

PR #36738 完整报告

sgl-project/sglang

[HiCache] Fence load-back behind the forward stream

合并时间: 2026-08-29 02:12

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36738>

执行摘要

- 一句话: HiCache 加载回写增加前向流栅栏, 修复数据竞争
- 推荐动作: 此 PR 值得精读, 因为它展示了并发控制中流同步的正确用法, 以及如何在不变计算逻辑的情况下解决数据竞争。值得关注的设计决策包括使用实例属性而非类级属性、将前向流引用注入控制器、以及测试 mock 的调整方式。对于涉及缓存并发控制或流同步的开发者有参考价值。

功能与动机

重叠调度中, 已结束或撤回请求的 KV 页在其已启动的前向轮次仍写入时被释放, 而复用这些页的计算流会隐含等待写入完成, 但 HiCache 的 load-back H2D 传输流没有此顺序保证, 导致加载操作可能覆盖前向写入, 损坏恢复的前缀。PR body 指出: 'The hicache load-back H2D has no such ordering: it writes the reallocated pages from the transfer stream. In-flight writes from the earlier round can therefore race the restore and corrupt a restored prefix.'

实现拆解

实现分三步:

1. 在 HiCacheController 初始化中新增 load_fence_stream 属性 (默认 None), 并在 start_loading 中当该属性非空时, 调用 l2_transfer_engine.host_to_device_stream.wait_stream(load_fence_stream) 前向流等待, 确保 H2D 复制在前向写入之后执行。
2. 在 Scheduler 初始化中, 当启用层级缓存时, 将 cache_controller.load_fence_stream 指向 model_runner.forward_stream, 为控制器提供前向流引用。
3. 测试更新: 在 test_hybrid_load_forwards_merged_pool_transfers 中显式设置 controller.load_fence_stream = None, 确保测试行为稳定。此改动不改变任何计算逻辑, 仅增加流顺序约束, 成本为每次 load-back 提交一次 wait_stream 调用。

关键文件:

- python/sglang/srt/managers/cache_controller.py (模块 缓存控制器; 类别 source; 类型 entrypoint): 添加 load_fence_stream 属性及实现栅栏等待逻辑, 是核心改动。
- python/sglang/srt/managers/scheduler.py (模块 调度器; 类别 source; 类型 core-logic): 在调度器初始化时注入前向流, 是触发栅栏的关键设置。

- test/registered/unit/mem_cache/test_hicache_staged_write_back_dispatch.py（模块 缓存测试；类别 test；类型 test-coverage）：测试适配新属性，确保 mock 控制器包含 load_fence_stream。

关键符号：start_loading, init

关键源码片段

python/sglang/srt/managers/cache_controller.py

添加 load_fence_stream 属性及实现栅栏等待逻辑，是核心改动。

```
# cache_controller.py
class HiCacheController:
    def __init__(self, ...):
        ...
        # Set by the scheduler to the forward stream; gates load-back H2D
        # behind in-flight forwards (see start_loading).
        self.load_fence_stream = None
        ...

    def start_loading(self) -> int:
        if len(self.load_queue) == 0:
            return -1

        producer_id = self.layer_done_counter.update_producer()
        op = CacheOperation.merge_ops(self.load_queue)
        host_indices, device_indices, pool_transfers = self._move_op_indices(op)
        self.load_queue.clear()
        producer_event = self.layer_done_counter.events[producer_id]
        producer_event.start_event.record()

        if self.load_fence_stream is not None:
            # in overlap scheduling, reclaimed pages might still be written by the forward thread
            # therefore a fence is needed for loading thread to prevent memory corruption
            # todo: it's possible to use a finer-grained fence
            self.l2_transfer_engine.host_to_device_stream.wait_stream(
                self.load_fence_stream
            )

        completion = self.l2_transfer_engine.submit_host_to_device(
            self._l2_load_transfers(host_indices, device_indices, pool_transfers),
            start_event=producer_event.start_event,
            on_layer_done=producer_event.complete,
            layer_num=self.layer_num,
        )
        ...
```

python/sglang/srt/managers/scheduler.py

在调度器初始化时注入前向流，是触发栅栏的关键设置。

```
# scheduler.py
# 在初始化完成后设置栅栏流
if self.enable_hierarchical_cache:
    cache_controller = self.tree_cache.cache_controller
    if cache_controller is not None:
        cache_controller.load_fence_stream = (
            self.tp_worker.model_runner.forward_stream
        )
```

评论区精华

PR 无 review 评论，但提交历史显示作者调整了属性声明方式：从类级属性改为实例属性，并在测试 mock 中显式设置，以匹配 `mock.Mock(spec=...)` 和 `__new__` 构造的实例行为。作者在提交消息中指出 'An instance-only attribute is therefore absent from the spec'，说明此调整旨在修正测试 mock 的规范匹配问题。

- 暂无高价值评论线程

风险与影响

- 风险：主风险在于流同步延迟可能影响性能：`wait_stream` 会使加载线程等待前向完成，若前向耗时较长，可能延迟 load-back，但 PR 说明这是预期行为，且 `wait_stream` 每次 load-back 仅一次。另一个风险是属性依赖：若 `model_runner.forward_stream` 在调度器初始化时未就绪，可能引用未初始化对象，但该属性在 hierarchical cache 启用时一定可用。此外，改动仅适用于 `HiCacheController` 及子类 `HybridCacheController`，其他控制器不受影响。
- 影响：影响范围限于启用层级缓存（`HiCache`/`HybridCache`）的场景，修复了恢复前缀的数据竞争问题，提高缓存正确性。对非层级缓存无影响。团队需关注该改动引入的流同步性能开销，但通常较小。测试覆盖有限，仅调整了现有测试，未新增针对数据竞争的单元测试，因为该竞争具有时序依赖性，难以确定性地复现。
- 风险标记：核心路径变更，并发控制，缺少确定性测试

关联脉络

- PR #36572 将模型前向流绑定到传输引擎的流（推断）：PR body 指出 #36572 实现了将 H2D 复制放入栅栏传输流的变更，本 PR 依赖该变更才能真正生效。
- PR #35931 [HiCache] Reject load-back specs that claim nodes pinned by an in-flight load-back: 同为 HiCache 调度器崩溃 / 数据完整性修复，涉及 load-back 线程安全。
- PR #36759 bugfix for `index_fill_` on NPU: 同为内存池分配器相关修复，但涉及 NPU 性能。