

# 中期报告

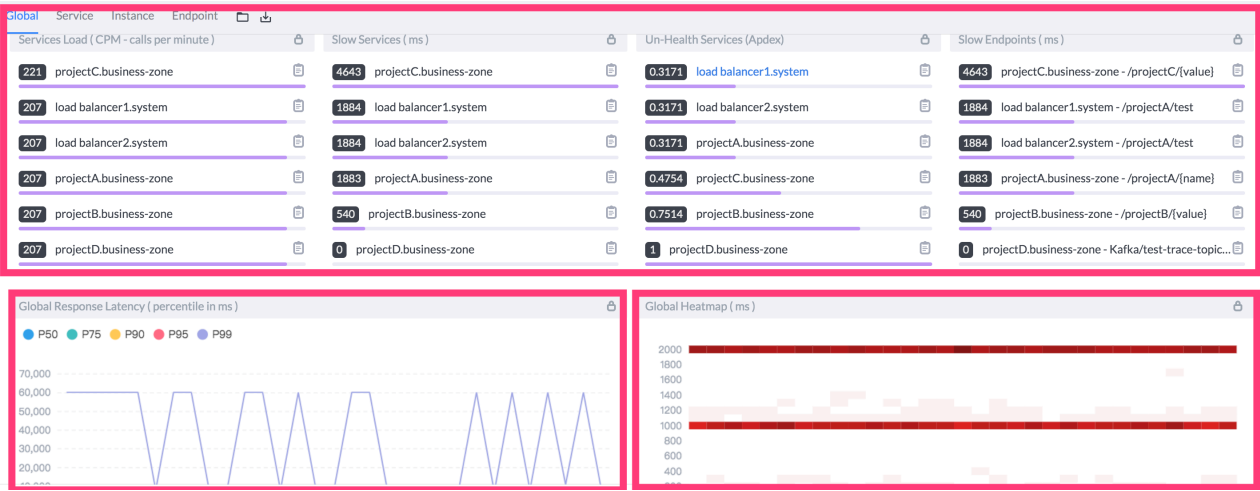
## 一、项目信息

### 1.1 项目名称

命令行 Dashboard （项目ID：2005027）

### 1.2 方案描述

根据 [web ui](#) 的界面，将要展示的内容划分为三块分步实现：服务和端点相关信息；全局响应延迟信息折线图；全局热力图。最后实现这三部分信息的自动刷新功能。



### 1.3 时间规划

- 熟悉阶段：本阶段的主要任务是熟悉项目代码以及涉及的相关知识。

时间段	事项
7月1日 -- 7月7日	通读项目源代码以及文档；复习 <code>Golang</code> 和 <code>GraphQL</code> 相关知识
7月8日 -- 7月14日	搭建 <code>SkyWalking</code> 环境，用于之后调试与测试

- 实施阶段：本阶段的主要任务是编写代码，实现产出要求。

时间段	事项
7 月 15 日 -- 7 月 21 日	实现第一部分全局信息的展示
7 月 22 日 -- 7 月 28 日	实现第二部分全局信息的展示
7 月 29 日 -- 8 月 4 日	实现第三部分全局信息的展示
8 月 5 日 -- 8 月 11 日	完成中期报告并提交
8 月 12 日 -- 8 月 18 日	实现按照给定参数自动刷新

