

PR #28762 完整报告

sgl-project/sglang

[diffusion] CI: refactor CI

合并时间: 2026-06-27 19:22

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28762>

执行摘要

- 一句话: 重构 diffusion CI, 抽取公共 runner 并简化测试组织
- 推荐动作: 值得精读, 尤其关注如何通过抽取公共模块消除重复代码。测试基础设施重构的套路可以复用。建议团队在合并后密切监控 NPU/AMD CI, 并确保文档更新反映新的文件组织结构。

功能与动机

原始 diffusion 测试代码中存在大量重复的 pytest 运行逻辑 (如 `collect_test_items`、`parse_junit_xml` 等), 分布于 `run_suite.py`、`run_suite_musa.py` 等多个文件中, 不利于维护。同时, 部分测试文件组织混乱, 测试用例设计冗余。本次重构旨在统一测试基础设施, 提高可维护性和可扩展性, 并为后续平台 (NPU/AMD) 的 CI 集成打下基础。

实现拆解

1. 创建公共 runner: 在 `python/sglang/multimodal_gen/test/runner/pytest_runner.py` 中汇集了所有与 pytest 运行相关的工具函数, 包括 `collect_test_items`、`parse_junit_xml_for_executed_cases`、`parse_junit_xml_for_case_results`、以及内部重试辅助函数 `_run_pytest_attempt`、`_extract_collection_line`、`_extract_short_test_summary`、`_extract_failure_tail`、`_summary_has_retryable_failure`。
2. 删除 musa 专用 runner: 删除了独立的 `run_suite_musa.py` (270 行), 新增 `server/musa/run_suite.py` (172 行), 该文件复用公共 runner, 实现了相同的套件分区执行逻辑, 但代码量大幅减少。
3. 简化主 runner: 修改 `run_suite.py`, 将其内联的 `collect_test_items`、`parse_junit_xml_for_executed_cases`、`parse_junit_xml_for_case_results` 等函数定义删除 (约 357 行), 改为从 `runner.pytest_runner` 导入 `partition_items_by_index` 和 `run_pytest`。此外, 将 `write_execution_report` 和 `run_component_accuracy_files` 等逻辑保留在原地但进行了精简。
4. 测试文件重组织: 将 `test_update_weights_from_disk.py` 从 `server/` 移动到 `single_test_file/`, 并将原来的三个单独测试 (`nonexistent_model`、`missing_model_path`、`nonexistent_module`) 合并为一个参数化的 `test_update_weights_rejects_invalid_requests`, 同时更新了文档注释。类似地, `cli_generate_common.py` 从 `cli/` 移动到 `single_test_file/`, `test_consistency_metrics.py` 从根测试目录移动到 `unit/`。

5. 清理旧文件：删除了不再使用的 `cli/test_generate_t2i_perf.py` (23 行)，并更新了 `.github/workflows` 中的 CI 配置以反映新的运行器路径。
6. 基准数据更新：由于移除了 LTX2 的 snapshot 模式 (改为弃用别名)，重新生成了 `ltx_2_3_hq` 的一致性测试基准和性能基准，将 CI 的 GT 快照指向新生成的原始模式结果。

关键文件：

- `python/sglang/multimodal_gen/test/runner/pytest_runner.py` (模块 测试运行器；类别 test；类型 test-coverage；符号 `collect_test_items`, `parse_junit_xml_for_executed_cases`, `parse_junit_xml_for_case_results`, `_run_pytest_attempt`)：核心新增文件，抽取了所有公共 pytest 运行逻辑，是整个重构的基石。
- `python/sglang/multimodal_gen/test/run_suite_musa.py` (模块 MUSA 运行器；类别 test；类型 deletion；符号 `parse_args`, `collect_test_items`, `run_pytest`, `main`)：被删除的独立 musa runner，体现了代码冗余的消除。
- `python/sglang/multimodal_gen/test/run_suite.py` (模块 主运行器；类别 test；类型 test-coverage；符号 `collect_test_items`, `parse_junit_xml_for_executed_cases`, `parse_junit_xml_for_case_results`, `_run_pytest_attempt`)：主运行器，删除了 357 行重复定义，改为导入公共模块，是重构的核心体现。
- `python/sglang/multimodal_gen/test/server/musa/run_suite.py` (模块 MUSA 运行器；类别 test；类型 test-coverage；符号 `parse_args`, `_resolve_suite_files`, `main`)：新增的 musa runner，替代删除的 `run_suite_musa.py`，代码量精简且复用公共 runner。
- `python/sglang/multimodal_gen/test/single_test_file/test_update_weights_from_disk.py` (模块 权重更新测试；类别 test；类型 rename-or-move；符号 `test_update_weights_nonexistent_model`, `test_update_weights_rejects_invalid_requests`, `test_update_weights_missing_model_path`, `test_update_weights_nonexistent_module`)：移动并简化了测试用例，体现了测试组织优化。

