

PR #33644 完整报告

sgl-project/sglang

[CI] Free hosted-runner disk space only when it is low

合并时间: 2026-08-05 12:55

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33644>

执行摘要

- 一句话: CI 磁盘清理改为按空间条件触发
- 推荐动作: 值得快速阅读, 作为 CI 按需执行昂贵清理的典型示例。设计上先量化开销 (80 秒删除约 1M 个文件) 再引入条件化, 思路清晰; 后续可考虑把 df 解析包成可复用的 Action 或统一放到 prepare_runner.sh, 并补充空值防御。

功能与动机

PR body 指出: ubuntu-latest 现在自带 145G 根盘 (清理前约 65G 空闲), CPU 套件峰值约 30G, 但每个 CPU job 仍花费约 80 秒删除约 1M 个预装文件。因此需要将删除操作的门槛设为可用空间低于 40G, 正常路径零成本, 镜像或套件占用变化时兜底逻辑仍能保证 job 安全。

实现拆解

变更入口: 两个 GitHub Actions workflow 文件 `_pr-test-stage-cpu.yml` 与 `rerun-test.yml` 中的 `Free disk space` 步骤。

1. 定位步骤: 两个文件各自的 Free disk space 步骤原先无条件执行 `sudo rm -rf /usr/share/dotnet /usr/local/lib/android /opt/ghc`, 用于给 CPU 测试套件腾出磁盘空间。
2. 加入空间探测: 在删除前执行 `df --output=avail -BG /`, 通过 `tail -1` 取数据行、`tr -dc '0-9'` 提取可用 GB 数值, 存入 `avail_gb`。
3. 阈值判断: 仅当 `avail_gb < 40` 时才执行删除; 否则跳过, 保留 `df -h` 输出当前磁盘状态。
4. 改动影响: 正常路径不再支付约 80 秒的删除开销; 若镜像或套件占用变化导致空间不足, 40G 阈值仍会触发清理, 维持 job 安全。两个文件改动完全一致, 保证主套件与重跑流程行为统一。无测试或配置配套改动; workflow 本身即部署对象。

关键文件:

- `.github/workflows/_pr-test-stage-cpu.yml` (模块 workflow; 类别 infra; 类型 infrastructure): CPU 测试主套件的工作流, 本 PR 的核心改动点之一, 将无条件磁盘清理改为按可用空间条件触发。
- `.github/workflows/rerun-test.yml` (模块 workflow; 类别 infra; 类型 infrastructure): 重跑测试的工作流, 包含相同的磁盘清理步骤, 一并修改以保持行为一致。

关键符号: 未识别

关键源码片段

[.github/workflows/_pr-test-stage-cpu.yml](#)

CPU 测试主套件的工作流，本 PR 的核心改动点之一，将无条件磁盘清理改为按可用空间条件触发。

```
# 步骤：按需释放磁盘空间
# 背景：ubuntu-latest 镜像自带约 65G 可用空间，CPU 套件峰值约 30G，
# 但预装文件约 100 万个，无条件删除需约 80 秒，因此只在空间低时执行。
- name: Free disk space
  run: |
    # 用 df 取根分区可用空间（GB），并剥离非数字字符得到纯数字
    avail_gb=$(df --output=avail -BG / | tail -1 | tr -dc '0-9')
    if [ "${avail_gb}" -lt 40 ]; then
      # 低于 40G 才删除大目录，保证测试有足够空间
      sudo rm -rf /usr/share/dotnet /usr/local/lib/android /opt/ghc
    fi
    # 始终打印磁盘状态，便于 CI 日志排查
    df -h
```

评论区精华

该 PR 无实质 review 讨论。作者在 PR 内通过 [/tag-and-rerun-ci](#) 命令触发 CI 重跑，随后自行合并；唯一交互来自机器人 CI 状态检查（base 未通过、extra 失败），未涉及设计争论。

- 暂无高价值评论线程

风险与影响

- 风险：风险点集中在 shell 探测逻辑：
 - df --output=avail 是 GNU coreutils 扩展，当前仅适用于 Ubuntu runner；若工作流被复用到 macOS 或 Windows runner 会直接失败。
 - tr -dc '0-9' 依赖数字为 ASCII，若 locale 不同或输出异常，avail_gb 可能为空，此时 ["\${avail_gb}" -lt 40] 会报整数表达式错误，在 set -e 下可能导致步骤失败；当前 GitHub hosted 环境实测正常，但缺少空值保护。
 - 40G 阈值大于 CPU 套件峰值（30G），但当镜像预装文件减少或套件增长到 40G 以上时，该阈值需要人工演进；好在低于阈值仍会兜底清理。
 - 本 PR 不触及任何运行时代码，不影响推理、调度等核心路径。
- 影响：影响范围限定在 GitHub Actions CI：
 - 所有 CPU 测试 job（_pr-test-stage-cpu.yml）与重跑 job（rerun-test.yml）在磁盘充足时跳过删除，每个 job 平均节省约 80 秒。
 - 低磁盘场景仍保留原有删除行为，与旧逻辑一致。
 - 对用户无感知，对团队收益是 CI 反馈速度提升；对系统无新增依赖。
 - 风险标记：依赖 df 输出解析，空值缺乏防御，40G 阈值需随演进跟进

关联脉络

- PR #33619 [CI] Speed up dependency install: dual-ABI Rust ext cache and prevalidation pruning: 同样聚焦 CI 提速，通过缓存与裁剪预校验避免重复开销；与本 PR 的磁盘清理条件化同属 CI 性能优化线。
- PR #33597 [CI] Extract download-rust-ext and give every install step a cache fallback: 为 CI 安装步骤引入缓存回退，与本 PR 一样希望降低 job 固定开销，改动文件也都在 .github/workflows 下。