

彭子豪

13786629856 | xiaobazeo@gmail.com

工作经历

小米科技有限责任公司

2024.07 - 至今

后端开发工程师 汽车部-自动驾驶与机器人部

负责内容:

- 带领 5 人团队主导部门在线生产协作与任务流转平台建设, 围绕数据采集、算法处理、人工作业、自动化清洗等地图生产场景, 建设分布式任务调度中心与流程引擎, 支撑月均 200w+ 流程实例流转, 日均 100w+ 算法任务调度, 累计接入迭代 40+ 条业务线。
- 参与建设内部 AI Agent 工程平台, 降低AI使用门槛和重复劳动, 落地 10+ Agent 工作流, 支撑部门 70+ 用户多协同工作。

项目经历

流程引擎

2024.07 - 至今

核心开发

项目简介: 旨在提升部门产线业务上下游协作 (比如: 数据采集准备、地图数据算法处理、人工作业等) 自动化能力, 同时支持全链路监控告警、生产人员产出交付量统计、可视化灵活配置多节点、多角色、多条件等能力, 支撑月均200w+ 流程任务流转调度, 累计支撑接入迭代 40+ 条业务线。

痛点:

- 流程实例与任务管理业务耦合, 没有实际的任务概念, 生产人员使用困难、任务监控能力弱、统计效率低等。
- 地图生产流程缺少统一的任务 / 节点抽象, 生产人员难以自助编排业务流程, 业务接入依赖研发重复开发, 搭建效率低时间长。
- Webhook 现状存在执行与 Pipeline 强耦合、无法区分服务并发、外部系统接入、回调链路扩展和异常恢复成本高等问题。
- 滚动更新、容器重启等case存在部分本地事务和消息发送丢失。

解决方案:

- 设计流程引擎与任务管理解耦方案, 将产线业务流程实例生命周期同步至任务管理服务, 支撑生产人员结算、产线交付量统计和全链路任务监控, 人力投入降低 50%, 人效提升 100%。
- 抽象 Flow / SubTask / 变量 / 节点类型 等领域模型, 建设自研 DAG 流程编排引擎, 支持流程图提交、节点配置、变量继承、导入导出与同步, 新业务接入开发周期缩短 40%。
- 设计 Webhook 与流程引擎解耦方案, 支持多服务来源异步限流执行, 并结合幂等校验机制, 任务流转时间缩短约 33%。
- 通过事务边界收敛、RocketMQ 有序消息与失败重试补偿、优雅停机机制, 保障本地状态变更与外部事件链路的最终一致性, 将事务丢失率降低至 0.01%。

分布式任务调度中心

2024.07 - 至今

核心开发

项目简介: 面向地图生产 Pipeline、SD 服务关联等多来源算法实例建设统一任务调度平台, 支持任务批量提交、优先级决策、Worker 拉取、进度回传、异常终止、超时重置和实例健康探测, 提升地图生产任务的自动化调度能力与系统可用性, 支撑日均 100w+ 任务调度, 计算节点同时协同运行 1000+。

痛点:

- 地图算法任务来源多、批次与任务状态难统一。
- Worker 并发拉取下重复领取、缓存脏数据和高优任务响应慢。
- 无法自动化有效调度 Worker 资源, 任务量过少时导致服务器资源过度浪费。
- Worker 异常退出、心跳中断或任务执行超时后, 执行中任务容易长期阻塞, 影响后续生产链路。

解决方案:

- 设计“批次-任务-执行记录”核心模型, 支持小批次聚合提交与大批次 startBatch / submitBulk / finishBatch 分阶段提交, 通过批次唯一号校验、事务化落库和 latestRecord 执行记录标记, 打通任务提交、分发、执行、回调、通知全生命周期, 支撑日均任务提交调度量100w+。
- 建设“探测-决策-下发”拉取式调度链路, 定时探测 DB 可执行任务并预热到 Redis 有界优先级队列, 结合实例类型、Worker 容量及版本、任务优先级和创建顺序进行调度决策, 并通过缓存候选、DB double-check、条件状态更新完成任务抢占, DB 查询量下降95%+, 重复派发率降低至0.01%。
- 设计可配置化动态扩缩容机制, 根据任务积压情况与机器负载调整算力资源, 资源利用率提升 50%, 有效解决算法执行过程中的热点倾斜与资源浪费问题, 节省部门年度预算 500w+。
- 建设服务实例注册、轻量级心跳、实例缓存、健康探测和超时重置机制, 通过定时探测识别异常实例并标记

TIDYING / DISCONNECTED, 联动重置执行中任务, DB写入量降低约70%, 将异常任务平均恢复时间降低至1分钟内。

AI Agent 工作流协作平台

2026.01 - 至今

核心开发

项目简介: 面向研发、算法及业务团队的内部建设可编排、可追踪、可恢复的多阶段 Agent 工作流能力的工程平台, 将部门知识库、MCP、SKILL等资产化, **已落地 10+ Agent 工作流, 支撑部门 70+ 用户多角色协同。**

痛点:

- 非技术用户直接使用 AI Agent 门槛高, 一次性对话难沉淀为可复用部门内部资产。
- Agent 缺少结构化运行上下文, 复杂任务启动依赖用户反复补充背景。
- Pod 初始化慢时, 前端、后端、K8s、Pod /init 与上下文预加载链路难以定位瓶颈。
- Agent 执行过程黑盒化, 工具调用、异常、进度与用户确认难以实时追踪。

解决方案:

- 实现 Agent Runtime 与工作流会话能力, 接入 Claude Agent SDK、MCP、workflow-context、Skills、WebSearch / WebFetch、AskUserQuestion 等工具体系, 将 Agent 能力封装为多阶段工作流, 支持任务探测、上下文注入、工具调用、人工确认、阶段性知识沉淀等功能。
- 建设上下文注入 Harness, 按 L0 / L1 分层探测 workflow、artifact、节点路径与资产上下文, 整合下发至 Agent Runtime; 通过 12s 总超时与 partial 降级策略, 保障上下文异常不阻塞会话启动。
- 建设 Trace / Perf 三层观测体系, 贯穿 openStage、ensurePod、K8sProvider、Pod /init、SSE ready 等关键阶段, 基于 traceId 聚合链路耗时并驱动优化, **将 Pod 初始化平均耗时由约 10s 降至约 3s。**
- 搭建事件驱动运行轨迹链路, 基于 SSE 实现 Pod → Backend → Frontend 纯透传, 实时下发消息、工具调用、异常、心跳、结构化提问与任务进度事件; 后端仅注入心跳、不缓存事件, 降低状态复杂度并提升可观测性。

教育经历

湖南工业大学

2020.09 - 2024.06

计算机科学与技术 本科 计算机学院 全日制

其他

- **技能:** 熟悉分布式事务、分布式锁、Java、SpringBoot、Kubernetes、Redis、RocketMQ、Mysql、Mybatis、RAG 等