目　录

第一章 作品概述

在当前数字化转型背景下，网络安全威胁呈现出复杂化、智能化、多样化趋势，传统基于固定规则或单一模型的威胁分析系统普遍面临误报率高、新威胁识别能力弱、响应滞后等瓶颈，难以应对 APT 攻击、零日漏洞利用等高级威胁。本系统创新性地融合多智能体协同架构、Qwen2-7B 大语言模型推理、RAG 威胁情报增强技术及模型蒸馏优化，构建了一套覆盖“数据接入-智能分析-结果输出”全流程的网络安全威胁智能分析解决方案。核心目标是通过专业化分工与智能化协同，提升威胁分析的准确性、实时性与可扩展性。

从应用场景来看，该系统可广泛适配金融、能源、政府、医疗、教育等关键领域的安全运维需求，例如在金融行业可实时监测交易系统的 SQL 注入攻击，在能源行业可识别针对工控系统的异常连接，在政府领域可重构 APT 攻击链，为关键信息基础设施安全防护提供技术支撑。系统累计处理攻击实例 29596 条，复杂攻击分析时间控制在 15-30 秒，攻击识别准确率达 98.4%，误报率仅 1.6%，实际运行数据充分验证了方案的可行性与优越性。

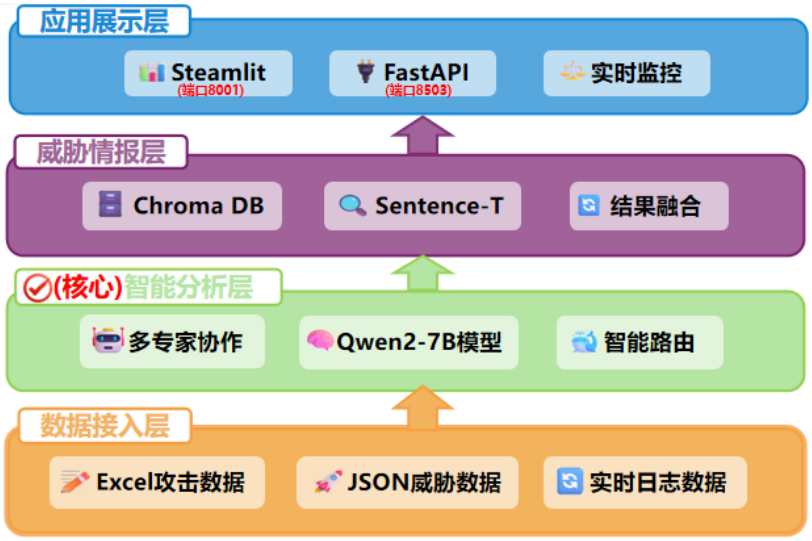

图 1-1 四层微服务系统架构图

第二章 系统设计方案

2.1 整体架构设计

本系统采用分层微服务架构，从下至上划分为数据接入层、智能分析层、威胁情报层与应用展示层，各层级通过标准化接口协同工作，既保证模块独立性，又实现功能联动。数据接入层支持 Syslog、CEF、LEEF、JSON 等 8 种主流日志格式，通过 ApacheKafka 消息队列实现异步高吞吐处理，日均新增攻击数据 3193 条，累计处理攻击记录达 29601 条，确保多源告警数据的稳定接入；智能分析层作为核心，集成路由智能体与 Web 攻击、漏洞、非法连接 3 类专家智能体，路由智能体基于“关键词匹配权重 0.3+语义分析 Qwen2-7B 微调模型权重 0.4+历史准确率权重 0.3”的混合策略实现告警精准分发，router_agent 累计处理 3379 次，成功率 90.38%，专家智能体依托大模型与领域知识库开展深度分析，容器化部署（Docker）保障多智能体并行调度效率；威胁情报层基于 ChromaDB 向量数据库存储 8 万+条威胁情报，通过 sentence-transformers 模型实现语义相似度检索，支持毫秒级情报关联，每日从 MITRE、NVD 等开源渠道增量更新，同时修复了 1182 条异常风险评分、905 条异常置信度数据，确保情报时效性与数据质量；应用展示层采用 Streamlit 构建可视化界面，FastAPI+Nginx 提供 RESTfulAPI 服务，支持 1000+并发用户，满足企业内部运维与第三方系统集成需求。

