

PR #35686 完整报告

sgl-project/sglang

[AMD][CI] Name the ROCm Image That Actually Ran in AMD Job Names

合并时间: 2026-08-24 16:07

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35686>

执行摘要

- 一句话: AMD CI 作业名展示真实 ROCm 版本, 联动更新监控与失败报告
- 推荐动作: 值得精读, 尤其是对维护复杂 GitHub Actions 矩阵 CI 的团队。本 PR 包含三个可借鉴的设计决策: 一是展示名与内部 id 解耦 (id 保留 `-rocm720` 后缀稳定 needs 依赖, 展示名携带可读的 flavor 与 runner 坐标); 二是规避 `wait-for-jobs` 的 `job.name` 前缀匹配陷阱并留下清晰的 commit 解释; 三是展示名变更时的历史数据策略——"宁可丢弃、不可误映射", 用 `_filter_legacy_amd_job_rows` 独立于共享分析器做局部过滤, 避免旧失败顶替新名字显示。搭配的单元测试虽然只有两个用例, 但精准覆盖了过滤函数的边界行为。

功能与动机

PR body 开宗明义: "This PR changes the job names shown in GitHub Actions for the AMD ROCm 7.0 and 7.2 test workflows." 根因是前序改动造成展示名与实际镜像脱节: #35602 之后 "the ROCm 7.2 PR gate pulls the 7.2.4 image while every job in it is still named `...-rocm720`"; #35603 之后 "one nightly run covers both images but no nightly job declares a `name:`, so GitHub auto-appends the matrix value and prints `nightly-test-1-gpu-unit-rocm720 (rocm724)`". 于是 `-rocm720` 既可能表示 7.2.0 也可能表示实际跑的 7.2.4, 运维无法从作业名判断真实环境, 定时任务报告与失败归因都会失真。方案是 "Move the flavor out of the name stem and into the paren coordinates", 并且无条件打印 (`pr-test-amd-extra` 之前对默认版本会抑制后缀)。

实现拆解

1. 统一展示名格式: 在五个工作流中为 GPU 作业显式声明 `name:`, 格式为 `<test> (<rocm版本>, <runner>[, <part>])`。其中 `pr-test-amd.yml` 硬编码 `rocm700`; `pr-test-amd-rocm720.yml` 用 `inputs.rocm_version || 'rocm724'` 支持按调用方选择镜像; `nightly-test-amd.yml` 硬编码 `rocm700` 并补上 runner; `nightly-test-amd-rocm720.yml` 用 `matrix.rocm_version` 动态拼名 (一个运行覆盖 7.2.0 与 7.2.4 两个镜像); `pr-test-amd-extra.yml` 删掉 "默认版本不带后缀、非默认版本才加后缀" 的条件逻辑, 统一为 `(rocm724, runner)`。内部 job id 全部保留, `needs` 依赖与 `lint / pr-test-finish` 合并检查不受影响。
2. 修复 `wait-for-jobs` 前缀失配: commit 说明 `wait-for-jobs` 匹配的是 Actions API 的 `job.name` 而非 job id, 规则为 `name === prefix || name.startsWith(prefix + ' ')`。展示名去掉 `-rocm720` 后缀后, 7.2 gate 的四个 wait 前缀全部匹配不到任何作业, stage A 会

空转满 240 分钟、stage B 空转 480 分钟。修复方式是让 `pr-test-amd-rocm720.yml` 的 `wait` 作业声明 `name: wait-for-stage-a-amd`，并把 `stage-name` 与 `jobs[].prefix` 改回裸茎干；裸前缀靠末尾空格与 `-nondeterministic`、`-mi35x` 变体区分。

