

PR #38632 完整报告

sgl-project/sglang

[AMD][CI] Consolidate AMD workflows and retire ROCm 7.0 CI

合并时间: 2026-09-15 23:45

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38632>

执行摘要

- 一句话: AMD CI 合并为一对 workflow 并退役 ROCm 7.0
- 推荐动作: 值得精读, 尤其适合负责多硬件平台 CI 的工程师。三个设计点值得借鉴: 一是用「单一 workflow + rocm_version matrix + 调度 cron 区分版本组合」替代「每版本一套 workflow」的重复布局, 将约 3900 行重复配置收敛为一份; 二是 `_filter_legacy_amd_job_rows` 的「历史 job 名识别 + 旧数据不污染当前 streak」的过滤策略, 是 workflow 改名场景下的实用技巧; 三是通过 `fromJson` 让上游 `check-changes` 输出版本数组、下游所有测试 job 统一展开矩阵的编排方式。NPU、Intel 等平台 CI 若面临类似的多版本多 workflow 重复问题, 可直接复用此模式。

功能与动机

PR body 明确指出目标是「Consolidate AMD workflows to one pair of yaml files」并「Removes the duplicate ROCm 7.0 workflows」, 让每个受支持的 ROCm 版本共用同一套标准 PR/Nightly 流程。Issue 评论中 michaelzhang-ai 补充了关键背景: 「Rocm7 is retired and still takes lots of computation resource」, 即 ROCm 7.0 已不再需要维护, 但其定时 Job 仍持续消耗 GPU 计算资源, 因此需要尽快合并并释放资源。

实现拆解

实现按五个步骤完成, 核心是「一套 workflow + rocm_version 矩阵」的架构替换原先「每版本一套 workflow」的重复布局:

1. 重写主 PR workflow (`.github/workflows/pr-test-amd.yml`): workflow 名从「PR Test ROCm 7.0 (AMD)」改为「PR Test (AMD)」, 新增 `rocm_version` 输入 (`all` / `rocm10` / `rocm724` / `rocm720`, 默认 `rocm10`); 调度 cron 改为 `0 */12 * * *` 每日两次跑 ROCm 10、`30 17 * * *` 每日一次跑 ROCm 7.2.4 与 7.2.0; `check-changes` job 新增 `rocm_versions` 输出, 按事件类型推导版本数组; 各测试 job (`sgl-kernel-unit-test-amd`、`stage-a-test-1-gpu-small-amd`、`jit-kernel-unit-test-amd` 等) 全部改为 `strategy.matrix` + `fromJson(needs.check-changes.outputs.rocm_versions)` 展开, 并把 `--rocm-version` 传给容器启动脚本。
2. 合并 Nightly workflow (`.github/workflows/nightly-test-amd.yml`): 吸收原 `nightly-test-amd-rocm720.yml` 的 2392 行内容, 改为同一文件内按 `rocm_version` 矩阵跑 ROCm 10 / 7.2.4 / 7.2.0, 同时删除重复的 `nightly-test-amd-rocm720.yml` 与 `pr-test-amd-rocm720.yml` (合计约 3900 行)。

3. 退役 ROCm 7.0 产物链：删除 `release-docker-amd-nightly.yml` (ROcm 7.0 专属镜像构建)，并在 `release-docker-amd.yml` 的 `matrix` 与 `rocm_tag` 分支中移除 `rocm700` 选项，产品镜像从 8 个 flavor (含 7.0 双架构) 减为 6 个 (ROcm 10 / 7.2.4 / 7.2.0 × gfx942 / gfx950)。
4. 同步监控与分析配套：`amd-ci-job-monitor.yml` 删除 `pr_rocm700_jobs` / `nightly_rocm700_jobs` 两组矩阵报告 job，`workflow` 参数统一改写为 `pr-test-amd.yml` / `nightly-test-amd.yml`；`scripts/ci_monitor/ci_failures_analysis.py` 中 `SGLangFailuresAnalyzer.main` 的 `workflow_filter` 全部从 `pr-test-amd-rocm720.yml` / `nightly-test-amd-rocm720.yml` 切换为新文件名，同时扩展 `_filter_legacy_amd_job_rows` 的 `legacy_callers` 集合 (新增 `call-nightly-amd-rocm720` 与 `call-pr-test-amd-rocm720`) 并增加 `rocm700` 行跳过逻辑，防止旧名与历史行污染 `streak` 统计；对应测试文件 `test_ci_failures_analysis.py` 随之删除。
5. 周边引用与脚本更新：`amd-aiter-scout.yml` 由同时调用 4 个 workflow 简化为 2 个调用并传 `rocm_version: all`；`scripts/ci/amd/amd_ci_start_container.sh` 与 `amd_ci_start_container_disagg.sh` 增加 `--rocm-version` 参数支持指定镜像；`pr-states.yml`、`release-branch-cut.yml`、`bot-bump-sglang-version.yml`、`pr-test-amd-extra.yml` 与 `.claude/skills/babysit-pr-to-pass-ci/SKILL.md` 中的 workflow 文件名引用同步更新。

