PR #29782 完整报告

sgl-project/sglang

[AMD] Register 3 unit mem_cache + utils tests for stage-b-test-1-gpu-small-amd

合并时间：2026-07-02 03:57

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29782

执行摘要

- 一句话：为 AMD CI 注册 3 个 unit/mem_cache 测试
- 推荐动作：值得阅读以了解如何将测试注册到 AMD CI 中。对于关注 AMD 平台测试覆盖的开发者尤其相关。

功能与动机

这些测试已在 NVIDIA 每提交 CI（base-b-test-1-gpu-small）中运行，但并未在 AMD CI 中注册。PR body 指出，它们仅使用纯 torch 操作 /Triton 内核，而 Triton 注意力后端已是 AMD 的主代码路径，因此只需在现有的 register_cuda_ci 旁添加 register_amd_ci 即可，无需 ROCm 特定代码变更。

实现拆解

1. 导入调整（3 个文件）：在每个文件顶部，将 from sglang.test.ci.ci_register import register_cuda_ci 改为 from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci。
2. 注册调用（3 个文件）：在现有的 register_cuda_ci 行下方添加 register_amd_ci(...) 调用，指定 stage="stage-b"、runner_config="1-gpu-small-amd" 和相应的估计时间。 - test_common.py：est_time=5 - test_triton_kernel_layout.py：est_time=5（相比 CUDA 的 30s 大幅降低，因为 AMD 上实测仅需 5s） - test_hiradix_cache_unit.py：est_time=15
3. 候选排除：在第一次提交通道中曾包含 test_store_cache_4d.py，但在 rocm720 上发现 Triton 编译器缺陷（TritonAMDGPUCanonicalizePointers → PassManager::run failed），故在第二个提交中将其移除，并仅保留 CUDA 注册。
4. 验证：通过 collect_tests 确认 AMD stage-b-test-1-gpu-small-amd 套件测试数从 112 增至 115；三个文件均在 mi3xx 和 rocm720 上通过。

关键文件：

- test/registered/unit/utils/test_common.py（模块 工具函数；类别 test；类型 test-coverage）：注册了 TestFlattenArraysToInt64Tensor 测试的 AMD CI 运行。测试 flatten_arrays_to_int64_tensor 在 CPU 和 CUDA 设备上的正确性，以及 nvidia-smi 辅助函数的 monkey-patched 行为。

- test/registered/unit/mem_cache/test_triton_kernel_layout.py（模块 Triton 内核；类别 test；类型 test-coverage）：注册了 TestTritonKernelLayoutParity 测试的 AMD CI 运行，该测试验证 Triton *decode/extend* 注意力内核的页面布局奇偶性。
- test/registered/unit/mem_cache/test_hiradix_cache_unit.py（模块 HiRadix 缓存；类别 test；类型 test-coverage）：注册了 TestHiRadixCacheKVEvents 测试的 AMD CI 运行，该测试基于纯 torch 操作和 gloo 进程组，无 NVIDIA 特定内核。

关键符号：未识别

关键源码片段

test/registered/unit/utils/test_common.py

注册了 TestFlattenArraysToInt64Tensor 测试的 AMD CI 运行。测试 flatten_arrays_to_int64_tensor 在 CPU 和 CUDA 设备上的正确性，以及 nvidia-smi 辅助函数的 monkey-patched 行为。

test/registered/unit/utils/test_common.py

```
import unittest
from array import array

import torch

from sglang.srt.utils.common import (
    flatten_arrays_to_int64_tensor,
    get_device_sm_nvidia_smi,
    get_nvidia_driver_version_str,
)
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci
from sglang.test.test_utils import CustomTestCase

# 保持原有的 CUDA 注册不变
register_cuda_ci(est_time=5, stage="base-b", runner_config="1-gpu-small")
# 新增 AMD 注册，估计时间与 CUDA 相同（纯 CPU 和 torch 操作）
register_amd_ci(est_time=5, stage="stage-b", runner_config="1-gpu-small-amd")
```

test/registered/unit/mem_cache/test_triton_kernel_layout.py

注册了 TestTritonKernelLayoutParity 测试的 AMD CI 运行，该测试验证 Triton *decode/extend* 注意力内核的页面布局奇偶性。

test/registered/unit/mem_cache/test_triton_kernel_layout.py

```
"""Triton-kernel parity test for the page-aware decode / extend kernels."""

import unittest

import torch

from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci
```

```
_HAS_CUDA = torch.cuda.is_available()

# 保持原有 CUDA 注册不变
register_cuda_ci(est_time=30, stage="base-b", runner_config="1-gpu-small")
# 新增 AMD 注册：实测在 AMD GPU 上该测试只需约 5s，远低于 CUDA 预期
register_amd_ci(est_time=5, stage="stage-b", runner_config="1-gpu-small-amd")
```

test/registered/unit/mem_cache/test_hiradix_cache_unit.py

注册了 TestHiRadixCacheKVEvents 测试的 AMD CI 运行，该测试基于纯 torch 操作和 gloo 进程组，无 NVIDIA 特定内核。

```
test/registered/unit/mem_cache/test_hiradix_cache_unit.py

"""Unit tests for srt/mem_cache/hiradix_cache.py KV cache events."""

import os
import unittest
from array import array

import torch

from sglang.srt.disaggregation.kv_events import BlockStored, StorageMedium
from sglang.srt.mem_cache.allocator import TokenToKVPoolAllocator
from sglang.srt.mem_cache.base_prefix_cache import InsertParams, MatchPrefixParams
from sglang.srt.mem_cache.cache_init_params import CacheInitParams
from sglang.srt.mem_cache.hiradix_cache import HiRadixCache
from sglang.srt.mem_cache.memory_pool import MHATokenToKVPool, ReqToTokenPool
from sglang.srt.mem_cache.radix_cache import RadixKey
from sglang.srt.server_args import ServerArgs, set_global_server_args_for_scheduler
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci
from sglang.test.test_utils import CustomTestCase

# 保持原有 CUDA 注册不变
register_cuda_ci(est_time=15, stage="base-b", runner_config="1-gpu-small")
# 新增 AMD 注册：该测试不需要 NVIDIA 特定内核，使用 gloo 通信和纯 torch
register_amd_ci(est_time=15, stage="stage-b", runner_config="1-gpu-small-amd")
```

评论区精华

无 review 评论或讨论。PR 由 HaiShaw 审查并批准，无额外讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅为在测试文件中添加 CI 注册声明，不修改任何生产代码或测试逻辑。唯一风险是已通过两个 AMD 渠道的实际运行验证的。
- 影响：影响范围仅限 AMD CI 基础设施。NVIDIA CI 不受影响（register_cuda_ci 的调用未被修改）。三个测试现在将在 AMD 的每提交 CI 中运行，提供对 Triton 注意力内核、

HiRadix 缓存和通用工具函数的 AMD 回归保护。

- 风险标记：无风险（仅 CI 注册变更）

关联脉络

- PR #29784 [AMD][DI][CI] 2/N Add DSV4 DP8/EP8 and MTP MI355X 1P1D nightly recipes: 同一作者 michaelzhang-ai 对 AMD CI 基础设施的系列贡献之一，专注于 AMD MI355X 的测试配置。
- PR #29678 feat(mem_cache): unified memory pool for hybrid Mamba / SWA models: 涉及相同的 mem_cache 测试文件（test_triton_kernel_layout.py 和 test_hiradix_cache_unit.py），这些测试在此 PR 中被注册到 AMD CI。