

联系方式

- 手机: 15868759135
 - Email: zhujingdi1998@gmail.com
 - QQ/微信号: 986494553/b986494553
 - Github: [xiantang](#)
-

个人信息

- 朱静迪/男/1998
 - 本科: 温州大学城市学院计算机系
 - 技术博客: <http://xiantang.info>
 - Github: <https://github.com/xiantang>
 - 期望职位: Java后端实习生/数据采集实习生
 - 期望城市: 杭州/上海
-

开源项目和作品

1,196 contributions in the last year

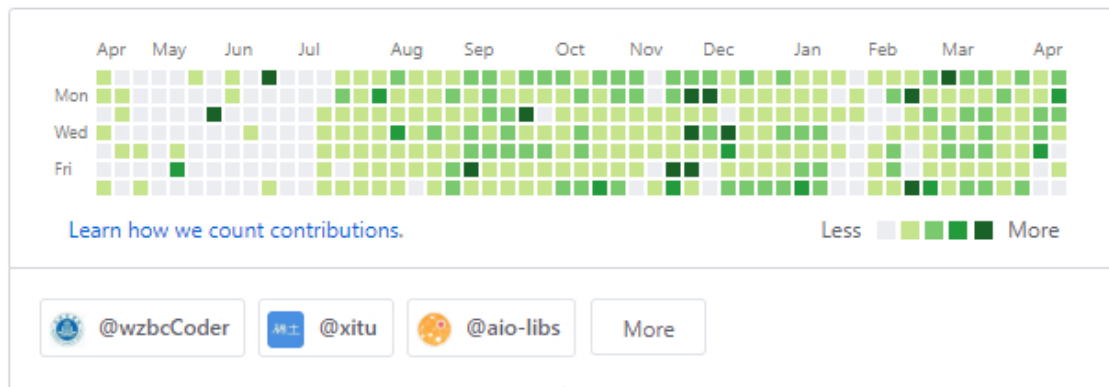


Contribution settings ▾

2019

2018

2017



Java项目

NIO WebServer

- 简介:
 - 基于Java NIO 多线程、socket网络编程、XML 解析、log4j 日志的 HTTP 服务器和 Servlet 容器。
- 技术细节:
 - 完成了对HTTP协议的部分支持, 解析POST GET 请求并且封装为自定义的 request 对象解析 url 返回对应的servlet 并封装为 response, 写入浏览器。
 - 使用NIO Reactor 模型实现复用socket连接, 并且通过反注册实现 keep-alive 长连接。
- 总结:

- 使用JMeter进行压力测试：connection:close 以下测试总请求次数都为 20000 次2个线程，每个线程循环访问10000次，吞吐量为 556 个请求/sec。

京东全站商品监控系统

- 简介：
 - 基于 scrapy-redis 和 Spring Boot 的分布式价格监控平台。
- 技术细节：
 - 使用 RabbitMQ 作为消息队列，通过 Quartz 定时扫描数据库，发送邮件提醒用户价格更新。提高性能用redis进行缓存减少数据库访问压力。
 - 使用Redis作为缓存数据库，加快了 50% 的加载速度。
 - 实现Spring-anti-spider插件，为恶意采集设置三个等级，分别是限制接口每分钟调用次数，检查 UserAgent，用户IP封禁，并运用到项目中。
 - 重构爬虫的 pipeline，使用有限状态机省去了繁杂的 if-else 代码。通过阅读和修改 scrapy-redis 源码，将原来基于 Redis set 的去重对列，改为基于布隆过滤器的实现，提高了去重的效率。
 - 使用 scrapyd 的api编写脚本控制爬虫的状态，达到增量爬取的效果。
 - 使用 docker 进行爬虫的批量部署。
- 总结：
 - 爬虫部署在三台 Ubuntu 云服务器上，并稳定爬取。
 - 只需三天就可以完成对京东的 1100w 商品数据的价格监控，每日能够记录 3W+ 条价格更新。

工作以及开发经历

2018.07-2018.09 我和我们文化传播

- 简介：
 - 负责豆瓣全站电影数据增量采集。
- 技术细节：
 - 使用芝麻代理并编写代理池，解决豆瓣IP封禁。
 - 优化爬虫架构，将评论爬取和详细页面爬取模块解耦。
- 总结：
 - 爬取 7w 电影数据，以及 500w 电影评价。
 - 使用aiohttp以多进程的方式编写爬虫框架，增量爬取周期从 7 天降低到 3 天。

技术文章

- [爬虫遭遇状态码521陷阱 破解js加密cookie](#) 收获 73 个赞同，330 个收藏。

获得奖项

- 2018第一届全国大数据竞赛二等奖
- 2018年学业三等奖学金

技能清单

以下均为我熟练使用的技能

- Web开发: Java/Python
 - 后端框架: Spring Boot/Django/SSM
 - 项目构建: Maven
 - ORM框架: Mybatis
 - 前端框架: Bootstrap
 - 数据库相关: MySQL/Redis
 - 版本管理、文档和自动化部署工具: git
 - 数据采集: scrapy/selenium/requests
 - 运维: linux terminal, docker
-

致谢

感谢您花时间阅读我的简历，期待能有机会和您共事。