

PR #55887 完整报告

vllm-project/vllm

[ROCm][Bugfix] Support shared KV prefill in AITER attention

合并时间: 2026-09-10 05:51

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/55887>

执行摘要

- 一句话: 修复 AITER 注意力后端在共享 KV 缓存场景下预填充失败的问题。
- 推荐动作: 本 PR 是一个针对性强、影响明确的关键 bug 修复。建议精读 `vllm/v1/attention/backends/rocm_aiter_fa.py` 中 `AiterFlashAttentionImpl.forward` 方法的变更, 以理解共享 KV 缓存的 gather 逻辑。同时, 新增的测试用例 `test_aiter_fa_shared_kv_matches_reference` 设计全面, 是理解 AITER 后端共享缓存行为的优秀参考。该修复值得纳入主干。

功能与动机

PR body 明确指出, 通过 `git bisect run` 从 nightly CI 中识别出 PR#54917 是导致回归的元凶。该 PR 让 Gemma 模型停止计算冗余 K/V 投影, 但 AITER 注意力后端仍尝试切片这些缺失的张量, 破坏了 MI300 和 MI355 上的翻译服务器作业。本 PR 旨在恢复后端对共享缓存的支持。

实现拆解

1. 引入共享 KV 元数据结构: 在 `vllm/v1/attention/backends/rocm_aiter_fa.py` 中新增 `AiterKVSharingMetadata` 数据类, 用于携带共享 KV 缓存操作所需的工作空间、查询位置、token 到批次的映射以及序列起始位置。
2. 修改元数据构建器: 在 `AiterFlashAttentionMetadataBuilder.__init__` 中检测是否存在共享 KV 层, 若存在则根据其 KV 维度 (`num_kv_heads`, `head_size`) 预分配一个共享工作空间张量。在 `build` 方法中, 当存在预填充 / 扩展且共享工作空间可用时, 构建并填充 `kv_sharing_metadata` 实例。
3. 调整前向传播逻辑: 在 `AiterFlashAttentionImpl.forward` 方法中, 当检测到当前层是共享层且存在预填充 / 扩展操作时, 从共享元数据的工作空间中获取为预分配的 key/value 张量, 然后调用 `cp_mha_gather_cache` 函数从 KV 缓存中 gather 新的预填充 / 扩展 token 的 K/V 数据, 填充到这些张量中, 供后续计算使用。同时, 将 `key` 和 `value` 参数的类型注解从 `torch.Tensor` 改为 `torch.Tensor | None`, 以允许在共享场景下传入 `None`。
4. 全面测试覆盖: 在 `tests/kernels/attention/test_rocm_aiter_fa.py` 中新增 `test_aiter_fa_shared_kv_matches_reference` 测试函数。该测试覆盖了 NHD/SHUFFLE 布局、FP8 缓存、滑动窗口和缓存不可变性等多种场景, 并通过参数化组合生成 18 个回归测试用例, 验证共享层读取的输出与参考实现一致, 且缓存内容在操作后保持不变。

关键文件：

- vllm/v1/attention/backends/rocm_aiter_fa.py（模块 注意力后端；类别 source；类型 core-logic；符号 AiterKVSharingMetadata, AiterFlashAttentionMetadataBuilder, AiterFlashAttentionImpl）：核心源码文件，包含共享 KV 缓存元数据结构的定义、元数据构建逻辑以及前向传播中的 gather 操作实现。
- tests/kernels/attention/test_rocm_aiter_fa.py（模块 注意力测试；类别 test；类型 test-coverage；符号 test_aiter_fa_shared_kv_matches_reference）：关键测试文件，新增了针对共享 KV 缓存场景的 18 个回归测试用例，覆盖了多种缓存布局、量化类型和批次模式，验证了修复的正确性和缓存的不可变性。

关键符号：AiterKVSharingMetadata, AiterFlashAttentionMetadataBuilder.init, AiterFlashAttentionMetadataBuilder.build, AiterFlashAttentionImpl.forward, test_aiter_fa_shared_kv_matches_reference

