

魏成龙/本科/JAVA/三年经验

联系方式

- 电话: 177 3995 9602
- Email: yjgbg1239461052@gmail.com
- 微信: yujgbg

个人信息

- 魏成龙/男/1995
- 本科/兰州理工大学/软件工程专业/18届毕业
- 期望职位: java/scala web后端开发程序员

工作经历

工作时间 2019/02-2019/09

公司名称 矽甲（上海）科技有限公司

工作描述

- 与客户进行沟通业务内容
- 把握整体项目开发进度
- 根据需求进行项目的开发，服务器应用版本的迭代和维护
- 参与数据库的设计

工作时间 2019/10-2020-09

公司名称 上海云联计算机系统有限公司

工作描述

- 对现有项目进行维护
- 根据需求进行项目的开发，web应用的迭代
- 数据库的设计

工作时间 2020/09-2021/08

公司名称 上海嗨普智能信息科技股份有限公司

工作描述

- 把握整体项目开发进度
- 根据需求进行项目的开发，服务器应用版本的迭代和部署
- 数据库的设计

项目经历

广告交易需求方平台 (DSP)

责任描述

- 负责代码的编写, 优化
- 数据库的设计
- 项目的构建与部署

技术架构简介

- 主要分为主控机, 投放端, 管理端 (还有一些其他端并未列举), 管理端实则为用户界面, 用户在管理端提交订单, 投放端接收来自媒体的请求, 做出投放响应, 或者不做出投放响应, 主控机根据订单和投放计划等数据, 周期性地向投放机提供时间片文件。

遇到的问题以及解决方案

- 技术难点1: 投放机的响应超时时间为10ms, 需要维持小于1%的超时率
- 解决方案1: 请求内不进行IO, 只是从内存中操作,降低了绝对响应时间; 使用Vert.X web, eventloop模型和NIO使得可以用少量线程hold大量请求, 降低了内核上下文切换次数, 再次降低了绝对响应时间; 投放机无状态化, 然后横向扩容至多台, 降低单机压力, 降低了高并发下的响应时间
- 技术难点2: 管理端业务高复杂度, 以及迭代过程中不断对业务的修改带来的混乱(复杂度主要来自于MyBatis mapper中大量的SQL, 数据类型转换, 还有数据校验)
- 解决方案2: 引入JPA, 使用关系注解标注数据之间的关系, 使得关系型数据在逻辑上呈现图状, 查询时使用Specification定位需要的数据对象, 使用EntityGraph表达所需对象的数据深度(放弃JPA中常用的函数名映射SQL语句思路), 以此来简化对数据的读取, 并且提高了程序的可组合性, (对Specification和EntityGraph的使用有一套[完整且简洁的封装](#), 使得上层调用者不用关注Specification和EntityGraph这两个晦涩的概念/关注点分离), 通过ActiveRecord模式简化对数据的操作 (以上, 在JPA上层提供了领域层抽象)。再引入了基于lambda表达式的声明式[对象校验工具](#), 降低了数据校验代码的代码量和复杂度, 提高了代码的可读性 (整体思想: 抽象业务中一些无关业务的逻辑, 做成公用组件, 用于业务代码的编写, 以此减少业务代码量, 从而提高了代码的可维护性)

项目的构建和部署

- 早期使用Jenkins进行构建和部署, 后期使用公司研发的devops平台

用到的组件

- AWS S3, Rabbit mq, Pulsar, Redis, MySQL

技术文章

- [孔乙己: 单例的五种写法](#)
- [Spring事务的前世今生](#)
- [剖析代理模式](#)
- [G1收集器简介](#)
- [代数数据类型的size和域模型\(翻译\)](#)
- [红黑树解析](#)
- [以函数式编程思维设计一个通用数据校验器](#)

技能清单

- JAVA基础：熟悉jdk8,11的新特性,对11之后的新特性也略有了解，包括但不限于Lambda，Optional，Stream，类型推导，了解VM内存模型，gc机制，8个垃圾收集器（G1以及之前的）各自的算法，对FP(函数式编程)和OOP(面向对象编程)有一定程度的见解
- Web框架：Spring MVC/Spring Boot/Vert.x/Finagle（Finagle是一个推特开源的基于netty的scala web框架）
- 操作系统相关：熟悉Linux中常见GNU工具的使用，有长期的Ubuntu Desktop和Manjaro使用经验
- 数据库相关：了解MySQL，会编写常见SQL语句，明白MyBatis的原理和使用，对Hibernate/ORM有比较深入的理解
- 版本管理、文档、容器：Git/Swagger2/Docker/Gitlab CI/Docker/Kubernetes，参与了公司devops平台的研发
- 编程语言：有入门较多种编程语言,包括但不限于ruby，scheme，clojure，haskell，prolog，偏好scala，偏好函数式编程，对多种编程范式各有所了解，对类型系统有较为深入的认识(这些很大程度上提高了我设计API的能力)
- 中间件：熟悉rabbit mq,aws s3(minio,swift),redis的使用
- 杂项：对分布式系统有一些学习和了解，了解redis集群的方案，对缓存的使用略有心得
- 最后：没提到的东西不一定是不会的，而且即使不会的也可以学习，我非常注意“提高自己的学习能力”这件事

个人评价

我是一个乐于学习，并善于学习，热爱技术，勇于尝试新事物，个性张扬，热爱交流，性格活泼，对代码中的坏味道过敏的洁癖重度患者。

另外，程序设计能力出众（非自我评价），抽象能力强（自我评价）。