PR #34811 完整报告

sgl-project/sglang

test: restore GLM-4.1V nightly latency threshold

合并时间：2026-08-15 18:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34811

执行摘要

- 一句话：恢复 GLM-4.1V nightly 延迟阈值至校准值 35.5s
- 推荐动作：值得快速浏览以理解测试阈值治理的教训：一行改动虽小，但背后是"重构移除 guard、恢复时带错旧值"的典型回归链。若要精读，重点看 MODEL_THRESHOLDS 字典的注释约定（1024-token CoT 预算对延迟基线的影响），并思考是否把阈值列表纳入静态 ratchet 防止再次丢失。

功能与动机

PR body 明确指出：阈值最初在 a685806d 中按 1.2 倍五次 nightly 运行最慢值校准为 35.5s；随后多模型 eval 重构移除了延迟守卫，而 #34662 在恢复守卫时误将旧值 30.4s 一并带回。这导致实测 33.259s 的运行时被误判为失败，尽管该值仍在校准上限 35.5s 之内。本次变更即是消除该回归引入的误报。

实现拆解

1. 根因定位：对比 git 历史确认 35.5s 是 a685806d 基于五次 nightly 运行中最慢值放大 1.2 倍得出的校准阈值；#34662 在恢复 latency guard 时意外带回了旧的 30.4s 阈值，造成 33.259s 的正常结果被误报为超时。
2. 单行恢复：在 test/registered/eval/test_vlms_mmmu_eval.py 的 MODEL_THRESHOLDS 字典中，将 ModelLaunchSettings("zai-org/GLM-4.1V-9B-Thinking") 的延迟阈值从 30.4s 改回 35.5s，分数阈值 0.280 保持不变。
3. 验证配套：作者执行了 pre-commit 检查、py_compile 编译检查和 git diff --check；随后两次在 Issue 评论中发起 /rerun-test，github-actions[bot] 均在 2-gpu-h100 上报告 test_vlms_mmmu_eval.py 测试通过。

关键文件：

- test/registered/eval/test_vlms_mmmu_eval.py（模块 多模态评测；类别 test；类型 configuration）：唯一改动文件，将 GLM-4.1V-9B-Thinking 的 MMMU 延迟阈值从 30.4s 恢复为校准值 35.5s，消除 nightly 误报。

关键符号：未识别

关键源码片段

test/registered/eval/test_vlms_mmmu_eval.py

唯一改动文件，将 GLM-4.1V-9B-Thinking 的 MMMU 延迟阈值从 30.4s 恢复为校准值 35.5s，消除 nightly 误报。

```
# MMMU 100 样本评测的分数与延迟阈值表。
# 延迟单位为秒。这里特意恢复 GLM-4.1V 的校准阈值 35.5s，
# 而不是保留 #34662 恢复 guard 时误带入的旧值 30.4s：
# 35.5s 是 a685806d 按 1.2 倍五次 nightly 中最慢一次校准的，
# 实测 33.259s 仍在 35.5s 上限之内。
MODEL_THRESHOLDS = {
    ...
    # 顺序为：MMMU 分数阈值、延迟阈值
    ModelLaunchSettings("zai-org/GLM-4.1V-9B-Thinking"): (0.280, 35.5),
    ...
}
```

评论区精华

本 PR 没有正式 review 评论，讨论集中在 Issue 评论区：作者 mickqian 两次发起 /rerun-test test/registered/eval/test_vlms_mmmu_eval.py，github-actions[bot] 两次报告在 2-gpu-h100 上测试通过。这体现了团队使用 /rerun-test 机器人对 nightly 测试做定向复验的协作模式，也侧面确认了阈值恢复后误报消除。

- 阈值恢复后的 rerun-test 验证 (testing): 恢复为 35.5s 阈值后 nightly MMMU eval 测试通过，误报消除。

风险与影响

- 风险：风险整体很低，但仍需注意：
  - 35.5s 阈值是按当时 2-gpu-large runner 环境校准的，若硬件调度、模型权重或评测样本变化，可能再次出现误报（收紧）或漏报（放宽）。
  - 仅修改一个模型条目，其他模型的阈值不受影响。
  - 没有新增自动化测试固定该值；未来若再经历多模型 eval 类似的重构，阈值仍有被误带或丢失的可能。可考虑将 MODEL_THRESHOLDS 的治理纳入静态 ratchet 检查（与近期 config ratchet 系列 PR 的思路一致）。
  - 影响：影响范围限于 nightly MMMU eval 测试：消除了 GLM-4.1V-9B-Thinking 的重复误报，减少 CI 噪音和人工 rerun 成本，让维护者更信任 nightly 结果。对运行时和用户无任何影响。团队层面，这是一次低成本的测试卫生修复，提醒后续重构在恢复 guard 时应回溯 git 历史核对阈值。
  - 风险标记：测试阈值调整，依赖运行环境稳定性，重构回归风险

关联脉络

- PR #34662 （标题未提供）：PR#34811 body 明确指出 #34662 在恢复 latency guard 时误将 GLM-4.1V 的延迟阈值恢复为旧值 30.4s，是本次误报的直接引入方。