

PR #7429 完整报告

PaddlePaddle/FastDeploy

[CI] Add approval check for logging-related modifications

合并时间: 2026-04-16 14:50

原文链接: <http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7429>

执行摘要

- 一句话: 在 CI 审批检查脚本中新增日志相关修改的审批检测逻辑。
- 推荐动作: 该 PR 是典型的 CI 流程优化, 适合基础设施维护者精读以了解审批检查机制。值得关注的设计决策包括: 通过 git diff 过滤实现精准检测、排除脚本自身修改避免循环触发、以及如何平衡检测覆盖与误报风险。对于普通开发者, 了解此变更可避免在修改日志代码时意外触发审批要求。

功能与动机

根据 PR body 说明, 动机是确保对日志行为的修改得到适当审查, 因为日志变更 (如 `.info/.debug/.error/log_request`) 可能影响调试、可观测性和系统稳定性。

实现拆解

1. 定义日志关键词数组: 在 `scripts/check_approval.sh` 中新增 `LOG_KEYWORDS` 数组, 包含 `.info()`、`.debug()`、`.error()`、`log_request()`、`log_request_error` 五个模式, 用于匹配日志调用。
2. 检测日志修改: 通过 `git diff` 命令获取与上游分支的差异, 过滤出新增行 (`grep -E "\+"`), 并排除差异头 (`grep -vE "\+\+"`), 然后使用 `grep -E` 结合关键词数组检测是否包含日志相关修改。同时通过 `-- ':exclude)scripts/check_approval.sh'` 排除了脚本自身修改, 避免自我触发检测。
3. 触发审批检查: 如果检测到日志修改且 PR ID 存在, 则输出提示信息并调用 `check_approval` 函数, 要求指定 RD (`xyxinyang`、`zyyzghb`) 审批。
4. 测试与调试: 提交历史显示作者曾添加临时测试文件 `tests/test_log_check_tmp.py` 进行验证, 但根据 review 建议, 该文件在最终提交前被移除, 未包含在合并版本中。

关键文件:

- `scripts/check_approval.sh` (模块 CI 脚本; 类别 `infra`; 类型 `core-logic`): 这是唯一被修改的文件, 实现了日志修改检测的核心逻辑, 直接影响 CI 审批流程。

关键符号: 未识别

关键源码片段

`scripts/check_approval.sh`

这是唯一被修改的文件，实现了日志修改检测的核心逻辑，直接影响 CI 审批流程。

```
# 定义日志相关关键词数组，用于匹配常见的日志调用模式
LOG_KEYWORDS=(
    "\.info\" # 匹配 .info() 调用
    "\.debug\" # 匹配 .debug() 调用
    "\.error\" # 匹配 .error() 调用
    "log_request\" # 匹配 log_request() 调用
    "log_request_error" # 匹配 log_request_error 字符串（注意：代码库中当前不存在此方法）
)

# 检测PR中是否包含日志相关修改
# 使用git diff获取与上游分支的差异，排除脚本自身修改（避免自我触发）
HAS_LOG_MODIFY=$(git diff upstream/$BRANCH \
    -- . ': (exclude)scripts/check_approval.sh' \
    | grep -E "^+" \
    | grep -vE "^+\\+\\+" \
    | grep -E $(printf -- "%s" "${LOG_KEYWORDS[@]}" | sed 's/|$/|/' ) || true)

# 如果检测到日志修改且PR ID存在，则触发审批检查
if [ -n "${HAS_LOG_MODIFY}" ] && [ -n "${PR_ID}" ]; then
    # 输出提示信息，说明需要RD审批
    echo_line1="You must have one FastDeploy RD (xyxinyang(zhouchong),
    zyyzghb(zhangyongyue)) approval for modifying logging behavior (.info/.debug/.error/log_
    request).\"
    # 调用check_approval函数，传入提示信息和审批者列表
    check_approval "$echo_line1" 1 xyxinyang zyyzghb
fi
```

评论区精华

1. 关键词准确性争议：AI reviewer 指出 `log_request_error` 在代码库中不存在（只有 `log_request` 方法），建议确认是否需要检测或移除。讨论中未明确回应，但最终代码保留了该关键词。
 2. 测试覆盖建议：AI reviewer 建议添加测试用例验证检测逻辑，包括真实日志调用、删除、注释和字符串场景的准确性。作者未直接回应，但通过临时测试文件进行了调试。
 3. 自我触发风险：AI reviewer 指出脚本自身修改可能误触发检测，建议改进过滤逻辑。作者在最终实现中通过 `-- . ': (exclude)scripts/check_approval.sh'` 排除了脚本文件，解决了此问题。
 4. 临时文件处理：AI reviewer 发现提交的临时测试文件 `tests/test_log_check_tmp.py` 不应提交，建议删除。最终提交中该文件已被移除，表明作者采纳了建议。
- 日志关键词准确性 (correctness): 未明确回应，但最终代码保留了该关键词，可能存在冗余。
 - 测试覆盖 (testing): 作者通过临时测试文件调试，但最终未提交正式测试，测试覆盖不足。
 - 自我触发风险 (design): 作者通过 `-- . ': (exclude)scripts/check_approval.sh'` 排除了脚本文件，解决了此问题。
 - 临时文件处理 (style): 作者在最终提交中移除了该文件，采纳了建议。

风险与影响

- 风险：1. 误报风险：关键词 `log_request_error` 在代码库中不存在，可能导致检测逻辑冗余或未来误报，但当前影响有限。 2. 漏报风险：检测逻辑依赖 `git diff` 和 `grep` 模式匹配，若日志调用格式变化（如换行、空格差异）可能漏检，但现有模式覆盖了常见调用。 3. 维护风险：CI 脚本变更可能影响所有 PR 的审批流程，若脚本逻辑错误（如过滤条件不当）可能导致审批要求误触发或缺失。 4. 兼容性风险：无，纯 CI 脚本变更不影响运行时功能。
- 影响：1. 对团队流程影响：所有涉及日志相关代码修改的 PR 现在需要指定 RD 审批，增加了变更控制粒度，可能略微延长合并周期，但提升了代码质量保障。 2. 对系统影响：仅影响 CI 流程，不改变 FastDeploy 核心功能、性能或安全性。 3. 对用户影响：无直接影响，用户不可见。
- 风险标记：关键词冗余，测试覆盖不足

关联脉络

- PR #7190 [Feature] implement log channel separation and request log level system: 该 PR 涉及日志系统的重大功能变更，与本 PR 的日志修改审批检测直接相关，体现了对日志变更加强管控的背景。