PR #49987 完整报告

vllm-project/vllm

Fix MQA with tensor parallelism on transformers modeling backend

合并时间：2026-07-28 06:51

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49987

执行摘要

- 一句话：修复 MQA 在 TP>1 时 split 错误，新增 PackedQKVFuser
- 推荐动作：建议阅读本 PR 以了解 Transformers 后端的 fuser 设计模式，特别是 PackedQKVFuser.match 和 update_forward 中的 AST 重写技巧，以及基类重构的分层思路。对于维护或扩展现有 StackedFuser 子类的开发者，需关注 RewriteFuser 的 fuse 方法行为是否与预期一致。

功能与动机

在 PR #30966 将 GPTBigCode/Starcoder2 迁移到 Transformers 建模后端后，Multi-Query Attention (MQA) 在 tensor parallelism > 1 时运行失败。根本原因是 MQA 使用单个 packed 投影 c_attn 输出 q + 2*kv 维度的向量，并在前向中通过 split((q_size, kv_size, kv_size), dim=-1) 分解。在张量并行下，每个 rank 只拥有部分 q/kv 头，但 split 仍使用全局 head 宽度，导致形状错误。本 PR 通过新增 PackedQKVFuser 自动检测并重写 split 尺寸为 [s // tp_size for s in output_sizes]，从而修复 TP 兼容性。

实现拆解

主要分为以下几个步骤：

1. 新增 PackedQKVFuser 类（vllm/model_executor/models/transformers/fusers/packed_qkv.py）：
   - 继承自新抽取的 RewriteFuser 基类，专用于处理“packed QKV”模式：即 HF 模型中单个 c_attn 线性投影输出后通过 split((q, kv, kv)) 分解为 Q、K、V 的场景（典型如 GPTBigCode/Starcoder2 的 MQA）。
   - match 方法遍历 FX 图，找到匹配 split((q_size, kv_size, kv_size)) 的节点，并回溯到上游线性层，验证尺寸一致且输出投影匹配，从而确定需要融合的层。
   - _split_call 方法在 AST 级别定位该 split 调用，确保唯一性。
   - update_forward 方法通过 AST 重写，将 split 的尺寸部分替换为 local_output_sizes(self.merged_name) 生成的动态分片表达式 [s // self.qkv_proj.tp_size for s in self.qkv_proj.output_sizes]，从而在张量并行下每个 rank 使用正确的局部宽度。
   - validate 方法检查 head 尺寸兼容性，确保融合可行。
2. 重构 fuser 基类（vllm/model_executor/models/transformers/fusers/base.py）：

- 将原来的 StackedFuser 拆分为 RewriteFuser 和 StackedFuser 两层：RewriteFuser 提供通用的 forward 重写和编译流程（update_forward、update_attrs、fuse），StackedFuser 在其基础上增加堆叠投影的语义（shards、orig_to_new_stacked 等）。
- 新增 local_output_sizes 辅助函数，返回用于 AST 替换的字符串表达式，抽取了之前 QKVFuser 中硬编码的内联代码，便于复用。

3. 调整 upstream_linear（vllm/model_executor/models/transformers/fx_utils.py）：
   - 允许穿过非线性的子模块调用（如 GPT 风格 attention 中的 resid_dropout），使得在存在 dropout 等模块时仍能正确回溯到输出投影。
4. 注册新 Fuser（vllm/model_executor/models/transformers/fusers/__init__.py 和 fuser.py）：
   - 在 __init__.py 中导出 PackedQKVFuser，并在 fuser.py 中调整控制流使其参与自动融合流程。
5. 添加单元测试（tests/models/transformers/fusers/test_linear.py）：
   - 新增 PackedQKVAttention 模拟 GPTBigCode 风格的 attention，包含 c_attn 和 c_proj。
   - 新增 ResidDropoutAttention 验证在 dropout 存在时输出投影仍能被正确识别。
   - 新增 FakeMQASelfAttn（基于 FakeAttention）模拟 MQA 场景。
   - _apply_packed_qkv_fuser_with_stubs 辅助函数使用普通 nn.Linear 模拟融合后的投影，并绑定重写后的 forward。
   - test_detects_and_rewrites_packed_qkv 测试函数验证匹配、重写和数值正确性。

关键文件：

- vllm/model_executor/models/transformers/fusers/packed_qkv.py（模块 后端融合器；类别 source；类型 core-logic；符号 PackedQKVFuser, info, _packed_sizes, match）：新增 PackedQKVFuser 类，核心变更，实现 MQA packed QKV 的自动检测与 TP 感知 split 重写。
- vllm/model_executor/models/transformers/fusers/base.py（模块 融合器基类；类别 source；类型 refactor；符号 local_output_sizes, StackedFuser, RewriteFuser, update_forward）：重构 fuser 基类，抽取 RewriteFuser 层，新增 local_output_sizes 辅助函数，为所有 fuser 提供统一的前向重写基础。
- tests/models/transformers/fusers/test_linear.py（模块 测试；类别 test；类型 test-coverage；符号 ResidDropoutAttention, init, forward, PackedQKVAttention）：添加 PackedQKVAttention 等测试模块和 test_detects_and_rewrites_packed_qkv 测试，确保 fuser 的正确性和回归覆盖。
- vllm/model_executor/models/transformers/fx_utils.py（模块 工具函数；类别 source；类型 data-contract）：调整 upstream_linear 支持穿过非线性子模块，确保带有 dropout 的 GPT 式 attention 中输出投影仍能被发现。
- vllm/model_executor/models/transformers/fusers/__init__.py（模块 注册；类别 source；类型 data-contract）：导出新类 PackedQKVFuser 和 RewriteFuser，使它们可被 auto-fuser 发现。

