

PR #39474 完整报告

sgl-project/sglang

[qwen 3.8 next] reuse old cuda stream instead of endlessly creating streams

合并时间: 2026-09-15 16:36

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39474>

执行摘要

- 一句话: 调整 QSA indexer 流上下文, 复用已有 CUDA 流
- 推荐动作: 建议精读但不必深挖: 整个 diff 只有 4 行, 真正值得关注的是三点——(1) stream 上下文的 fork 点从 QKV 准备之前后移到之后, 以及 wait_stream 为什么没有跟着移动; (2) record_stream 与成对 wait_stream 构成的跨流契约; (3) PR 标题的“流不断创建”缺少量化与根因说明。若后续要修改 _prepare_qkv_gate 或 indexer 的读集合, 务必回到这段代码确认 wait_stream 的语义仍然成立; 同时建议给这类多流 overlap 改动补一条可复现的流数量断言或 profiler 基线, 避免只靠截图背书。

功能与动机

PR 标题直接点明问题: “reuse old cuda stream instead of endlessly creating streams”, 即 Qwen 3.8 next (qwen4_exp) 模型的双流 overlap 路径会持续创建 CUDA 流。PR body 的 Motivation 段没有文字说明, 只给了两张 profiler 前后对比截图 (prev / after), 用执行轨迹的差异来证明改动效果。结合代码上下文, 该路径位于 QSA 稀疏注意力的 indexer overlap 分支, 而 self_attention 会在每一层、每个 decode step 被调用, 任何流上下文层面的额外开销都会被层数放大, 因此作者选择在热点路径上做定点收敛。

实现拆解

1. 变更入口: `python/sglang/srt/models/qwen4_exp.py` 的 `Qwen4ExpAttention.self_attention`, 具体落在 `overlap_indexer` 分支 (条件为 `self.is_qsa` and `self.alt_stream` is not None and `get_is_capture_mode()` and `hidden_states.shape[0] < _QSA_INDEXER_OVERLAP_TOKEN_THRESHOLD`)。全 PR 共 4 增 4 删, 未触碰其他模块。
2. 核心调整: 把 `with torch.cuda.stream(self.alt_stream):` 的位置从 `self._prepare_qkv_gate(...)` 调用之前移到之后, 使其只包住 `self._compute_qsa_topk_indices(hidden_states, positions, forward_batch)`。`topk_indices` 依旧在 alt stream 上产生, 并继续通过 `attention_kwargs["topk_indices"]` 传给 `self.attn(q, k, v, forward_batch, **attention_kwargs)`, 数据契约没有变化。
3. 跨流同步契约保持不变: `current_stream = torch.cuda.current_stream()` 与 `self.alt_stream.wait_stream(current_stream)` 仍发生在 `_prepare_qkv_gate` 之前, 因此 alt stream 只等待更早入队的主流工作, indexer 与 QKV/gate 准备仍可重叠; `current_stream.wait_stream(self.alt_stream)` 与

`topk_indices.record_stream(current_stream)` 仍位于 `self.attn` 之前，保证 attention 消费前可见、并向缓存分配器登记跨流生命周期。

4. 对其他分支的影响：非 overlap 的 `elif self.is_qsa` 分支、以及非 capture 模式下的 QSA 路径保持单流执行，行为完全不变；`fused_sigmoid_mul` 与 `o_proj` 部分未改动。
5. 测试与配置配套：本次没有任何测试文件、配置项或文档改动。验证依赖作者反复重跑的 `test/registered/e2e/models/test_qwen4_exp_models.py` (4-gpu-b200, 最终通过) 以及 PR body 中的 profiler 前后截图，缺少针对流行为的单元级断言。

关键文件：

- `python/sglang/srt/models/qwen4_exp.py` (模块 模型实现；类别 source；类型 core-logic；符号 `self_attention`, `_compute_qsa_topk_indices`, `_prepare_qkv_gate`)：本 PR 唯一改动文件。Qwen 3.8 next (`qwen4_exp`) 模型实现中的 QSA 稀疏注意力双流 overlap 逻辑在此，stream 上下文位置调整即发生在 `Qwen4ExpAttention.self_attention`。
- `test/registered/e2e/models/test_qwen4_exp_models.py` (模块 端到端测试；类别 test；类型 test-coverage)：本 PR 未修改该文件，但它是唯一的验证手段：作者三次重跑该端到端测试 (4-gpu-b200) 才拿到通过，说明改动缺少单元级回归保护。

