

PR #39611 完整报告

sgl-project/sglang

[CI][Disaggregation] Fix EADDRINUSE flake in test_disaggregation_dwdp_gpt_oss

合并时间: 2026-09-15 23:05

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39611>

执行摘要

- 一句话: 拆分测试改走基类启动器, 修复 NCCL 端口竞争
- 推荐动作: 值得精读 (10 分钟内可读完), 重点不在代码体量而在根因链路: 测试 launcher 被覆写 → --nccl-port 丢失 → get_free_port() 返回临时端口后立刻释放 → 背靠背启动的两侧抢到同一端口 → init_process_group EADDRINUSE。建议关注三个设计决策: 一是“声明差异、复用基类启动”的 fixture 设计模式; 二是端口固定值与 dp-attention 派生端口段之间的边界核算; 三是 launch_all() 的 fail-fast 看护。同时建议尽快补一个同源 PR 修复 test_disaggregation_dp_attention.py, 否则同一类 flake 会在别的队列上重演。

功能与动机

该注册测试在 CI 的 setUpClass 阶段反复 flake: 日志显示

torch.distributed.DistNetworkError: The server socket has failed to listen on any local network address. port: 45623 ... EADDRINUSE, 随后 RuntimeError: Rank 0 scheduler died during initialization (exit code: -3) 与 RuntimeError: Server process exited with code 1。PR 描述指出基类 PDDisaggregationServerBase 本就为规避该问题固定了两侧 rendezvous 端口 (注释写明 Pin distinct nccl (torch.distributed rendezvous) ports below the ephemeral range; otherwise both sides derive them from get_free_port() and can race onto the same port), 但该测试自建 launcher, 参数列表中缺少 --nccl-port, 两侧退回 PortArgs.init_new 里绑 0 号端口取回的临时端口并立即释放, 背靠背启动时内核可能把同一端口分给两侧; prefill 侧是非 dp-attention 分支, 没有 wait_port_available 预检, 因此冲突没有被更早拦住。

实现拆解

1. 删除自建 launcher: 移除测试内的 start_prefill/start_decode 以及 setUpClass 中手工的起服、wait_server_ready 健康检查和 launch_lb() 序列, 同时删掉只服务于这条路径的 DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH、popen_launch_pd_server 导入, 共减少 63 行。原因是自建启动器会重建整个参数列表, 无法继承基类的端口固定、transfer 后端与 RDMA 设备拼装逻辑。
2. 改为声明式配置: 新增类属性 prefill_tp_size = NUM_PREFILL_GPUS、decode_tp_size = NUM_DECODE_GPUS、decode_base_gpu_id = NUM_PREFILL_GPUS, 并用 extra_prefill_args (--dwdp-size 2、--disable-flashinfer-autotune、--mem-fraction-static 0.85) 与 extra_decode_args (--dp 2、--enable-dp-attention、

--disable-flashinfer-autotune、--mem-fraction-static 0.85) 只描述与基类默认值的差异。这样 --trust-remote-code、--disaggregation-mode、--disaggregation-bootstrap-port、--tp、--nccl-port、transfer_backend、rdma_devices 全部由 PDDisaggregationServerBase 补齐。

- 收敛到 `launch_all()`: `setUpClass` 精简为 `super().setUpClass()` → 设置 `cls.model` → `cls.launch_all()`, 与 `test_disaggregation_basic.py` 的写法保持一致。基类会统一拉起 `prefill/decode/load balancer`、做就绪等待, 并额外安装子进程 `fail-fast` 看护, 使服务端中途死亡立刻暴露, 而不是表现成下游超时。
- 端口冲突的收口验证: 改动后命令行新增 `prefill --nccl-port 11300`、`decode --nccl-port 11400`, 二者在临时端口范围之外, 且不与 `decode` 侧 `dp-attention` 派生端口段 (11433-11439)、`bootstrap` 端口 (11500) 和 `HTTP` 端口重叠, 因此根因被真正消除而不是被掩盖。
- 测试 / 配套改动: 本 PR 无源码主路径改动, 也无新增测试文件, 验证方式为在关联 issue 中触发 `/rerun-test registered/disaggregation/test_disaggregation_dwdp_gpt_oss.py`, 4-gpu-b200 上 1 个测试通过。
- 已知未处理项: `test_disaggregation_dp_attention.py` 存在完全相同的潜在缺陷 (同样覆写两个 `launcher` 且缺 `--nccl-port`), 作者在 PR 描述中明确说明为控制范围未一并修复, 留给后续 PR。