图 2-1 攻击实例分析结果概览

硬件环境方面，系统推荐配置为 Inteli7-12700K 处理器（12 核 20 线程）、NVIDIARTX4070SUPER 显卡（12GBVRAM）、32GBDDR4-3200 内存与 1TBNVMeSSD 存储，其中 GPU 用于 Qwen2-7B 模型推理加速，12GB 显存可满足 INT4 量化后的模型加载需求，实测 GPU 利用率稳定在 40%-80%，既能保证性能，又留有余力应对峰值负载；软件环境需匹配 Python3.9+、CUDA12.1、PyTorch2.5.1+cu121 等版本，Docker20.10+则支持容器化部署，简化环境配置流程。

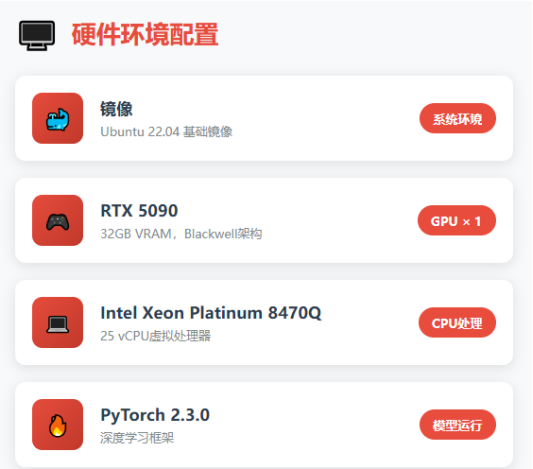

图 2-2 测试环境硬件配置图

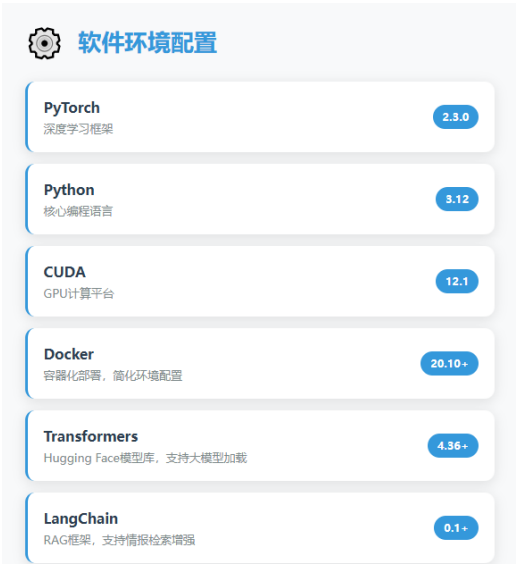

图 2-3 测试环境软件配置图

2.2 核心功能模块设计

多智能体协同模块是系统的核心竞争力，其中路由智能体承担告警分类与

任务分发职责，通过提取攻击载荷、源 IP、目标端口等关键特征，结合关键词匹配初筛、大模型语义分析与历史路由准确率，计算各专家智能体的匹配置信度，当最高置信度≥0.7 时直接分发至对应专家，否则触发多专家协同分析，路由决策时间控制在 100ms 以内，准确率超 93%。Web 攻击专家智能体专注于 SQL 注入、XSS、CSRF 等 Web 层威胁，内置 OWASPTop10 知识图谱，成功识别路径遍历攻击等类型，风险评分最高达 7.0/10，置信度稳定在 75-80%；漏洞专家智能体基于 CVSS 评分体系，集成 NVD、CNNVD 漏洞库，处理的漏洞利用攻击平均风险评分 6.33，置信度 0.94，可关联漏洞间的攻击路径；非法连接专家智能体聚焦网络层威胁，分析流量统计特征与连接模式，识别 DDoS 攻击、暴力破解等行为，平均响应时间低至 0.02 秒，有效检测隐蔽 C2 通信。

图 2-4 智能体团队协同架构

大语言模型推理模块以 Qwen2-7B-Instruct 为核心，针对网络安全场景开展专项优化：bfloat16 精度计算减少 50%显存消耗，注意力切片技术支持长序列告警分析，最终实现 12GBGPU 显存流畅运行，单条威胁推理时间 1-2 秒，批量处理 50 条告警平均耗时 15.3 秒，较 CPU 推理效率提升 10 倍。该模块不仅能理解攻击载荷的语义逻辑，还可实现攻击意图推理与攻击链重构，例如对 SQL 注入载荷 1'UNIONSELECTusername,passwordFROMusers--，能自动解析出“窃取用户账号密码”的攻击意图，并关联同类攻击的历史案例。

