PR #24630 完整报告

sgl-project/sglang

[NPU] Diffusion CI Ground Truth Generation (NPU)

合并时间：2026-06-04 05:14

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24630

执行摘要

- 一句话：为 NPU 扩散测试增加 GT 生成并重构套件
- 推荐动作：值得仔细阅读，特别是 run_suite.py 中的条件导入模式和 compute_diffusion_partitions.py 的环境变量切换，这是多平台测试框架的设计模板。同时关注后续的路径优化和 URL 迁移，避免遗留硬编码风险。

功能与动机

PR 描述指出需要在 Ascend 测试中增加一致性检查的 GT 生成，这是实现扩散模型在 NPU 上持续集成的基础设施改造。

实现拆解

1. 新增 NPU GT 生成 workflow：创建 .github/workflows/diffusion-ci-gt-gen-npu.yml，支持 workflow_dispatch 触发，包含计算分区、1-NPU 和 2-NPU 的 ground truth 生成作业，测试结果发布到 sgl-project/ci-data 仓库。
2. 重构测试套件结构：修改 python/sglang/multimodal_gen/test/run_suite.py，移除内联的套件定义（FILE_SUITES、PARAMETRIZED_CASE_GROUPS 等），改为通过 current_platform.is_npu() 条件导入对应平台配置：NPU 从 testcase_configs_npu.py 导入，GPU 从 gpu_cases.py 导入。这消除了原有的 run_suite_npu.py 重复代码。
3. 删除冗余文件和测试：删除 run_suite_npu.py（299 行）和 test_server_8_npu.py（31 行），将 8-NPU 测试用例（如 wan2_2_t2v_14b_w8a8_8npu）转换为 2-NPU，减少设备占用。
4. 移动配置集中化：在 python/sglang/multimodal_gen/test/server/gpu_cases.py 尾部新增 _discover_unit_tests 函数及一系列套件常量，在 testcase_configs_npu.py 中导出 NPU 版配置，形成平台独立的配置模块。
5. 更新性能基线：修改 perf_baselines_npu.json，将所有基线数据替换为在 Ascend A3 硬件上重新测量后的数值，并调整 hardware 字段为 Ascend A3。
6. 调整 CI 脚本：修改 compute_diffusion_partitions.py 和 diffusion_case_parser.py，通过环境变量 USE_NPU_CONFIGS 切换 NPU 配置，避免在 CPU 协调器上误判平台。同时更新 pr-test-npu.yml 移除 8-NPU 相关作业。
7. 添加辅助配置：在 test_utils.py 中添加 Ascend 专用的 ground truth URL 常量（指向 sgl-project/ci-data），并在 CODEOWNERS 中新增 NPU 测试目录的所有者行。

关键文件：

- .github/workflows/diffusion-ci-gt-gen-npu.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：新增 NPU 扩散 CI ground truth 生成工作流，是整个 PR 的核心交付物。定义了 workflow_dispatch 触发、计算分区、1-NPU 和 2-NPU 的 GT 生成作业，以及结果发布到 ci-data 仓库。
- python/sglang/multimodal_gen/test/run_suite.py（模块 测试套件；类别 test；类型 test-coverage；符号 _discover_unit_tests）：重构核心文件，改为通过平台检测动态导入配置，消除了 run_suite_npu.py 的重复。
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 GPU 用例；类别 test；类型 test-coverage；符号 _discover_unit_tests, FILE_SUITES, PARAMETRIZED_CASE_GROUPS, STANDALONE_FILES）：原 run_suite.py 中的套件配置被移到该文件尾部，使其成为 GPU 套件配置的中心文件。
- python/sglang/multimodal_gen/test/run_suite_npu.py（模块 测试套件；类别 test；类型 deletion；符号 parse_args, collect_test_items, run_pytest, main）：因功能重复被删除，是 PR 中最大的删除文件（299 行），反映重构意图。
- python/sglang/multimodal_gen/test/server/ascend/test_server_8_npu.py（模块 8-NPU 测试；类别 test；类型 deletion；符号 TestDiffusionServerEightNpu, case）：被删除，8-NPU 测试被 2-NPU 替代，以节省设备。
- python/sglang/multimodal_gen/test/server/ascend/perf_baselines_npu.json（模块 性能基线；类别 test；类型 test-coverage）：更新了所有性能基线数据，反映 Ascend A3 硬件和老化的 2-NPU 测试。
- python/sglang/multimodal_gen/test/server/ascend/testcase_configs_npu.py（模块 NPU 配置；类别 test；类型 test-coverage）：导出 NPU 版套件配置，是条件导入的目标模块。