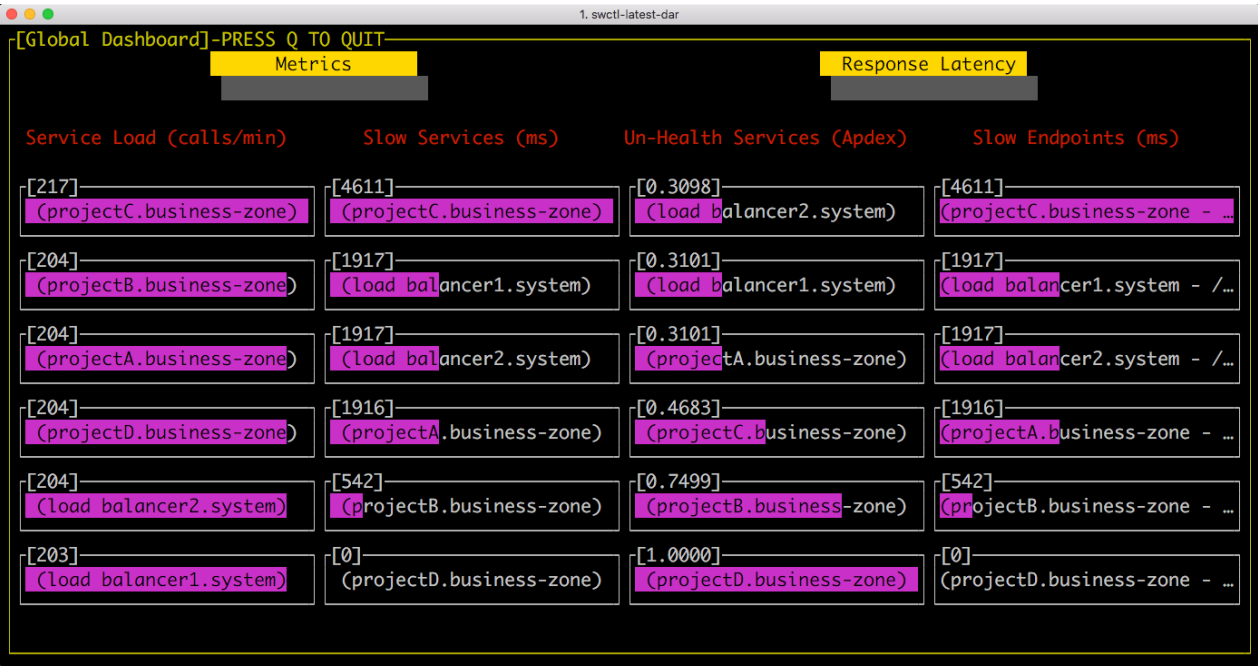
- 收尾阶段：本阶段的主要任务是测试、调试代码，完善文档。

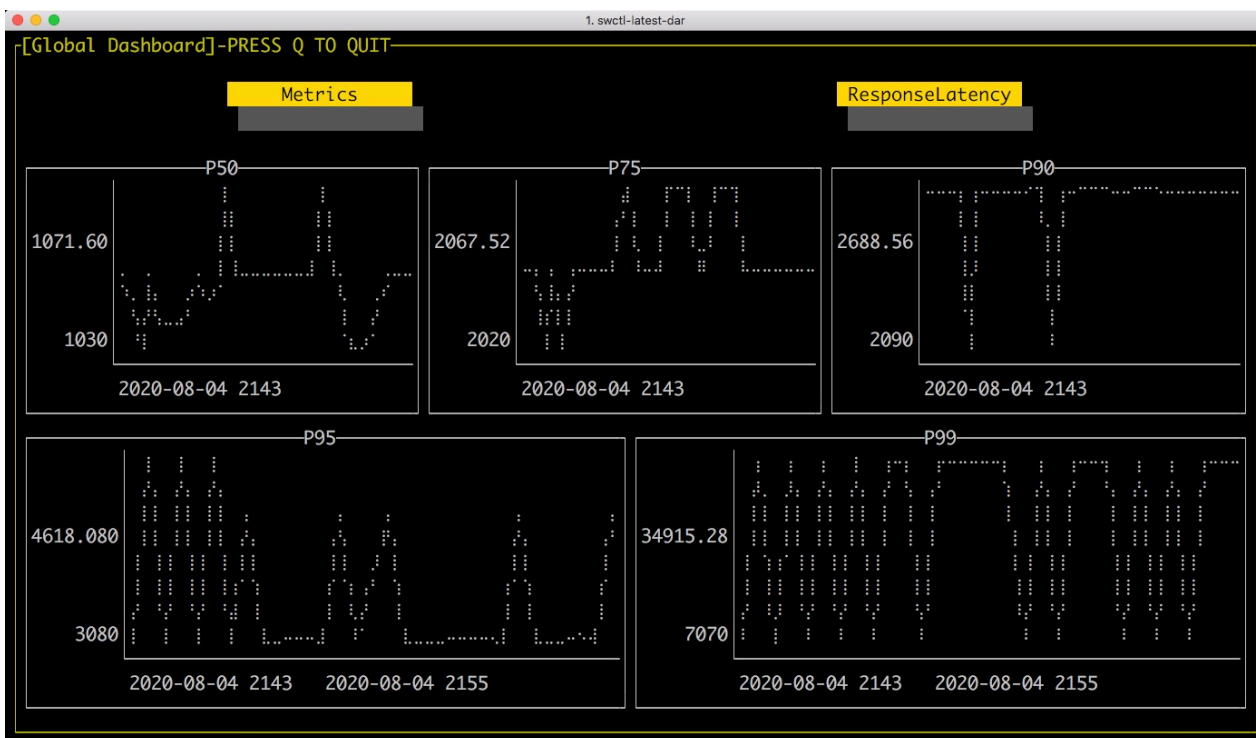
时间段	事项
8 月 19 日 -- 8 月 25 日	测试、调试代码
8 月 26 日 -- 9 月 1 日	完善文档，提交PR
9 月 2 日 -- 9 月 8 日	完成结项报告并提交

