Bai Zixuan

+86-156-5070-5080 bzx0619@gmail.com github.com/napallday

Basic Information

- Nationality: Chinese, **Singapore Permanent Resident** (No visa sponsorship required)

Education

Bachelor, Electronics Engineering, Peking University(北京大学 信息科学技术学院电子学系) | *Beijing, China*
- GPA: 3.6/4.0 (Rank:10/50); **Dual Degree: Economics** **Sep 2014 - July 2018**

Master, Signal & Information Processing, Peking University | *Beijing, China* **Sep 2018 - April 2021**
(voluntarily quit master program because it's not related to computer science)

Awards & Publications

- 2019 IEEE Student Competition **First Prize**
- Publications: ICASSP, INFOCOM, ICC, IEEE IoT Journal, IEEE Network

Working Experience

Senior Software Engineer, Marketplace Tech Services, Shopee *Singapore* **May 2021 - June 2024(3 years)**

- Responsible for building reliable and sustainable technical products and providing solid solutions in middleware areas to empower the technical robustness and sustainability for Marketplace product lines and backend teams
- **Promoted to Senior Software Engineer in one year**
- **Projects**
  - **In-house Golang Kafka Client Library**

    Built an in-house unified Golang Kafka client library in Shopee, used by **20+ teams** in **2000+ projects**. The design initiated from experiences and best practices, making Kafka interactions performant, safer and easier. I also gave a company-wide technical sharing on Kafka

    - Supported Tracing, Metrics, Full Chain Stress Test, Message Persistence, Traffic Routing(forward, double write, filter), Config Hot Reload
    - Built an **out-of-the-box consumer model** -- With this model, it only takes several minutes to build up a performant and reliable consumer service from scratch

      Core features includes:

      - Concurrent Message Transformation
      - Message Processing in different concurrent levels(by partition, by key, by any field in biz model, randomly)
      - Hot Retry, Multi-layer Cold Retry, Dead Letter Queue
      - Message Batch Processing
      - Advanced Offset Management to avoid message loss
    - Provided Kafka SPI/SP in a company-wide service development framework(like SpringBoot)
    - Avoided common pitfalls(infinite rebalance), integrated best practices(new partitions automatic detection) and optimizations(best-effort message de-duplication), etc
  - **High Performant Message Transformation/Cache Invalidation Platform**

    A high performant & scalable solution to monitor Database events for cache invalidation or message format transformation

    - Built a light-weight and concurrent ETL pipeline framework, adopted in 1000+ projects
    - **Cache Invalidation Platform**: A centralized place where developers can effortlessly manage their cache invalidation process

    - Read from Kafka binlog messages to invalidate cache
    - Adopted delay double deletion policy to avoid DB slave delay issue
  - **Message Transformation**: A general solution that transforms database binlog to customized data model by Kafka
    - Features include: Data Filtering, Field Mapping, Field Deletion, Field Generation, Triggering Conditions, Debounce, Re-partition, Split by Region, etc

Intern, Ads Infra, ByteDance *Beijing* **April 2020 - September 2020 (6 months)**

Open Source Contributions And Personal Projects

IBM Sarama - Golang Client for Apache Kafka with over 11k stars

- Implemented **incremental cooperative rebalance protocol**
  - Rewrote core consumer logic including setup/cleanup, rebalance, offset management and heartbeat
  - **Eliminated STW(Stop-The-World) problem** during rebalances, reduced rebalance overhead by 90+% and made a company-wide tech sharing
- Resolved the long-standing issue of **automatic partition detection**, which is the main reason Alibaba Cloud's recommend not using Sarama
- Implemented **multiple rebalance strategies support**, enabling zero-downtime rolling upgrades
- Fixed various performance issues including **race conditions**, **CPU burning**, and **memory optimization**

Collateralized Lending Protocol - Solidity DeFi Project

A lending protocol allowing users to mint synthetic stablecoins against their collateral

- Implemented **floating bonus mechanism** to incentivize liquidators and enhance protocol robustness
- Applied **CEI(Check-Effect-Interaction) pattern** and added reentrancy guard to avoid reentrancy attacks
- Integrated with Chainlink Price Feeds and added **stale price protection**
- Applied several **gas optimization** tricks
- Used **static analysis** tools(slither, aderyn) to find potential issues
- Developed comprehensive test suite with mock tests and **handler-based invariant(stateful) fuzzing** using Foundry

Huff Counter - Low-level Smart Contract Implementation

- Implemented the same counter contract in four different abstraction levels: **Solidity, Solidity & Yul, pure Yul, and Huff**
- Applied **differential fuzzing tests** to ensure behavioral consistency across implementations

Skills

- **Programming Languages**: Golang, Java, Solidity
- **Web2**: Kafka, Redis, MySQL
- **Web3**: Solidity, Foundry, EVM(opcodes)
  - Familar with Common Attack Vectors, Gas Optimization, Common ERC protocols, Uniswap, Compound, etc
- **Strengths**: Eager to learn new things, Problem Solving, Communication