关键文件：

- `.github/workflows/pr-test-amd.yml` (模块 CI 流程；类别 infra；类型 infrastructure)：AMD CI 合并后的主入口 workflow，从 ROCm 7.0 影子流程升级为标准多版本矩阵流程，新增 `rocm_version` 输入、双 cron 调度与全 job 矩阵展开，是本次变更的核心载体。
- `.github/workflows/nightly-test-amd.yml` (模块 CI 流程；类别 infra；类型 infrastructure)：合并后的 Nightly 主文件，吸收原 `nightly-test-amd-rocm720.yml` 的 2392 行内容并改为多版本矩阵，是 ROCm 10 / 7.2.4 / 7.2.0 旧版本回归的主阵地。
- `scripts/ci_monitor/ci_failures_analysis.py` (模块 CI 监控；类别 source；类型 core-logic；符号 `_filter_legacy_amd_job_rows`, `SGLangFailuresAnalyzer.init`, `main`)：CI 连续失败分析器的核心逻辑文件：`workflow_filter` 全部切换到新命名，`_filter_legacy_amd_job_rows` 扩展 `legacy_callers` 并新增 `rocm700` 历史行过滤，直接决定合并后 `streak` 统计的准确性。
- `scripts/ci_monitor/test_ci_failures_analysis.py` (模块 CI 监控；类别 test；类型 deletion；符号 `TestFilterLegacyAmdJobRows`, `test_drops_legacy_names_and_nested_utilities`, `test_keeps_current_flavors_and_nested_callers_separate`)：随 `ci_failures_analysis.py` 过滤逻辑重构被整体删除的测试文件；其断言基于旧命名集合，已无法覆盖新的 `legacy_callers` 逻辑。
- `.github/workflows/pr-test-amd-rocm720.yml` (模块 CI 流程；类别 infra；类型 deletion)：被退役的 ROCm 7.2 主 PR workflow，删除 1506 行；其全部能力收敛进合并后的 `pr-test-amd.yml`。
- `.github/workflows/nightly-test-amd-rocm720.yml` (模块 CI 流程；类别 infra；类型 deletion)：被退役的 ROCm 7.2 Nightly workflow，删除 2392 行；是本次删除量最大的文件，其矩阵内容整体移入 `nightly-test-amd.yml`。

- `.github/workflows/amd-ci-job-monitor.yml` (模块 CI 监控; 类别 infra; 类型 infrastructure) : AMD CI 状态监控 workflow 的配套改造: 删除两套 ROCm 7.0 影子报告 job 与输出, workflow 引用统一指向新文件。
- `.github/workflows/amd-aiter-scout.yml` (模块 CI 流程; 类别 infra; 类型 infrastructure) : 每周 AITER 回归从 4 个 workflow 调用简化为 2 个, 通过 `rocm_version: all` 让下游矩阵自动展开三版本, 是合并模式在 Scout 场景的直接体现。
- `.github/workflows/release-docker-amd.yml` (模块 镜像发布; 类别 infra; 类型 infrastructure) : 产品 Docker 镜像发布矩阵移除 rocm700, 从 8 个 flavor 收敛为 6 个, 是 ROCm 7.0 退役在发布链路上的落点。
- `scripts/ci/amd/amd_ci_start_container.sh` (模块 CI 脚本; 类别 infra; 类型 infrastructure) : CI 容器启动脚本新增 `--rocm-version` 参数, 是 workflow 矩阵能按版本拉取镜像的关键配套改动。

