# PR #22879 完整报告

sgl-project/sglang

[Diffusion] [NPU] Fix multimodal gen CI

合并时间：2026-04-17 09:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22879

## 执行摘要

- 一句话：分离 GPU 测试用例并新增 NPU 专用测试运行器，修复 NPU CI 因下载 GPU 模型而失败的问题。
- 推荐动作：建议团队在合并前验证 CI 自动分区功能是否受影响，并检查所有导入路径是否已正确更新。对于学习价值，此 PR 展示了如何通过分离平台特定逻辑来优化 CI 流程，适合关注测试架构和跨平台兼容性的工程师参考。

## 功能与动机

根据 PR body 描述，在 PR #22810 后，NPU CI 开始尝试下载 GPU CI 的所有模型，但 NPU 服务器无法完成这些下载操作，导致 CI 失败。为解决此问题，需要将 GPU 测试用例与 NPU 测试分离，避免不必要的模型下载。

## 实现拆解

1. 新增 NPU 测试运行器：创建 python/sglang/multimodal_gen/test/run_suite_npu.py，定义 NPU 专用的测试套件（如 1-npu、2-npu）和运行逻辑，核心函数包括 parse_args、collect_test_items、run_pytest 和 main，用于管理测试并行执行。
2. 分离 GPU 测试用例：新增 python/sglang/multimodal_gen/test/server/gpu_cases.py，从原 testcase_configs.py 中迁移所有 GPU 相关测试用例（如 ONE_GPU_CASES_A），并调整导入以移除平台依赖。
3. 清理原配置文件：修改 python/sglang/multimodal_gen/test/server/testcase_configs.py，删除 GPU 测试用例定义和不再需要的导入（如 current_platform 和测试工具），使其专注于通用配置和 NPU 基线加载。
4. 更新现有测试文件：修改多个测试文件（如 run_suite.py、test_server_1_gpu.py、test_server_2_gpu.py），将导入从 testcase_configs 切换到 gpu_cases，并移除 NPU 套件定义以避免冲突。
5. 调整 CI 工作流：更新 .github/workflows/pr-test-npu.yml，确保使用新的 run_suite_npu.py 运行器，并可能调整测试分区逻辑。 配套改动包括更新性能基线文件（如 perf_baselines_npu.json）和修复文档字符串，以保持一致性。

关键文件：

- python/sglang/multimodal_gen/test/run_suite_npu.py（模块 测试运行器；类别 test；类型 entrypoint；符号 parse_args, collect_test_items, run_pytest, main）：新增的 NPU

专用测试运行器，是修复 CI 的核心入口，定义了 NPU 测试套件和并行执行逻辑。

- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 测试用例；类别 test；类型 test-coverage）：新增的 GPU 测试用例集合，从原配置文件中分离出来，避免了 NPU CI 下载 GPU 模型。
- python/sglang/multimodal_gen/test/server/testcase_configs.py（模块 测试配置；类别 test；类型 configuration）：核心配置文件，移除了 GPU 测试用例和平台相关导入，现在专注于通用配置和 NPU 基线加载。
- python/sglang/multimodal_gen/test/run_suite.py（模块 测试运行器；类别 test；类型 test-coverage）：修改了原有测试运行器，更新导入以使用 gpu_cases，并移除了 NPU 套件定义，防止冲突。
- .github/workflows/pr-test-npu.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：CI 工作流文件，更新以确保使用新的 run_suite_npu.py 运行 NPU 测试，影响实际 CI 执行。