关键符号：`Qwen4ExpAttention.self_attention`, `_compute_qsa_topk_indices`, `_prepare_qkv_gate`

关键源码片段

`python/sglang/srt/models/qwen4_exp.py`

本 PR 唯一改动文件。Qwen 3.8 next (`qwen4_exp`) 模型实现中的 QSA 稀疏注意力双流 overlap 逻辑在此，stream 上下文位置调整即发生在 `Qwen4ExpAttention.self_attention`。

```
# Qwen4ExpAttention.self_attention (改动后形态)
# 变更实质：torch.cuda.stream(self.alt_stream) 上下文只包住 indexer 计算，
# 位置从 _prepare_qkv_gate 之前后移到之后；跨流同步语句保持原位。
def self_attention(
    self,
    positions: torch.Tensor,
    hidden_states: torch.Tensor,
    forward_batch: ForwardBatch,
) -> torch.Tensor:
    # 仅当 QSA、存在 alt_stream、处于 CUDA Graph 捕获 / 回放模式，
    # 且 token 数低于阈值时才启用双流 overlap 路径。
    overlap_indexer = (
        self.is_qsa
        and self.alt_stream is not None
        and get_is_capture_mode()
        and hidden_states.shape[0] < _QSA_INDEXER_OVERLAP_TOKEN_THRESHOLD
    )
    attention_kwargs = {}
    if overlap_indexer:
        # 仍然先让 alt_stream 等主流，但只等“更早入队”的主流工作：
        # indexer 只读 hidden_states / positions，因此故意不等待后面的 QKV 准备，
```

```

# 这样 indexer 才能与 _prepare_qkv_gate 的 GEMM 真正重叠。
current_stream = torch.cuda.current_stream()
self.alt_stream.wait_stream(current_stream)

# 顺序变化点之一: QKV + gate 准备先入队到主流。
q, k, v, gate = self._prepare_qkv_gate(
    positions=positions,
    hidden_states=hidden_states,
    forward_batch=forward_batch,
)

if overlap_indexer:
    # 顺序变化点之二: stream 上下文只包住 indexer 计算本身。
    # 旧写法把这个上下文放在 _prepare_qkv_gate 之前, PR 标题称其会导致
    # “不断创建流”; 改动后 indexer 仍在同一条 self.alt_stream 上执行,
    # 只是 fork 时机挪到了 QKV 准备入队之后。
    with torch.cuda.stream(self.alt_stream):
        topk_indices = self._compute_qsa_topk_indices(
            hidden_states, positions, forward_batch
        )
    # 主流在此等待 alt_stream: 保证 attention 消费 topk_indices 前已完成。
    current_stream.wait_stream(self.alt_stream)
    # topk_indices 在 alt_stream 上分配、由主流上的 attention 消费,
    # 因此在 alt_stream 被复用前把生命周期记到主流上, 避免提前回收。
    topk_indices.record_stream(current_stream)
    attention_kwargs["topk_indices"] = topk_indices
elif self.is_qsa:
    # 非 overlap (非 capture 或超阈值) 路径保持单流执行, 行为不变。
    attention_kwargs["topk_indices"] = self._compute_qsa_topk_indices(
        hidden_states, positions, forward_batch
    )

attn_output = self.attn(q, k, v, forward_batch, **attention_kwargs)
if gate is not None:
    if attn_output.is_cuda:
        # strided 3D gate 视图直接喂融合内核, 省掉 reshape 拷贝。
        attn_output = fused_sigmoid_mul(attn_output, gate, inplace=True)
    else:
        gate = gate.reshape(gate.shape[0], -1) if gate.ndim == 3 else gate
        attn_output = attn_output * torch.sigmoid(gate)
output, _ = self.o_proj(attn_output)
return output

```

评论区精华

本 PR 没有行内 review 评论, 讨论几乎全部集中在 CI 重跑: 作者 Qiaolin-Yu 连续三次发送 [/rerun-test test/registered/e2e/models/test_qwen4_exp_models.py](#), 第一次 4-gpu-b200 失败, 第二次被 CI Maintenance Mode (issue #21065) 拦截未派发, 第三次才通过。评审

