# PR #20736 完整报告

sgl-project/sglang

[AMD] Enable share expert fusion with router experts for Qwen3.5 BF16 & FP8

合并时间：2026-04-15 09:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/20736

## 执行摘要

- 一句话：为 AMD 平台 Qwen3.5 MoE 模型启用共享专家融合，减少内核启动以提升推理效率。
- 推荐动作：推荐工程师精读 can_fuse_shared_expert 条件判断和权重映射逻辑，理解 AMD 特定优化路径；关注 FP8 兼容性为待办事项，可参考讨论中的技术权衡。

## 功能与动机

PR body 指出：Qwen2 MoE 和 Qwen3.5 MoE 模型使用共享专家，当 shared_expert_intermediate_size == moe_intermediate_size 时，可将共享专家与路由专家融合（topk+1），从而减少内核启动次数并提升推理效率。

## 实现拆解

1. 环境检测与条件判断：在 python/sglang/srt/models/qwen2_moe.py 中添加 can_fuse_shared_expert 函数，检查服务器参数（--disable-shared-experts-fusion）、配置属性（shared_expert_intermediate_size）和后端兼容性（DeepEP），确保融合仅在 HIP 平台且 SGLANG_USE_AITER=1 时启用。
2. 专家计数与初始化扩展：Qwen2MoeSparseMoeBlock 初始化时新增 support_shared_expert_fusion 参数，计算 num_fused_shared_experts，并更新 top_k 和 num_experts 以包含共享专家，影响后续 MoE 调度。
3. 路由逻辑增强：_forward_router_experts 方法中，在 top-k 选择后调用 _append_shared_to_topk_output 将共享专家 ID 和权重追加到输出，实现单次调度中的融合前向。
4. 权重加载适配：在 python/sglang/srt/models/qwen3_5.py 中，Qwen3_5MoeForConditionalGeneration.load_weights 方法使用 _get_num_fused_shared_experts 获取融合专家数，调整 expert_params_mapping 和 fused_expert_params_mapping，将共享专家权重（如 mlp.shared_expert.*）重映射到路由专家布局。
5. 测试与配置配套：PR body 包含准确性测试（GSM8K）和基准测试结果，但未直接修改测试文件；作者提及 FP8 精度问题需 aiter 升级，当前仅限 BF16，后续 PR 将补全。

关键文件：

- python/sglang/srt/models/qwen2_moe.py（模块 模型层；类别 source；类型 core-logic；符号 can_fuse_shared_expert, _get_shared_expert_weights, _append_shared_to_topk_output）：核心实现文件，新增共享专家融合判断函数并扩展MoE 块逻辑，直接影响路由调度和专家计数。
- python/sglang/srt/models/qwen3_5.py（模块 模型层；类别 source；类型 data-contract；符号 _get_num_fused_shared_experts）：权重加载适配文件，添加方法获取融合专家数量并更新映射逻辑，确保模型正确加载融合共享专家权重。

关键符号：can_fuse_shared_expert, _get_shared_expert_weights, _append_shared_to_topk_output, _get_num_fused_shared_experts

## 关键源码片段

### python/sglang/srt/models/qwen2_moe.py

核心实现文件，新增共享专家融合判断函数并扩展MoE块逻辑，直接影响路由调度和专家计数。

```
def can_fuse_shared_expert(
    config: PretrainedConfig,
) -> bool:
    """共享专家是否可作为额外MoE专家融合（Qwen3.5 + Aiter）。

    调用方仍需基于`support_shared_expert_fusion`和`_use_aiter`门控。
    """
    if (
        get_global_server_args().disable_shared_experts_fusion is True # 服务器参数禁用融合
        or getattr(config, "shared_expert_intermediate_size", 0) <= 0 # 配置未定义共享专家尺寸
        or config.shared_expert_intermediate_size != config.moe_intermediate_size # 尺寸不匹配
        or get_moe_a2a_backend().is_deepep() # 后端为DeepEP时不兼容
    ):
        return False
    return True


def _append_shared_to_topk_output(
    self,
    topk_output: StandardTopKOutput,
    shared_expert_weights: torch.Tensor,
) -> StandardTopKOutput:
    """将共享专家追加到top-k输出，用于融合调度。

    共享专家ID设为`self.num_experts`（即基础专家数），权重来自sigmoid(gate)。
    """
    shared_expert_ids = torch.full_like(
        topk_output.indices[:, :1],
        self.num_experts,
        device=topk_output.indices.device,
    )
    # 扩展indices和weights以包含共享专家
```

```
    new_indices = torch.cat([topk_output.indices, shared_expert_ids], dim=-1)
    new_weights = torch.cat([topk_output.weights, shared_expert_weights], dim=-1)
    return StandardTopKOutput(new_indices, new_weights)
```

