PR #34074 完整报告

sgl-project/sglang

[CI] Move tests onto the right CI stages

合并时间：2026-09-01 03:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34074

执行摘要

- 一句话：66 项低频测试移入 weekly，workflow 矩阵化
- 推荐动作：值得精读。这是 sglang CI 体系的一次系统性梳理：测试频率分层策略、无效注册清理、GitHub Actions matrix 化与 runner_filter 下沉，以及 compute_partitions.py 的 matrix 静态解析。对维护多 runner 测试矩阵的团队有直接参考价值。重点关注：注册即声明 (suite, stage, runner) 的约定、静默丢失的防御（白名单校验 + 人工核对）、以及 " 从不执行的注册比报错的注册更危险 " 这一运维洞察。

功能与动机

PR body 明确指出："Nightly is for evaluation and perf regression checking -- numbers you want tracked daily. A feature or UT test that breaks a couple of times a year does not need a daily slot; knowing which week it broke is enough to bisect." 这些低频测试此前占用 nightly runner，且部分 CPU 注册因过滤器不匹配从未执行；作者希望按测试价值分层，把 nightly 槽位留给真正需要每日追踪的评估与性能回归。

实现拆解

1. 测试注册迁移（66 文件 / 70 注册）：将 test/registered/ 下 8-gpu-models（GLM-4.6、MiniMax-M2.5、Mistral-Large3、Ring-2.5-1T）、cuda_graph/piecewise（PCG + 投机解码）、kernels/benchmark/kv_canary、lora、attention 确定性测试、model_loading、backends、eval、scheduler 及 debug_utils 等文件的注册从 stage="nightly" 改为 stage="weekly"，部分同时调整套件名。原因：这些测试一年只坏几次，周级频率足够二分定位，且不占用每日 eval/perf 槽位。
2. 清理矛盾与无效注册：debug_utils 45 个文件此前同时携带 base-a-test-cpu 的 nightly=True（永不执行，因为 _pr-test-stage-cpu.yml 跑 run_suite.py 时不带 --nightly）与 base-c-test-cpu 注册（会在 per-commit Xeon 箱上执行）；统一为单条 stage="weekly" 的 CPU 注册，把 CPU-only debug 工具从 per-commit Xeon 移除。删除 test/registered/utils/test_model_file_verifier.py：其真实服务器测试依赖 vllm._custom_ops，CPU runner 上无法启动。
3. 裁剪 nightly eval 模型列表：python/sglang/test/test_utils.py 中 DEFAULT_MODEL_NAME_FOR_NIGHTLY_EVAL_* 常量移除已被其他套件覆盖的模型（如 Llama-3.1-8B、Qwen3-8B 等），避免同一模型回归在多个套件重复报警；test_text_models_gsm8k_eval.py 与 AMD test_gsm8k_eval_amd.py 同步删除对应模型组。

4. Weekly workflow 矩阵化与新增 CPU workflow：weekly-test-nvidia.yml 从每 runner 一个手工 job 折叠为单个 weekly-test job + strategy.matrix.include 6 行（1-gpu-large、2-gpu-large、4-gpu-h100、4-gpu-b200、8-gpu-h200、8-gpu-b200），self_name 与 run_timeout_minutes 使用 ${{ matrix.}}；新增 weekly-test-cpu.yml（每周日 UTC 00:00 调度 + workflow_dispatch），因为 uses: 不接受表达式，CPU 无法作为 Nvidia matrix 的一行。两个 weekly workflow 都复用 per-commit 相同的共享 stage _pr-test-stage.yml，保证 PR 内与 weekly 行为一致。
5. runner_filter 下沉与分区解析适配：runner_filter 输入从每个 caller 移到共享 _pr-test-stage.yml 的 job if（job 级条件可读 inputs 但读不到 matrix，因此 caller 无法本地过滤）；输入类型从 choice 改为 string（on: 块不接受表达式，options 会成为第二份需同步的 matrix）。scripts/ci/utils/compute_partitions.py 新增 _resolve_matrix_refs，在静态读取时把 ${{ matrix.key }} 替换为 strategy.matrix.include 行内实际值；test/run_suite.py 的 OTHER_SUITES 声明 6 个新的 weekly CUDA 套件名与 weekly-test-cpu 有效。
6. 验证：weekly dispatch 全 6 行展开并运行真实测试，6/7 绿（唯一失败 moe/test_hybrid_dp_ep_tp_mtp.py 与本 PR 无关，main 的 weekly 同样失败）；逐一核对每个 scheduled 注册落在声明的 job；debug_utils 测试以 sglang[test] 本地通过，确保首个 weekly CPU 运行不红。

