

肖世良

(86) 137 5710 6997 | slxiao1988@163.com | 杭州市滨江区信诚路 567 号 | GitHub: [slxiao](#) | 公众号: [测试不将就](#)

工作/项目经历

诺基亚杭州研发中心(2015.08-至今), 先后担任工程师, 专家, 技术 leader, 敏捷教练 (4 年升 2 级)

持续集成与持续交付, 2018.04 ~ 至今

- 带领 6~10 人的持续集成和持续交付团队, 服务 300+人跨国研发部门, 参与公司第一个 5G 商用产品交付。运用数据驱动和事故驱动策略识别研发流程与技术瓶颈, 采用敏捷模式落地改进点, 持续提高研发质量与效率。
- 基于 Python+Elasticsearch+React JS 全栈技术, 开发了 CI/CD 监控平台。基于自研的 Jenkins 插件([pipeline-monitor-plugin](#), 已开源), 提升监控实时性, 降低平台维护成本。平台日访问量近 2000 次, 备受各方好评。

面向大规模自动化测试的工程算法, 2017.04 ~ 至今

- 针对大规模自动化测试精准性低的问题, 基于源文件依赖关系算法设计了能够依据源文件变化动态选择待执行用例子集的测试漏斗工具([test-funnel](#), 已开源), 平均减少 80%的 JavaScript 白盒测试用例执行次数。
- 针对大规模自动化测试运行效率低的问题, 基于贪心策略和动态规划策略, 设计了多种测试用例集合切割算法([partition](#), 已开源), 实现了运行时间最小化的并行自动化测试, 将测试执行效率提高 2-4 倍。
- 针对自动化测试实现与维护成本高的问题, 基于 TensorFlow 框架实现了智能化的软件缺陷预测工具 [prober](#) (已开源), 在公共数据集上准确率 80%+。结合面向 Java 语言的代码特征提取算法, 将工具应用于产品开发。通过周期性扫描代码并发送缺陷风险提示报告, 辅助开发团队进行代码重构与改进工作。

菜鸟模拟器及性能测试平台开发, 2016.01 ~ 2017.04

- 带领 3 人开发团队, 基于 Python 实现了若干以 protobuf、TCP、HTTP(s)等为核心的私有网络协议。采用敏捷开发模式, 运用 TDD, CI/CD, docker, DevOps 等理念或工具, 累计产生 3000+代码提交(其中本人提交 1000+次), 累计执行包含 260+个用例的测试集 6000+次, 累计发布 320+个版本, 实现了高质量与高效率软件开发。
- 在模拟器基础上, 基于 Python +React JS 全栈技术自研了性能测试平台。基于协程实现异步 IO, 基于消息队列实现主从模式与分布式执行, 持续改进并数倍提升性能测试平台的并发能力, 圆满承担了产品性能测试任务。

公众号与社区活动, 2018.04 ~ 至今

- 创建公众号《[测试不将就](#)》, 以“速度之上的质量”为愿景, 践行“质效合一”理念, 聚焦软件质量保障, 工程效率, CI/CD/DevOps, AI 软件工程, AI 测试等方向。累计输出 60 余篇近 20 万字原创技术文章, 拥有万名读者。
 - 中国 Python 大会杭州站演讲嘉宾(17), 连续三届公司技术嘉年华演讲嘉宾(17-19)。
-

教育/研究经历

- 浙江大学本科(06~10), 中国科学院大学[中科院上海微系统与信息技术研究所]硕博连读(10~15), 专业: 通信与信息系统, 研究方向: 无线网络和 IoT, 获工学博士学位。
 - 参与多项国家重大专项, 以第 1 作者身份在 IEEE/ACM 所属期刊发表 5 篇 SCI 论文, 以第 2 作者身份发表 6 篇 SCI 论文, 发明专利 6 项, 5 篇 1 作 SCI 论文他引 50+次, 为多个权威期刊会议审稿数十次(参见[学术主页](#))。
-

荣誉奖励

2019 Nokia HZ TC Recognition Award, 2017 Nokia HZ TC Trainer Award, 2017 Nokia LTE 编程大赛个人奖, 2014-2015 年博士研究生国家奖学金(全所通信专业 100+在读博士, 仅 2 人获奖, 奖金 ¥ 30000), 2014-2015 年中科院所长奖学金, 2007-2008 年浙江省高等数学竞赛一等奖, 2007-2008 年浙江大学研究与创新一等奖学金。

附技能清单: 熟练—Python, Linux, Jenkins, Docker, Gitlab/GitHub/Gerrit 熟悉—Kubernetes, ELK, React JS, MySQL, Groovy 了解—Java, Node JS, TensorFlow, Maven/Gradle