PR #34380 完整报告

sgl-project/sglang

[CI] Add a scheduled workflow to close stale PRs

合并时间：2026-08-11 15:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34380

执行摘要

- 一句话：新增定时工作流自动关闭 stale PR，带 dry-run 与回滚
- 推荐动作：值得精读。该 PR 是"高影响自动化操作"的安全设计范本：dry-run 默认开启、单轮上限、一键回滚、多级豁免、用 vars 做 kill switch。重点关注三个决策：用全量分页替代 search API 避免截断、approving review 豁免的成本收益权衡、以及 updated_at 偏差方向的取舍。团队若要运营大型 GitHub 仓库，可直接复用这套模板。

功能与动机

PR body 给出明确数据：仓库 open PR 积压超过 4000，其中 90 天无活动的 stale 约 1072 个、WIP / 草稿且超 21 天约 331 个，手动 triage 已难以跟上。设计者刻意选择慢速清理，理由是"a bad rule shows up in the reopen rate long before it can close a thousand PRs"；同时为 approved 但 stale 的 PR 保留豁免，因为 43 个里至少 8 个（#22441、#21194、#21153、#20427、#19777、#19857、#19530、#22178）已被人工判定值得保留，关闭它们等于覆盖维护者的审核结论。

实现拆解

1. 入口与触发：新增 .github/workflows/close-stale-prs.yml，通过 schedule cron 0 1 * * * 每日 01:00 UTC 触发；workflow_dispatch 暴露 dry_run、rules、max_actions 三个输入便于手动演练。配置 concurrency: group: close-stale-prs 防止 cron 与手动运行重叠，permissions 限定为 pull-requests: write + contents: read。
2. 候选收集与预筛：脚本用 github.paginate(github.rest.pulls.list) 拉取全部 open PR（按 updated 升序），理由在 body 中明确：search API 上限 1000 会静默截断 4000+ 的积压量。随后为每个作者统计 open PR 数量与最老空闲天数，预先算出 overQuota 作者集合。
3. 规则判定：classify() 按 quota -> wip -> stale 顺序匹配。wip 用 WIP_MARKER 正则匹配标题里的 WIP / DNM / draft 标记或草稿状态 + 空闲超过 21 天；stale 为 90 天无更新；quota 为非协作者、open PR 数 > 5 且所有 PR 都空闲超过 7 天。
4. 豁免与执行：依次跳过带 high priority / keep-open / good first issue 标签的 PR、write 以上权限作者、以及存在 APPROVED review 的 PR（避免覆盖维护者审核结论）。非 dry-run 时先评论（body() 模板含目录迁移提示）再调用 pulls.update 关闭，单轮最多 MAX（默认 32）个。

5. 可观测性与回滚：运行结束向 job summary 写入扫描总数、关闭数、各规则计数和四列表格（reason / author / title / link），并输出一行 gh pr reopen 批量撤销命令；dry-run 文案为“would close”。schedule 事件强制 DRY=false，手动运行默认 dry-run。
6. 配置与配套：RULES 与 MAX_ACTIONS 读取顺序为 workflow 输入 -> vars.STALE_RULES / vars.STALE_MAX_ACTIONS -> 默认值；MAX=0 作为 kill switch。未新增测试文件，安全设计完全依赖 dry-run + 上限 + 撤销命令。

关键文件：

- .github/workflows/close-stale-prs.yml（模块 CI 工作流；类别 infra；类型 infrastructure；符号 classify, body, hasWrite）：唯一变更文件，定义了整个 stale PR 清理机制：触发方式、分类规则、豁免条件、限速与回滚能力。

关键符号：classify, body, hasWrite

评论区精华

该 PR 没有 review 评论，设计权衡主要沉淀在 PR body 的 Notes for reviewers 中：

> search caps at 1000 results and would silently truncate a 4000+ PR backlog.

这决定了用分页 pulls.list 而非 search API。

> updated_at is refreshed by some non-substantive events ... That direction is safe here (skipped, never wrongly closed).

设计者明确接受这一偏差方向，并在文件注释中记录了未来可能的修复。

> schedule runs force dry_run=false. Before enabling the cron, run it manually with dry_run=true and check the summary.

启用 cron 前必须先手动 dry-run 验证。

> The exemption costs 43 out of 1115.

approved 豁免以约 3.9% 的遗漏换取不覆盖维护者审核结果，被认为值得。

- 用 pulls.list 分页替代 search API (design): 采用 github.paginate 遍历全量 open PR，避免 search API 1000 条上限导致的静默截断。
- updated_at 被非实质性事件刷新 (correctness): 该偏差方向安全：只会跳过看似活跃的 PR，不会误关；已在文件注释中记录备选修复思路。
- schedule 强制 dry_run=false 的启用节奏 (design): 先在手动 dry-run 下检查 summary，再启用 cron；手动运行默认 dry-run。
- approved review 豁免的成本与理由 (design): 保留 APPROVED 豁免，成本可接受。

风险与影响

- 风险：
  1. 自动关闭误伤风险：即使有标签、write 权限、approving review 三重豁免，stale 规则仍可能关闭仍在低频维护的 PR。updated_at 被事件刷新只保证“跳过”方向安全，无法识

别“有真实工作但无 GitHub 事件”的 PR。缓解措施包括每次 32 个上限、一键 reopen 命令、dry-run 预检。

2. API 限流风险（推断）：全量分页拉取 4000+ PR，且每个候选 PR 都要调用 listReviews，单次运行 REST 调用量可能达到数千次，接近仓库级 GITHUB_TOKEN 的限流阈值（通常每小时 1000 次）。PR 中未提及限流处理，首次实跑需观察。
3. cron 启用风险：schedule 事件强制 dry_run=false，若未先手动验证规则或 vars 配置错误，可能直接开始关闭；目前默认上限 32 已比较保守。
4. 并发风险：concurrency group 已避免重叠，但 cancel-in-progress: false 意味着手动运行与 cron 并发时会排队而非取消。

- 影响：
  1. 仓库维护：自动消化约 1400 个积压 PR，预计每天净关闭约 19 个，约 2.5 个月清完；同时保持每日上限低于日新增量，避免规则错误造成批量误伤。
  2. 贡献者：被关闭 PR 会收到评论说明原因，并附带最近目录迁移提示；gh pr reopen 一条命令可撤销，quota 规则当前匹配数为 0，不影响活跃贡献者。
  3. 团队：减少人工 triage 负担，approving review 豁免保留维护者结论；job summary 与回滚命令让每次操作可审计。
  4. 范围：仅涉及 GitHub CI 与仓库治理，不触碰 SRT 运行时逻辑。 - 风险标记：自动关闭 PR 可能误伤低频维护，大批量 API 调用可能触及限流，cron 首次启用前需手动 dry-run，updated_at 判定活跃度存在失真

关联脉络

- PR #33086 [Fix] Make wait_port_available actually wait timeout_s seconds: 同属 CI 基础设施与仓库治理方向，与本 PR 无直接依赖。
- PR #34324 [AMD][CI] Run MI300 8-GPU stage-C shards two at a time: 同为 CI 工作流类变更，可视为仓库 CI 治理持续演进的一部分。