PR #36673 完整报告

sgl-project/sglang

[CI] Read subprocess stdout on a background thread to avoid EOF deadlock

合并时间：2026-08-29 01:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36673

执行摘要

- 一句话：修复 CI 子进程 stdout 读取 EOF 死锁
- 推荐动作：该 PR 值得快速浏览以了解 CI 死锁修复模式，尤其是后台线程处理子进程输出的技巧。但技术深度有限，不涉及核心推理逻辑，可跳过精读。关注点：join(timeout) 的合理性，以及是否需要在极端情况下处理 stdout 不关闭的问题。

功能与动机

PR 描述指出：run_unittest_files 使用阻塞的 for line in process.stdout 循环读取子进程输出，然后才调用 process.wait()。当子进程输出足够多填满管道缓冲区时，该循环可能在 EOF 上无限阻塞，导致死锁。该修复旨在消除 CI 中的潜在挂起。

实现拆解

1. 定位问题：在 python/sglang/test/ci/ci_utils.py 的 run_one_file 函数（capture_output=True 分支）中，原先用 for line in process.stdout 同步读取输出，再 process.wait()，存在 EOF 死锁风险。
2. 引入后台读取线程：新增内部函数 read_output，将逐行读取循环移入其中，并通过 threading.Thread(target=read_output, daemon=True) 创建后台守护线程并启动。
3. 调整主流程：先 process.wait() 等待子进程结束，再 reader_thread.join(timeout=60) 限时等待读取线程收尾，确保主线程不会无限阻塞。
4. 配套影响：仅改动 CI 测试工具函数，未涉及配置、部署或测试用例本身；由于是死锁修复，未新增单元测试，依赖现有 CI 流程验证。

关键文件：

- python/sglang/test/ci/ci_utils.py（模块 CI 工具；类别 test；类型 test-coverage；符号 read_output）：核心变更文件：修复 CI 子进程 stdout 读取死锁，将阻塞读取移至后台线程并超时 join。

关键符号：read_output

关键源码片段

python/sglang/test/ci/ci_utils.py

核心变更文件：修复 CI 子进程 stdout 读取死锁，将阻塞读取移至后台线程并超时 join。

```
# python/sglang/test/ci/ci_utils.py

def run_one_file(filename, capture_output=False):
    nonlocal process, output_lines
    # ... 省略前置代码 ...
    if capture_output:
        # 创建子进程，合并 stdout/stderr 到管道
        process = subprocess.Popen(
            cmd,
            stdout=subprocess.PIPE,
            stderr=subprocess.STDOUT,
            text=True,
            errors="ignore", # 忽略非 UTF-8 字节，防止 UnicodeDecodeError
        )
        output_lines = []

        # 后台线程持续读取 stdout，避免主线程在管道缓冲满时陷入 EOF 死锁。
        def read_output():
            for line in process.stdout:
                logger.info(line.rstrip())
                output_lines.append(line)

        # 使用 daemon 线程，确保即使主线程异常退出，线程也不会阻止进程结束。
        reader_thread = threading.Thread(target=read_output, daemon=True)
        reader_thread.start()
        process.wait() # 等待子进程结束
        # 限时等待读取线程收尾，最多 60 秒，防止线程永久阻塞。
        reader_thread.join(timeout=60)
    else:
        process = subprocess.Popen(cmd, stdout=None, stderr=None)
        process.wait()
    # ... 省略后续代码 ...
```

评论区精华

该 PR 无 review 评论。仅有一条维护者 iforgetmyname 的评论 /tag-run-ci-label 用于触发 CI。审核人 iforgetmyname 直接批准（APPROVED），未提出异议。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 读取不完整风险：reader_thread.join(timeout=60) 若超时，可能丢弃子进程部分输出，影响重试决策和日志完整性，但概率低（正常子进程结束后 stdout 应立即 EOF）。
  2. 线程安全：output_lines 由后台线程追加，主线程仅在 join 后读取，无并发访问，但若未来主线程在 join 前访问则有风险。

3. 死锁消除不彻底：若 process.stdout 在子进程结束后仍不 EOF（极端情况），连接线程仍可能阻塞在 read_output 内，但 daemon 线程不会阻止进程退出。
4. 影响范围：仅 CI 工具，不涉及生产路径，风险低。 - 影响：影响用户 / 开发者：CI 稳定性提升，避免因死锁导致的挂起和超时，减少 CI 重试。影响系统：改善 run_unittest_files 在输出较大时的可靠性。影响团队：维护者需了解后台线程机制，但代码改动小，易于理解。影响程度较小，限于 CI 测试运行器。 - 风险标记：CI 稳定性，后台线程超时可能丢输出

关联脉络

- PR #36434 [AMD] Add ROCm 10 (gfx942 / gfx950) release images: 同为 CI/ 基础设施相关 PR，调整 CI 流程，可能受影响 CI 稳定性相关改动。
- PR #36308 [AMD][CI] Limit HiCache MGSM eval concurrency on ROCm: 同为 CI 测试并发 / 稳定性调整的 PR，与本 PR 同属 CI 可靠性改进方向。