

PR #27817 完整报告

sgl-project/sglang

[AMD] ci: register 8 attention-backend unit tests to run on AMD CI

合并时间: 2026-06-13 11:52

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27817>

执行摘要

- 一句话: 在 AMD CI 注册 8 个注意力后端单元测试
- 推荐动作: 值得合并。变更模式清晰, 验证充分, 两条 AMD 流水线均通过。建议后续关注 PCG 在 ROCm 上的进展并及时恢复跳过。

功能与动机

缩小 NVIDIA 与 AMD 之间的 CI 覆盖差距, 确保 AMD GPU 上可移植的注意力后端测试得到 CI 覆盖。第三批重点覆盖 dense、GDN、KDA、Lightning、SWA 等注意力变体的 Triton 和 torch_native 后端。

实现拆解

1. 注册 AMD CI: 在 8 个测试文件中, 将 `from sglang.test.ci.ci_register import register_cuda_ci` 改为同时导入 `register_amd_ci`, 并添加一行 `register_amd_ci(est_time=..., suite="stage-b-test-1-gpu-large-amd")`。这些文件原本已注册 CUDA 的 `base-b` 套件, 新增注册使同一批测试也在 AMD CI 的 `stage-b-test-1-gpu-large-amd` 分区运行。
2. 跳过 AMD 不兼容的 PCG split-op-extend: 在 5 个文件 (`dense/test_triton`、`gdn/test_torch_native`、`gdn/test_triton`、`kda/test_triton`、`swa/test_triton`) 的 `test_runner_mode_split_op_extend_cases` 方法上添加 `@unittest.skipIf(is_hip(), ...)`。该方法使用 `piecewise-CUDA-graph (PCG)` extend 运行器, 依赖 `TcPiecewiseForwardContext.num_tokens`, 而 PCG 尚未在 ROCm 上实现。
3. 移除 Mamba2 测试的 AMD 注册: 初始尝试中包含了 `mamba/test_mamba2.py`, 但 `rocm720` 验证发现 `causal_conv1d_fn` 在 CUDA 条件导入中, 核心 extend 路径也需要该函数, 无法简单跳过。因此决定将该文件从 AMD 注册中移除, 恢复为仅 CUDA。
4. 无 workflow 变更: 所有目标均为已有的 `stage-b-test-1-gpu-large-amd` 分区, 无需修改 CI 配置文件。

关键文件:

- `test/registered/attention/unittests/dense/test_triton.py` (模块 注意力测试; 类别 test; 类型 test-coverage; 符号 `test_runner_mode_split_op_extend_cases`): 核心注意力测试, 新增 AMD 注册并跳过 PCG split-op-extend 方法

- test/registered/attention/unittests/gdn/test_triton.py (模块 注意力测试; 类别 test; 类型 test-coverage; 符号 test_runner_mode_split_op_extend_cases) : GDN Triton 测试, 同样需要 PCG 跳过
- test/registered/attention/unittests/swa/test_triton.py (模块 注意力测试; 类别 test; 类型 test-coverage; 符号 test_runner_mode_split_op_extend_cases) : SWA Triton 测试, 同样需要 PCG 跳过
- test/registered/attention/unittests/dense/test_torch_native.py (模块 注意力测试; 类别 test; 类型 test-coverage) : 仅新增 AMD 注册, 无跳过
- test/registered/attention/unittests/gdn/test_torch_native.py (模块 注意力测试; 类别 test; 类型 test-coverage; 符号 test_runner_mode_split_op_extend_cases) : 既有注册又有 PCG 跳过
- test/registered/attention/unittests/kda/test_triton.py (模块 注意力测试; 类别 test; 类型 test-coverage; 符号 test_runner_mode_split_op_extend_cases) : 既有注册又有 PCG 跳过
- test/registered/attention/unittests/lightning/test_triton.py (模块 注意力测试; 类别 test; 类型 test-coverage) : 仅新增 AMD 注册
- test/registered/attention/unittests/swa/test_torch_native.py (模块 注意力测试; 类别 test; 类型 test-coverage) : 仅新增 AMD 注册

关键符号: test_runner_mode_split_op_extend_cases

关键源码片段

test/registered/attention/unittests/dense/test_triton.py

核心注意力测试, 新增 AMD 注册并跳过 PCG split-op-extend 方法

```
import sys
import unittest
from pathlib import Path

import torch

from sglang.srt.model_executor.forward_batch_info import ForwardMode
from sglang.srt.utils import is_hip # 新增: 用于 AMD 跳过检查
from sglang.test.test_utils import CustomTestCase

sys.path.insert(0, str(Path(__file__).resolve().parents[1]))

from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci # 新增: 同时注册 AMD CI

# ... 中间省略 import ...

register_cuda_ci(est_time=25, stage="base-b", runner_config="4-gpu-b200")
register_cuda_ci(est_time=25, stage="base-b", runner_config="1-gpu-large")
register_amd_ci(est_time=25, suite="stage-b-test-1-gpu-large-amd") # 新增: 注册 AMD CI 套件
```

