PR #35949 完整报告

vllm-project/vllm

[MoE Refactor] Move the shared/fused expert output sum into MoERunnerBase

合并时间：2026-04-21 00:28

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/35949

执行摘要

- 一句话：将共享 / 融合专家输出求和移入 MoERunnerBase
- 推荐动作：该 PR 是 MoE 重构系列的核心部分，值得精读。重点理解 apply_routed_scale_to_output 的设计决策以及基类如何通过 _fused_output_is_reduced 状态跟踪 reduce 状态。建议关注后续 MoE runner 的进一步抽象。

功能与动机

PR 描述指出目标是将共享和融合专家输出的求和以及最终的 shared/fused all reduce 代码移入 MoERunnerBase，并删除使用 SharedFusedMoE 的模型中的相应代码。这样可以减少模型代码中的重复逻辑，确保所有 MoE 层共享一致的 reduce 行为，降低维护成本。

实现拆解

1. 在 MoERunnerBase 中新增输出后处理逻辑（moe_runner_base.py）新增 _unpack 辅助函数，用于统一处理 _moe_forward 返回的张量或元组。新增 apply_routed_output_transform、trunc、reduce_and_trunc 方法，负责共享专家输出求和、缩放因子应用及 TP all-reduce。
2. 重构 FusedMoE 层配置（layer.py）移除 reduce_results 参数，新增 routed_output_transform 和 apply_routed_scale_to_output 参数。当 apply_routed_scale_to_output 为 True 时，将路由器的 routed_scaling_factor 设为 1.0，避免重复缩放，由运行器在输出时统一处理。
3. 简化 MoERunner 抽象接口（moe_runner.py）从抽象基类中移除 must_reduce_shared_expert_outputs 和 maybe_all_reduce_tensor_model_parallel 方法，forward 返回类型统一为单个张量。这些逻辑现在由 MoERunnerBase 直接提供。
4. 更新所有使用 SharedFusedMoE 的模型修改 transformers/moe.py、deepseek_v2.py、exaone_moe.py、kimi_linear.py、ernie45_vl_moe.py 等模型文件，删除它们各自的手动共享专家求和与 all-reduce 代码，改为直接调用 SharedFusedMoE 的 forward，由运行器自动处理后处理。

关键文件：

- vllm/model_executor/layers/fused_moe/runner/moe_runner_base.py（模块 MoE 运行器；类别 source；类型 data-contract；符号 _unpack, apply_routed_output_transform, trunc, reduce_and_trunc）：核心更改文件，新增了 apply_routed_output_transform、

trunc、reduce_and_trunc 等方法，移除了 reduce_results 参数，集成了 shared expert 求和与 TP reduce 逻辑。

- vllm/model_executor/layers/fused_moe/layer.py（模块 MoE 层；类别 source；类型 data-contract；符号 must_reduce_shared_expert_outputs, maybe_all_reduce_tensor_model_parallel）：移除了 reduce_results 参数，新增了 routed_output_transform 和 apply_routed_scale_to_output 配置，调整了缩放因子传递逻辑。
- vllm/model_executor/models/transformers/moe.py（模块 模型适配层；类别 source；类型 data-contract；符号 add_all_reduce, MLPWithAllReduce, forward）：移除了 add_all_reduce 内联函数和 MLPWithAllReduce 动态类，不再手动包裹 all-reduce，简化模型代码。
- vllm/model_executor/layers/fused_moe/runner/moe_runner.py（模块 运行器接口；类别 source；类型 data-contract；符号 must_reduce_shared_expert_outputs, maybe_all_reduce_tensor_model_parallel）：从 MoERunner 接口中移除了 must_reduce_shared_expert_outputs 和 maybe_all_reduce_tensor_model_parallel 抽象方法，forward 返回类型简化。
- vllm/model_executor/models/exaone_moe.py（模块 EXAone 模型；类别 source；类型 data-contract）：迁移到 SharedFusedMoE，将共享专家创建逻辑提前，删除手动求和与 reduce 代码。

关键符号：_unpack, apply_routed_output_transform, trunc, reduce_and_trunc, maybe_all_reduce_tensor_model_parallel

