An LLM-Based Decision Engine for NearbyOne

December 17, 2024

Name: Sepideh Shamsizadeh
Sending Organization: ZHAW, School of Engineering, Institut für Angewandte Informationstechnologie
Host Organization: Nearby Computing SL
Supervisors:
Host: Maria A. Serrano
Sending: Josef Spillner
Secondment Duration: October 1, 2024 – December 31, 2024

Introducation

The CLOUDSTARS project (HORIZON-MSCA Staff Exchanges) aims to facilitate international research collaboration between academia and industry to tackle emerging technological challenges.
During my secondment at Nearby Computing SL, my research focused on addressing the limitations of Kubernetes orchestration in dynamic edge computing environments. With the rapid growth of IoT devices, resource-constrained environments require intelligent decision-making platforms to minimize latency, optimize bandwidth, and improve system efficiency.
The project led to the development of LLM-Orch, a Large Language Model-based decision engine that enhances workload distribution and resource optimization for Kubernetes-based orchestration. The work involved integrating Prometheus for real-time monitoring and aligning with NearbyOne, Nearby Computing's flagship orchestration platform.

Objectives

The following objectives guided the secondment:

1. Gain familiarity with **NearbyOne** platform architecture and functionalities.
2. Design and develop an **ML-based decision engine** for Kubernetes orchestration.
3. Integrate the solution into the **NearbyOne closed-loop orchestration framework**.
4. Demonstrate the system's improvements in:
    - **Latency reduction**.
    - **Resource optimization**.
    - **Scalability** for edge-cloud deployments.

Activities During Secondment

Task 1: Introduction to NearbyOne

During the initial phase, I focused on gaining an in-depth understanding of the **NearbyOne platform**, its user interface, and overall orchestration workflow. Key activities included:

- Exploring core components such as **resource monitoring systems**, decision-making modules, and integration points for external tools like **Prometheus**.

Challenges: As an external researcher, I encountered difficulties accessing **Prometheus** and **Thanos** for observability, which delayed the initial hands-on testing and workflow validation.

Task 2: ML Model Design and Implementation

I designed and developed an **LLM-driven decision engine** to improve Kubernetes-based orchestration, focusing on the following key functionalities:

- **User Intent Interpretation**: Analyzing and translating natural language inputs into actionable commands for service management.
- **Dynamic Workload Allocation**: Allocating workloads to Kubernetes clusters based on real-time metrics collected via Prometheus.

The system architecture included the development of five specialized **LLM-based agents**:

1. **User Intent Interpreter Agent**: Processes natural language inputs to generate actionable commands.
2. **ServiceOps Agent**: Handles service deployments, deletions, and workload migrations.
3. **Autoscaling Agent**: Monitors system metrics and dynamically adjusts resource allocations.
4. **Platform Data Extraction Agent**: Collects and aggregates system metrics and cluster performance data to support decision-making.
5. **Ranking Agent**: Ranks available Kubernetes clusters based on Prometheus metrics and user intent, providing optimal cluster recommendations for deployments.

Innovative Techniques: The decision engine leveraged **GPT-3.5 Turbo** with customized **prompt engineering** for each agent to ensure precise user intent interpretation and decision-making. By tailoring prompts for specific tasks, we achieved an efficient and context-aware orchestration workflow.

Task 3: Integration into NearbyOne Framework

The final phase focused on integrating the **LLM-Orch engine** into the **NearbyOne orchestration framework**. The main integration activities included:

- Establishing connections with the **Prometheus monitoring system** to enable real-time collection of resource metrics.
- Implementing **dynamic workload scheduling** powered by LLM agents for decision-making.

Validation

We verified the system's functionality by checking the generated **JSON files** for deployment and deletion actions. These files were successfully processed by the **NearbyOne platform**, demonstrating the complete **closed-loop workflow** from user intent to execution. The agents accurately translated user inputs into actionable configurations, ensuring smooth and reliable orchestration.

Outcomes and Achievements

Research Outcomes

- Developed and validated the **LLM-Orch engine**, a Large Language Model-based decision engine to address Kubernetes orchestration challenges in dynamic edge-cloud environments.
- Introduced a modular architecture with specialized agents for user intent interpretation, resource monitoring, and workload optimization.
- Demonstrated a complete **closed-loop workflow** where user intents were successfully translated into actionable JSON files and executed on the **NearbyOne platform**.

Technical Outcomes

- Established seamless integration with the **Prometheus monitoring system** for real-time metrics collection and analysis.
- Developed five LLM-based agents:
    1. **User Intent Interpreter Agent**: Translates natural language inputs into structured JSON commands.
    2. **ServiceOps Agent**: Executes service deployments, deletions, and workload migrations.
    3. **Autoscaling Agent**: Dynamically adjusts resource allocations based on system metrics.
    4. **Platform Data Extraction Agent**: Aggregates and analyzes system metrics to support decision-making.
    5. **Ranking Agent**: Ranks Kubernetes clusters based on Prometheus metrics and user preferences, recommending the most optimal clusters.
- Leveraged **GPT-3.5 Turbo** with **customized prompt engineering** for each agent, ensuring context-aware and efficient orchestration decisions.

Challenges and Solutions

Reflections and Learnings

The secondment experience at **Nearby Computing SL** provided the following key insights:

- Gained hands-on experience in integrating **LLM-based decision engines** into an industrial orchestration platform.

Challenge	Solution
Difficulty accessing **Prometheus** and **Thanos** for observability	Collaborated with the NearbyOne team to acquire access credentials and troubleshoot integration issues.
Limited time for end-to-end testing and performance evaluation	Focused on functional validation by verifying JSON file generation and ensuring platform execution.
Computational overhead of LLM execution in real-time	Optimized GPT-3.5 Turbo prompts to reduce latency and improve processing efficiency.

Table 1: Challenges encountered and corresponding solutions.

- Recognized the importance of real-time observability tools, such as **Prometheus**, for optimizing Kubernetes performance.
- Improved collaboration and problem-solving skills by working in an industrial environment, bridging academic research with real-world applications.

Future Work

The following areas have been identified for future work:

- **Performance Testing and Benchmarking:** Conduct comprehensive testing to evaluate latency, resource optimization, and scalability under varying workloads.
- **Enhanced Resource Efficiency:** Optimize LLM execution to further reduce computational overhead and improve real-time processing.
- **Integration with Additional Tools:** Expand the system to integrate with tools like **Grafana** for advanced visualization and insights.
- **Explainability of Decisions:** Incorporate explainability features to provide insights into LLM-driven decisions, ensuring transparency and trust.

Conclusion

The secondment at **Nearby Computing SL** successfully achieved its objectives. The development of the **LLM-Orch engine** demonstrated the potential of integrating **Large Language Models** into Kubernetes orchestration for dynamic edge-cloud environments. The successful validation of the closed-loop workflow, where user intents were executed seamlessly via the NearbyOne platform, highlights the system's capabilities in intelligent workload distribution and resource optimization.
This work marks a significant step toward scalable, adaptive, and intelligent orchestration solutions. The experience gained from collaborating with Nearby Computing provided

valuable insights into real-world implementation challenges, advancing both my research and practical knowledge.