

李忠泽

求职岗位: Golang后端工程师

年 龄: 25岁

婚姻状况: 未婚

民 族: 满族

政治面貌: 共青团员

电 话: 18941174662

性 别: 男

身高体重: 173cm/75kg

籍 贯: 辽宁省丹东市

工作年限: 3年经验

邮 箱: zonzeeeli0523@gmail.com



教育背景

2016-09 ~ 2020-06

大连理工大学

电子信息工程 (本科)

主修课程: 信号与系统、数字电路与系统、程序设计与基础、数据结构等等。

在校经历: 参加学生会文化部门, 担任学部和校级文化活动的安排与组织。

实习经验

2019-06 ~ 2019-08

东软信息集团 (大连)

嵌入式软件工程师

- 工作总结: 完成智能机器车的嵌入式软件开发;
- 学习成果: 学习如何用QT语言设计用户界面、使用蓝牙/ZigBee进行模块数据无线透传。

工作经历

2020-08 ~ 至今

珠海格力电器股份有限公司

后端开发工程师

- 负责物理网设备及格力+软件语音模块的后端业务功能开发;
- 负责语音后端的框架设计、业务设计、中间件模块开发;
- 负责语音功能模块与外部合作部分的接口对接。

项目经验

2022-08 ~ 2023-02

任务管理异步调度框架

主要负责人

项目内容: 对任务进行自动调度、可续做与重试的异步处理框架, 可灵活配置任务信息。比如用在管理离线语音的合成任务、设备控制的预设任务、语音入口输入文本数据分析认为的调度执行。

个人职责及技术难点:

- 负责框架的架构设计, 分成任务管理调度模块、任务治理监控模块和任务执行处理模块;
- 使用分布式锁避免并发竞争, 获取锁的worker节点执行任务;
- 对任务的优先级、重试、续做进行设计, 并对任务状态进行监控, 超时判定失败交由人工手动处理;
- 才用分表设计, 按照大小分表, 将冷热任务分开;
- 增大MySQL连接池对性能调优, 提高qps;
- 涉及到的技术栈: gin、mysql、redis、consul、zap等。

2021-01 ~ 2021-06

自定义名称预处理

主要负责人

项目内容: 对设备名称自定义后的输入话术文本进行替换或删除处理。比如设备“空调”自定义为“小白”, “打开小白”处理为“打开空调”, 房间“房间1”自定义为“xxx家”, “打开xxx家的空调”处理为“打开空调”, 并携带“xxx家”的信息。

个人职责及技术难点:

- 选择并发处理, 使用优先级返回处理的结果, 降低了程序执行耗时;
- 为解决语音识别出错问题, 增加拼音处理, 使用动态规划进行匹配, 优化时间复杂度;
- 涉及到的技术栈: grpc、consul、opentelemetry、jarger等。

2022-04 ~ 2022-07

语音合成api接口

主要负责人

项目内容: 对第三方语音合成api接口统一接入, 整合协议将结果返回。

个人职责及技术难点:

- 使用流式传输优化音频文件的传输效果;
- 使用心跳机制定时获取OSS服务器配置信息;
- 异步获取错误信息, 非核心功能崩溃后不影响合音频返回功能, 并捕获异常记录。
- 涉及到的技术栈: grpc、consul、websocket、zap、viper等。

2021-08 ~ 2022-02

多轮对话流程调度框架

主要负责人

项目内容: 对语音入口的话术的控制指令进行调度的框架设计, 并对多轮语音对话流程进行管理。比如“打开空调”, 家庭下有多个“空调”, 返回设备信息供用户选择控制, 并进行第二轮控制。

个人职责及技术难点:

- 设计协议, 选用session来判断一套会话流程是否结束, 不结束则获取信息缓存拼接;
- 设计整体流程, 在每一轮文本处理和语义处理后, 将信息进行比对后缓存, 控制意图或设备不同则进行清空处理, 否则获取拼接进行功能的控制调度。

