PR #29463 完整报告

sgl-project/sglang

[DeepSeek V3] Reland: run routed experts on main stream in dual-stream MoE

合并时间：2026-06-30 06:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29463

执行摘要

- 一句话：修复 DeepSeek V3 双流 MoE 中 CUDA 图捕获顺序导致的精度崩溃
- 推荐动作：
  - 值得精读：本 PR 揭示了 CUDA 图捕获中一个极其微妙且危险的陷阱——host-side 张量元数据修改与 CUDA 图捕获的交互问题。
  - 值得学习的设计权衡：为性能收益（~1%），作者选择了"按顺序即正确"的方案而非更健壮的修复，但记录了明确的注释和未来改进方向。
  - 测试覆盖：虽然无直接新增的单元测试，但通过 test_moe_ep_extra.py 的 gsm8k 测试（精度 > 0.60）验证了正确性。

功能与动机

原 PR #29142 通过将路由专家放置在主流上以利用 PDL-fuse 获得约 1% 的性能增益，但被回滚因为 gsm8k 精度从 ~0.7 跌至 0.005。根因并非跨流竞争，而是 dispose_tensor（host-side 通过 set_ 将 hidden_states 的 data_ptr 置零）在共享专家内核捕获之前被执行，导致CUDA 图记录了无效地址。本 PR 保留性能收益的同时，通过调整语句顺序修复精度问题。

实现拆解

1. 保留流分配方案：沿用 #29142 的流分配——路由专家在主流上执行，共享专家在替代流上执行，使得路由专家成为主流的最后一个内核，可以与 MoE 后的残差相加进行 PDL-fuse。
2. 调整执行顺序：将共享专家计算块（with torch.cuda.stream(self.alt_stream): shared_output = ...）移动到路由专家管道之前，确保共享专家 GEMM 内核在 CUDA 图捕获时读取到有效的 hidden_states.data_ptr()。
3. 路由专家保持不变：路由管道的 self.gate, self.topk, self.experts() 调用顺序不变，仍在主流上执行。
4. 同步不变：current_stream.wait_stream(self.alt_stream) 仍放在路由专家之后，合并 shared_output 之前。
5. 关键注释：在 forward_normal_dual_stream 方法头部新增了说明注释，记录必须先启动共享专家的原因，防止未来类似问题。

关键文件：

- python/sglang/srt/models/deepseek_v2.py（模块 模型路由；类别 source；类型 core-logic；符号 forward_normal_dual_stream）：核心变更文件：调整了 forward_normal_dual_stream 中共享专家与路由专家的执行顺序，修复了 CUDA 图捕获中的 data_ptr 失效问题，保留了路由专家在主流上执行的性能优化。

关键符号：forward_normal_dual_stream

关键源码片段

python/sglang/srt/models/deepseek_v2.py

核心变更文件：调整了 forward_normal_dual_stream 中共享专家与路由专家的执行顺序，修复了 CUDA 图捕获中的 data_ptr 失效问题，保留了路由专家在主流上执行的性能优化。

```
# python/sglang/srt/models/deepseek_v2.py

def forward_normal_dual_stream(
    self,
    hidden_states: torch.Tensor,
    should_allreduce_fusion: bool = False,
    use_reduce_scatter: bool = False,
    gemm_output_zero_allocator: BumpAllocator = None,
    input_ids: Optional[torch.Tensor] = None,
    input_ids_global: Optional[torch.Tensor] = None,
    *,
    use_flashinfer_trtllm_bypass: bool = False,
) -> torch.Tensor:
    # 关键注释：必须在 routed call 之前启动 shared expert.
    # routed 的 deep_gemm 预置中调用 dispose_tensor,
    # 它会将 hidden_states 的 data_ptr 通过 set_() 置为空张量 (host-side 操作 ).
    # 如果 shared expert 的内核在 routed call 之后才启动，则它捕获到 data_ptr == 0,
    # 在 decode CUDA 图重放时会读空地址，导致垃圾输出 .
    current_stream = torch.cuda.current_stream()
    self.alt_stream.wait_stream(current_stream)
    server_args = get_global_server_args()
    dispatch_info = (
        ExpertLocationDispatchInfo.init_new(layer_id=self.layer_id)
        if server_args.enable_eplb
        else None
    )

    # 第 1 步：在 alt 流上执行 shared expert ( 先执行，确保捕获有效地址 )
    with torch.cuda.stream(self.alt_stream):
        shared_output = self._forward_shared_experts(
            hidden_states, gemm_output_zero_allocator
        )

    # 第 2 步：在主流上执行 routed expert (gate + topk + experts)
    # 此时 dispose_tensor 才被调用，但 shared_output 已捕获完成
    router_logits = self.gate(hidden_states, gemm_output_zero_allocator)
```

```
# ... topk / experts 调用序列不变 ...
# ... finalize 和 allreduce 保持不变 ...
current_stream.wait_stream(self.alt_stream)
# ... 后续 merge shared_output 和 allreduce ...
return final_hidden_states
```

评论区精华

无 reviewer 讨论。作者 kpham-sgl 在 PR 描述中详细分析了根因：

- 排除了“跨流竞争”和“torch.compile flattening”的误判。
- 明确写入者 / 读者对：写入者是 pre_permute_standard_to_deep_gemm 中的 hidden_states.set_(empty(0))，读者是共享专家 GEMM 内核启动。
- 通过二分测试确认：共享专家先执行时精度 0.66，路由先执行时精度 0.005。
- CUDA 图捕获中 dispose_tensor 写后读依赖 (correctness)：结论：将 shared expert 启动移前即可规避，无需更复杂的修复。作者同时指出了更健壮的未来方向：让 dispose_tensor 操作 dispach/permuted 缓冲区而非原始张量。

风险与影响

- 风险：
  1. 时序脆弱性：修复依赖 Python 语句顺序，仍然脆弱。任何未来新增的 dispose_tensor 调用或类似 host-side 修改张量元数据的操作都可能再次触发相同问题。PR 中已指出更健壮的修复方案（让 dispose_tensor 在 dispach/permuted 缓冲区上操作，而非原始张量），但未实施。
  2. 回归风险：仅修改了 deepseek_v2.py 中的控制流顺序，无风险文件。
  3. 性能影响：保留了 #29142 的性能收益（路由专家作为主流最后一个内核，可 PDL-fuse），无性能回归。
- 影响：
  - 用户影响：所有使用 DeepSeek V3 或类似模型且启用 deep_gemm 后端的用户将受益于约 1% 的性能提升（来自 #29142），同时避免精度崩溃。
  - 系统影响：仅影响单文件 deepseek_v2.py，且仅在 forward_normal_dual_stream 路径中生效，其他路径无影响。
  - 团队影响：无额外维护负担。
  - 风险标记：核心路径变更，时序依赖脆弱，无新增测试，性能收益与精度修复

关联脉络

- PR #29142 [DeepSeek V3] run routed experts on main stream in dual-stream MoE: 原 PR：将路由专家迁移到主流以利用 PDL-fuse，但因精度崩溃被回滚。本 PR 是重新提交，保留了其流分配方案并修复了时序问题。
- PR #29452 Revert #29142 due to gsm8k regression: 回滚 PR：因 #29142 导致 test_moe_ep_extra.py 精度从 0.7 降至 0.005，本 PR 修复了该问题，不再需要回滚。