PR #52976 完整报告

vllm-project/vllm

[CI][ROCm] Standardize AMD test job labels by device

合并时间：2026-08-20 11:33

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52976

执行摘要

- 一句话：统一 193 个 AMD CI 任务命名，修复 MI300 超时
- 推荐动作：建议 CI / 测试基础设施负责人精读：本文是大规模 CI 配置重构的样板，重点看命名唯一性策略、CPU 任务转 GPU 的取舍与注释保留方式；khluu 对 PR body 与 diff 不一致的核查也值得作为 review 流程范例。普通工程师仅需了解超时与路径修复即可，无需深入。

功能与动机

PR body 明确了主要动机：'Standardize all 193 .buildkite/test-amd.yaml labels on :amd: (<Device>) <Purpose> [Shard %N], matching the convention introduced for test_areas in #52659'，即与 test_areas 区引入的设备级命名约定对齐，同时移除旧式 GPU 计数与跨平台后缀。khluu 在评论中补充了超时修复的真实背景：main 分支 #84629 / #84630 的多模态任务在 50 分钟超时（原始 + 重试）均失败，因此需将 MI300 任务超时提升到 70 分钟。

实现拆解

1. 统一命名约定：在 .buildkite/test-amd.yaml 中将全部 193 个测试任务 label 规范为 ：amd: (<Device>) <Purpose> [Shard %N]，与 #52659 为 test_areas 引入的约定对齐；移除旧式 GPU 计数与跨平台后缀，改用语义限定符保证唯一性，避免依赖 label 的过滤与通知失效。
2. CPU 任务转 GPU 并保留备份：将 7 个活动的 CPU 任务（如 Basic Models Other、Multimodal Processor、V1 Others、Async Engine/Inputs/Utils/Worker/Config 等）替换为等效 MI250 ROCm GPU 任务，显式保留 1 个 MI250 GPU；原始 CPU 定义以 no_gpu: true、optional: true 注释形式保留在文件末尾并标记 # TBD。提交历史 Restore CPU-only AMD test groups 表明该方案经过返工，最终选择“替换 + 注释保留”而非直接删除。
3. 配套修复：
   - .buildkite/test_areas/models_multimodal.yaml：MI300 Multimodal Models (Standard) 4 的 timeout_in_minutes 从 50 提升到 70，修复 main 分支 #84629 / #84630 的持续超时；
   - .buildkite/test_areas/distributed.yaml：multi-API server 测试路径从 entrypoints/openai/test_multi_api_servers.py 更新为

entrypoints/launchers/api_server/test_multi_api_servers.py，与当前仓库目录结构对齐；

- tests/models/test_vision.py：4 处分布式视觉测试 worker 的设备字符串从 current_platform.device_name 改为 current_platform.device_type。

4. 验证与收敛：多次触发 AMD CI 与主 CI（Buildkite AMD CI #12247、#12249、#12250、#12264 与主 CI #84656、#84691），MI300 多模态任务在超时提升后以 44m55s 通过，MI355 paired job 以 32m23s 通过；重复 PR #52991 被关闭并合入本 PR。

关键文件：

- .buildkite/test-amd.yaml（模块 CI 编排；类别 config；类型 configuration）：核心变更文件，193 个任务标签统一标准化，7 个 CPU 任务替换为 MI250 GPU 任务并注释保留，同时更新文件头 mirror_hardwares 语义。
- tests/models/test_vision.py（模块 视觉测试；类别 test；类型 test-fix；符号 run_dp_sharded_vision_model_vs_direct, run_dp_sharded_mrope_vision_model_vs_direct, run_dp_sharded_mrope_vision_model_empty_input_worker, run_dp_sharded_mrope_vision_model_uneven_load_worker）：本次唯一的源码级改动，4 处分布式视觉测试 worker 的设备选择从 device_name 调整为 device_type，适配 AMD 平台。
- .buildkite/test_areas/models_multimodal.yaml（模块 多模态 CI；类别 config；类型 configuration）：MI300 多模态标准任务超时 50→70 分钟，修复 main #84629 / #84630 的真实超时失败，且该变更与 PR body 描述不一致，是 review 争论焦点。
- .buildkite/test_areas/distributed.yaml（模块 分布式测试；类别 config；类型 configuration）：更新 multi-API server 测试路径，与当前仓库目录结构对齐，避免 NVIDIA / AMD 镜像中的路径失效。

关键符号：run_dp_sharded_vision_model_vs_direct, run_dp_sharded_mrope_vision_model_vs_direct, run_dp_sharded_mrope_vision_model_empty_input_worker, run_dp_sharded_mrope_vision_model_uneven_load_worker

