张笑冬

邮箱：zhaxd123@126.com

手机号：13917846274

教育背景

2015.08-2017.10	**瑞典皇家理工学院**	**通信系统（网络工程方向）**	**硕士**
2011.09-2015.06	**东南大学**	**通信工程**	**本科**

个人优势

- 熟练掌握 **C++**，具备**良好的编码规范**和大型系统下的**架构优化**经验
- 熟悉常见**设计模式**，具备 **UML** 使用经验
- 熟悉 **Linux** 操作，具备 **Bash** 脚本编写经验，具备 **Linux** 环境下开发经验
- 具备**数据结构与算法**、**计算机网络**、**操作系统**相关理论知识
- 具备 **Python** 开发经验，熟练使用 **Git**、**SVN** 等主流版本控制工具进行团队协作开发
- **华为可信考试 C++专业级认证，证券投资基金业从业资格证书**

工作经历

2023.06-至今　　仲阳天王星量化　　C++核心开发工程师

个人职责：

- 期间主要承担期货项目开发工作，负责期货行情模块的开发和部署工作，alpha 信号数据的实盘化适配工作，期货仓位的配置和校对模块开发以及 CTP 柜台开发工作；
- 完整承担了期货项目实盘上线工作，并持续维护上线后续的可靠性提升和盘后回测数据分析工作；

2017.12-2023.06　　华为技术有限公司　　C++软件开发工程师

个人职责：

- 承担 5G 通信系统开发工作，负责小区算法模块的 C++开发和架构设计，确保组件的代码框架能够应对产品的持续迭代和需求演进；
- 先后承担多个业务需求负责人，能够高效协调周边组件达成高质量交付，月均代码开发量约 2KLOC；

项目经历

2023.06-至今　　期货柜台开发和持续维护　　开发负责人

个人职责：

- 负责 CTP 交易柜台的开发和维护工作。上游对接内部的交易算法单元模块；
- 实现了柜台的登陆授权管理、结算管理、订单管理、仓位管理以及资金管理等功能的开发；
- 支持基于本地费率表和在线费率查询的自动锁仓判断功能；

- 支持定时询仓和仓位校准功能（配合上游的仓位较准模块）；
- 实现柜台风控模块的设计和开发，在设计上能够做到风控逻辑和数据分离解耦，支持以表驱动的方式注册风控项函数并由底层框架定时执行。风控间隔支持用户文件灵活配置。当前风控支持持仓风险度控制和持仓合约临近交割风控功能等；

2023.06-至今　　期货行情开发工作　　开发负责人

个人职责：

- 负责上期所 CTP 行情接收模块开发工作；
- 负责大商所 Level2 nano 行情接收模块开发工作；
- 负责对接中信期货郑商所 Level2 行情接收模块开发工作；
- 承担宏汇 TDF 期货行情开发工作；

2023.06-至今　　Alpha 信号数据实盘化改造工作（Python）　　开发负责人

个人职责：

- **实盘化 tradecode 改造**
- 实盘改造信号数据处理流程，将研究员提供的基于文件和矩阵的离线版本代码改为基于流式输入的实时计算版本，同时对接上游实时行情数据流，使用文件映射和共享内存的方式对模块间数据流的传递加以改造；
- 在后续测试过程中持续针对实时模式和离线模式的信号进行校准和差异修复的工作；
- **仓位配置模块开发**
- 开发仓位配置模块，上游对接研究信号生成的仓位权重数据，下游对接交易系统；
- 该模块主要完成分时点的任务拼接，同时支持合约换期的清仓处理以及基于给定面值的分时点仓位计算和文件导出功能；

2023.06-至今　　盘后数据分析（Python）　　开发负责人

个人职责：

- 基于当日数据运行盘后回测，并根据结果进行当日理论收益分析，生成天级、周级、年化统计报表；
- 基于当日行情延时数据生成日内延时统计报表；
- 基于盘后回测对比结果生成盘中实时信号差异报表；
- 基于策略预热结果生成次日的合约换期报表；
- 自动化整合报表并以邮件形式定时发送给相关人员；

2023.06-至今　　期货交易可靠性　（Python）　　开发负责人

个人职责：

- **行情延时监控：**支持 tick 级行情延时监控管理，大于给定阈值实时告警，支持盘后数据导出以供分析；
- **仓位对比功能：**日内分钟级仓位对比和告警，同时附带支持换期旧合约未清仓告警和锁仓告警功能；
- **费率变更提醒：**对比当前最新柜台查询费率和本地理论回测费率差异并告警，同时支持两者的自动适配；

