CS209A-Fall Final Project Report

> You should provide a written report that describes the GitHub repos you selected for this project.
>
> The written report should also introduce **the architecture design of your project**, as well as the **important classes, fields, and methods**.
>
> Finally, your report should highlight **the insights you obtained from the data analysis results**, e.g., what are **the answers to the above questions**, what can we **learn about the repo according to your answers**, and what can **be improved about this repo**, etc.

我们项目的体系结构设计大概如下：

前端部分：使用python+HTML编写，读取本地txt文件并进行数据展示。

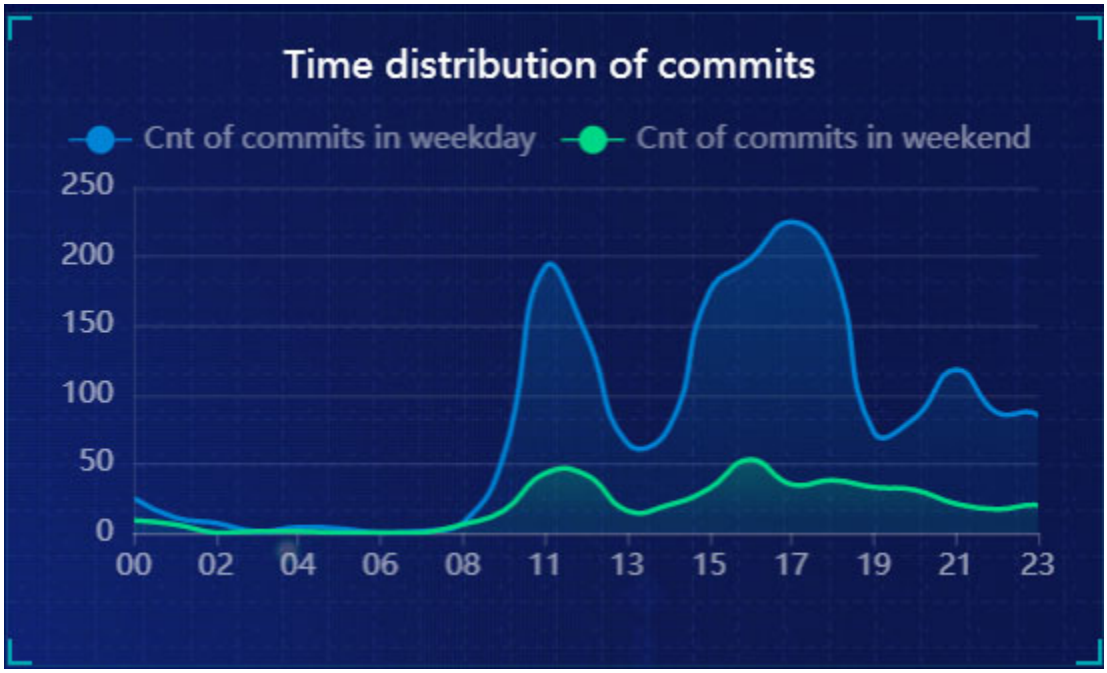
后端部分：首先是爬虫部分，用的是java编写，结构是普通的maven项目架构，主要是调用github本身的restful接口获取原始的json数据然后使用alibaba的fastjson工具进行解析处理获取到我们想要的数据，第二部分是搭建我们自己的restful接口，这里使用的是springboot的框架，当目录下的application运行后可以直接通过浏览器调用对应的restful接口获取电脑本地的数据,调用的格式为/id/rep/order。

我们所选择的Github Repos分别是

1. alibaba/**fastjson**
2. alibaba/**arthas**
3. alibaba/**spring-cloud-alibaba**

对于spring-cloud-alibaba这个repo：

- 它的developer总数为2320，commits前6位的id分别是fangjian0423，mercyblitz，flystar32，yuhuangbin，DanielLiu1123，theonefx。
- 关于issue，open issue的总数为393，closed issue的总数为2570，解决issue的时间的平均值为1074h，极值差为30,026h，方差为9,612,111.
- release的总数为30，最新7代release之间所相差的commits数量分别为0，62，173，129，50，24。其中，关于commit的时间分布如下图所示：