## 二、项目进度

### 2.1 已完成工作

根据原定方案和时间规划，目前已经完成了第一部分和第二部分的信息展示，如下图所示，点击黄色按钮可以在两部分信息之间进行切换。此外用户可以传入自己的模板来自定义要展示的信息。

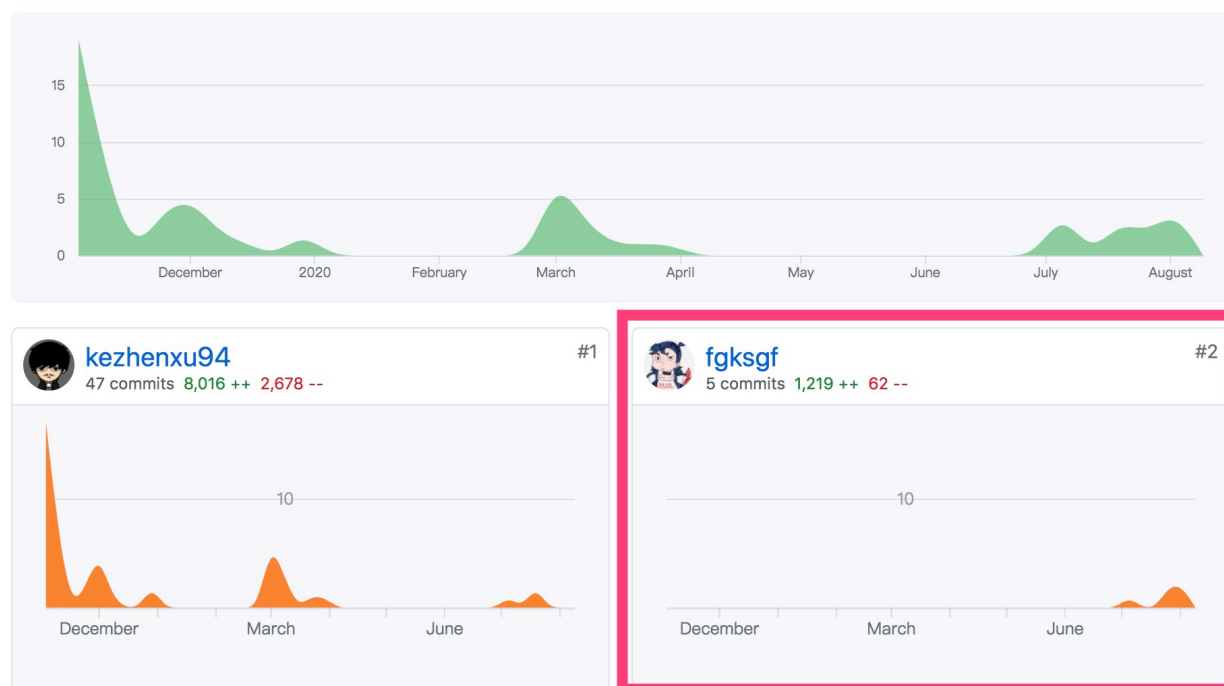




目前已提交五个 `pull requests`，代码已经过导师 `code review` 合并到了 `GitHub` 仓库的 `master` 分支，贡献了一千多行改动，贡献量排第二：

- [Fix a bug and make several improvements](#) ✓ bug feature  
#51 by fgksgf was merged 3 days ago • Approved 0.4.0
- [Integrate global metrics and global response latency into `dashboard global` command](#) ✓ feature  
#50 by fgksgf was merged 7 days ago • Approved 0.4.0
- [Fix `dashboard global` command cannot return json result](#) ✓ bug  
#49 by fgksgf was merged 10 days ago • Approved 0.4.0
- [chore: add changelog for 0.3.0](#) ✓ chore  
#48 by kezhenxu94 was merged 17 days ago • Approved 0.3.0
- [Chore: add missing license](#) ✓ chore  
#47 by kezhenxu94 was merged 19 days ago • Approved 0.3.0
- [Add a subcommand of `dashboard`: `global-metrics`](#) ✓ feature  
#46 by fgksgf was merged 13 days ago • Approved
- [Miss compiling commands](#) ✓ documentation  