3. 改写作业监控解析：`amd-ci-job-monitor.yml` 中 PR 7.2 通道用 `sed 's/-rocm720$/'` 把 job id 还原成展示名前缀；nightly 通道把每个 job id 与 `runs-on` 配对后，对 `rocm_flavors="rocm724 rocm720"` 各展开一行，避免 `stem-only` 前缀把所有 flavor 合并成一行而掩盖 " 只在某个镜像上复现 " 的回归；同时排除 `wait-for-stage-*`（跑在 `ubuntu-latest`，自托管快照没有数据），保留 `call-pr-test-amd-extra` 调用方以覆盖其委派的 GPU 作业。
4. 失败报告历史过滤：`scripts/ci_monitor/ci_failures_analysis.py` 新增模块级函数 `_filter_legacy_amd_job_rows`，在 `main()` 中对 pr / nightly 的 `scheduled` 与 `general` 四路 AMD 数据统一调用，丢弃旧茎干以 `-rocm<数字>` 结尾的行、中间态只含 flavor 的 nightly 行、被改名的 `call-pr-test-amd-extra-rocm720` 外层调用方及各类 `wait/finish/gate` 工具作业。设计取向是 " 宁可丢弃、不可误映射 "——旧名的版本号并不可靠，所以不清洁映射而是整体重置 AMD 失败历史；共享分析器与非 AMD 报告完全不动。
5. 测试与收尾：新增 `scripts/ci_monitor/test_ci_failures_analysis.py`，两个用例分别覆盖 " 旧名 + 嵌套工具作业全部丢弃 " 与 " 新名 + 仍有效的嵌套调用方全部保留 (对象身份不变) "。`scripts/ci/utils/query_job_status.py` 注释示例更新为完整展示名并提示 `stem` 前缀会匹配所有 flavor。验证项包括 YAML 解析、`actionlint`、`workflow job-name lint`、Python 编译、`Black`、`git diff --check` 与新增单测。

关键文件：

- `scripts/ci_monitor/ci_failures_analysis.py`（模块 失败分析；类别 `source`；类型 `core-logic`；符号 `_filter_legacy_amd_job_rows`）：新增 `_filter_legacy_amd_job_rows` 模块级过滤函数，并在 `main()` 中对 pr / nightly 的 `scheduled` 与 `general` 四路 AMD 数据统一调用，是改名后失败报告不出现旧名顶替新名的核心保障；共享分析器保持不动。
- `scripts/ci_monitor/test_ci_failures_analysis.py`（模块 单元测试；类别 `test`；类型 `test-coverage`；符号 `TestFilterLegacyAmdJobRows`，`test_drops_legacy_names_and_nested_utilities`，`test_keeps_current_flavors_and_nested_callers_separate`）：新增单测覆盖过滤函数的丢弃与保留两条路径，验证旧名行、中间态名、嵌套调用方、工具作业与新名行的边界行为，是本次唯一的行为级测试保障。
- `.github/workflows/nightly-test-amd-rocm720.yml`（模块 工作流；类别 `infra`；类型 `infrastructure`）：改动量最大的工作流 (+92/-46)：约 23 个 nightly 作业统一改为 `format('... ({0}, ...runner)', matrix.rocm_version)` 动态展示名，`Setup docker` 步骤也改为打印所选 flavor，一个运行覆盖 7.2.0 与 7.2.4 两个镜像。
- `.github/workflows/pr-test-amd-rocm720.yml`（模块 工作流；类别 `infra`；类型 `infrastructure`）：ROCm 7.2 PR 通道的核心改造文件：作业展示名改为携带 `inputs.rocm_version || 'rocm724'` 与 `runner`，同时修复 `wait-for-jobs` 前缀失配（`stage-name` 与 `jobs prefix` 改回裸茎干），否则 stage A / stage B 会空转 240 / 480 分钟。

- .github/workflows/nightly-test-amd.yml (模块 工作流; 类别 infra; 类型 infrastructure) : ROCm 7.0 回退影子 nightly 通道: 约 20 个作业补上 name: <stem> (rocm700, <runner>), 使 7.0 与 7.2 两条 nightly 的名字形状完全一致。
- .github/workflows/amd-ci-job-monitor.yml (模块 工作流; 类别 infra; 类型 infrastructure) : 定时任务监控的解析逻辑同步适配新名: PR 通道用 sed 去掉 -rocm720 后缀还原前缀; nightly 通道把 job id 与 runs-on 配对后按 flavor 各展开一行, 避免多镜像合并成一行掩盖单镜像回归; 并排除跑在 ubuntu-latest 的 wait 作业。
- .github/workflows/pr-test-amd.yml (模块 工作流; 类别 infra; 类型 infrastructure) : ROCm 7.0 PR 回退影子通道: 16 个 GPU 作业展示名硬编码插入 rocm700 坐标, 补齐 stage-c-test-large-8-gpu-amd-mi35x 等此前没有展示名的作业。
- .github/workflows/pr-test-amd-extra.yml (模块 工作流; 类别 infra; 类型 infrastructure) : opt-in 扩展套件: 删除 " 默认版本不带后缀、非默认版本才加后缀 " 的条件逻辑, 统一为 (rocm724, runner), 消除 #35602 静默改变展示的隐患。
- scripts/ci/utils/query_job_status.py (模块 查询脚本; 类别 infra; 类型 infrastructure) : 查询脚本注释示例更新为完整展示名 (含 flavor 与 runner), 并说明 stem 前缀会匹配所有 flavor、需用完整名钉住单镜像。

