结项报告

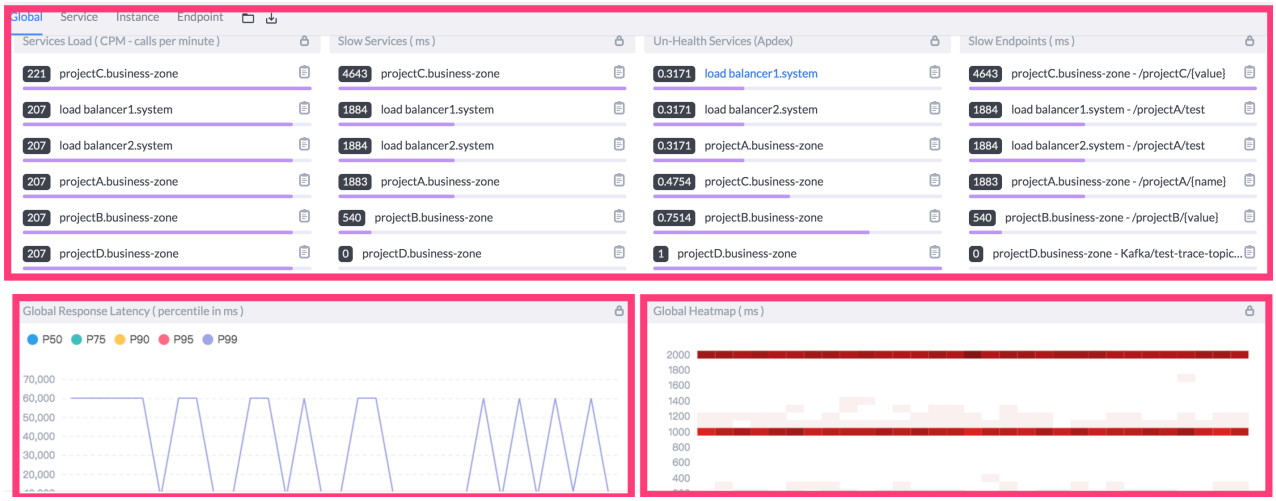
一、项目信息

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`Golang`和`GraphQL`相关知识
7 月 8 日 -- 7 月 14 日	搭建`SkyWalking`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

