

应聘Rust工程师/C++工程师(北京)_吴翔翔的简历

17150313884| me@tute.edu.cn| 男/1997/本科
技术博客:pymongo.github.io| Github:pymongo

工作经历

- | | | |
|---|---------|-------------------|
| 海南自贸区威中杰信息技术有限公司 | 全栈开发工程师 | 2019.10 - 至今 |
| · 主要负责交易所订单撮合引擎的开发(Rust)、WebSocket维护、安卓端的维护、后台管理系统的开发 | | |
| 北京邦友科技开发有限公司 | 软件测试工程师 | 2018.11 - 2019.10 |
| · 业务是汽车ECU(电子控制器)的功能测试, 以日方提供的脚本语言进行编程, 调试 | | |

项目经历

股票交易所撮合引擎	Rust开发	2020.06 - 至今
-----------	--------	--------------

[项目链接] <http://trading.sctajik.com/>

[项目介绍] 塔吉克证券是一个用塔国货币索莫尼进行股票交易的平台, 撮合引擎负责集合竞价和成交价格相近的订单

[项目背景] 基于数字货币交易所进行二次开发, 参照上交所添加了涨跌停机制、T+1交易方式、集合竞价等功能

[技术架构] 1. 开发了一个proc_macro_derive(过程宏), 可以给不同的表单结构体生成实现某个Trait的代码

通过过程宏提取了一些公共字段例如lang的getter方法、以及根据参数生成接口签名校验的方法

2. 使用泛型函数复用了检查不同表单的参数是否合法的代码, 多个接口共用一部分代码降低了维护成本

3. 通过Mutex锁实现了系统的幂等性, 用降低锁的粒度或无锁编程的方式去优化性能

4. 根据做市团队的需求, 用Rust的Rocket框架还开发了一个市场监控系统, 具有躲单、下单报警等功能

5. 集合竞价、开收盘等时间段的切换的方案是, 通过tokio异步线程内轮询时间判断时间段是否需要切换

如需切换则修改相应的全局AtomicBool, 缺点是多个股票进程没有共用一个判断时间的线程

[主要工作] 1. 设计节假日的数据结构, 做完了开休盘时间段切换、工作日/节假日切换的功能

2. 重新设计股票订单相关表的表结构, 设计股票交易的账户对账计算方式

3. 实现了当日加权平均成交收盘价的自动更新, 解决了股票订单成交后账户余额会变负数的问题

4. 接口添加了中英俄语言的国际化支持, 用Python调用谷歌翻译API将中文文案翻译成多国语言的json文件

[开源贡献] 股票交易的价格和手续费有严格的精度要求, 而Rust的Decimal类型库没有round/ceil等常用的精度转换API
我阅读相关源码实现了decimal类型的round方法, 并给bigdecimal-rs开源项目提了PR#66

[项目收获] 1. 学会管理一个Rust项目内部有多个子项目(lib crate)、多个Cargo.toml之间依赖的处理

2. 发现了thread::spawn内打印log(日志)会运行时异常, 而用tokio::spawn则不会发生该错误

3. 解决了函数中泛型入参直接获取结构体字段会报错的问题, 学会用Rust泛型实现mixin编程范式

4. 解决了获取到的时区会受到夏令时影响的问题, 学会使用impl Fn和impl Future传递同步/异步的代码块

币市/杠杠交易订单撮合引擎	Rust开发	2020.04 - 2020.06
---------------	--------	-------------------

[项目链接] <http://trading-account-type.testcadae.top/margin/omgusdt>

[项目介绍] 虚拟货币交易所的撮合引擎的功能是将价格相近的买单和买单匹配在一起进行成交

[项目背景] 老版撮合引擎是用Ruby单线程框架写的, 创建一个订单耗时40ms, 与一流交易所的撮合引擎性能差距太大

Ruby撮合引擎CPU/内存负载很高, 空闲时30%负载, 并发量高时负载会暴涨到90%需要重启去释放资源

[项目成绩] Rust重写的撮合引擎创建一个订单耗时1~2ms, Apache Benchmark并发10的测试结果是每秒处理1100个请求

[项目架构] 1. Rust技术栈主要是: actix-web+tokio+sqlx, 除了初始化应用的方法, 其余所有方法都是异步的(async)

