PR #23076 完整报告

sgl-project/sglang

[diffusion] CI: fix auto-partition

合并时间：2026-04-17 22:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23076

执行摘要

- 一句话：修复扩散模型 CI 自动分区逻辑，支持多配置文件并防止空套件。
- 推荐动作：对于负责 CI 基础设施或扩散测试的工程师，建议精读以了解动态配置解析的设计。关注 resolve_case_config_path 函数和 validate_suite_case_coverage 验证逻辑，这些是防止 CI 回归的关键设计决策。

功能与动机

从 review 评论推断，主要动机是防止 CI 信号回归，确保扩散测试的动态套件不因配置变更而变为空，从而避免测试覆盖丢失。

实现拆解

1. 动态解析测试用例配置路径：修改 scripts/ci/utils/diffusion/diffusion_case_parser.py，新增 resolve_case_config_path 函数，通过解析 run_suite.py 中的导入语句来动态确定扩散用例配置文件路径，替代硬编码路径 testcase_configs.py。这样支持多配置文件（如 gpu_cases.py），并确保 CI 保持单一数据源。
2. 添加套件验证逻辑：在 scripts/ci/utils/diffusion/compute_diffusion_partitions.py 中新增 validate_suite_case_coverage 函数，检查动态扩散套件是否包含参数化用例，防止空套件导致分区错误。
3. 更新覆盖验证脚本：修改 scripts/ci/utils/diffusion/verify_diffusion_coverage.py，使用动态解析路径并引入相同 guardrails，在覆盖验证中拒绝空动态套件。
4. 调整性能基准数据：更新 python/sglang/multimodal_gen/test/server/perf_baselines.json，修正 fast_hunyuan_video 等测试用例的阶段时间和去噪步骤数据，确保基准准确反映当前测试性能。

关键文件：

- scripts/ci/utils/diffusion/diffusion_case_parser.py（模块 CI 脚本；类别 infra；类型 infrastructure；符号 resolve_case_config_path）：核心解析逻辑修改，新增动态配置路径解析函数，支持多配置文件并替代硬编码路径。
- scripts/ci/utils/diffusion/compute_diffusion_partitions.py（模块 CI 脚本；类别 infra；类型 infrastructure；符号 validate_suite_case_coverage）：分区计算脚本中添加验证逻辑，防止动态扩散套件为空，确保 CI 分区正确。

- python/sglang/multimodal_gen/test/server/perf_baselines.json（模块 性能基准；类别 test；类型 test-coverage）：更新性能基准数据，确保扩散测试的基准准确反映当前性能，影响测试结果验证。
- scripts/ci/utils/diffusion/verify_diffusion_coverage.py（模块 CI 脚本；类别 infra；类型 infrastructure）：覆盖验证脚本中使用动态解析路径并添加相同 guardrails，确保覆盖验证不通过空动态套件。

关键符号：resolve_case_config_path, validate_suite_case_coverage

关键源码片段

scripts/ci/utils/diffusion/diffusion_case_parser.py

核心解析逻辑修改，新增动态配置路径解析函数，支持多配置文件并替代硬编码路径。

```
def resolve_case_config_path(repo_root: Path, run_suite_path: Path) -> Path:
    """
    从 run_suite.py 的导入中解析扩散用例配置路径。
    run_suite.py 必须从同一个模块导入 ONE_GPU_CASES 和 TWO_GPU_CASES，
    该模块被视为单一数据源。
    """
    with open(run_suite_path, "r", encoding="utf-8") as f:
        content = f.read()

    tree = ast.parse(content, filename=str(run_suite_path))
    one_gpu_module: Optional[str] = None
    two_gpu_module: Optional[str] = None

    # 遍历 AST 查找导入语句
    for node in ast.walk(tree):
        if not isinstance(node, ast.ImportFrom) or not node.module:
            continue
        imported_names = {alias.name for alias in node.names}
        if "ONE_GPU_CASES" in imported_names:
            one_gpu_module = node.module
        if "TWO_GPU_CASES" in imported_names:
            two_gpu_module = node.module

    # 验证必须导入两个常量
    if one_gpu_module is None or two_gpu_module is None:
        raise RuntimeError(
            "run_suite.py must import BOTH ONE_GPU_CASES and TWO_GPU_CASES."
        )
    if one_gpu_module != two_gpu_module:
        raise RuntimeError(
            f"run_suite.py imports ONE_GPU_CASES and TWO_GPU_CASES from different modules:
            {one_gpu_module} vs {two_gpu_module}"
        )

    # 构建相对路径并返回
```

```
    rel_path = Path(*one_gpu_module.split(".")).with_suffix(".py")
    config_path = repo_root / rel_path
    if not config_path.exists():
        raise FileNotFoundError(f"Case config not found: {config_path}")
    return config_path
```

评论区精华

review 中仅有一个 bot 评论总结了变更，没有实质讨论或争议。bot 指出变更支持多配置文件并添加了 guardrails 以防止空动态套件。

- 暂无高价值评论线程

风险与影响

- 风险：风险包括：1) 配置解析逻辑可能错误（如导入解析失败）导致 CI 失败或误报测试覆盖；2) 性能基准更新若不准确（如数据未校准），可能影响测试结果的可靠性；3) 新增的验证逻辑可能引入额外开销，但在 CI 环境中影响有限。
- 影响：对最终用户无直接影响，但显著提升了扩散模型 CI 测试的可靠性和健壮性。对于开发团队，减少了因空套件导致的 CI 失败风险，确保测试信号不丢失。影响范围限于扩散模块的 CI 流程和测试基础设施。
- 风险标记：配置解析错误，测试覆盖不准确

关联脉络

- PR #22955 [Diffusion] Fix ModelOpt B200 CI artifact coverage: 同为扩散模块的 CI 修复，涉及测试覆盖和权重文件选择，共享扩散测试基础设施。
- PR #23052 [diffusion] doc: update doc: 同为扩散模块的文档更新，表明近期扩散功能持续演进，CI 修复是其中的基础设施调整。