

赵龙燕

全栈工程师 | 9年经验 | 杭州

xhgua.top 微信: uuuyki captainzhaoly@gmail.com

[alyzhao](#) 19564175201

自我简介

资深全栈架构师，拥有 9 年互联网大厂核心业务研发经验。专注于构建 AI 驱动的高性能系统。深谙大模型工程化落地，精通基于 LangChain / LangGraph 的复杂多 Agent 状态机 workflow 构建，具备丰富的高级 RAG(混合检索/重排/GraphRAG) 系统调优经验，且熟悉 MCP、SSE 等大模型交互协议。技术栈横跨 Node.js、Python 与 React 极客生态。曾主导自动化测试执行引擎及交互式调试系统的设计，在自动化测试、IDE、数据可视化、音视频领域有深厚积累。坚持“测试先行”的研发理念，善于利用 AI 工具将复杂业务逻辑转化为可验证的规格说明，极大地提升了工程交付的确定性。具备跨职能团队协作与项目管理经验，享受解决难题的过程，致力于通过技术创新创造业务价值。是终身学习者，也是健身、音乐、摄影爱好者。

专业技能

AI Coding & AI Agent 开发

- 大模型微调与底层机制：**了解 Transformer 底层架构与模型量化原理 (FP16/GGUF)。具备一定的 LoRA 参数微调 (PEFT) 实践经验；了解基于 LlamaFactory 的模型微调流程、基于 vLLM 的高并发推理部署以及基于 EvalScope 的量化评测体系。
- 大模型应用架构：**精通 LangGraph 复杂状态机管理，具备设计健壮的多 Agent 协同系统的能力；熟悉 MCP 协议与大模型工具流 (Tool Calling) 交互机制。
- 高级 RAG 工程：**精通多链路检索优化。熟练运用 RRF 混合检索、Cross-Encoder 深度重排以及 GraphRAG 图谱语义关联检索；熟悉 DeepDoc 多模态文档解析。
- AI Coding 与研发工程化：**深谙 Spec-Driven Coding 范式，配合 Claude Code、Cursor、OpenCode 等工具实现高准确度的人机协作闭环；能够利用 AI 技术重构传统研发链路，实现从需求到测试用例的自动化。
- 工程化自动化：**能够利用 AI 技术重构传统研发链路，实现从设计稿到代码、从需求到测试用例的自动化闭环。

IDE 架构与研发工具链

- IDE 框架定制：深入理解 VSCode 插件生态与 OpenSumi 框架，具备从 0 到 1 构建工业级 IDE 产品的经验。
- 底层协议应用：精通 DAP (Debug Adapter Protocol) 协议，曾自研基于 Monaco Editor 的 JS 单步调试器，实现断点管理、变量监控及物理级线程暂停。
- 源码静态分析：熟练使用 Babel AST 解析与代码转换，具备开发代码转换器、静态分析工具及自动化插桩工具的编译器级实战经验。

高性能全栈架构

- 高并发架构与数据库：深入理解 SQLAlchemy / SQLModel 异步 ORM 原理；精通 MySQL 锁机制与并发控制；熟练设计高可用 Redis 缓存防御架构 (布隆过滤器、逻辑过期等) 解决缓存穿透与击穿难题。
- Node.js 生态：精通 Node.js 全栈开发，深入理解 Midway (FaaS)、Express、Egg 等后端框架原理；具备高性能 RPC 通信与大规模数据通信架构设计能力。
- Python 后端：熟练使用 FastAPI + uv，专注于构建高并发、可扩展、响应式的流式微服务架构。
- 容器化与部署：熟练使用 Docker 进行环境隔离与容器化部署，掌握 MinIO 对象存储，熟悉 Jenkins 自动化 CI/CD 流转。

垂直领域技术积累

- 低代码与图形化：拥有成熟的低代码平台架构设计经验，熟悉无限画布、Canvas 2D/WebGL 渲染优化及 WebAssembly 预览加速技术。
- 多端与音视频：精通 React 生态 (Hooks/Redux/Saga)，熟练掌握 Taro 多端开发；了解 WebRTC 实时通信、设备管理 (WebUSB) 及浏览器音视频 API 应用。
- 质量保障体系：熟练掌握 Puppeteer/Playwright 进行 Web 自动化测试，具备设计高仿真、抗重定向的自动化执行引擎能力。

项目经历

一、MAX-IDE 自动化测试 Web/H5 执行引擎 (Todhorn Web Executor)

技术栈：Node.js + Playwright + WebSocket + Docker + OSS

项目简介：

为解决跨端跨工程自动化测试痛点，主导开发了基于 Node.js 与 Playwright 的高吞吐、高仿真 Web/H5 自动化测试引擎。作为低代码解决方案质量保障体系的关键基础设施，具备高仿真真实交互能力、独立沙箱隔离运行环境及对接宏观调度的核心能力，为长链路自动化业务提供了关键的 Web 端能力支撑。

