|  |  |  |
| --- | --- | --- |
|  |  | **赵世令的简历** |
| 个人网站：https://www.lenzhao.com  Github主页：https://github.com/zhaoshiling1017 | |
| 个人博客：http://weibo.com/lenzhao1017 | |
|  | |
| lenzhao@yahoo.com | |
| (+86) 186-1270-0346 | |
|  | |

|  |
| --- |
| **求职意向：软件工程师** |

|  |
| --- |
| 1. 精通Java、NodeJs、python、shell编程语言；精通mysql、postgres、mongdb等数据库；熟练Atlas、Mycat、Cobar等数据库中间件技术。 2. 精通redis、memcached等内存数据库；熟练Twemproxy中间件技术。 3. 精通Java Web后端开发、熟练NodeJs全栈开发。 4. 熟练JVM虚拟机、类加载机制、内存模型、NIO、网络等Java基础架构体系。 5. 熟练高性能事件驱动程序库libuv。 6. 熟练DNS解析原理；熟练Http、Https工作原理。 7. 熟练面向对象设计原则，掌握设计模式应用场景。 8. 熟练使用常用数据结构及算法。 9. 熟练Spring、SpringMVC、Hibernate、MyBatis、Play Framework等Java框架；熟悉Express、KOA、Vue、AngularJs等NodeJs框架；熟悉Web.py等python框架。 10. 熟练SpringBoot、SpringCloud等Java分布式开源框架。 11. 熟练Docker容器技术。 12. 熟练ActiveMQ、Kafka等分布式消息系统。 13. 熟练ELK(ElasticSearch+Logstach+Kibana)日志分析系统。 14. 熟悉Dubbo、Zookeeper等分布式服务框架。 15. 熟悉Lucene、Solr、ElasticSearch等实时分布式搜索和分析引擎。 16. 熟悉Nginx、Varnish、Squid等代理服务器技术。 17. 熟悉LVS、Haproxy等负载均衡技术。 18. 熟悉Actor高并发编程模型；以及其实现Akka框架。 19. 熟悉C、C++、lua、golang、groovy、scala等编程语言。 20. 熟练使用Linux操作系统命令；熟练使用git版本控制；熟练使用Maven构建工具。 21. 熟悉Hadoop、Spark等大数据架构。 22. 了解数据挖掘、机器学习、数字图像处理等相关技术。 23. 擅长技术架构的选型与搭建，技术攻关，服务端和客户端性能分析与优化。 24. 具有完善的产品设计思维与理念，参与产品设计与讨论。 25. 针对实际的项目，有处理大数据量与高并发的经验，其中涉及数据库的sql优化、分库、分表、读写分离，应用的负载均衡，前端静态资源的分离等。 26. 平常喜欢接触新鲜事物，特别是一些新技术；爱逛开源社区，喜欢结识技术能力较强的人，以此来畅谈和分享技术。 |

