PR #30290 完整报告

sgl-project/sglang

[AMD] Register 5 CI-verified 1-GPU kernel/attention unit tests for AMD PR CI

合并时间：2026-07-08 05:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30290

执行摘要

- 一句话：为 AMD CI 注册 5 个已有单元测试
- 推荐动作：该 PR 是标准基础设施维护，没有需要深入理解的设计决策。适合快速通过。

功能与动机

扩展 AMD per-commit (PR-tier) 测试覆盖，追踪 ROCm CI 仪表盘。所有 5 个测试已在 NVIDIA CI 上运行，且硬件无关（Triton kernel / plain-torch），因此只需标准 register_amd_ci(...) 两行编辑即可，无需 ROCm 特定代码。

实现拆解

1. 导入 register_amd_ci：在每个目标测试文件中将导入语句 from sglang.test.ci.ci_register import register_cuda_ci 修改为 from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci。
2. 注册 AMD CI：在 register_cuda_ci(...) 之后添加 register_amd_ci(est_time=..., stage="stage-b", runner_config="...")，其中 runner_config 区分 1-gpu-large-amd 和 1-gpu-small-amd 以适应不同超时和资源需求。
3. 包含的文件：test_dsa_metadata.py、test_trtllm_mha_page_table.py、test_gdn_noncontiguous_stride.py、test_kda_kernels.py、test_unified_mamba_views.py，每个文件仅修改两行。

关键文件：

- test/registered/attention/test_gdn_noncontiguous_stride.py（模块 GDN；类别 test；类型 test-coverage）：FLA GDN Triton kernels 的非连续步幅测试，注册到 AMD large suite。
- test/registered/attention/test_kda_kernels.py（模块 KDA；类别 test；类型 test-coverage）：KDA (Kimi delta attention) FLA Triton kernels 测试，注册到 AMD large suite。
- test/registered/attention/test_trtllm_mha_page_table.py（模块 trtllm_mha；类别 test；类型 test-coverage）：Triton 设备端 page-table 构建测试，注册到 AMD small suite。
- test/registered/kernels/test_dsa_metadata.py（模块 DSA；类别 test；类型 test-coverage）：Triton DSA metadata kernels 测试，注册到 AMD large suite。

- test/registered/unit/mem_cache/test_unified_mamba_views.py（模块 Mamba views；类别 test；类型 test-coverage）：UnifiedKVPool._build_mamba_views 的 plain-torch view/stride round-trip 测试，注册到 AMD small suite。

关键符号：未识别

关键源码片段

test/registered/attention/test_gdn_noncontiguous_stride.py

FLA GDN Triton kernels 的非连续步幅测试，注册到 AMD large suite。

```
# test/registered/attention/test_gdn_noncontiguous_stride.py
# 该测试验证 fused_gdn_gating 和 fused_sigmoid_gating_delta_rule_update
# 在非连续 a/b 输入下的正确性，模拟 Qwen3.5-27B v_per_group=3 场景

import unittest
import torch
from sglang.srt.layers.attention.fla.fused_gdn_gating import fused_gdn_gating
from sglang.srt.layers.attention.fla.fused_sigmoid_gating_recurrent import (
    fused_sigmoid_gating_delta_rule_update,
)
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci

# NVIDIA CI 注册
register_cuda_ci(est_time=7, stage="base-b", runner_config="1-gpu-large")
# AMD CI 注册，估计时间相同，分配到 stage-b large AMD 套件
register_amd_ci(est_time=7, stage="stage-b", runner_config="1-gpu-large-amd")

# ... 测试主体不变 ...
```

test/registered/attention/test_kda_kernels.py

KDA (Kimi delta attention) FLA Triton kernels 测试，注册到 AMD large suite。

```
# test/registered/attention/test_kda_kernels.py
# 测试 KDA fused sigmoid gating recurrent 及相关辅助函数

import unittest
import torch
from sglang.srt.layers.attention.fla.cumsum import chunk_local_cumsum
from sglang.srt.layers.attention.fla.fused_recurrent import (
    fused_recurrent_kda_packed_decode,
)
from sglang.srt.layers.attention.fla.fused_sigmoid_gating_recurrent import (
    fused_sigmoid_gating_delta_rule_update,
)
from sglang.srt.layers.attention.fla.index import prepare_chunk_indices
from sglang.srt.layers.attention.fla.kda import (
    fused_recurrent_kda,
    kda_gate_chunk_cumsum,
```

```
)
from sglang.srt.utils.common import get_device
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci

register_cuda_ci(est_time=12, stage="base-b", runner_config="1-gpu-large")
register_amd_ci(est_time=12, stage="stage-b", runner_config="1-gpu-large-amd")

# ... 测试主体不变 ...
```

test/registered/attention/test_trtllm_mha_page_table.py

Triton 设备端 page-table 构建测试，注册到 AMD small suite。

```
# test/registered/attention/test_trtllm_mha_page_table.py
# 单元测试：验证 trtllm_mha 的设备端 page-table 构建与主机端 gather 结果一致

import unittest
from typing import Optional
import torch
from sglang.srt.layers.attention.triton_ops.trtllm_mha_page_table import (
    build_trtllm_mha_page_table,
)
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci
from sglang.test.test_utils import CustomTestCase

register_cuda_ci(est_time=14, stage="base-b", runner_config="1-gpu-small")
register_amd_ci(est_time=14, stage="stage-b", runner_config="1-gpu-small-amd")

# ... 测试主体不变 ...
```

评论区精华

无 review 评论。

- 暂无高价值评论线程

风险与影响

- 风险：极低风险。变更仅涉及测试注册的元数据添加，不修改任何生产代码、测试逻辑或基础设施配置。已在 AMD MI325 (rocm700) 和 ROCm 7.2.0 (rocm720) 两条 CI 流水线上验证通过。
- 影响：影响范围小，仅限于 AMD PR CI 流水线。受益于提升 AMD 平台的测试覆盖，确保 Triton kernel 和 plain-torch 代码在 AMD GPU 上持续回归。对用户无直接影响。
- 风险标记：仅测试注册变更，已在两条 AMD 流水线验证

关联脉络

- PR #30309 [AMD] ci: run multimodal_gen unit suite on AMD: 同为提升 AMD CI 测试覆盖的 PR，延续同一工作方向。

- PR #30386 [AMD] Run MI355X disaggregation Nightly Test with runtime checkout code mechanism: 同为 AMD CI 基础设施改进，但侧重于 nightly 而非 PR CI。