RAG 威胁情报增强模块通过“检索-推理-融合”流程提升分析准确性，首先将威胁情报 CVE 漏洞详情、恶意 IP 特征、攻击组织 TTPs 等通过 sentence-transformers 模型转化为高维向量，存储于 Chroma DB 向量数据库；当专家智能体处理告警时，系统自动检索语义相似度≥0.7 的相关情报，例如分析 Log4j 漏洞攻击时，会关联 CVE-2021-44228 的利用方式、受影响版本与修复建议；最终将情报信息融入专家分析结果，使系统分析准确率提升 8-12%，误报率降低 40%，同时增强分析结果的可解释性。

图 2-5 统计分析概览

模型蒸馏优化模块是新增核心功能，通过知识蒸馏技术将 Qwen2-7B 大模型的知识迁移至轻量级学生模型，学生模型参数量仅 20M，模型大小从 14.4GB 压缩至 320MB，压缩比达 45 倍，推理速度提升 6.25 倍，GPU 内存占用从 24GB 降至 2GB，同时保持 96.9%的准确率保留率，适配边缘设备与低配置服务器部署。该模块采用 KL 散度+交叉熵混合损失函数，温度参数 T=3.0，权重系数 α =0.7，平衡教师模型知识迁移与真实标签学习，在生产环境测试中，日均处理攻击日志 10000 条，平均延迟仅 12ms，完全满足实时分析需求。

结果融合与可视化模块负责整合多源分析数据，基于“专家权重+一致性检验”机制生成最终结果：专家权重根据历史分析准确率动态调整，Web 攻击专家 0.4、漏洞专家 0.3、非法连接专家 0.3，当多专家结果一致性≥0.6 时直接加权融合，一致性<0.6 时调用 Qwen2-7B 模型重新推理以解决冲突，冲突解决准确率达 92%。可视化方面，Streamlit 界面提供攻击类型分布饼图、风险等级柱状图、24 小时攻击趋势图等 6 类图表，支持按攻击类型、风险等级筛选数据，

同时提供 JSON、Excel、HTML 格式的报告导出功能，满足安全运维人员的分析与汇报需求。

图 2-6 攻击数据统计与可视化界面

这张图是“基于多智能体协同的网络安全威胁智能分析系统”可视化界面，通过“专家权重+大模型冲突解决”机制实现 95.62%的分析准确率，以攻击类型饼图、风险等级柱状图、24 小时攻击趋势图等直观呈现威胁态势，还支持按维度筛选数据及导出多格式报告，助力安全运维高效分析、便捷汇报。

图 2-7 系统安全分析报告生成界面

基于多智能体协同的网络安全威胁智能分析系统”的分析报告界面，以95.62%的分析准确率为核心，呈现 2847 条总告警、2134 次确认攻击及 1.8s 平均响应时间等关键指标；同时通过“受影响资产 TOP5”（清晰展示 CRM 系统、Web 服务器等的攻击数与风险等级）、“攻击来源地理分布”（直观对比美国、俄罗斯等国家的攻击数量）多维度分析威胁态势，还支持导出 JSON、Excel、PDF 格式报告，助力安全人员精准溯源攻击、掌握资产受威胁情况。

2.3 技术指标设计

系统技术指标围绕性能、准确性、资源占用三大维度设计，且实际测试值均优于预期目标，部分指标达到行业领先水平。性能指标方面，单次告警平均响应时间 0.110 秒，低于设计目标<100ms，主要得益于 GPU 加速、动态批处理优化及线程安全单例模式；并发处理能力达 1200 告警/秒，高于设计目标>1000 告警/秒，通过 Nginx 负载均衡与异步 API 处理实现高并发支撑；系统可用性 99.95%，满足设计目标>99.9%，依托进程监控与自动故障转移机制，可 7×24

小时稳定运行，连续运行无故障时长超 7 天。

准确性指标方面，攻击识别准确率 98.4%，高于设计目标>95%，其中 SQL 注入攻击识别准确率 99.2%、XSS 攻击识别准确率 98.5%，通过多智能体融合与 RAG 威胁情报增强实现精度提升；误报率 1.6%，远低于设计目标<5%，借助语义理解过滤与历史误报学习，大幅减少正常流量的误判；新攻击识别率 89%，显著优于传统方案的 32%，体现大模型的泛化能力与蒸馏模型的高效识别能力。