|  |
| --- |
| 4年多的软件开发经验，负责过大型项目，有丰富的处理应急问题的经验，有较强的学习和实践能力。  薪资期望：25K |

|  |
| --- |
|  |

|  |
| --- |
| **开发项目** |

|  |  |  |
| --- | --- | --- |
| 云修助手 | **技术负责人** | **2016.10-至今** |
| |  |  |  | | --- | --- | --- | | • | 在ubuntu环境下，使用IntelliJ IDEA完成系统开发工作，开发语言java。 | **中国北京** | | • | 该平台集成了设备管理、系统管理以及移动巡检手机APP端的管理等功能，是为大数据云服务时代中成长的企业打造的设备运维移动管理平台。该平台支持企业的任务管理、设备检查、工单执行、安防管理和施工管理，为企业安全快速的处理各种任务提供保障；系统能够灵活的配置用户和职务权限，方便不同层级的人员独立的进行工作，更安全，更高效。 | | | • | 我在此项目中负责需求调研、详细设计、系统架构搭建、主代码编写以及管理与指导开发工作；由于这是一款面向互联网的产品，所以在刚开始架构选型的时候，需要充分考虑架构的可扩展性，最后敲定选用微服务的架构体系，整个架构图如下：  介绍如下：  1. 系统请求分APP、企业网站和运营管理网站三个来源。  2. 采用Spring Cloud微服务框架。  3. 请求首先经过阿里云SLB负载均衡，分发到下层的zuul代理层。  4. Html、css、image等静态资源采用七牛云存储，七牛提供CDN加速。  5. 在zuul通过加载动态过滤机制、实现请求验证、动态路由和负载分发等功能，将请求分发到下层的各web应用。  6. Web应用注册到Eureka注册中心，在web层调用下层的Service。  7. Service同样注册到Eureka，web应用和Service启动后将自己的信息注册到Eureka。  8. zuul和web调用下层服务时，通过Eureka服务发现取得服务列表并选取其中的服务进行调用，并在本地缓存，注册到Eureka的服务通过心跳维持和注册中心的连接，Eureka会剔除心跳消失或变慢的注册信息，保持服务的可用性。  9. Service通过MyCat中间件进行数据库的操作，MyCat来实现数据库集群的读写分离、分库分表、多租户等功能。  10. 缓存由redis集群挂一层twemproxy来实现代理分片，twemproxy会根据路由机制，转发给下层的redis集群。  11. 缓存存放数据库的查询数据，也用来解决kafka消费者重复消费的问题，另外也存放token和用户ID的对应关系。  12.系统监控通过Spring Boot actuator和admin实现。  13.系统配置文件托管在公司搭建的git服务器中，并通过消息总线和kafka队列进行热更新。  14. 每个Service采用SpringMVC+Mybatis的Web架构，对外提供RestFul风格API。  15. 由于各个模块采用Maven作为构建工具，所以很方便的通过插件的方式进行Docker构建、发布。  日志收集与分析系统架构图如下：  介绍如下：  1. 系统日志通过ELK(ElasticSearch+Logstach+Kibana)来收集整理展示，应用级别的日志通过Zipkin来收集。2. Zipkin可以通过请求嵌入traceid、spanid来实现跨服务的日志跟踪。  3. Logstach和Zipkin收集日志后通过kafka消息队列到ElasticSearch进行存储归类检索，Zipkin收集的应用级别的日志通过Zipkin WenUI进行展示，Logstach收集的系统日志通过Kibana进行展示。  遇到的问题：   1. 在实际开发过程中，遇到了关于分布式事务的通用问题，即分布式事务的一致性的问题，通过借鉴CAP、BASE等分布式理论，刚开始借助Atomikos实现了二阶段提交协议，不过两阶段提交涉及到多个节点的网络通信，通信时间如果过长，事务的相对时间也就会过长，那么锁定资源的时间也就长了，在高并发的服务中,就会存在严重的性能瓶劲；最终采用了基于消息队列+定时补偿机制，在我们的大多数业务逻辑中，第二阶段失败的概率比较小，所以单独独立补偿任务表出来，可以更加清晰，能够比较明确的直到当前多少任务是失败的。 2. 在使用Kafka作为消息队列如何保证在高并发下消息不丢失的问题，我的解决方案是先保证不丢失，然后在应用中处理重复的问题。 3. 由于通过Twemproxy作为代理，解决了单个Redis实例承载能力的问题，但是Twemproxy本身也是单点，后期会通过Keepalived做高可用方案。 4. 由于阿里云SLB负载均衡无法映射单个ECS服务器中的多个端口(所以会需要很多台ECS服务器)，所以后期公司考虑自建机房，通过架设LVS+Keepalived自己实现负载均衡配置。 | | | • | 网址：http://www.ducetech.com；运营端网址：http://manager.ducetech.com。 | | | | |
| 北京地铁通号分公司生产计划进度管理系统 | **技术负责人** | **2016.06-2016.09** |
| |  |  |  | | --- | --- | --- | | • | 在ubuntu环境下，使用IntelliJ IDEA完成系统开发工作，开发语言java。 | **中国北京** | | • | 该系统集成了流程发起、进度查询、任务管理、通知、统计报表、系统管理等功能，极大方便了通号分公司下的各部门、各科室之间的协作，同时方便通号分公司领导查看监控流程进度，从而与考核挂钩。 | | | • | 需求调研、详细设计、系统架构搭建、主代码编写、管理与指导开发工作。 | | | • | 系统架构列表：  Spring MVC + [MyBatis](http://code.google.com/p/mybatis/) + Mysql  遇到的问题：   1. 如何统一捕获处理异常？自定义实现HandlerExceptionResolver接口，来全局统一拦截处理功能异常；其中注入contentNegotiationManager，判断根据不同请求类型构造对应的数据格式响应，如JSON或JSP页面；根据不同异常类型，做一定的错误消息友好转义处理，区分控制不同异常是否需要进行logger日志记录；logger记录时，生成全局唯一的错误编号并合理的传递到前端界面显示以便用户反馈，把相关请求数据基于MDC方式记录下来，以便问题排查。 2. 计划任务的支持；对于定时任务这部分，在框架中除了按照常规的Spring Asynchronous & Scheduling用法外，主要从以下层面进行封装扩展：框架主要基于Spring Scheduling与Quartz集成进行封装扩展，提供一个BaseQuartzJobBean，所有Quartz Job统一继承此基类，便于统一定义处理、异常处理、日志记录等；提供一个扩展SchedulerFactoryBean的任务加载机制，实现动态从数据库加载定时任务配置；提供一个扩展LoggingJobHistoryPlugin的任务历史记录插件，异步方式把执行记录写入指定数据库表，并提供UI查看界面。 3. 数据库连接池刚开始采用dbcp，由于需要查看、统计SQL请求监控信息，所以将连接池更换为阿里的druid，它提供了更强大的监控统计功能，并且扩展性更好。 4. 由于每个实体类定义的字段特别多，所以出现很多GETER/SETER方法，通过引入Lombok采用注解的形式，简化消除了一些必须但有显得臃肿的Java代码。 5. 由于系统是部署在Linux环境中，通过编写一些常用的Shell脚本有效的提高应用更新的效率和简化人工操作流程，尤其是在开发环境可以实现全自动更新部署。 | | | | |
| 北京地铁志愿者官方APP | **技术负责人** | **2015.11-2016.05** |

