

PR #32735 完整报告

sgl-project/sglang

[CI] Fail lint when a registered file's TestCase classes never run

合并时间: 2026-07-29 15:45

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32735>

执行摘要

- 一句话: 防止注册测试文件中的死测试类
- 推荐动作: 值得合并, 修复了潜在的测试静默跳过问题, 提升了 CI 可靠性。变更简单、针对性明确。

功能与动机

PR 描述指出: CI 以 `python3 file.py` 执行注册文件, 导致自定义 `__main__` 若包含 `argparse` CLI 且未正确处理测试运行, `TestCase` 类永远不会实际执行, 文件却一直显示绿色。需要增加 lint 检测并修复唯一的违规文件。

实现拆解

1. 新增静态检查函数: 在 `scripts/ci/check_registered_tests.py` 中新增 `_defines_testcase(tree)` 函数, 通过 AST 遍历判断文件是否定义继承 `TestCase` 的类 (包括 `type()` 动态创建); 新增 `_main_runs_tests(tree)` 函数, 检查 `if __name__ == "__main__"` 块是否包含 `unittest.main()` 或 `pytest.main()` 调用。
2. 集成到检查流程: 在现有 `main()` 函数的文件遍历循环中, 对每个注册文件进行 AST 分析; 若文件定义了 `TestCase` 类但 `__main__` 未运行测试, 则收集到 `dead_tests` 列表; 最后如果有死测试文件, 打印错误信息并设 `exit_code=1`。
3. 修复违规测试文件: 修改 `test/registered/unit/mem_cache/test_unified_radix_cache_bench.py`, 将原直接的 CLI 参数解析和逻辑迁移到 `_run_bench_cli()` 函数中; 在 `if __name__ == "__main__"` 中判断 `--bench` 参数: 若存在则调用 `_run_bench_cli()` 运行基准测试, 否则调用 `unittest.main()` 执行 `TestCase` 类。同时修补了 `make_req()` 函数, 增加了 `req.kv = ReqKvInfo(...)` 赋值以兼容 #29427 引入的 `req.kv` 字段。

关键文件:

- `scripts/ci/check_registered_tests.py` (模块 CI 脚本; 类别 `infra`; 类型 `infrastructure`; 符号 `_defines_testcase`, `_main_runs_tests`): 核心 lint 检查逻辑, 新增 AST 静态分析检测死测试类。
- `test/registered/unit/mem_cache/test_unified_radix_cache_bench.py` (模块 测试; 类别 `test`; 类型 `test-coverage`; 符号 `_run_bench_cli`): 修复了 15 个测试静默跳过的问题, 并添加 `req.kv` 兼容性补丁。

关键符号: `_defines_testcase`, `_main_runs_tests`, `_run_bench_cli`

关键源码片段

scripts/ci/check_registered_tests.py

核心 lint 检查逻辑，新增 AST 静态分析检测死测试类。

```
# scripts/ci/check_registered_tests.py
import ast
# ... (other imports) ...

def _defines_testcase(tree: ast.AST) -> bool:
    """True if the file defines unittest classes, statically or via type()."""
    for node in ast.walk(tree):
        if isinstance(node, ast.ClassDef):
            # Check if any base class contains "TestCase"
            if any("TestCase" in ast.unparse(b) for b in node.bases):
                return True
        elif isinstance(node, ast.Call):
            # Handle dynamic class creation: type("Name", (Base,), {...})
            if (isinstance(node.func, ast.Name) and node.func.id == "type"
                and len(node.args) >= 2
                and isinstance(node.args[1], ast.Tuple)
                and any("TestCase" in ast.unparse(e) for e in node.args[1].elts)):
                return True
    return False

def _main_runs_tests(tree: ast.Module) -> bool:
    """Check if __main__ block calls unittest.main() or pytest.main()."""
    for stmt in tree.body:
        if not (isinstance(stmt, ast.If)
            and ast.unparse(stmt.test).replace("'", '"') == '__name__ == "__main__"'):
            continue
        body = ast.unparse(ast.Module(body=stmt.body, type_ignores=[]))
        if "unittest.main" in body or "pytest.main" in body:
            return True
    return False

def main() -> int:
    # ... (existing code) ...
    dead_tests = [] # (file) -- TestCase classes that `python3 file.py` never runs
    for f in files:
        # ... existing registration parsing ...
        with open(f, "r", encoding="utf-8") as fh:
            tree = ast.parse(fh.read(), filename=f)
        if _defines_testcase(tree) and not _main_runs_tests(tree):
            dead_tests.append(f)
    # ... rest of parsing ...
    if dead_tests:
        print("ERROR: Test file(s) define TestCase classes that CI never runs: ...")
        for f in dead_tests:
```

```
        print(f" {f}")
    exit_code = 1
    return exit_code
```

test/registered/unit/mem_cache/test_unified_radix_cache_bench.py

修复了 15 个测试静默跳过的问题，并添加 req.kv 兼容性补丁。

```
# test/registered/unit/mem_cache/test_unified_radix_cache_bench.py
# ... (earlier part unchanged) ...

def _run_bench_cli():
    parser = argparse.ArgumentParser(description="UnifiedRadixCache benchmark")
    parser.add_argument("--num-seqs", type=int, default=5000)
    # ... other arguments ...
    args = parser.parse_args()
    # ... run benchmark logic ...

if __name__ == "__main__":
    # CI runs `python3 file.py`; it must execute the TestBench_* classes
    if "--bench" in sys.argv:
        sys.argv.remove("--bench")
        _run_bench_cli()
    else:
        unittest.main()
```

评论区精华

无 review 评论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。修改仅影响 CI 检查脚本和单个已确认有问题的测试文件。修复的 req.kv 赋值是 fabricate req 场景下的兼容补丁，不会影响正常流程。
- 影响：CI 流程：lint 阶段现在会检测所有注册测试文件，确保 TestCase 类能被 CI 实际执行。测试修复：test_unified_radix_cache_bench.py 的 15 个 TestBench_* 测试现在会在 CI 中正常执行，提高了 UnifiedRadixCache 的测试覆盖率。
- 风险标记：暂无

关联脉络

- PR #29427 Introduce req.kv container for coupled owned kv field lifecycle: 本 PR 修复的测试文件中需要补充 req.kv = ReqKvInfo(...) 以兼容该 PR 引入的新字段。