PR #51695 完整报告

vllm-project/vllm

[MOE] Standardize and abstract fused shared expert optimization selection

合并时间：2026-08-19 00:56

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51695

执行摘要

- 一句话：标准化 FSE 检测，模型构建与权重加载统一量化兼容判定
- 推荐动作：值得精读。核心价值在于治理“双分辨率 bug 源”：把 FSE 判定收敛为单一来源（建模期 resolve_layer_fused_shared_expert + 模型级 is_model_fused_shared_expert_compatible），并让权重加载复用同一状态。值得关注的设计决策包括：量化兼容性检查只在用户显式请求 FSE 时执行（避免无谓开销）、AITER 不可用时归一化为 False、混合状态直接显式报错而非静默回退、Quark 层配置查找的精确 / 通配双模式。

功能与动机

PR body 指出 FSE 判定逻辑在多个模型实现中重复且口径不一：例如 qwen3_next.py 有独立的 _is_shared_expert_fse_compatible，DeepSeek-V4 AMD 路径有 _shared_experts_are_fp4，而 fused_moe/layer.py 的 determine_expert_counts 又用另一套条件二次解析，容易出 bug。作者希望模型构建和 checkpoint 加载使用同一套“量化兼容”决策，并让 FSE API 不绑定 ROCm/AITER，为 online quantization（#51285/#51392）和 FlashInfer 侧 shared expert fusion（issue #50963）铺路。

实现拆解

1. 新增统一兼容性判定入口：vllm/model_executor/layers/quantization/utils/config_utils.py（新增 155 行）定义 is_shared_expert_quant_fse_compatible(quant_config, expert_prefix, shared_expert_prefix, projection_names)。对 DeepseekV4FP8Config 校验路由专家是否为 MXFP4、shared expert 是否被 exclude、w1 权重 dtype 是否为 fp4，并处理 MTP 层的前缀规整；对 QuarkConfig 通过 get_projection_quant_configs 展开 packed projection，对比 routed 与 shared expert 各投影的量化配置是否完全一致；其余量化方法暂返回“未实现兼容性检测”。
2. 重构 Quark 层配置查找：vllm/model_executor/layers/quantization/quark/quark.py 将原先内嵌的 _matches_pattern 逻辑外提为公开方法 get_layer_quant_config_from_name，支持精确匹配与 fnmatch 通配符；_find_matched_config 保留 layer_type/global 回退语义并复用新方法，消除 packed 展开处的重复代码。
3. 新增 FSE 解析与模型级校验：vllm/model_executor/layers/fused_moe/utils.py 新增 resolve_layer_fused_shared_expert（仅在 AITER 开关开启时做兼容性检查，AITER 不可用时归一化为 False）和 is_model_fused_shared_expert_compatible（遍历模型所有

MoE 层，混合启用 / 禁用状态直接抛 NotImplementedError，报告启用与禁用层数）。

4. 模型接入：DeepSeek-V4（AMD）、MiniMax-M3（AMD）、Qwen3-Next、Qwen3.5、DeepSeek-V2、GLM4-MoE、AXK1 及多个 MTP 模块改为在 MoE 构造时调用 resolve_layer_fused_shared_expert 并保存 is_fused_shared_expert_enabled，load_weights/get_expert_mapping/maybe_fuse_shared_experts 等权重路径统一读取该状态；原 _shared_experts_are_fp4、_fuse_shared_experts_enabled(config, prefix) 等重复实现被删除。fused_moe/layer.py 的 determine_expert_counts 也改为直接接收建模期解析出的 fuse_shared_experts 参数，消除双重解析。
5. 测试配套：新增 tests/model_executor/layers/test_fused_shared_expert.py（700 行），覆盖 DeepSeek-V4/Quark 的 exclude 正则、MiniMax-M3/DeepSeek-V4/Qwen3.5/DeepSeek-V2/GLM4 五类模型的实例化、resolve_layer_fused_shared_expert 三种分支（禁用跳过兼容性检查、AITER 不可用归一化、拒绝不兼容量化），并验证 determine_expert_counts 的 fuse_shared_experts 参数覆盖。MI350 与 H100 上均为 29 passed。

