

“中国电子杯”第三届“高校 ICT 产教融合创新大赛

赛题七

一、赛题名称

MaaS 平台的智能路由与多级缓存引擎设计

二、命题单位

中电金信软件有限公司

三、所需知识背景

编程与工程能力：熟练掌握至少一门高性能编程语言（如 Go、Java、Rust），熟悉常用的 Web 框架与网络编程（HTTP/gRPC）。

大模型与 AI 基础：了解主流大语言模型（LLM）的交互协议（如 OpenAI API 兼容格式），理解 Embedding（文本嵌入）、向量检索以及 Transformer 架构的基本原理。

中间件与分布式系统：熟悉 Redis/Kafka 等常见中间件的使用与原理，了解分布式系统中的负载均衡、缓存一致性等核心概念。

四、赛题背景

在 MaaS 平台接入模型数量和应用场景激增的背景下，单纯依赖推理引擎层的底层优化（如算子融合、PD 分离等）虽然能提升单机性能，但面临着极高的技术门槛、以及模型厂商、硬件绑定等局限性。

相比之下，在 MaaS 网关层进行工程侧优化，具有“全局视野、即插即用、业务无感”的特点，能够以更高的投入产出比（ROI）解决大流量下的首字延迟（TTFT）、Token 吞吐速率（TPOT）和 GPU 利用率问题。

传统的负载均衡策略（如轮询、最小连接数），难以感知 GPU 负载、上下文长度和语义复杂度等关键 AI 特征。因此，在 MaaS 网关层构建一套具备推理全链路感知能力的智能‘交通指挥系统’，通过精细化调度来平抑推理延迟抖动并优化算力成本，已成为 AI 工程化落地不可或缺的关键基础设施。

五、赛题描述

高性能的智能路由与缓存网关模块设计：要求能够根据后端多个模型实例的实时负载（如显存水位、当前并发数），动态分配用户请求；同时，针对重复的系统提示词（System Prompt）或高频问答，设计一套应用层或推理层的前缀缓存（Prefix Cache）机制，以减少重复计算。

产出可独立部署的网关系统代码：支持多模型动态接入、基于语义的智能流量分发以及多级缓存加速，显著降低 MaaS 平台的推理延迟与运营成本。

六、评价指标

各赛程阶段任务划分参考：

校内初赛阶段：侧重系统架构设计，核心调度算法实现与缓存策略设计，同时提供实验室仿真测试报告作为参考依据；

决赛小组赛阶段：侧重实际的效果验证，模拟生产环境，由评委组配合完成部署与测试验证。接入 Qwen 或 DeepSeek 系列 等主流开源模型。通过自动化脚本压测，对比“开启网关优化”与“直连后端模型（Baseline）”的性能差异。

校内初赛

1.校内初赛作品提交要求

(1) 技术文档-设计报告:提交 Word 版本及 PDF 版本各 1 份，建议封面或目录后的第 1-2 页项目为重要内容预览页，包含内容：

1、架构设计：阐述 MaaS 网关的整体逻辑架构，包括请求接入、路由转发、后端模型管理等模块的设计思路。

2、核心算法：重点描述动态调度算法（如负载均衡策略、优先级队列设计）和缓存优化策略（如多级缓存设计、缓存一致性机制、语义缓存匹配算法）。

3、创新点分析：对比传统网关或开源方案，说明本作品在性能提升、成本控制或功能扩展上的独特创新。

(2) 技术文档--测试与性能分析报告：格式要求同上（Word & PDF），内容需包含详实的数据支撑。内容要求：

1、测试环境：明确说明测试所用的硬件配置（CPU、内存、GPU 型号）及软件

环境（操作系统、依赖库版本）。

2、性能基准：列出在固定算力下，部署指定参数规格模型（如 Qwen2.5-14B）时的基准性能数据（Baseline）。

3、优化效果：提供开启网关优化前后的对比数据，核心指标需包含 首字延迟降低率、系统吞吐量提升率、缓存命中率 等，并附带压力测试工具（如 JMeter, Locust, wrk2）的配置截图及结果图表。

（3）技术数据--源代码与部署包：作品原始文件（ZIP 数据包），主要包含可编译、可运行的完整工程文件。内容要求：

1、源代码：网关核心代码、调度插件代码、缓存模块代码等。代码需有清晰的目录结构和关键注释。

2、依赖说明：提供 requirements.txt (Python)、pom.xml (Java) 或 package.json (Node.js) 等依赖管理文件。

