

郝晓波

电话: 15714514997

邮箱: 15714514997@163.com



基本信息

具备3年Go语言后端开发经验，精通高并发、微服务架构设计与实现。热衷于构建高效、可扩展的网络应用，有良好的团队协作和项目管理能力。曾主导公司关键项目的大规模机器学习模型部署，有效提升了应用的预测能力和实时响应速度。在运维领域，积累了丰富的实践经验，熟练掌握Linux环境下的系统优化及各类中间件（如ELK、Consul、Nginx）的部署与调优，确保了服务的稳定运行和卓越性能。

教育背景

黑龙江大学- 本科 - 电子信息工程专业 (2017.09 - 2021.06)

专业技能

- 编程语言:** Go (熟练掌握), Python (基础), C(基础)
- 框架/库:** Gin、GORM、gRPC、go-zero
- 数据库:** MySQL、Redis、Elasticsearch
- 工具与技术:** Docker、Consul、jaeger、Kafka、traefik、Jenkins、Git
- 操作系统:** Linux(Ubuntu、CentOS)

工作经验

Golang后端开发&大模型运维 - 深圳昂楷科技

2023.08 - 至今

- 微服务架构设计与实现:** 精心规划并构建了微服务生态系统，利用微服务的特性实现模块化、独立部署和横向扩展，显著提升了系统的灵活性和可维护性。
- 高并发与高可用性解决方案:** 采用前沿的负载均衡技术与策略，结合微服务架构，打造了能够应对大规模用户访问的Web服务平台。
- 压力测试与容错机制:** 实施了全面的压力测试，模拟各种极端场景，以验证系统极限性能。
- 大模型部署与优化:** 主导了大模型的部署流程，针对模型加载和推理速度进行了优化，大幅提升了模型响应速度和预测精度。
- 系统健康监控与维护:** 建立了全方位的监控体系，实时监测系统性能指标和健康状况，通过主动预警和预防性维护，及时发现并解决潜在问题，保障了模型服务的稳定运行和用户满意度。

Golang后端开发 - 珠海格力电器

2021.07 - 2023.08

- 负责智能窗帘与智能开关项目的整体技术规划与实施,** 重点在于设计并部署智能语音控制解决方案，确保系统能够准确解析用户指令并控制相应设备。
- 搭建并优化gRPC服务端,** 实现与语音识别系统的无缝对接，提升了系统的响应效率和用户交互体验。
- 引入Consul服务注册与发现机制,** 强化服务的动态管理能力，显著提高系统的可用性和稳定性。
- 负责日志系统的构建与维护,** 利用Elasticsearch集中存储设备交互数据，为后续的数据分析和系统性能优化提供有力支持。
- 推动运维自动化,** 通过创建Dockerfile和Jenkinsfile，建立基于Git的持续集成与持续部署（CI/CD）流程，提高项目部署的效率与准确性。

- 管理并优化Nginx配置，实施服务在Nginx中的负载均衡与反向代理设置，保障服务的高效运行与良好的网络性能。

项目经历

大模型后端开发

- 模型加速与跨语言通信：采用PyTorch和VLLM框架，深度优化模型推理流程，显著缩短响应时间，提升模型处理能力。利用gRPC技术实现跨语言高效通信，确保了服务间低延迟交互与高吞吐量，为异构系统集成提供了坚实基础。
- 微服务生态建设与管理：实施服务注册与健康监测机制，借助Consul服务，构建了一个动态、健壮的微服务架构，有效管理和监控各服务实例的状态，确保了系统的高可用性和弹性。
- 高性能Web服务构建：选用Gin框架构建高并发、高可用的API网关，为前端应用提供了稳定、高效的数据接入点，支撑大规模用户访问。
- 日志监控与可视化：集成ELK栈，实现了日志的实时收集、分析与可视化展示，极大便利了故障排查与性能调优，提升了运维效率和系统可观测性。

智能窗帘&智能开关项目

- gRPC服务端设计与实现：构建了一个高性能、低延迟的gRPC服务端，专门用于接收来自客户端的远程控制请求，解析并理解请求的具体意图。
- 设备状态与mesh网络信息查询：根据请求中携带的设备标识，查询数据库中存储的设备mesh网络信息及其当前状态，确保对设备的准确识别与定位。
- 控制逻辑与决策制定：基于请求意图与设备的实时状态，智能判断并决定设备应执行的具体动作，如开启、关闭或调整参数。
- 设备控制命令下发与结果回传：将经过决策的控制命令连同必要的mesh信息一并发送至设备控制端，确保命令的准确执行，并接收执行结果，形成闭环控制。
- 服务注册与发现机制集成：紧密集成服务注册与发现机制，自动获取项目相关的配置信息，保证服务的高可用性和动态伸缩能力。
- 日志记录与Elasticsearch整合：实现了服务日志的标准化记录，并将所有关键操作日志实时写入Elasticsearch，便于后续的数据分析、监控及故障排查。

运维

- Dockerfile与Jenkinsfile设计与实施：精心编写Dockerfile，确保项目服务能够高效、一致地打包为容器镜像，同时创建Jenkinsfile以自动化构建和测试流程，实现从代码提交到生产环境部署的无缝衔接。
- CI/CD流水线构建与维护：利用Git作为版本控制系统，搭建并维护了高度自动化的CI/CD流水线，确保每次代码更新都能快速、可靠地通过测试并部署至目标环境，显著提高了开发效率和软件质量。
- Elasticsearch系统管理：负责Elasticsearch集群的日常运维，包括性能调优、索引管理、数据备份与恢复，以及监控与故障排除，确保搜索和数据分析服务的高可用性和响应速度。
- Nginx负载均衡与反向代理配置：优化Nginx配置，实施负载均衡策略，确保请求在多个服务器间均匀分配，提高应用的可扩展性和响应能力。同时，设置反向代理以增强安全性，并改善用户体验。
- Consul服务注册与发现：在微服务架构中，利用Consul进行服务注册和发现，实现了服务间的动态通信和故障切换，增强了系统的稳定性和灵活性。

智能开关自动测试项目

- 智能开关自动化测试平台设计与开发：运用Gin框架构建了一个高效、灵活的自动化测试平台，用于持续监控和评估智能开关的性能和稳定性。
- Grpc服务交互与状态验证：通过gRPC协议与智能开关建立低延迟、高性能的双向通信，精确调用其控制接口，同时捕获返回结果，确保指令执行的有效性和响应的及时性。
- 设备状态对比与异常检测：自动发起对智能开关的控制命令，实时监测设备状态变化，通过比对预设条件下的预期状态与实际状态，精准识别任何偏离正常行为的迹象。

- 测试记录与错误追踪：系统化记录每一次测试过程中的详细信息，包括但不限于命令、响应、设备状态等，一旦检测到异常，立即生成详尽的错误报告。
- 邮件通知与报告机制：集成邮件服务，当测试发现任何异常时，自动发送包含测试记录和错误详情的报告至指定接收人，确保问题的及时反馈与处理。
- 定时任务与持续监控：利用Cron作业或类似定时任务调度器，确保测试服务每天自动执行，持续监控智能开关的健康状况，提供不间断的性能保障。