个人项目

秒杀系统：实现简易秒杀系统的后端处理。

- 用mysql对商品信息进行存储，可以查询、添加，并用etcd进行同步，从etcd中加载商品信息，监听数据变化；
- 对库存扣减方案进行优化，不使用etcd，使用redis进行预扣方案，防止超卖，并轮询mysql将失效订单加载回redis，防止少卖；
- 用redis做队列，对创建订单的请求通过redis通知到逻辑处理模块；
- 安全校验，对用户信息进行签名校验，做白名单和黑名单限制；
- 流量限制，对入口及用户ip进行请求频率限制。
- 涉及到的技术栈：gin、mysql、etcd、redis等。
- <https://github.com/Zonzeeli/SecKill>
- <https://github.com/Zonzeeli/SpikeSystem>

短链接：实现短链的转链、存链、跳转等功能。

- 使用mysql存储短链和长链的映射关系，并加入缓存设计优化查询；
- 使用布隆过滤器防止缓存穿透，并使用singleflight提高并发性能；
- 基于MySQL主键自增，采用replace替代insert，保证数据量只有一条，实现高可用的发号器，转换成62进制的序列，保证短链尽可能短；
- 考虑分布式部署，对MySQL进行主从方案设计。
- 涉及到的技术栈：go-zero、mysql、redis等。
- <https://github.com/Zonzeeli/shortener>

IM聊天后台：实现一个简易实时聊天的后台管理系统。

- 用mysql存储用户信息，并完成注册登陆验证；
- 使用redis做缓存和队列，实现双人在线聊天，流式传输文本；
- 用mongodb存储聊天记录信息，对双方发送和接受的信息进行存储；
- 消息存储包含已读/未读状态，接受消息时可返回历史聊天记录和未读信息，在mongodb中更新状态，刷新过期时间；
- 涉及到的技术栈：gin、websocket、zap、viper、mongodb、mysql、redis等。
- https://github.com/Zonzeeli/Chat_demo

gRPC Gateway 网关框架：实现一个http和grpc的网关框架，添加中间件功能及手写负载均衡。

- 鉴权认证，使用jwt做鉴权；
- 对proto文件的参数做validate校验；
- 用zap自定义日志系统；
- 用Prometheus监控客户端和服务端埋点统计；
- 对访问进行限流策略；
- 添加重试和异常恢复机制；
- 使用Jaeger+OpenTracing做链路追踪；
- 涉及到的技术栈：grpc、prometheus、opentracing、zap、jwt、validate等。
- <https://github.com/Zonzeeli/Balancer>
- <https://github.com/Zonzeeli/grpc-gateway-demo>

社区帖子后端：实现帖子处理的后端，并根据投票点赞量返回帖子。

- 使用mysql存储帖子及用户信息，并做增删改查操作；
- 用jwt做用户校验，且使用雪花算法生成userid保证全局唯一以便分布式冲突；
- 用Redis的有序集合存储帖子分数和帖子id的对应信息，并对投票进行时间和分数的设计，采用投一票加分制(86400/200=432)保证帖子的时间和分数关系，且对投票进行限制；
- 使用桶令牌做限流策略；
- 涉及到的技术栈：gin、mysql、redis、zap、viper等。
- https://github.com/Zonzeeli/Go_web_bluebell

个人技能

技术栈：熟悉使用 Go、MySQL、Redis、gRPC、gin框架、go-zero框架及中间件等，了解消息队列、Kubernetes、Docker等。

证书：大学英语6级证书。

自我评价

1. 工作积极认真，细心负责，会针对架构、业务考虑优化方案，并与团队良好的配合。
2. 热爱学习，经常主动组织培训，与团队成员分享学习知识与新工具，并进行代码评审，帮助团队成员有良好的代码规范。
3. 个人心态乐观，积极向上，期望能增长视野，开阔眼界，了解更多新奇的事物。