核心职责与难点攻克：

- 设计 Node.js 宿主进程与 Browser 页面的 IPC 通信桥接：下行通过 page.addInitScript 埋入执行引擎，上行通过 page.exposeFunction 打破沙箱限制，实现低延迟双向通信
- 实现自适应延迟与抗重定向的探活策略：建立 waitForExecutorAndStart 多级熔断探测算法，解决页面跳转、异步加载及多重 JS Redirect 导致上下文丢失的难题，执行成功率大幅提升
- 抽象策略模式组件（Operation / Assert / ControlFlow / Script），支持 If/Else、For 循环、自定义断言与沙箱代码段，并实现变量快照存取与跨端变量穿透
- 设计 1-N 复用单例并发拓扑：单一 Chromium 常驻进程 + BrowserContext Pool，毫秒级创建纯净隐身独立执行上下文（Cookie/Storage 100% 硬隔离），单台 4核8G 云主机承载 6+ 并发链路
- 通过 WebSocket Agent 长连接实现横向内网穿透，调度中心可随时掌控节点活性并实时获取串联状态
- 精细化资源保护：MAX_CONCURRENCY 阈值控制 + 僵尸 Context 垃圾回收 + 带防抖指数退避的 WebSocket 重连 + OSS 异步非阻塞上报，避免单次缺陷导致宿主机雪崩

二、MAX-IDE 交互式断点调试（基于 DAP 的 JS 单步调试器）

技术栈：TypeScript + Monaco Editor + Babel AST + Web Worker + SharedArrayBuffer

项目简介：

为解决模型 IDE 中日志调试反馈慢、可视化差、影响性能等痛点，主导设计了基于 VSCode DAP（Debug Adapter Protocol）的交互式断点调试系统。支持断点调试、单步执行（Step Over/Into/Out）、变量查看、调用栈跟踪等功能。

核心职责与难点攻克：

- 三层架构解耦设计（Debug UI / Debug Adapter / Debugger），各模块通过 RPC（JSON-RPC over postMessage）通信，UI 与 debugger 完全解耦可插拔，借鉴 VSCode DAP 协议标准

- 突破核心技术难点：使用 `Atoms.wait + SharedArrayBuffer` 实现 Worker 线程级别的物理暂停与恢复，而非 `async/await` 方案的全量异步传染，完美保留原始调用栈
- 通过 Babel AST 插桩在每个表达式注入 `checkpoint`，配合 `@babel/traverse` 生成作用域链；使用 Proxy 拦截变量访问，巧妙规避 TDZ（暂时性死区）问题
- 使用 `WeakMap` 实现循环引用对象的图结构记忆（Graph Memoization），保留真实环状拓扑，利用结构化克隆算法（Structured Clone）安全跨线程传递，解决 OOM 与 `DataCloneError`
- JIT 即时按需编译：Proxy 代理触发插桩编译，仅在调用对应文件函数前一刻才 AST 拉取处理，毫秒级首屏启动；配合缓存复用大幅提升大型项目调试体验
- UI 层 Lazy Node Expansion：变量树首次仅渲染顶层节点，按需展开避免数十万 DOM 渲染卡顿，还原 Chrome DevTools 级无限递归展开体验

三、MAX-IDE 自动化测试平台（低代码 IDE 集成测试系统）

技术栈：**React + useSyncExternalStore + Babel AST + Web Worker + Docker**

项目简介：

打造深度融入低代码 IDE 生态的自动化测试平台，支持沉浸式无痕录制、复杂场景逻辑编排（If/For）、跨多端协同、组件级精确匹配定位及模型交互，是具备工业级深度的底座解决方案。

核心职责与难点攻克：

- 基于 AOP 劫持渲染引擎 `useCreateElement`，对组件级 `nodeId` 进行事件拦截录制，摒弃脆弱的 DOM Selector，实现语义级零侵入录制，解决传统 E2E 测试因 DOM 微调导致的大面积失效问题
- 双向绑定与复杂手势捕获：对不抛出外部事件的双向绑定组件建立 `componentTwoWayBindMap` 探针；基于 BFS 广度优先搜索算法定位真实滚动 DOM 节点，精准还原列表滚动操作
- 自研 `Promise Defer + operateStack` 出入栈架构处理“打断与恢复”：当用户插入 `Assert/If` 配置表单时挂起状态机，配好后 `resolveAndPopOperation` 唤醒，实现多层次异步无缝缝合
- `Executor + Middleware` 管道模式：统一策略抽象 `Click/Swipe/Assert/If` 等执行器，支持执行前强制等待、完成延迟等切面批量处理