关键符号：parse_args, collect_test_items, run_pytest, main

## 关键源码片段

### python/sglang/multimodal_gen/test/run_suite_npu.py

新增的 NPU 专用测试运行器，是修复 CI 的核心入口，定义了 NPU 测试套件和并行执行逻辑。

以下是从 run_suite_npu.py 中整理的 collect_test_items 函数，展示了如何收集测试项并处理错误：

```
def collect_test_items(files, filter_expr=None):
    """Collect test item node IDs from the given files using pytest --collect-only."""
    cmd = [sys.executable, "-m", "pytest", "--collect-only", "-q"]
    if filter_expr:
        cmd.extend(["-k", filter_expr]) # 添加pytest过滤表达式，用于筛选特定测试
    cmd.extend(files) # 将测试文件列表加入命令

    print(f"Collecting tests with command: {' '.join(cmd)}")
    result = subprocess.run(cmd, capture_output=True, text=True) # 执行pytest收集命令

    # 检查收集错误：pytest退出码处理，确保在错误时抛出异常
    # 退出码说明：0=成功，1=收集时有错误，2=执行中断，3=内部错误，4=命令行错误，5=
    无测试收集（可能因过滤）
    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"
        logger.error(error_msg)
```

```
        raise RuntimeError(error_msg) # 抛出运行时错误，便于CI日志追踪

    if result.returncode == 5:
        print(
            "No tests were collected (exit code 5). This may be expected with filters."
        ) # 处理无测试收集的情况，避免误报

    # 解析输出以提取测试节点ID：pytest -q 输出格式为 test_file.py::TestClass::test_method[param]
    test_items = []
    for line in result.stdout.strip().split("\n"):
        line = line.strip()
        # 跳过空行和摘要行（以=、-或空格开头），仅处理包含::的有效测试ID
        if line and "::" in line and not line.startswith(("=", "-", " ")):
            test_id = line.split()[0] if " " in line else line # 处理可能附加额外信息的行
            if "::" in test_id:
                test_items.append(test_id) # 添加到测试项列表

    print(f"Collected {len(test_items)} test items")
    return test_items # 返回收集到的测试项，用于后续分片执行
```

## 评论区精华

- 关键正确性问题：gemini-code-assist[bot] 指出 gpu_cases.py 中使用通配符导入，缺少了 current_platform，可能导致运行时 NameError。建议改为显式导入，以确保平台检测功能正常。
- 文档不一致：gemini-code-assist[bot] 和 ping1jing2 发现 run_suite_npu.py 的文档字符串示例中文件名和套件名不一致（如引用 run_suite.py 和 1-gpu），要求更新为 run_suite_npu.py 和 1-npu，以提高准确性。
- 未使用代码：gemini-code-assist[bot] 提到 run_suite_npu.py 中定义了未使用的函数 _is_in_ci，建议移除以避免代码冗余。
- 设计权衡：mickqian 在批准后补充评论，指出此 PR“完全破坏了 multimodal_gen CI 的自动分区功能”，关联 Issue #23076，暗示分离测试可能影响了 CI 的智能调度机制，需进一步验证。
  - gpu_cases.py 中缺少 current_platform 导入 (correctness): 建议改为显式导入，已在后续提交中修复。
  - run_suite_npu.py 文档不一致 (documentation): Makcum888e 回复已修正为 run_suite_npu.py 和 1-npu。
  - 破坏 CI 自动分区功能 (design): 未在 PR 内解决，需团队后续验证和修复。

## 风险与影响

- 风险：- 运行时错误风险：gpu_cases.py 中缺少 current_platform 导入（已在 review 中识别并修复），若未正确处理，可能导致测试执行时平台检测失败。
- CI 分区功能破坏：根据 mickqian 的评论，此变更可能破坏了 CI 的自动分区特性（如基于运行时估计的负载均衡），影响测试执行的效率和可靠性。

- 维护复杂性增加：新增文件（如 run_suite_npu.py 和 gpu_cases.py）引入了平台特定的代码路径，可能增加后续测试维护和更新时的复杂性。
- 兼容性风险：修改了多个测试文件的导入逻辑，如果其他模块仍依赖原 testcase_configs.py 的结构，可能导致导入错误或测试遗漏。
- 影响：- 对用户的影响：主要影响开发者和 CI 系统，修复后 NPU CI 能正常运行，避免因模型下载失败而阻塞开发流程；但潜在的自动分区问题可能导致测试执行时间不稳定。
- 对系统的影响：增强了测试的平台隔离性，使 NPU 和 GPU 测试能独立管理，有利于未来扩展多平台支持；但代码库中测试文件数量增加，可能略微增大构建和运行开销。
- 对团队的影响：工程师需适应新的测试结构和运行方式，review 中识别的问题已部分解决，但未解决的自动分区疑虑需团队后续跟进。
- 风险标记：缺少测试导入，CI 分区破坏，维护复杂性增加

## 关联脉络

- PR #22810 （未提供标题，但从 PR body 引用）：PR body 明确指出此变更用于修复 PR #22810 引入的 NPU CI 下载 GPU 模型问题，是直接关联的前序变更。
- PR #23076 （从评论中引用，但未在历史 PR 列表中，假设为相关 Issue）：mickqian 在评论中提及此 PR 破坏了 CI 自动分区功能，关联到 Issue #23076，表明可能存在后续修复需求。