PR #48072 完整报告

vllm-project/vllm

[CI][CPU] Add Qwen2-VL multimodal tests for CPU backend and fix incompatibilities

合并时间：2026-07-12 12:30

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48072

执行摘要

- 一句话：CPU 后端 Qwen2.5-VL 多模态测试覆盖与 CI 整合
- 推荐动作：值得快速阅读，学习如何在多平台项目中利用 current_platform 条件化测试参数，以及如何设计 CI job 隔离以减少测试时间。核心逻辑变更不大，技术决策清晰。

功能与动机

PR body 明确指出需要支持 CPU 后端的多模态嵌入减少。修复两个关键问题：

1) pin_memory 在 CPU 设备上的不兼容； 2) 跨进程 RPC 中本地函数序列化失败。最终目标是使 Qwen2-VL / Qwen2.5-VL 能够在纯 CPU 部署下运行多模态推理。

实现拆解

实现步骤

1. 测试文件动态参数化(tests/models/multimodal/generation/test_qwen2_5_vl.py) - 引入 from vllm.platforms import current_platform，在 @pytest.mark.parametrize 装饰器中使用 current_platform.is_cpu() 条件表达式，使 CPU 后端只保留最核心的参数组合（例如 video_pruning_rate 仅保留 [0.0]、use_bytecode_hook 仅保留 [True]），避免运行不支持的配置并缩短测试时间。
2. CI 配置调整(.buildkite/hardware_tests/cpu.yaml) - 在已有的 CPU-Multi-Modal Model Tests job 中，通过 --ignore=tests/models/multimodal/generation/test_qwen2_5_vl.py 将 Qwen2.5-VL 测试排除，避免重复运行。 - 新增独立的 CPU-Qwen2.5-VL Multimodal Tests job，使用与通用多模态相同的 source_file_dependencies（vllm/model_executor/layers/rotary_embedding 和 tests/models/multimodal/generation/），通过 VLLM_CI_ENV=0 环境变量并专门运行 test_qwen2_5_vl.py。
3. pin_memory 修复尝试（最终撤回） - 在 review 过程中，曾尝试在 vllm/multimodal/inputs.py 的 reduce_data 函数中增加 CPU 检测逻辑以禁用 pin_memory，但经 reviewer 确认，测试在没有该修改的情况下也能通过，最终该修改被移除。

关键文件：

- tests/models/multimodal/generation/test_qwen2_5_vl.py（模块 Qwen2.5-VL；类别 test；类型 test-coverage）：测试核心文件，通过条件参数化适配 CPU 后端，避免运行不支持的参数组合。

- .buildkite/hardware_tests/cpu.yaml（模块 CPU CI；类别 test；类型 test-coverage）：CI 配置文件，将 Qwen2.5-VL 测试从通用多模态 job 中分离并新增独立 job，实现并行化。

关键符号：未识别

关键源码片段

tests/models/multimodal/generation/test_qwen2_5_vl.py

测试核心文件，通过条件参数化适配 CPU 后端，避免运行不支持的参数组合。

```
# SPDX-License-Identifier: Apache-2.0
import pytest
from vllm.platforms import current_platform

# 根据当前平台动态裁剪参数组合：
# CPU 后端只运行最核心的配置，跳过 video_pruning_rate=0.75 和 use_bytecode_hook=False 组合，
# 以加速 CI 同时避免潜在的不兼容性。
@pytest.mark.core_model
@pytest.mark.parametrize("model", ["Qwen/Qwen2.5-VL-3B-Instruct"])
@pytest.mark.parametrize(
    "video_pruning_rate",
    [0.0] if current_platform.is_cpu() else [0.0, 0.75]
)
@pytest.mark.parametrize("num_frames", [16])
@pytest.mark.parametrize("dtype", ["bfloat16"])
@pytest.mark.parametrize("max_tokens", [128])
@pytest.mark.parametrize(
    "use_bytecode_hook",
    [True] if current_platform.is_cpu() else [True, False]
)
def test_qwen2_5_vl_evs_functionality(
    vllm_runner, video_assets, model,
    video_pruning_rate: float, num_frames: int,
    dtype: str, max_tokens: int, use_bytecode_hook: bool, monkeypatch
) -> None:
    """Test EVS functionality."""
    monkeypatch.setenv("VLLM_USE_BYTECODE_HOOK",
                       "1" if use_bytecode_hook else "0")
    # 其余主体逻辑保持不变，以聚焦参数化变更
```

评论区精华

Review 讨论主要集中在以下方面：

- 测试参数动态化：reviewer bigPYJ1151 建议使用 current_platform.is_cpu() 来分配不同参数，作者据此修改。
- CI 分离：reviewer 建议将新启用测试排除在通用 job 之外并独立运行，避免 CI 时间过长；同时建议使用与通用 job 相同的依赖列表，作者均采纳。

- pin_memory 修复：作者尝试增加 CPU 设备检测禁用 pin_memory，但 reviewer 指出可以更优雅地使用已有 device 变量，最终作者验证测试无需此修改，将其撤回。
- 测试参数动态化条件化 (testing): 采用条件参数化，CPU 上只保留 video_pruning_rate=0.0 和 use_bytecode_hook=True，其他平台保持全组合。
- CI job 拆分与依赖配置 (infra): 在通用 job 中添加 --ignore 排除 Qwen2.5-VL 测试，新增独立 job 并复用相同的 source_file_dependencies。
- pin_memory 修复必要性确认 (correctness): 移除 inputs.py 中的 pin_memory 修改，确认 CPU 环境下无需额外修复。

风险与影响

- 风险：本 PR 仅涉及测试与 CI 配置变更，未改动任何核心运行时逻辑，引入回归风险极低。主要风险包括：
  - 新添加的 CI job 可能因环境（如缺少 CPU 硬件标签或依赖配置错误）而失败，但已使用与现有 job 相同的依赖模式。
  - 测试参数缩减可能导致某些隐藏的 CPU 兼容性问题未被检出；不过保留的组合验证了最核心功能。
  - 影响：
  - 用户影响：无直接影响，纯内部 CI 改进。
  - 系统影响：降低 CPU CI 整体耗时（将 Qwen2.5-VL 测试分离后可并行执行）。
  - 团队影响：为 CPU 多模态提供了自动化回归检测，有助于后续 CPU 版本演进。
  - 风险标记：CI 配置变更，测试覆盖缩减风险

关联脉络

- 暂无明显关联 PR