## python/sglang/srt/models/qwen3_5.py

权重加载适配文件，添加方法获取融合专家数量并更新映射逻辑，确保模型正确加载融合共享专家权重。

```
  def _get_num_fused_shared_experts(self):
    """获取融合共享专家的数量，用于权重加载时调整专家计数。

    通过检查首层MLP的`num_fused_shared_experts`属性实现；若未启用则返回0。
    """
    if not (
        hasattr(self.model, "layers")
        and len(self.model.layers) > 0
        and hasattr(self.model.layers[0].mlp, "num_fused_shared_experts")
    ):
        return 0
    return self.model.layers[0].mlp.num_fused_shared_experts


  def load_weights(self, weights: Iterable[Tuple[str, torch.Tensor]]):
    """加载权重，适配融合共享专家布局。

    当启用融合时，`num_experts`增加`num_fused_shared_
    experts`，并将共享专家权重映射到路由专家索引。
    """
    num_experts_base = self.config.num_experts
    num_fused_shared_experts = self._get_num_fused_shared_experts()
    num_experts = num_experts_base + num_fused_shared_experts # 调整总专家数

    # 构建专家参数映射，包含基础专家和融合共享专家
    expert_params_mapping = FusedMoE.make_expert_params_mapping(
        ckpt_gate_proj_name="gate_proj",
        ckpt_down_proj_name="down_proj",
        ckpt_up_proj_name="up_proj",
        num_experts=num_experts, # 使用调整后的专家数
    )

    # 处理共享专家权重重映射
    if self.enable_shared_expert_fusion:
        for name, loaded_weight in weights:
            if f"mlp.shared_expert.gate_up_proj" in name:
                # 将共享专家gate_up_proj拆分为gate和up，加载到对应专家索引
                loaded_weight = loaded_weight.chunk(2, dim=-2)
                weight_loader(param, loaded_weight[0], name_mapped, "w1", num_experts_base)
                weight_loader(param, loaded_weight[1], name_mapped, "w3", num_experts_base)
            elif f"mlp.shared_expert.down_proj" in name:
```

weight_loader(param, loaded_weight, name_mapped, "w2", num_experts_base)

## 评论区精华

- 变量名一致性：yichiche 评论“Is there a specific reason for changing the name from router_logits to gate_logits? If not, we should keep it as is to avoid unnecessary changes.”，强调保持代码一致性以避免混淆。
- 日志清理：yichiche 多次指出“We should remove this logging code from the production environment.”，要求移除调试日志以提升代码整洁性。
- DeepEP 兼容性：yichiche 建议“Add or get_moe_a2a_backend().is_deepep() to check if deepep is enable.”，防止融合逻辑与 DeepEP 后端冲突，已整合到 can_fuse_shared_expert 中。
- 权重映射细节：yichiche 询问“You may also need fused_expert_params_mapping here?”，zhentaocc 解释共享专家通过现有逻辑映射，无需额外条目。
- FP8 问题与后续工作：zhentaocc 提到“FP8 accuracy issue identified, will need aiter upgrade to fix split_k issue.”，指出当前 PR 限 BF16，FP8 支持需后续 PR 解决，并计划扩展共享 gate 融合。
- 变量名一致性与日志清理 (style): 作者可能已调整变量名，日志代码在后续提交中被移除，强调代码整洁性和一致性。
- DeepEP 后端兼容性检查 (correctness): 检查已整合到 can_fuse_shared_expert 函数中，确保在 DeepEP 启用时禁用融合。
- 权重映射逻辑适配 (design): 权重加载已适配，共享专家权重被重映射到路由专家布局，设计决策基于现有架构扩展。
- FP8 精度问题与后续工作 (correctness): FP8 支持被识别为待办事项，作者计划后续 PR 解决，不影响当前 BF16 功能。

## 风险与影响

- 风险：- 回归风险：融合条件依赖 SGLANG_USE_AITER 环境变量和 HIP 后端，若配置不当或平台不支持，可能导致 MoE 层行为异常或性能退化；权重加载改动涉及共享专家重映射，错误处理可能引发模型加载失败。
- 性能风险：融合逻辑增加初始化复杂度，但在启用时减少内核启动，实测吞吐量提升；未启用时对性能无影响。
- 兼容性风险：对 FP8/MXFP4 量化支持不完整，作者提及精度问题，需后续 aiter 升级，可能影响量化模型用户。
- 安全风险：无直接安全漏洞，但新代码分支需验证边界条件，防止配置错误导致服务中断。
- 影响：- 用户影响：AMD ROCm 用户通过设置 SGLANG_USE_AITER=1 可自动启用融合，获得约 5% 吞吐量提升（基准测试显示 BF16 下输出 token 吞吐量从 767.82 tok/s 增至 802.79 tok/s），且准确性测试显示精度变化可忽略（GSM8K exact_match 差异 <0.003）。
- 系统影响：优化 MoE 层调度，减少 GPU 内核启动次数，提升整体推理效率；权重加载逻辑扩展支持融合布局，增强模型兼容性。

- 团队影响：需跟进 FP8 支持和共享 gate 融合等后续工作，可能推动 AMD 平台优化和 MoE 模块演进。
- 风险标记：条件依赖环境变量，FP8 支持不完整，权重加载复杂化

## 关联脉络

- PR #21773 [AMD][CI] Add GLM-5-MXFP4 accuracy and perf nightly tests for MI35x: 同为 AMD 平台相关 PR，涉及 CI 测试扩展，可辅助理解 AMD 环境验证背景。
- PR #22606 [serving] replace O(n²) stream_buffer string concat with integer offset: 同为性能优化 PR，共享减少计算开销的目标，可对比学习性能改进策略。
- PR #21232 [sgl] perf optimization for eplb: 涉及算法性能优化，与本 PR 的 MoE 层效率提升主题相关，反映团队持续性能优化趋势。