

PR #32638 完整报告

sgl-project/sglang

test: fix NIXL EP Mooncake FT test

合并时间: 2026-08-06 18:00

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32638>

执行摘要

- 一句话: 修复 NIXL Mooncake 弹性 EP 故障测试的进程击杀与恢复验证
- 推荐动作: 该 PR 值得测试与 EP 相关同学快速浏览: 展示了如何通过精确 pkill 与引导请求让故障注入测试变得可复现。但整体是测试修复, 不涉及核心代码, 不必精读。

功能与动机

PR body 指出该测试需要 kill 实际的 DP scheduler。旧代码 pkill 目标进程名 `sglang::scheduler_DP1_TP8_EP8` 与实际进程布局不符, 故障注入并未真正生效; 同时 `os.system` 的 pkill 没有锚定, 匹配范围过宽, 测试既可能漏杀也可能误杀。

实现拆解

1. 调整启动配置: 在 `TestNixlMoeMooncakeElasticEP.server_args` 中追加 `--moe-dense-tp-size 1` 与 `--enable-dp-lm-head`, 使目标进程名变成 `sglang::scheduler_DP1_TP1_EP1`, 与进程实际命名保持一致。
2. 精确进程击杀: 用 `subprocess.run(["pkill", "-f", f"^{self.pkill_process_1}$"], check=True)` 替换 `os.system` 调用, 正则锚定首尾, `check=True` 保证进程存在, 若未命中直接报错, 避免静默失败。
3. 故障后引导请求: 在运行 GSM8K 评测前, 先向 `{self.base_url}/generate` 发送一个 `max_new_tokens=1`、`routed_dp_rank=0` 的最小请求, 让控制器学习 post-fault active rank mask, 再开始并发负载, 避免评测期间因掩码未就绪而失败。
4. 测试配套: 仅修改此测试文件, 未改动源码、配置或其他测试; 原有断言 `metrics["score"] > 0.60` 保持不变。

关键文件:

- `test/manual/ep/test_nixl_ep.py` (模块 弹性 EP; 类别 test; 类型 test-coverage; 符号 `TestNixlMoeMooncakeElasticEP.test_gsm8k_fault_1`): 唯一变更文件, 核心修复 NIXL Mooncake 弹性 EP 故障测试的进程击杀目标与恢复验证流程。

关键符号: `TestNixlMoeMooncakeElasticEP.test_gsm8k_fault_1`

关键源码片段

[test/manual/ep/test_nixl_ep.py](#)

唯一变更文件，核心修复 NIXL Mooncake 弹性 EP 故障测试的进程击杀目标与恢复验证流程。

```
class TestNixlMoeMooncakeElasticEP(_EPTestBase):
    # 通过显式配置 MoE dense TP size 与 DP LM head,
    # 使故障目标的 scheduler 进程名变为 "sglang::scheduler_DP1_TP1_EP1",
    # 与原实现中的错误进程名 "sglang::scheduler_DP1_TP8_EP8" 对齐。
    server_args = [
        *NIXL_COMMON,
        *DP_ATTN,
        *ELASTIC_MOONCAKE,
        "--moe-dense-tp-size",
        "1",
        "--enable-dp-lm-head",
    ]

    # 使用 ^...$ 锚定进程名，避免 pkill 误杀同名前缀的其他进程
    pkill_process_1 = "sglang::scheduler_DP1_TP1_EP1"

    def test_gsm8k_fault_1(self):
        # check=True 强制 pkill 至少命中一个进程，
        # 若进程名因配置变化失效，测试会立即失败而不是静默通过
        subprocess.run(
            ["pkill", "-f", f"^{self.pkill_process_1}$"],
            check=True,
        )
        # 先向幸存 rank 发送一次最小 forward (bootstrap) ,
        # 让控制器在并发负载前学习 post-fault active rank mask
        response = requests.post(
            f"{self.base_url}/generate",
            json={
                "text": "Hello",
                "sampling_params": {"max_new_tokens": 1},
                "routed_dp_rank": 0,
            },
            timeout=120,
        )
        self.assertEqual(response.status_code, 200, response.text)
        metrics = self._run_gsm8k()
        self.assertGreater(metrics["score"], 0.60)
```

评论区精华

UNIDY2002 在批准时提到: "I encountered the same case on the Mooncake side as well, and this solution looks sound to me." 说明他在 Mooncake 侧也遇到过同类问题，并认可该修复方案。ShangmingCai 先评论 LGTM，随后正式批准，无其他技术争议。

- 修复方案可靠性 (design): 方案获维护者认可并合并。

风险与影响

- 风险：本 PR 只改测试，无产品代码风险。测试依赖真实 RDMA 设备与 NIXL/Mooncake 环境，只能在 test/manual/ep 手动测试目录中运行；pkill 通过精确正则锚定进程名，若进程名因后续配置变化而改变，check=True 会直接导致测试失败，需要同步更新进程名。bootstrap 请求增加了一次推理调用，略微延长测试时间。
- 影响：影响范围限于 test/manual/ep/test_nixl_ep.py 的 Mooncake 弹性 EP 用例，不会影响默认 CI 或产品端。对验证 NIXL + Mooncake 容灾能力的团队而言，该用例从 "可能未真正注入故障" 变为 "确定性故障注入 + 恢复验证"，提升了该场景测试的可信度。
- 风险标记：依赖真实 RDMA 环境，pkill 进程名强耦合，仅 manual 测试覆盖

关联脉络

- PR #33498 Build and release sgl-deep-ep wheels: 同属 MoE EP 后端工程化配套，反映 SGLang 在 EP 后端构建、分发与弹性容灾上的持续建设。