



罗 勇

应聘岗位: **Android 高级工程师**
Android 架构师

关于我

性别: 男
出生: 1989 年 08 月
现居: 深圳南山
户口: 江西吉安
经验: 10 年
电话: 18682256298
邮箱: jefferson@youdubo.com

求职意向

地区: 深圳
行业: 互联网金融、互联网、智能硬件
薪资: 面谈
现状: 目前离职

教育背景

时间: 2008/09-2012/07
学校: 北京吉利大学
专业: 计算机应用与科学
学历: 统招大专

自我评价

- 1: 热爱专研技术、爱运动。
- 2: 做事追求效率、追求完美。
- 3: 不畏困难, 也勇于接受挑战。

掌握技能

一: Java

- Java 基础开发知识。
- 熟练掌握 Java 序列化、泛型、反射机制、线程、并发。
- 熟练掌握 JVM 、DALVIK、 ART 虚拟机底层原理、JVM 内存管理、垃圾回收算法、MMU 内存管理单元工作机制。
- 熟练掌握编译时技术 OOP【面向切面】、JavaSSit【字节码操作】、ASM【字节码插桩】、动态代理。

二: Android

- 熟练掌握 Android 四大组件、网络、数据持久化、线程等基础知识。
- 深入理解 Handler 机制、事件机制、Binder 四层源码的工作原理。
- 深入理解 AMS、PMS、WMS、init、Zygote、system_manager、service_manager 等主要关键进程。
- 深入理解 Activity、Service 等的全启动流程以及工作原理、UI 的全绘制过程以及渲染机制、沙箱机制等。
- 熟练掌握 Android 性能优化【内存优化、弱网优化、图片加载、启动优化、UI 优化、卡顿优化、崩溃优化、安全优化、Apk 瘦身】、OOM 分析处理、ANR 底层原理以及解决方案、Crash 方案。

三: 第三方框架 && 数据结构 && 设计模式

- 深入阅读并理解 OKHttp、Retrofit、Glide、GreenDao、RxJava、RxAndroid、LeakCannary、ButterKnife、Dagger2、EventBus 等三方框架, 自己重新手写过其中一部分框架。
- 掌握 Hook 系统 API 代码、Jetpack、AndroidR 下的沙箱存储等多种较难的技术。
- 掌握排序【冒泡、归并、计数】、二叉树、链表、栈/队列、二分、哈希表、堆/优先队列、数组/双指针、动态规整、矩阵、LRU 等数据结构
- 工厂模式、单例模式、建造者模式、原型模式、适配器模式、观察者模式、代理模式、责任链模式、策略模式、备忘录模式等设计模式。

四: 优势项

- 掌握 Android NDK 的开发, 对 Framework 层中主要服务有比较深入的理解。
- 掌握音视频开发技术【MPEG-4/H.264 AVC 编解码标准、AAC 音频编解码标准、流媒体协议 HLS 和 RTMP】, 对 FFmpeg 的编解码、libavformat、libavcodec 等核心模块。可将 FFmpeg 移植到 Android 平台下。
- 掌握多种语言, Java、C、C++、Kotlin。

工作经历

深圳一点半科技有限公司

2018.12 ~ 至今 | 研发部 | Android 架构师 & 技术总监

工作内容:

- 打造、构建公司产品，后期迭代升级产品以便更符合市场需求。
- 组建公司的 IT 研发团队，指定开发计划和推荐项目的研发，带领新手。
- 搭建 Android 端 App 的架构【基于 JetPack 组件搭建 MVVM 架构、性能优化和 Crash 捕获框架、网络框架、线程池管理框架、进程通信轻量级封装框架、弱网缓存以及缓存框架、Hook 技术实现的换肤框架、AndroidR 下分区存储框架、Tinker 热修复框架、增量升级框架、多渠道打包框架】。
- 带领团队成员进行代码 review 以及定期的技术分享和培训。

好未来教育集团

2017.12 ~ 2018.12 | 励步英语事业部 | Android 组组长

工作内容:

- App 的性能优化【内存优化、大图加载优化、UI 卡顿优化、OOM & ANR & Crash 监控和处理、弱网优化】
- 手表端 App 动作识别算法的研发【动态时间规整算法 DTW、数据收集、采样、数据过滤、匹配特征】。
- 移植 FFmpeg 项目到 Android 端，打造多协议【HLS、RTMP】的框架嵌入到项目并对外输出 API 供第三方进行使用。
- 带领新成员、带领团队进行代码的 review 以及技术分享培训。

百度（中国）有限公司

2015.10 ~ 2017.12 | 小度创新事业部 | Android 高级工程师 (T5)

工作内容:

- 开发全语音场景的下的进程通信项目 lib。
- 开发小度音响项目，通过进程通信 lib 实现无界面化的全场景语音小度音响。
- 开发大屏电视的 Launcher 项目，通过语音实现全场景实现大屏电视的全场景语音控制，包括第三方应用的语音控制。

TCL 多媒体科技控股有限公司

2012.5 ~ 2015.10 | 研发部 | Android 工程师

工作内容:

- 七喜 TV 盒子中 Launcher 项目的开发。
- 音视频播放器的开发【基于 FFmpeg 进行移植】，编解码、流媒体协议。
- 车机中系统 App 的开发【导航 App、音乐播放器、收音机、设置等】。

项目经历

项目一：智能硬件水质监测浮漂 | 2018.12 ~ 至今 | 深圳一点半科技有限公司

使用技术： Rxjava2 + ViewPager/Fragment + Glide + Retrofit

职责描述：

- 整个项目架构的搭建，主要描述在技术描述中。
- 制定整个项目的开发计划，推荐项目的进度。
- 带领团队成员进行代码的 review 以及团队建设。

技术描述：

- **MVP 架构搭建**：客户端采用了面向接口编程思想将 View 层与 Model 层进行分离，业务代码和逻辑代码解耦，结合 LiftCycle 对 View 生命周期更好地管理
- **UI 框架**：页面采用了 ViewPager+Fragment 嵌套方式开发，进行完全 lazyLoad，严格根据 Fragment 生命周期和可见性来控制加载数据的时机，根据数据动态添加 View 的展示，提升性能。
- **图片显示**：基于 Glide 做了图片显示的优化，对列表中图片出现的错位进行了改良，封装了一套对所有 View 都能方便使用的图片加载框架。
- **网络模块**：基于 Okhttp/Retrofit/RxJava 三个框架做了二次封装，形成独立的 Module，为应用层提供了丰富的功能接口，例如多域名切换、日志/Header/证书配置等，可以进行单元测试，开发者只需要在使用的时候添加自己对应的接口和 Bean 即可。
- **播放器**：采用 FFmpeg 移植到 Android 端，对软硬解码进行封装，对 H.264 压缩方式提供多种上层的选择，对 AAC 音频格式提供多种选择方式，然后将项目打包成.so 包直接在 Java 层进行应用。

项目二：在线直播和点播教育 | 【2017.12 ~ 2018.12】 | 好未来教育集团

使用技术： Rxjava2 + Retrofit + Jet Pack + ViewPager\Fragment + LuBan+ Hook + Tinker + bsDiff

职责描述：

- 通过 JetPack 进行项目整体架构的搭建。
- 通过 rtmp 进行推流 将摄像头的 nv21 视频格式转换成 h264 然后通过 rtmpdump 进行推流。
- App 的性能优化【内存优化、大图加载优化、UI 卡顿优化、OOM & ANR & Crash 监控和处理、弱网优化】
- 利用动态规整算法对手表端的 App 进行动作的识别。
- 开发音视频播放器。

技术描述：

- **MVVM 架构**：客户端通过 Lifecycle 进行组件管理、通过 livedata 将业务数据进行隔离，通过 databinding 来反馈 UI，同时通过 Hilt 实现项目的高度解耦，搭建出一套 MVVM 的框架。
- **UI 框架**：页面采用了 ViewPager+Fragment 嵌套方式开发，进行完全 lazyLoad，严格根据 Fragment 生命周期和可见性来控制加载数据的时机，根据数据动态添加 View 的展示，提升性能。
- **图片显示**：通过 LuBan 开源框架对图片进行压缩，后期改为自己开发编译的 Jpeg 引擎生成.so 文件，通过哈夫曼算法在 Native 层对图片进行压缩，大大提高图片压缩效率和内存性能，尤其是 GIF 文件的播放。
- **网络模块**：基于 Okhttp/Retrofit/RxJava 三个框架做了二次封装，形成独立的 Module，为应用层提供了丰富的功能接口，例如多域名切换、日志/Header/证书配置等，可以进行单元测试，开发者只需要在使用的时候添加自己对应的接口和 Bean 即可。
- **播放器**：采用 FFmpeg 移植到 Android 端，对软硬解码进行封装，对 H.264 压缩方式提供多种上层的选择，对 AAC 音频格式提供多种选择方式，添加对多种流媒体协议的支持，然后将项目打包成.so 包直接在 Java 层进行应用。
- **性能优化**：内存优化通过 Jvmti+LeakCanary+systrace 工具封装一套框架，实现对内存泄漏和抖动进行监控，UI 卡顿通过 layout Inspector 分析层级，通过 systrace 监控运行，自动注入 Choreographer 完成帧率的监控，最后运行 Looper 机制完成快速定位。
- **升级和修复**：通过 Tinker 框架进行项目的热修复，编译 bsDiff 代码生成 bsDiff 和 bsDaptch 对项目进行增量升级。