关键文件：

- test/registered/utils/test_model_file_verifier.py（模块 文件校验；类别 test；类型 deletion；符号 _FakeModelTestCase, _RealModelTestCase, TestModelFileVerifier, test_detect_bit_rot）：整个测试文件被删除（349 行）。其 _RealModelTestCase 要启动真实服务器，但 CPU runner 上未安装 vllm._custom_ops，服务器根本起不来；这也是 " 注册看起来在跑、实际永不通过 " 的典型案例。
- .github/workflows/weekly-test-cpu.yml（模块 周测工作流；类别 infra；类型 infrastructure）：新增的 CPU-only weekly 工作流，承接从 per-commit Xeon 箱上移出的 45 个 debug_utils 测试；独立文件的原因是用例 uses: 不接受表达式，CPU 无法并入 Nvidia matrix。
- scripts/ci/utils/compute_partitions.py（模块 分区脚本；类别 infra；类型 infrastructure；符号 _resolve_matrix_refs, load_run_timeouts）：唯一有实质算法逻辑的变更：新增 _resolve_matrix_refs，在静态读取分区与超时时自行解析 ${{ matrix.* }} 表达式，支撑 weekly-test-nvidia.yml 的 matrix 化。
- .github/workflows/weekly-test-nvidia.yml（模块 周测工作流；类别 infra；类型 infrastructure）：weekly 工作流矩阵化的核心载体：6 类 runner 从逐个手写 job 收敛为一个 matrix job，self_name、超时全部由 include 行驱动；同时把 runner_filter 从 choice 改为 string。
- test/registered/eval/test_text_models_gsm8k_eval.py（模块 评估测试；类别 test；类型 test-coverage；符号 TestNightlyGsm8KEval）：GSM8K 全模型 eval 从 nightly 移入 weekly 并裁剪模型列表，是测试频率分层策略的典型代表；显示 eval 类回归也接受周级发现。

- python/sglang/test/test_utils.py（模块 测试工具；类别 test；类型 test-coverage；符号 DEFAULT_MODEL_NAME_FOR_NIGHTLY_EVAL_TP2, DEFAULT_MODEL_NAME_FOR_NIGHTLY_EVAL_FP8_TP1, DEFAULT_MODEL_NAME_FOR_NIGHTLY_EVAL_FP8_TP2）：集中定义 nightly eval 模型常量，本 PR 在此裁剪重复模型，并加注释说明 " 刻意省略 " 的设计原则，影响多个 eval 测试文件的模型组构成。
- test/run_suite.py（模块 套件校验；类别 test；类型 test-coverage；符号 OTHER_SUITES）：作为 suite 白名单校验的入口，必须同步声明新出现的 weekly 套件名，否则校验会 fail fast；这是防止测试静默丢失的防御层。

