PR #48876 完整报告

vllm-project/vllm

[Model] Add Inkling compressed-tensors dynamic FP8 support

合并时间：2026-07-29 10:27

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48876

执行摘要

- 一句话：动态 FP8 checkpoint w2 权重尺度加载支持
- 推荐动作：值得精读。该 PR 展示了如何在不改变原有架构的前提下，通过一个简洁的分支处理来适配特殊的权重尺度格式，是增量式扩展的范本。

功能与动机

RedHatAI/Inkling-FP8-dynamic checkpoint 使用了每输出通道的 w2_weight_scale，但原有的加载逻辑将其当作 w2 权重进行分片，导致错误。PR 描述指出需要 'replicate per-output-channel w2_weight_scale values across TP ranks'。

实现拆解

1. 修改 vllm/models/inkling/nvidia/moe.py 的 InklingMoE.load_expert_weight 方法：在 elif key.endswith(('_scale_2', '_global_scale')): 全局尺度分支之后，elif key.startswith('w13'): 之前，插入一个新的 elif key == "w2_weight_scale" and weight.shape[-1] == 1: 分支。该分支直接加载对应专家槽位的权重数据到参数中，不进行分片操作，实现复制。
2. 新增测试 test_moe_loads_channelwise_scale_for_tp：在 tests/models/inkling/test_moe_weight_layout.py 中添加参数化测试，验证 w13 和 w2 两种投影的每输出通道尺度加载。测试模拟了有 3 个全局专家、本地 2 个专家（槽位映射为 0→0, 2→1）、TP rank=1 的场景，并检查 w13 的去交织和 w2 的直接复制逻辑。
3. 删除调试代码：在 PR 的最终版本中，根据 review 评论移除了一个用于调试的环境变量 INKLING_FP8_SCALE_INTERLEAVED。

关键文件：

- vllm/models/inkling/nvidia/moe.py（模块 模型文件；类别 source；类型 data-contract）：核心加载逻辑修改：添加 w2_weight_scale 特殊分支处理每输出通道尺度复制。
- tests/models/inkling/test_moe_weight_layout.py（模块 测试；类别 test；类型 test-coverage；符号 test_moe_loads_channelwise_scale_for_tp）：新增测试覆盖 w13 和 w2 投影的每输出通道尺度加载，确保 TP 下复制逻辑正确。

关键符号：InklingMoE.load_expert_weight, test_moe_loads_channelwise_scale_for_tp

关键源码片段

vllm/models/inkling/nvidia/moe.py

核心加载逻辑修改：添加 w2_weight_scale 特殊分支处理每输出通道尺度复制。

```
# vllm/models/inkling/nvidia/moe.py 中的 load_expert_weight 方法片段

        if key.endswith(('_scale_2', '_global_scale')):
            # 原有逻辑：处理全局尺度（标量）
            ...
        elif key == "w2_weight_scale" and weight.shape[-1] == 1:
            # 新增分支：处理每输出通道尺度（最后一维为 1）
            # 这类尺度的形状为 [global_experts, output_channels, 1]
            # 在 TP 下需要复制到每个 rank，而不是像 w2 权重那样分片
            # 直接根据本地专家槽位索引加载对应的数据行即可
            param.data[lids] = weight[gids].to(device=param.device, dtype=param.dtype)
        elif key.startswith('w13'):
            # 原有逻辑：w13 权重去交织
            ...
        else:
            # 原有逻辑：w2 权重按中间维度分片
            ...
```

tests/models/inkling/test_moe_weight_layout.py

新增测试覆盖 w13 和 w2 投影的每输出通道尺度加载，确保 TP 下复制逻辑正确。

```
# tests/models/inkling/test_moe_weight_layout.py 中的测试函数片段

@pytest.mark.parametrize(("projection", "checkpoint_rows"), [("w13", 8), ("w2", 4)])
def test_moe_loads_channelwise_scale_for_tp(
    projection: str, checkpoint_rows: int
) -> None:
    # 创建一个形状为 [2, 4, 1] 的参数，模拟 2 个本地专家、每专家 4 个输出通道
    param = torch.nn.Parameter(torch.empty(2, 4, 1))
    # 模拟专家容器，包含一个 w13_weight_scale 或 w2_weight_scale 参数
    experts = SimpleNamespace(
        **{f"{projection}_weight_scale": param},
        moe_config=SimpleNamespace(moe_parallel_config=SimpleNamespace(tp_rank=1)),
    )
    layer = SimpleNamespace(
        experts=SimpleNamespace(routed_experts=experts),
        # 模拟本地专家槽位映射：全局专家 0→本地 0，全局专家 2→本地 1
        _local_expert_slots=lambda: {0: 0, 2: 1},
    )
    # 创建全局 checkpoints 尺度：形状为 [3, checkpoint_rows, 1]
    checkpoint_scale = torch.arange(3 * checkpoint_rows).reshape(3, checkpoint_rows, 1)

    # 调用加载方法
    loaded = moe.InklingMoE.load_expert_weight(
        layer, f"experts.{projection}_weight_scale", checkpoint_scale
    )
```

```
    # 预期：只取本地专家对应的行（全局索引 0 和 2）
    expected = checkpoint_scale[[0, 2]]
    if projection == "w13":
        # w13 的尺度在保存时也是交织的 [g0, u0, g1, u1, ...]
        # TP rank=1 意味着取后半部分（索引 4:），然后重组恢复连续布局
        expected = expected[:, 4:].reshape(2, 2, 2, 1).transpose(1, 2).flatten(1, 2)
    # 断言结果与预期一致
    torch.testing.assert_close(param, expected.float())
    assert loaded == [f"experts.routed_experts.{projection}_weight_scale"]
```

评论区精华

Review 中仅有一条评论：mgoin 询问某处代码是否为 'debug cruft'（指向一个环境变量 INKLING_FP8_SCALE_INTERLEAVED），但该代码在最终提交中已被移除。此外，Claude bot 提示未启用完整审查。

- 调试代码残留 (style): 该代码（环境变量 INKLING_FP8_SCALE_INTERLEAVED）在最终提交中已被移除。

风险与影响

- 风险：风险较低。变更仅 3 行核心逻辑，且通过测试验证。主要风险是：
  - 如果未来 checkpoint 格式变化（例如 w2_weight_scale 的最后一维不为 1），新的分支条件 weight.shape[-1] == 1 可能不匹配，导致回退到旧的错误分片逻辑。但测试已覆盖该场景。
  - 不影响现有 checkpoint（w13 全局 / 输入尺度、w2 全局 / 输入尺度等）。
  - 影响：直接影响 Inkling 模型压缩张量动态 FP8 checkpoint 的加载。对于 RedHatAI/Inkling-FP8-dynamic 模型，PR 使其在 TP 模式下能正确初始化。对其他模型无影响。
  - 风险标记：缺少真实精度验证

关联脉络

- 暂无明显关联 PR