

钱立超

C++ 开发工程师 | Linux 后端 | 高并发实时数据

15829259537 | xdeduvip@gmail.com | 3 年 C++ 后端开发经验

个人优势

- **C++ 与并发**：熟悉 C++11/14/17、STL、RAII、智能指针、模板、继承与多态；掌握 thread、mutex、condition_variable、atomic 等并发工具，能够处理资源生命周期、线程同步、并发分片和批处理任务。
- **AI 研发工程化**：能将领域经验结构化为包含上下文约束、代码模板、检查清单、测试验证和验收规则的可执行 Skill，形成“AI 生成/修改—自动校验—人工审查”闭环，将个人经验转化为团队可复用的研发资产。
- **高并发网络与协议**：熟悉 TCP、HTTP/REST、WebSocket/WSS、TLS、JSON-RPC、Protobuf、YAML；具备 Boost.Asio/Beast、OpenSSL、CPR/libcurl、zlib 实践，能够设计事件循环、连接分片、串行写队列、ACK、限频、心跳、分层超时和指数退避重连。
- **实时数据与大数据量处理**：实践进程内消息总线、有状态 Dataset、事件时间、乱序/过期过滤、按 Key/高度 Hash 分片、有界队列、批内去重、条件变量触发及“锁内交换、锁外组批”，能够平衡吞吐、时延、顺序性和数据正确性。
- **存储选型与时序入库**：熟悉 MySQL 与 QuestDB 的适用场景，负责 QuestDB 技术调研、模型设计、ILP v2 批量写入、连接管理和部署落地，面向持续追加、时间范围查询和海量明细分析优化存储链路。
- **可观测与工程化**：设计 Prometheus 指标、Grafana 看板、异常上报与告警闭环；熟悉结构化日志、健康检查、录制回放、ASan、CMake、动态库插件化和 Docker 多阶段构建。

教育经历

西安电子科技大学	2020.09~2023.06	网络与信息安全	硕士研究生
西安电子科技大学	2016.08~2020.07	网络工程	本科

工作经历

同花顺（杭州星锐网讯科技有限公司） | C++ 后端开发工程师 | 2023.07—至今

- 基于 C++17/Linux 构建高并发实时数据链路，技术范围覆盖异步 Adapter、进程内消息总线、有状态处理、协议分发及 QuestDB 时序入库，负责接入与管理数据源包括 **8 家交易所**、**3 家银行**、**Bitcoin 账本**和 **3 类数字协议资产**。
- 将交易所接入全流程沉淀为 AI 领域 Skill，并在 3 家数字货币交易所复用，使单家接入周期由约 **2 周**缩短至 **3 天**，按 **10 个工作日**口径缩短约 **70%**。
- 推动 QuestDB 在组内链上时序数据场景落地，完成存储选型、表模型、ILP v2 批量写入、连接自愈、指标监控和容器化交付，形成可复用的时序存储基线。

- 建设 Prometheus 指标、Grafana 看板、异常上报和分级告警，围绕连接中断、订阅失败、队列积压、数据过期、同步失败与写库异常建立可观测闭环。

项目经历

多交易所数字货币行情接入与管理

项目概述： 针对多交易所协议差异、并发订阅扩展和重复开发问题，负责 8 家交易所实时数据接入，建设统一 Adapter/Protobuf 链路、可观测体系及可复用的 AI 领域 Skill。

技术栈： C++17、Boost.Asio/Beast、OpenSSL、HTTP/REST、WebSocket/WSS、JSON、Protobuf、YAML、zlib、Prometheus、Grafana、CMake、Docker、AI 领域 Skill

关键实现：

- 面向大规模交易对并发订阅，将每个标的拆分为 **Ticker+日 K 两类 Topic**，按单连接 **200 Topic**（约 **100 个标的**）稳定分片；代码支持最多 **60 条连接**，按当前参数计算理论分片容量约 **6000 个标的**。
- 实现增量订阅与连接分配复用，配置 **10 次/秒**订阅限频、**5 秒 ACK** 超时和最多 **3 次**重试，断线重连后按原分片自动恢复，降低订阅风暴和行情空窗风险。
- 构建 **5 秒心跳**、**15 秒无读检测**、**1~30 秒指数退避**及 **0~500ms jitter** 的连接容错机制；为 TCP/TLS/WSS/写操作设置 5 秒分层超时，并增加 **1MiB** 入站帧保护。
- 动态码表按 **30 分钟**周期刷新，失败后 **30 秒**重试，支持最多 **100 页**、**单页最多 4 次请求尝试**及 **8MiB 响应保护**；新快照校验失败时保留旧映射与在线订阅。
- 将 Ticker 与 UTC 日 K 按事件时间合并，过滤时间回退、乱序、跨日和过期数据；有状态处理链路配置 **20 万条消息队列**，并以 Ticker **2 分钟**、日 K **2 天**的新鲜度窗口控制数据质量。
- 建设连接、订阅 ACK、码表刷新、最后收包、队列水位、数据新鲜度和发布状态等指标，接入 Prometheus/Grafana；对连接超时、订阅失败、数据断流及过期异常主动上报并触发告警。
- 将协议分析、码表接入、连接状态机、字段映射、测试用例、监控项和交付检查清单固化为 AI 领域 Skill，形成生成、自动校验与人工复核闭环。

