

PR #34300 完整报告

sgl-project/sglang

test: isolate metal profiler tests from ambient SGLANG_USE_MLX

合并时间: 2026-08-11 09:44

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34300>

执行摘要

- 一句话: 用 patch 固定 use_mlx, 修复 MLX CI 车道 profiler 测试误走策略
- 推荐动作: 值得快速浏览: 变更短小, 但“环境变量被 lru_cache 缓存打败、因此在调用点 mock”这一测试隔离模式有通用借鉴意义。关注点放在 setUp 的 patch 目标路径与 addCleanup 的组合用法, 以及 docstring 中记录的设计理由。

功能与动机

PR body 明确说明: 该修复针对 #34166 合并评论中点名的已知 CI 失败。

`apply_metal_profiler_patches()` 通过 `use_mlx()` 选择捕获策略, 而 `stage-a-unit-test-mlx` CI lane 在 job 级设置了 `SGLANG_USE_MLX=1`, 导致 `TestSchedulerProfilerManagerMPS` 每轮都静默运行 MLX 策略而非它要测试的 MPS 策略。`use_mlx()` 带有 `@lru_cache(maxsize=1)`, 进程内已缓存的值会让环境变量 `pin` 失效, 所以必须在 `profiler.py` 的调用点直接 mock。

实现拆解

1. 变更入口: 唯一修改文件是 `test/registered/unit/hardware_backend/mlx/test_metal_profiler.py`, 为 `TestSchedulerProfilerManagerMPS` 新增 `setUp` 方法。
2. 核心逻辑: 在 `setUp` 中通过 `unittest.mock.patch` 将 `sglang.srt.hardware_backend.mlx.profiler.use_mlx` 替换为固定返回 `False` 的桩, 并用 `self.addCleanup(patcher.stop)` 保证每个用例结束后自动恢复, 避免污染同进程内的其他测试。
3. 设计原因: patch 操作发生在 `use_mlx()` 的实际调用点 (模块内符号), 绕开了 `lru_cache` 的进程级缓存; 同时扩展了类 docstring, 记录“为什么不用环境变量 `pin`”这一决策, 供后续维护者理解。
4. 配套说明: sibling 类 `TestMetalCaptureProfilerMLX` 直接调用 `MetalCaptureProfiler.start_mlx`, 不会经过 `use_mlx()` 分发, 因此无需 `pin`。作者在 Apple M4 Pro 上用 4 种环境组合 (`SGLANG_USE_MLX=1`、未设置、`=0`、以及 CI lane 的同款调用方式) 验证均为 10 passed; 全量 MLX 单测套件在无环境变量覆盖下为 184 passed、4 skipped、0 failed, 与修复前基线一致。无配置、schema 或部署配套改动。

关键文件:

- `test/registered/unit/hardware_backend/mlx/test_metal_profiler.py` (模块 MLX 探针; 类别 test; 类型 test-coverage; 符号 `setUp`): 唯一变更文件: 为

TestSchedulerProfilerManagerMPS 新增 setUp, 在 profiler.py 调用点 mock use_mlx() 为 False, 隔离环境变量 SGLANG_USE_MLX 对测试策略选择的干扰, 是本次修复的核心与全部。

关键符号: setUp

关键源码片段

test/registered/unit/hardware_backend/mlx/test_metal_profiler.py

唯一变更文件: 为 TestSchedulerProfilerManagerMPS 新增 setUp, 在 profiler.py 调用点 mock use_mlx() 为 False, 隔离环境变量 SGLANG_USE_MLX 对测试策略选择的干扰, 是本次修复的核心与全部。

```
@unittest.skipUnless(_IS_APPLE_SILICON and _HAS_MLX, _SKIP_REASON)
class TestSchedulerProfilerManagerMPS(unittest.TestCase):
    """SchedulerProfilerManager._start_profile 处理 Metal 捕获失败。

    apply_metal_profiler_patches() 会依据 use_mlx() 选择捕获策略,
    而 use_mlx() 读取 SGLANG_USE_MLX 且被 lru_cache 缓存。
    本测试类专测 MPS 策略, 因此在 setUp 中直接 patch 调用点,
    而不是改环境变量 —— 环境变量 pin 无法覆盖已缓存的值。
    """

    def setUp(self):
        # profiler.py 中调用的是 `use_mlx()`, 所以 patch 模块内符号,
        # 强制走 MPS 策略; addCleanup 保证每个用例结束后自动恢复。
        patcher = patch(
            "sglang.srt.hardware_backend.mlx.profiler.use_mlx", return_value=False
        )
        patcher.start()
        self.addCleanup(patcher.stop)

    def _make_manager(self, output_dir):
        from sglang.srt.managers.scheduler_components.profiler_manager import (
            SchedulerProfilerManager,
        )

        class FakePS:
            tp_rank = dp_rank = pp_rank = moe_ep_rank = 0
            dp_size = pp_size = moe_ep_size = 1
            gpu_id = 0

        mgr = SchedulerProfilerManager(
            ps=FakePS(), dp_tp_cpu_group=None, get_forward_ct=lambda: 0
        )
        mgr._init_profile(output_dir, None, None, None, None, None, False, "test")
        return mgr
```

评论区精华

本 PR 没有实质性的 review 评论：alexnaills 直接 APPROVED（评论内容为空）；Issue 侧仅有 yeahdongcn 触发 /tag-and-rerun-ci 重跑 CI。核心设计决策——为什么 mock 调用点而不是改环境变量——已通过 setUp 上方扩展的 docstring 沉淀下来，没有在评论中展开讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
 - mock 路径依赖风险：当前 profiler.py 内部以模块内 use_mlx() 方式调用，patch 目标字符串正确；若未来改为 from tensor_bridge import use_mlx 直接导入符号，该 patch 会失效，需要同步更新测试。
 - 覆盖范围局限：该测试类现在被固定为永远验证 MPS 策略分支，MLX 策略分支依赖 sibling 测试覆盖；未来新增用例时需保持两个策略分支的覆盖对称。
 - 影响面可控：addCleanup 确保 patch 在每个用例后恢复，不会泄漏到其他测试；纯测试变更，无性能、安全、兼容性风险。
 - 影响：对用户与运行时系统零影响（test-only 变更）。对团队的影响主要是 CI 稳定性：修复 pr-test-mlx lane 每轮都会出现的 profiler 测试失败，减少合并时的噪音与误判。影响范围限定在 Apple Silicon / MLX 测试路径，影响程度较小但价值明确。
 - 风险标记：mock 目标路径依赖 profiler.py 内部符号，仅覆盖 MPS 策略分支，纯测试变更，无运行时风险

关联脉络

- PR #34166 [MLX] Window-bounded SWA KV storage and in-graph sampling: 本 PR 直接引用其合并评论中点名的已知 CI 失败；该 MLX 大 PR 引入了 profiler 相关路径，ambient SGLANG_USE_MLX 导致 TestSchedulerProfilerManagerMPS 静默跑错策略。
- PR #34301 Fix CI server warmup progress logging: 同为测试侧修复 CI 稳定性问题的近期 PR（修复 CI 禁用 tqdm 时预热进度误报），与本 PR 属于同一波 CI 噪音清理。
- PR #34272 [CI] Fix GSM8K floating-point tolerance boundary: 同为减少 CI 误失败的修复（浮点容差边界），与本 PR 一起反映仓库对 CI 稳定性的持续收敛。