... 类定义省略 ...

```
@unittest.skipIf(
    is_hip(), # 跳过条件: AMD/HIP 平台
    "split-op extend runner exercises the piecewise-CUDA-graph path "
    "(TcPiecewiseForwardContext.num_tokens), which is not wired on ROCm.",
)
def test_runner_mode_split_op_extend_cases(self):
    # 该方法使用 PCG extend runner, 在 ROCm 上不可用
    for case, static_num_tokens in self.SPLIT_OP_CASES:
        for breakable in (False, True):
            runner = "bcg" if breakable else "pcg"
            with self.subTest(case=case.name, backend=case.backend, runner=runner):
                run_dense_split_op_extend_case(self, case, breakable=breakable, static_num_
                    tokens=static_num_tokens)
```

test/registered/attention/unittests/gdn/test_triton.py

GDN Triton 测试, 同样需要 PCG 跳过

```
from sglang.srt.utils import is_hip
# ...
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci

register_cuda_ci(est_time=20, stage="base-b", runner_config="4-gpu-b200")
register_cuda_ci(est_time=20, stage="base-b", runner_config="1-gpu-large")
register_amd_ci(est_time=20, suite="stage-b-test-1-gpu-large-amd")

# ...

@unittest.skipIf(
    is_hip(),
    "split-op extend runner exercises the piecewise-CUDA-graph path "
    "(TcPiecewiseForwardContext.num_tokens), which is not wired on ROCm.",
)
def test_runner_mode_split_op_extend_cases(self):
    # 同 dense/test_triton 的跳过原因
    for case, static_num_tokens in self.SPLIT_OP_CASES:
        for breakable in (False, True):
            runner = "bcg" if breakable else "pcg"
            with self.subTest(...):
                run_gdn_split_op_extend_case(...)
```

test/registered/attention/unittests/swa/test_triton.py

SWA Triton 测试, 同样需要 PCG 跳过

```
from sglang.srt.utils import is_hip
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci

register_cuda_ci(est_time=20, stage="base-b", runner_config="4-gpu-b200")
```

```

register_cuda_ci(est_time=20, stage="base-b", runner_config="1-gpu-large")
register_amd_ci(est_time=20, suite="stage-b-test-1-gpu-large-amd")

# ...
@unittest.skipIf(
    is_hip(),
    "split-op extend runner exercises the piecewise-CUDA-graph path "
    "(TcPiecewiseForwardContext.num_tokens), which is not wired on ROCm.",
)
def test_runner_mode_split_op_extend_cases(self):
    # 同上述跳过原因
    for case, static_num_tokens in self.SPLIT_OP_CASES:
        for breakable in (False, True):
            runner = "bcg" if breakable else "pcg"
            with self.subTest(...):
                run_swa_split_op_extend_case(...)

```

评论区精华

作者在 PR 评论中报告了两轮 AMD 运行发现的问题：

- 第一轮：PCG split-op-extend 在 AMD 上抛出 `AttributeError: 'TcPiecewiseForwardContext' object has no attribute 'num_tokens'`，因此在 5 个文件上用 `@skipIf(is_hip())` 跳过。
- 第二轮 (rocm720)：发现 Mamba2 的 `causal_conv1d_fn` 不仅影响 layout 测试，还影响核心 extend 路径，最终决定完全移除该文件的 AMD 注册。
- 最终验证：8 个文件在 mi3xx 和 rocm720 两条 AMD 流水线上全部通过。
- PCG split-op-extend 在 AMD 上失败 (performance)：在 5 个文件中用 `@skipIf(is_hip())` 跳过该测试方法。
- Mamba2 `causal_conv1d_fn` 在 AMD 上的依赖问题 (correctness)：将 `mamba/test_mamba2.py` 从 AMD 注册中完全移除，恢复为仅 CUDA。

风险与影响

- 风险：低风险。变更仅涉及测试文件中的 CI 注册和跳过装饰器，不影响任何产品代码或 CUDA CI。主要风险是：
 - PCG split-op-extend 在 AMD 被跳过后，若未来 PCG 在 ROCm 上实现，需手动移除此跳过。
 - Mamba2 测试被完全排除在 AMD 之外，可能导致该功能在 AMD 上的回归不被发现。
 - 跳过条件 `is_hip()` 在 AMD 平台返回 `True`，若未来推理芯片也使用 HIP，可能误跳过。
 - 影响：对用户无直接影响。对团队而言，AMD CI 覆盖范围扩大，有利于早期发现注意力后端在 AMD GPU 上的回归。影响范围限定于 CI 基础设施，CUDA CI 完全不变。
 - 风险标记：PCG 路径在 AMD 上跳过，未来需恢复，Mamba2 测试被完全排除在 AMD 之外

关联脉络

- PR #25208 first batch of AMD CI registration: 同一系列的第一批，建立了模式
- PR #25939 second batch of AMD CI registration: 同一系列的第二批，继续缩小覆盖差距
- PR #27811 Restore AMD piecewise CUDA graph support dropped by #23906: PCG
AMD 兼容性问题正在被修复，相关跳过将来可能移除