PR #48688 完整报告

vllm-project/vllm

[ROCm][Bugfix] Enable the fp32 head_dtype torch.mm fast path on ROCm

合并时间：2026-07-15 16:18

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48688

执行摘要

- 一句话：为 ROCm 启用 fp32 head_dtype torch.mm 快速路径
- 推荐动作：此 PR 值得精读，尤其适合关注 ROCm 支持和性能优化的工程师。它展示了一个小而精准的性能修复案例：通过分析底层库源码将已有的快速路径扩展到新平台，并配以充分的测试验证。

功能与动机

PR #48390 添加了 fp32 head_dtype 的 torch.mm(out_dtype=torch.float32) 快速路径，但仅对 CUDA 启用。ROCm 始终回退到 cast 路径，会产生不必要的 fp32 权重复制。提交者通过分析 PyTorch 源码发现 ROCm 支持该 out_dtype 组合（通过非 hipBLASLt 的 GEMM 路径），因此将快速路径扩展到 ROCm 以消除性能浪费。

实现拆解

1. 扩展快速路径条件：在 vllm/model_executor/layers/logits_processor.py 的 _apply_head 方法中，将条件 current_platform.is_cuda() 改为 (current_platform.is_cuda() or current_platform.is_rocm())，并在注释中说明 ROCm 通过非 Lt GEMM 路径支持该功能。
2. 新增测试用例：在 tests/v1/sample/test_head_dtype.py 中新增 test_fp32_head_uses_mm_fast_path_on_device 函数，使用 mock.patch 确保 cast 路径（F.linear）不被调用，同时验证输出精度与 fp32 参考一致。测试使用 @pytest.mark.skipif 条件跳过无 CUDA 的环境。

关键文件：

- vllm/model_executor/layers/logits_processor.py（模块 逻辑处理器；类别 source；类型 core-logic）：核心变更文件，修改了 _apply_head 方法中的平台判断条件，将快速路径扩展到 ROCm。
- tests/v1/sample/test_head_dtype.py（模块 测试；类别 test；类型 test-coverage；符号 test_fp32_head_uses_mm_fast_path_on_device）：新增测试用例，验证 ROCm 上快速路径被使用且结果正确。

关键符号：_apply_head

关键源码片段

vllm/model_executor/layers/logits_processor.py

核心变更文件，修改了 _apply_head 方法中的平台判断条件，将快速路径扩展到 ROCm。

```
# 关键变更：_apply_head 方法中的快速路径条件
if (
    self.head_dtype == torch.float32
    and (current_platform.is_cuda() or current_platform.is_rocm()) # 原为 is_cuda()
    and hidden_states.is_cuda
):
    # 直接将投影累积到 fp32，避免每步都 materialize 一份 fp32 权重副本。
    # torch.mm(out_dtype=...) 仅支持 fp16/bf16 输入并输出 fp32，
    # 并且只对 CUDA 和 ROCm 实现（ROCm 通过非 Lt 的 GEMM 路径）；
    # 其他平台回退到下面的 cast 路径。
    flat = hidden_states.reshape(-1, hidden_states.shape[-1])
    logits = torch.mm(flat, lm_head.weight.t(), out_dtype=self.head_dtype)
    if embedding_bias is not None:
        logits = logits + embedding_bias.to(self.head_dtype)
    return logits.reshape(*hidden_states.shape[:-1], -1)
# cast 路径：回退到 F.linear，会产生 fp32 权重副本
return F.linear(
    hidden_states.to(self.head_dtype),
    lm_head.weight.to(self.head_dtype),
    embedding_bias.to(self.head_dtype) if embedding_bias is not None else None,
)
```

tests/v1/sample/test_head_dtype.py

新增测试用例，验证 ROCm 上快速路径被使用且结果正确。

```
@pytest.mark.skipif(
    not torch.cuda.is_available(),
    reason="Exercises the torch.mm(out_dtype=...) device fast path, "
    "available on CUDA and ROCm.",
)
def test_fp32_head_uses_mm_fast_path_on_device(default_vllm_config):
    # 在 ROCm 上，current_platform.is_cuda() 为 False，之前会回退到 cast 路径
    # (F.linear) 而不是使用 torch.mm(out_dtype=...)，尽管 ROCm 也支持
    # 该 out_dtype mm 功能（通过非 Lt 的 GEMM 路径）。
    from unittest import mock

    vocab_size, hidden_size, num_tokens = 64, 16, 4
    lp = _build_processor(vocab_size)
    lp.head_dtype = torch.float32

    hidden_states = torch.randn(
        num_tokens, hidden_size, dtype=torch.bfloat16, device="cuda"
    )
    weight = torch.randn(vocab_size, hidden_size, dtype=torch.bfloat16, device="cuda")

    # mock 掉 cast 路径的 F.linear，确保快速路径被使用
```

```
with mock.patch(
    "vllm.model_executor.layers.logits_processor.F.linear"
) as linear_mock:
    logits = lp._get_logits(hidden_states, _FakeLmHead(weight), None)

linear_mock.assert_not_called() # 快速路径不应调用 F.linear
assert logits.dtype == torch.float32
# 与 fp32 参考比较精度
expected = torch.nn.functional.linear(hidden_states.float(), weight.float())
torch.testing.assert_close(logits, expected)
```

评论区精华

无实质性 review 讨论。审核者 noooop 直接批准，提交者 wjabbour 表达感谢。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅将条件从 is_cuda() 扩展为 is_cuda() or is_rocm()，实际上 CUDA 路径完全不受影响；提交者已在 ROCm 硬件（Radeon RX 9070 XT）上验证了正确性和精度。唯一的潜在风险在于未来 PyTorch 可能修改 ROCm 上 torch.mm(out_dtype=...) 的行为，但此变更与 PyTorch 内部实现一致，且测试会捕获回归。
- 影响：性能：消除每步 fp32 权重矩阵的冗余分配和复制，对 ROCm 上使用 fp32 head_dtype 的模型（如 DeepSeek）有显著性能提升，尤其在大 vocabulary 场景下。正确性：无影响，数学等价于 cast 路径。兼容性：不影响 CUDA，仅扩展支持到 ROCm。
- 风险标记：核心路径变更（条件判断）

关联脉络

- PR #48390 fp32 head_dtype torch.mm fast path: 本 PR 扩展了 #48390 引入的快速路径到 ROCm。
- PR #48525 LoRA path support for fp32 head_dtype: #48390 的后续，增加了 LoRA 路径支持，本 PR 不涉及 LoRA。
- PR #48654 ROCm CI test fix for unrelated CPU-dispatch issue: 同一测试文件的 CI 修复，但不涉及快速路径逻辑。