PR #34017 完整报告

sgl-project/sglang

[Fix] Judge the phase-checker device-assert test by its FAIL line, not the exit code

合并时间：2026-08-08 07:19

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34017

执行摘要

- 一句话：修复 phase-checker 设备断言测试的 flaky 判定逻辑
- 推荐动作：值得快速浏览，了解 CUDA 设备端 assert 在 CI 环境下的非确定性退出行为。核心设计决策是“以证据行而非退出码作为断言标准”，对类似需要跨进程判断内核错误的测试有借鉴意义。无需精读，但可作为 CI 稳定性改进的参考范例。

功能与动机

PR body 明确指出：CI 运行在 SGLANG_CUDA_COREDUMP=1 下，coredump handler 可能 abort 进程；同时 torch.cuda.synchronize() 会抛出 cudaErrorAssert（"device-side assert triggered"）或泛化的 cudaErrorLaunchFailure（"unspecified launch failure"），子进程只匹配了第一种措辞并 exit 0，第二种走 exit 2，被断言拒绝。这直接导致 base-b-test-1-gpu-small (3) 在两个不同 runner 上失败，而同一验证内容在另一个 shard 通过，属于典型的 flaky 测试。

实现拆解

1. 根因分析

- 在 test/registered/utils/test_phase_checker.py 的 test_update_mismatch_after_enable_raises_in_subprocess 中，原逻辑要求子进程退出码为 (0, -6)。但 CI 的 coredump 机制会引入额外退出路径，且 torch.cuda.synchronize() 的错误措辞不唯一，导致断言不稳定。

2. 修改核心断言

- 保留 stdout 中 SimplePhaseChecker FAIL 行作为“内核侧 assert 已触发”的唯一证据。
- 将 assertIn(result.returncode, (0, -6)) 改为 assertNotEqual(result.returncode, 1)，因为退出码 1 表示子进程明确报告“没有抛 RuntimeError”，即 assert 从未触发。
- 失败信息中加入 returncode，便于后续诊断。

3. 清理相关测试文件

- test/registered/vlm/test_vision_openai_server_a.py：将通配符导入 from sglang.test.vlm_utils import * 替换为显式导入 DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH、DEFAULT_URL_FOR_TEST、popen_launch_server、IMAGE_MAN_IRONING_URL，并删除未使用的

OmniOpenAITestMixin 导入及删除列表中的对应项，避免通配符泄漏和隐式依赖。

- test/registered/models_e2e/test_qwen3_next_models.py：删除两个测试类中未使用的 gsm8k_accuracy_thres = 0.93 配置，保留 kl_div_thres 作为有效阈值。

4. 验证

- 通过 /rerun-test test/registered/utils/test_phase_checker.py 在 1-gpu-5090 runner 上重跑，结果 ?。
- 无产品代码、配置或部署配套改动，全部变更集中在测试目录。

关键文件：

- test/registered/utils/test_phase_checker.py（模块 阶段检查；类别 test；类型 test-stability；符号 test_update_mismatch_after_enable_raises_in_subprocess）：核心修改：将设备端 assert 触发的判据从子进程退出码改为 stdout 中的 FAIL 行，并保留退出码不为 1 的最低约束，修复 CI 上的 flaky 失败。
- test/registered/vlm/test_vision_openai_server_a.py（模块 视觉服务；类别 test；类型 test-maintenance）：将通配符导入改为显式导入，并删除未使用的 OmniOpenAITestMixin，避免通配符泄漏导致的隐式依赖。
- test/registered/models_e2e/test_qwen3_next_models.py（模块 模型端到端；类别 test；类型 test-maintenance）：删除两个测试类中未使用的 gsm8k_accuracy_thres 配置，精简测试配置，避免误导后人以为 GSM8K 精度被校验。

关键符号：test_update_mismatch_after_enable_raises_in_subprocess

关键源码片段

test/registered/utils/test_phase_checker.py

核心修改：将设备端 assert 触发的判据从子进程退出码改为 stdout 中的 FAIL 行，并保留退出码不为 1 的最低约束，修复 CI 上的 flaky 失败。

```
def test_update_mismatch_after_enable_raises_in_subprocess(self) -> None:
    """在子进程中运行，因为设备端 assert 会污染 CUDA 上下文。"""
    script = textwrap.dedent("""
        import sys
        import torch
        from sglang.srt.utils.phase_checker import SimplePhaseChecker

        device = torch.device("cuda:0")
        checker = SimplePhaseChecker(initial_phase=0, device=device)
        checker.enable_assert()
        # phase=0 但期望 phase=99 —— 内核必须触发 device_assert。
        checker.update(expect_phase=99, next_phase=1, caller_name="bad")
        try:
            torch.cuda.synchronize()
        except RuntimeError as e:
            msg = str(e).lower()
            if "device-side assert" in msg or "phase mismatch" in msg:
```

```
            sys.exit(0)
        print(f"Unexpected RuntimeError: {e}", file=sys.stderr)
        sys.exit(2)
    print("expected RuntimeError but none was raised", file=sys.stderr)
    sys.exit(1)
    """)
result = subprocess.run(
    [sys.executable, "-c", script],
    capture_output=True,
    text=True,
    timeout=180,
)
# FAIL 行是内核侧检查已触发的证据；进程如何退出则不是：
# CI 开启 SGLANG_CUDA_COREDUMP=1 时 coredump handler 可能 abort 进程，
# 而同步时可能抛出 cudaErrorAssert 或泛化的 cudaErrorLaunchFailure。
# 只有 exit code 为 1 才意味着 assert 从未触发。
self.assertIn(
    "SimplePhaseChecker FAIL",
    result.stdout,
    f"returncode {result.returncode}; "
    f"stdout: {result.stdout}\nstderr: {result.stderr}",
)
self.assertNotEqual(
    result.returncode,
    1,
    "the mismatch did not fire device_assert; "
    f"stdout: {result.stdout}\nstderr: {result.stderr}",
)
```

评论区精华

本 PR 没有 review 评论。作者在 PR body 中给出了完整的根因解释：同测试在不同 runner 上通过 / 失败的原因在于 coredump 生成是否发生，而 stdout 中的 FAIL 行始终存在，因此应以 FAIL 行为准而非退出码。commit 记录中提交了“trim comment”和清理死代码的独立提交，说明作者在合并 main 后主动清理了相关测试文件。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险是断言放宽可能引入漏报：如果 coredump handler 在打印 SimplePhaseChecker FAIL 行之前就 abort，stdout 将缺失该行，测试会误判为失败（false negative），但这种情况原本也会失败，因此可接受。另一个隐患是测试依赖 CI 环境变量 SGLANG_CUDA_COREDUMP 的行为，本地或非 CI 环境可能无法复现原始 flaky，但这不影响测试通过性。此外，若未来 stdout 输出策略变化（如缓冲导致 FAIL 行未 flush），同样可能导致误判，但目前没有迹象。
- 影响：影响范围限于 CI 测试稳定性：修复了 base-b-test-1-gpu-small (3) 等在 phase-checker 上的间歇性失败，降低开发者在 main 分支看见红色 CI 的概率。对用户无

感知，团队 CI 可靠性得到提升。清理导入和死代码也减少了测试文件对隐式通配符导入的依赖，提升可维护性。

- 风险标记：违反退出码断言放宽，额外依赖 FAIL 行 stdout 输出，CI 环境变量假设，清理导入可能影响其他测试

关联脉络

- 暂无明显关联 PR