

PR #29710 完整报告

sgl-project/sglang

[HiSparse]: Skip flaky hisparse-nixl ci

合并时间: 2026-06-30 14:04

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29710>

执行摘要

- 一句话: 跳过不稳定的 NIXL HiSparse 测试
- 推荐动作: 可合并。作为临时缓解措施, 长期应分析测试失败根因并修复。

功能与动机

HiSparse NIXL 测试在 CI 中不稳定 (**skip until stabilized**), 通过跳过 CI 执行来避免频繁失败, 同时保留本地手动运行能力。

实现拆解

1. 修改测试装饰器: 在 `test/registered/disaggregation/test_disaggregation_hisparse.py` 中, 将 `TestDisaggregationDSV4HiSparseNixl` 类的 `@unittest.skipUnless(is_in_ci() or _has_nixl(), ...)` 替换为 `@unittest.skipIf(is_in_ci(), ...)`。
2. 行为变化: 变更前, CI 中如果 `_has_nixl()` 返回 `True` 则会执行测试; 变更后, CI 中始终跳过, 非 CI 环境 (如本地) 仍可运行。
3. 删除冗余注释: 移除了旧的 `skipUnless` 消息参数。

关键文件:

- `test/registered/disaggregation/test_disaggregation_hisparse.py` (模块 测试; 类别 `test`; 类型 `test-coverage`): 修改了测试装饰器, 跳过了 CI 中不稳定的 HiSparse NIXL 测试。

关键符号: 未识别

关键源码片段

`test/registered/disaggregation/test_disaggregation_hisparse.py`

修改了测试装饰器, 跳过了 CI 中不稳定的 HiSparse NIXL 测试。

```
# 变更前: CI 中如果 NIXL 可用则执行测试
# @unittest.skipUnless(is_in_ci() or _has_nixl(), ...)
# 变更后: CI 中始终跳过, 非 CI 环境仍可运行
@unittest.skipIf(is_in_ci(), "Flaky in CI — skip until stabilized.")
class TestDisaggregationDSV4HiSparseNixl(TestDisaggregationDSV4HiSparseBase):
    ...
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。仅修改测试装饰器，不影响任何生产代码。CI 中 HiSparse NIXL 测试覆盖暂时缺失，但测试本身不稳定，跳过可减少 CI 噪音。
- 影响：
 - 用户：无影响。
 - 系统：CI 中断言失败减少。
 - 团队：需后续跟进修复测试稳定性后再恢复 CI 执行。
 - 风险标记：测试覆盖临时丢失

关联脉络

- PR #28348 [AMD]: Enable NIXL PD disaggregation for ROCm(1/n): 同为 NIXL 传输后端相关，涉及 test_disaggregation 测试。