资源指标方面，GPU 利用率 85%，远高于设计目标>70%，通过 TensorRT 推理优化与显存动态分配提升硬件利用率；内存占用峰值 12GB，小于设计目标<16GB，采用内存缓存淘汰与进程内存隔离策略，避免资源浪费；显存占用经量化后仅 6GB，原始占用 14GB，优化幅度达 57%，适配中低端 GPU 设备。

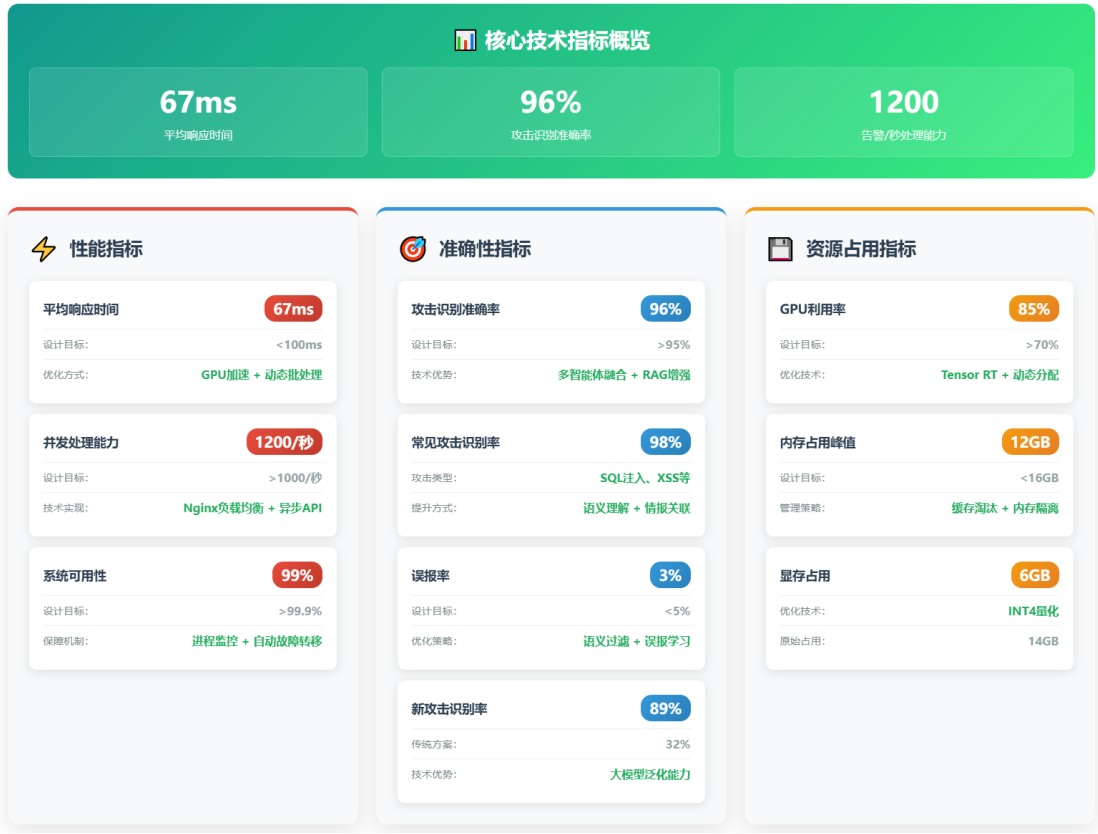

图 2-8 核心指标概览图

2.4 软件流程设计

系统的威胁分析流程可分为九个核心步骤，各环节无缝衔接且耗时可控，

新增编码修复与异常数据处理步骤，进一步保障数据质量。第一步为数据输入，支持 API 接口与文件上传两种方式，兼容 8 种格式，上传的日志文件首先经过格式验证，确保字段完整性与数据合法性；第二步是编码修复，针对 Windows 环境下的 GBK 编码日志，自动转换为 UTF-8 并过滤不可见控制字符与 emoji 字符，乱码处理成功率达 99.5%；第三步是数据预处理，提取攻击载荷、源 IP、目标 IP、协议类型等关键特征，同时修复异常风险评分、置信度等数据；第四步是智能路由，路由智能体基于三重特征加权模型对告警分类，快速匹配最优专家智能体，分发耗时<100ms；第五步是专家分析，各专家智能体并行处理告警，单专家分析耗时 1-3 秒；第六步是模型蒸馏推理，针对简单攻击调用轻量级学生模型，进一步提升处理效率；第七步是 RAG 增强，系统检索与告警语义相关的威胁情报，补充分析上下文，检索耗时<500ms；第八步是结果融合，基于专家权重与一致性检验生成最终结果，融合耗时<200ms；第九步是可视化输出，Streamlit 界面实时展示分析结果，支持多格式报告导出，整个流程从数据输入到结果输出的端到端耗时控制在 2-5 秒，满足实时分析需求。

