

PR #1920 完整报告

radixark/miles

fix(ci): stabilize SGLang and FSDP E2E

合并时间: 2026-07-29 09:46

原文链接: <http://prhub.com.cn/radixark/miles/pull/1920>

执行摘要

- 一句话: 修复 SGLang E2E 失败不可见与 FSDP 长任务误入 nightly
- 推荐动作: 值得精读。PR body 的 root cause 编号分析质量高, 把「失败为什么不可见」拆成 6 个独立根因并逐一修复, 是排查 CI 假性红绿的范本; resolve_policy 的「减法不是否决、显式请求并集兜底」语义值得作为 CI 策略设计参考。对需要在仓库中注册长时间 E2E 用例的工程师, 本 PR 的标签分类与 est_time 预算规则有直接指导意义。建议顺带确认 run-ci-fsdp 标签当前是否仍有归属用例。

功能与动机

PR body 明确列出 6 个根因: MODEL_PATH 把默认权重获取推迟到 SGLang 启动时; _wait_for_ready 在 fixture 清理前抛错导致服务进程无人回收; pytest.main 返回失败但 __main__ 丢弃退出码; est_time=300 低估 60-rollout FSDP 负载, 旧 runner 在 1800 秒超时时任务仍在推进; resolve_policy 把 long 纳入 nightly 标准范围; labels 把长 FSDP 用例分到 fsdp 或 short 范围。作者的目标是让 setup 失败在 CI 中对齐可见, 并让长任务只能经显式 run-ci-long / run-ci-all 选择, 同时保留显式请求的并集兜底语义。

实现拆解

1. SGLang 失败可见化: tests/e2e/sclang/test_chat_input_ids_equivalence.py 新增 _prepare_model(), 在 sclang_server fixture 中启动服务前用 snapshot_download 把 Qwen/Qwen3-0.6B 预下载到 /root/models/Qwen3-0.6B; MODEL_PATH 改为环境变量 SGLANG_E2E_MODEL_PATH 优先、默认本地路径兜底; __main__ 由 pytest.main(...) 改为 raise SystemExit(pytest.main(...)), 让直接执行脚本时的失败传导到进程退出码。
2. SGLang 服务清理: tests/e2e/sclang/utils/sclang_server.py 的 start_sclang_server 用 try/except (RuntimeError, TimeoutError) 包住 _wait_for_ready, 就绪失败时先 server.stop() 再 raise, 回收子进程与日志句柄; 由于失败发生在 fixture yield 之前, fixture finally 中的停止逻辑不会重复触发。
3. FSDP 长任务重分类: tests/e2e/short/test_qwen3_0.6B_fsdp_colocated_2xGPU.py 的 est_time 从 300 提到 3000 秒 (按 runner 规则获得约 3750 秒文件预算), 标签从 short 改为 long, 工作负载与 gsm8k 0.71 阈值不变; tests/e2e/fsdp/test_qwen3_0.6B_fsdp_distributed.py 标签从 fsdp 改为 long, 使其只能经显式长任务请求选择。

4. nightly 策略调整: tests/ci/ci_policy.py 的 resolve_policy 中 nightly 分支从 _ALL - {"ft-long"} 改为 _ALL - {"long", "ft-long"}; 减法语义不变 (不是逐用例否决), 显式 run-ci-long / run-ci-all 仍通过并集重新引入被减标签, docstring 同步改写为作用域列表。
5. 配套测试与文档: tests/ci/test/test_run_suite.py 更新 TestResolvePolicy 参数化期望、在 nightly alias CLI 测试中新增 'long' not in alias_policy 断言, 并把 nightly 范围用例重命名为 test_nightly_scope_excludes_long_and_ft_long_but_selects_ft_short; docs/ci/00-stage.md、docs/ci/01-label.md 同步 nightly 排除项与显式 run-ci-long 加回语义; .github/workflows/pr-test.yml 仅更新 cron 注释。7 个 commit 展示了标签策略的反复: 先加 fsdp 域标签、再退回 short、最终统一归入 long。

关键文件:

- tests/ci/ci_policy.py (模块 CI 策略; 类别 test; 类型 core-logic; 符号 resolve_policy): 本 PR 的策略核心: resolve_policy 的 nightly 分支从排除 ft-long 扩展为同时排除 long, 是「FSDP 长任务不进 nightly」的关键开关, subtraction 语义与显式请求并集优先级的 docstring 同步改写。
- tests/e2e/sglang/test_chat_input_ids_equivalence.py (模块 SGLang 用例; 类别 test; 类型 test-coverage; 符号 _prepare_model, slang_server): SGLang 失败可见化的主要载体: 新增 _prepare_model 预下载默认模型到 /root/models/Qwen3-0.6B, 并把 pytest.main 退出码通过 raise SystemExit 传播给 CI runner。
- tests/e2e/sglang/utils/sglang_server.py (模块 SGLang 服务; 类别 test; 类型 test-coverage; 符号 start_sglang_server, _wait_for_ready): 就绪失败清理逻辑: start_sglang_server 捕获 RuntimeError / TimeoutError 后先 server.stop() 再抛出, 修复服务进程与日志句柄泄漏。
- tests/ci/test/test_run_suite.py (模块 套件测试; 类别 test; 类型 test-coverage; 符号 test_nightly_scope_excludes_long_and_ft_long_but_selects_ft_short): 策略改动的主要测试证明: 更新 TestResolvePolicy 参数化期望、新增 nightly alias 不包含 long 的断言, 并把 nightly 范围用例重命名为 test_nightly_scope_excludes_long_and_ft_long_but_selects_ft_short。
- tests/e2e/short/test_qwen3_0.6B_fsdp_colocated_2xGPU.py (模块 FSDP 用例; 类别 test; 类型 test-coverage): est_time 从 300 秒提升到 3000 秒、标签从 short 改为 long, 让 60-rollout 负载获得约 3750 秒预算并退出 short/nightly/image 默认范围。
- tests/e2e/fsdp/test_qwen3_0.6B_fsdp_distributed.py (模块 FSDP 用例; 类别 test; 类型 test-coverage): 标签从 fsdp 改为 long, 使该 65-rollout 分布式 FSDP 用例只能经显式 run-ci-long / run-ci-all 选择, 而非通过 run-ci-fsdp 域标签进入范围。
- docs/ci/01-label.md (模块 CI 文档; 类别 docs; 类型 documentation): CI 标签语义文档同步: nightly 与 image 范围的减法说明、显式 run-ci-long 加回语义, 以及「长任务只打 long 标签」的注册约定。
- docs/ci/00-stage.md (模块 CI 文档; 类别 docs; 类型 documentation): stage 文档同步 nightly 默认范围排除 long 与 ft-long 的新语义。
- .github/workflows/pr-test.yml (模块 工作流; 类别 infra; 类型 infrastructure): 仅更新 nightly cron 注释, 说明默认范围排除 long 与 ft-long, 无行为变更。

关键符号: resolve_policy, _prepare_model, start_sglang_server, _wait_for_ready, test_nightly_scope_excludes_long_and_ft_long_but_selects_ft_short, test_nightly_alias_matches_explicit_cadence

关键源码片段

tests/ci/ci_policy.py

本 PR 的策略核心: resolve_policy 的 nightly 分支从排除 ft-long 扩展为同时排除 long, 是「FSDP 长任务不进 nightly」的关键开关, subtraction 语义与显式请求并集优先级的 docstring 同步改写。

```
def resolve_policy(cadence: str, raw_labels: set[str]) -> RunPolicy:
    # 分支优先级从高到低为 run-ci-all > nightly > run-ci-image;
    # 显式请求的 run-ci-<x> 标签最后再参与并集, 保证 run-ci-long
    # 与 run-ci-all 在 nightly 下仍能重新引入被减去的 long 标签
    requested = strip_run_ci_prefix(raw_labels) & set(KNOWN_LABELS)
    if "run-ci-all" in raw_labels:
        # run-ci-all 覆盖全部注册标签, 不受任何减法影响
        scope = set(KNOWN_LABELS)
    elif cadence == NIGHTLY_CADENCE:
        # nightly 默认范围减去 long 与 ft-long: 长耗时 FSDP 任务
        # 不再默认进入 nightly, 只有显式 run-ci-long 才执行
        scope = set(KNOWN_LABELS) - {"long", "ft-long"}
    elif "run-ci-image" in raw_labels:
        # image 构建范围维持原语义: 排除 long 与全部 FT 标签
        scope = set(KNOWN_LABELS) - {"long", "ft-short", "ft-long"}
    else:
        scope = set()
    return RunPolicy(
        cadence=cadence,
        include_labels=frozenset(scope | requested),
        bypass_fastfail=cadence == NIGHTLY_CADENCE or "bypass-fastfail" in raw_labels,
    )
```

tests/e2e/sglang/test_chat_input_ids_equivalence.py

SGLang 失败可见化的主要载体: 新增 _prepare_model 预下载默认模型到 /root/models/Qwen3-0.6B, 并把 pytest.main 退出码通过 raise SystemExit 传播给 CI runner。

```
DEFAULT_MODEL_ID = "Qwen/Qwen3-0.6B"
DEFAULT_MODEL_PATH = "/root/models/Qwen3-0.6B"
# 环境变量提供本地权重路径时优先使用, 便于 CI 复用已缓存的模型
MODEL_PATH_OVERRIDE = os.environ.get("SGLANG_E2E_MODEL_PATH")
MODEL_PATH = MODEL_PATH_OVERRIDE if MODEL_PATH_OVERRIDE is not None else
DEFAULT_MODEL_PATH
```

```
def _prepare_model() -> None:
```

```
# 只有使用默认模型路径时才预下载权重；显式设置
# SGLANG_E2E_MODEL_PATH 时假定环境已备好权重，不重复拉取
if MODEL_PATH_OVERRIDE is None:
    snapshot_download(DEFAULT_MODEL_ID, local_dir=DEFAULT_MODEL_PATH)
```

```
@pytest.fixture(scope="module")
def sglang_server():
    # 在启动 SGLang 之前完成权重准备，把网络下载失败提前暴露在
    # setup 阶段，而不是让服务在启动超时后才发现模型缺失
    _prepare_model()
    server = start_sglang_server(model_path=MODEL_PATH)
    try:
        yield server
    finally:
        server.stop()
```

```
if __name__ == "__main__":
    # pytest.main 返回退出码但不会自动影响进程退出状态；
    # raise SystemExit 让直接执行脚本时的失败能正确传给 CI runner
    raise SystemExit(pytest.main([__file__, "-v"]))
```

