个人信息

- 姓名：王康；性别：男；生日：1989/12/24；婚否：已婚；
- 电话：18610842638； Email：focusj.x@gmail.com；
- 毕业院校：河北大学工商学院（本科）；毕业时间：2012/7/01；专业：信息管理与信息系统；
- 博客：http://www.jianshu.com/users/13542edebea3/

工作经历

1. 2017年11月~今：北京摩拜信息技术有限公司

工作职责：

- 负责从零搭建公司业务监控系统，包括：系统架构设计，子系统开发及项目进度把控。
- 推进监控系统在公司业务部门的对接和使用。

2. 2016年03月~2017年11月：北京第三石信息科技有限公司

工作职责：

- 后端核心开发，主要负责后端系统的设计和开发。
- 系统重构和微服务架构改造。

3. 2014年10月~2015年11月： ThoughtWorks

工作职责：

- 业务系统全栈开发

4. 2012年04月~2014年09月：北京尤尼信息科技有限公司

工作职责：

- 负责公司后端系统的开发和维护。
- 兼职项目经理，负责和产品经理对接和梳理需求，制定迭代计划。

技术总结

- 熟练使用Java，Scala，Golang，了解面向对象和函数式编程思想.
- 熟悉多线程编程和异步编程，了解多种并发模型。了解Java多线程工具库，Akka，Netty，Vert.x框架。
- 了解微服务架构，ServiceMesh。了解SpringCloud和Istio框架的使用。
- 了解Redis，RabbitMQ，Kafka，MySQL，MongoDB的使用。
- 了解Docker和Kubernetes的使用，有一定的DevOps经验。了解敏捷项目开发，了解软件的CI/CD流程。

项目经验

摩拜监控系统

项目描述

从零开始搭建摩拜内部监控系统，目前所有的核心服务都已接入监控服务。每天监控系统处理7亿+指标信息(100G)，发出有效报警上百条。

项目职责

1. 从零搭建公司业务监控系统，协助所有核心业务系统均已接入监控服务，现在每天处理7亿+指标量。
2. 使用Golang开发基于内存的实时报警策略计算模块，报警策略表达式可以支持配置阈值和同环比，同时还提供了降噪功能，防止瞬间抖动造成误报。实时性方面：一个异常指标从应用端上报到报警发出整个链路耗时可控制在1min之内。
3. 基于influx-proxy的做了InfluxDB双区多实例高可用方案。在原有开源项目上开发了动态更新数据节点配置的功能。
4. 优化数据消费服务指标判重逻辑。原有的实现方式是：直接查询MySQL数据库，程序使用线程池提升处理能力。但由于数据量太大导致了线程池队列积压，引发了OOM。通过引入布隆过滤器重新优化了这部分实现，不仅解决了原有问题还加快了系统处理速度。
5. 基于Kubernetes和Istio的ServiceMesh改造，并且基于Kubernetes和Istio API实现了灰度发布工具。
6. 使用Golang和Mux开发了内部短域名服务，上线半月的时间已经生成了7000多万个code。

技术栈：SpringCloud，Java8，Golang，gRPC，Istio，Kubernetes，InfluxDB，ElasticSearch，Grafana，Telegraf。

订单系统

项目描述

北京第三石在美国市场做C2C业务，主要是线下二手物品交易。为了让用户交易更安全方便，公司要做担保交易和物流。

项目职责:

1. 引入Akka框架实现异步编程（将业务流程中的非关键流程异步话，比如发送push，订单扣费邮件等都可异步化），缩短服务的响应时间：订单接口从2~3s缩短到1s内。
2. 引入Spring StateMachine重构订单状态流转逻辑，增强了程序的可维护性。并通过其提供的回调机制实现订单历史状态持久化。
3. 基于Redlock算法实现了Java版本的Redis分布式锁。
4. 按照REST风格设计API，并通过swagger提供在线文档，降低前后端对接的难度。

技术栈: Java 8，Spring Boot，Spring StateMachine，JOOQ，Akka，Flyway，RabbitMQ，Redis，MySQL

后端服务拆分

项目描述

北京第三石是成立了三年多的创业公司，由于缺乏对后端服务重构和优化，所有的业务都耦合在一个系统里，导致系统臃肿复杂。开发维护成本越来越高。而且Python + Django也导致了严重的性能问题。为了解决这些问题，开始推进后端服务的拆分，以及技术栈转型.

项目职责

1. 系统边界划分以及微服务拆分规划。通过对现在业务的盘点和梳理，一共抽象了5个基础服务：用户，商品，订单，物流，聊天。实施方案如下：将现有系统中和基础服务相关的数据操作代码提取出来，包装为服务；接下来将剩余代码，按照不同的业务归属拆分为不同服务。
2. 技术调研和技术选型。用户、商品这些基础服务对性能要求比较高，综合对比了SpringBoot和Vertx，最终选择Vert.x作为基础服务框架。
3. 结合Redis缓存，最终用户服务完成后性能测试（4G/4Core）：读7000+QPS，写3000+QPS。

技术栈: Java8，Vert.x，JOOQ，MySQL，Redis，Docker，etcd。

名称：业务系统

项目描述

该项目是ThoughtWorks为某美国公司开发的业务处理系统。该公司主要负责为第三方公司提供财务审计，国际报税等业务。每项业务都由单独的系统处理。为了方便第三方客户查看及使用，需要作一个聚合的网站，把现有的业务聚合到一个网站内操作.

项目经历:

- 引入Pact契约测试框架，提前暴露由接口协议改变而引发的微服务集成失败。
- 引入React封装前端通用组件，减少团队的重复性工作，保证网站体验的一致性。
- 丰富的敏捷开发实战体验：TDD，CodeReview，CI，CD等。

技术栈：.Net Web API，ReactJS，SQL Server