**自我评价**

**5年**Java开发经验，其中**3年**以上**大型系统架构设计经验**，以及**3年**以上的**团队管理经验**，具有**分布式、高并发、高可用、大数据量的系统架构设计以及研发经验**，目前正负责**注册用户1.5亿，日活300万，日访问量1亿+**的手机生活服务平台的架构设计与研发。同时拥有扎实的技术功底，**对dubbo、spring cloud、spring、mybatis等开源框架均深度阅读过源码**。

**个人技能**

1、精通Java、设计模式、网络编程（Netty）、并发编程、JVM内存管理及调优

2、精通spring、springMVC、Mybatis，阅读过相关源码并根据需要扩展

3、精通dubbo、spring cloud（Eureka、Ribbon、Feign、Hystrix、zuul），阅读过相关源码

4、精通分布式事务，阅读过2PC、TCC相关组件的源码，设计可靠消息最终一致性方案、最大努力通知方案、saga

5、精通Mysql，具有sql优化、Mycat分库分表、索引优化、性能调优、数据库灾备等丰富的实战经验

6、精通Redis，具有集群搭建（Twemproxy、Codis、Redis Cluster），冷热备份，性能调优、数据迁移等实战经验

7、熟练使用MongoDB，具有集群搭建（副本集、分片集群），冷热备份、数据迁移、安全管理等实战经验

8、精通ElasticSearch，熟练使用Solr、Logstash、Beats及Kibana，搭建ELK日志收集

9、消息中间件：掌握activeMQ、rocketMQ、rabbitMQ、kafka原理及集群布署

10、负载均衡：熟练使用Nginx（Tengine、Openresty）、Haproxy、LVS、KeepAlived、zookeeper等负载均衡组件

11、自动化布署：Git、Jenkins、Gradle

12、容器化布署：docker，具有搭建swarm、mesos、marathon、kubernetes集群并运维经验

13、自动化运维：熟练使用Saltstack，数据监控（zabbix）

14、其他：了解并使用过Hadoop、Hive、Strom等大数据相关技术，使用过其他技术如lua、Go、Python、规则引擎（drools）等

15、建模工具：PowerDesigner、Rose、Visio、等UML建模工具

16、遵循华为测试规范，功能测试（单元测试、冒烟测试、集成测试、QA测试）性能测试（Jmeter、LoadRunner）、自动化测试（selenium、QTP）

17、遵循华为安全体系，代码安全、加密安全、设计安全

18、遵循华为完善的层级文档规范

**项目经验**

**（3）分布式系统架构**

1、分布式系统基于dubbo划分为个人中心、交易中心、充话费/流量服务、水电煤服务、电影票、海淘等十几个模块，dubbo配置关闭启动时检查、服务分组、多版本、延迟暴露等。基于dubbo如何做服务治理、服务降级以及重试

2、配置中心、广告系统、二级页采用spring cloud（Eureka、Ribbon、Feign、Hystrix、zuul）微服务框架实现。

3、基于分布式数据库MongoDB解决消息中心、广告中心的大批量终端Id的信息更新

4、基于分布式搜索系统ElasticSearch+Kafka满足站内搜索和ELK日志系统

5、分布式事务主要应用于交易中心的支付功能采用TCC事务，保障支付，风控，优惠券处理的一致性；支付网关回调采用可靠消息最终一致性设计，保证扣款，积分，抽奖的最终一致性；支付反馈消息采用最大努力通知型设计

6、分布式session使用Tomcat-redis-session-manager实现共享

7、采用zookeeper的分布式锁解决消息中心多模块竞争请求HW密钥的问题

**（4）海量数据架构**

1、个人中心、交易中心、充话费/流量服务采用Mycat对Mysql分库分表，分散存储数据

2、消息中心、广告中心采用MongoDB副本集，利用内嵌文档的Bson结构和过期索引应对大批量数据的增删

3、redis采用redis cluster做集群保障动态扩容，个人中心、交易中心使用bitmap、HyperLogLog类型进行大数据量统计

4、日志系统采用ElasticSearch的Rollover Index不断滚动切换增量数据，定时删除过期数据

5、监控系统监控Rabbitmq的Dead-Letter-Exchange应对消息积压

**（5）高并发系统架构**

1、SLB+Tengine分散压力，优化并发连接数

2、Tomcat采用非阻塞协议Http11Nio2Protocol，启用压缩，优化并发连接数，JVM优化

3、个人中心、交易中心、充话费/流量采用Mysql+Mycat+SLB进行分库分表，读写分离

4、充话费/流量采用Redis+Ehcache多级缓存保存热点数据

5、交易中心采用Disrupter进行多分支并行校验

6、个人中心、交易中心采用Rabbitmq对低耦合业务进行异步处理

7、充话费/流量采用strom实时统计热点数据，在redis保存权重，提高成功率

8、对热点数据预处理

**（6）高可用架构**

1、各业务模块多点布署保证高可用

2、充话费/流量、配置中心等采用Hystrix对异常的充值渠道进行熔断降级

3、用户信息、渠道信息等采用Redis+Ehcache多级缓存

4、MHA+Mysql+Mycat+SLB，MongoDB副本集，Redis Cluster集群保证存储高可用

5、Rabbitmq镜像集群保证消息队列高可用

6、ElasticSearch+Zookeeper集群保证搜索引擎高可用

7、Zabbix自动化检测实时报警

**（7）高性能架构**

解决手机充话费/充流量延迟问题，提高缴费成功率

1、采用享元模式缓存各渠道充值商品，采用状态模式构建充值记录，采用原型模式构建订单VO转换

2、采用strom实时统计渠道良萎率，在redis保存各渠道权重，提供充值成功率

3、采用hystrix对问题渠道进行限流，隔离，降级

4、对充值记录进行Mycat分库分表，按时间线分片

5、新增补充工程，轮询各类失败记录进行补充

解决交易中心安全性，卡单，请求缓慢，代码重构

1、采用Disrupter进行多分支并行校验

2、不同类型订单处理采用策略模式，支付状态采用状态模式，参数对象转换采用原型模式

3、支付功能采用TCC事务，网关回调采用可靠消息最终一致性设计，发消息采用最大努力通知型设计

4、交易表采用分库分表

解决个人中心用户联合登录阻塞问题，数据迁移，分库分表，代码重构

1、渠道校验采用策略模式

2、更换手机号，账号解绑定采用责任链模式

3、临时账号采用延迟队列

4、将热点数据迁移到redis缓冲

5、项目优化拆分，通过rabbitmq异步通讯

6、采用mycat对联合登录表和用户表进行分库分表，数据迁移，mycat高可用布署，预生产压测（Jmeter+AB）

解决消息中心数据膨胀，查询缓慢问题

1、采用mongodb层级格式重新设计表结构

2、利用mongodb过期索引动态批量删除

3、采用redis的bitmap类型记录大批量分发数据

4、采用hystrix对超时更新进行服务降级

对广告系统、common server、二级页系统引入sring cloud，docker mesos、marathon、推荐引擎等

协助搭建自动化测试环境、搭建ELK日志分析环境等