|  |  |  |
| --- | --- | --- |
| • | 在ubuntu环境下，使用eclipse完成系统开发工作，开发语言java。 | 中国北京 |

|  |  |
| --- | --- |
| • | 该系统集成了志愿者管理、培训、积分统计、消息发送等功能，使管理人员能够方便、高效地对北京地铁志愿者进行有序的管理和记录，极大的提高了有关人员的工作效率，为工作的开展和日志的记录提供了强有力的保障。 |

|  |  |
| --- | --- |
| • | 我在此项目中主要负责需求调研、详细设计、接口服务及后台系统的架构搭建、核心代码编写及技术指导工作；  系统架构如下：  APP接口采用SOAP WebService的形式对外提供服务，架构为Spring+Apache CXF+Hibernate。  后台系统采用Play Framework1.x开发。  数据库: Mysql。  缓存: Redis。  遇到的问题：   1. 刚开始APP接口服务与数据库交互部分采用存储过程的方式，项目上线之后，随着并发量和数据量的增加，数据库压力不断增大，所以后来去除全部存储过程，改用HQL实现。 2. 由于Play框架采用的是Hibernate作为ORM，用c3p0作为数据源，然后Play源码对c3p0的配置相当有限，所以后来通过调整Play的源码，增强了对c3p0的支持。 3. 项目开始上线应用和数据库在同一台阿里云ECS服务器上，随着时间推移，志愿者注册人数不断增加，服务器的负载不断增大，后来增加了一台ECS服务器，把数据库迁移出去，从而缓解了服务器的压力。 4. 项目上线后，数据库中数据量随着用户访问量不断增加，开始出现有些数据库查询缓慢的情况，整理与分析所有的数据库查询SQL语句，根据SQL执行计划建立索引，并且优化了数据库的配置参数，并对数据库进行了分区处理。 5. 随着访问量和数据量的不断增加，单个数据库的读写开始出现压力，又增加了一台ECS服务器作为Slave数据库，通过Qihoo360的Atlas数据库中间件实现了数据库的读写分离，Master与Slave数据库通过数据库自带的半同步复制机制进行数据同步。 6. 数据库读写分离会遇到数据不一致的问题，有时候新增了一条数据结果跳转到列表页没有显示，这个时候在应用中强制把写和读放在同一个事务中，从而都是访问的主库，避免了数据的不一致。 7. 把一些不经常变化的数据放在缓冲中，然后通过缓存失效期或者更新数据时触发缓存刷新等，可以明显改善系统的性能问题。 8. 由于数据库中志愿者积分表数据超过了500万级别，所以利用Atlas中间件根据创建时间对积分表做了简单的分表操作。 9. 由于生产环境服务器没有开启防火墙，受到黑客团队攻击，主要通过redis 6379端口，登录之后清理了服务器内存，后来开启了服务器防火墙，并架设Openvpn，后台工作人员通过登录VPN进行系统操作。 |