关键文件：

- vllm/model_executor/layers/quantization/utils/config_utils.py（模块 量化配置；类别 source；类型 data-contract；符号 is_shared_expert_quant_fse_compatible, get_projection_quant_configs）：新增统一 FSE 量化兼容性判定入口，是本次标准化的核心数据契约；对 DeepseekV4FP8Config 与 QuarkConfig 分别实现兼容性检测，返回布尔值与原因。
- vllm/model_executor/layers/fused_moe/utils.py（模块 融合 MoE；类别 source；类型 data-contract；符号 resolve_layer_fused_shared_expert, is_model_fused_shared_expert_compatible, get_moe_layer）：新增 resolve_layer_fused_shared_expert 与 is_model_fused_shared_expert_compatible，形成 FSE 解析与模型级一致性校验的统一入口，所有模型接入的核心载体。
- vllm/model_executor/layers/quantization/quark/quark.py（模块 量化配置；类别 source；类型 data-contract；符号 _find_matched_config, get_layer_quant_config_from_name, _matches_pattern）：重构 Quark 层配置查找，新增公开的 get_layer_quant_config_from_name，供 FSE 兼容性判定按层名精确 / 通配匹配配置，是 config_utils 判定能力的地基。
- vllm/models/deepseek_v4/amd/model.py（模块 模型定义；类别 source；类型 data-contract；符号 _shared_experts_are_fp4, _fuse_shared_experts_enabled）：DeepSeek-V4 AMD 路径从旧的 _shared_experts_are_fp4 层内精度检查迁移到统一兼容性判定，并让权重映射 / 加载读取模型级 is_fused_shared_expert_enabled，是行为变更最明显的模型接入点。
- vllm/models/minimax_m3/amd/model.py（模块 模型定义；类别 source；类型 data-contract；符号 _fuse_shared_experts_enabled, _aiter_moe_fused_shared_experts_enabled）：MiniMax-M3 AMD 路径重构 FSE 解析：_aiter_moe_fused_shared_experts_enabled 改为接收解析后的布尔值，消除对 config 的重复判断，并保留 gfx950 + AITER 的专属子路径。

- tests/model_executor/layers/test_fused_shared_expert.py（模块 FSE 测试；类别 test；类型 test-coverage；符号 get_deepseek_v4_quark_config, get_fse_test_model_config, test_determine_expert_counts_fuse_shared_experts_override, test_resolve_layer_fused_shared_expert_skips_compatibility_when_disabled）：新增 700 行测试，是本次重构的质量保障核心：覆盖 Quark/DeepSeek-V4 配置构造、五类模型实例化、resolve 三分支与 determine_expert_counts 覆盖，双平台验证。

关键符号：is_shared_expert_quant_fse_compatible, get_projection_quant_configs, get_layer_quant_config_from_name, _find_matched_config, resolve_layer_fused_shared_expert, is_model_fused_shared_expert_compatible, get_moe_layer, _aiter_moe_fused_shared_experts_enabled

