PR #25210 完整报告

sgl-project/sglang

[AMD] Add amd jit resolve token ids bench ci

合并时间：2026-05-14 15:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25210

执行摘要

- 一句话：将 JIT kernel benchmark 加入 AMD CI 套件
- 推荐动作：此 PR 为纯粹的 CI 配置扩展，变更清晰且经过测试验证。建议快速合并。

功能与动机

为 JIT kernel 提供 AMD 设备上的持续集成基准测试，填补 AMD 平台缺乏 JIT kernel CI 覆盖的空白。PR body 提到："To complete the CI benchmark of JIT kernels with AMD devices."

实现拆解

1. 导入注册函数：在文件 python/sglang/jit_kernel/benchmark/bench_resolve_future_token_ids.py 中，将导入语句从 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，同时引入 AMD 和 CUDA 的 CI 注册函数。
2. 添加 AMD CI 注册行：在原有 register_cuda_ci 调用之后，新增一行 register_amd_ci(est_time=10, suite="jit-kernel-unit-test-amd")，将该 benchmark 注册到名为 jit-kernel-unit-test-amd 的 AMD CI 套件中。
3. 保持向后兼容：保留原有的 CUDA CI 注册，确保 CUDA 平台 CI 流程不受影响。

关键文件：

- python/sglang/jit_kernel/benchmark/bench_resolve_future_token_ids.py（模块 JIT 内核；类别 source；类型 dependency-wiring）：单文件变更，注册函数导入与 AMD CI 套件注册添加（+2/−1），是本次 PR 的唯一内容。

关键符号：未识别

关键源码片段

python/sglang/jit_kernel/benchmark/bench_resolve_future_token_ids.py

单文件变更，注册函数导入与 AMD CI 套件注册添加（+2/−1），是本次 PR 的唯一内容。

```
# 文件：python/sglang/jit_kernel/benchmark/bench_resolve_future_token_ids.py
import itertools

import torch
```

```
import triton
import triton.testing

from sglang.jit_kernel.benchmark.utils import (
    DEFAULT_DEVICE,
    get_benchmark_range,
    run_benchmark,
)
from sglang.jit_kernel.resolve_future_token_ids import resolve_future_token_ids_cuda
from sglang.srt.utils import get_compiler_backend
# 同时导入 register_cuda_ci 和新增加的 register_amd_ci
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci

# 保留原有的 CUDA CI 注册
register_cuda_ci(est_time=10, suite="stage-b-kernel-benchmark-1-gpu-large")
# 新增 AMD CI 注册，将 benchmark 加入 AMD 专用套件
register_amd_ci(est_time=10, suite="jit-kernel-unit-test-amd")

SIZE_LIST = get_benchmark_range(
    full_range=[2**n for n in range(4, 16)], # 16 ⋯ 32K elements
    ci_range=[256, 4096],
)

configs = list(itertools.product(SIZE_LIST))

def _torch_resolve(input_ids, future_map):
    input_ids[:] = torch.where(
        input_ids < 0,
        future_map[torch.clamp(-input_ids, min=0)],
        input_ids,
    )

_compiled_resolve = torch.compile(
    _torch_resolve, dynamic=True, backend=get_compiler_backend()
)

@triton.testing.perf_report(
    triton.testing.Benchmark(
        x_names=["size"],
        x_vals=configs,
        line_arg="provider",
        line_vals=["jit", "torch_compile", "torch"],
        line_names=["SGL JIT Kernel", "torch.compile", "PyTorch"],
        styles=[("blue", "-"), ("green", "-."), ("red", "--")],
        ylabel="us",
```

评论区精华

无 review 讨论。PR 获得两次批准（均由 HaiShaw 提交），无其他评论。

- 暂无高价值评论线程

风险与影响

- 风险：本 PR 仅修改一行导入和一行注册调用，无核心逻辑变更。风险极低：新增的注册函数可能因 CI 配置错误（如套件名称不匹配）导致 AMD CI 任务失败，但不影响现有 CUDA CI 流程。
- 影响：
  - 对用户（开发者和维护者）无直接影响，主要提升 CI 覆盖完整性。
  - 对 AMD CI 流水线，将在 jit-kernel-unit-test-amd 套件中额外运行 bench_resolve_future_token_ids.py 基准测试。
  - 对 CUDA CI 无影响，CUDA 端保持原行为。
  - 风险标记：低风险

关联脉络

- 暂无明显关联 PR