Zhang Jia

18292060445 · plus_zzz@163.com · 2002-09
Homepage: pluszzz.top
GitHub: https://github.com/Pluszzz

Education

2024 - Present | **Northwest University** (211 & Double First-Class) | Software Engineering / M.S.

- "Huawei Cup" China Graduate Mathematical Modeling Contest — National Second Prize

2020-2024 | **Xi'an University of Posts & Telecommunications** | Information & Computing Science / B.S.

- National Undergraduate Mathematical Modeling Contest — Provincial Second Prize
- CET-4

Skills

- AI Agent & LLM Applications: Familiar with the development workflow of LLM-powered intelligent applications, understanding core mechanisms such as Agent workflow design, tool calling, task planning, and context management; experienced in LLM application architecture design for complex business scenarios
- LLM Application Development: Experienced in building enterprise-grade RAG systems, familiar with the complete pipeline of document parsing, Embedding, Milvus/Elasticsearch retrieval, hybrid recall, knowledge enhancement, and generation optimization
- Prompt Engineering & Context Optimization: Understand LLM prompt design methodologies, familiar with context window management, token optimization, structured output constraints, and inference optimization techniques; experienced in long-text scenario optimization
- LLM Service Engineering: Familiar with mainstream LLM API protocols and service integration, understanding streaming output, Function Calling, multi-model adaptation, and model service call chain design
- Java Backend Engineering: Proficient in Java and the Spring Boot ecosystem, mastering the WebFlux/Reactor asynchronous programming model; experienced in enterprise service development, API design, and system performance optimization
- Databases & Data Processing: Familiar with MySQL data modeling, index optimization, execution plan analysis, and transaction mechanisms; experienced in data storage design and SQL optimization for complex business scenarios
- Engineering Capability: Familiar with Git, Maven, Docker and other engineering tools, understanding Spec-Driven Development (SDD) workflows, with the ability to decompose requirements, design modules, deploy services, troubleshoot issues, and deliver projects

Internship Experience

Xi'an Hengpin Electronic Technology | AI Agent Engineer | 2026-03 ~ 2026-08

PlusAgent (Super Agent)

Project Overview: Enterprise-grade Multi-Agent Runtime platform built on the Orchestrator-Workers architecture, implementing Agent lifecycle management, task orchestration, tool ecosystem integration, context management, and structured artifact generation, enabling multi-Agent collaborative execution in complex business scenarios
Tech Stack: SpringBoot 3 + WebFlux(Reactor) + MCP + Groovy + MyBatis Plus + MySQL + Vue3
Responsibilities:

- Multi-Agent Orchestration Engine: Self-developed state-machine-driven Orchestrator-Workers orchestration engine, implementing task decomposition, sub-Agent scheduling, result review, and terminal state convergence; controlling Agent behavior via tool whitelists, execution round limits, and state guardrails
- Agent Runtime & Event System: Built multi-Agent asynchronous event flow architecture on WebFlux/Reactor, designing a unified Event Protocol to enable real-time interaction for Agent switching, tool invocation, and streaming output via SSE
- MCP Tool Runtime: Implemented an MCP JSON-RPC Client supporting stdio/HTTP dual transport protocols, realizing tool discovery, Schema transformation, and dynamic registration to integrate external MCP Server capabilities into the Agent Tool Runtime
- Context Engineering & Memory System: Built an Agent Context Pipeline implementing role isolation, context trimming, Token Budget control, and Prompt assembly; combining incremental summarization, quality scoring, and long-term memory accumulation for long-session management
- LLM Gateway & Artifact Generation Pipeline: Implemented a model access layer compatible with Anthropic/OpenAI protocols, resolving streaming response protocol differences; improving structured artifact generation stability via Markdown AST and JSON Repair

mechanisms

- Skill Runtime: Built a dynamic skill loading framework on GroovyClassLoader, supporting runtime compilation of tool scripts, incremental scanning, and zero-downtime updates for hot extension of Agent capability modules

PlusNexus (Intelligent Retrieval System)

Project Overview: Enterprise-grade intelligent search and knowledge retrieval platform, building natural language query parsing, search plan generation, multi-source retrieval fusion, and intelligent result generation on LLM Query Understanding, enabling users to query enterprise structured and unstructured data in natural language
Tech Stack: Spring Boot + Elasticsearch + Milvus + FastAPI + gRPC + MySQL + SSE
Responsibilities:

