PR #40708 完整报告

vllm-project/vllm

[BugFix] Fix Gemma4 'layers.0.moe.experts.0.down_proj_packed' KeyError issue

合并时间：2026-05-10 01:20

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40708

执行摘要

- 一句话：修复 Gemma4 AWQ 权重加载 KeyError，支持下划线命名
- 推荐动作：值得阅读。该 PR 清晰展示了如何处理不同量化框架的命名约定差异，并合理利用已有的工具函数 fused_moe_make_expert_params_mapping 减少重复代码。测试添加也体现了回归预防的实践，适合作为类似权重加载问题的参考。

功能与动机

Issue #40247 和 #40591 报告了 Gemma4 26B AWQ 模型在 v0.19.1 版本加载时崩溃，错误信息 KeyError: 'layers.0.moe.experts.0.down_proj_packed'。该问题由 expert_params_mapping 仅支持点号后缀（如 .qweight ），未处理 compressed-tensors 使用的下划线后缀（如 _packed ）导致。

实现拆解

1. 引入 fused_moe_make_expert_params_mapping 工具函数（从 vllm.model_executor.layers.fused_moe 导入），用于生成标准点号后缀的 expert_params_mapping。
2. 基于 dot_suffix_expert_params_mapping 构建 underscore_suffix_expert_params_mapping：将 weight_name 中的 '.' 替换为 '' 并附加 weight 前缀，以匹配 compressed-tensors 的命名规范（如 experts.0.gate_proj. → experts.0.gate_proj_weight_）。
3. 合并两个映射列表作为新的 expert_params_mapping，替代原有的手动映射生成代码。
4. 在 tests/models/quantization/test_awq.py 中新增 test_awq_load 测试函数，参数化测试两个模型：标准 AWQ 格式（dot-suffix）和 compressed-tensors 格式（underscore-suffix），验证加载不抛出异常且能生成输出。
5. （辅助）在 .gitignore 中添加 .codex 忽略。

关键文件：

- vllm/model_executor/models/gemma4.py（模块 模型加载；类别 source；类型 data-contract；符号 load_weights）：核心修复文件，修改了 load_weights 方法中的 expert_params_mapping 生成逻辑，添加下划线后缀命名支持。
- tests/models/quantization/test_awq.py（模块 量化测试；类别 test；类型 test-coverage ；符号 test_awq_load）：添加了 test_awq_load 回归测试，覆盖两种 AWQ 命名约定。

关键符号：load_weights, test_awq_load

关键源码片段

vllm/model_executor/models/gemma4.py

核心修复文件，修改了 load_weights 方法中的 expert_params_mapping 生成逻辑，添加下划线后缀命名支持。

```
# 文件：vllm/model_executor/models/gemma4.py

# 在文件顶部的导入部分，新增了以下导入
from vllm.model_executor.layers.fused_moe import (
    FusedMoE,
    GateLinear,
    fused_moe_make_expert_params_mapping, # 新增
)

# 在 load_weights 方法中，变更后的核心逻辑：
def load_weights(self, weights):
    # ...（之前的 stacked_params_mapping 等）

    num_experts = getattr(self.config, 'num_experts', None) or 0

    # Strategy A: dot-separated suffix（标准 AWQ/GPTQ，如 .qweight, .scales）
    dot_suffix_expert_params_mapping = fused_moe_make_expert_params_mapping(
        self,
        ckpt_gate_proj_name='gate_proj',
        ckpt_down_proj_name='down_proj',
        ckpt_up_proj_name='up_proj',
        num_experts=num_experts,
    )

    # Strategy B: underscore-separated suffix
    # (CompressedTensors 格式，如 _packed, _scale)
    underscore_suffix_expert_params_mapping = [
        (
            f'{param_name}weight_', # 参数名，如 experts.w13_weight_
            f'{weight_name.rstrip(".")}_', # 权重名，如 experts.0.gate_proj_
            expert_id,
            shard_id,
        )
        for (param_name, weight_name, expert_id, shard_id)
        in dot_suffix_expert_params_mapping
    ]

    expert_params_mapping = (
        dot_suffix_expert_params_mapping + underscore_suffix_expert_params_mapping
    )
```

```
    # 后续加载逻辑不变 ...
```

tests/models/quantization/test_awq.py

添加了 test_awq_load 回归测试，覆盖两种 AWQ 命名约定。

```
# 文件：tests/models/quantization/test_awq.py

# 新增的测试函数
@pytest.mark.parametrize(
    ('model', 'quantization', 'dtype'),
    [
        ('mattbucci/gemma-4-26B-AWQ', 'awq', 'float16'),
        ('cyankiwi/gemma-4-26B-A4B-it-AWQ-4bit', 'compressed-tensors', 'bfloat16'),
    ],
    ids=[
        'gemma4-moe-standard-awq-dot-suffix',
        'gemma4-moe-compressed-tensors-underscore-suffix',
    ],
)
@torch.inference_mode()
def test_awq_load(
    vllm_runner: type[VllmRunner],
    example_prompts: list[str],
    model: str,
    quantization: str,
    dtype: str,
) -> None:
    '''Regression test: AWQ weight loading must not KeyError.'''
    with vllm_runner(
        model,
        quantization=quantization,
        dtype=dtype,
        max_model_len=128,
        enforce_eager=True,
    ) as vllm_model:
        outputs = vllm_model.generate_greedy(example_prompts[:2], max_tokens=32)
    assert len(outputs) == 2
```

评论区精华

Review 中有两个主要讨论：

- Isotr0py 建议使用 fused_moe_make_expert_params_mapping 替代手动构造映射，减少冗余代码。作者采纳。
- 作者最初创建了单独测试文件 test_gemma4_weight_loading.py，但 Isotr0py 认为无需独立文件，且现有 test_awq.py 更适合 CI GPU 限制，作者随后将测试合并到 test_awq.py。
- 使用 fused_moe_make_expert_params_mapping 重构专家参数映射 (design): 作者采纳，改用工具函数生成点号后缀映射，并基于此构造下划线后缀映射。

- 测试用例位置选择 (testing): 作者接受，将测试合并到 test_awq.py。

风险与影响

- 风险：主要风险是新引入的下划线后缀映射可能仍无法覆盖所有 compressed-tensors 命名变体（如 weight 之外的 scale 参数）。但通过遍历 dot_suffix_expert_params_mapping 并动态生成，其覆盖面与标准映射一致；测试覆盖了两种主流模型，降低了遗漏风险。对性能无影响，因为修改仅在模型加载阶段执行一次。
- 影响：直接影响使用 Gemma4 系列模型并采用 AWQ/compressed-tensors 量化的用户，修复了此前无法加载模型的 bug。对其他模型和功能无影响。该修复有助于稳定 Gemma4 在 v0.19.1 中的回归问题。
- 风险标记：专家权重命名兼容性

关联脉络

- 暂无明显关联 PR