关键源码片段

tests/kernels/attention/test_rocm_aiter_fa.py

关键测试文件，新增了针对共享 KV 缓存场景的 18 个回归测试用例，覆盖了多种缓存布局、量化类型和批次模式，验证了修复的正确性和缓存的不可变性。

```
# 新增测试函数，验证共享 KV 缓存层在预填充和扩展时的行为
@pytest.mark.skipif(not on_mi3xx(), reason="MI300/MI350 ROCm only")
@pytest.mark.parametrize(
    "shuffle,sliding_window", [(False, None), (False, 16), (True, None)]
)
@pytest.mark.parametrize("kv_cache_dtype", ["auto", "fp8"])
@pytest.mark.parametrize("batch", ["prefill", "extend", "mixed"])
def test_aiter_fa_shared_kv_matches_reference(
    monkeypatch, shuffle, sliding_window, kv_cache_dtype, batch
):
    """Shared layers read cached K/V for prefill and extend without modifying it."""
    # ... 导入测试工具和核心组件 ...
    _assert_aiter_supported()
    # 通过 monkeypatch 设置 KV 缓存布局 (NHD 或 SHUFFLE)
    monkeypatch.setattr(rocm_aiter_ops, "_SHUFFLE_KV_CACHE_ENABLED", shuffle)
    # ... 根据 batch 参数创建不同的 BatchSpec ...
    # ... 创建 VllmConfig 和两个 Attention 层 (target 和 shared)，其中 shared 层配置了 kv_
    sharing_target_layer_name ...
    # ... 创建元数据构建器并构建 metadata ...
    # ... 准备共享的 KV 缓存并填充随机数据，确保缓存内容相同 ...
    # ... 复制缓存状态作为基准 ...
    # ... 执行共享层的 forward 方法，key 和 value 传入 None ...
    output = torch.empty_like(query)
    layer.impl.forward(layer, query, None, None, kv_cache, metadata, output)
    # ... 使用参考实现计算期望输出 ...
    expected = ref_paged_attn(...)
    # ... 比较输出，对解码和预填充 / 扩展部分使用不同的容差 ...
```

```
torch.testing.assert_close(output, expected, atol=2e-2, rtol=2e-2)
# 关键断言：操作后 KV 缓存内容必须保持不变（不可变性）
torch.testing.assert_close(
    kv_cache.view(torch.uint8), cache_before, atol=0, rtol=0
)
```

评论区精华

Review 讨论较少，仅有一条由 [claude\[bot\]](#) 发出的自动评论，说明仓库已配置手动代码审查。主要审批者 [dllehr-amd](#) 简单确认了修复的有效性。无实质性技术争议或设计权衡讨论。

- PR 审批与确认 (other): 修复被接受并合并。

风险与影响

- 风险：回归风险：修复直接针对由 PR#54917 引入的回归，逻辑清晰，但依赖于对 `cp_mha_gather_cache` 函数的正确调用，需确保 `gather` 操作在各种缓存布局和量化类型下的正确性。性能影响：为共享层引入了额外的元数据构建和 KV 缓存 `gather` 操作，这会带来微小的开销，但这是功能正确性所必需的。兼容性：修改了 `forward` 方法的签名（允许 `key/value` 为 `None`），这是一个内部 API 变更，但由于 `AiterFlashAttentionImpl` 是后端实现细节，影响范围可控。
- 影响：用户影响：恢复了使用 AITER 注意力后端的 ROCm 用户（如使用 Gemma 等共享 KV 缓存模型）在预填充和扩展阶段的正常推理功能，解决了导致 CI 翻译服务器作业失败的关键问题。系统影响：确保了 AITER 后端在共享 KV 缓存场景下的功能完整性，是维护 ROCm 平台支持的关键一步。团队影响：为 ROCm 团队提供了可靠的共享 KV 缓存支持测试用例，有助于未来开发中的回归检测。
- 风险标记：核心路径变更，平台特定逻辑

关联脉络

- PR #54917 优化 Gemma 模型 K/V 投影：本 PR 明确指出是 PR#54917 引入的回归，后者优化了 Gemma 模型使其不再计算冗余 K/V 投影，但未同步更新 AITER 后端，导致其尝试切片缺失的张量。