3、部署脚本：提供 Dockerfile、docker-compose.yml 或一键部署 Shell 脚本，确保评审环境可快速拉起服务。

4、配置文件：脱敏后的网关配置文件（如 config.yaml），需包含路由规则、后端模型地址、缓存参数等示例。

（4）演示视频

提交格式：MP4 格式，时长控制在 3-5 分钟。

内容要求：

1、功能演示：录屏展示网关的接入过程、API 调用流程及并发请求下的响应情况。

2、可视化展示：如有开发可视化监控面板（展示实时 QPS、延迟、缓存命中情

况)，请重点演示。

3、讲解配音：配合画面讲解系统运行逻辑及性能优化效果，无需露脸。

评分标准（总分 100 分）

大项	内容	评分要求	分值
现场表现	文档质量	文档结构清晰，原理分析合理，代码注释规范	10 分
	演示效果	系统演示流畅，功能展示完整	5 分
	答辩表现	回答问题准确，逻辑清晰	5 分
方案设计创新性	架构合理性	网关整体架构清晰，模块划分合理（路由、限流、缓存解耦）。能够清晰阐述数据在网关中的流转路径。	15 分
	算法先进性	调度算法：有明确的数学模型或逻辑图描述（如加权轮询、最小连接数、或基于预测的动态调度）。 缓存策略：设计了合理的缓存淘汰算法（LRU/LFU）或语义缓存匹配算法。	20 分
技术实现与代码质量	代码规范	代码结构清晰，命名规范，关键逻辑有详细注释。遵循主流编程语言的工程化标准。	5 分
	功能完整性	实现核心的调度与缓存模块接口，即使未接入大规模集群，代码逻辑应完整。	10 分
验证与仿真分析	配置与部署	提供清晰的 README 和配置文件（YAML/JSON），能够通过 Docker 或脚本快速在本地启动服务。	10 分
	仿真测试方案	重点考察：由于缺乏大规模真机，学生需提供基于 Mock 服务的仿真测试报告。例如：使用脚本模拟后端不同延迟，验证网关的超时熔断或慢节点剔除逻辑是否生效。	10 分
	理论性能分析	能够通过公式推导或架构图分析，论证其方案在理论上如何降低延迟或提升吞吐（例如：分析加锁粒度对并发的影响）。	10 分

决赛小组赛

1.决赛小组赛作品提交要求

（1）答辩 ppt：凝练作品内容、方案、亮点、创新点，并阐述作品相较于。注意：为确保比赛公平公正，ppt 中不能出现队伍单位、人名等识别信息。答辩 PPT 模板由主办单位提供，请关注组委会和赛题交流群通知。

（2）技术文档-设计报告:提交 Word 版本及 PDF 版本各 1 份，建议封面或目录

后的第 1-2 页项目为重要内容预览页，包含内容

- 1、架构设计：阐述 MaaS 网关的整体逻辑架构，包括请求接入、路由转发、后端模型管理等模块的设计思路。
- 2、核心算法：重点描述动态调度算法（如负载均衡策略、优先级队列设计）和缓存优化策略（如多级缓存设计、缓存一致性机制、语义缓存匹配算法）。
- 3、创新点分析：对比传统网关或开源方案，说明本作品在性能提升、成本控制或功能扩展上的独特创新。

（3）技术文档--测试与性能分析报告：格式要求同上（Word & PDF），内容需包含详实的数据支撑。内容要求：

- 1、测试环境：明确说明测试所用的硬件配置（CPU、内存、GPU 型号）及软件环境（操作系统、依赖库版本）。
- 2、性能基准：列出在固定算力下，部署指定参数规格模型（如 Qwen2.5-14B）时的基准性能数据（Baseline）。
- 3、优化效果：提供开启网关优化前后的对比数据，核心指标需包含 首字延迟降低率、系统吞吐量提升率、缓存命中率 等，并附带压力测试工具（如 JMeter, Locust, wrk2）的配置截图及结果图表。

（4）技术数据--源代码与部署包：作品原始文件（ZIP 数据包），主要包含可编译、可运行的完整工程文件。内容要求：

- 1、源代码：网关核心代码、调度插件代码、缓存模块代码等。代码需有清晰的目录结构和关键注释。
- 2、依赖说明：提供 requirements.txt (Python)、pom.xml (Java) 或 package.json (Node.js) 等依赖管理文件。

- 3、部署脚本：提供 Dockerfile、docker-compose.yml 或一键部署 Shell 脚本，确保评审环境可快速拉起服务。
- 4、配置文件：脱敏后的网关配置文件（如 config.yaml），需包含路由规则、后端模型地址、缓存参数等示例。

2.决赛小组赛阶段评分标准（总分 100 分）

参赛作品需部署在指定规格的服务器集群上(模拟生产环境)，并接入 Qwen2.5-14B 或 DeepSeek-V3-33B 等主流开源模型。通过自动化脚本压测，对比“开启网关优化”与“直连后端模型（Baseline）”的性能差异。

大项	内容	评分要求	分值
现场表现	文档质量	文档结构清晰，原理分析合理，代码注释规范	10 分
	演示效果	系统演示流畅，功能展示完整	5 分
	答辩表现	回答问题准确，逻辑清晰	5 分
方案设计创新性	架构合理性	网关整体架构清晰，模块划分合理（路由、限流、缓存解耦）。能够清晰阐述数据在网关中的流转路径。	15 分
	算法先进性	调度算法：有明确的数学模型或逻辑图描述（如加权轮询、最小连接数、或基于预测的动态调度）。 缓存策略：设计了合理的缓存淘汰算法（LRU/LFU）或语义缓存匹配算法。	20 分
技术实现与代码质量	代码规范	代码结构清晰，命名规范，关键逻辑有详细注释。遵循主流编程语言的工程化标准。	5 分
	功能完整性	实现核心的调度与缓存模块接口，即使未接入大规模集群，代码逻辑应完整。	10 分
效果验证	性能水位提升	测试环境：阿里 PPU 单机 / 32K 上下文 测试模型：Qwen27B（可多副本） 测试方法：使用统一的压力测试工具，混合发送 50% 重复请求（考察缓存）和 50% 随机新请求，持续运行 10 分钟。 相较于基线 baseline ● 卓越（30 分） P99 TTFT 提升 30% TPOT 提升 20% ● 进阶（20 分） P99 TTFT 提升 15 ~ 30% TPOT 提升 10 ~ 20%	30 分

		● 基础 (10 分) P99 TTFT 提升 5~15% TPOT 提升<10%	
--	--	--	--

注：Baseline（基线）指不经过智能网关，直接调用后端 vLLM/TensorRT-LLM 服务的原始性能数据。

七、参赛平台

无