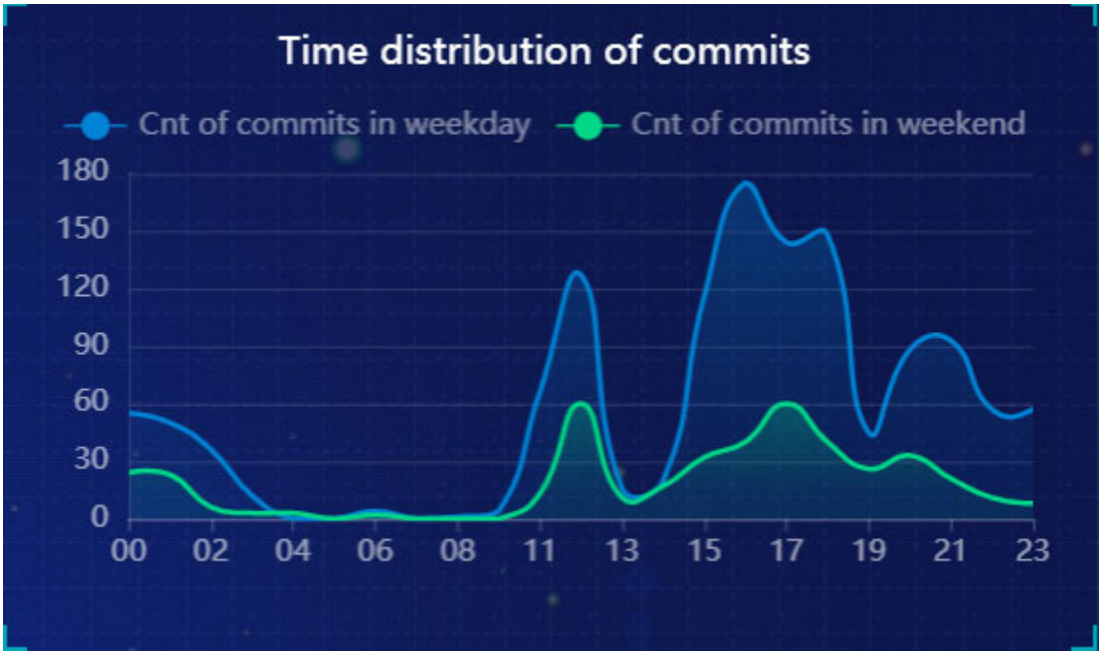
从获取的数据中我们可以得知：

- 该repo解决issue的时间跨度较大，小的issue可能几天内就能解决，而大的issue则需要几百天的时间。
- 同时，该repo的developers都比较喜欢在8~12点，14~18点之间内提交commits，在20~22点会有个小高峰，而凌晨至清晨的时间（0点~7点）则几乎没有commits。
- 对于每个releases之间，既有0个commits之差的releases，也有多至173个commits之差的releases，在一定程度上反映出版本之间更新迭代的速度和差别有所区别。
- 根据时间段划分的话，我们把时间段在(6,11]的划分为morning，(11,13]的划分为noon，(13,18]的划分为afternoon，(18,22]的划分为evening，(22,6]（晚22点到次日6点）的划分为night。结果显示，在afternoon时间段的commits是最多的，占比45.48%；其次是morning，占比21.85%。

我们认为，该repo具有足够高的commits数和developers数，但是可能还得在issues的解决这块做出改进，例如说及时关闭已经解决了的issue等；同时，对于每个releases的管理可能也还需加强。

对于arthas这个repo：

- 它的developer总数为1834，commits前6位的id分别是hengyunabc，HollowMan6，kylixs，Hearen，beiwei30，ralf0131。
- 关于issue，open issue的总数为272，closed issue的总数为2051，解决issue的时间的平均值为837h，极值差为31,199h，方差为7,298,297.
- release的总数为47，最新7代release之间所相差的commits数量分别为33，8，16，15，3，12。其中，关于commit的时间分布如下图所示：



从获取的数据中我们可以得知：

- 该repo解决issue的时间跨度较大，小的issue可能几天内就能解决，而大的issue则需要几百天的时间。
- 同时，该repo的developers都比较喜欢在9~13点，14~19点之间内提交commits，在20~22点，23~次日3点分别有个小高峰，而凌晨至上午的时间（4点~9点）则几乎没有commits。
- 对于每个releases之间，既有3个commits之差的releases，也有多至33个commits之差的releases，最多的commits之差也仅有33个，反映出版本之间更新迭代的速度较快。
- 根据时间段划分的话，我们把时间段在(6,11]的划分为morning，(11,13]的划分为noon，(13,18]的划分为afternoon，(18,22]的划分为evening，(22,6]（晚22点到次日6点）的划分为night。结果显示，在afternoon时间段的commits是最多的，占比45.01%；其次是evening，占比20.05%。

我们认为，该repo具有足够高的commits数和developers数，但是可能还得在issues的解决这块做出改进，例如说及时关闭已经解决了的issue等。

对于fastjson这个repo：

- 它的developer总数为3971，commits前6位的id分别是wenshao，VictorZeng，Omega-Ariston，'Anonymous'（匿名用户），kimmking，'Anonymous'（匿名用户）。
- 关于issue，open issue的总数为1976，closed issue的总数为2324，解决issue的时间的平均值为1909h，极值差为39,645h，方差为17,971,527.
- release的总数为84，最新7代release之间所相差的commits数量分别为25，68，82，43，56，44。其中，关于commit的时间分布如下图所示：

从获取的数据中我们可以得知：

- 该repo解决issue的时间跨度较大，小的issue可能几天内就能解决，而大的issue则需要几百天的时间。
- 同时，该repo的developers都比较喜欢在8~12点，13~19点，20~23点之间内提交commits，在0~2点会有个小高峰，凌晨至清晨的时间（3点~7点）也有少数commits。
- 对于每个releases之间，既有25个commits之差的releases，也有多至82个commits之差的releases，整体来看releases更新迭代的速度和commits之差呈现出较为均衡的趋势。
- 根据时间段划分的话，我们把时间段在(6,11]的划分为morning，(11,13]的划分为noon，(13,18]的划分为afternoon，(18,22]的划分为evening，(22,6]（晚22点到次日6点）的划分为night。结果显示，在afternoon时间段的commits是最多的，占比28.32%；其次是evening，占比23.09%；再者是morning，占比21.7%。

我们认为，该repo具有足够高的commits数和developers数，但是可能还得在issues的解决这块做出改进，例如说及时关闭已经解决了的issue等。同时，在凌晨也会有commits提交，说明developers比较国际化且都比较喜欢解决问题，~~但是太晚工作对身体不好。~~