钟轶

AI Agent 工程师 / Go 后端研发工程师

广东深圳 · 10 年+经验
186-8068-8653
doraemon.dream@qq.com
https://www.emonzhong.com

个人摘要

10 年+ 互联网与金融科技研发，具备企业级 **分布式训练与推荐工程 → 金融高并发后端 / 稳定性治理 → 企业级 AI Agent 落地** 的相关经验。

Agent 方向：基于 Hermes Agent 从 0 到 1 搭建企业自动化编码系统，熟悉 Agent 应用（流程标准化沉淀成业务 skills、MCP/CLI 对接业务平台、知识库搭建等）、Agent 系统原理（感知、推理、记忆、规划、执行/验证）和 Agent 评测。

Go 后端方向：具备券商 / 交易所交易系统和行情系统的研发与稳定性治理经验，参与高并发、故障隔离与可运维体系建设。

专业技能

AI Agent	Skill 编写、MCP 接入、Tool Call、上下文治理工程、记忆/检索系统、可观测/评测系统搭建、知识库搭建
Go 后端	高并发 WebSocket；连接池 / GC 优化；熔断限流；链路监控与故障 SOP；交易 / 行情相关稳定性治理
机器学习与推荐系统	分布式训练；推荐精排；ABTest 实验体系
语言	Go、Python、C++、Shell
工程与工具	GitLab CI/CD、MySQL；Cursor / Codex / Claude / Hermes Agent 等 AI Agent 系统
算法与数据结构	熟练，LeetCode 排名

工作经历

金证股份有限公司 · AI Agent 研发工程师

2026.04 – 2026.07

企业自动化编码 Agent 平台

实践总结：基于 Hermes Agent 的企业自动化编码落地

问题定义：企业编码交付要求可验收、可重入、可人工收敛。采用 **LLM Workflow（Orchestrator-Workers）**：人定阶段模板，编排 Agent 根据用户语意，路由场景，拆阶段任务，专责 Worker 按阶段执行。

系统分层：LLM 负责 Thinking / Plan；Hermes Agent 负责循环控制、工具 / Skill 执行与流程编排；外围用企微人审 + GitLab CI 构成 Agent 反馈/校验，形成「思考 → 执行 → 校验 → 反馈」闭环。

多 Agent 编排：经 Hermes Gateway（企微入口）接入；编排 Agent 按场景路由到编排模板，任务图写入 **Hermes Kanban**；各阶段 Agent 仅认领单一任务，经 handoff 交接，失败进入 blocked 状态并通知到人。

AutoCode 全链路：需求分析 → 详细设计 → 编码 → 单测。分析 / 设计阶段经 MCP 读写内部需求管理平台、和产品同事对齐需求盲点；通过分阶段检索、注入企业知识库内容，生成覆盖完整编码、测试（涵盖风险清单和关键决策）的执行方案后，进入人审 Review 对齐。编码/测试阶段通过沙盒隔离不同需求执行环境、提交代码触发 CI 检测流程；Agent 经 MCP 查询异机 GitLab Runner Pipeline 红绿并回写看板状态，通知用户。

工程价值：阶段边界清晰、可埋人审卡点、可重入；轨迹可追踪、评测/反馈方式更客观、支持多 Agent / 多模型按阶段分工，把不同模型能力做成可分阶段编排的能力。

技术栈：Hermes Agent · Kanban · Skill / MCP · GitLab CI · Python · Context Engineering

独立开发 / AI 应用 · 独立开发者 2025.09 – 2026.03

校园暴力行为检测（WiFi CSI 感知 + 深度学习）

主导基于 WiFi CSI 的人体姿态 / 行为识别，融合 CSI、语音与图像多模态信号，做非接触式暴力行为探测。

使用 WiFi CSI 完成信号采集与预处理（小波去噪、Hampel 滤波、相位校准）+ ABi-LSTM（Attention）进行模型训练和时序活动识别，来判别用户长时序行为（暴力检测场景，在优先保证召回率的情况下，训练准确率可以达到 85%+）。

在学校场景完成 CSI 数据收集和试点验证。

技术栈：WiFi CSI · Python · TensorFlow / PyTorch · 信号处理

BingX 交易所 · 后端研发工程师（T10） 2023.09 – 2025.08

合约交易稳定性治理 / 行情推送系统

负责 Go 高性能行情 WebSocket 推送系统的迭代；针对 GC 与连接池优化，服务单机峰值 CPU 消耗从 40% 降低至 20% 左右，压测连接数性能提升接近 **1 倍**。

负责 Go 高性能行情 K 线系统的迭代，完成离线 / 实时 K 线数据生成、数据合并的优化改造，K 线数据的 P99 查询延迟从原先的 1 秒+ 降低至 20ms 以内，单机 8 核 QPS 达到千万级查询能力。

参与单边行情引发的核心交易链路雪崩复盘与治理优化（一次金融系统雪崩的完整复盘）。

推动体系化系统改进：

- **连接数上限管控**：包含前端对非登录态用户连接数的限制，以及后端基于线上 WebSocket 8 核单机 4 万连接数的上限限制
- **核心服务熔断限流**：柜台、行情 K 线、WebSocket 等核心服务，完成熔断设置（根据服务压测 QPS、P95 延时作为熔断指标）
- **核心指标定义、监控完善**：对每个服务指标进行梳理，定义服务核心指标和下钻维度的指标，构建较全面的「指标分析 → 问题定位」的全链路服务健康度分析方式，方便日常服务巡检。
- **故障 SOP**：梳理和构建基于问题发现（告警）、快速问题定位（核心指标下钻分析 + 链路追踪日志串联分析）、快速应急止损（应急预案）的服务 Runbook，以此来快速响应并解决突发线上故障。