侧只有 YAMY1234 的一条 APPROVED “LGTM”，没有对 stream 语义、执行顺序变化或“流不断创建”的根因做技术追问。另一个值得注意的事实是合并者与作者同为 Qiaolin-Yu，属于自审自并。

- 端到端回归测试三次重跑才通过 (testing): 最终 end-to-end 通过，但过程中出现过失败，且没有为本次改动补充任何针对流行为的单元测试。
- 评审意见只有 LGTM，缺少并发语义讨论 (design): 快速通过；跨流依赖前提 (indexer 只读 hidden_states/positions) 依赖代码注释而非测试来保障。
- “流不断创建”的根因与量化证据缺失 (documentation): 未解决：读者需要自行用 profiler 复核因果，后续维护者难以判断该顺序约束是否是硬性要求。

风险与影响

- 风险：
 1. 同步契约脆弱（跨流顺序变更）：self.alt_stream.wait_stream(current_stream) 的调用位置没有跟随 with 块一起后移，这是保留 overlap 窗口的有意设计，但它隐含依赖“indexer 只读 hidden_states/positions、只写 QSA 私有 pool buffer”这条注释里的前提。一旦后续有人让 _prepare_qkv_gate 原地改写 hidden_states，或让 indexer 读取 Q/K/V，就会出现跨流读后写竞态，而当前没有测试能捕获。
 1. 缺少测试覆盖：没有新增单测；唯一覆盖是端到端模型测试 `test/registered/e2e/models/test_qwen4_exp_models.py`，且该测试在本 PR 期间曾失败过一次，说明其稳定性与定位能力有限，无法验证“流不再增长”这一诉求。
 2. 缺少量化证据与根因说明：diff 中没有新增或删除任何 stream 创建语句，PR body 的 Motivation 也只有两张截图，没有 step time、显存或流数量的量化对比，因此“旧写法导致流持续创建”的因果链条需要读者自行用 profiler 复核。
 3. CUDA Graph 捕获路径：该分支只在 `get_is_capture_mode()` 为真时生效，改变 kernel 的入队顺序会改变捕获到的图拓扑（fork/join 位置），若图稳定性或 replay 行为出现异常，排查成本较高。
 4. 审查强度不足：仅一个 LGTM、无行内讨论，且作者自行合并，对热点路径上的并发改动而言把关偏弱。
 - 影响：对用户而言，改动只影响 Qwen 3.8 next (qwen4_exp) 模型在 CUDA Graph 捕获 / 回放模式下的 QSA 稀疏注意力路径，预期收益是 profile 中流对象数量与流上下文开销下降；其他模型、其他后端（非 QSA 或非 capture 场景）不受影响，因此回归面很窄。对系统而言，收益上限受限于 `_QSA_INDEXER_OVERLAP_TOKEN_THRESHOLD` 阈值以下的请求比例，长序列 / 大 batch 不会走到该分支。对团队而言，这是一次典型的“热点路径复用资源而非反复申请”的微优化，改动仅 4 行却涉及跨流依赖前提，值得作为多流 overlap 代码契约维护的反面 / 正面案例记录到相关代码注释或贡献指南中。
 - 风险标记：跨流依赖契约脆弱，缺少单元测试覆盖，无量化性能证据，作者自审自并，捕获路径入队顺序变更

关联脉络

- PR #37748 [CPU] Implement fused QK Norm and RoPE kernels: 同属 `sglang/srt/models/` 下 qwen 系列模型的注意力路径性能改造：该 PR 修改

qwen3_moe.py 与 models/utils.py 并接入共享融合实现，与本 PR 一样在模型前向热点路径做小范围性能优化，可对照其测试与接入方式。

- PR #39278 [Fix][Qwen-VL] Normalize sentinel on artifact fast path: 同为 Qwen 系列模型的“快路径 / 快速分支”行为修复 (multimodal/processors/qwen_vl.py)，反映 Qwen 模型线正在密集迭代快速路径上的边界行为，与本 PR 在 capture 快路径上的定点修复属于同一类问题形态。
- PR #36821 [KDA] Support ReplaySSM ring-write in the fused chain-verify kernel: 同样关注 decode / verify 阶段的 kernel 组合与执行路径优化 (fused KDA verify + backend 接入)，是本仓库近期“流与内核组合优化”方向的可比案例，可作为评估本 PR 收益证据强度的参照。