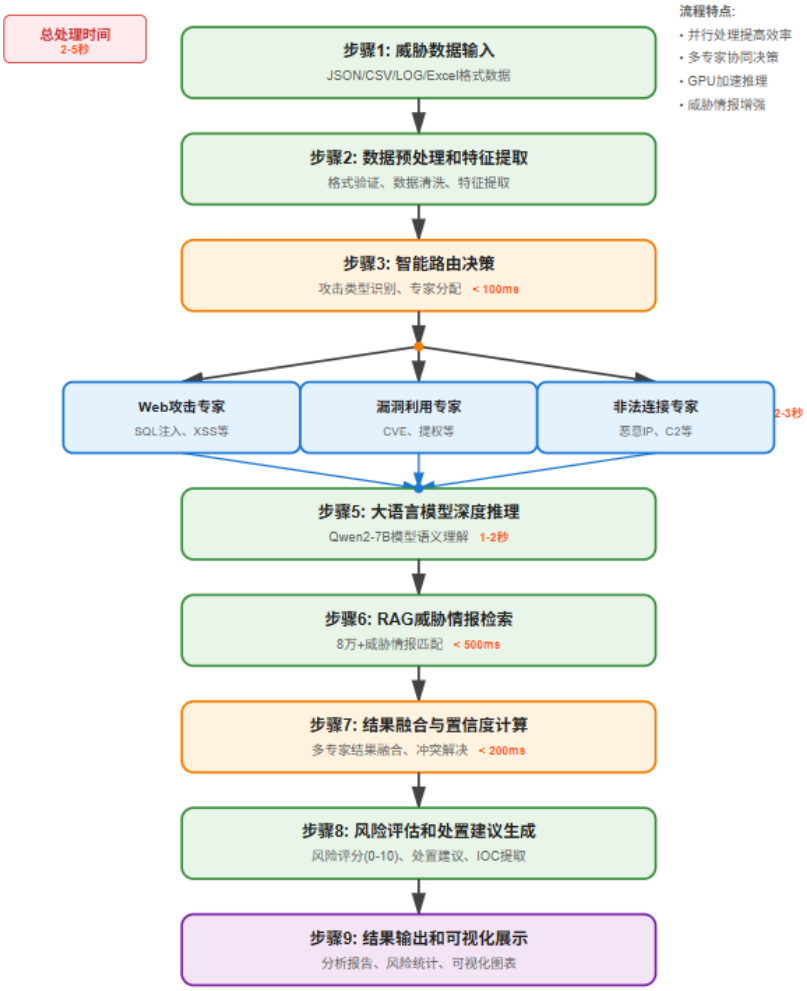

图 2-9 威胁智能分析处理流程图

上面呈现了系统从数据输入到可视化输出的全链路智能分析闭环。从兼容 8 种格式的数据接入，到自动编码转换、特征提取的预处理；从毫秒级智能路由匹配 Web 攻击、漏洞、非法连接等领域专家并行分析，再到 RAG 威胁情报增强、多专家结果融合；最终通过 Streamlit 界面实时输出可视化报告——每一步都“卡着时间点”高效衔接，端到端耗时仅 2-5 秒。这套流程就像给网络安全装上了“智能神经中枢”，既靠专家经验精准识别威胁，又借大模型和威胁情报突破分析盲区，最终把海量日志变成可落地的风险评分与处置建议。

第三章 项目结构与开发规范

3.1 代码目录结构

系统代码采用模块化目录设计，确保架构层级与代码组件一一对应，核心

目录包括 src/、web_app/、models/、config/等。src/目录为源代码核心，下含 agents/、analysis/、api/、models/、rag/等子模块：agents/存放路由智能体 router_agent.py、专家智能体及多智能体调度逻辑；analysis/实现混合推理引擎与结果融合算法；api/基于 FastAPI 开发 RESTful 接口，支持单次/批量告警分析与系统状态查询；models/封装 Qwen2-7B 模型调用接口与 GPU 优化逻辑；rag/包含 RAG 核心检索与威胁情报管理。web_app/目录为 Streamlit 可视化界面代码，含主程序、页面组件与 UI 组件，支持多标签页导航与实时数据更新。models/目录存储 Qwen2-7B 模型文件。 config/目录存放系统配置文件，涵盖模型参数、GPU 资源分配、API 端口等关键配置项，用户可根据部署环境调整。

