PR #25248 完整报告

sgl-project/sglang

[CI] Support new-style `register_cuda_ci(stage=, runner_config=)` in slash handler + est-time updater

合并时间：2026-05-14 14:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25248

执行摘要

- 一句话：修复新式 register_cuda_ci 不被识别
- 推荐动作：建议合并后关注后续是否有遗漏的新式参数变体（如参数顺序不一致），可考虑未来将 _extract_suite 进一步抽象为更通用的参数解析器。此 PR 适合 CI/Infra 团队精读，对理解 CI 注册机制的演进有参考价值。

功能与动机

224 个已注册的测试文件已改用新式 register_cuda_ci(..., stage="...", runner_config="...") 写法，但 slash command handler 和 est-time updater 的正则只匹配旧式 suite="..."，导致 /rerun-test <file> 报错 No register_cuda_ci()/register_cpu_ci() found（例如 test_hicache_storage_file_backend.py），且 update_est_time.py 静默跳过这些文件的 est_time 更新。

实现拆解

1. 抽取公共解析函数 _extract_suite：在 scripts/ci/utils/slash_command_handler.py 中新增该函数，先尝试匹配旧式 suite="..."，若不匹配则提取参数，再从 stage 和 runner_config 组合出 "{stage}-test-{runner_config}" 格式的 suite。
2. 重写 detect_suite 调用点：将原内联的正则匹配替换为调用 _extract_suite(content, "register_cuda_ci") 和 _extract_suite(content, "register_cpu_ci")，消除代码重复。
3. 同步 update_est_time.py 的匹配逻辑：增加对新式格式的 fallback 匹配：在旧式正则失败后，如果 suite 包含 -test- 分隔符，则构建新式正则并检查是否匹配；若匹配则使用新式 pattern 进行替换。

关键文件：

- scripts/ci/utils/slash_command_handler.py（模块 CI 脚本；类别 infra；类型 infrastructure；符号 _extract_suite, detect_suite）：新增 _extract_suite 函数，重构 detect_suite 使其同时支持新旧两种 register_cuda_ci / register_cpu_ci 写法，是 PR 的核心改动文件。
- scripts/ci/update_est_time.py（模块 CI 脚本；类别 infra；类型 infrastructure）：增加对新式 register_cuda_ci 的 est_time 更新支持，确保该脚本不会静默跳过使用新写法的测试文件。

关键符号：_extract_suite, detect_suite

关键源码片段

scripts/ci/utils/slash_command_handler.py

新增 _extract_suite 函数，重构 detect_suite 使其同时支持新旧两种 register_cuda_ci / register_cpu_ci 写法，是 PR 的核心改动文件。

```
# scripts/ci/utils/slash_command_handler.py

def _extract_suite(content, func_name):
    """Pull a suite name out of a `register_{cuda,cpu}_ci(...)` call.

    Two styles are supported:
    1. legacy: register_cuda_ci(..., suite="stage-X-test-Y")
    2. new: register_cuda_ci(..., stage="stage-X", runner_config="Y")
       -> suite = f"{stage}-test-{runner_config}"
    """
    # 先尝试匹配旧式写法
    legacy = re.search(
        rf'^[^#\n]*{func_name}\([^)]*suite\s*=\s*["\']([^"\']+)["\']',
        content,
        re.MULTILINE,
    )
    if legacy:
        return legacy.group(1)
    # 若失败，尝试匹配新式写法并提取 stage 和 runner_config
    args = re.search(rf"^[^#\n]*{func_name}\(([^)]*)\)", content, re.MULTILINE)
    if args:
        stage_m = re.search(r'stage\s*=\s*["\']([^"\']+)["\']', args.group(1))
        rc_m = re.search(r'runner_config\s*=\s*["\']([^"\']+)["\']', args.group(1))
        if stage_m and rc_m:
            return f"{stage_m.group(1)}-test-{rc_m.group(1)}"
    return None

def detect_suite(file_path_from_test):
    # ... 读取文件内容后，调用 _extract_suite 替代内联正则
    suite = _extract_suite(content, "register_cuda_ci")
    if suite:
        runner = CUDA_SUITE_TO_RUNNER.get(suite)
        # ... 后续处理
    suite = _extract_suite(content, "register_cpu_ci")
    if suite:
        return suite, "ubuntu-latest", False, True, None
```

scripts/ci/update_est_time.py

增加对新式 register_cuda_ci 的 est_time 更新支持，确保该脚本不会静默跳过使用新写法的测试文件。

```
# scripts/ci/update_est_time.py（相关片段）

# 先尝试旧式正则匹配
legacy_pattern = re.compile(
    rf"(register_{backend}_ci\(est_time=)(\d+)"
    rf'(,\s*suite="{re.escape(suite)}")'
)
pattern = legacy_pattern if legacy_pattern.search(new_content) else None

# 若旧式不匹配且 suite 符合新式格式，则构造新式正则
if pattern is None and "-test-" in suite:
    stage, _, rc = suite.partition("-test-")
    new_style_pattern = re.compile(
        rf"(register_{backend}_ci\(est_time=)(\d+)"
        rf'(,\s*stage="{re.escape(stage)}",\s*runner_config="{re.escape(rc)}")'
    )
    if new_style_pattern.search(new_content):
        pattern = new_style_pattern

if pattern is None:
    continue # 跳过不匹配的文件
```

评论区精华

该 PR 无 review 评论，仅作者使用 /rerun-test 触发了一次 CI 验证（test_hicache_storage_file_backend.py）并成功执行。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更集中在两个 CI 基础设施脚本的套件提取逻辑，不涉及核心运行时。_extract_suite 的 fallback 逻辑确保对旧格式完全向后兼容；新格式的 suite 组合方式与现有 CUDA_SUITE_TO_RUNNER 字典中的 key 格式一致（如 stage-c-test-4-gpu-h100），因此 detect_suite 中获取 runner 的映射不会中断。存在轻微风险：若某些文件混用 stage/runner_config 顺序或存在非标准写法，可能导致组合错误，但已有 CI 验证覆盖。
- 影响：影响范围：所有使用新式 register_cuda_ci 的 224 个测试文件。对用户：开发者现在可以通过 /rerun-test <file> 重新运行这些文件的 CI，不会收到错误提示。对系统：est_time 自动更新脚本能正确更新新式文件的预估时间。影响程度：中等，解决了 CI 工具链中长期存在的兼容性问题。
- 风险标记：测试文件兼容性，静默跳过风险

关联脉络

- PR #25255 ci: read est_time from sglang-ci-stats instead of scraping CI logs: 同样修改了 scripts/ci/update_est_time.py，属于同一 CI 基础设施改进方向。

- PR #25138 ci: extract cuda stage actions + runner_config mapping: 引入了 stage/runner_config 的概念，是新式注册格式的背景 PR。