

成亮_技术经理_简历

热爱写代码的技术经理

联系方式

- 手机/微信 : (+86) 186-7297-8279
- Email : stevenchengmask@gmail.com

个人信息

download doc

- 成亮
- 计算机专业 / 5年经验
- 博客 : <https://amao12580.github.io/>
- GitHub : <https://github.com/amao12580/>

- 期望职位：技术经理 / 架构师
 - 期望薪资：税前25~35k / 月
 - 期望城市：深圳
 - 预计到岗：Offer后一周
-

工作经历

深圳市铺铺旺电子商务有限公司 (2016年3月 - 至今)

小旺项目

- [view](#) [Android](#)
 - [view](#) [iOS](#)
-

担任系统架构师，主导O2O平台基础架构的维护。按照业务特性，对系统进行拆分，以10多个微服务的形式进行重组，结合Docker容器技术，使系统容量调控非常灵活，扩容1倍仅需30分钟，采用多级缓存设计，从容化解流量洪峰。促进运维团队参与开发，量化平台业务指标，提倡面向运维的系统设计和技术选型，实施ELK日志分析方案，监控微服务和中间件的运行时状态，达到运维可视化，平台的可靠性不再是短板。可视化的监控，还使生产环境的问题更容易解决，迅速完成故障定位和流量隔离，简单的问题仅需运维替换集群节点，5分钟内即可完成，复杂的问题在隔离后，移植到线下环境调试修复，团队因此跳出加班写bug的困境。保持聚焦前沿技术，建立团队技术分享周会，在分享中一起成长，相互促进，共同沉淀，凝聚力也随之提高。

参与业务需求评审，结合团队实际资源，对需求进行优先级排期。在完成需求细化后，进行开发任务分解，甄别任务难易程度，分配合适的人完成。版本发布后，带领高级开发人员，进行Code review，重构不合规范和折衷的代码。我们相信持续交付过程中，适量的编写Quick & Dirty代码并无不妥，但必需有记录

(TO-DO) ，快速完成版本迭代，快速交付团队价值。平台的持续发展，一定会有隐患和性能问题，综合每周技术会议的讨论，定期安排重构，对外的目标也是更快和更好地交付。每完成一次重构，团队战斗力也整体提高，架构也完成一次进化。为了不影响持续的业务需求开发，重构在少数成员间进行，在SVN Branch 开发，待评审和测试完成后Merge到Trunk。对开发规范和代码重构的重视，使得我们的代码质量可控。

引入Docker容器技术，还使团队协作成本更低。结合docker-compose编排工具，生产、测试、开发等运行时环境，变得统一起来，程序真正的达到Build once, Run anywhere！开发与运维的协作，只需要交换Dockerfile文件（仅数KB）或容器Export文件，开箱即可用，摆脱在不同Linux环境的不一致。新加入的成员也可以在极短时间搭建环境，迅速融入团队，更快的为团队贡献价值！配合Jenkins持续集成工具，对新提交代码自动化进行：构建、测试（集成和性能）、部署，各阶段出错均有Report，整个代码生命周期变得可视化，团队的工作效率从未有这样高！

开源项目

维护

- RSSReader：个人RSS源管理和新文章推送，分为PC端和Android端。
- JOOQ-With-Spring：提供Spring与JOOQ的深入集成解决方案，并梳理了良好的分层架构。

贡献

Elastic-Job

Description：Elastic-Job是当当网后台的基础架构设施，是dd-job作业模块中分离出来的分布式弹性作业框架。去掉了和dd-job中的监控和ddframe接入规范部分。该项目基于成熟的开源产品Quartz和Zookeeper及其客户端Curator进行二次开发。

- [Improvement] 校对作业服务器与注册中心时间误差：对基础组件进行改良，已被采纳。
 - [Bug] 复杂网络环境下IP地址获取不准确的问题：提出并修复第一个Bug，已被采纳。
-

技术文章

- Nginx专题研究总结：Nginx与Docker容器系列 <02:在生产环境的实践>
 - JOOQ专题研究总结：JOOQ 3.8.2 使用 教程：从入门到提高
 - 分布式理论总结
 - 当我们谈事务时，我们在谈什么？
-

演讲和讲义

- 5月公司内部分享：Nginx_从入门到提高.ppt
-

技能清单

以下均为我熟练使用的技能

- 编程语言：Java
- DI框架：Spring
- MVC框架：Spring-MVC/Blade
- ORM框架：MyBatis/JOOQ
- 数据库：MySQL/Redis/MongoDB
- 消息队列：ActiveMQ

- 搜索引擎：Solr
 - 版本管理：SVN/Git
 - 代码构建管理：Maven
 - 代码质量管理：FindBugs/Checkstyle/Sonar
 - 文档管理：MediaWiki/Markdown
 - 持续集成：Jenkins
 - 缺陷跟踪管理：JIRA
 - 单元测试：JUnit
 - 功能测试：Postman/Fiddler
 - 性能测试：Gatling/Jvisualvm
 - 容器服务：Docker/Docker Compose/Docker Swarm
 - 开发环境：Linux/IntelliJ IDEA/Eclipse/MyEclipse
 - 服务器：Nginx/Tomcat
 - 负载均衡：Keepalived/HAProxy
-

致谢

感谢您花时间阅读我的简历，期待能有机会和您共事。