- Concurrent Logic Planning Engine: Developed the `Query Plan Tree` framework, using LLM to decompose complex Chinese queries into `AND/OR/NOT/SEARCH/COUNT` logic trees. Executed sub-queries concurrently with thread pools and short-circuit optimization, supporting full-set/difference-set computation and result branch tracing
- Metadata-Driven LLM Routing: Dynamically maintained database field Schema on the Java side while the Python side automatically extracted entities and injected rules, enabling new search dimensions with just one row added to the database — zero code or Prompt changes, solving the pain point of Prompt tightly coupling business knowledge
- Multi-Level Fallback & Dynamic DSL: Designed a three-level fault-tolerant chain of "LLM structured analysis → rule engine → keyword dictionary matching", ensuring stable retrieval even when models malfunction; developed `DynamicQueryBuilder` for automatic routing and dynamic assembly of ES `bool query` conditions
- Dual Protocol & Streaming Response Optimization: Adopted HTTP + gRPC dual protocol architecture, serving frontend access and high-performance Java service calls respectively; implemented staged streaming output via SSE, splitting result generation into "deterministic data construction → LLM summarization → citation association" to prevent LLM from directly generating structured factual data and improve perceived responsiveness for long-running tasks
- Microservice Decoupling & Dual-Level Cache: Independently packaged a Chinese relative-time parsing microservice (Time-MCP-server) to uniformly handle fuzzy time expressions; designed Plan/Parse dual-level cache for frequent identical queries, significantly reducing LLM call costs and improving response speed

Projects

RagentAI

https://github.com/Pluszzz/RagentAI 2025-09 ~ 2026-02

Project Overview: Enterprise-grade RAG intelligent agent platform, implementing a full-pipeline RAG system from document ingestion to intelligent Q&A. The system fully covers intent recognition, question rewriting, multi-channel retrieval, session memory, and model routing with fault tolerance for production environments. Distributed queue-based rate limiting controls concurrency, while circuit breakers and priority degradation ensure high availability across multi-model scenarios
Tech Stack: SpringBoot + MyBatis Plus + Milvus + Redis + Redisson + Apache Tika + Sa-Token
Responsibilities:

- Multi-Channel Retrieval Architecture: Designed a multi-channel retrieval engine (intent-directed + global vector retrieval + post-processing pipeline), improving retrieval precision while maintaining recall rate
- Question Understanding Optimization: Implemented LLM-based question rewriting and decomposition to automatically complete context and break down complex questions
- Session Memory Management: Designed a session memory compression strategy (sliding window + auto-summary) to reduce token consumption and solve long-context token overflow
- Model Scheduling & Fault Tolerance: Implemented model routing with a three-state circuit breaker, supporting multi-model priority scheduling and automatic failover
- Intent Recognition System: Built a tree-structured multi-level intent recognition architecture that proactively guides users to clarify requirements when confidence is low
- Tool Calling Framework: Implemented an MCP tool integration framework with registry-based auto-discovery, fusing knowledge retrieval with tool invocation
- Distributed Rate Limiting: Implemented a Redis ZSET + Lua sliding window rate limiting scheme supporting global and user-level concurrency control

Short Link Platform

https://github.com/Pluszzz/ShortLinkPlatform 2025-05 ~ 2025-08

Project Overview: SaaS short link system providing an efficient, secure, and reliable URL management platform for enterprise and individual users, with in-depth analytics and tracking features for flexible link management and optimization
Tech Stack: Spring Boot + Spring Cloud Alibaba + MySQL + Redis + RocketMQ + ShardingSphere + Sentinel
Responsibilities:

- Message Idempotency Control: Implemented MQ consumption idempotency with Redis, ensuring each message is consumed exactly once within a specified time window

- **Cache Penetration Protection:** Encapsulated a cache penetration protection scheme using double-checked locking and null-value caching to optimize queries for large volumes of non-existent data
- **Bloom Filter Optimization:** Used a Redisson Bloom filter for short link existence checks, improving efficiency and effectively reducing invalid query requests
- **Traffic Governance & Rate Limiting:** Integrated Sentinel for QPS-level rate limiting with graceful degradation to maintain system stability under traffic spikes
- **Async Peak-Shaving Architecture:** Adopted RocketMQ asynchronous consumption to shave peaks in high-concurrency monitoring data writes
- **Cache Consistency Design:** Designed the "update DB, delete cache" strategy to maintain consistency between short link cache and database
- **Distributed Concurrency Control:** Used Redisson distributed read-write locks to support concurrent data modification in high-concurrency short link scenarios
- **Sharding & Scaling:** Implemented shard routing with ShardingSphere to support targeted queries for redirect and pagination scenarios