

PR #26338 完整报告

sgl-project/sglang

Signal CUDA coredumps to tracker issue

合并时间: 2026-05-26 11:21

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26338>

执行摘要

本 PR 在 [upload-cuda-coredumps](#) GitHub Action 中集成了自动信号上报功能。当 CI 任务因 CUDA coredump 失败时, action 会检查目录并向上报 issue 发布一行评论, 包含 PR 引用和运行 URL, 使团队可通过单个 API 调用查询 30 天内的 coredump 历史, 避免从大量日志中手动 grep。

功能与动机

如 PR body 所述: “post a one-line comment (PR ref + run URL) to sgl-project/sglang-ci-stats#2 so we can query 30-day coredump history with a single API call instead of grepping job logs.” 关联 Issue [sgl-project/sglang-ci-stats#2](#) 也明确说明要自动收集 coredump 事件。

实现拆解

1. [action.yml](#) 改造 ([.github/actions/upload-cuda-coredumps/action.yml](#)) :
 - 新增 check 步骤, 基于环境变量 SGLANG_CUDA_COREDUMP_DIR 目录是否存在且非空输出 has_dumps 布尔值。
 - 将上传步骤的 if-no-files-found 从 ignore 移除 (因前置检查已确保触发条件)。
 - 新增 Signal coredump to tracker issue 步骤, 通过 GitHub REST API 解析当前 job_id, 构建精确的运行 URL 并使用 bot token 向 tracker issue POST 评论。
2. 工作流参数传递 ([.github/workflows/_pr-test-stage.yml](#)) :
 - 在调用 [upload-cuda-coredumps](#) 的 if: failure() 分支下, 新增两个参数 tracker-issue: "26340" 和 bot-token: `${{ secrets.GH_PAT_FOR_PULL_REQUEST }}`。
3. 信号步骤关键逻辑:
 - 利用 GitHub Jobs API 按 runner_name 和 status: in_progress 筛选当前 job, 避免矩阵展开后的命名不确定性。
 - 构建 job 级 URL 作为首选, 若 job_id 查询失败则回退到 run attempt URL。
 - 评论格式为 @hnyls2002 [Coredump Tracker] PR #<num> - <job_url>, 便于在 issue 中快速识别。

以下为 [action.yml](#) 中新增的信号步骤完整定义 (YAML), 包含其前置检查和 run 逻辑:

```
- name: Check for coredumps
```

```

id: check
shell: bash
run: |
  dir="${SGLANG_CUDA_COREDUMP_DIR:-/tmp/sglang_cuda_coredumps}"
  if [ -d "$dir" ] && [ -n "$(ls -A "$dir" 2>/dev/null)" ]; then
    echo "has_dumps=true" >> "$GITHUB_OUTPUT"
  else
    echo "has_dumps=false" >> "$GITHUB_OUTPUT"
  fi

```

- name: Signal coredump to tracker issue

```

if: steps.check.outputs.has_dumps == 'true' && inputs.tracker-issue != '' && inputs.bot-token != ''

```

```

shell: bash

```

```

env:

```

```

  BOT_TOKEN: ${inputs.bot-token}
  PR_NUM: ${github.event.pull_request.number}
  EVENT_NAME: ${github.event_name}
  TRACKER_ISSUE: ${inputs.tracker-issue}

```

```

run: |

```

```

  if [ -n "$PR_NUM" ]; then
    ref_label="PR #${PR_NUM}"
  else
    ref_label="$EVENT_NAME"
  fi

```

```

  job_id=$(curl -sS \
    -H "Authorization: Bearer ${BOT_TOKEN}" \
    -H "Accept: application/vnd.github+json" \
    "${GITHUB_API_URL}/repos/${GITHUB_REPOSITORY}/actions/runs/${GITHUB_RUN_ID}/attempts/${GITHUB_RUN_ATTEMPT}/jobs?per_page=100" \
    | python3 -c 'import json,sys,os; print(next((j["id"] for j in json.load(sys.stdin)["jobs"] if j.get("runner_name")==os.environ["RUNNER_NAME"] and j.get("status")==in_progress), ""))')

```

```

  if [ -n "$job_id" ]; then
    run_url="${GITHUB_SERVER_URL}/${GITHUB_REPOSITORY}/actions/runs/${GITHUB_RUN_ID}/job/${job_id}"
  else
    run_url="${GITHUB_SERVER_URL}/${GITHUB_REPOSITORY}/actions/runs/${GITHUB_RUN_ID}/attempts/${GITHUB_RUN_ATTEMPT}"
  fi

```

```

  body_json=$(printf '{"body": "@hnyls2002 [Coredump Tracker] %s - %s"}' "${ref_label}" "${run_url}")

```

```

  curl -sS -X POST \
    -H "Authorization: Bearer ${BOT_TOKEN}" \
    -H "Accept: application/vnd.github+json" \
    "${GITHUB_API_URL}/repos/${GITHUB_REPOSITORY}/issues/${TRACKER_ISSUE}/comments" \

```

-d "\${body_json}"

评论区精华

本 PR 未发生公开 review 讨论。但 8 次提交历史显示作者进行了多轮实际测试（包括强制注入 coredump 环境、从 gh CLI 改为 curl、调整 issue 编号等），最终在合并前清理了所有调试提交，保证了最终代码的干净。

风险与影响

风险分析：

- 信号步骤依赖 GitHub REST API 的可用性和 bot-token 的权限；若 API 暂时不可用，脚本已内置 fallback 机制（降级到 run attempt URL 而非崩溃）。
- Bot token 通过 secrets.GH_PAT_FOR_PULL_REQUEST 注入，风险可控，但需确保该 PAT 具有对目标仓库 issue 的写入权限。
- 该步骤仅在 coredump 存在且配置 token 后执行，不会影响正常 CI 流程。

影响分析：

- 对开发者：coredump 事件自动记录到单一 issue，便于查询和追踪历史趋势。
- 对 CI 系统：增加一次 curl 调用，开销极小。
- 对团队：未来可基于 tracker issue 的评论构建自动化告警或 dashboard，提升调试效率。

关联脉络

本 PR 直接关联跨仓库 issue [sgl-project/sglang-ci-stats#2](#) (CUDA Coredump Tracker)，该 issue 作为 coredump 事件的消息汇集点。在 CI 基础设施层面，此 PR 与近期其他 CI 改进（如 #26281 启用 EPD CI、#26298 添加 PD 子进程快速终止机制）共同提升了 SGLang 的 CI 可观测性和健壮性。