关键符号: `_filter_legacy_amd_job_rows`, `TestFilterLegacyAmdJobRows.test_drops_legacy_names_and_nested_utilities`, `TestFilterLegacyAmdJobRows.test_keeps_current_flavors_and_nested_callers_separate`

关键源码片段

scripts/ci_monitor/ci_failures_analysis.py

新增 `_filter_legacy_amd_job_rows` 模块级过滤函数, 并在 `main()` 中对 pr / nightly 的 scheduled 与 general 四路 AMD 数据统一调用, 是改名后失败报告不出现旧名顶替新名的核心保障; 共享分析器保持不动。

```
def _filter_legacy_amd_job_rows(job_data: Dict[str, Dict]) -> Dict[str, Dict]:
    """丢弃 AMD 改名前的旧作业名行, 共享分析器保持不变。"""

    filtered = {}
    for full_name, data in job_data.items():
        # 全名可能是嵌套调用链, 例如 call-pr-test-amd-rocm720 / call-pr-test-amd-extra / extra-a-test-1-gpu-small-amd (...)
        # 只有叶子名才是真正的测试作业名; 此处只需过滤被改名的那一个外层调用方,
        # 包括 AITER 在内的其他 *-rocm720 调用方仍然有效。
        name_parts = full_name.split(" / ")
        if "call-pr-test-amd-extra-rocm720" in name_parts[:-1]:
            continue

        leaf_name = name_parts[-1]
        # wait 作业跑在 ubuntu-latest 上, 不承载 GPU 测试, 报告里应跳过
        if leaf_name.startswith(("wait-for-stage-a-amd", "wait-for-stage-b-amd")):
            continue
        # 工具作业 (gate / finish / 汇总) 同样不入报告
```

```

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",
}:
    continue

# 旧版茎干以 -rocm< 数字 > 结尾, 例如 ...-rocm720;
# 但旧名中的版本号不可靠 (-rocm720 的 PR 作业可能实际跑 rocm724) ,
# 无法安全映射到新名, 故整体丢弃而不是清洗。
stem = leaf_name.split("(", 1)[0]
_, separator, version = stem.rpartition("-rocm")
if separator and version.isdigit():
    continue

# 中间态 nightly 名只带 flavor, 例如 nightly-test-1-gpu-unit (rocm724);
# 新名始终同时携带 flavor 与 runner, 形如 (rocm724, linux-mi300-1gpu-sglang)。
if leaf_name.endswith(")") and "(" in leaf_name:
    details = leaf_name.rsplit("(", 1)[1][:-1]
    if details.startswith("rocm") and details[4:].isdigit():
        continue

filtered[full_name] = data

return filtered

```

scripts/ci_monitor/test_ci_failures_analysis.py

新增单测覆盖过滤函数的丢弃与保留两条路径, 验证旧名行、中间态名、嵌套调用方、工具作业与新名行的边界行为, 是本次唯一的行为级测试保障。

"""AMD 作业名切换的过滤单测: 验证丢弃旧名、保留新名与嵌套调用方。"""

```

import os
import sys
import unittest

sys.path.insert(0, os.path.dirname(os.path.abspath(__file__)))
from ci_failures_analysis import _filter_legacy_amd_job_rows # noqa: E402

class TestFilterLegacyAmdJobRows(unittest.TestCase):
    def test_drops_legacy_names_and_nested_utilities(self):
        # 旧茎干带 -rocm< 数字 > 后缀、中间态只含 flavor 的 nightly 名、
        # 已改名的 call-pr-test-amd-extra-rocm720 外层调用方、
        # 以及 wait / finish / gate 工具作业, 全部应被丢弃。
        rows = {
            "stage-b-test-1-gpu-small-amd-rocm720 (linux-mi300-1gpu-sglang, 0)": {},
            "nightly-accuracy-2-gpu-rocm720 (rocm724)": {},
            "nightly-accuracy-2-gpu-rocm724": {},