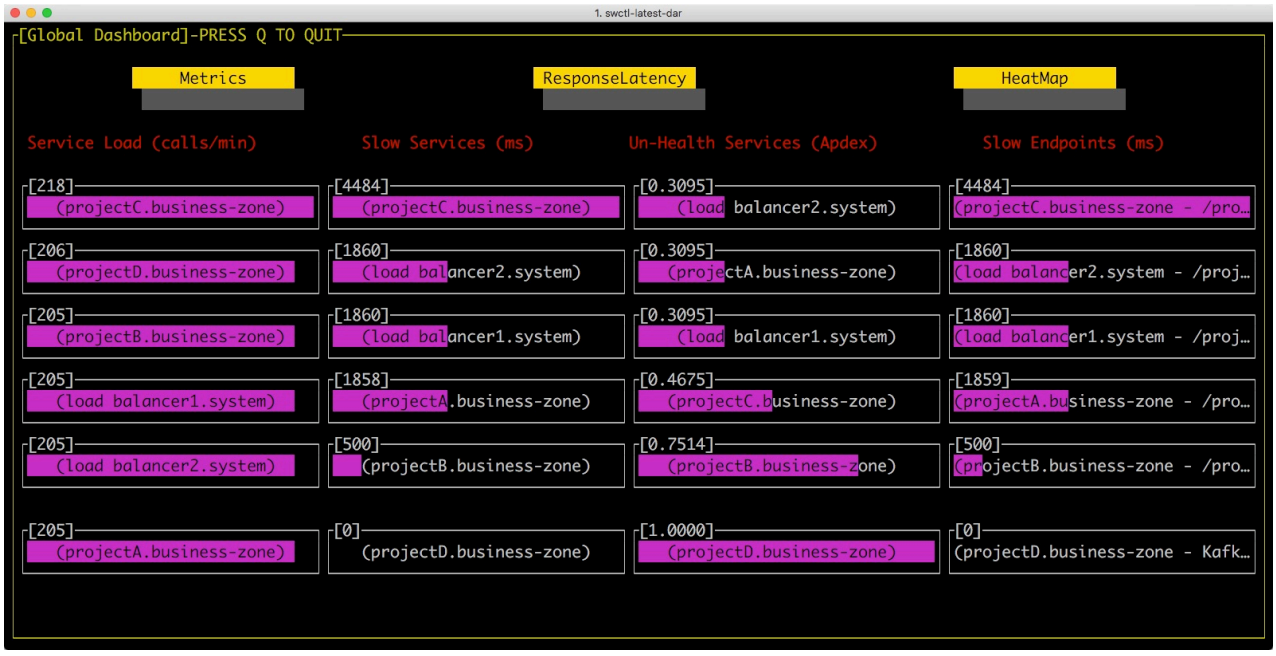
- 收尾阶段：本阶段的主要任务是测试、调试代码，完善文档。

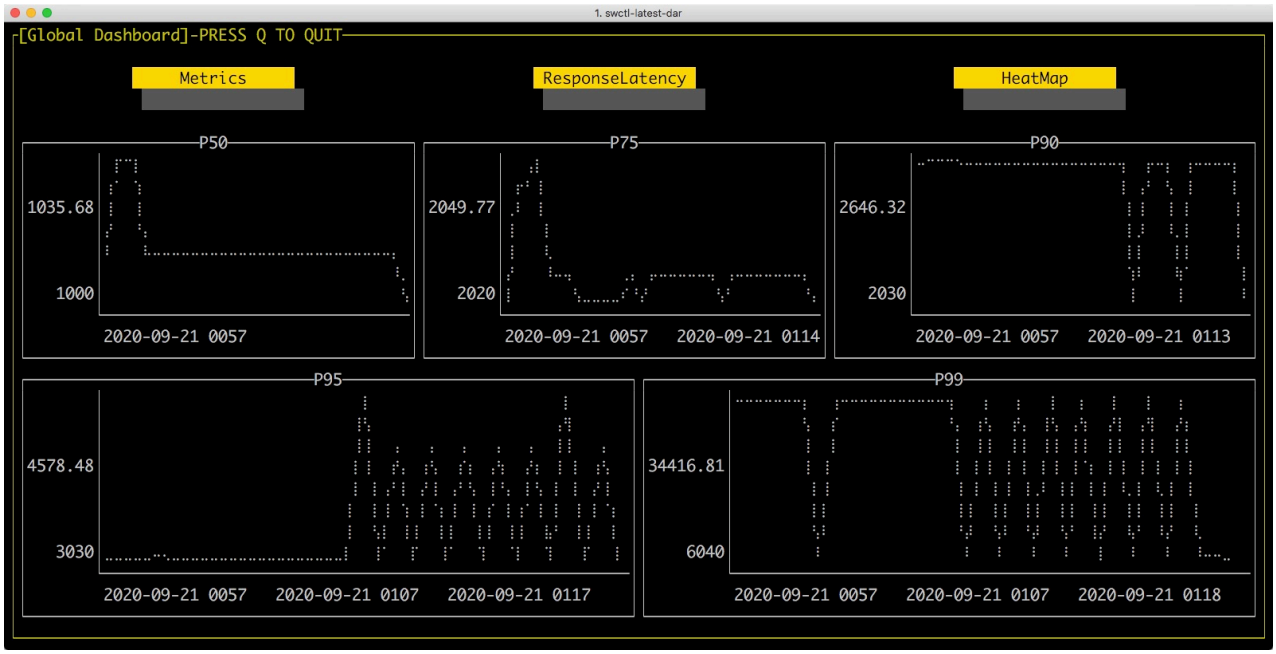
时间段	事项
8 月 12 日 -- 8 月 18 日	完善代码和文档
8 月 19 日 -- 8 月 25 日	实现第三部分全局信息的展示
8 月 26 日 -- 9 月 1 日	实现按照给定参数自动刷新
9 月 2 日 -- 9 月 8 日	完成结项报告并提交