tests/e2e/sglang/utils/sglang_server.py

就绪失败清理逻辑：start_sglang_server 捕获 RuntimeError / TimeoutError 后先 server.stop() 再抛出，修复服务进程与日志句柄泄漏。

```
process = subprocess.Popen(cmd, stdout=log_file, stderr=subprocess.STDOUT, env=env)
server = SGLangServer(process=process, host=host, port=port, log_path=log_path, _log_file=
log_file)

try:
    _wait_for_ready(server, timeout_secs=startup_timeout_secs)
except (RuntimeError, TimeoutError):
    # 就绪失败时主动回收子进程与日志句柄，避免 setup 报错后
    # 残留 SGLang 服务进程拖住后续用例或污染其他测试
    server.stop()
    raise
return server
```

评论区精华

本 PR 没有实质性的 review 讨论：审核人 Zhichenzzz 直接 APPROVED 且未留评论；仓库唯一一条评论来自 gemini-code-assist 机器人，声明其 GitHub 代码审查服务已停用。真正的设计权衡写在 PR body 的 Review Focus 与 7 个 commit 的演进中：作者要求重点审视 **MODEL_PATH** 解析、**start_sglang_server** 的就绪清理、以及 **resolve_policy** 对 long 的处理；2xGPU FSDP 用例的标签经历「加 fsdp 域标签 → 退回 short → 最终定为 long」三次调整，反映了对「能否被 run-ci-fsdp 选中」与「是否过长混入 short/nightly」之间的取舍，最终

选择了语义更严格的 long。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 长任务预算未实测：est_time=3000 是估算值，PR 明确说明未在 2xH200 上重跑完整 60-rollout 负载；若真实耗时仍超过约 3750 秒，用例依旧会被 runner 超时终止，只是把失败点推后。
2. run-ci-fsdp 可能失配：两个 FSDP 用例标签都改为 long 后，run-ci-fsdp 可能不再命中任何用例（仓库中未见其他 fsdp 标签用例），依赖该标签的 PR 会静默失去 FSDP 覆盖。
3. nightly 默认覆盖收窄：nightly 范围排除 long 后，依靠 nightly 做长任务回归的场景需要显式 run-ci-long；新用例若不遵守「长任务只打 long 标签」的约定，也可能漏出覆盖。
4. SGLang 启动路径变更：snapshot_download 依赖 CI 节点网络与 HuggingFace 可达性，且 /root/models 需可写并持久化；环境异常会在 setup 阶段快速失败，这是本次预期行为，但失败原因指向网络而非代码。
5. 清理逻辑安全性：就绪失败发生在 fixture yield 之前，fixture finally 不会重复 stop，逻辑上安全；若未来有人把 stop 逻辑提前，需注意幂等性。

- 影响：

1. CI 可靠性：SGLang 用例从「等待约 900 秒后假成功」变为「setup 阶段快速失败并清理进程」，直接文件运行与 pytest 运行的失败行为对齐。
2. 覆盖范围：nightly 与 run-ci-image 默认范围都排除 long，两个 FSDP 长任务只经显式 run-ci-long / run-ci-all 执行；对测试注册标签的语义要求更严格。
3. 协作与文档：docs/ci/00-stage.md 与 01-label.md 同步了新语义，后续注册用例的工程师以文档为准。
4. 影响面：全部为测试与 CI 配置变更，无产品代码、schema 或部署变更，对用户侧无直接影响。 - 风险标记：FSDP 长任务预算未实测，run-ci-fsdp 可能失配用例，nightly 默认覆盖收窄，SGLang 启动路径变更

关联脉络

- PR #1795 Bump sglang to v0.5.16: 同一 SGLang E2E CI 基础设施线：本 PR 的 SGLang 用例与镜像构建依赖其升级后的依赖栈，两者共同维护 sglang E2E 的稳定性。
- PR #1961 docker: keep the cu12 dependency markers after checking out sglang-miles: 同为 CI/ 构建稳定性修复，聚焦 docker 镜像构建与 sglang 依赖一致性，与本 PR 的「让 CI 故障及时正确暴露」目标同源。