制回退到 cpu_tensor，直到 #35233 修复。
        backend = (
            "host_pool"
            if disagg.disaggregation_mode == "decode"
            and not is_hip() # 新增：避免在 HIP 上自动选择 host_pool
            and not get_parallel().dcp_enabled
            and not disagg.disaggregation_decode_enable_radix_cache
            and not disagg.disaggregation_decode_enable_offload_kvcache
            and not priority_preemption
            and supports_host_pool
            else "cpu_tensor"
        )
        fields["disaggregation_decode_retraction_backup"] = backend

    # 根据后端设置 pool 比例
    if memory.hicache_ratio is None:
```

```
        if backend == "host_pool" and not memory.enable_hierarchical_cache:
            fields["hicache_ratio"] = BACKUP_ONLY_HICACHE_RATIO
        else:
            fields["hicache_ratio"] = 2.0

    get_context().override("kv_cache_builder.decode_retraction", **fields)
    return backend
```

评论区精华

该 PR 没有 review 评论，但提交历史显示删除了一个 *decode* retraction resolver 单元测试，可能因为该测试需要适配新的回退逻辑。

- 暂无高价值评论线程

风险与影响

- 风险：该变更是兼容性回退，不修复 ROCm host_pool retraction 路径的根本问题，因此风险主要是功能上的：在 ROCm 上使用 host_pool 回退的性能优势将暂时不可用。此外，删除单元测试可能降低对相关逻辑的测试覆盖。
- 影响：该变更影响 ROCm 平台（MI35x 等）上使用 PD disaggregation 的 *decode* 服务器。它避免了潜在的 GPU 进程崩溃，但可能降低 retraction 性能，因为回退到 cpu_tensor。对于明确指定 host_pool 的用户，行为不变。对非 ROCm 平台无影响。
- 风险标记：核心路径变更，缺少测试覆盖，兼容性回退

关联脉络

- PR #34801 [HiCache] Host pool as implicit PD *decode* retraction backend: 该 PR 引入的变化导致此 PR 需要回退。
- PR #35233 [HiCache] Device-accessible pointer alias handling for host pool: 此 PR 计划修复导致此回退的根本问题。