2018-2023　　5G 通信系统业务需求开发　　C++开发负责人

个人职责：

- 承担 5G 算法组特性需求的架构设计、统筹开发及代码检视工作，参与完成 200+个特性需求的开发和交付工作；

- 重点负责调优保留参数、互调干扰检测（PIM）、符号自适应等大粒度特性的方案分析和开发工作。在 5G 产品引入风险排查包机制，确保软件版本升级可靠性；

2018-2019　　保留参数机制优化和组件透传通道的设计与开发　　C++开发负责人

个人职责：

保留参数是一种实验性质的调优参数，在产品演进过程中频繁增删和修改，引入了大量的冗余代码，造成代码可读性降低和性能损失。基于这样的业务痛点完成如下架构优化：

- 自顶向下的系统分析，将参数结构由原有的单键扁平化布局优化为双键多实例布局，原有的数据层静态下发调整为代码层动态创建，实现数据实体创建流程下沉；
- 重新制定保留参数的模型默认值方案原则，以及代码映射机制的开发，实现配置面设计原则的统一；
- 引入组件间配置透传通道机制，避免中间组件对数据的不必要感知，大幅降低参数零散配置的冗余代码；
- 本次重构将相关代码规模由 5KLOC 降低至 2KLOC，代码精简约 60%。架构优化后，每新增一个参数所需工作量由百余行降低至 10 行代码即可完成，从而显著提升保留参数启用的开发效率；

2019-2020　　事件中心服务框架开发　　C++开发负责人

个人职责：

- 在早期代码中过度随意的使用函数回调进行状态，导致调用关系散乱，不同特性间的代码存在紧密耦合。为此我们在系统内增设统一的事件中心模块，采用发布订阅模式集中管理不同特性事件之间的监听和回调关系；
- 引入事件中心后，通过将原有回调函数形式的状态通知机制抽象为不同的事件项注册到事件中心中，发布者将状态以对应事件项的形式通知事件中心，监听者只需在事件中心中订阅自己关心的是事件项，发布和订阅行为均只和事件中心交互，从而达到特性间松耦合的设计目的；

2020-2021　　算法模块事件订阅机制设计与开发　　C++开发负责人

个人职责：

早期代码事件中心的使用存在事件项制定随意、发布和订阅代码零散的问题。一些本质相同的状态被作为不同事件项独立发布和订阅，导致长期代码难以维护。此问题的根源在于，业务代码一般是依据场景进行编写，而事件的发布和订阅则应以事件维度进行组织。本人对组件代码进行如下的分步优化：

- 对存量事件项进行抽象整合，提取出正交的事件项集合并完成统一收编；
- 使用代理模式对每个事件项定义单独的代理类，专门负责和事件中心的交互工作，处理事件的注册和分发；
- 定义事件处理基类作为处理行为的抽象，针对具体事件项分别派生不同的具体处理类。处理类向代理类注册自己关注的事件类型，并在订阅响应动作中负责进一步分发到下级的实际订阅者；
- 使用建造者模式，将处理和通知行为解耦。处理类依据事件项维度加以组织，通知类依据被通知模块的维度加以组织。事件处理类会根据订阅的模块组合不同的通知类，而一个具体通知类仅需和一个特定模块进行交互；
- 经过重构后的代码，逻辑清晰。分层式行为架构保证了各功能模块的单一职责原则，同时提供了更好的扩展性；

2022-2023　　5G Rantrans 无线翻译设计平台与开发　　C++开发负责人

个人职责：

- 承担 5G 产品侧无线翻译工具 Rantrans 从零到一的设计和开发工作。该项目旨在面向 5G 产品部门提供线上翻译提交的平台。工具集成资源文件扫描、远端自动提交、翻译进度查询、翻译一键回导等功能于一体，将原本需要手动导出邮件提交的翻译流程做到线上自动化；
- 工具直接对接线上语料库平台，大幅降低重复翻译的内容。该工具引入后，与原有的人工提交相比，翻译侧工作量降低 75%，翻译周期从 3 天缩短至当日回稿，帮助开发侧翻译效率提升近 3 倍；
- 项目代码采用了松耦合设计，将源文件扫描、翻译导出和提交、翻译线上状态查询、翻译回导分属到不同职责模块独立完成。将基本的解析功能、回导功能代码上移到抽象基类，有效保证了代码的复用性和灵活性。代码整体架构合理，易于扩展，本项目已在公司内部开源并持续维护，项目获得华为无线公司部门优秀实践奖项（QCC）；

科研经历

2017　　Multi-Cloud Simulation environment for WebRTC streams

- 在地理分布式多重云架构下，用户对于计算资源的获取会很大程度依赖于分布式主机的物理位置和网络链路条件；
- 针对以 WebRTC 为代表的的分布式端到端流式传输模型，本文提出并实现了一种基于 Python/Mininet 实现的 Multi　Cloud 网络仿真器。能够依据用户指定的网络拓扑配置和调度策略，自动化模拟音视频数据流在多数据中心之间的传输，并进行性能测量；
- https://kth.diva-portal.org/smash/get/diva2:1165928/FULLTEXT01.pdf