- 极限内存优化: LazyModule 按需加载 + instanceId 动态缓存生命周期管理 + removePageRootCache 主动释放泄漏节点, 根治大型低代码应用的 OOM 隐患

四、Magic Design 可视化低代码工具

技术栈: **React + Redux + Redux-Saga + Konva + Electron**

- 负责前端架构设计, 包括数据结构设计、技术选型、工程搭建及核心功能设计, 以 Node.js + VSCode 为底座, WebView 为交互入口
- 设计 path-map-tree 数据结构, 通过节点 id 与路径索引实现高性能树节点操作, 结合最优索引更新策略提升整体应用性能
- 设计并实现 RPC 通信模式 (Node ↔ Browser), 通过 Proxy 代理通信方, 支持 TypeScript 类型定义与自动提示, 接口注册简单语义化, 配合消息缓冲机制大幅提升可维护性
- 使用 WebAssembly + Canvas 实现所见即所得的预览层, 抽象合理数据结构实现绘制、平移、缩放等功能, 设计高性能事件模型支持拖拽、框选、边角拖拽等操作
- 实现 JS 沙箱计算 bindings 表达式, 结合沙箱隔离防止 XSS 攻击, 使用 Proxy 维护 bindings 与组件依赖关系, 实现用户操作实时更新
- 设计 Error Boundary 错误处理组件, 捕获运行时、语法、异步、React Render、Saga 等多类型错误并显示 Fallback UI, 写入日志支持数据恢复

五、Magic Code 代码IDE (AliOS 开发者桌面工具)

技术栈: **Electron + OpenSumi + Webpack + Gulp + Jenkins**

- 负责 IDE 功能设计与可行性调研, 在 Electron 中集成 OpenSumi 框架, 通过插件化架构主导主要功能开发, 并为 OpenSumi 贡献代码, 并制定团队工程化规范: Husky + Lint-Staged + ESLint + VSCode Save Format 代码提交校验, Gulp + Webpack + Jenkins 自动化发布流程
 - 设计并实现插件内 Node 与 Browser 的通信模式, 通过 Context 与自定义 Hooks 封装多个模块能力, 开发 magic-code-cli 脚手架支持 IDE 插件项目快速创建
 - 自研多线程分片下载器 (支持断点续传、错误重试), 发布为独立 npm 包供其他业务使用; 设计 SDK 开放接口支持多业务调用, 提升项目扩展性
 - 设计 IDE 增量更新功能, 配合 Gulp 构建增量包与 Jenkins 自动上传, 实现用户无感知最小化更新, 解决多平台更新与 Windows 文件占用替换难题
-

工作经历

2024.07 – 至今 商米科技 【前端工程师】

- IDE 调试业务负责人，主导 IDE 单步调试的架构设计和项目管理，推动功能顺利落地
- 负责应用 IDE 中的在线 ADB 功能，通过 WebUSB 实现设备管理和 APK 安装，结合 WebRTC 实现设备日志、网络请求与性能数据的实时查看，大幅提升真机调试效率
- 负责用户预览性能优化及 Web 运行时框架的维护，提升系统整体性能和用户体验

2019.07 – 2024.02 阿里巴巴 【前端工程师】

◆ 飞猪事业部 (2023.10 – 2024.02)

- 负责飞猪数据相关中后台的开发维护，负责飞猪数据资产平台的开发维护，负责飞猪报表生产平台的功能和架构设计以及项目管理
- 以全栈工程师身份主导飞猪报表生产平台的设计到上线全过程

◆ 斑马智行事业部 (2021.04 – 2023.09)

- 负责 Magic Design 前端架构设计，负责核心功能开发与技术难点的突破
- 负责 AliOS 开发者 IDE 工具的设计与开发，包括 SDK 分片下载、断点续传、IDE 增量更新、Magic Design 自定义编辑器等核心技术设计与开发
- 沉淀多个工具包：下载器、ESLint Config、脚手架及 IDE 工具方法库，支持其他业务复用
- 多次在组内做技术分享：OpenSumi 能力介绍、Redux 源码与原理、Redux 中间件与 Redux-Saga 原理

◆ 新华智云事业部 (2021.04 – 2023.09)

- 负责 Web 视频编辑器的开发与维护，包括海报视频卡片式编辑器等延伸业务探索与支持
- 负责公司微信小程序的开发管理与维护，沉淀小程序开发技术，编写小程序脚手架

2018.09 – 2019.07 中软国际 【前端工程师】

2017.09 – 2018.09 杭州数空科技有限公司 【前端工程师】

教育经历

2013.09 – 2017.06 嘉兴学院 【信息管理与信息系统专业】