项目成果： 完成 **8 家交易所**统一接入和标准模型输出，形成单连接 200 Topic、最大 60 条连接、理论约 6000 个标的的分片设计容量；有状态链路配置 20 万条消息队列并实现 7×24 小时可观测。Skill 在 **BingX、OKX、Gate.io** 3 家交易所复用后，单家开发周期由约 **2 周**缩短至 **3 天**，按 **10 个工作日**口径缩短约 **70%**。

Bitcoin 链上账本采集与 QuestDB 时序入库

项目概述： 针对 Bitcoin 链上数据持续增长及 MySQL 在海量追加写和时序查询中的成本问题，推动 QuestDB 落地，负责采集、事件分发、并发批处理、时序入库、连接自愈和监控告警链路建设。

技术栈： C++17、Linux、Bitcoin Core JSON-RPC、CPR/libcurl、JSON、Protobuf、YAML、Asio、MySQL、QuestDB ILP v2、Prometheus、Grafana、CMake、Docker

关键实现：

- 针对链上数据“持续追加、写多改少、按时间/高度范围查询”的负载特征，对比 MySQL 行存、事务与索引维护开销，选用 QuestDB 时序分区、列式存储和 ILP v2 批量写入，并沉淀模型、连接、监控及部署规范。
- 封装 getblockcount/getblockhash/getblockstats/getblock 4 类 Bitcoin Core RPC，单区块通过 3 次 RPC 完成哈希、统计与详情采集，统一设置 60 秒请求超时，并对异常高度最多重试 5 次。
- 建设指定高度区间历史同步、启动补数、实时追块和本地高度水位续跑，以 Protobuf 和进程内事件总线解耦网络采集与数据库写入，避免慢 I/O 反压采集线程。
- 设计 Block、Transaction、Vin、Vout 4 类 Protobuf 模型、34 个结构化字段和 4 张 QuestDB 时序表，支持基础区块与可选交易明细采集模式。
- 实现可配置工作线程与按区块高度 Hash 分片，使用 4 组缓冲、批内复合键去重、条件变量阈值/超时触发及“锁内交换、锁外组批”；设置单批上限 1 万条、9950 条阈值唤醒和默认 1 秒超时兜底，降低数据库 I/O 对接收线程的阻塞。
- 基于 QuestDB ILP v2 实现 4 条写表路径，为每个目标端点和工作线程维护 Sender；存储侧每秒检查连接、异常后置空并重建，每 5 秒刷新写入统计。
- 建设 11 组同步成功/失败、失败高度、最新高度、积压、写入成功/失败、批次及连接状态指标，接入 Prometheus/Grafana；对同步失败、积压增长、写入异常和 Sender 断开主动上报并触发告警。
- 以 3 个动态插件解耦 HTTP 服务、BTC 采集和 QuestDB 存储，配套 Docker 两阶段构建及进程/指标端口双项健康检查；参与 Runes、BRC-20、Ordinals NFT 等协议资产数据接入与管理。

项目成果：完成 QuestDB 在组内链上数据场景的工程化落地，形成 4 类模型、34 个结构化字段、4 张时序表及统一 ILP v2 写入基线；批处理支持单批 1 万条、9950 条阈值唤醒和 1 秒超时兜底，并建立 11 组指标、Grafana 看板、异常告警、Sender 自动重建和双项健康检查，为后续链上时序数据接入提供可复用基础设施。

多银行积存金实时行情接入与分发

项目概述：针对三路 TCP/二进制数据源的并发接入、时间轴差异和状态污染风险，负责建设多路异步采集、独立状态管理、有序快照更新、实时协议分发及监控告警链路。

技术栈：C++17、Asio/TCP、自定义二进制协议、JSON、Protobuf、YAML、进程内消息总线、有状态数据集、Prometheus、Grafana、CMake、Docker

关键实现：

- 为 3 路上游分别创建独立 Asio 客户端、事件循环和 I/O 线程，并行完成二进制报文接收与解析；下游通过 1 条串行消费线程更新有状态快照，兼顾多源并发接入与单 Key 时序一致性。
- 构建 3 路 DataID → 3 套独立 Dataset → 3 个实时分发 Index 的一一映射，隔离代码表、状态、时间轴和故障域，同时以模板复用公共快照合并逻辑。

- 实现 **INIT** → **INITOK** → **全量快照**三阶段初始化，设置 **60 秒**阶段超时与 **10 秒**会话巡检；超时或数据许可变化时自动重建初始化状态。
- 覆盖普通交易日、周五、节假日、特殊日期 **4 类时间规则**；以 **3 秒**周期检测行情延迟/断流、**30 秒**周期检查交易日，并支持连续 **2 次**重连失败后切换备用端点。
- 使用有界队列和串行状态更新避免并发写入导致快照时间倒退，配套队列水位、溢出/恢复、PPS、包序号和最后收包时间等背压诊断能力。
- 建设 **13+** 项连接、流量、源/本地时延、丢包、队列、处理耗时及业务状态指标，接入 Prometheus/Grafana；对连接断开、数据延迟、丢包、队列溢出和状态异常进行上报并触发告警。
- 支持静态码表初始化、全量/增量协议转换、按交易日录包及文件回放，并完成收盘 OHLC 数据上传，提升复杂时序边界的回归验证和问题复现效率。

项目成果：完成 **3 路 DataID**、**3 套独立 Dataset**、**3 个实时分发 Index** 的隔离链路，覆盖 **4 类日期规则**和三阶段初始化；形成 **13+ 项指标 + Grafana 看板 + 异常告警**的诊断体系，并通过按交易日录包回放实现初始化、实时、节假日和收盘边界的可重复验证。