关键源码片段

vllm/model_executor/layers/fused_moe/runner/moe_runner_base.py

核心更改文件，新增了 apply_routed_output_transform、trunc、reduce_and_trunc 等方法，移除了 reduce_results 参数，集成了 shared expert 求和与 TP reduce 逻辑。

```
def apply_routed_output_transform(
    self,
    hidden_states: torch.Tensor,
    shared_experts_output: torch.Tensor | None,
) -> torch.Tensor:
    # 如果存在共享专家输出，先相加
    if shared_experts_output is not None:
        hidden_states = hidden_states + shared_experts_output

    # 如果需要将路由缩放因子应用到输出（而非路由器），则相乘
    # 避免 fp16 溢出，通常用于 DeepSeek 等模型
    if self._apply_routed_scale_to_fused_output:
        hidden_states = hidden_states * self.routed_scaling_factor

    # 如果融合输出尚未经过 TP all-reduce，则执行
    if not self._fused_output_is_reduced:
        hidden_states = tensor_model_parallel_all_reduce(hidden_states)
```

```
    return hidden_states
```

vllm/model_executor/layers/fused_moe/layer.py

移除了 reduce_results 参数，新增了 routed_output_transform 和 apply_routed_scale_to_output 配置，调整了缩放因子传递逻辑。

```
# 在 FusedMoE.__init__ 中
# 当 apply_routed_scale_to_output 为 True 时，将路由器中的缩放因子设置为 1.0，
# 避免路由器再次应用缩放；缩放由运行器在输出时统一处理
self.routed_scaling_factor = (
    routed_scaling_factor if not apply_routed_scale_to_output else 1.0
)
self.apply_routed_scale_to_output = apply_routed_scale_to_output
```

评论区精华

> "The apply_scale_to_output is very confusing... It should have a better name" ——robertgshaw2-redhat 决策：重命名为 apply_routed_scale_to_fused_output。
>
> "I find this to be confusing --- who must_reduce_shared_expert_output?" ——robertgshaw2-redhat 决策：添加注释说明。
>
> "are we sure we want to make this default to false? we have missed some cases (e.g. glm4)" ——robertgshaw2-redhat 决策：显式设置标志。
>
> "this function was dead code right?" ——robertgshaw2-redhat 关于 _maybe_reduce_shared_out 回答：该函数仍被使用，已迁移。
>
> "I dont think this is right. We are now going to apply this twice I think?" ——robertgshaw2-redhat 关于 deepseek_v2 结论：不会重复，因为标志已阻止。

- apply_scale_to_output 命名改进 (design): 重命名为 apply_routed_scale_to_fused_output
- must_reduce_shared_expert_outputs 接口混淆 (design): 添加注释说明该函数仅由 MoERunnerBase 内部使用
- 默认值导致 glm4 缩放因子缺失 (correctness): 显式传递 apply_routed_scale_to_output=True 给需要该行为的模型
- shared_experts 中 _maybe_reduce_shared_out 功能迁移 (correctness): 功能保留在 MoERunnerBase 中，SharedExperts 类中删除该方法
- deepseek_v2 缩放因子重复应用担忧 (correctness): 不会重复，因为 apply_routed_scale_to_output 标志阻止了路由器中的重复应用

风险与影响

- 风险：主要风险包括：1）TP all-reduce 移动可能导致新路径下的额外通信或遗漏，需覆盖多种并行配置的测试。2）多个模型共享同一基类逻辑，若条件判断有误可能引入回归。3）缩放因子应用时机变化可能影响某些量化方法的数值精度。

- 影响：对用户：MoE 模型推理结果应保持不变（测试验证了 17 个模型的精度基线）。对开发者：新增 MoE 模型不再需要手动处理 shared expert 求和与 reduce，简化了模型实现。对系统：统一了 MoE 运行器架构，降低了技术债务。
- 风险标记：核心路径变更，跨模型影响，多模型测试覆盖

关联脉络

- PR #40574 [MoE] Move cutlass moe to fused_moe/experts/: 该 PR 将 CUTLASS MoE 移动到 experts 子目录，与本 PR 的 MoE 运行器重构属于同一功能线。
- PR #40794 [Bugfix][MoE] Unpad routed output before shared expert add [Fixes #35949]: 该 PR 修复了共享专家添加前的解填充问题，与本 PR 的共享专家求和逻辑直接相关。