项目三：小度音响 & 大屏电视 | 2015.10 ~ 2017.12 | 百度（中国）有限公司

使用技术： Recycle View + QUIC【Http3.0】 + OCR + VEAR + JPEG 引擎 + Jet Pack

职责描述：

- 负责最为核心的 lib 库 VEAR，此库包含了语音识别、进程通信、网络通信、线程通信和线程切换管理等功能。
- 开发小度音响项目，通过剥离 VEAR 库，将语音识别单独提出并对性能方面进行优化。
- 开发大屏电视的 Launcher 和 setting 项目，并且通过中间 lib 项目 VEAR 与各个系统的 App 进行大量的进程通信以及网络通信。
- 打造专属的网络框架，此框架基于 Http3.0 实现，底层对应 UDP 协议，大大提升通信的效率。

技术描述：

- **系统框架**：整体系统框架通过一个中间库项目，对所有的系统 App 进行 Binder 通信，同时该库实现语音识别功能，同时对第三方 App，采用截屏+OCR 技术识别图片结果，再与中间库的语音识别结果进行二次对比后达到语音识别功能，从而达到实现对整个设备全场景语音控制的效果。
- **网络架构**：网络架构采用最新的 Http3.0 协议，同时底层采用 UDP 封装一套专属的网络架构，传输数据采用 Protobuffer 机制，同时实现多头阻塞\多路复用、拥塞控制、流量控制的高效、快速、安全网络通信。
- **图片显示**：图片显示上采用编译 JPEG 引擎源码生成.so 文件库，其中压缩方案采用哈夫曼算法，大图加载采用区域加载方式。
- **进程通信**：整个中间库 VEAR 实现一套自己封装的轻量级 binder、控件化的 AIDL、控件化后的 ContentProvider 三种通信机制，以实现对于不同需求下面的进程通信。
- **启动优化**：采用有向图的方式实现一套 Android 系统启动优化机制，对整个系统的所有 Task 进行分层管理，以便最大程度上达到一个快速的启动。
- **应用监控**：通过拦截系统的 UnCaughtExceptionHandler 对整个系统的崩溃体验进行优化，添加重新 Loop 或者重启应用等多种友好的用户体验方式解决系统崩溃的问题，同时对系统的 Error 和 Exception 日志进行统计和上传。
- **线程管理**：通过 CAS 原理，实现一套公平锁机制下的 ReentrantLock，实现智能管理线程池和线程的通信和切换调度
- **播放器**：采用 FFmpeg 移植到 Android 端，对软硬解码进行封装，对 H.264 压缩方式提供多种上层的选择，对 AAC 音频格式提供多种选择方式，然后将项目打包成.so 包直接在 Java 层进行应用。

项目四：七喜 TV 盒子 & 车机系统 App | 【2012.5 ~ 2015.10】 | TCL 多媒体控股有限公司

使用技术： Rxjava2 + OKHttp + ViewPager/Fragment + EventBus + ButterKnife

职责描述：

- 开发电视盒子中的 Launcher、单独抽离音视频播放器形成独立的 lib 库。
- 开发车机系统中的多个系统 App，导航、音乐播放器、收音机等。
- 通过 rtmp 进行推流 将摄像头的 nv21 视频格式转换成 h264 然后通过 rtmpdump 进行推流。
- App 的性能优化【内存优化、大图加载优化、UI 卡顿优化、OOM & ANR & Crash 监控和处理、网络优化】
- 开发音视频播放器。

技术描述：

- **MVC 框架**：自定义一套 Activity 管理栈实现对控制的管理，通过 EventBus 对 View 进行更新。
- **UI 框架**：页面采用了 ViewPager+Fragment 嵌套方式开发，进行完全 lazyLoad，严格根据 Fragment 生命周期和可见性来控制加载数据的时机，根据数据动态添加 View 的展示，提升性能。
- **图片显示**：图片显示直接采用 Android.jar Api 提供的复用和比例、质量压缩两种方式。
- **网络模块**：基于 Okhttp 框架做二次封装，形成独立的 Module，为应用层提供了丰富的功能接口，例如多域名切换、日志/Header/证书配置等，可以进行单元测试，开发者只需要在使用的时候添加自己对应的接口和 Bean 即可。
- **播放器**：采用 FFmpeg 移植到 Android 端，对软硬解码进行封装，对 H.264 压缩方式提供多种上层的选择，对 AAC 音频格式提供多种选择方式，添加对多种流媒体协议的支持，然后将项目打包成.so 包直接在 Java 层进行应用。
- **性能优化**：做一些内存上的优化、UI 卡顿、和崩溃优化处理。