

# PR #25232 完整报告

sgl-project/sglang

ci: emit machine-readable TIMINGS block at end of run\_unittest\_files

合并时间: 2026-05-14 12:56

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25232>

## 执行摘要

- 一句话: CI 测试框架输出机器可读 TIMINGS 块
- 推荐动作: 作为 CI 基础设施改进, 值得合并。后续可基于 TIMINGS 块构建更丰富的可视化或监控面板。新引入的 `_repo_relative_path` 函数可被其他需要稳定文件路径的模块复用。

## 功能与动机

CI 测试结果目前通过 `grep` 调试日志行提取耗时信息, 缺乏稳定接口。本次变更提供机器可读的 TIMINGS 块, 便于下游 scrapers/dashboards 消费。

## 实现拆解

1. 新增辅助函数: `_repo_relative_path(p: str) -> str` 将任意路径转换为仓库相对路径 (如 `test/registered/foo.py`), 确保 CI runner 不同布局下文件 key 稳定。
2. 引入 `file_elapsed` 字典: 在 `run_unittest_files` 函数中增加 `file_elapsed: Dict[str, float]` 字典, 记录每个文件的最新一次执行耗时 (重试时覆盖旧值)。
3. 在关键位置记录耗时: 在 `run_one_file` 正常返回时记录 `file_elapsed[filename] = elapsed`; 在超时异常处理分支也记录 `file_elapsed[filename] = float(timeout_per_file)`, 确保超时文件出现在 TIMINGS 块中。
4. 输出 TIMINGS 块: 在函数末尾, 遍历 `file_elapsed`, 对每个文件输出一行 JSON: `{"file": repo_relative_path, "passed": fname in passed_set, "elapsed": round(elapsed)}`。用 `'===== TIMINGS BEGIN ====='` 和 `'===== TIMINGS END ====='` 固定包裹。
5. 保持人类可读输出: 原有的 PASSED/FAILED/RETRIED 段落不变, 仅新增机器可读块。

关键文件:

- `python/sglang/test/ci/ci_utils.py` (模块 测试工具; 类别 `test`; 类型 `test-coverage`; 符号 `_repo_relative_path`): 唯一变更文件, 包含 TIMINGS 块输出逻辑和 `_repo_relative_path` 辅助函数

关键符号: `_repo_relative_path`, `run_unittest_files`

## 关键源码片段

[python/sglang/test/ci/ci\\_utils.py](python/sglang/test/ci/ci_utils.py)

唯一变更文件，包含 TIMINGS 块输出逻辑和 \_repo\_relative\_path 辅助函数

```
# python/sglang/test/ci/ci_utils.py
```

```
def _repo_relative_path(p: str) -> str:
```

```
    """Return path stripped to repo-relative form (e.g. 'test/srt/foo.py').
```

```
    Used in the machine-readable TIMINGS block so downstream scrapers
    get a stable key regardless of CI runner checkout layout.
```

```
    """
```

```
    if not os.path.isabs(p):
```

```
        p = os.path.join(os.getcwd(), p)
```

```
    marker = "/sglang/"
```

```
    idx = p.rfind(marker)
```

```
    return p[idx + len(marker):] if idx >= 0 else p
```

```
def run_unittest_files(
```

```
    files: Union[List[TestFile], List[CIRegistry]],
```

```
    timeout_per_file: float,
```

```
    continue_on_error: bool = False,
```

```
    enable_retry: bool = False,
```

```
    max_attempts: int = 2,
```

```
    retry_wait_seconds: int = 60,
```

```
):
```

```
    # ... 原有逻辑不变 ...
```

```
    # Per-file elapsed seconds, latest attempt wins.
```

```
    file_elapsed: Dict[str, float] = {}
```

```
    for i, file in enumerate(files):
```

```
        # ... 文件循环 ...
```

```
        def run_one_file(filename, capture_output=False):
```

```
            # ... 原有逻辑 ...
```

```
            elapsed = time.perf_counter() - file_tic
```

```
            file_elapsed[filename] = elapsed # 记录耗时，重试覆盖
```

```
            # ... 其余代码 ...
```

```
        # ... 重试、超时处理等 ...
```

```
        # 超时分支中:
```

```
        except TimeoutError:
```

```
            kill_process_tree(process.pid)
```

```
            time.sleep(5)
```

```
            file_elapsed[filename] = float(timeout_per_file) # 超时也记录
```

```
    # 所有文件执行完毕后，输出 TIMINGS 块
```

```
    passed_set = set(passed_tests)
```

```
    logger.info("===== TIMINGS BEGIN =====")
```

```
    for fname, elapsed in file_elapsed.items():
```

```
        logger.info(
```

```
            json.dumps({
```

```
        "file": _repo_relative_path(fname),
        "passed": fname in passed_set,
        "elapsed": round(elapsed),
    })
)
logger.info("===== TIMINGS END =====")
```

## 评论区精华

PR 讨论极少，仅有一条 Gemini 机器人的每日配额警告和一条 `/tag-and-rerun-ci` 命令。无实质 review 意见。

- 暂无高价值评论线程

## 风险与影响

- 风险：本次变更仅涉及 CI 测试框架的工具函数，不改变测试执行逻辑，风险极低。可能的风险是 TIMINGS 块格式变更或 JSON 序列化异常导致下游消费端解析失败，但由于输出到日志而非关键 pipeline，影响有限。
- 影响：影响范围仅限于 CI 测试流程中的 `run_unittest_files` 函数。对开发者本地运行无影响。下游仪表盘可按契约解析 TIMINGS 块，替代不稳定的 `grep` 方式。
- 风险标记：下游解析契约变更风险

## 关联脉络

- PR #24253 ci: combine H200 8-GPU warmup steps and surface server log on every path: 同属 CI 基础设施改进，涉及测试流程日志输出优化
- PR #25236 ci: H200 conditional split + dsv4 est\_time recalibration (h200 partition 6→2): 同为 CI 调整，后续可利用 TIMINGS 块数据辅助分区时间校准