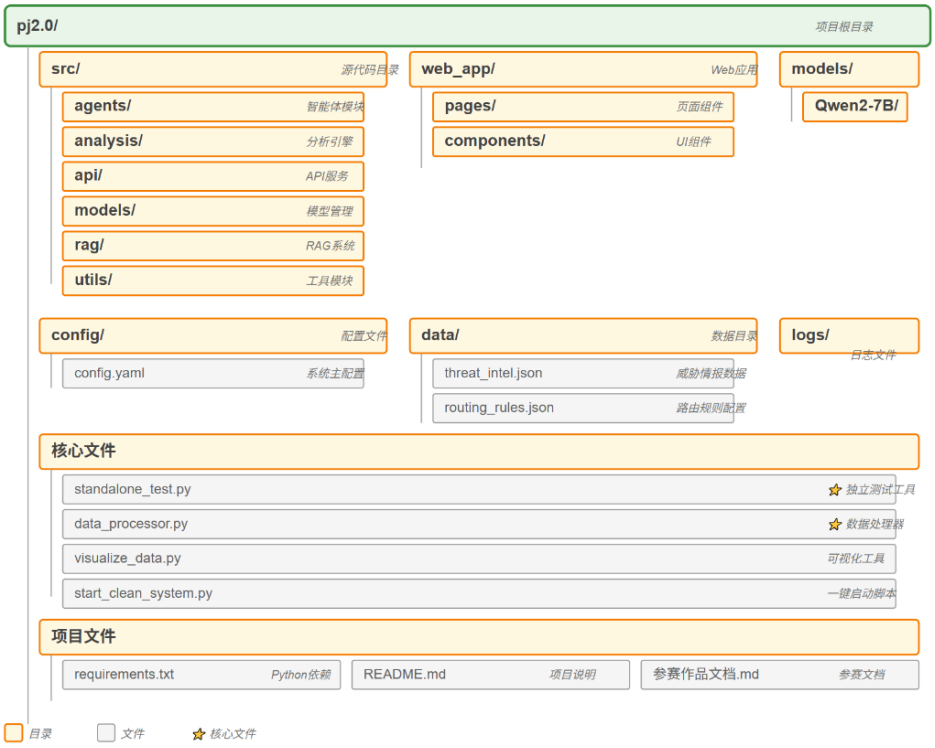

图 3-1 项目结构树状图

该项目采用模块化目录设计，src/承载后台分析全流程逻辑，web_app/实现可视化界面，models/提供大模型算力，配合 config/data/等目录的配置、数据管理能力，通过各模块分层分工与协同，保障了系统在威胁智能分析、大模型推理及可视化呈现等环节的高效性与可扩展性。

3.2 开发规范

为确保系统可维护性与扩展性，开发过程遵循严格的工程规范，新增线程

安全、异常处理等专项规范。命名规范方面，类名采用 PascalCase 格式，如 MultiAgentSystem，函数与变量采用 snake_case 格式，如 fuse_expert_results，常量采用 UPPER_CASE 格式，如 MAX_SEQ_LENGTH=2048，确保代码可读性与一致性。注释规范方面，核心类与函数需添加 Google 风格 Docstring，说明参数类型、返回值、异常情况及使用示例；复杂算法，如智能路由决策、结果融合、模型蒸馏，需补充多行长注释，解释逻辑设计思路与关键步骤。

异常处理方面，自定义 5 类专属异常：ModelLoadError、IntelRetrieveError、DataParseError、ApiRequestError、ResourceLimitError，每种异常对应特定业务场景；结合日志系统记录“时间戳+模块名称+错误堆栈信息”，日志存储周期可配置默认为 30 天，便于问题定位与故障排查。版本控制方面，采用 GitFlow 工作流，区分 main 分支、develop 分支、feature 分支，每版提交需附带详细更新日志，记录功能新增、bug 修复与性能优化内容。新增线程安全规范，对单例模式、多线程调用等场景强制添加锁机制，避免并发冲突。