2. 一个进程只负责一个交易对, 所有进程都是共用一个二进制运行文件, 通过环境变量/命令行参数

区分交易对, 例如请求'/btcusdt/create_order'会被nginx转发到负责btcusdt交易对的撮合进程

老版撮合是用四级域名作不同交易对的分发, 域名暴露在公网无法配置防火墙

3. 服务器启动时将数据库的订单、交易对数据加载成Rust全局变量, 减少了每次撮合时订单数据的SQL查询

4. 项目早期用的是diesel ORM, 遇到高并发下MySQL FOR UPDATE锁超时后换成sqlx

5. 不同服务器, 不同Ruby/Rust项目之间共用一个redis服务器, 利用redis pubsub机制实现通信

[主要工作] 1. 项目早期负责技术选型, 调研并学习了Go和Rust, 理解老版Ruby撮合的代码和单元测试用例

2. 使用rust-toolchain file和rustup override配置项目的Rust版本

3. 为了支持杠杠交易要对账户进行拆分, 给订单、账户表等设计新字段, 同时修改老版撮合去兼容新字段

[项目收获] 1. 异步环境下要用异步的Mutex锁, 不能用std::sync::Mutex, 否则在高并发请求时容易发生死锁

2. redis的pubsub机制实现跨语言跨进程数据共享的性能较差, 调研了TcpStream或UnixStream等方法

交易所安卓端APP

安卓开发

2020.01 - 2020.04

[项目链接] <https://github.com/wildfirechat/android-chat/pull/330>

[项目背景] 由于安卓开发的同事离职，我接手安卓项目继续开发与维护

[项目架构] 1. 一些新需求用到的库只有androidx版本，我将安卓项目升级为androidx架构
2. 老的Activity/Fragment页面保持MVP模式不变，新页面尝试使用新技术MVVM
3. 通过Parcelable序列化/反序列化接口实现两个页面之间传递Java实例对象

[主要工作] 1. 参考PC端法币交易系统的聊天功能，开发安卓端的聊天功能
2. 开发安卓端WebSocket工具，包括心跳包、断线重连、恢复订阅、取消订阅等功能
3. 我做了币币订单列表、法币广告列表、邀请返佣等共计15~20个页面
4. 新增了SQLite数据库(Room API)缓存WebSocket数据
5. 集成了友盟SDK，收集用户错误日志以及统计用户的使用信息，方便产品运营

交易所移动端API接口

Ruby开发

2019.12 - 2020.04

[项目链接] http://v-api.testcadae.top/api/v2/swagger_doc

[项目介绍] 该项目作用是给交易所移动端APP提供API和WebSocket数据

[项目架构] 1. 践行TDD，使用REST API框架grape，了解RPC，通过单元测试/swagger/postman调试接口
2. 对于时效性短的数据，存入redis减轻MySQL负载，例如当日取消订单3次就禁止交易的功能
3. 使用FactoryBot库结合Faker库通过工厂模式生成测试数据，借助单元测试进行代码重构

[主要工作] 1. 维护老的API接口，为APP的新页面开发接口，我参与开发了30~40个接口以及相应的单元测试
2. 给所有接口加上数字签名校验，同时开发了安卓端的生成签名的工具
3. 分析多表联查时出现N+1查询的原因，通过加单向/双向关联和includes语句解决N+1查询问题

开源社区代码贡献

开源项目bigdecimal-rs

contributor

[PR链接] <https://github.com/akubera/bigdecimal-rs/pull/66>

[项目介绍] 类似Python内置的Decimal，bigdecimal是Rust解决浮点数运算精度丢失的库

[主要工作] 我实现了四舍五入的round API并提供了16个测试用例，round最大可支持i128范围内的Decimal数据

开源项目sqlx

contributor

[PR链接] <https://github.com/launchbadge/sqlx/pull/319>

[项目介绍] 与Go的sqlx类似，Rust的sqlx也是一个轻量的高性能数据库工具

[主要工作] 我阅读sqlx的MySQL源码时发现docstring中有一个typo(单词拼写错误)，我提交PR修复了该错误

开源项目actix

contributor

[PR链接] <https://github.com/actix/examples/pull/298>

[项目介绍] actix是一款Rust语言Actor框架，actix/examples代码仓库主要是actix-web框架的一些例子

[主要工作] 删掉了关闭服务器代码例子中两个无用变量

开源项目野火IM

Java后端开发+安卓开发

[PR链接] <https://github.com/wildfirechat/android-chat/pull/330>

[项目介绍] 野火IM是一款仿微信的聊天软件，我参与了安卓端的开发

[项目背景] 基于野火二次开发，将交易所的聊天功能拆分成独立APP，由于视频会议效果不佳，目前项目处于暂停状态

[项目架构] 安卓端的主要技术栈是jetpack MVVM，服务端的技术栈是maven+spring，聊天协议部分没有开源

[主要工作] 二次开发野火IM的服务端+安卓端，提PR将聊天RecyclerView仅用于UI预览下的文案改成tools:text属性