```

```

        "nightly-test-1-gpu-unit (rocm724)": {},
        "call-pr-test-amd-rocm720 / call-pr-test-amd-extra-rocm720 / extra-a-test-1-gpu-small-amd (linux-mi300-1gpu-sglang)": {},
        "wait-for-stage-a-amd": {},
        "call-pr-test-amd-extra / pr-test-amd-extra-finish": {},
        "call-pr-test-amd-extra / call-gate / pr-gate": {},
    }
    self.assertEqual(_filter_legacy_amd_job_rows(rows), {})

def test_keeps_current_flavors_and_nested_callers_separate(self):
    # 新名同时携带 flavor 与 runner；嵌套调用方只过滤被改名的那个，
    # 其他外层 caller 的 -rocm720 前缀仍被视为当前有效。
    new_success = {"current_streak": 0}
    rows = {
        "stage-b-test-1-gpu-small-amd (rocm724, linux-mi300-1gpu-sglang, 0)": new_success,
        "nightly-accuracy-2-gpu (rocm720, linux-mi300-2gpu-sglang)": {
            "current_streak": 1
        },
        "call-pr-test-amd-rocm720 / stage-c-test-4-gpu-amd (rocm724, linux-mi300-4gpu-sglang, 0)": {
            "current_streak": 0
        },
        "call-pr-test-amd-rocm720 / call-pr-test-amd-extra / extra-a-test-1-gpu-small-amd (rocm724, linux-mi300-1gpu-sglang)": {
            "current_streak": 0
        },
    }

    filtered = _filter_legacy_amd_job_rows(rows)
    self.assertEqual(set(filtered), set(rows))
    self.assertIs(
        filtered["stage-b-test-1-gpu-small-amd (rocm724, linux-mi300-1gpu-sglang, 0)"],
        new_success,
    )

```

```

if __name__ == "__main__":
    unittest.main()

```

[.github/workflows/nightly-test-amd-rocm720.yml](https://github.com/ROCm/ROCm/actions/workflows/nightly-test-amd-rocm720.yml)

改动量最大的工作流 (+92/-46)：约 23 个 nightly 作业统一改为 `format('... ({0}, ..runner)', matrix.rocm_version)` 动态展示名，Setup docker 步骤也改为打印所选 flavor，一个运行覆盖 7.2.0 与 7.2.4 两个镜像。

```

# nightly-test-amd-rocm720.yml: 一个运行通过矩阵覆盖 rocm724 / rocm720 两个镜像。
# 展示名不再依赖 GitHub 自动附加矩阵值（那会产生 -rocm720 (rocm724) 这种误导名），
# 而是显式 format 成 "<stem> (<rocm 版本>, <runner>)"，与 PR 通道保持一致。
nightly-test-1-gpu-unit-rocm720:
    name: ${{ format('nightly-test-1-gpu-unit ({0}, linux-mi300-1gpu-sglang)', matrix.rocm_version)

```

```

}}
strategy:
  fail-fast: false
  matrix:
    # 与 amd-ci-job-monitor.yml 里硬编码的 rocm_flavors="rocm724 rocm720" 保持同步
    rocm_version: [rocm724, rocm720]
  # ...
steps:
  # 启动语句也改为打印实际选中的版本，而不是固定文本 ROCm 7.2
  - name: Setup docker (${{ matrix.rocm_version }})
    run: |
      touch github_summary.md
      bash scripts/ci/amd/amd_ci_start_container.sh --rocm-version ${{ matrix.rocm_version }}

```

评论区精华

该 PR 没有任何 review 评论，HaiShaw 直接 APPROVED，核心设计权衡全部沉淀在 8 个 commit message 与 PR body 中，最值得记录的三处：

"wait-for-jobs matches the Actions API job.name, not the job id, using `name === prefix || name.startsWith(prefix + ' ')`. Dropping the -rocm720 suffix from the display names left all four prefixes in the 7.2 gate matching nothing, so stage A would have spun for its full 240 minutes and stage B for 480 after that."