#41 by wu-sheng was merged on Jul 8 • Approved 0.3.0
- [Fix metrics graphql path](#) ✓ bug  
#40 by fgksgf was merged on Jul 6 • Approved 0.3.0

Contributions to master, excluding merge commits



## 2.2 遇到的问题及解决方案

主要遇到了两个问题：

- `hard code` 问题。第一次提交的功能性 `pull request`，导师在 `review` 时指出我的代码中有一些 `hard code`，很多请求的参数都写死在了代码里，非常不灵活。我之前并没有考虑过这种问题，幸好导师给出了修改的建议：将这些参数放在配置文件中，通过读取配置文件中的配置来请求数据和展示数据，这样一来很多地方都可以改成用循环语句实现，大大减少了重复代码和硬编码。
- 第三方库不兼容问题。原项目中使用 `termui` 库实现了热力图组件，使用 `termdash` 库实现了折线图组件，但实现命令行的 `dashboard` 要将这三部分集成在一个命令同一个界面里展示，这三部分必须使用相同的库。因此经过对比权衡、和导师讨论后，最终选择了目前仍很活跃、功能更强大的 `termdash` 库。所以需要我用 `termdash` 重新实现一个热力图新组件，于是只能去看 `termdash` 库的源码，找到其已有组件中与热力图最相似的折线图组件代码，在其基础上修改来实现一个新的组件。

除此之外还学到了很多東西，比如 `make` 命令、`go module`、`git submodule`、`git rebase` 等知识，获益匪浅。

## 2.3 后续工作安排

由于第三部分的实现无法复用已有的组件而需要从头实现一个新，因此未能按照初始规划在中期报告前完成，所以将该任务调整在中期报告后完成。后续工作安排调整如下：

时间段	事项
8 月 12 日 -- 8 月 18 日	实现第三部分全局信息的展示
8 月 19 日 -- 8 月 25 日	实现按照给定参数自动刷新
8 月 26 日 -- 9 月 1 日	完善代码和文档
9 月 2 日 -- 9 月 8 日	完成结项报告并提交