关键文件:

- `test/registered/disaggregation/test_disaggregation_dwdp_gpt_oss.py` (模块 拆分测试; 类别 `test`; 类型 `test-coverage`; 符号 `setUpClass`, `start_prefill`, `start_decode`, `extra_prefill_args`): 唯一改动文件: 删除自建的 `start_prefill/start_decode`, 改由 `PDDisaggregationServerBase` 的 `launch_all()` 起服, 从而恢复 `--nccl-port` 固定, 消除 `EADDRINUSE` flake。

关键符号: `TestDisaggregationDWDPGptOss.setUpClass`,
`TestDisaggregationDWDPGptOss.start_prefill`,
`TestDisaggregationDWDPGptOss.start_decode`,
`PDDisaggregationServerBase.launch_all`, `PDDisaggregationServerBase.setUpClass`

关键源码片段

`test/registered/disaggregation/test_disaggregation_dwdp_gpt_oss.py`

唯一改动文件: 删除自建的 `start_prefill/start_decode`, 改由 `PDDisaggregationServerBase` 的 `launch_all()` 起服, 从而恢复 `--nccl-port` 固定, 消除 `EADDRINUSE` flake。

```
class TestDisaggregationDWDPGptOss(PDDisaggregationServerBase):
    """PD 拆分: prefill 侧 DWDP (2 GPU), decode 侧 DP attention (2 GPU) 。”
```

```
    NUM_PREFILL_GPUS = 2
    NUM_DECODE_GPUS = 2
```

```
    # 复用基类 fixture 的启动器, 而不是在测试里重写 start_prefill / start_decode。
    # 只有走基类路径, 两侧才会带上固定好的 --nccl-port; 否则各自调用 get_free_port()
```

```

# 会拿到同一个临时端口，并在 init_process_group 阶段以 EADDRINUSE 崩掉。
prefill_tp_size = NUM_PREFILL_GPUS
decode_tp_size = NUM_DECODE_GPUS
decode_base_gpu_id = NUM_PREFILL_GPUS

# 只声明本用例与基类默认值的差异项：--trust-remote-code、--disaggregation-mode、
# --disaggregation-bootstrap-port、--tp、--nccl-port、transfer_backend、
# rdma_devices 等公共参数全部交由基类补齐，避免再次漏参数。
extra_prefill_args = [
    "--dwdp-size",
    str(NUM_PREFILL_GPUS),
    "--disable-flashinfer-autotune",
    "--mem-fraction-static",
    "0.85",
]
extra_decode_args = [
    "--dp",
    str(NUM_DECODE_GPUS),
    "--enable-dp-attention",
    "--disable-flashinfer-autotune",
    "--mem-fraction-static",
    "0.85",
]

@classmethod
def setUpClass(cls):
    super().setUpClass()
    cls.model = GPT_OSS_MODEL_PATH
    # launch_all() 依次拉起 prefill / decode / load balancer 并等待就绪，
    # 同时注册子进程 fail-fast 看护：服务端中途死亡会立刻报错，
    # 而不是退化成下游的健康检查超时。
    cls.launch_all()

def test_gsm8k(self):
    metrics = run_eval(
        SimpleNamespace(
            base_url=self.base_url,
            model=self.model,
            eval_name="gsm8k",
            api="chat",
            num_shots=5,
            num_examples=100,
            max_tokens=4096,
            num_threads=8,
            repeat=1,
            temperature=0.0,
            top_p=1.0,
            host="http://127.0.0.1",
            port=int(self.base_url.split(":")[-1]),

```

```

    )
)
# 精度门限与修复前保持一致，说明本次只是启动方式的调整。
self.assertGreaterEqual(metrics["score"], GSM8K_BASELINE_ACCURACY)

# 基类 PDDisaggregationServerBase 中被本次修复“重新接回”的端口固定逻辑
# （引自 PR 描述，用于说明根因）：把两侧的 torch.distributed rendezvous 端口
# 钉在临时端口范围之外，否则两侧各自 get_free_port() 会抢到同一端口，
# 在 init_process_group 处以 EADDRINUSE 失败。
# 该基类逻辑原本被测试覆写的 launcher 绕过，因此进程只能退回
# PortArgs.init_new 的 get_free_port() 路径（无 wait_port_available 预检）。
cls.prefill_nccl_port = f"{int(base_port) + 300}" # 本用例下为 11300
cls.decode_nccl_port = f"{int(base_port) + 400}" # 本用例下为 11400

