

# PR #34761 完整报告

sgl-project/sglang

[AMD][CI] Restore gfx942 Grok-1 INT4 and Grok-2 schedules

合并时间: 2026-08-14 06:57

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34761>

## 执行摘要

- 一句话: 恢复 gfx942 Grok-1 INT4 与 Grok-2 夜测调度, 仅移除 FP8
- 推荐动作: 值得 CI/ 基础设施维护者快速浏览: 变更本身是纯 workflow 调度恢复, 逻辑直白; 真正的价值在于 PR body 中“与 #34643 叠加做机器净差异验证”的方法论, 可作为同类 CI 调度调整的验证模板。不需要精读源码, 也不涉及运行时行为。

## 功能与动机

这是对已合并 #34643 的 follow-up, PR body 明确说明: Keep the gfx942/MI30x Grok-1 INT4 and Grok-2 scheduled jobs; only Grok-1 FP8 should remain removed on gfx942。与 #34643 合并后, net 差异应为仅移除 MI30x GPT-OSS job 中内嵌的 Grok-1 FP8 步骤, 而不是把所有 Grok 系列任务都从 gfx942 移除。

## 实现拆解

1. 恢复 job 定义: 在两个 workflow (.github/workflows/nightly-test-amd.yml 与 .github/workflows/nightly-test-amd-rocm720.yml) 中恢复 gfx942 专属 job nightly-8-gpu-grok1-int4[-rocm720] 与 nightly-8-gpu-grok2[-rocm720], runs-on 均为 linux-mi300-8gpu-sglang。每个 job 含 Accuracy Test 与 Performance Test 两步, 分别调用注册套件 nightly-amd-accuracy-8-gpu-grok1-int4 / nightly-amd-accuracy-8-gpu-grok2 及对应 perf 套件, 沿用 RCCL\_MSCCL\_ENABLE=0、timeout-minutes: 60、GITHUB\_STEP\_SUMMARY 回写与 continue-on-error 等既有约定。
2. 恢复调度入口: 在 workflow\_dispatch 的 job\_select 列表和 check-all-jobs 的 needs 列表中加回上述 job 名, 保持与 #34643 之前一致的顺序, 确保手动触发、定时调度与全量聚合均能覆盖这两个任务。
3. 修正注释: 将 GPT-OSS 段落注释统一为 8-GPU GPT-OSS (MI35x mixes Qwen3-Coder-Next), 删除“MI30x mixes Grok1-FP8”这一失效子句, 避免调度清单与注释不一致。
4. 验证配套: 无测试、模型、内核或配置阈值改动; PR 通过 workflow YAML 解析、重复 job 名检查、registered-test 校验与 codespell, 并与 pre-#34643 版本做机器 diff, 确认 job 数量、job 体、job\_select 顺序与 check-all-jobs needs 顺序完全一致。

关键文件:

- `.github/workflows/nightly-test-amd.yml` (模块 AMD 夜测; 类别 infra; 类型 configuration; 符号 `nightly-8-gpu-grok1-int4`, `nightly-8-gpu-grok2`) : 主 AMD nightly workflow (ROCm 7.0) : 恢复 `gfx942` 的 Grok-1 INT4 与 Grok-2 job 定义、`workflow_dispatch` 选项及 `check-all-jobs` 依赖, 并修正 GPT-OSS 注释。
- `.github/workflows/nightly-test-amd-rocm720.yml` (模块 AMD 夜测; 类别 infra; 类型 configuration; 符号 `nightly-8-gpu-grok1-int4-rocm720`, `nightly-8-gpu-grok2-rocm720`) : ROCm 7.2 版 AMD nightly workflow: 与主 workflow 同步恢复 `gfx942` Grok-1 INT4 与 Grok-2 job、`dispatch` 选项及聚合依赖。

关键符号: 未识别

## 评论区精华

本 PR 没有任何 review 评论或讨论线程, 维护者 HaiShaw 直接 approve。核心论证都集中在 PR body 的验证表格中: 与 #34643 叠加后的净差异仅为移除 9 行 Grok-1 FP8 步骤; Grok-1 INT4 与 Grok-2 的 job 块与 pre-#34643 字节级一致; `job_select` 与 `check-all-jobs` 顺序一致且无 stale 条目。另一个重要观察是 Grok-2 的 perf 步骤当前不会启动, 因为其 `accuracy` 步骤失败, 恢复后可能出现持续红测。

- 恢复调度与保留 FP8 移除的取舍 (other): 维护者 HaiShaw approve; 恢复 `gfx942` Grok-1 INT4 与 Grok-2, 保留 Grok-1 FP8 移除。

## 风险与影响

- 风险:
  1. GPU 资源成本: 恢复两个任务后, 每对 ROCm 7.0/7.2 nightly 运行增加约 34.5 GPU-hours; 虽然相对 #34643 净节省 4.6 GPU-hours/ 对, 但绝对值仍不小, 需关注 AMD CI 队列压力。
  2. 已知失败任务: PR body 自述 Grok-2 perf 当前因 `accuracy` 步骤失败而不启动, 恢复调度可能带来持续红色 nightly; `continue-on-error` 只保护 perf 步骤, `accuracy` 失败仍会使 job 变红。
  3. job 与套件名耦合: job 内硬编码了 `nightly-amd-accuracy-8-gpu-grok1-int4` 等套件名, 若未来测试套件重命名或注册路径变化, 会导致 job 空跑或失败, 需在改动测试注册时同步检查这些 workflow。
  4. 无任何 Python/C++ 代码变更, 因此无运行时回归风险。 - 影响: 对 AMD nightly CI: 恢复 `gfx942` (MI30x) 上 Grok-1 INT4 与 Grok-2 的精度与性能持续覆盖, 同时保留 Grok-1 FP8 移除, 使 `test_grok1_fp8_eval_amd.py` 与已无人调度的 `test_grok1_fp8_perf.py` 一样成为 `registered-but-not-run`。对 AITER Scout: `amd-aiter-scout.yml` 以 `job_filter: all` 调用这两个 workflow, 会自动继承恢复的 Grok 覆盖, 无需单独定义。对团队: AMD/CI 维护者需要监控恢复后 nightly 的稳定性与 GPU 资源消耗。 - 风险标记: CI 资源消耗增加, 已知失败任务可能持续红测, job 与套件注册名强耦合

## 关联脉络

- PR #34643 [AMD][CI] Stop scheduling Grok-1 and Grok-2 on MI30x: 本 PR 的直接 follow-up: 撤销其对 gfx942 上 Grok-1 INT4 与 Grok-2 的过度移除, 仅保留 Grok-1 FP8 的移除, 并与该 PR 叠加做了净差异机器验证。