

PR #25145 完整报告

sgl-project/sglang

Fix tests for decode radix cache

合并时间: 2026-05-13 20:31

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25145>

执行摘要

- 一句话: 临时禁用不稳定的 NIXL 测试
- 推荐动作: 这是一个紧急止血的临时变更, 适合快速合并。建议团队在后续 PR 中优先解决 NIXL 后端稳定性问题, 并恢复测试覆盖。

功能与动机

PR body 中明确指出: "Seeing these errors on main", 并提供了两个 CI 失败的链接。作者希望临时禁用 NIXL 测试, 直到后端稳定, 以免阻塞主线开发。

实现拆解

1. 修改测试装饰器: 在文件 `test/registered/distributed/test_disaggregation_decode_radix_cache.py` 中, 将 `TestDisaggregationDecodeRadixCacheNixl` 类的 `@unittest.skipUnless` 装饰器替换为 `@unittest.skip("Temporarily disabled until nixl backend is stable.")`。
2. 保持其他测试不变: Mooncake 后端的测试类 `TestDisaggregationDecodeRadixCacheMooncake` 及其它代码未受影响。
3. 后续计划: 作者在评论中表示只合入此临时修复以解除 CI 阻塞, 将在另一 PR 中彻底修复 NIXL 后端问题。

关键文件:

- `test/registered/distributed/test_disaggregation_decode_radix_cache.py` (模块测试; 类别 test; 类型 test-coverage; 符号 `TestDisaggregationDecodeRadixCacheNixl`): 这是唯一修改的文件, 通过修改装饰器临时禁用了 NIXL 后端的 `decode radix cache` 测试。

关键符号: 未识别

关键源码片段

`test/registered/distributed/test_disaggregation_decode_radix_cache.py`

这是唯一修改的文件, 通过修改装饰器临时禁用了 NIXL 后端的 `decode radix cache` 测试。

```
# 变更前: 仅在 CI 或本地有 NIXL 环境时运行
# @unittest.skipUnless(
#     is_in_ci() or _has_nixl(),
```

```
# "NIXL is required for decode radix cache disaggregation coverage.",
# )
# 变更后：始终跳过，直到 NIXL 后端稳定
@unittest.skip("Temporarily disabled until nixl backend is stable.")
class TestDisaggregationDecodeRadixCacheNixl(
    DisaggregationDecodeRadixCacheTestMixin, PDDisaggregationServerBase
):
    transfer_backend_name = "nixl"
```

评论区精华

无 review 讨论，仅作者自行合并。作者在评论中提供了 CI 通过的截图，确认修改生效。

- 暂无高价值评论线程

风险与影响

- 风险：此变更仅禁用一个测试类，不会影响生产代码。风险极低：
 - NIXL 后端的功能测试将短暂缺失，可能掩盖该后端新的回归问题。
 - Mooncake 后端的测试仍在运行，保障了解码 radix cache 的基本功能验证。
 - 需要在后续 PR 中及时修复 NIXL 后端并重新启用测试，否则 NIXL 相关的回归可能长期未被发现。
- 影响：
 - 对用户：无直接影响。仅影响 CI 流程。
 - 对系统：H20 CI 将恢复稳定，不再因 NIXL 测试失败而阻塞合并。
 - 对团队：为 NIXL 后端的修复争取了时间，但需尽快跟进。
- 风险标记：测试覆盖缺失

关联脉络

- PR #24973 [CI] Add DSV4 Flash disaggregation test: 同为分布式测试的 CI 改进，本次禁用的测试也属于同类测试框架。
- PR #24671 fix(nixl): close file descriptors after each FILE transfer: 与 NIXL 后端相关，可能涉及相同的稳定性问题。