```

评论区精华

本 PR 没有人工 review 评论（`review_comments_count = 0`），作者自行合并，讨论集中在与 PR 关联的 issue 区：

ShangmingCai: [/rerun-test registered/disaggregation/test_disaggregation_dwdp_gpt_oss.py](#)

github-actions[bot]: [4-gpu-b200 \(1 test\)](#) : [🔗 View workflow run](#)

即修复的实证来自 CI 复跑通过，而非代码评审；此外 PR 描述里的 Note 是唯一实质性的设计讨论点——`test_disaggregation_dp_attention.py` 有 identical latent bug，作者表示 *happy to fold it in if reviewers prefer*，但最终无人跟进，该隐患仍留在 main 上。

- CI 复跑验证修复效果 (testing): 修复在目标 runner 上验证有效，EADDRINUSE 未再现；但样本仅一次复跑，长期稳定性仍依赖后续 CI 观察。
- `test_disaggregation_dp_attention.py` 存在相同潜在缺陷 (correctness): 未解决：无 reviewer 回应，该隐患仍留在 main 上，同类 flake 可能在别的 CI 队列复现。
- 缺少人工代码评审，作者自合并 (other): 对测试类小改动可接受，但设计层面的“测试夹具复用 vs 重写 launcher”模式未经过第二人确认。

风险与影响

- 风险：
 1. 启动参数控制权转移：起服命令行改由 `PDDisaggregationServerBase.launch_all()` 拼接，若基类后续调整默认参数（新增 / 改默认 flag、改超时、改 transfer backend 拼装顺序），该测试会静默跟随变化，测试与“实际起服命令”之间的可读性下降，需要靠基类单测兜底。
 2. 端口段重叠需持续关注：新增的 `--nccl-port` 取值为 11300/11400，目前与 decode 侧 `dp-attention` 派生块（11433-11439）、bootstrap（11500）不冲突；一旦基类的 `base_port` 或派生偏移调整（例如 +300/+400 改成其他步长），这份“不冲突”的结论需要重新核对。

3. 同类隐患未修: test/registered/disaggregation/test_disaggregation_dp_attention.py 仍覆写 launcher 并省略 --nccl-port, 仍可能在 CI 上以同样的 EADDRINUSE 形式 flake, 属于本次修复的残留缺口。
4. fail-fast 看护带来的行为变化: launch_all() 会在服务端进程死亡时立刻失败。这对 CI 是改进, 但会让过去靠等待超时“偶然混过”的场景变成显式失败, 短期内可能看到更多红色流水线。
5. 无生产代码风险: 改动严格限定在测试文件内 (+25/-63), 不触及调度器、传输层或 KV 路径, 回归风险仅限于该用例自身的覆盖范围。- 影响: 对 CI: 消除了 4-gpu-b200 extra-b 队列中一个可复现的初始化阶段 flake, 减少无意义的 rerun 和排查成本; 对开发者: 确立了“DWDP/ 拆分类测试复用基类 launch_all()”的推荐模式, 后续新增拆分测试可直接照抄, 避免再次踩端口竞争; 对产品运行时: 零影响, 用户侧服务、性能与精度行为均不改变; 对团队: 由于无人工 review、作者自合并, 修复的正确性主要依赖 CI 复跑证据, 建议把防御性写法的经验沉淀到基类文档或测试写作规范中。- 风险标记: 仅测试代码改动, 同类端口隐患未修, 启动参数改由基类拼接, 缺少人工 review

关联脉络

- PR #39600 [CI] Fix SWA decode radix cache NIXL import: 同样是拆分 (disaggregation) 注册测试的初始化阶段崩溃修复, 属于同一条“拆分测试稳定性”工作线。
- PR #39544 [misc] Trim redundant variants from the 8-gpu-h20 disaggregation test suite: 改动同一批 test/registered/disaggregation/* 测试文件, 方向为拆分套件瘦身与耗时优化, 与本 PR 的精简 launcher 思路一致。
- PR #39545 [CI] Wait for a killed test server's GPU memory before the next launch: 改动测试基础设施 python/sglang/test/test_utils.py / runners.py, 处理被杀测试服务器的资源与启动时序问题, 与本 PR 关注的起服竞态同源。
- PR #39368 [CI][PD] Skip the flaky decode HiCache file-backend disaggregation test: 同样处理 PD 拆分测试的 flaky 问题, 但采用跳过 / 放宽阈值的临时手段; 本 PR 则给出了根因级修复, 二者可对比参考。