

PR #33904 完整报告

sgl-project/sglang

[CI] Refresh the CPU HF cache base only on main-ref runs

合并时间: 2026-08-07 10:38

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33904>

执行摘要

- 一句话: CI 缓存刷新增条件, 仅 main 刷新基础缓存
- 推荐动作: 该 PR 展示了 CI 缓存分层与条件写回的设计模式, 适合运维 CI 的团队快速浏览。虽为核心代码路径外的变更, 但缓存策略的思考方式值得借鉴。

功能与动机

PR body 指出: Actions cache 读取作用域限制在当前 ref 和默认分支, 因此每个 PR 的 ref 每次运行都写缓存副本, 对其他 ref 毫无帮助, 反而因频繁写入把 main 分支的基础缓存挤出, 导致定时运行冷启动并遭遇 Hub 429 限流。需要将基础缓存刷新收敛到 main 分支, PR 仅在获得基础缓存缺失内容时才写自身缓存。

实现拆解

实现拆解如下:

1. 将原单一“Cache HF hub”步骤拆为“Restore HF hub cache”, 改用 actions/cache/restore@v4 进行恢复, 保持原有 key (hf-cpu-<partition>-<run_id>) 与 restore-keys 滚动策略。
2. 新增“Record restored HF cache size”步骤, 通过 du -sb 统计 HF_HOME 目录字节数并写入 GITHUB_OUTPUT, 记录恢复时的大小。
3. 新增“Decide whether to save the HF cache”步骤, 再次统计运行结束后的缓存大小, 并与恢复前比较; 若当前 ref 为 refs/heads/main 或 after > before, 则输出 save=true, 否则 save=false。
4. 新增“Save HF hub cache”步骤, 仅在 save=true 时调用 actions/cache/save@v4 写入缓存。该方案保证 main 始终刷新基础缓存, PR 只在有新数据下载时写回, 避免无效写入和缓存淘汰。

关键文件:

- .github/workflows/_pr-test-stage-cpu.yml (模块 CI 工作流; 类别 infra; 类型 infrastructure): 调整 CPU 测试阶段 HF 缓存刷新策略, 仅 main 刷基础缓存, PR 按需保存

关键符号: 未识别

评论区精华

该 PR 无 review 评论，作者直接合并，讨论内容主要体现在 PR body 与提交信息中，属于单作者独立完成的基础设施调整。

- 暂无高价值评论线程

风险与影响

- 风险：风险点包括：
 - 缓存大小比较依赖 `du` 命令，不同文件系统或重叠挂载可能影响结果，但整体可靠。
 - 若 PR 首次运行时无基础缓存 (`before=0`)，`after` 通常大于 0 会触发保存，符合预期；若测试过程中缓存被清理导致 `after < before`，则不会保存，但后续运行仍可从 `restore-keys` 恢复旧缓存。
 - `main` 分支判断基于 `GITHUB_REF`，若未来该 workflow 被其他非 `main` 分支或 `tag` 触发，可能不会刷新基础缓存，但当前用途为 PR 测试，风险低。
 - 拆分 `restore` 与 `save` 后，需要确保 `HF_HOME` 环境变量在所有步骤中一致。
 - 影响：影响 CI 基础设施：减少了 PR 运行时对缓存的无效写入，保护 `main` 基础缓存不被挤出，提升主分支调度运行的缓存命中率，降低 HuggingFace Hub 429 错误概率，同时不破坏 PR 重跑的热缓存效果（仅在新增下载时保存）。对开发者而言，PR 重跑仍然快速。
 - 风险标记：缓存策略变更，依赖 Actions 缓存行为，大小判断逻辑

关联脉络

- 暂无明显关联 PR