PR #35500 完整报告

sgl-project/sglang

[CI/NPU] Isolate multi-node tests by run_id to prevent concurrent-run···

合并时间：2026-09-01 11:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35500

执行摘要

- 一句话：NPU 多机测试按 run_id 隔离源码与 pod，避免并发互相干扰
- 推荐动作：值得快速浏览，对 CI 并行隔离设计有参考价值：run_id 贯穿 " 存储路径 + job 名 + pod label + 日志路径 " 四个层面，是一种轻量且完整的运行级隔离范式。重点借鉴 prepare_cm_data 的 "label selector 收窄 + 名称前缀兜底 " 双保险过滤方式；同时注意模板与 Python 脚本必须同步演进，未来新增 job 类型时要同步补 run-id label。

功能与动机

PR body 明确说明："Apply run-scoped isolation across the NPU multi-node test pipeline so that concurrent runs sharing the same k8s namespace and PVC no longer overwrite each other's source code." 并发运行共享同一 k8s namespace 与 PVC 时存在三类问题：源码固定写入 /root/.cache/tests/sglang/ 会互相覆盖；Clear Resources 阶段按 KUBE_JOB_NAME 前缀清理可能误删其他运行的 pod 导致死锁；prepare_cm_data 以宽泛的 app=sgl-ascend label selector 收集 pod 会把其他运行的 pod 写入自己的 ConfigMap。

实现拆解

1. workflow 入口隔离（.github/workflows/nightly-test-npu-e2e-multi-node.yml）：源码复制目标路径改为 /root/.cache/tests/sglang-${github.run_id}，sglang_source_relative_path 同步跟进；KUBE_JOB_NAME 从 ascend-sglang-${inputs.test_type}-test 改为 ascend-sglang-test-${github.run_id}，使每个运行拥有全局唯一的 job/pod 前缀；调用 run_npu_e2e_test.py 时新增 --run-id ${github.run_id} 参数，并在测试结束后 rm -rf 本运行的专属源码目录。
2. 测试编排脚本透传（python/sglang/test/ascend/e2e/run_npu_e2e_test.py）：run_npu_e2e_test_case 新增 run_id: str = "" 参数并注入 single / multi-pd-mix / multi-pd-separation 三种 k8s_context；CLI 新增 --run-id 参数；prepare_cm_data 接收 run_id 后把 label selector 收窄为 app=sgl-ascend,run-id={run_id}，final_kube_job_name 的 pod 名前缀过滤保留作兜底。
3. k8s 模板统一打标（5 个 jinja2 模板）：k8s_single.yaml.jinja2、k8s_multi_pd_mix.yaml.jinja2、k8s_multi_pd_separation.yaml.jinja2 及两个 green 变体，在每个 pod template 的 labels 中加入 run-id: "{{ run_id }}"，与 prepare_cm_data 的 selector 严格对齐，这是整个方案成立的前提。

4. 日志路径隔离（python/sglang/test/ascend/e2e/run_npu_testcase.sh）：plog 备份路径加入 ${run_label} 段（run_label 已携带 run_id 前缀），与 log 路径布局一致，避免同一节点上不同运行的 plog 互相覆盖。
5. 配套与回滚：本 PR 未新增单元测试，正确性依赖 NPU e2e 实跑验证；review 期间提出的 pre-test bounded retry、post-test 精确 job 名清理、|| true 幂等化等增强均被作者回滚，最终合并版本只保留最小的隔离改动。

关键文件：

- python/sglang/test/ascend/e2e/run_npu_e2e_test.py（模块 测试编排；类别 test；类型 test-coverage；符号 prepare_cm_data, run_npu_e2e_test_case）：核心隔离逻辑所在：run_npu_e2e_test_case 新增 run_id 参数并注入三种 k8s_context，prepare_cm_data 按 run-id label selector 收窄 pod 收集范围，CLI 新增 --run-id 参数。
- .github/workflows/nightly-test-npu-e2e-multi-node.yml（模块 CI 流水线；类别 infra；类型 infrastructure）：隔离的入口层：源码目录、KUBE_JOB_NAME、--run-id 传参、结束清理都在这条 workflow 中串联，是并发隔离方案能落地到 CI 的关键。
- python/sglang/test/ascend/e2e/run_npu_testcase.sh（模块 测试脚本；类别 test；类型 test-coverage）：plog 备份路径加入 run_label，与 log 路径对齐，避免同一节点上不同运行的 plog 文件互相覆盖。
- python/sglang/test/ascend/e2e/k8s_multi_pd_separation.yaml.jinja2（模块 部署模板；类别 test；类型 test-coverage）：pd-separation（火山引擎 Job 模式）模板的 prefill/decode/router 三类 pod 均补充 run-id label，是 prepare_cm_data 按 run_id 过滤的前提。
- python/sglang/test/ascend/e2e/k8s_single.yaml.jinja2（模块 部署模板；类别 test；类型 test-coverage）：单机模板同样补充 run-id label，保持所有 job 类型行为一致，避免 single 类型与其他类型并存时 selector 不一致。