关键符号：_discover_unit_tests, parse_args, collect_test_items, run_pytest, TestDiffusionServerEightNpu.case

关键源码片段

python/sglang/multimodal_gen/test/run_suite.py

重构核心文件，改为通过平台检测动态导入配置，消除了 run_suite_npu.py 的重复。

```
# python/sglang/multimodal_gen/test/run_suite.py (head)
"""
Test runner for multimodal_gen that manages test suites and parallel execution.

For diffusion 1-gpu/2-gpu suites, cases are partitioned by estimated runtime
using LPT so each CI shard has a similar total runtime.
"""

from sglang.multimodal_gen.runtime.platforms import current_platform
from sglang.multimodal_gen.test.server.testcase_configs import (
    BASELINE_CONFIG,
    DiffusionTestCase,
```

```
)

# 根据当前平台动态导入对应的套件配置
# 避免维护两份 run_suite.py 复制，TODO: remove duplicated code
if current_platform.is_npu():
    from sglang.multimodal_gen.test.server.ascend.testcase_configs_npu import (
        _UPDATE_WEIGHTS_FROM_DISK_TEST_FILE,
        COMPONENT_ACCURACY_SUITES,
        DEFAULT_EST_TIME_SECONDS,
        DEFAULT_STANDALONE_EST_TIME_SECONDS,
        FILE_SUITES,
        PARAMETRIZED_CASE_GROUPS,
        STANDALONE_FILES,
        STARTUP_OVERHEAD_SECONDS,
        SUITES,
    )
else:
    from sglang.multimodal_gen.test.server.gpu_cases import ( # noqa: F401
        _UPDATE_WEIGHTS_FROM_DISK_TEST_FILE,
        _UPDATE_WEIGHTS_MODEL_PAIR_ENV,
        _UPDATE_WEIGHTS_MODEL_PAIR_IDS,
        COMPONENT_ACCURACY_FILE_NUM_GPUS,
        COMPONENT_ACCURACY_SUITES,
        DEFAULT_EST_TIME_SECONDS,
        DEFAULT_STANDALONE_EST_TIME_SECONDS,
        FILE_SUITES,
        ONE_GPU_CASES,
        PARAMETRIZED_CASE_GROUPS,
        STANDALONE_FILE_EST_TIMES,
        STANDALONE_FILES,
        STARTUP_OVERHEAD_SECONDS,
        STRICT_SUITES,
        SUITES,
        TWO_GPU_CASES,
    )
```

python/sglang/multimodal_gen/test/server/gpu_cases.py

原 run_suite.py 中的套件配置被移到该文件尾部，使其成为 GPU 套件配置的中心文件。

```
# python/sglang/multimodal_gen/test/server/gpu_cases.py (head 末尾新增部分 )

def _discover_unit_tests() -> list[str]:
    """自动发现 unit 目录下的测试文件"""
    unit_dir = Path(__file__).resolve().parent.parent / "unit"
    if not unit_dir.is_dir():
        return []
    return sorted(
        f"../unit/{f.name}" for f in unit_dir.glob("test_*.py") if f.is_file()
    )
```

```
FILE_SUITES = {
    "unit": _discover_unit_tests(),
    "component-accuracy": ["test_component_accuracy_1_gpu.py", "test_component_accuracy_2_
    gpu.py"],
    "component-accuracy-1-gpu": ["test_component_accuracy_1_gpu.py"],
    "component-accuracy-2-gpu": ["test_component_accuracy_2_gpu.py"],
    "1-gpu-b200": ["test_server_b200.py"],
}

PARAMETRIZED_CASE_GROUPS = {
    "1-gpu": [("test_server_1_gpu.py", ONE_GPU_CASES)],
    "2-gpu": [("test_server_2_gpu.py", TWO_GPU_CASES)],
}
# ... 其他常量和向后兼容的 SUITES 字典
```