二、项目总结

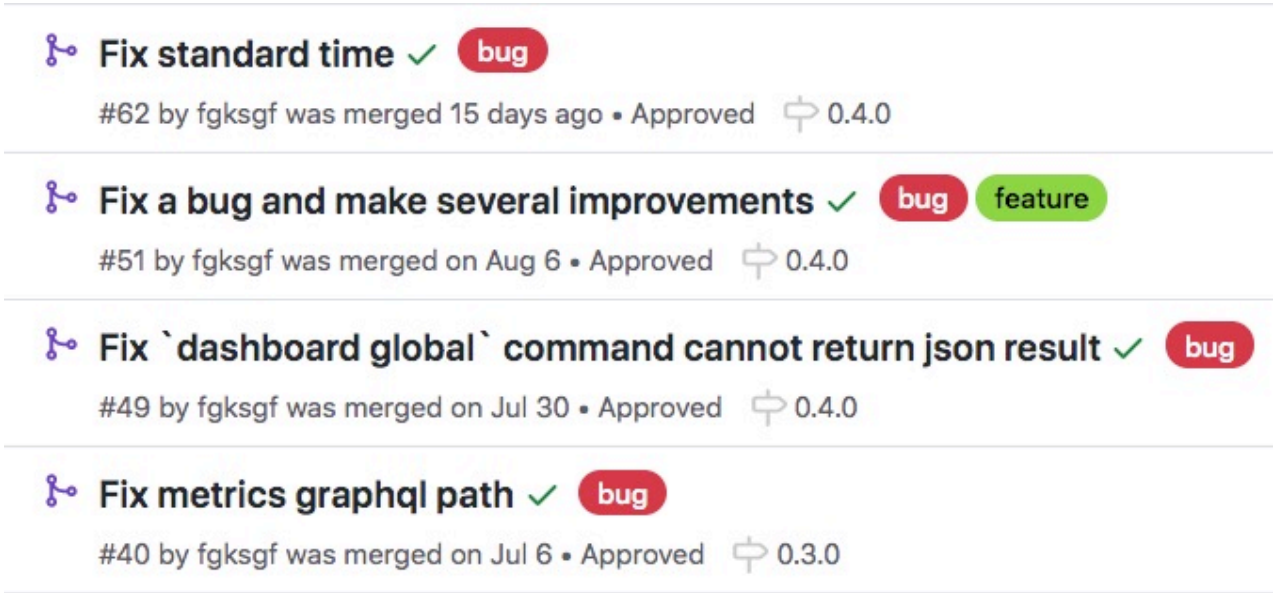
2.1 项目成果

1. 为`SkyWalking CLI`实现了一个可以查看全局数据`dashboard`视图，且`dashboard`可以**定制化**和**自动刷新**；

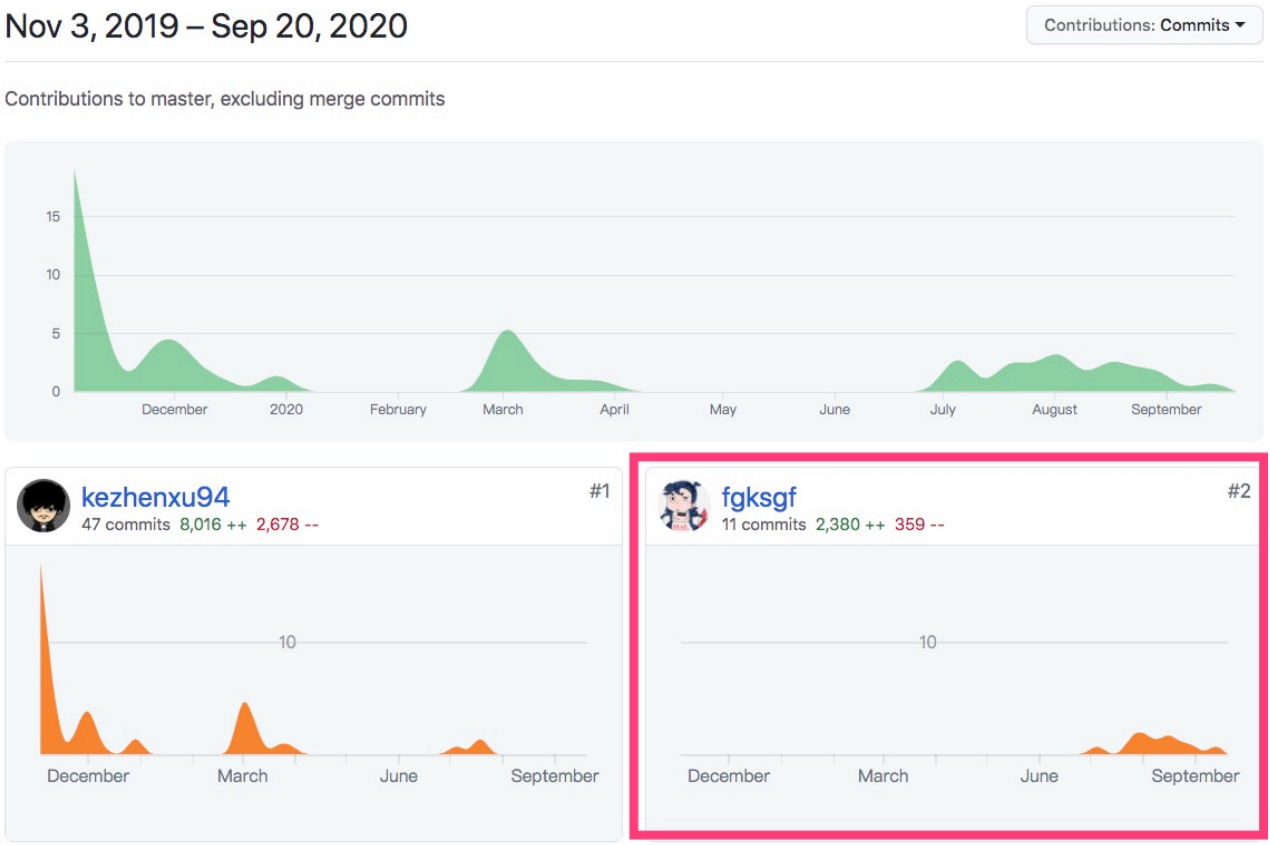




2. 在阅读项目源码过程中，发现并修复了若干 `BUG`；



3. 共提交了**11**个`Pull Requests`，代码已经过导师`code review`合并到了`GitHub`仓库的`master`分支，贡献了两千多行改动，对`SkyWalking CLI`项目的贡献数量排名第二



2.2 遇到的问题及解决方案

主要遇到了三个问题：

- `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。
- 定时刷新问题。默认的刷新间隔为6秒，如果没过6秒重新请求一次数据，时间粒度可能就会发生改变，比如从“分”变为了“秒”，且返回的数据可能与之前是一样的，造成资源浪费。于是想了一种方案：设置一个变量记录“秒”粒度的当前时间，每次更新这个时间然后转化为用户所给时间的粒度并进行比较，如果没有发生改变则不更新，否则进行更新。