PR #29166 完整报告

sgl-project/sglang

[Fix]: Inline H2D during CUDA graph capture to avoid stream isolation in Offloader

合并时间：2026-07-01 08:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29166

执行摘要

- 一句话：修复 Offloader 在 CUDA graph 捕获时的流隔离错误
- 推荐动作：值得精读，尤其是理解 CUDA graph stream capture isolation 约束和异步 prefetch 的交互。设计上采用了条件分支来动态适配捕获状态，思路清晰。后续可考虑更优雅的方案（如预分配 pinned memory 或利用 cudaMemcpyAsync 的流顺序语义）来恢复部分 overlap 收益。建议在 merge 后跟踪 torch.compile 兼容性 issue。

功能与动机

CPU weight offloader 在 decode CUDA graph 捕获时，通过 alt_stream 异步 prefetch 并通过 CUDA event 同步，这引入了对未捕获流的依赖，触发 cudaErrorStreamCaptureIsolation 错误。PR body 明确指出根因在于 prefetch 和 event.wait() 跨流操作违反了 CUDA graph stream capture isolation 约束。

实现拆解

1. start_onload 方法改造（offloader.py:308-317）：在方法开头插入 torch.cuda.is_current_stream_capturing() 检测。若正在 CUDA graph 捕获，则跳过 alt_stream.wait_stream 和 alt_stream 上下文，直接在当前捕获流上同步创建 device tensors，并将 _load_event 置为 None。否则走原有 alt_stream 异步路径。
2. wait_and_get_device_tensors 方法改造（offloader.py:322-332）：同样先检测是否正在捕获。若正在捕获且 _load_event 不为 None（表明之前有跨流 prefetch 残留），则重新在当前流上内联创建 device tensors 并清除 event；若正在捕获但 _load_event 为 None，直接返回已有的 _device_tensors。非捕获路径则等待 event 后返回 tensors。
3. 测试与配置配套：本 PR 未包含测试文件变更，依赖已有 CI 验证（JustinTong0323 在 4×GB300 上复现并验证修复有效，GSM8K 准确率正常，但 decode 吞吐因内联 H2D 降至约 35 tok/s）。

关键文件：

- python/sglang/srt/utils/offloader.py（模块 Offloader；类别 source；类型 core-logic；符号 start_onload, wait_and_get_device_tensors）：CPU weight offloader 核心实现，本 PR 全部变更集中于此文件的两个方法：start_onload 和 wait_and_get_device_tensors。

关键符号：start_onload, wait_and_get_device_tensors

关键源码片段

python/sglang/srt/utils/offloader.py

CPU weight offloader 核心实现，本 PR 全部变更集中于此文件的两个方法：start_onload 和 wait_and_get_device_tensors。

```
# python/sglang/srt/utils/offloader.py (.head)

def start_onload(self):
    # 若当前流正在 CUDA graph 捕获，则跳过 alt_stream 异步路径，
    # 直接在捕获流上同步创建 device tensors，避免跨流事件隔离错误。
    if torch.cuda.is_current_stream_capturing():
        self._device_tensors = self._create_device_tensors()
        self._load_event = None # 清除 event 引用，告诉 wait 方法无需等待
        return
    # 非捕获路径：保持原有 alt_stream 异步 prefetch + event 记录
    self.alt_stream.wait_stream(torch.cuda.current_stream())
    with torch.cuda.stream(self.alt_stream):
        self._device_tensors = self._create_device_tensors()
        self._load_event = torch.cuda.Event()
        self._load_event.record()

def wait_and_get_device_tensors(self):
    assert self._device_tensors is not None
    if torch.cuda.is_current_stream_capturing():
        # 捕获模式：若存在残留 event（来自 warmup 期间的非捕获 prefetch），
        # 则重新内联创建 tensors 并清除 event；否则直接返回已有 tensors。
        if self._load_event is not None:
            self._device_tensors = self._create_device_tensors()
            self._load_event = None
        return self._device_tensors
    # 非捕获模式：等待异步 prefetch 完成
    if self._load_event is not None:
        self._load_event.wait()
    return self._device_tensors
```

评论区精华

JustinTong0323 指出了两个值得关注的 caveat：

1. 吞吐下降：内联 H2D 被捕获进 decode graph 后每个 step 都会重放，导致 offload 的 experts 每步都需拷贝一次，GLM-5.2-FP8 上 decode 降至约 35 tok/s，失去了 alt_stream 的 overlap 收益。建议在 PR 中明确标注性能前后对比。
2. torch.compile 兼容性：torch.cuda.is_current_stream_capturing() 不是 Dynamo traceable 的，在 tc_piecewise 模式下（prefill decode 分别 graph 捕获）可能触发 torch.compile 回退到 eager 模式，影响 prefill 性能。

这些反馈未被驳回，表明该 fix 在功能正确性上被接受，但性能和兼容性代价需在后续迭代中优化。

- 内联 H2D 带来的性能下降 (performance): 该 fix 在功能正确性上被接受，但性能代价需要在后续优化中解决，目前无进一步处理。
- torch.compile 兼容性 (design): 被记录为已知缺陷，但当前修复先解决崩溃问题，兼容性问题留待后续。

风险与影响

- 风险：
  1. 性能回归：内联 H2D 导致 decode 吞吐明显下降（约 35 tok/s），失去 alt_stream 重叠收益。对高吞吐敏感场景可能接受度低。
  2. torch.compile 兼容风险：is_current_stream_capturing() 不是 traceable，可能引起 Dynamo 回退至 eager，影响 prefill 性能。需在 tc_piecewise 或 torch.compile 场景下额外验证。
  3. 回归风险：非捕获路径行为完全保留，回归风险较小。 - 影响：影响范围仅限于启用了 CPU offload 且同时使用 CUDA graph 捕获的场景（如 decode 阶段 full cudagraph）。禁用 offload 或不使用 GPU graph 捕获的用户无影响。修复了 stream capture isolation 崩溃，但引入了 decode 阶段的性能降级。 - 风险标记：核心路径变更，缺少测试覆盖，性能潜在回归，CUDA graph 兼容性

关联脉络

- PR #29622 Budget EAGLE/STANDALONE draft KV pool in SWA pool configurators: 同为 offload 相关修复，关注 CUDA graph 和 offload 交互。
- PR #29352 [bug2] skip swa recovery on locked full kv: 同为 kv-cache 和 CUDA graph 相关 bugfix。