Collection and Integration of Application Telemetry for Cloud-Edge Distributed Systems

Proposal

Orchestration decisions in distributed systems depend not only on network characteristics, but also on application behaviour (request volume between services, CPU/memory usage, response latency, workload variability, and temporal patterns such as peaks, daily cycles, or bursty behaviour). These factors directly influence choices such as service placement, the number of replicas, or the need for migrations between cloud and edge nodes.

In the state of the art, real workload traces and application telemetry from clusters and microservice benchmarks exist, but these data are often isolated, collected using different metrics, heterogeneous formats, and non-uniform levels of detail. This makes it difficult to systematically reuse them for studies on distributed systems, orchestration, and resource management.

The goal of this thesis is to collect, analyse, and integrate real application telemetry by building a coherent, clean, and structured dataset for the study of cloud-edge distributed systems. The resulting dataset must represent services, application instances, requests, usage metrics, and response times in a homogeneous way, so that it can be used as a basis for simulations, resource management studies, and the evaluation of orchestration techniques.

The work includes the analysis of public benchmarks and traces of distributed applications, the definition of a unified data model to describe services and metrics, data cleaning (handling missing values, units of measurement, and formats), and temporal alignment across different sources. A part of the thesis work will be dedicated to the characterization of application patterns, for example by studying workload variability over time, the relationship between resource usage and latency, and the presence of peaks or anomalous behaviours. The result will be a documented and reusable dataset, accompanied by an analysis describing its main properties.

Tools and Technologies

- Python
- Pandas / NumPy
- Microservice benchmarks and public traces
- Matplotlib / Seaborn
- JSON / CSV / Parquet

Objectives

1. Analyse sources of application telemetry data.
2. Design a unified data schema for services and metrics.
3. Implement integration and cleaning pipelines.
4. Study workload patterns and resource usage.
5. Produce a documented and reusable dataset for distributed systems.
6. Demonstrate a concrete use case that uses the produced dataset.

Prerequisites

- Good knowledge of Python.
- Basics of operating systems and distributed systems.
- Interest in data analysis and cloud-edge computing.