评论区精华

核心讨论聚焦于平台检测的正确性、代码复用和环境适配：

- 平台检测问题：gemini-code-assist 指出在 compute_diffusion_partitions.py 中直接使用 current_platform.is_npu() 是错误的，因为分区脚本运行在 CPU 协调器上，应当使用环境变量或参数。作者后续改用 USE_NPU_CONFIGS 环境变量解决。
- 代码重复：ping1jing2 询问 run_suite.py 中的重复代码（FILE_SUITES 等）能否避免，作者添加了 TODO，但最终通过条件导入消除了重复。
- 硬编码路径：gemini-code-assist 指出 /root/... 等绝对路径不可移植，建议使用 repo_root 构造路径。作者表示将替换为变量。
- 个人仓库 URL：ground truth 数据 URL 指向 e-martirosian 个人仓库，被要求迁移到 sgl-project 组织仓库以确保长期维护。
- 性能基线波动：Makcum888e 要求仔细检查基线数字，因为不同 CI 服务器可能给出不同值。作者表示数字来自测试输出，后续又因 CI 失败再次更新。
  - 平台检测方法选择 (correctness): 改用环境变量 USE_NPU_CONFIGS 来切换 NPU 配置。
  - 硬编码路径不可移植 (correctness): 作者表示将替换为变量，但最终代码中路径仍存在，是否完全解决存疑。
  - 个人仓库 URL 可靠性 (security): 标记为待修复，但本次 PR 未完全迁移。
  - 代码重复问题 (design): 作者添加 TODO，最终通过条件导入消除了重复，但留下了 TODO 标记。
  - 8-NPU 测试必要性 (question): 确认不需要 8-npu，移除。
  - 性能基线准确性 (testing): 多次调整后通过，但基线仍然可能因环境波动。
  - 测试失败上报问题 (other): 未在讨论中明确解决，但最终批准。

风险与影响

- 风险：

1. 平台检测混杂：虽然改用环境变量，但 diffusion_case_parser.py 中仍可能残留 current_platform.is_npu() 调用（需要确认最终代码），若在非 NPU 环境下运行会选错配置。
2. 硬编码路径：testcase_configs_npu.py 中的模型缓存路径（ /root/.cache/modelscope/...）依赖特定 CI 环境，在其他环境（如开发者本地）会直接失败。
3. 外部数据源可靠性：Ground truth 数据存储于个人仓库 e-martirosian/ci-data，若账号变动或仓库迁移将导致 GT 下载失败。
4. 性能基线波动：NPU CI 机器性能可能波动，更新后的基线（如 wan2_1_t2v_1.3b_1_npu 的 DenoisingStage 从 26240ms 变为 27796ms）可能仍需多次调整。
5. 删除 8-NPU 测试影响：如果未来需要 8-NPU 规模扩散测试，需重新添加，当前改动彻底移除了基础设施。 - 影响：用户影响：无。系统影响：

- CI 基础设施：新增一个手动触发的 workflow 用于 GT 生成，NPU CI 测试套件从 8-NPU 缩减为 2-NPU，降低了设备需求。
- 代码库：删除 299+31 行冗余代码，新增 214 行 workflow 和约 200 行配置移动，整体行数减少。
- 团队影响：NPU 测试维护者现在可以直接在 testcase_configs_npu.py 中管理套件，无需同步 run_suite_npu.py。但条件导入增加了 run_suite.py 的理解成本。
- 风险标记：平台检测依赖环境变量，硬编码路径不稳定，个人仓库 URL 可靠性，性能基线可能波动，删除 8-NPU 测试无保留

关联脉络

- 暂无明显关联 PR