|  |  |
| --- | --- |
| • | 后台网址：http://manager.bjsubway.cc。 |

|  |  |  |
| --- | --- | --- |
| 清凉地App分享服务 | 技术负责人 | **2015.09-2015.10** |

|  |  |  |
| --- | --- | --- |
| • | 在ubuntu环境下，使用pycharm完成平台开发工作，开发语言python。 | 中国北京 |

|  |  |
| --- | --- |
| • | 清凉地App是慈音文化倾力打造的一款佛门修行记录App，在使用过程中需要把修行轨迹等相关信息分享出去，该系统就是为此开发的一款后台服务程序。 |
| • | 需求调研、详细设计、框架搭建、代码编写。  系统架构采用轻量级Python Web框架web.py，数据库Mysql。 |
| • | 网址：http://share.sharinglove.cn。 |

|  |  |  |
| --- | --- | --- |
| 清凉地App后台系统 | 技术负责人 | **2015.08-2015.10** |

|  |  |  |
| --- | --- | --- |
| • | 在ubuntu环境下，使用sublime完成平台开发工作，开发语言nodeJs。 | 中国北京 |

|  |  |
| --- | --- |
| • | 该系统主要对修行课程、注册用户、系统用户等相关信息的维护，亦包括多修行报表的统计，由于其中部分课程资源存储在七牛云上，所以还包括与七牛云存储的对接。 |

|  |  |
| --- | --- |
| • | 需求调研、详细设计、系统架构搭建、核心代码编写、管理与指导开发工作。  系统架构: Express Web框架+Grunt构建工具+Bower依赖管理工具，通过pm2部署。  数据库: Mysql。  遇到的问题：   1. 由于NodeJS采用异步编程模型，会出现很多嵌套函数，刚开始采用Q、eventproxy模块，后期用bluebird进行了重构。 2. 在高并发的情况下，长时间运行系统会出现内存泄露的问题，首先利用memwatch来检测和监控内存泄露，后来发现存在一些数组push操作大于pop操作，从而导致数据容量不断增加，当使用内存达到v8引擎限制(1.4G)时，就会出现内存泄露问题，逐一排查代码，对这些数组进行了优化改进。 |

