PR #27057 完整报告

sgl-project/sglang

[AMD] move shared expert check function to quark

合并时间：2026-06-13 11:46

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27057

执行摘要

- 一句话：将共享专家融合检查移至 Quark 配置，并添加精确判断
- 推荐动作：该 PR 展示了将量化后端特殊逻辑从模型代码中分离的实践，值得关注。虽然改动较小，但 can_fuse_shared_expert 的深度比较设计为可复用模式。建议在类似涉及量化配置判断的场景中参考此方式。

功能与动机

For shared expert fusion feature, precise checks are added. This is aimed for a more general use for all models.

实现拆解

1. Quark 配置新增 can_fuse_shared_expert：在 quark.py 的 QuarkConfig 类中添加 can_fuse_shared_expert 方法。首先检查排除层中是否包含共享专家（gate 除外），若包含则返回 False；然后检查是否有逐层量化配置，若无则返回 True；最后通过 _find_matched_config 和 deep_compare 比较路由专家与共享专家的量化配置是否一致，不一致则返回 False。
2. Qwen3.5 模型添加自动禁用：在 qwen3_5.py 中添加 _maybe_autodisable_shared_experts_fusion 方法，在模型初始化时于 ROCm 环境下调用。如果模型类型为 qwen3_5_moe_text、融合未被手动禁用且 can_fuse_shared_expert 返回 False，则自动设置 disable_shared_experts_fusion 为 True，使多流路径（#25885）仍能生效。
3. Qwen2MoE 函数委托：在 qwen2_moe.py 的 can_fuse_shared_expert 函数中，移除原有的直接排除层检查，改为通过 getattr 获取 quant_config 的 can_fuse_shared_expert 方法并调用。若方法不存在则保持 True，确保非 Quark 后端行为不变。

关键文件：

- python/sglang/srt/layers/quantization/quark/quark.py（模块 量化层；类别 source；类型 core-logic；符号 can_fuse_shared_expert）：新增 can_fuse_shared_expert 方法，通过排除层检查和配置深度比较精确判断共享专家融合是否可行。
- python/sglang/srt/models/qwen3_5.py（模块 模型层；类别 source；类型 data-contract；符号 _maybe_autodisable_shared_experts_fusion）：新增 _maybe_autodisable_shared_experts_fusion 方法，在 ROCm 上自动禁用共享专家融合，

确保多流路径可用。

- python/sglang/srt/models/qwen2_moe.py（模块 模型层；类别 source；类型 data-contract）：修改 can_fuse_shared_expert 函数，委托给 quant_config 的 can_fuse_shared_expert 方法，保留向后兼容。

关键符号：can_fuse_shared_expert, _maybe_autodisable_shared_experts_fusion

关键源码片段

python/sglang/srt/layers/quantization/quark/quark.py

新增 can_fuse_shared_expert 方法，通过排除层检查和配置深度比较精确判断共享专家融合是否可行。

```
def can_fuse_shared_expert(self) -> bool:
    # 检查排除层：如果共享专家 body 被排除（gate 除外），则不可融合
    if any(
        "shared_expert" in layer
        and "shared_expert_gate" not in layer
        and not layer.startswith("mtp.")
        for layer in self.exclude_layers
    ):
        return False

    # 如果没有逐层量化配置，说明所有层使用统一规格，可以融合
    layer_quant_config = self.quant_config.get("layer_quant_config") or {}
    if not layer_quant_config:
        return True

    # 对比 layer 0 的路由专家与共享专家各投影的量化配置
    lookup_stub = torch.nn.Module()
    try:
        for base in _MOE_SHARED_EXPERT_QUANT_LAYER0_BASES:
            moe_name = f"{base}.mlp.experts"
            moe_cfg = self._find_matched_config(moe_name, lookup_stub)
            for suffix in _SHARED_EXPERT_BODY_PROJ_SUFFIXES:
                shared_name = f"{base}.mlp.shared_expert.{suffix}"
                shared_cfg = self._find_matched_config(shared_name, lookup_stub)
                if not deep_compare(moe_cfg, shared_cfg):
                    return False
    except ValueError:
        # 配置名不匹配视为不可融合
        return False
    return True
```

python/sglang/srt/models/qwen3_5.py

新增 _maybe_autodisable_shared_experts_fusion 方法，在 ROCm 上自动禁用共享专家融合，确保多流路径可用。

```
def _maybe_autodisable_shared_experts_fusion(self, config, quant_config):
```

```
    # 当检查点无法融合时自动禁用（例如 MXFP4 Qwen3.5），
    # 这样模型仍能走多流路径（#25885）。仅适用于 ROCm。
    server_args = get_global_server_args()
    if (
        config.model_type == "qwen3_5_moe_text"
        and not server_args.disable_shared_experts_fusion
        and not can_fuse_shared_expert(config, quant_config)
    ):
        server_args.disable_shared_experts_fusion = True
        logger.info(
            "Qwen3.5: shared-expert fusion not supported for this checkpoint; "
            "auto-disabling (multi-streaming #25885 still applies)."
        )

# 在 __init__ 中调用
if _is_hip:
    self._maybe_autodisable_shared_experts_fusion(config, quant_config)
```

评论区精华

主要讨论围绕 gemini-code-assist[bot] 的担忧：将原始 exclude_layers 检查替换为 can_fuse_shared_expert 方法后，其他量化后端（未实现该方法）是否会绕过安全检查。作者解释新逻辑只对 Quark 生效，因为通过 getattr 检查方法是否存在；若不存在，则直接跳过，行为与之前相同。最终 reviewer 接受此解释，PR 合并。

- 非 Quark 后端的原始 exclude_layers 检查是否会被绕过 (correctness): 作者的解释被接受，逻辑正确。PR 合并，其他后端无影响。

风险与影响

- 风险：
  - 缺少测试覆盖：PR 未添加针对新检查逻辑的单元测试，但原有逻辑未受影响，且改动较小。
  - 非 Quark 后端行为：通过 getattr 委托确保向后兼容，非 Quark 后端无行为变化。
  - 平台依赖性：自动禁用仅作用于 ROCm 平台（_is_hip），其他平台行为不变。
- 影响：
  - 用户：对用户透明，AMD ROCm 用户在使用 Qwen3.5 MXFP4 模型时可能观察到多流性能提升（因融合被自动禁用）。
  - 系统：仅影响模型初始化时的配置判断，对运行时性能无直接影响。
  - 团队：将量化专用逻辑从模型代码中分离，便于后续维护和扩展。
  - 风险标记：缺少测试覆盖

关联脉络

- PR #22948 Original shared expert exclusion check: 原始排除层检查的参考来源

- PR #25885 Multi-streaming for transformer flow: 该 PR 支持在共享专家融合开启时仍启用多流，本 PR 进一步确保自动禁用融合以激活多流