关键源码片段

vllm/model_executor/layers/fused_moe/utils.py

新增 resolve_layer_fused_shared_expert 与 is_model_fused_shared_expert_compatible，形成 FSE 解析与模型级一致性校验的统一入口，所有模型接入的核心载体。

```
# vllm/model_executor/layers/fused_moe/utils.py
def resolve_layer_fused_shared_expert(
    quant_config: "QuantizationConfig | None",
    prefix: str,
    shared_expert_name: str = "shared_experts",
) -> bool:
    """Resolve whether AITER fused shared-expert execution is enabled.

    Returns:
        Whether AITER fused shared experts are enabled.
    """
    # 注意：`is_fusion_moe_shared_experts_enabled` 被 `@if_aiter_supported`
    # 装饰，AITER 不可用时返回 None
    fse_requested = bool(rocm_aiter_ops.is_fusion_moe_shared_experts_enabled())
    # 仅在用户显式请求 FSE 时才做量化兼容性检查；
    # 未请求时跳过检查，避免无谓告警与开销
    fse_compatible, fse_reason = (
        is_shared_expert_quant_fse_compatible(
            quant_config,
            f"{prefix}.experts",
            f"{prefix}.{shared_expert_name}",
        )
        if fse_requested
        else (True, None)
    )
    is_fused_shared_expert_enabled = fse_requested and fse_compatible
    if fse_requested and not is_fused_shared_expert_enabled:
        logger.warning(
            "VLLM_ROCM_USE_AITER_FUSION_SHARED_EXPERTS is enabled but "
            "cannot be enabled: %s.",
```

```
            fse_reason,
        )
    return is_fused_shared_expert_enabled


def is_model_fused_shared_expert_compatible(
    layers: nn.ModuleList | Iterable[nn.Module],
    moe_cls: type[nn.Module],
    moe_name: str,
) -> bool:
    """Resolve one fused-shared-expert state for a model's MoE layers."""

    def get_moe_layer(layer: nn.Module) -> nn.Module | None:
        # 沿 `moe_name`（如 "mlp"/"block_sparse_moe"）逐级取子模块
        for name in moe_name.split("."):
            layer = getattr(layer, name, None)
            if layer is None:
                return None
        return layer

    moe_layers = (
        moe_layer
        for layer in layers
        # 跳过流水线并行占位层，并只统计实际存在的 MoE 层
        if not isinstance(layer, PPMissingLayer)
        and (moe_layer := get_moe_layer(layer)) is not None
        and isinstance(moe_layer, moe_cls)
    )

    enabled = [
        getattr(layer, "is_fused_shared_expert_enabled", False) for layer in moe_layers
    ]
    enabled_count = sum(enabled)
    disabled_count = len(enabled) - enabled_count
    # 模型内所有 MoE 层必须保持一致 FSE 状态；
    # 混合状态直接报错，避免同一 PP rank 内出现无法对齐的权重加载路径
    if enabled_count > 0 and disabled_count > 0:
        raise NotImplementedError(
            "Fused shared experts must be enabled for all MoE layers; found "
            f"{enabled_count} enabled and {disabled_count} disabled layers. "
            "Per-layer fused shared experts is not yet supported. Please open "
            "an issue."
        )
    return enabled_count > 0 and disabled_count == 0
```

评论区精华

1. mgoin 指出 MTP 前缀映射错误：model.layers.<N>.ffn.shared_experts 在 MTP 层应规整为 mtp.<N-num_hidden_layers>.ffn.shared_experts，而不是直接截断为 layers.<N>...

；作者在 a56fa44 修复。

2. mgoin 质疑 layer_quant_config 缺失："Not checking layer_quant_config seems pretty bad"，建议至少检查 layer_quant_config 是否只有一个条目作为最简保护；作者随后补上了逐层精确加正则匹配的完整查找。
3. mgoin 提醒 "shared_expert" 字符串检查不贴合 Quark 语义："Just blindly checking for "shared_expert" doesn't seem like it matches Quark's semantics……保守起见可以接受，但应该做一个更好的基于实际量化前缀匹配的公共工具"；作者同意并改进了 exclude 判定。
4. bnellnm 追问 all(enabled)：注释声称"必须所有层启用"，返回逻辑却只是有启用层即可；作者将返回条件改为 enabled_count > 0 and disabled_count == 0，与注释一致。
5. fxmarty-amd 自述双重解析问题：fused_moe/layer.py 的 determine_expert_counts 与建模文件各自解析 FSE，"双分辨率容易出 bug"，并指出 is_act_and_mul 条件源自 #32244、可疑的 or 条件源自 #46545；最终在 1707ae5 让 FusedMoEFactory -> determine_expert_counts 直接依赖传入的 fuse_shared_experts。

