PR #47534 完整报告

vllm-project/vllm

[Test][LoRA] Use lightweight CPU reference and skip heavy cleanup in punica ops tests

合并时间：2026-07-06 11:29

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47534

执行摘要

- 一句话：使用 CPU 参考加速 Punica 测试 5.1×
- 推荐动作：值得精读，尤其是设计合理 CPU 参考并消除无用夹具以加速测试的方法。对于需要优化测试性能或规避特定平台算子 bug 的场景有参考价值。

功能与动机

原测试使用 torch_ops.sgmv_shrink/expand 作为参考，内部通过 torch.einsum 实现，会物化大型 lora_weight[exploded_indices] 张量（大 batch/rank 下可达 ~8 GB），且 XPU 上 oneDNN/oneMKL 在大量 Triton kernel 启动后返回错误结果。此外，继承自 conftest.py 的 cleanup_fixture 每次测试后执行重量级分布式清理（含 ray.shutdown、gc.collect 等），而 punica 内核测试完全不使用分布式或 Ray，造成巨大开销。

实现拆解

1. 新增 CPU 参考实现：在 tests/lora/test_punica_ops.py 中新增 _cpu_bgmv_shrink 和 _cpu_bgmv_expand 两个轻量函数。它们将输入、权重和输出移动到 CPU，通过 torch.repeat_interleave 展开 LoRA 索引，然后逐 LoRA 执行 inp @ w.T 循环。这种做法避免物化完整 lora_weight[exploded_indices] 大张量，且使用 CPU 计算绕过 XPU 上的算子 bug。
2. 重写参考接口函数：将原有的 sgmv_shrink_for_nslices 和 sgmv_expand_for_nslices 改为调用上述 CPU 函数，并移除了对 vllm.lora.ops.torch_ops 的导入依赖。现在所有参考计算都在 CPU 上完成，最终通过 out_tensor.copy_(out_cpu) 写回 GPU。
3. 覆写无操作夹具：增加两个自动使用的 fixture cleanup_fixture 和 dynamo_reset，它们只是 yield 而不做任何清理。这覆盖了 conftest.py 中全局的对应 fixture，从而避免了每测试后执行 cleanup_dist_env_and_memory（含 destroy_model_parallel、destroy_distributed_environment、ray.shutdown()、gc.collect()、torch.accelerator.empty_cache()）和 torch._dynamo.reset()。

该变更仅涉及一个测试文件，无其他代码或配置改动。

关键文件：

- tests/lora/test_punica_ops.py（模块 LoRA 测试；类别 test；类型 test-coverage；符号 cleanup_fixture, dynamo_reset, _cpu_bgmv_shrink, _cpu_bgmv_expand）：唯一变更文件，包含所有改动：新增 CPU 参考函数、覆写夹具、修改参考接口。

关键符号：cleanup_fixture, dynamo_reset, _cpu_bgmv_shrink, _cpu_bgmv_expand, sgmv_shrink_for_nslices, sgmv_expand_for_nslices

关键源码片段

tests/lora/test_punica_ops.py

唯一变更文件，包含所有改动：新增 CPU 参考函数、覆写夹具、修改参考接口。

```
# tests/lora/test_punica_ops.py

import pytest
import torch

# ...（其他导入）

@pytest.fixture(autouse=True)
def cleanup_fixture():
    """覆盖 conftest 中全局的 cleanup_fixture —— punica 测试不需要分布式清理。"""
    yield # 无操作


@pytest.fixture(autouse=True)
def dynamo_reset():
    """覆盖 conftest 中全局的 dynamo_reset —— punica 测试不使用 torch.compile。"""
    yield # 无操作


def _cpu_bgmv_shrink(
    inputs, lora_weight, output, seq_len_tensor, lora_indices, scaling=1.0
):
    """
    轻量级 shrink 参考：在 CPU 上逐 LoRA 执行 matmul，
    避免物化大张量 lora_weight[exploded_indices]（~8 GB）。
    output[mask] = scaling * inputs[mask] @ weight.T
    """
    # 将每个 token 映射到其所属的 LoRA ID
    exploded = torch.repeat_interleave(lora_indices, seq_len_tensor)
    for lid in exploded.unique():
        if lid < 0:
            continue
        mask = exploded == lid
        inp = inputs[mask].to(output.dtype)
        w = lora_weight[lid].to(output.dtype)
        output[mask] = scaling * (inp @ w.T)


def _cpu_bgmv_expand(
    inputs, lora_weight, output, seq_len_tensor, lora_indices,
    offset=0, add_inputs=False
```

```
):
    """
    轻量级 expand 参考：与 shrink 类似，处理输出切片偏移和残差连接。
    output[mask, offset:offset+n] (+)= inputs[mask] @ weight.T
    """
    exploded = torch.repeat_interleave(lora_indices, seq_len_tensor)
    for lid in exploded.unique():
        if lid < 0:
            continue
        mask = exploded == lid
        inp = inputs[mask].to(output.dtype)
        w = lora_weight[lid].to(output.dtype)
        n = w.shape[0]
        result = inp @ w.T
        if add_inputs:
            output[mask, offset : offset + n] += result
        else:
            output[mask, offset : offset + n] = result

# 后续修改了 sgmv_shrink_for_nslices 和 sgmv_expand_for_nslices 以调用上述 CPU 函数
```

评论区精华

该 PR 只有一次实质审核，来自 *jeejeelee* 的批准（“Thank you”），无争议讨论。Mergify bot 自动指出合并冲突和 pre-commit 检查失败，已由作者解决。

- Approval and CI issues (other): 无争议，已合并。

风险与影响

- 风险：风险极低：
  - 仅修改测试文件，不影响生产代码路径。
  - CPU 参考实现通过 assert_close 与 Triton kernel 对比，精度风险可控；且原本的 torch_ops 参考在 XPU 上已知有 bug，切换为 CPU 后反而更可靠。
  - 覆写夹具为无操作不会引发问题，因为 punica 测试确实不需要分布式或 torch.compile。
  - 可能在极少数场景下 CPU 结果与 GPU 结果存在浮点差异，但 assert_close 容差足够覆盖。
  - 影响：
  - 用户：无直接影响，该变更仅影响 CI 测试。
  - 系统：Punica 操作测试的 CI 时间从 13 分 55 秒降至 2 分 43 秒（5.1× 加速），显著提升开发者迭代速度。
  - 团队：后续新增 LoRA kernel 测试时可以参考此模式；XPU CI 不再因 oneDNN bug 出现误报。
  - 风险标记：低风险，仅测试文件，无核心逻辑变更

关联脉络

- PR #43957 [XPU] Fix Eagle3 initialization on XPU: 同为 XPU 平台相关修复，本 PR 的 CPU 参考也用于规避 XPU oneDNN/oneMKL bug。
- PR #43092 [XPU] Fix CUDA API shims breaking Torch Dynamo during AOT compile: 同为 XPU 平台问题，本 PR 覆写 dynamo_reset 避免 Torch Dynamo 相关开销。