关键符号：_resolve_matrix_refs, load_run_timeouts

关键源码片段

.github/workflows/weekly-test-cpu.yml

新增的 CPU-only weekly 工作流，承接从 per-commit Xeon 箱上移出的 45 个 debug_utils 测试；独立文件的原因是用例 uses: 不接受表达式，CPU 无法并入 Nvidia matrix。

```
# Weekly CPU tests. Holds the CPU-only unit tests that are not worth a
# per-commit slot -- debug tooling (dump comparator, source patcher) whose
# breakage is fine to notice once a week.
#
# Separate from weekly-test-nvidia.yml because CPU goes through a different
# reusable workflow, and `uses:` takes no expressions -- so it cannot be another
# row in that file's matrix. Both run the same shared stage their per-commit
# counterparts do, so a test behaves identically in a PR and in the weekly run.
name: Weekly Test (CPU)

on:
  schedule:
    - cron: '0 0 * * 0' # Sunday 00:00 UTC, alongside the Nvidia weekly run
  workflow_dispatch:

concurrency:
  group: weekly-test-cpu-${{ github.ref }}
  cancel-in-progress: true

permissions:
  actions: write
  contents: read
  issues: read
  pull-requests: read

jobs:
  # run_all_tests skips the paths-filter, so every test runs regardless of what
  # the last commit touched.
  check-changes:
    uses: ./.github/workflows/_pr-test-check-changes.yml
```

```
    with:
      pr_test_yml: '.github/workflows/weekly-test-cpu.yml'
      run_all_tests: true
      force_continue_on_error: true
    secrets: inherit

  # No rust_ext_artifact: nothing builds one here, so the stage falls back to
  # its cache and compiles on a miss.
  weekly-test-cpu:
    needs: check-changes
    if: github.repository == 'sgl-project/sglang'
    uses: ./.github/workflows/_pr-test-stage-cpu.yml
    with:
      self_name: weekly-test-cpu
      check_changes: ${{ toJson(needs.check-changes.outputs) }}
      caller_inputs: ${{ toJson(inputs) }}
      partitions: ${{ needs.check-changes.outputs.partitions }}
      run_timeout_minutes: '90'
    secrets: inherit
```

scripts/ci/utils/compute_partitions.py

唯一有实质算法逻辑的变更：新增 _resolve_matrix_refs，在静态读取分区与超时时自行解析 ${{ matrix.* }} 表达式，支撑 weekly-test-nvidia.yml 的 matrix 化。

```
# 匹配形如 ${{ matrix.runner_config }} 的表达式
_MATRIX_REF = re.compile(r"\$\{\{\s*matrix\.([A-Za-z_][A-Za-z0-9_]*)\s*\}\}")


def _resolve_matrix_refs(value, row: dict):
    """把 ${{ matrix.key }} 替换为 matrix.include 行里的实际值。

    在 weekly-test-nvidia.yml 中，self_name 与 run_timeout_minutes 都是
    ${{ matrix.* }} 表达式，GitHub 会在 dispatch 时解析；而这里的静态读取
    发生在 dispatch 之前，必须自己完成同样的替换，才能拿到每个 suite 的
    真实超时值。
    """
    return _MATRIX_REF.sub(lambda m: str(row[m.group(1)]), str(value))


def load_run_timeouts(pr_test_yml_path: str) -> dict:
    """从 pr-test*.yml 读取 self_name -> run_timeout_minutes 映射。

    兼容两种 job 形态：普通单个 job，以及带 strategy.matrix 的 job。
    非 matrix 路径与旧逻辑完全一致。
    """
    # 省略文件解析与 job 遍历的公共部分，核心分支如下
    for job_id, job in jobs.items():
        if not isinstance(job, dict) or job.get("uses") not in _REUSABLE_STAGE_USES:
            continue
```

```
        with_ = job.get("with") or {}
        # matrix 行缺省时视为单个 job：rows 只含一个 None
        rows = (
            ((job.get("strategy") or {}).get("matrix") or {}).get("include")
            or [None]
        )
        for row in rows:
            if row is None:
                suite = with_.get("self_name", job_id)
                timeout = with_["run_timeout_minutes"]
            else:
                # matrix 行存在时，把表达式逐个替换成该行的实际值
                suite = _resolve_matrix_refs(with_.get("self_name", job_id), row)
                timeout = _resolve_matrix_refs(with_["run_timeout_minutes"], row)
            timeouts[suite] = int(timeout)
    # 若没有匹配到任何 job，直接报错，避免静默丢失超时配置
    if not timeouts:
        raise RuntimeError(
            f"load_run_timeouts: no jobs matched uses in {_REUSABLE_STAGE_USES!r} "
            f"in {pr_test_yml_path}"
        )
    return timeouts
```