关键符号：PackedQKVFuser.match, PackedQKVFuser._packed_sizes, PackedQKVFuser._split_call, PackedQKVFuser.update_forward, PackedQKVFuser.validate, local_output_sizes, RewriteFuser.fuse, StackedFuser.update_forward, upstream_linear

关键源码片段

vllm/model_executor/models/transformers/fusers/packed_qkv.py

新增 PackedQKVFuser 类，核心变更，实现 MQA packed QKV 的自动检测与 TP 感知 split 重写。

```
@dataclass
class PackedQKVFuser(RewriteFuser):
    """Fuser for attention with q, k and v packed into one projection."""

    qkv_name: str
    o_name: str | None
    q_size: int
    kv_size: int

    @staticmethod
    def _packed_sizes(node: fx.Node) -> tuple[int, int] | None:
        # 检查是否为 split((q, kv, kv)) 调用且 kv 尺寸相等
        if not is_method(node, "split") or len(node.args) < 2:
            return None
        sizes = node.args[1]
        if not isinstance(sizes, (tuple, list)) or len(sizes) != 3:
            return None
        if not all(isinstance(size, int) for size in sizes):
            return None
        q_size, k_size, v_size = sizes
        if k_size != v_size or q_size < k_size:
            return None
        return q_size, k_size

    @classmethod
    def match(cls, graph: fx.Graph, module: nn.Module) -> "PackedQKVFuser | None":
        # 遍历 FX 图节点，寻找 split 调用匹配 MQA 模式
        for node in graph.nodes:
            if (sizes := cls._packed_sizes(node)) is None:
                continue
            q_size, kv_size = sizes
            # 通过 split 的输入向上回溯到线性投影
            qkv_node = upstream_linear(node.args[0], module)
            if qkv_node is None:
                continue
            qkv_name = str(qkv_node.target)
            # 验证 split 消耗了整个投影输出
```

```
            if module.get_submodule(qkv_name).out_features != q_size + 2 * kv_size:
                continue
            # 尝试找到输出投影（o_proj）
            o_name = returned_linear(graph, module)
            if o_name == qkv_name or (
                o_name is not None
                and module.get_submodule(o_name).in_features != q_size
            ):
                o_name = None
            return cls(
                source_cls=type(module).__name__,
                qkv_name=qkv_name,
                o_name=o_name,
                q_size=q_size,
                kv_size=kv_size,
            )
        return None
```

评论区精华

本 PR 的作者为 microslaw，合并者 hmellor 在合并前进行了代码优化（命名和 DRY 改进，见最终 commit）。无实质技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险包括：
  - 张量并行兼容性：PackedQKVFuser 依赖 local_output_sizes 动态分片，若不正确调用（如未设置 tp_size），可能导致 split 尺寸不匹配。validate 方法只检查 head 尺寸兼容性，未校验 tp_size 是否已正确初始化。
  - 重构影响：将 StackedFuser 基类拆分为 RewriteFuser + StackedFuser，所有原继承 StackedFuser 的类（如 QKVFuser、GLUFuser、MoEBlockFuser）现在继承 RewriteFuser 的 fuse 实现，可能导致行为差异（虽然语义一致，但需要回归测试）。
  - 匹配逻辑的鲁棒性：_packed_sizes 假设 split 的三个尺寸均为整数且 kv 相等，若 HF 模型使用动态表达式或非对称 QKV，会静默跳过融合，导致依然运行在非优化路径。
  - AST 重写错误：_split_call 要求唯一一个三次 split 调用，若 forward 存在多个类似调用的模式会抛异常。
  - 影响：用户影响：修复了 bigcode/starcoder 等 MQA 模型在 --tensor-parallel-size > 1 时运行报错的问题，使用 Transformers 后端的用户可直接受益。系统影响：新增的 fuser 机制对其他 packed QKV 风格模型（如 GPT-2 的双线性？）也有潜力，但当前仅匹配 MQA 模式。团队影响：基类重构为后续新 fuser 的开发提供了更清晰的抽象（RewriteFuser 通用前向重写，StackedFuser 堆叠投影）。测试覆盖了常见变体（含 dropout），增强信心。
  - 风险标记：张量并行兼容性，新增 fuser 可能影响其他模型，重构影响 StackedFuser 子类

关联脉络

- PR #30966 Migrate GPTBigCode and Starcoder2 to the Transformers modeling backend: 这是引入 MQA TP 问题的根因 PR，本 PR 修复了该迁移未处理 MQA 张量并行兼容性的问题。