关键源码片段

tests/models/test_vision.py

本次唯一的源码级改动，4 处分布式视觉测试 worker 的设备选择从 device_name 调整为 device_type，适配 AMD 平台。

```
def run_dp_sharded_vision_model_vs_direct(
    local_rank: int, world_size: int, batch_size: int, master_port: int
):
    """验证 run_dp_sharded_vision_model 与直接调用模型的输出一致。"""
    set_random_seed(0)

    # 设备字符串改用 device_type 而非 device_name，以适配 AMD CI 的设备选择方式；
    # 设备号由 local_rank 拼接，最终形如 "<device_type>:<rank>"
    device = f"{current_platform.device_type}:{local_rank}"
```

```
    torch.accelerator.set_device_index(device)
    torch.set_default_device(device)

    update_environment_variables(
        {
            "RANK": str(local_rank),
            "LOCAL_RANK": str(local_rank),
            "WORLD_SIZE": str(world_size),
            "MASTER_ADDR": "localhost",
            "MASTER_PORT": str(master_port),
        }
    )

    # 初始化分布式环境与 2 卡 tensor parallel，随后对比直接推理与分片推理结果
    init_distributed_environment()
    with ensure_current_vllm_config():
        initialize_model_parallel(tensor_model_parallel_size=world_size)

    image_input = torch.randn(batch_size, 3, 224, 224)
    vision_model = SimpleLinearModel()
    with torch.inference_mode():
        direct_output = vision_model(image_input)
    with torch.inference_mode():
        sharded_output = run_dp_sharded_vision_model(image_input, vision_model)
    # 剩余部分：在 rank 0 上断言两者形状一致且 torch.allclose
```

评论区精华

核心讨论来自 khluu 的合并前把关：她指出 PR body 声称超时不变，但 diff 实际将 MI300 Multimodal Models (Standard) 4 超时从 50 提升至 70 分钟（失败证据为 main #84629 / #84630 在 50 分钟窗口内原始与重试均超时），并要求合并前更正描述；同时确认 head 提交 1656b2782c 的修复覆盖在 AMD CI #12250 上通过（MI300 44m55s、MI355 32m23s），并关闭了重复 PR #52991。提交历史还揭示了两个设计反思：Restore CPU-only AMD test groups 表明最初删除 CPU 任务的方案被推翻，改为“GPU 替换 + 注释保留”；Avoid invalid Buildkite step keys 说明重命名过程中同步处理了 step key 合法性约束。最终 reviewer dllehr-amd 直接批准。

- PR body 与超时改动不一致 (correctness)：合并前需更正 PR 描述；调整后 AMD CI #12250 以 44m55s 通过，修复得到验证。
- CPU-only 任务替换为 MI250 GPU 任务的设计 (design)：采用“GPU 替换 + 注释保留 + optional”方案，保留区标记 # TBD 供后续决策。
- vision worker 设备字符串 device_name → device_type (question)：已随 AMD CI 与主 CI 验证通过，但缺少针对设备选择语义的专项测试。

风险与影响

- 风险：

1. .buildkite/test-amd.yaml 单文件 700+ 行重排，193 个 label 全部变化，依赖旧 label 的 Buildkite 通知、过滤规则或外部脚本可能失效。
2. 7 个 CPU-only 任务转为 MI250 GPU 任务，AMD GPU 资源占用显著增加；保留的 CPU 注释区（no_gpu: true）与 GPU 替换版（device: mi250_1）共存，存在误读误配风险，且保留区均标记 optional: true，不参与默认调度。
3. tests/models/test_vision.py 的 device_name → device_type 只随全量 CI 回归验证，缺少针对设备选择语义的专项测试；若某平台 device_type 返回不合法前缀，会导致 2 卡视觉测试失败。
4. 超时提升至 70 分钟会让真正卡死的任务多等待 20 分钟，且该变更未在 PR body 中披露，属于 review 盲区（已被 khluu 指出）。
5. 三次 merge main 带来的冲突掩盖风险，已由 head 上的 AMD CI 与主 CI 双绿解除。 - 影响：面向最终用户无行为变化，全部影响集中在 CI 基础设施：job 命名从碎片化格式（含 GPU 计数、跨平台后缀）收敛为统一模板 :amd: (<Device>) <Purpose> [Shard %N]，Buildkite dashboard 的可读性与可筛选性提升，并与 #52659 的 test_areas 约定对齐；AMD CI 资源模型变化，7 个任务从 CPU 转向 GPU，多模态任务超时窗口延长；团队后续新增 AMD 任务可直接套用命名模板，降低心智成本；重复 PR #52991 被关闭，避免维护分叉。 - 风险标记：大规模 CI 配置重命名，CPU 转 GPU 资源变化，超时与描述不一致，设备选择变更缺少专项验证

关联脉络

- PR #52659 (title unavailable) test_areas AMD device label convention: PR body 明确本 PR 的命名约定是 matching the convention introduced for test_areas in #52659，是本次标准化的模板来源。
- PR #52991 (title unavailable) Duplicate of #52976, closed by khluu: khluu 在评论中确认关闭了窄范围重复 PR #52991，将相关改动并入本 PR。