例如，用户给的时间段起始时间为`12:00`，粒度为“分”，经过6秒后，当前时间为`12:00:06`，将当前时间转换为“分”粒度进行比较发现时间并没有发生改变于是不会进行更新；而如果经过60秒后，当前时间为`12:01:00`，此时转换为“分”粒度后与原时间不同，因此会请求新数据进行刷新，粒度为“小时”时同理。

2.3 总结与心得

“万事开头难”，我一直都希望参与一些开源项目锻炼自己，但苦于不知从何入手。而此次的暑期2020活动则正好提供了这样一个机会：丰厚的奖金、清晰的任务要求与预期成果以及导师的联系方式，让我顺利地踏上了开源这条道路。

回顾这两个多月的参与过程，我感觉我能够顺利完成这个项目主要有以下两个原因：

- 项目处于起步期。我选择的`SkyWalking CLI`项目目前最新的版本号为`0.3.0`，还处于一个非常年轻的阶段，代码量也相对较少，而且项目结构清晰，各个目录放什么功能的代码也有文档说明，这对于我理解整个项目非常有帮助，从而能够更快地上手。
- 导师认真负责。每次我遇到什么问题，导师都能及时地回复我；对于我提交的`PR`也能够很快地被`review`，而且不时给予肯定的评论与鼓励，这极大地提高了我的成就感，让我更加积极地投入到下一阶段的工作，形成一个正向的循环。

最后，我想说，活动虽然即将结束，但我的开源之路还刚刚开始。