End-to-end Autonomic Management of Cloud-Edge Continuum Systems using Reinforcement Learning: Design Approach and Formulation

Theodoros Aslanidis, Andreas Chouliaras,
John Byabazaire, Dimitris Chatzopoulos

ENHANCE – HiPEAC 2024 - Munich, Germany

UCD DUBLIN

Funded by the European Union

European Commission | HORIZON EUROPE

MLSysOps

Cloud computing

- **Involves delivering on-demand computing services** including storage, processing power, and applications, over the internet, enabling flexible and scalable access to shared resources.
- **Enables users to access and utilize resources** without the need for physical ownership or maintenance of infrastructure.
- **Provides a centralized platform** for deployment and management of applications, fostering collaboration and efficiency.
- **Limitation:** The escalating number of IoT devices produce massive amounts of data that cloud computing cannot handle.

Edge computing

- A distributed computing paradigm that brings processing and storage capabilities **closer to data sources**.
- Significantly **decreases latency** for real-time applications and IoT systems.
- **Enhances reliability and bandwidth efficiency** by minimizing network distance and transmitting only relevant data.
- Employed in **real-time applications** across various sectors, including IoT, smart cities, manufacturing, healthcare, and transportation.
- **Limitation:** not enough computing power

The cloud-edge continuum

- A **hybrid model** that emerges to **overcome the limitations** and challenges opposed by the previous models.
- Combines **robust computing power** and management capabilities of cloud computing and **real-time data processing** of edge computing.
- Includes the spectrum of computing resources from **centralized cloud data centers** to **decentralized edge devices.**
- Ideal for scenarios requiring a **balance** between **centralized control** and **local processing** such as versatile applications in IoT, cyber-physical systems, smart cities, healthcare, and industries demanding real-time responsiveness.

ML-assisted cloud-edge continuum

- To **enhance adaptability**, ML techniques can be incorporated into the cloud-edge continuum.
- The ML is responsible **managing** and **controlling** the dynamic and heterogeneous nature of the cloud-edge continuum.
- This helps **optimizing decision-making** and **improving overall efficiency** in managing its dynamic and heterogeneous nature for **optimal performance** and **responsiveness**.

The outer picture

- The system, viewed externally, operates as an AI-enabled entity
- Makes complex decisions to strike a balance between system or application metrics such as:
  - overall performance
  - resource utilization
  - energy consumption

Under the hood

- Internally, this whole system will be a conventional resource management framework with **AI-assisted decision-making**.
- Depending on whether the decision is of high importance, this can be a rule-based or heuristic approach or ask AI to make a more informative decision.
- Different ML models can assist in different types of decisions.
- The AI-assistant can be switched on and off by a system administrator.

Three-layer hierarchical agent structure

- Due to the high **heterogeneity** and the **dynamic** nature of the cloud-edge continuum, we propose a **three-level** agent hierarchy.
  - Provide scalability, adaptability, and efficient resource utilization.
  - Enables specialized agents at each level to handle specific tasks, from fine-grained resource control at the node level to strategic deployment decisions at the continuum level.
- Each agent at every level has access to a diverse set of **ML models**, enhancing decision-making capabilities by incorporating machine learning techniques tailored to specific objectives.

Node-level agent responsibilities

- **Device selection (acceleration):**
  - CPU, GPU or FPGA
- **Frequency selection:**
  - Select the optimal CPU or GPU frequency
- **Other selections:**
  - affinity to cores,
  - RAM,
  - storage

Cluster-level agent responsibilities

- **Node allocation:**
  - Allocate nodes for running application components.
  - Determine connectivity between nodes.
- **Resource distribution:**
  - Split resource budget among nodes based on application requirements.
- **Resource optimization:**
  - Optimize resource utilization to achieve desired objectives.
- **Load balancing within cluster:**
  - Plan migration of application components for cluster-level load balancing.

Continuum-level agent responsibilities

- **Application Deployment:**
  - Determine where applications should be deployed within the continuum.
- **Objective Switching:**
  - Dynamically switch among different objectives to meet system goals.
- **Efficiency Optimization:**
  - Strive for efficient resource usage while meeting application requirements.
- **System administrator interaction:**
  - Receive signals from the system administrator

Extra responsibilities of all agent levels

- **ML Model Selection:**
  - Select the appropriate ML model based on the higher-level command and system optimization metric/objective.
- **ML Retraining Process:**
  - Inject the retraining process of an ML model into the system.
- **Anomaly Detection and Trust Evaluation:**
  - Control and trigger anomaly detection and trust evaluation mechanisms.
- **Communicate with agents of other levels**
  - Propagate higher-level objectives to lower-level agents.
  - Give feedback of current internal state to higher-level agents.
- **Logging:**
  - Keep a log of actions with timestamps.

Using RL techniques

- **Why?**
  - Handling non-periodic and complex user patterns.
  - Adapts to dynamic and heterogeneous environments.
  - Providing long-term, strategic decision-making and planning.

- **Exploring RL Paradigms:**
  - Hierarchical RL
  - Multi-agent RL
  - Multi-objective RL

Challenges

- **RL agent actions design:**
  - Crafting actions that align with system objectives and application requirements.
- **Telemetry data granularity:**
  - Determining the level of detail in telemetry data to capture system dynamics accurately.
- **Dynamic objective switching:**
  - Implementing the ability to seamlessly switch among diverse system objectives.
- **Global-to-local objective translation:**
  - Translating global objectives into specific, low-level system actions.
- **Learning Feasibility and Robustness:**
  - Ensuring model learning processes are not only feasible but also robust and efficient.

Thank you!

Any questions?

Feel free to contact us at:

theodoros.aslanidis@ucdconnect.ie