PR #25279 完整报告

sgl-project/sglang

DeepseekV2MoE: defer shared experts when routed kernel is non-mutating

合并时间：2026-05-15 13:20

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25279

执行摘要

- 一句话：延迟共享专家计算以降低峰值显存
- 推荐动作：该 PR 值得关注其设计思路：通过改变算子执行顺序而非引入新算子来优化显存。建议精读 deepseek_v2.py 中 forward_normal 方法的变更逻辑。

功能与动机

当 routed-MoE 内核不就地修改输入时（inplace=False），hidden_states 在 self.experts(...) 调用后仍然存活。原有的先计算共享专家再调用 routed 的顺序，导致共享专家激活和 routed workspace 同时驻留 GPU，在长 prefill 场景下推高瞬时峰值显存。

实现拆解

1. 确定延迟条件：在 forward_normal 方法顶部读取 self.experts.moe_runner_config.inplace，当为 False 时设置 defer_shared = True，表示可延迟共享专家计算。
2. 调整调用顺序：
   - 当 defer_shared 为 False（如 Triton fused_moe 等 in-place 内核）时，保持原有顺序：先调用 _forward_shared_experts，再执行 routed 分发。
   - 当 defer_shared 为 True 时，跳过 pre-routed 的共享专家调用，在 routed 调用完成后再调用 _forward_shared_experts，使得 routed workspace 可在共享专家分配前被释放。
3. 保持向后兼容：FusedMoE 和 MoeRunnerConfig 定义未改动，仅修改调用点顺序。对于 in-place 内核，行为 bit-for-bit 不变。

关键文件：

- python/sglang/srt/models/deepseek_v2.py（模块 模型层；类别 source；类型 core-logic；符号 forward_normal）：DeepSeek V2/V3/V4 模型的核心 MoE 前向逻辑，通过调整共享专家调用顺序降低长 prefill 的峰值显存。

关键符号：forward_normal

关键源码片段

python/sglang/srt/models/deepseek_v2.py

DeepSeek V2/V3/V4 模型的核心 MoE 前向逻辑，通过调整共享专家调用顺序降低长 prefill 的峰值显存。

```
# python/sglang/srt/models/deepseek_v2.py — DeepseekV2MoE.forward_normal 部分
# 判断 routed kernel 是否就地修改输入；若不就地修改，则可延迟共享专家计算
# 以降低峰值显存（routed workspace 可在共享专家前释放）
defer_shared = not self.experts.moe_runner_config.inplace

if hidden_states.shape[0] > 0:
    # 只有非延迟场景（in-place kernel）才在 routed 之前计算共享专家
    if not defer_shared and not self._fuse_shared_experts_inside_sbo:
        shared_output = self._forward_shared_experts(
            hidden_states, gemm_output_zero_allocator
        )
    # ... 后续 router_logits, topk 计算不变 ...

# 执行 routed MoE 内核（此时不会修改 hidden_states）
final_hidden_states = self.experts(hidden_states, topk_output)

# 在 routed 调用之后，若为延迟场景且需要共享专家，则在此计算
# 此时 routed workspace 已可释放，避免与共享专家激活同时驻留 GPU
if (
    defer_shared
    and hidden_states.shape[0] > 0
    and not self._fuse_shared_experts_inside_sbo
):
    shared_output = self._forward_shared_experts(
        hidden_states, gemm_output_zero_allocator
    )
```

评论区精华

Review 中 gemini-code-assist[bot] 建议在 forward_normal 函数开头初始化 shared_output = None，以避免潜在的 UnboundLocalError（当前逻辑虽能确保使用前赋值，但初始化更安全清晰）。该评论未被采纳或回复。

- shared_output 变量初始化安全性 (correctness): 未采纳或回复，当前代码未作修改。

风险与影响

- 风险：
  1. 仅当 defer_shared = True 且 hidden_states.shape[0] > 0 且 _fuse_shared_experts_inside_sbo 为 False 时才会走新路径，其他情况行为不变，回归风险低。
  2. 若未来引入新的 routed 内核且 inplace 属性设置错误，可能导致共享专家未被计算。依赖 moe_runner_config.inplace 的正确性。
  3. 未新增测试覆盖延迟路径，需依赖现有 CI。 - 影响：影响范围：仅影响使用 non-mutating routed MoE 内核（如 flashinfer_trtllm_routed）的 DeepSeek

V2/V3/V4 类模型，TP=4 的 Kimi-K2.5-NVFP4 在 16K prefill 场景下显存峰值降低。影响程度：中等，属于性能优化且不改动接口和语义。 - 风险标记：核心路径变更，缺少测试覆盖，依赖配置正确性

关联脉络

- PR #24884 [MoE] Decouple Mega MoE from DeepEP backend: 都涉及 DeepSeek V2/V3/V4 的 MoE 层逻辑，同一文件 deepseek_v2.py 被修改。
- PR #24691 [UnifiedTree]: Support HiCache For DeepSeek_V4: DeepSeek V4 相关优化，同一模型系列。