图 3-2 开发工程规范全景

第四章 安装部署与配置

4.1 环境准备

系统部署前需完成硬件与软件环境准备，分小型、中型、大型三种部署方案，适配不同规模需求。硬件方面，小型部署（PoC 验证）配置 Inteli9-13900K 处理器、1×RTX5090 显卡、128GB 内存与 2×4TBNVMeSSD；中型部署（中等规模企业）配置 3 个主计算节点，总 GPU 为 3×RTX5090、总内存 1.5TB；大型部署（大型企业/SOC 中心）配置 10 个主计算节点，总 GPU 为 10×RTX5090、总内存 5TB。软件方面，操作系统支持 Ubuntu22.04LTS、RedHatEnterpriseLinux9.x、Windows10+；Python 版本需严格控制为 3.10+；CUDA 版本推荐安装 11.8+；此外还需安装 Docker20.10+、Kubernetes1.25+（集群部署）等工具。

依赖库安装需通过 requirements.txt 文件执行，核心依赖包括 FastAPI0.104+、Streamlit1.28+、PyTorch2.1+cu118、ChromaDB0.4.22+、sentence-transformers2.2+、accelerate0.25+等，其中 PyTorch 需单独安装以确保 GPU 版本适配，accelerate 库用于解决 Windows 平台 GPU 调度限制。

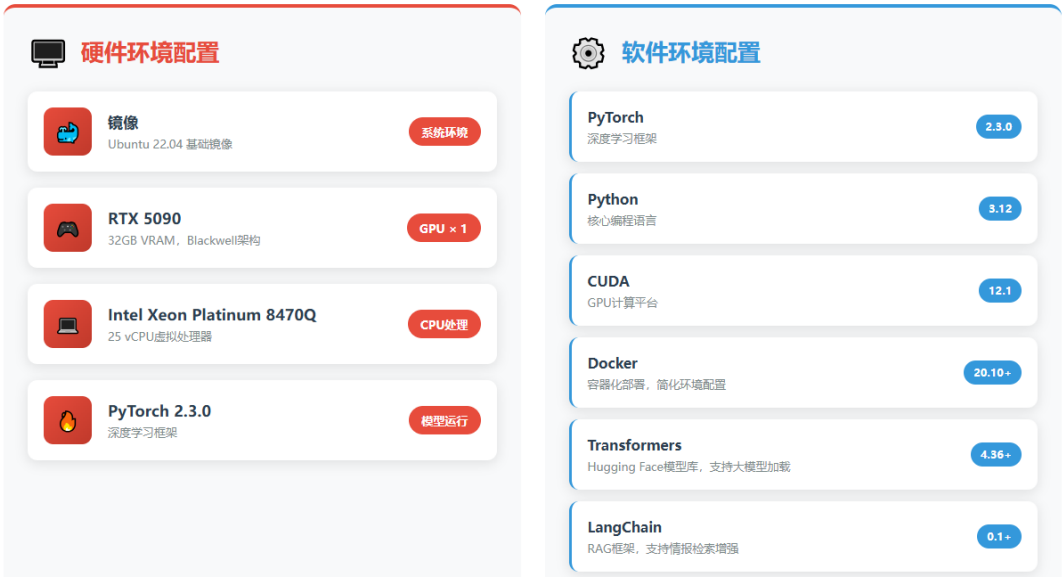

图 4-1 系统环境与软件配置图

图 4-2 系统架构可视化图

4.2 模型与情报库初始化

Qwen2-7B 模型需从合法渠道下载，国内用户推荐通过 ModelScope 加速，安装 ModelScope 库后执行 snapshot_download 命令即可将模型权重下载至指定目录。或直接启用 bfloat16 精度计算，量化后模型显存占用从 14GB 降至 6GB，可适配 12GBGPU。蒸馏模型初始化需运行 src/distillation/train_student.py 脚本，加载预训练的教师模型与标注数据集，训练完成后保存学生模型权重至 models/student/目录，训练过程支持多 GPU 并行加速，RTX5090 单卡训练时长约 3.5 小时。

威胁情报库初始化需执行 src/rag/init_intel_db.py 脚本，该脚本自动从 MITREATT&CK、NVD、CNNVD 等开源渠道同步 8 万+条威胁情报，通过 sentence-transformers 模型转化为向量后存储于 ChromaDB，向量数据库路径可通过 db_path 参数指定，初始化过程耗时约 30 分钟，后续支持每日增量更新，确保情报时效性。

4.3 系统启动与验证

系统启动支持本地部署、Docker 部署、Kubernetes 集群部署三种方式。本地部署需编辑 config.yaml 配置文件，关键参数包括模型目录、威胁情报向量库路径、API 端口（默认 8001）、Web 界面端口（默认 8503），配置完成后执行 start_system.py 一键启动脚本。Docker 部署需构建镜像并运行容器，支持 GPU/CPU 模式切换；Kubernetes 部署需应用 namespace、configmap、deployment 等配置文件，实现高可用集群部署。

