PR #26616 完整报告

sgl-project/sglang

Let unittest._ShouldStop propagate through retry() so subTest+failfast works

合并时间：2026-05-29 07:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26616

执行摘要

- 一句话：修复 retry() 未传播 _ShouldStop 导致 CI 误报
- 推荐动作：建议精读。此 PR 虽小，但揭示了 unittest 内部 _ShouldStop 信号与重试装饰器交互的微妙问题，对理解 Python 测试框架和 CI 失败调试有参考价值。代码注释清晰，是良好的异常处理实践案例。

功能与动机

CI 中 attention-backend 单元测试套件在 -f (failfast) 模式下运行时，任何在 with self.subTest(...) 内的 self.skipTest() 调用（例如 FA3 硬件门控、DSA 的 tilelang/trtllm/aiter 能力门控、Mamba2 树验证、draft-extend 后端不可用等）都会导致 retry() 将 _ShouldStop 误判为可重试异常，最终报告为错误。PR body 详细描述了触发链：SkipTest → subTest.__exit__ 捕获并记录跳过 → 因 result.failfast 为 True 而抛出 _ShouldStop → retry() 未识别 → 重试并最终包装为 Exception。

实现拆解

1. 新增导入：在 python/sglang/srt/utils/common.py 中从 unittest.case 导入 _ShouldStop。
2. 异常链处理：在 retry() 函数的 except SkipTest 分支之后、except Exception 之前新增 except _ShouldStop: raise 分支，使该异常直接穿透重试逻辑，由 unittest 框架的 testPartExecutor 处理。
3. 注释说明：在新增分支处添加多行注释，解释 _ShouldStop 的语义（由 subTest.__exit__ 在 failfast 模式下抛出，表示测试方法应停止，不是错误，不应重试）。

关键文件：

- python/sglang/srt/utils/common.py（模块 工具函数；类别 source；类型 core-logic；符号 retry）：唯一修改文件；在 retry() 函数中添加对 _ShouldStop 异常的特殊处理，修复 CI failfast 模式下 subTest+skipTest 导致误报的 bug。

关键符号：retry

关键源码片段

python/sglang/srt/utils/common.py

唯一修改文件；在 retry() 函数中添加对 _ShouldStop 异常的特殊处理，修复 CI failfast 模式下 subTest+skipTest 导致误报的 bug。

```
# 新增导入
from unittest.case import _ShouldStop

# retry() 函数中的异常处理链
def retry(fn, max_retry, initial_delay=2.0, max_delay=60.0, should_retry=lambda e: True):
    for try_index in itertools.count():
        try:
            return fn()
        except SkipTest:
            # 不重试 skip 的测试，直接透传给 TestCase
            raise
        except _ShouldStop:
            # 新增：`_ShouldStop` 是 `unittest.case` 的私有信号，
            # 当 `subTest.__exit__` 捕获到 `SkipTest` 且 `result.failfast`
            # 为 True （CI 使用 `python3 file.py -f`）时抛出。
            # 它的作用是通知外层 `testPartExecutor` 停止当前测试方法，
            # 不应被重试，也不应被当作 error 报告。
            raise
        except Exception as e:
            traceback.print_exc()
            if try_index >= max_retry:
                raise Exception(f"retry() exceed maximum number of retries.")
            if not should_retry(e):
                raise Exception(f"retry() observe errors that should not be retried.")
            delay = min(initial_delay * (2**try_index), max_delay) * (
                0.75 + 0.25 * random.random()
            )
            logger.warning(
                f"retry() failed once ({try_index}th try, maximum {max_retry} retries). "
                f"Will delay {delay:.2f}s and retry. Error: {e}"
            )
            time.sleep(delay)
```

评论区精华

无 review 评论。PR 由作者自行审查合并。

- 暂无高价值评论线程

风险与影响

- 风险：变更极小（+8 行，仅控制流新增一个异常捕获分支），且异常类型为 unittest 内部私有类，外部不应抛出。风险极低。唯一潜在风险是如果未来 Python 版本移除或重命名 _ShouldStop，但可通过版本兼容性注释缓解。
- 影响：直接影响所有使用 retry() 的 CI 测试用例，特别是 attention-backend 单元测试套件（27 个文件）。修复后，-f 模式下 subTest 内 skipTest() 将正确报告为跳过而非错误，消

除 CI 误报。不影响非 unittest 场景下的 retry() 行为。

- 风险标记：极小变更

关联脉络

- PR #26302 [UnifiedTree] gate load back pre-evict on full-attn availability only: 间接关联：同样涉及 CI 测试套件的稳定性改进。