PR #51196 完整报告

vllm-project/vllm

[Kimi][MM] disable kimi_vit's dynamic torch.compile for TPU

合并时间：2026-08-08 08:42

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51196

执行摘要

- 一句话：TPU 上禁用 Kimi ViT 动态 torch.compile
- 推荐动作：值得快速浏览。该 PR 展示了用装饰器参数做平台条件编译的简洁写法，disable 与 skip 的取舍讨论也有参考价值；对后续 TPU 平台适配和 torch.compile 使用有借鉴意义。

功能与动机

PR body 明确说明目的：Conditionally decorate torch.compile because XLA doesn't support dynamic compiling。TPU 上测试表现为断言消失且 ViT 正常工作（on TPU, the assertion is gone and ViT works without other change），即不禁用动态编译时 XLA 后端会触发断言失败，导致视觉编码器无法运行。

实现拆解

1. 定位编译入口：vllm/model_executor/models/kimi_k25_vit.py 中 get_rope_shape 是 rope 频率插值的核心函数，装饰器链由外层 get_rope_shape_decorate（首次调用用固定 (64, 64) shape 预热编译 guard）和内层 @torch.compile(dynamic=True) 组成。
2. 按平台条件禁用编译：将内层装饰器改为 @torch.compile(dynamic=True, disable=current_platform.simple_compile_backend == "tpu")。TPU 平台 simple_compile_backend 返回 tpu，此时 disable=True 让函数直接以 eager 执行、完全绕过编译；其余平台保持原动态编译行为。
3. 方案取舍：review 中 Isotr0py 建议用 skip 参数，作者确认改用 disable 参数并更新 PR；两种写法都能绕过编译，disable 语义更直白地表示当前平台不支持、关闭编译。
4. 配套与测试：未新增测试文件，验证依赖 TPU CI 构建；本次变更不涉及数据结构、权重格式或对外接口，CUDA/ROCm/CPU 路径零影响。

关键文件：

- vllm/model_executor/models/kimi_k25_vit.py（模块 视觉模型；类别 source；类型 platform-gating；符号 get_rope_shape）：唯一变更文件，在 get_rope_shape 的 @torch.compile 装饰器上按平台禁用动态编译，修复 TPU 上 XLA 不支持动态编译导致的失败。

关键符号：get_rope_shape

关键源码片段

vllm/model_executor/models/kimi_k25_vit.py

唯一变更文件，在 get_rope_shape 的 @torch.compile 装饰器上按平台禁用动态编译，修复 TPU 上 XLA 不支持动态编译导致的失败。

```
# 预热装饰器：首次调用时用固定 shape (64, 64) 触发一次编译，
# 以捕获 guard 信息；后续调用直接返回真实 shape 的结果。
def get_rope_shape_decorate(func):
    _get_rope_shape_first_call_flag = set()

    def wrapper(org, interpolation_mode, shape):
        key = (org.requires_grad, torch.is_grad_enabled(), interpolation_mode)
        if key not in _get_rope_shape_first_call_flag:
            _get_rope_shape_first_call_flag.add(key)
            _ = func(org, interpolation_mode, shape=(64, 64))
        return func(org, interpolation_mode, shape)

    return wrapper


# TPU 上 XLA 不支持 dynamic compile，因此当 simple_compile_backend 为 tpu 时
# 通过 disable=True 完全跳过 torch.compile，回退到 eager 执行；
# 其他平台保持 dynamic=True 的动态形状编译加速。
@get_rope_shape_decorate
@torch.compile(dynamic=True, disable=current_platform.simple_compile_backend == "tpu")
def get_rope_shape(org, interpolation_mode, shape):
    # F.interpolate 在 (H, W) 维度上调整 rope 频率形状，再拍平成 1D 返回
    return (
        F.interpolate(
            org.permute((2, 0, 1)).unsqueeze(0),
            size=shape,
            mode=interpolation_mode,
        )
        .squeeze(0)
        .permute((1, 2, 0))
        .flatten(end_dim=1)
    )
```

评论区精华

核心讨论围绕禁用 torch.compile 的参数选择。Isotr0py 在 review 评论中建议使用 skip 参数并附 PyTorch 文档链接；作者 lk-chen 回复确认存在 disable 参数，并立即更新 PR。最终采用 disable 写法，评审人批准合并，无遗留问题。

- 使用 skip 还是 disable 禁用 torch.compile (design): 采用 disable 参数实现条件禁用，评审人批准合并。

风险与影响

- 风险：

1. 平台字符串依赖：禁用条件依赖 current_platform.simple_compile_backend == "tpu"，若 TPU 平台该返回值变化，禁用条件会静默失效。
2. TPU 性能回退：get_rope_shape 在多模态推理中高频调用，TPU 上从编译执行变为 eager 后开销上升，但功能正确优先。
3. 测试覆盖缺口：无专门单测覆盖该分支，回归保障依赖 CI 中 TPU 作业，若未来 CI 未覆盖此路径可能存在遗漏。 - 影响：影响范围极小：仅限 Kimi K25 视觉编码器在 TPU 平台上的 get_rope_shape 路径，单文件单行改动。对 CUDA/ROCm/CPU 等平台无行为变化；TPU 上从动态编译回退 eager，以功能修复优先于性能。对团队而言是一次低成本平台兼容性修复，不引入对外接口或数据结构变更。 - 风险标记：未新增测试覆盖，平台字符串判断，TPU 性能回退

关联脉络

- PR #50585 [K3 Perf] Optimize k3 dspark fused kv, 4.5~4.6x kernel performance improvement: 同属 Kimi 模型族，体现 Kimi 系列在多平台性能与适配上的持续投入。
- PR #51435 [Bugfix][MM] Avoid device sync in FusedInputNorm initialization: 同属多模态视觉模型路径，近期多模态代码在多平台行为修复上密集迭代。