评论区精华

该 PR 没有正式的 review comment，但 PR body 中作者详细论证了多个设计决策，值得作为讨论精华引用：

> "Nightly is for evaluation and perf regression checking -- numbers you want tracked daily. A feature or UT test that breaks a couple of times a year does not need a daily slot; knowing which week it broke is enough to bisect."
>
> "The nightly=True ones never executed. _pr-test-stage-cpu.yml runs run_suite.py --hw cpu --suite base-a-test-cpu with no --nightly, and the filter is an equality test on that flag."
>
> "A job-level if can see github / needs / vars / inputs but not matrix, so a caller that declares its runners as a matrix cannot filter there."
>
> "uses: takes no expressions, so CPU cannot be another row in the Nvidia matrix."
>
> "_RealModelTestCase classes launch a real server that cannot start on a CPU runner -- RotaryEmbedding.__init__ imports vllm._custom_ops, which is not installed there."

核心结论：测试按价值分层（per-commit / nightly / weekly）、CPU 与 Nvidia 必须分文件、runner_filter 下沉到共享 stage、无效注册宁可删除也不要留一个 " 看起来在跑 " 的假象。

- 测试频率策略：weekly vs nightly (design): 将 66 个低频测试移动到 weekly，为 nightly 腾出资源；这是按测试价值分层的明确决策。

- runner_filter 下沉与 matrix 化限制 (design): 采用共享 job if + string 输入；compute_partitions.py 静态解析 matrix 引用。
- 为何删除 test_model_file_verifier.py (correctness): 删除文件而非移动到 weekly（即使 weekly 也无法运行）；保留假注册只会掩盖问题。

风险与影响

- 风险：
  1. 回归发现延迟：66 项测试从每日变为每周，若其中某项实际属于高频回归，最长 7 天才会暴露；这是对 " 低频测试 " 的刻意取舍，但依赖作者的分类判断。
  2. 静默停跑风险：注册指向 (stage, runner_config) 对若没有对应 workflow job，测试会无声消失（正是 debug_utils 之前的状况）。作者已逐一核对，但未来新增 weekly 注册或改动 workflow 时仍可能踩中；test/run_suite.py 的 suite 白名单校验能兜底一部分，但不能覆盖 (stage, runner_config) 与 job 的完整对应。
  3. 删除测试文件：test_model_file_verifier.py 被整个删除，model_file_verifier 工具目前没有自动化覆盖；若未来在 GPU/ 真实模型路径上复用该工具，需要重新补测试。
  4. matrix 静态解析新逻辑：compute_partitions.py 的 _resolve_matrix_refs 用正则替换 ${{ matrix.key }}，若 include 行缺少某个 key 会抛 KeyError；当前 6 行都完整，但新增 runner 配置时需同步保证 key 齐全。
  5. eval 覆盖裁剪：gsm8k 模型列表缩水（移除与其他套件重复的模型），依赖 " 回归会在对应套件浮现 " 的假设，若某模型只在这次 eval 中出现，会失去覆盖。 - 影响：对用户 / 生产：无运行时影响，纯测试与 CI 配置变更。对 CI 资源：nightly GPU 槽位释放约 21 个低频测试；per-commit Xeon CPU 箱不再跑 45 个 debug_utils 文件；每周日统一承载 6 类 Nvidia runner + CPU 测试。对团队：新增机器只需在 matrix 加一行；调整测试频率只需改 stage=；新增测试无额外 workflow 改动。CI 架构从 " 每 runner 一个 job" 收敛为 " 一个 matrix job"，可维护性显著提升。对开发流程：低频回归最迟一周发现，需要接受这一延迟作为成本。 - 风险标记：覆盖频率降为周级，静默停跑风险，删除测试文件，matrix 静态解析

关联脉络

- PR #36814 xpu: move prefill-only model tests to the nightly-xpu-1-gpu grid: 同类操作：将测试迁移到更高层的 CI 网格（per-commit → nightly），说明仓库正在建立按频率分层的 CI 体系。
- PR #37203 [CI] Speed up lint: cache pre-commit envs + mint, drop redundant work: 同属 CI 基础设施优化方向，与本 PR 一起体现对 CI 效率与资源利用的持续投入。
- PR #37214 test: re-enable DSV4-Flash W8A8 8p nightly perf cases: 涉及 nightly 测试槽位管理；本 PR 释放 nightly 槽位后，这类真正需要每日追踪的性能用例有了更充足的空间。