关键符号: `_filter_legacy_amd_job_rows`, `SGLangFailuresAnalyzer.get_recent_runs`, `SGLangFailuresAnalyzer.main`

关键源码片段

`scripts/ci_monitor/ci_failures_analysis.py`

CI 连续失败分析器的核心逻辑文件: `workflow_filter` 全部切换到新命名, `_filter_legacy_amd_job_rows` 扩展 `legacy_callers` 并新增 rocm700 历史行过滤, 直接决定合并后 `streak` 统计的准确性。

```
def _filter_legacy_amd_job_rows(job_data: Dict[str, Dict]) -> Dict[str, Dict]:
    """丢弃 AMD workflow 规范化前的旧 job 名, 避免历史行污染当前 streak."""

    # AMD workflow 合并后, 以下外层 caller 名称已被新命名取代。
    # 仍要识别它们的旧拼写, 防止旧记录并入当前连续失败统计。
    legacy_callers = {
        "call-nightly-amd-rocm720",
        "call-pr-test-amd-extra-rocm720",
        "call-pr-test-amd-rocm720",
    }
    filtered = {}
    for full_name, data in job_data.items():
        # job 全名按 "/" 分层: 外层是调用链, 末段是叶子 job 名。
        # 只要外层出现任一旧 caller, 整行直接丢弃。
        name_parts = full_name.split("/")
        if any(part in legacy_callers for part in name_parts[:-1]):
            continue

        leaf_name = name_parts[-1]
        # 管理类叶子 job 不参与测试统计: 等待门、gate、finish 汇总等
        if leaf_name.startswith(("wait-for-stage-a-amd", "wait-for-stage-b-amd")):
            continue
        if leaf_name in {
            "call-gate", "call-pr-test-amd-extra", "check-all-jobs",
```

```

"check-changes", "pr-gate", "pr-test-amd-extra-finish",
"pr-test-amd-finish",
"pr-test-amd-rocm720-finish", # 规范化前的旧 teardown job
}:
    continue

# 旧命名风格是叶子名带 -rocm<digits> 后缀，但后缀版本不可信
# （例如 -rocm720 的 PR job 实际可能跑 rocm724），一律丢弃。
stem = leaf_name.split(" ", 1)[0]
_, separator, version = stem.rpartition("-rocm")
if separator and version.isdigit():
    continue

# 中间态 nightly 命名只有 ROCm flavor，没有 runner；
# 当前标准命名必须同时包含 flavor 与 runner。
if leaf_name.endswith("-") and " " in leaf_name:
    details = leaf_name.rsplit(" ", 1)[1][:-1]
    if details.startswith("rocm") and details[4:].isdigit():
        continue

# 合并后的 workflow 复用了已退役 ROCm 7.0 workflow 的历史记录，
# 在 lookback 窗口内不能让这些旧行计入多版本当前 streak。
if details.split("-", 1)[0] == "rocm700":
    continue

filtered[full_name] = data

return filtered

```

评论区精华

本 PR 的 review 评论为空，三位 reviewer (HaiShaw、michaelzhang-ai、yctseng0211) 直接 APPROVED，yctseng0211 评价「LGTM, AMD specific change」，说明变更范围被明确限定在 AMD CI 域内。核心讨论集中在 Issue 评论中：michaelzhang-ai 请求尽快合并，理由是「Rocm7 is retired and still takes lots of computation resource」，并 @ 了多位维护者推动合入。此外，commit 历史中的拆分决策也值得注意：[\[AMD\]\[CI\] Split workflow reference cleanup into follow-up](#) 表明作者有意将 workflow 引用的后续清理与本次合并主变更解耦，避免一次变更耦合过多清理工作。

