PR #25367 完整报告

sgl-project/sglang

Fix EPLB redundant experts with shared expert fusion and Waterfill

合并时间：2026-05-21 13:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25367

执行摘要

- 一句话：修复 EPLB 冗余专家与 DeepEP Waterfill 冲突
- 推荐动作：值得精读。PR 修复了一个在冗余专家场景下的非明显 bug，展示了 EPLB 与 DeepEP Waterfill 之间的交互依赖。设计决策（如区分逻辑 / 物理 expert 计数、共享权重的槽位映射）对理解 MoE 系统具有参考价值。

功能与动机

PR body 说明：当 EPLB 添加冗余专家后，DeepEP Waterfill 使用的 n_routed_experts（逻辑计数）与实际的物理 routed expert 数量不符，导致共享专家槽位 remap 错误；同时 fused shared checkpoint 的权重加载使用了 expert_id >= _num_global_routed 的判断，而该全局计数不含冗余专家，在冗余专家存在时共享权重无法正确映射到物理槽位。

实现拆解

1. 传递物理 routed expert 计数给 DeepEP Waterfill - 在 python/sglang/srt/model_executor/model_runner.py 的 _prepare_moe_topk() 中，将 num_routed_experts 从逻辑数（n_routed_experts）改为物理数（num_routed_experts + server_args.ep_num_redundant_experts），并传入 DeepEPWaterfillBalancer 的 num_routed_experts 参数。 - 确保 Waterfill 的共享专家槽位计算使用正确的物理 routed expert 分组大小。
2. 修复 fused shared checkpoint 权重映射 - 在 python/sglang/srt/layers/moe/fused_moe_triton/layer.py 的 weight_loader() 中，将原来的 expert_id >= _num_global_routed 判断替换为基于 num_logical_experts 的 shared_expert_id 计算。 - 新增逻辑：当 require_global_experts 且使用 DeepEP 后端时，根据 ep_rank 和 _num_local_routed 计算物理 shared expert id 列表；否则使用 _num_global_routed + shared_expert_id。
3. 更新 _remap_topk_for_deepep 函数签名与调用 - 在 python/sglang/srt/layers/moe/topk.py 中，将参数 n_routed_experts 改为 num_physical_routed_experts，并在调用处从 router_logits.shape[1] 改为从 expert_location_dispatch_info.num_physical_experts 获取（若存在），否则 fallback 到 router_logits.shape[1]。 - 确保 DeepEP interleaved layout 的 remap 步骤使用物理 routed expert 数量计算 num_local_routed。

关键文件：

- python/sglang/srt/layers/moe/fused_moe_triton/layer.py（模块 MoE 层；类别 source；类型 core-logic；符号 weight_loader）：修复 fused shared checkpoint 权重加载时物理 expert id 映射逻辑，核心变更点。
- python/sglang/srt/layers/moe/topk.py（模块 MoE 层；类别 source；类型 core-logic；符号 _remap_topk_for_deepep, _post_process_topk_ids）：修复 _remap_topk_for_deepep 函数中物理 routed expert 计数，确保 Waterfill 共享槽位计算正确。
- python/sglang/srt/model_executor/model_runner.py（模块 模型运行器；类别 source；类型 data-contract；符号 _prepare_moe_topk）：将 Waterfill balancer 初始化时的 expert 计数从逻辑数改为物理数，触发整个修复链路。

关键符号：_prepare_moe_topk, weight_loader, _remap_topk_for_deepep, _post_process_topk_ids

关键源码片段

python/sglang/srt/layers/moe/topk.py

修复 _remap_topk_for_deepep 函数中物理 routed expert 计数，确保 Waterfill 共享槽位计算正确。

```
# python/sglang/srt/layers/moe/topk.py ( 部分 )

def _remap_topk_for_deepep(
    topk_ids: torch.Tensor,
    topk_weights: torch.Tensor,
    num_fused_shared_experts: int,
    num_physical_routed_experts: int, # 从 n_routed_experts 改为物理计数
    topk_config: TopKConfig,
) -> tuple[torch.Tensor, torch.Tensor]:
    # ...
    ep_size = get_moe_expert_parallel_world_size()
    ep_rank = get_moe_expert_parallel_rank()
    # 由于 topk_ids 已经被 remap 为物理 id，因此这里必须使用物理 routed 计数
    num_local_routed = num_physical_routed_experts // ep_size
    num_local_experts = num_local_routed + num_fused_shared_experts
    # ... 后续 remap 逻辑不变

# 在 _post_process_topk_ids 中调用处
if num_fused_shared_experts > 0 and is_deepep_class_backend():
    # 优先从 expert_location_dispatch_info 获取物理 expert 数量
    num_physical_routed_experts = (
        expert_location_dispatch_info.num_physical_experts
        if expert_location_dispatch_info is not None
        else router_logits.shape[1] # fallback 到逻辑数（兼容旧模式）
    )
    topk_ids, topk_weights = _remap_topk_for_deepep(
        topk_ids,
```

```
        topk_weights,
        num_fused_shared_experts,
        num_physical_routed_experts,
        topk_config,
    )
```

评论区精华

该 PR 审核人 ch-wan 直接批准（APPROVED），无 review 评论。从 commits 看，作者在第一次提交后两次合并 main 分支，可能解决了 CI 冲突或与上游保持同步。PR body 中的性能数据（+2.27% / +3.30% throughput，MMLU 无回归）说明修改正确且有效。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 回归风险：修改了 Waterfill 和 fused shared expert 权重映射的核心逻辑，但已有性能测试和 MMLU 无回归验证，风险可控。
  2. 兼容性：num_physical_routed_experts 在 expert_location_dispatch_info 为 None 时 fallback 到 router_logits.shape[1]，与原有行为兼容。
  3. 耦合性：model_runner.py 中硬编码引用了 server_args.ep_num_redundant_experts，如果该参数未设置或在不同上下文中含义不同，可能导致计算错误。 - 影响：影响范围：仅限于使用 DeepEP Waterfill 且启用了 EPLB 冗余专家（ep_num_redundant_experts > 0）的 DeepSeek V3 等 MoE 模型。对于无冗余专家的场景，ep_num_redundant_experts 为 0，行为与之前一致。性能提升约 2-3%，无明显副作用。 - 风险标记：核心路径变更，依赖 server_args 参数

关联脉络

- PR #25907 Fix FlashInfer A2A token cap sizing: 同属 MoE 模块的 bugfix，涉及 token routing 的计数修正。
- PR #25824 [Refactor] Encapsulate SWA loc translation inside SWAKVPool with per-batch cache invalidation: 同样涉及 DeepSeek V4 系统的底层修复，与 MoE 负载均衡相关。