关键符号：`collect_test_items`, `parse_junit_xml_for_executed_cases`, `parse_junit_xml_for_case_results`, `_run_pytest_attempt`, `_extract_collection_line`, `_extract_short_test_summary`, `_extract_failure_tail`, `_summary_has_retryable_failure`, `partition_items_by_index`, `run_pytest`

关键源码片段

`python/sglang/multimodal_gen/test/runner/pytest_runner.py`

核心新增文件，抽取了所有公共 pytest 运行逻辑，是整个重构的基石。

```
def collect_test_items(
    files: Sequence[str], filter_expr: str | None = None
) -> list[str]:
    """Collect pytest node IDs from the given files or node selectors."""
    # 构造 pytest --collect-only 命令，-q 减少输出
    cmd = [sys.executable, "-m", "pytest", "--collect-only", "-q"]
    if filter_expr:
```

```

    cmd.extend(["-k", filter_expr])
cmd.extend(files)

filter_note = f" with filter: {filter_expr}" if filter_expr else ""
print(f"Collecting tests from {len(files)} item(s){filter_note}")
result = subprocess.run(cmd, capture_output=True, text=True)

# pytest 返回码 5 表示未收集到测试，是预期行为
if result.returncode not in (0, 5):
    error_msg = (
        f"pytest --collect-only failed with exit code {result.returncode}\n"
        f"Command: {' '.join(cmd)}\n"
    )
    if result.stderr:
        error_msg += f"stderr:\n{result.stderr}\n"
    if result.stdout:
        error_msg += f"stdout:\n{result.stdout}\n"
    raise RuntimeError(error_msg)

if result.returncode == 5:
    print(
        "No tests were collected (exit code 5). This may be expected with filters."
    )

# 解析 stdout 提取测试 ID（形如 file::class::method 的行）
test_items = []
for line in result.stdout.strip().split("\n"):
    line = line.strip()
    if line and "::" in line and not line.startswith(("=", "-", " ")):
        test_id = line.split()[0] if " " in line else line
        if "::" in test_id:
            test_items.append(test_id)

print(f"Collected {len(test_items)} test items")
return test_items

```

评论区精华

PR 作者 mickqian 在评论中 @ping1jing2 @bingxche 询问重构是否会影响 NPU/AMD CI（见 PR 评论）。目前没有进一步的讨论或 review 反馈，表明可能已通过内部沟通确认安全。

- 重构是否影响 NPU/AMD CI (question): 暂未看到回复，但 PR 已合并，推测已通过其他渠道确认无影响。

风险与影响

- 风险:

1. 测试 runner 回归：重构后的 pytest_runner.py 可能引入新的 bug，导致测试收集、执行或结果解析出现错误，尤其是 _run_pytest_attempt 的流式读取和重试逻辑需要仔细

验证。

2. 文件移动导致路径依赖失效：多个测试文件被移动（如 `test_update_weights_from_disk.py` 从 `server/` 移到 `single_test_file/`），若其他脚本或 CI 配置硬编码了旧路径，可能导致测试无法找到。
3. LTX2 snapshot 弃用影响：尽管保留了别名，但文档指出弃用窗口，现有使用 snapshot 模式的用户可能在未来版本中遇到兼容性问题。
4. NPU/AMD 平台风险：musa runner 被重写且公共 runner 未在 NPU/AMD 上充分测试，可能导致这些平台 CI 失败。
 - 影响：影响范围：主要影响 diffusion 模块的 CI 流程和测试开发者。影响程度：中等。功能上保持了等价性，但代码组织大幅改善。
 - NPU/AMD 平台可能需要验证新的 runner 是否正常工作。LTX2 用户需要注意 snapshot 模式的弃用通知。
 - 风险标记：测试核心路径变更，文件移动影响路径解析，平台兼容风险（NPU/AMD），LTX2 snapshot 弃用

关联脉络

- 暂无明显关联 PR