白子轩

+86-156-5070-5080 bzx0619@gmail.com github.com/napallday

基本信息

- 国籍：中国，新加坡永久居民 (无需签证)

教育背景

北京大学 信息科学技术学院 电子学系 学士 **2014 年 9 月 - 2018 年 7 月**

- GPA: 3.6/4.0 (排名: 10/50); 双学位: 经济学

北京大学 信号与信息处理 硕士（因方向和计算机无关，主动选择退学） **2018 年 9 月 - 2021 年 4 月**

科研成果

- 2019 年 IEEE 学生竞赛 一等奖
- 论文发表：ICASSP, INFOCOM, ICC, IEEE IoT Journal, IEEE Network

工作经历

Senior Software Engineer, Marketplace Tech Services, Shopee | *新加坡* **May 2021 - June 2024 (3 年)**

- 负责构建可靠且可持续的技术产品，为 Marketplace 产品线和后端团队提供中间件领域的解决方案，提升技术稳定性和可持续性
- 一年即晋升为 **Senior Software Engineer**
- 项目经历
  - **公司内 Golang Kafka 客户端库**

    为 Shopee 构建统一的 Golang Kafka 客户端库，被 **20 多个团队的 2000 多个项目使用**。基于最佳实践，使 Kafka 交互更高效、安全和便捷。同时进行了全公司范围的技术分享。

    - 支持链路追踪、监控指标、全链路压测、消息持久化、流量路由（转发、双写、过滤）、配置热更新
    - 构建**开箱即用的消费者模型** -- 使用该模型，几分钟内即可从零搭建高性能可靠的消费者服务

      核心功能包括：

      - 并发消息转换
      - 多级并发处理（按分区、按 key、按业务模型字段、随机并发）
      - 热重试、多层冷重试、死信队列
      - 消息批处理
      - 高级偏移量管理，避免消息丢失
    - 在公司级服务开发框架（类似 SpringBoot）中提供 Kafka SPI/SP
    - 避免常见错误使用（无限重平衡），集成最佳实践（新分区客户端自动检测）和优化（消息去重）等
  - **高性能消息转换/缓存删除平台**

    用于监控数据库事件的高性能可扩展解决方案，用于缓存失效或消息格式转换

    - 构建轻量级并发 ETL 框架，在 1000 多个项目中采用
    - **缓存删除平台**：开发者可以轻松管理缓存失效流程的集中平台
      - 从 Kafka 存储的 binlog 消息读取，删除缓存
      - 采用延迟双删策略避免数据库从库延迟问题
    - **消息转换**：通过 Kafka 将数据库 binlog 转换为自定义数据模型的通用解决方案
      - 功能包括：数据过滤、字段映射、字段删除、字段生成、触发条件、去抖动、重分区、按地区拆分等

实习生，广告基础架构部，字节跳动 | *北京* **2020 年 4 月 - 2020 年 9 月**

开源贡献与个人项目

IBM Sarama - Golang Kafka 客户端（11k+ Stars）

- 实现了增量协作再平衡协议
  - 重写消费者核心逻辑，包括启动/清理、再平衡、偏移量管理和心跳等
  - **消除了再平衡过程中的 STW 问题**，将再平衡开销降低 90% 以上，并进行了公司级技术分享
- 解决了新分区自动检测的长期问题，这是阿里云不推荐使用 Sarama 的主要原因
- 实现了多重平衡策略支持，实现零停机滚动升级
- 还修复了各种性能问题，包括竞态条件、**CPU 占用过高和内存优化**等

外生抵押借贷协议 - Solidity DeFi 项目

一个允许用户抵押资产铸造稳定币的借贷协议

- 实现**浮动奖励机制**以激励清算人，增强协议稳定性
- 应用 **CEI(检查-生效-交互)模式**并添加重入保护以防止重入攻击
- 集成 Chainlink 价格预言机并添加**过期价格保护策略**
- 应用了多种 **gas 优化** 技巧
- 使用**静态分析**工具(slither, aderyn)发现潜在问题
- 使用 Foundry 开发全面的测试套件，包括 Mock 测试和基于 **handler 的不变量（状态）模糊测试**

Huff 计数器 - 底层智能合约实现

- 用四种不同抽象层次实现相同的计数器合约：**Solidity、Solidity & Yul、纯 Yul 和 Huff**
- 应用**差分模糊测试**确保各实现行为一致性

技能

- **编程语言**：Golang、Java、Solidity
- **Web2**：Kafka、Redis、MySQL
- **Web3**：Solidity、Foundry、EVM(opcodes)
  - 熟悉常见攻击向量、Gas 优化、常见 ERC 协议、Uniswap、Compound 等
- **个人特长**：学习能力强、解决问题能力、沟通能力