- ROCm 7.0 退役与合并紧迫性 (design): 三位 reviewer 全部 APPROVED，yctseng0211 以「LGTM, AMD specific change」确认变更范围可控，PR 随后合入。
- workflow 引用清理拆分策略 (design): 后续多个 commit (pr-states.yml、release-branch-cut.yml、bot-bump-sglang-version.yml 等) 实际仍纳入本 PR 同步完成，拆分意图在最终合入版本中被吸收。

风险与影响

- 风险:

1. PR 阶段多版本覆盖下降：合并后普通 pull_request 事件只默认跑 ROCm 10 (rocm_versions 推导中 pull_request fall through 到 rocm10) ， ROCm 7.2.4 / 7.2.0 的回归验证仅靠每日定时调度覆盖。若某 PR 修改了仅在旧版本暴露的兼容性问题，PR 门禁无法拦截，属于有意为之的权衡但需要团队知晓。
2. 监控 streak 统计正确性：ci_failures_analysis.py 的 workflow_filter 从 pr-test-amd-rocm720.yml 切换到 pr-test-amd.yml 后，GitHub Actions 历史 lookback 窗口内的旧 workflow 行将不再被拉取；同时 _filter_legacy_amd_job_rows 对 rocm700 的过滤依赖 job 名括号内第一个字段恰好为 rocm700，若未来 job 命名再次调整，存在漏过滤导致 streak 虚高或误合并的风险。
3. 引用断裂风险：一次性删除 3 个 workflow 共约 4150 行，任何未同步的外部引用 (bot、其他 workflow 的 uses:) 都会导致运行时错误「workflow does not exist」。本 PR 已同步 pr-states.yml、release-branch-cut.yml、bot-bump-sglang-version.yml，但仓库内是否还有其他引用点 (如文档、脚本) 无法从本次 diff 完全确认。
4. AITER Scout 资源放大：amd-aiter-scout.yml 现在对 nightly-test-amd.yml 传 rocm_version: all，每周会同时展开三个版本的完整 nightly，即使并行执行，GPU 占用总时长仍明显高于此前单版本模式。- 影响：影响范围集中在 AMD CI 基础设施，对运行时产品代码零影响。对开发者而言，PR 页面上的 AMD 检查项名称从「PR Test ROCm 7.0 / ROCm 7.2」变为统一的「PR Test (AMD)」，调度节奏从 ROCm 7.0 每两天一次变为 ROCm 10 每日两次；对 AMD CI 维护者而言，后续新增或调整 ROCm 版本只需修改 pr-test-amd.yml / nightly-test-amd.yml 的矩阵与 cron，不再需要复制整套 workflow；对 CI 监控系统而言，job 名的 streak 统计跨版本合并方式发生改变，ci_failures_analysis.py 的过滤逻辑需要持续跟随命名规范。总体影响属于「工程流程收敛」，显著降低维护成本并释放 ROCm 7.0 占用的 GPU 计算资源。- 风险标记：PR 阶段多版本覆盖下降，CI 监控过滤逻辑变更，大量 workflow 删除，AITER Scout 资源放大

关联脉络

- PR #39411 [NPU] [CI] Fix multimodal_gen filter leakage and add daily scheduled run: 同为平台 CI 的调度节奏与 workflow 精简调整，与本次 AMD CI 合并属于同一类「平台 CI 收敛」演进方向，可对比 NPU 与 AMD 的 CI 治理模式。
- PR #39544 [misc] Trim redundant variants from the 8-gpu-h20 disaggregation test suite: 同样以削减冗余 CI 变体、降低资源占用为目标，与本次退役 ROCm 7.0、合并重复 workflow 的诉求一致，反映仓库近期 CI 资源治理趋势。
- PR #39545 [CI] Wait for a killed test server's GPU memory before the next launch: 改进 AMD 等自建 runner 的 GPU 释放等待逻辑，属于 AMD CI 稳定性配套，与本 PR 的容器启动脚本改动 (amd_ci_start_container.sh) 作用于同一 CI 运行链路。
- PR #39600 [CI] Fix SWA decode radix cache NIXL import: 针对 AMD/disaggregation CI 测试的导入崩溃修复，与本 PR 一样属于 AMD CI 日常维护面，表明 AMD CI 体系正在经历高频调整期。