

PR #56522 完整报告

vllm-project/vllm

[ROCm][CI] Extend timeout for `Basic Models (other)`

合并时间: 2026-09-12 06:12

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56522>

执行摘要

- 一句话: MI300 模型测试超时延长至 65 分钟, 提升 CI 可靠性
- 推荐动作: 不值得精读, 属于机械的 CI 配置调整。值得关注的一点是 vLLM 的 CI 中存在 test_areas mirror 机制与平台独立流水线 (.buildkite/test-amd.yaml) 并行的双维护模式, 同一任务的配置需要在两处重复维护, 这是后续 CI 重构时可考虑的收敛点。

功能与动机

PR body 明确说明: `Extend timeout for (MI300) Basic Models (Other) to improve reliability`, 并附上了一次超时的 Buildkite 构建链接作为证据。该任务在 `buildkite/test-amd.yaml` 中标记为 `optional: true`, 超时本身不会阻塞合并, 但会产生失败噪音、占用 MI300 agent 资源, 并在 mirror 流水线场景下造成任务失败误报。结合 PR #55667 将 `VLLM_ROCM_USE_AITER=1 pytest ... test_hyv4_rocm.py` 加入同一任务, 任务整体耗时上升, 原有 50 / 55 分钟上限已不足以覆盖正常执行时间。

实现拆解

本 PR 的变更入口是 Buildkite 流水线配置, 不涉及任何源码逻辑, 按以下 3 步完成:

1. 修改 mirror 定义中的 AMD 分支: 在 `.buildkite/test_areas/models_basic.yaml` 中, (H200 MIG 35GB) Basic Models (Other) 任务下的 `mirror.amd` 分支定义了名为 (MI300) Basic Models (Other) 的派生任务, 其 `timeout_in_minutes` 由 50 调整为 65。这个 mirror 机制让 AMD 任务从 NVIDIA 任务派生, 但超时上限需要独立配置, 因此无法通过修改父任务自动同步。
2. 同步修改独立 AMD 流水线: 在 `.buildkite/test-amd.yaml` 中, 存在一份独立定义的 (MI300) Basic Models (Other) 任务 (带 # TBD 注释), 其 `timeout_in_minutes` 由 55 调整为 65。该文件是 AMD 单独维护的流水线, 与 test_areas 的 mirror 机制并行存在, 两处必须保持数值一致, 否则会出现同一任务在两套流水线上超时行为不同的配置漂移。
3. CI 验证: 合并者 AndreasKaratzas 在 Issue 评论中触发 `/ci run`, 由 `github-actions[bot]` 启动 Buildkite CI #88415 对 commit `6dfb5de306ac` 进行验证, 确认修改后的流水线配置可正常解析和执行。

测试与部署配套: 无源码或测试变更, `parallelism`、`source_file_dependencies`、测试命令均保持不变, 仅调整超时上限这一单个配置键。

关键文件:

- `.buildkite/test_areas/models_basic.yaml` (模块 CI 配置; 类别 config; 类型 configuration) : 该文件通过 mirror 机制定义 AMD 派生任务, 本次将 (MI300) Basic Models (Other) 的超时从 50 分钟提升至 65 分钟, 是主变更点, 直接影响由 test_areas 生成的 AMD mirror 流水线。
- `.buildkite/test-amd.yaml` (模块 CI 配置; 类别 config; 类型 configuration) : AMD 独立维护的 Buildkite 流水线中同样定义了 (MI300) Basic Models (Other) 任务, 本次将超时从 55 分钟提升至 65 分钟, 与 mirror 定义保持一致, 避免配置漂移。

关键符号: 未识别

评论区精华

本 PR 没有实质性的技术讨论或设计交锋。关键记录如下:

- claude[bot] 自动评论指出: This pull request is from a fork — automated review is disabled, 即 fork 来源的 PR 跳过自动审核, 需维护者手动触发一次性 review。
- 维护者 AndreasKaratzas 直接批准 (APPROVED), 并通过 `/ci run` 触发 Buildkite CI #88415 完成验证后合入。
- 整个评审过程无反对意见、无未解决问题, 改动虽小但有真实构建日志作为超时佐证, 可信度较高。
- Fork PR 自动审核被跳过 (other): 维护者 AndreasKaratzas 手动完成审查并直接批准, 同时通过 `/ci run` 触发 Buildkite CI #88415 验证流水线, 风险已闭环。

风险与影响

- 风险: 风险总体很低, 但仍有两点值得留意:
- 配置双处同步风险: 同一任务的 `timeout_in_minutes` 同时存在于 `.buildkite/test_areas/models_basic.yaml` 的 mirror 分支和 `.buildkite/test-amd.yaml` 独立流水线中。未来若只修改其中一处, 会导致两套流水线行为不一致, 建议后续将该配置收敛到单一数据源。
- 超时放宽可能掩盖性能退化: 65 分钟上限相比原来的 50 / 55 分钟多出 10-15 分钟缓冲, 若模型初始化或 AITER 相关路径出现渐进式性能退化, 原先会在 50 分钟内触发超时告警的退化可能被新上限吸收, 延迟团队的感知时间。建议结合历史任务耗时监控定期审视该值是否仍合理。
 - 对 NVIDIA CI、运行时、用户均无影响, 因为改动只作用于 mi300_1 agent 池。
- 影响: 影响范围仅限于 AMD ROCm CI: (MI300) Basic Models (Other) 任务在 mirror_hardware 中覆盖 amdexperimental、amdproduction、amdgfx942nightly、amdmi300 等硬件池, 超时延长后这些池上的任务失败判定将延后最多 15 分钟。对用户无感知, 对团队的主要收益是减少频繁超时导致的失败噪音与重跑成本, 提升 CI 结果可信度。由于任务本身是 `optional: true`, 超时延长的紧急程度有限, 属于稳定性治理而非阻塞性修复。
- 风险标记: 配置双处同步, 超时放宽可能掩盖性能退化

关联脉络

- PR #55667 [ROCm][CI] Add HY-V4 generation coverage: 该 PR 将 HY-V4 生成测试 (test_hyv4_rocm.py) 接入 MI300 CI 的同一 (MI300) Basic Models (Other) 任务, 新增的 AITER 测试命令显著增加任务耗时, 是本次超时延长的直接诱因。
- PR #56459 [ROCm][Docker] Pin AINIC apt repo to snapshot 1.117.5-a-77: 同属近期 ROCm CI 稳定性治理线, 说明 AMD 侧 CI 正在通过固定依赖、调整超时等手段收敛环境不确定性。
- PR #56433 [ROCm][Bugfix] Fix AITER preshuffled FP8 block-scale kernel: 该修复涉及 vllm/models/deepseek_v4/amd/rocm.py 与 AITER 执行路径, 与同一测试任务覆盖的 test_deepseek_v4_vl_rocm.py 相关, 反映 ROCm 模型测试任务内容持续增长的背景。