应聘中级Rust服务器程序员_吴翱翔

17150313884| os.popen@gmail.com| 男/1997
技术博客: pymongo.github.io| Github账号: pymongo
leetcode账号: leetcode-cn.com/u/user0772-d

教育经历

天津职业技术师范大学(本科) 电子信息工程 2015 - 2019

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

· 主要负责数字货币交易所订单撮合引擎、WebSocket服务器、安卓端APP、PC端Vue页面的开发

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

WebSocket服务器 Rust开发 2020.07 - 至今

[项目链接] wss://ws.cadex.top
[项目背景] 老的Ruby版WebSocket服务器负载高内存泄露严重，需要每隔5分钟去重启一次去释放资源
[技术架构] 1. 服务器不生产任何数据，每个WebSocket频道各有一个Ruby脚本通过进程通信每隔2秒推数据给服务器
2. 弃用老版基于redis pubsub的进程间通信方式，改用效率更高性能更好UDP socket进行通信
3. 用的是actix框架，主要有两种Actor，一个是跟每个客户端一对一的SessionActor
另一个Actor是UDP socket Listener兼任存储每个频道有哪些客户端订阅
UdpActor会缓存其它Ruby脚本发来的上一条推送数据，一旦客户端订阅频道就立即推送缓存的数据
SessionActor的析构函数中会通知UdpActor删掉所有频道的订阅列表中已断开连接的Actor地址
[主要工作] 1. 设计WebSocket断线重连/心跳包机制，用Rust重写负载较高的聊天应用中WebSocket数据推送脚本
2. 尝试用Rust开发安卓端和IOS端的WebSocket客户端库，提高移动端的性能
[项目收获] 1. 在Actor Handler的同步函数中嵌入异步的代码块的两种方法
2. 在Xcode上通过Bridging Header给Mac/IOS引入Rust库，安卓引用Rust编程成的so库文件

数字货币交易所订单撮合引擎 Rust开发 2020.04 - 至今

[项目链接] http://trading.cadex.top
[项目介绍] 数字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的测试结果是每秒能处理400个下单请求
[技术架构] 1. Rust技术栈主要是：actix+tokio+sqlx，曾经用过diesel但由于无法满足某些查询已弃用
2. 撮合引擎本质上是一个对性能要求较高的API服务器，主要由创建订单/撤单/撤销全部三个接口组成
三个接口都需要将各参数按字典序结合private_key算出数字签名，Rust没法按字典序遍历结构体字段
用固定的format!去生成签名字符串维护成本太高，我开发了一个derive过程宏
可以让每个接口的表单结构体自动生成数字签名，以后增删字段也不用改代码
3. 使用泛型复用了所有接口表单中公共参数的校验，例如检查lang字段合法就是通过泛型实现
4. 将未完成的订单数据用「优先队列/堆」的数据结构持久化存储在内存中，减少MySQL的负载
撮合引擎的买单和卖单本质上是个有序数组，根据leetcode刷题经验，动态有序数据用堆，静态用快排
5. 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
6. 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
[主要工作] 1. 开发撮合引擎的单元测试宏，例如下单宏，验证账户余额宏，让不懂Rust的同事也能通过宏去编写单元测试
2. 管理一个大型Rust项目，内部有多个子项目(package)、多个Cargo.toml之间依赖的处理
3. 设计限价交易下单、市价下单、止盈止损下单的三种下单方式，理解杠杠交易的做空做多业务逻辑
4. 开发撮合成功后更新redis中最新价格的功能，将actix-redis重构成async-redis，测试redis断线重连
[项目收获] 1. 异步环境下要用tokio或async-std的Mutex锁，用std::sync::Mutex容易发生死锁
2. 简单了解了Rust和C++的多态/泛型/虚函数表，以及Rust如何实现静态分发和动态分发(dynmatic dispatch)

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. 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. 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的系统，具有躲单、下单报警等功能
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. 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
2. 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
3. 发现了rust_decimal库，不仅内存大小只有bigdecimal-rs的1/4，而且运算速度是bigdecimal的2-3倍

交易所安卓端APP 安卓开发 2020.01 - 2020.04

[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. 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
2. 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
3. 新增了SQLite数据库(Room API)缓存WebSocket数据
4. 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://new-api.cadex.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

Rust开源社区贡献

开源项目bigdecimal-rs contributor

[PR链接] https://github.com/akubera/bigdecimal-rs/pull/66
[项目介绍] bigdecimal-rs是Rust社区一个适用于较大数据之间精准运算的库
[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

[项目介绍] 与Go的sqlx类似，Rust的sqlx也是一个轻量的数据库工具
[PR#581] Remove decimal feature unnecessary dependency num-traits
[PR#319,PR#592] 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