关键符号：prepare_cm_data, run_npu_e2e_test_case

关键源码片段

python/sglang/test/ascend/e2e/run_npu_e2e_test.py

核心隔离逻辑所在：run_npu_e2e_test_case 新增 run_id 参数并注入三种 k8s_context，prepare_cm_data 按 run-id label selector 收窄 pod 收集范围，CLI 新增 --run-id 参数。

```
# run_npu_e2e_test.py —— 按 run_id 收窄 pod 收集范围
def prepare_cm_data(namespace, pod_string, run_id=None):
    """构造 ConfigMap 数据：{pod_name: pod_ip}。

    传入 run_id 时把 label selector 收窄到当前运行，
    避免同一 namespace 下并发运行互相污染 ConfigMap；
    pod_string 过滤（final_kube_job_name）仍作为兜底保障。
    """
    if run_id:
        # 与各 jinja2 模板中写入的 run-id label 保持一致
        label_selector = f"app=sgl-ascend,run-id={run_id}"
```

```
    else:
        # 未传 run_id 时保持原来的全量行为，兼容本地 / 非 CI 调用
        label_selector = "app=sgl-ascend"
    pods = core_api.list_namespaced_pod(
        namespace=namespace, label_selector=label_selector
    )
    data = {}
    for pod in pods.items:
        pod_name = pod.metadata.name
        if pod_string in pod_name:
            # 只登记本运行 pod 的名字与 IP，供 ConfigMap 下发
            data[pod_name] = pod.status.pod_ip
    return data
```

.github/workflows/nightly-test-npu-e2e-multi-node.yml

隔离的入口层：源码目录、KUBE_JOB_NAME、--run-id 传参、结束清理都在这条 workflow 中串联，是并发隔离方案能落地到 CI 的关键。

```
# nightly-test-npu-e2e-multi-node.yml —— run_id 贯穿源码路径与 job 命名
# 复制源码到共享 PVC，目录带 run_id 后缀，并发运行互不覆盖
current_path=$(pwd)
target_path=/root/.cache/tests/sglang-${{ github.run_id }}
rm -rf ${target_path}
mkdir -p ${target_path}
cp -r ${current_path}/* ${target_path}/

# job 名携带 run_id，保证清理阶段只命中本运行的前缀
KUBE_JOB_NAME=ascend-sglang-test-${{ github.run_id }}
sglang_source_relative_path=tests/sglang-${{ github.run_id }}

# 把 run_id 传给 Python 编排脚本，用于 pod label 与 ConfigMap 过滤
CMD="${CMD} --kube-job-name-prefix ${KUBE_JOB_NAME} \\
  --run-id ${{ github.run_id }}"

# 测试结束后清理本运行的专属源码目录，避免 PVC 累积残留
rm -rf /root/.cache/tests/sglang-${{ github.run_id }} || true
```

评论区精华

cherryblo 的核心关注点是清理阶段的安全性，这与本次隔离目标直接相关：针对 post-test 清理按精确 job 名删除 pod 的方案，cherryblo 指出 "it may cause pods from other parallel tasks to be deleted."，作者回复 "Already rolled back"。另一线程中 cherryblo 对 pre-test bounded retry 给出 "Optimize logging, add delete" 的评价，同样被作者以 "Already rolled back" 回应；对 `|| true` 的改动，作者自评 "it is a useless change" 并回滚。最终合并版本因此收敛为最小改动的隔离方案。

- post-test 精确 job 名清理方案可能误删并行任务 pod (correctness): 该增强被回滚，保留原清理逻辑，避免误删其他并行运行的 pod。

- pre-test bounded retry 清理增强 (design): 该方案未进入合并版本，回滚后保持原有清理与等待逻辑。
- kubectl get pods 幂等性改动的必要性 (question): 改动被认定为无效并回滚，workflow 保持原样。

风险与影响

- 风险：
  - 模板与脚本同步风险：prepare_cm_data 依赖 pod 上存在 run-id label，必须与 5 个 jinja2 模板同步演进；未来新增 job 类型（如新的 green 变体）若漏加 label，该类型测试将收集不到 pod。
  - PVC 残留累积：收尾的 rm -rf 只在正常流程末尾执行，job 超时或被 kill 时可能残留 /root/.cache/tests/sglang-${run_id} 目录，长时间累积会占用共享盘空间。
  - Job 名长度限制：KUBE_JOB_NAME 从 test_type 换成 run_id 后变长，PR body 也提到 "within the length limit"，后续若再拼接其他标签需注意 k8s 63 字符上限。
  - 兼容性：run_id 参数缺省为空时所有行为与改动前一致，本地调试不受影响，风险较低。
- 影响：
  - 对用户：无任何产品功能影响，纯测试 /CI 基础设施改动。
  - 对系统：NPU nightly e2e 多机测试可在同一 k8s namespace 与共享 PVC 上并行运行，不再互相覆盖源码、污染 ConfigMap 或误删 pod，CI 稳定性显著提升。
  - 对团队：影响范围限定于使用 nightly-test-npu-e2e-multi-node.yml 流水线的 NPU 多机测试场景；调试者排查并发问题时定位更快。
  - 风险标记：模板与脚本需同步演进，PVC 残留累积风险，Job 名长度上限约束，无新增单测覆盖

关联脉络

- PR #37214 test: re-enable DSV4-Flash W8A8 8p nightly perf cases: 同属 NPU nightly 测试流水线稳定性建设，依赖相同的 NPU e2e 多机运行环境与测试脚本。
- PR #36459 [NPU] Fix evalscope accuracy parsing and add glm5_1 aime26 request timeout: 同为 NPU 测试工具链修复，本 PR 的 run_npu_e2e_test.py 与 workflow 改动正是该类测试的运行底座。