PR #23382 完整报告

sgl-project/sglang

[AMD] skip deterministic inference for MLA FP8 test

合并时间：2026-04-23 15:43

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23382

执行摘要

- 一句话：为 AMD CI 跳过 MLA FP8 测试中的确定性推理标志，修复 CI 失败。
- 推荐动作：该 PR 值得快速浏览，以了解如何优雅处理跨平台 CI 测试中的后端差异。重点关注条件逻辑的设计，它展示了在存在上游 bug 时如何临时绕过问题而不破坏现有功能。对于涉及多后端支持的团队，这是一个实用的模式。

功能与动机

PR #23303 添加的 *--enable-deterministic-inference* 标志在 NVIDIA 上工作正常，但在 AMD CI 分区（*stage-b-test-1-gpu-small-amd*）上导致服务器启动失败，错误为：' ValueError: Currently only ['flashinfer', 'fa3', 'triton'] attention backends are supported for deterministic inference, but you explicitly specified 'aiter'.'。根本原因是 ROCm 后端存在三个叠加的 bug：

1) 默认注意力后端 aiter 不在确定性推理允许列表中； 2) 绕过此问题设置 --attention-backend triton 会触发 FUSED_ROPE_ROCM 问题； 3) 进一步禁用该功能后遇到 CUDA 图捕获时的 GPU 内存访问错误。因此，暂时在 AMD CI 上跳过该标志，恢复非确定性基线，直到上游 ROCm 问题解决。

实现拆解

1. 导入依赖调整：在 *test/registered/mla/test_mla_fp8.py* 中，从 sglang.test.test_utils 导入 is_in_amd_ci 函数，用于检测当前是否在 AMD CI 环境中。
2. 重构服务器启动参数：将 setUpClass 方法中的 other_args 列表从内联定义改为变量，并移除 *--enable-deterministic-inference* 标志的硬编码。
3. 条件添加标志：添加条件判断 if not is_in_amd_ci()：，仅在非 AMD CI 环境下将 *--enable-deterministic-inference* 标志追加到 other_args 中，从而在 AMD CI 上跳过该标志。
4. 测试配套：此变更仅影响测试逻辑，不修改生产代码；确保 NVIDIA 路径保持原有行为（添加标志以稳定 MGSM 分数），AMD 路径回退到非确定性推理以避免 CI 失败。

关键文件：

- *test/registered/mla/test_mla_fp8.py*（模块 MLA 测试；类别 test；类型 test-coverage；符号 TestMLA.setUpClass）：这是唯一变更的文件，包含测试逻辑的核心调整，通过条件判断跳过 AMD CI 上的确定性推理标志，直接解决 CI 失败问题。

关键符号：TestMLA.setUpClass, is_in_amd_ci

关键源码片段

test/registered/mla/test_mla_fp8.py

这是唯一变更的文件，包含测试逻辑的核心调整，通过条件判断跳过 AMD CI 上的确定性推理标志，直接解决 CI 失败问题。

```
class TestMLA(CustomTestCase, MGSMEnMixin):
    mgsm_en_score_threshold = 0.8

    @classmethod
    def setUpClass(cls):
        cls.model = DEFAULT_MLA_FP8_MODEL_NAME_FOR_TEST
        cls.base_url = DEFAULT_URL_FOR_TEST
        # 重构参数列表为变量，便于条件添加
        other_args = [
            "--trust-remote-code",
            "--kv-cache-dtype",
            "fp8_e5m2",
            # 注释说明确定性推理的作用：固定 MoE 专家调度和内核归约顺序，减少 MGSM 分数波动
        ]
        # 关键条件判断：仅在非 AMD CI 环境下添加确定性推理标志
        if not is_in_amd_ci():
            # 在 AMD
            上，默认注意力后端（aiter）不在确定性推理允许列表中，导致服务器启动失败，因此跳过此
            标志
            other_args.append("--enable-deterministic-inference")
        # 使用条件构建后的参数启动服务器
        cls.process = popen_launch_server(
            cls.model,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=other_args,
        )
```

评论区精华

Reviewer HaiShaw 在批准前要求作者创建一个 issue，以记录未来需要在 AMD ROCm 上为该测试添加 fp8_e4m3 kv-cache-dtype 和 --enable-deterministic-inference 支持。作者已创建 issue #23536 并链接到本 PR，确保有跟踪机制解决上游 ROCm 问题。讨论焦点在于长期修复的规划，而非当前变更的技术争议。

- AMD ROCm 兼容性跟踪 (other): 作者创建了 issue #23536 并链接到 PR，确保有长期修复计划。

风险与影响

- 风险：技术风险：

  - 回归风险：低。变更仅影响测试逻辑，NVIDIA 路径保持不变，AMD 路径回退到已知可工作的非确定性基线，不会引入新 bug。
  - 兼容性风险：无。不涉及生产代码或 API 变更。
  - 性能影响：无。测试中 AMD 路径禁用确定性推理可能略微增加 MGSM 分数的波动性，但这是预期内的临时措施。具体风险点：依赖 is_in_amd_ci() 函数的准确性；如果该函数误判环境，可能导致标志错误添加或跳过，但函数来自项目内部工具，风险可控。
- 影响：对用户的影响：无直接影响，因为这是内部 CI 测试调整。 对系统的影响：修复了 AMD CI 分区上的测试失败，确保 CI 流水线稳定运行。 对团队的影响：减少了 CI 阻塞，提高了开发效率；同时通过关联 issue #23536 明确了未来修复 ROCm 兼容性的任务，有助于长期维护。
- 风险标记：平台差异处理，临时绕过上游 bug

关联脉络

- PR #23303 [PR #23303]（假设标题为添加确定性推理标志到 MLA FP8 测试）：本 PR 直接修复了 PR #23303 引入的 AMD CI 失败问题，通过条件跳过该 PR 添加的 *--enable-deterministic-inference* 标志。