"An intermediate nightly schema showed only the ROCm flavor. Current names always include both the flavor and runner inside parentheses."

"The monitor's failure report drops only old-schema AMD ROCm 7.2 rows, so an old failure cannot look current under a name that no longer exists."

- 暂无高价值评论线程

风险与影响

- 风险：
 - 外部匹配失效（确定性风险）：PR body 明确提示 "Any external dashboard or script that matches an old full display name must be updated." 仓库内 monitor、失败分析器、查询示例已全部联动，但仓库外的消费方（如团队自建看板）不受本 PR 控制。
 - wait-for-jobs 前缀陷阱（高危隐患，本次已修复）：若后续再次调整展示名，必须同步检查 .github/actions/wait-for-jobs 的 `name.startsWith(prefix + ' ')` 匹配语义；本次差点让 stage A / stage B 分别空转 240 / 480 分钟。
 - 历史报告数据断层：改名前的 AMD 失败行被整体丢弃，且尚未迁移到新名的分支产生的失败也会被过滤（其实际 flavor 无法可靠映射），会造成短暂盲区；旧运行仍可在 GitHub Actions 页面查到。
 - nightly 监控与矩阵需手工同步：amd-ci-job-monitor.yml 硬编码 `rocm_flavors="rocm724 rocm720"` 与各作业的 `runs-on`，注释要求与 nightly 工作流的 `rocm_version` 矩阵保持同步；新增 flavor 而忘记更新 monitor 会漏报。

- 未展开矩阵的展示不确定性：作业在矩阵展开前被跳过时，展示名可能显示表达式而非解析后的名字（PR body 已如实说明）。
- 行为零变化：GPU 测试命令、镜像、runner、触发条件、调度与依赖均未改变，内部 job id 不变，稳定合并检查不受影响。
- 影响：影响范围集中在 AMD CI 可观测性链路：GitHub Actions 页面作业名、amd-ci-job-monitor 定时监控、ci_failures_analysis 失败报告与 query_job_status 查询示例四类消费方全部联动更新，覆盖 PR 测试（含 ROCm 7.0 回退影子通道与 7.2 通道）、nightly 测试（约 43 + 46 个作业定义）以及 AITER 侦察、release-branch-cut、bot-bump 等继承子名的流程。对用户与团队而言，运维能直接从作业名判断真实 ROCm 镜像，降低版本误判导致的失败归因成本；对系统而言 GPU 测试行为零变化，内部 job id 不变保证合并检查稳定。影响程度中等偏上，属于纯 CI 基础设施的展示与报告层标准化，不触及任何运行时路径。
- 风险标记：展示名变更破坏外部脚本匹配，wait-for-jobs 前缀匹配陷阱，历史报告数据断层，监控与矩阵需手工同步，未展开矩阵作业名不确定

关联脉络

- PR #35602 （标题未提供）ROCm 7.2 PR gate 拉取 7.2.4 镜像：PR body 指明本 PR 是其 follow-up；#35602 使 ROCm 7.2 PR gate 实际拉取 7.2.4 镜像而作业名仍为 -rocm720，是本次改名的直接触发点。
- PR #35603 （标题未提供）nightly 单运行覆盖双 ROCm 镜像：PR body 与首个 commit 提到 #35603 后一个 nightly 运行覆盖两个镜像且无显式 name，GitHub 自动附加矩阵值产生误导名，触发本次展示名标准化。
- PR #34483 （标题未提供）AMD multimodal-gen 单测并入 1-GPU lane: commit 记录合并 origin/main 时吸收 #34483：其删除了 multimodal-gen-unit-test-amd 并折叠到 1-GPU lane，本分支在合并时采用该删除，避免对已删除作业改名。
- PR #35383 [AMD][CI] Add the Qwen3.8 MXFP4 MI35x nightly: 同一功能线：修改了 .github/workflows/nightly-test-amd-rocm720.yml 与 test/run_suite.py；本 PR body 验证 "preserved all four current Qwen nightly jobs; the obsolete Qwen3-235 jobs remain removed"，二者在 nightly 作业集合上直接交互。