- MTP 层 shared_expert 前缀映射错误 (correctness): 作者在 a56fa4400ce17b5d64823807b2571166ca987fe2 修复，按 extract_layer_index 与 num_hidden_layers 区分主模型与 MTP 层。
- layer_quant_config 未被检查 (correctness): 作者补上逐层精确加 re: 正则匹配的完整查找，并回退到 global_quant_config。
- shared_expert 字符串检查不符合 Quark 语义 (design): 作者保留保守检查并在代码注释中标注 TODO；这是从旧 qwen3_next.py 逻辑平移的历史行为。
- is_model_fused_shared_expert_compatible 应为 all(enabled) (correctness): 作者改为 enabled_count > 0 and disabled_count == 0，与注释语义一致。
- FusedMoEFactory 双重 FSE 分辨率 (design): 提交 1707ae5 让 FusedMoEFactory -> determine_expert_counts 直接依赖传入的 fuse_shared_experts 单一解析结果。
- layer_type 检查是否可移入 resolve 函数 (design): 作者在 9f045cb5f9278e19f2700b8bfdcd1dcfd6ac2c62 处理，将检查收敛到统一 resolve 逻辑。

风险与影响

- 风险：
  1. 行为变更：模型内混合 FSE 状态现在抛 NotImplementedError（原为静默错乱或按层各自回退）；VLLM_ROCM_USE_AITER_FUSION_SHARED_EXPERTS=1 但量化不兼容时从部分模型可运行变为统一告警回退。PP>1 场景下主模型与 MTP 模型 FSE 策略可能不同，作者特意将 ValueErrors 放宽为 logger.warning 以兼容。
1. Quark 查找语义重构风险：get_layer_quant_config_from_name 改变了 packed projection 的 fallback 顺序（原先 packed 外层直接用 global_config，现在先逐 shard 查 layer_quant_config），可能影响既有非 FSE 量化模型的配置解析，需关注 quantization 相关 CI 测试。
2. 权重加载路径：deepseek_v4/amd/model.py 删除层内 _shared_experts_are_fp4 精度检查后，fuse_by_layer/get_expert_mapping 统一读取模型级状态，对层间共享专家精度不

一致的 checkpoint 行为有变化。

3. 跨模型影响面广：DeepSeek-V2/V4、MiniMax-M3、Qwen3-Next/Qwen3.5、GLM4-MoE、AXK1 及多个 MTP 模块的构建与权重加载路径均被改动，回归面大；好在 MI350 与 H100 双平台测试通过。 - 影响：影响所有实现 FSE 的 MoE 模型（DeepSeek-V2/V4、MiniMax-M3、Qwen3-Next/Qwen3.5、GLM4-MoE、AXK1 等）的模型构建、权重映射与加载路径；新增 700 行测试覆盖 29 个用例，在 MI350 与 H100 双平台验证；为后续 online quantization（#51285/#51392）和 FlashInfer 侧 FSE（#50963）提供了统一的兼容性检测扩展点。团队协作上，由 AMD 侧作者主导、vLLM 维护者（mgoin、bnellnm）深度 review 并 approve，属于跨厂商共建的 MoE 基础设施改动。 - 风险标记：行为变更：混合 FSE 状态抛 NotImplementedError，Quark 配置查找语义重构，跨 10+ 模型权重加载路径，PP>1 主模型与 MTP 策略差异

关联脉络

- PR #51285 Online quantization (PR body 中提及): PR body 明确说明本 PR 的 FSE 兼容性检测是 online quantization 的前置，使 FSE 能基于任意量化配置做更广的兼容性判定。
- PR #51392 Online quantization (PR body 中提及): 与 #51285 同属 online quantization 功能线，PR body 明确引用为本 PR 的后续受益方。
- PR #32244 fused_moe/layer.py 中 is_act_and_mul 条件来源：fxmarty-amd 在 review 中明确指出 determine_expert_counts 中 is_act_and_mul 条件的历史来源，与本 PR 消除双重解析直接相关。
- PR #46545 fused_moe/layer.py 中可疑 or 条件来源：fxmarty-amd 在 review 中指出该 PR 引入的 or 条件导致 mini-max-m3 可在非 AITER 下启用 FSE，本 PR 将其收敛为单一 fuse_shared_experts 传递。
- PR #52182 Remove VLLM_TEST_FORCE_FP8_MARLIN to replace with linear_backend/moe_backend: 同属 MoE/ 量化后端选择逻辑的抽象与标准化演进，与本 PR 的 FSE 后端选择统一方向一致。