

# PR #7405 完整报告

PaddlePaddle/FastDeploy

[CI] Add pytest failure log collection and persistence

合并时间: 2026-04-16 22:56

原文链接: <http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7405>

## 执行摘要

- 一句话: 为 CI 添加 pytest 失败日志收集与持久化功能, 提升调试效率。
- 推荐动作: 该 PR 值得快速浏览, 重点关注 `pytest_runtest_makereport` 钩子的实现方式, 以及文件名清洗和导入结构调整的设计决策。对于 CI 基础设施维护者, 可借鉴其日志收集机制以优化其他项目的测试调试流程。

## 功能与动机

根据 PR body 描述, 动机是“增强 CI 调试能力, 通过收集详细的 pytest 失败日志, 减少不稳定或失败用例的排查时间”。这旨在解决 CI 环境中测试失败时调试困难的问题, 提升开发和维护效率。

## 实现拆解

1. 新增 pytest 钩子实现日志收集: 在 `tests/conftest.py` 中新增 `pytest_runtest_makereport` 钩子, 捕获测试失败时的错误信息, 并写入 `FD_LOG_DIR` 目录下的日志文件, 文件名基于测试用例名生成。
2. 调整 `conftest.py` 导入和结构: 将 `e2e.utils.serving_utils` 导入移至文件顶部, 并添加日志目录创建逻辑到 `pytest_configure` 中, 同时重构代码注释以提升可读性。
3. 禁用不稳定测试文件: 将 `tests/e2e/4cards_cases/test_Qwen3_30b_tp4.py` 重命名为 `_test_Qwen3_30b_tp4.py`, 以禁用该不稳定测试。
4. 更新 CI 脚本排除干扰: 修改 `scripts/coverage_run.sh`, 在 `grep` 命令中添加 `--exclude="pytest*_error.log"` 选项, 避免日志收集产生的错误日志干扰 CI 输出。

关键文件:

- `tests/conftest.py` (模块 测试框架; 类别 `test`; 类型 `test-coverage`; 符号 `pytest_configure`, `pytest_collection_modifyitems`, `pytest_runtest_makereport`, `FDRunner`): 核心变更文件, 新增 pytest 钩子实现日志收集, 并调整导入和配置逻辑。
- `tests/e2e/4cards_cases/_test_Qwen3_30b_tp4.py` (模块 端到端测试; 类别 `test`; 类型 `rename-or-move`): 通过重命名禁用不稳定的测试文件, 避免 CI 失败干扰。
- `scripts/coverage_run.sh` (模块 脚本工具; 类别 `infra`; 类型 `configuration`): 更新 CI 脚本以排除 pytest 错误日志, 避免干扰 CI 输出。

关键符号: `pytest_runtest_makereport`, `pytest_configure`, `pytest_collection_modifyitems`

## 关键源码片段

### tests/confstest.py

核心变更文件，新增 pytest 钩子实现日志收集，并调整导入和配置逻辑。

```
import os
import re
import pytest

@pytest.hookimpl(tryfirst=True, hookwrapper=True)
def pytest_runtest_makereport(item, call):
    """
    捕获失败测试用例并将错误日志保存到FD_LOG_DIR。
    仅记录测试执行阶段的失败，不包含setup/teardown阶段。
    """

    outcome = yield
    report = outcome.get_result()

    if report.when == "call" and report.failed:
        # 从item.nodeid提取测试用例名，并清洗非法字符以确保文件名安全
        raw_name = item.nodeid.split("::", 1)[-1]
        case_name = re.sub(r'[\V:*?"<>|]', '_', raw_name) # 替换特殊字符为下划线

        log_dir = os.environ.get("FD_LOG_DIR", "log")
        log_file = os.path.join(log_dir, f"pytest_{case_name}_error.log")

        # 将错误信息写入日志文件
        with open(log_file, "w", encoding="utf-8") as f:
            f.write(f"Test failed: {item.nodeid}\n")
            if call.excinfo is not None:
                f.write(f"Exception: {call.excinfo}\n")
            if report.longrepr is not None:
                f.write(f"Traceback: {report.longrepr}\n")
```

## 评论区精华

1. 文件名安全性问题：PaddlePaddle-bot 指出 case\_name 未做安全清洗，参数化测试中含 / 等字符可能导致日志写入失败，建议添加清理逻辑。作者在后续提交中通过正则表达式修复了此问题。
  2. 导入风险疑问：PaddlePaddle-bot 询问将 e2e.utils.serving\_utils 提升为顶层导入是否会导致非 e2e 测试的 collection 失败，但未得到明确回复，此疑虑未解决。
  3. 脚本选项位置建议：PaddlePaddle-bot 建议将 --exclude 选项放在位置参数前以避免 POSIX 兼容性问题，但 PR 中未采纳此建议。
  4. 临时测试修改：Review 中发现了多个测试文件中的断言被反转为 !=，这是用于验证日志收集功能的临时修改，作者在后续提交中移除了这些调试代码。
- 文件名安全性清洗 (correctness): 作者在后续提交中添加了正则表达式清洗逻辑，替换非法字符为下划线。

- 顶层导入风险 (design): 未得到明确回复, 疑虑未解决, 可能依赖环境配置。
- 脚本选项位置 (style): PR 中未采纳此建议, 选项仍放在位置参数后。

## 风险与影响

- 风险: 1. 导入依赖风险: `conftest.py` 中顶层导入 `e2e.utils.serving_utils`, 若该模块在非 `e2e` 测试环境中不可用, 可能导致所有测试的 `collection` 阶段失败。 2. 日志文件路径安全: 尽管已添加文件名清洗, 但 `case_name` 清洗逻辑可能未覆盖所有边缘情况, 如过长文件名或特殊编码字符, 仍可能引发文件操作异常。 3. CI 脚本兼容性: `coverage_run.sh` 中 `--exclude` 选项放在位置参数后, 在严格 POSIX 环境下可能失效, 导致错误日志未被排除, 干扰 CI 输出。 4. 测试覆盖影响: 禁用 `test_Qwen3_30b_tp4.py` 可能降低对该模型特定配置的测试覆盖, 需确保其他测试能补偿。
- 影响: 1. 对用户影响: 无直接影响, 主要面向开发者和 CI 维护者, 提升测试失败时的调试效率。 2. 对系统影响: 新增日志文件可能增加磁盘使用量, 但通过环境变量 `FD_LOG_DIR` 可控制目录, 影响有限。 3. 对团队影响: 简化 CI 故障排查流程, 减少手动日志收集时间, 提升开发迭代速度。
- 风险标记: 导入依赖风险, 文件路径安全, 脚本兼容性

## 关联脉络

- PR #7429 [CI] Add approval check for logging-related modifications: 同属 CI 和日志相关改进, 涉及脚本和审批流程优化。
- PR #7190 [Feature] implement log channel separation and request log level system: 涉及日志系统增强, 本 PR 的日志收集机制可视为其补充。