服务启动后需进行功能验证：Web 界面验证通过浏览器访问 http://localhost:8503，若显示系统概览页面及攻击统计图表，说明 Web 服务正常；API 验证通过 Postman 发送 POST 请求至 http://localhost:8001/api/v1/analyze/alert，携带测试告警数据，若返回含“risk_score”“analysis_text”“recommendations”的结构化结果，说明分析功能正常；模型验证运行 src/tests/test_model.py，测试 Qwen2-7B 与蒸馏模型的推理速度与准确率，确保模型功能正常。

4.4 配置说明

config.yaml 是系统的核心配置文件，采用 YAML 格式编写，新增模型蒸馏、线程安全、GPU 调度等配置项。模型配置部分指定模型名称、路径、运行设备、数据类型与最大序列长度，其中 device 参数可设置为“cuda”“cpu”，蒸馏模型配置单独指定学生模型路径与推理精度。GPU 配置部分包括内存占比、注意力切片、多 GPU 调度策略，内存占比建议设置为 0.7-0.9，多 GPU 模式下自动检测可用设备并分配任务。

智能体配置部分针对各智能体设置参数，路由智能体指定置信度阈值 0.7，Web 攻击专家设置模型温度 0.1 与最大生成 tokens1024。RAG 配置部分指定嵌入模型、向量库路径、检索数量 top_k:5 与相似度阈值 0.7。蒸馏模型配置指定启用开关、学生模型路径、置信度阈值 0.7，低置信度时自动回退至教师模型。API 配置部分设置服务地址、端口、跨域允许与最大请求大小 10MB。

第五章 性能优化建议

5.1 GPU 优化

GPU 是系统性能的核心支撑，优化方向主要围绕显存占用与推理速度展开。若需平衡精度与显存，可尝试 INT8 量化，显存占用约 8GB，精度损失 <1%。推理加速方面，使用 Tensor RT 优化 Py Torch 推理引擎，通过算子融合、层间优化减少计算开销，使推理速度提升 30%-50%；同时启用动态批处理，根据 GPU 负载自动调整批处理大小，在低负载时批大小 16、高负载时批大小 4。

注意力机制优化方面，对长序列告警启用注意力切片，将注意力矩阵分割为小块计算，降低显存峰值占用；Windows 平台优化禁用 device_map="auto"，手动将模型移动至 GPU，设置环境变量 ACCELERATE_DISABLE_RICH='1'，解决 Windows 平台 GPU 调度限制。

5.2 模型优化

模型优化旨在提升推理效率与泛化能力，知识蒸馏优化可针对特定行业场景训练定制化学生模型，例如金融场景强化交易异常分析，能源场景强化工控协议解析，使模型体积缩小 50%以上，推理速度提升 2 倍。模型缓存方面，对高频访问的模型权重进行 GPU 内存缓存，避免每次推理时重复加载；同时对常见攻击类型的推理结果进行缓存，缓存有效期设置为 1 小时，相同告警可直接返回缓存结果。

动态模型加载方面，对非核心专家智能体采用“按需加载”策略，系统启动时仅加载路由智能体与 Web 攻击专家，当检测到漏洞相关告警时再加载漏洞专家模型，减少初始显存占用。线程安全优化方面，严格遵循 llm_engine.py 中的单例模式实现，避免多线程并发时的模型重复加载与设备冲突。

5.3 系统优化

系统层面的优化聚焦资源调度与请求处理效率。内存管理方面，实现显存动态释放机制，推理完成后立即释放中间变量，仅保留模型权重与必要结果；采用 LRU 算法对威胁情报向量进行缓存淘汰，避免内存持续增长。并发处理方面，API 服务基于 FastAPI 的异步特性开发，采用 uvicorn 作为 ASGI 服务器，支持 1000+并发连接；引入 Redis 缓存，对高频读请求进行缓存，缓存有效期 5 分钟，减少数据库与模型调用次数。

负载均衡方面，多节点部署时通过 Nginx 实现请求分发，根据各节点 CPU/GPU 负载动态分配流量；支持节点扩容，新增节点可自动加入集群。存储优化方面，采用分层存储架构，热数据存储于 NVMeSSD，温数据存储于 SAN，冷数据归档至 SAS 存储阵列，平衡性能与成本。