技术栈：Go · WebSocket · 熔断限流 · 监控告警 · 稳定性工程

富途控股有限公司 · 后端高级开发工程师 2022.05 – 2023.08

参与港股期货 / 衍生品系统的迭代和重构，并从 0 至 1 完成算法订单系统的架构研发与搭建。

参与港股期货 / 衍生品高并发低延迟交易相关系统开发重构。

- 参与港股期权、期货及衍生品交易系统的日常需求迭代和维护
- 参与系统重构的核心设计和研发、重构

负责算法订单模块从 0 至 1 完成全链路系统设计及拆单策略模块设计、研发、落地上线。

- 负责算法订单全链路系统架构设计
- 负责拆单模块和算法订单模块核心接口、交互方案和核心数据结构设计
- 负责完成拆单模块的开发，包含完整系统架构、上下游交互接口、幂等、无状态系统、Trace 链路和可观测性机制的设计研发。

网易云音乐 · 资深研发工程师 2020.11 – 2022.02

推荐精排：负责精排模块从老架构迁移至新架构相关研发工作。负责精排模块相关算子研发、云音乐业务接入等能力的建设。

ABTest 实验系统：在职期间，完成从 0 至 1 基于 Google 层叠实验架构原理 构建实验分流架构、假设检验指标分析、实时数据指标分析、数据埋点体系，支撑网易云音乐基于实验数据决策算法效果（在首页推荐、FM 等多个业务场景完成线上数据验证）。

阿里巴巴 · 资深研发工程师（P7） 2018.11 – 2020.11

1688 广告推荐 DMP：

- 系统重构：完成 B 类用户画像，通过优化系统架构，解耦系统下游依赖，治理/合并离线数据依赖产出，减少下游服务的调用量（减少 **70%** 左右），引入实时数据管道与离线/在线标签一体化体系，实现亿级用户 ID-Mapping 与秒级人群圈选能力，在线画像服务访问延迟 P95 从 1 秒+ 降至 50 毫秒内。
- 系统迭代：基于画像系统的访问特性和下游服务内容生产的时效性，引入带缓存的并发调度框架（Graph Call），将服务调用和逻辑构建之间的组件（算子）化，方便实验模型的可配置化和热插拔，配合算法同事，在单季度内完成几十次模型的迭代，单季度 RPM 最高提升 **10%+**。

腾讯科技 · 后端开发工程师 2013.07 – 2018.10

微信自研深度学习平台（Amber）

负责 Parameter Server 模块与底层网络通信（ZeroMQ / RDMA）优化。

基于版本号机制，完成 PS 服务 BSP、ASP、SSP 模式的梯度数据聚合存储与数据分发能力。

使用 RoCE 通信优化技术方案，将模型训练带宽从 **5Gb/s 提升至 40Gb/s**，支撑多机多卡线性扩展。

微信推荐系统 — 画像与精排工程（看一看）

画像系统：使用本地缓存优化画像系统，用户请求 **85%+** 命中缓存，针对冷启动的长尾用户，使用布隆过滤器优化查询访问效率，将服务平均延时从之前的 100ms+ 降低至 15ms 内。

精排系统：

- 线程池调度：由于文章之间的打分相互独立，将精排打分并行化处理，单次排序打分 item 数从 **400 提升至 2000**，平均调用耗时从 **80ms 降至 15ms**。
- 模型 KV 访问优化：精排模型中存在大量小型 KV 数据访问（LR / FM 模型），于是串联负责 KV 的同事、模型训练负责人，针对精排模型的训练、权重导出（Checkpoint）、线上访问（Batch Parameter Query 路由相同 KV 集群而不是原先分散的路由集群）等准实时模型的全流程、架构进行优化，将原先在线准实时模型的调用耗时从 50ms+ 降低至 5ms 以内。

微信负载均衡系统（MMLB）

负载均衡系统是微信所有后端服务 Infra 层用于服务流量访问控制的基础组件，在职期间核心完成快速拒绝和动态路由功能的开发。

- 快速拒绝（FastReject）：基于下游服务访问的延迟、失败率、队列堆积程度、端口可达情况（嗅探），判定是否应将下游某个节点加入拒绝访问列表，并在下游服务指标恢复后重新加入可访问列表。
- 动态路由：快速拒绝的方式在极端情况下（比如下游部分服务过载，或者服务分区）可能引起服务雪崩，因此引入了动态路由模式：基于服务端实际请求量、CPU 消耗动态计算对应关系；并针对特殊场景（比如所有服务都在短期内被判定为快速拒绝）引入服务降级策略（异常节点只拒绝部分流量，而不是整机流量拒绝），以此来平衡快速拒绝在资源达到瓶颈边界场景可能带来的服务雪崩问题，也尽可能平衡机器的流量。

技术栈：C++ · CUDA · ZeroMQ · RDMA · 分布式训练

教育背景

厦门大学 · 软件工程（本科） 2009.09 – 2013.07

ACM 区域赛银奖

微软编程之美前 50

Google Code Jam 前 500