|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | | 外呼营销系统 | 技术负责人 | **2015.01-2015.07** |  |  |  |  | | --- | --- | --- | | • | 在ubuntu环境下，使用eclipse完成平台开发工作，开发语言java(应用端)、nodeJs(CTI部分)、lua(脚本)，软交换freeswitch。 | 中国北京 |  |  |  | | --- | --- | | • | 该系统能够有效提高营销部分的工作效率、降低企业销售成本和方便管理人员对销售代表进行考核等功效。坐席只需签入硬电话的方式进行工作，该系统通过自动拨号的功能，自动识别出空号、错号、忙音和关机等情况，将有效的电话号码转接至坐席人员，从而发现更多的机会，节省了消耗，大大提高工作效率；同时改系统对所有接通的通话进行了录音。 |  |  |  | | --- | --- | | • | 需求调研、详细设计、系统架构搭建、部分核心代码编写、管理与指导开发工作。  后台系统架构: Play Framework1.x，CTI服务: NodeJS。  数据库: Postgres+ SQlite  遇到的问题：   1. 该系统通过Postgres的schema实现了多租户；由于每个企业用户每天的话单量比较大，有时高达50万条，所以对话单表每天做一个备份，以当天日期为后缀，在页面查询的时候，后台程序会根据查询日期自动切换话单表。 2. 为了减轻系统本身的压力，对系统进行了拆分，把定时任务这块提取出去，这样定时任务服务和系统通过消息队列进行通信。 | | • | 网址：http://101.254.188.181:81。 |  |  |  |  | | --- | --- | --- | | 电话会议系统 | 技术负责人 | **2014.09-2014.12** |  |  |  |  | | --- | --- | --- | | • | 在ubuntu环境下，使用sublime完成平台开发工作，开发语言nodeJs，软交换freeswitch。 | 中国北京 |  |  |  | | --- | --- | | • | 该系统主要解决公司内部电话会议问题，支持临时会议和预约会议两种形式；功能包含基本电话会议功能、权限验证、创建会议室、邀约会议、踢人、禁言、静音、会议录音、日志管理等功能。 |  |  |  | | --- | --- | | • | 需求调研、详细设计、系统架构搭建、部分核心代码编写、管理与指导开发工作。  系统框架: Express + Grunt + bower |  |  |  |  | | --- | --- | --- | | 迪信通电商管理系统 | 技术负责人 | **2014.03-2014.08** |  |  |  |  | | --- | --- | --- | | • | 在ubuntu环境下，使用eclipse完成平台开发工作，开发语言java。 | 中国北京 |  |  |  | | --- | --- | | • | 该系统主要对来自线上（淘宝、亚马逊、一号店等）客户订单线下的处理以及线下的采购、入库，商品的管理，模块包括订单管理、采购管理、调拨管理、借机管理、库房管理、售后管理、客户管理、基本信息管理及系统管理。 |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | • | | 需求调研、详细设计、系统架构搭建、部分核心代码编写（订单、基本信息、系统管理等模块）、管理与指导开发工作。 | | | | 126buy配购网 | | 软件工程师 | **2013.05-2014.02** | |  |  |  |  | | --- | --- | --- | | • | 在windows环境下，使用eclipse完成平台开发工作，开发语言java。 | 中国北京 |  |  |  | | --- | --- | | • | 该网站是一款类似京东的专卖汽车配件的购物商城，网站上线后，由于推广力度不够，现网站已经关闭。 |  |  |  |  | | --- | --- | --- | | • | 需求调研、详细设计、部分代码编写。 |  |  |  |  |  | | --- | --- | --- | | ERP进销存管理系统 | 软件工程师 | **2012.09-2013.04** |  |  |  |  | | --- | --- | --- | | • | 在windows环境下，使用eclipse完成平台开发工作，开发语言java。 | 中国北京 |  |  |  | | --- | --- | | • | 该系统主要对汽车配件的入库、出库、采购、销售等管理，亦包括销售客户的维护，及报表的查询与导出。 |  |  |  | | --- | --- | | • | 需求调研、部分模块代码编写。 | |

|  |
| --- |
| **工作经历** |

|  |  |  |
| --- | --- | --- |
| 北京卓成世纪科技发展有限公司 | 技术经理 | **2015.08-至今** |

|  |  |
| --- | --- |
| • | 负责产品(项目)架构选型、搭建、开发工作 |
| • | 针对产品(项目)中遇到的技术难点问题，进行技术攻关，组织制定技术解决方案 |

|  |  |
| --- | --- |
| • | 负责技术部门新知识点的培训工作 |

|  |  |  |
| --- | --- | --- |
| 中软融合(北京)软件技术有限公司 | 技术经理 | **2012.09-2015.07** |

|  |  |  |
| --- | --- | --- |
| • | 负责公司产品的调研、研发与维护 | 中国北京 |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | • | 负责公司项目的研发与维护 | |

|  |
| --- |
| **教育背景** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 德州学院 | | 计算机科学与技术 | **学士学位** | **2008.09-2012.07** |
| **开源项目和技术文章** | | | | |
| 开源项目：  [Voyage](https://github.com/zhaoshiling1017/voyage)(https://github.com/zhaoshiling1017/voyage): 采用Java实现的基于netty轻量的高性能分布式RPC服务框架。实现了RPC的基本功能，开发者也可以自定义扩展，简单，易用，高效。 | | | | |

|  |
| --- |
|  |