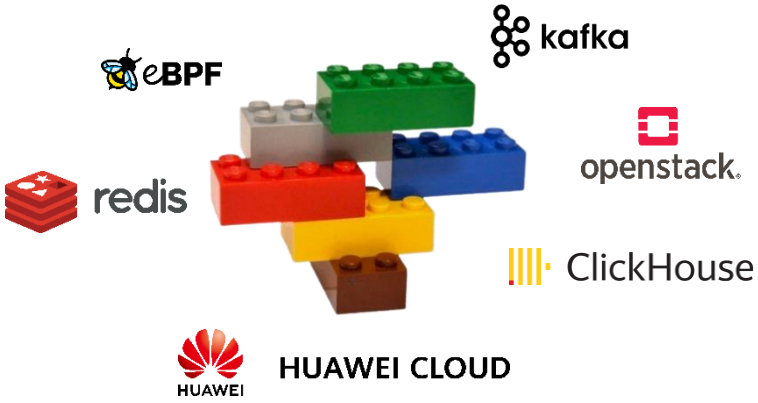
HUAWEI **Munich Research Center**

Non-intrusive Distributed Tracing for Cloud

(MSc/PhD Thesis/Internship)



Problem. Distributed tracing is an influential technique for understanding the flow of requests through a complex system composed of microservices. Nonetheless, for many systems it is not possible to instrument the code in each service to generate trace data. For example, it is typically difficult to instrument frontends, to asynchronous middleware (e.g., Redis and Kafka), and to backends. Thus, new emerging approaches use non-intrusive monitoring to collect metrics and data without modifying the existing system.

Objective. The objective of this project is to trace user requests made to OpenStack without relying on distributed tracing frameworks such as OSProfiler (a library that provides a tracing and profiling).

Approach. Built a distributed tracing tool based on eBPF (extended Berkeley Packet Filter) probes and BCC (BPF Compiler Collection) to monitor and store service calls in kernel space using BPF maps. Services include OpenStack components (e.g., Nova, Glance, Keystone) and middleware components (e.g., Kafka, Redis). In user space, monitoring data from BPF maps is accessed by Python scripts using the BCC API and stored in ClickHouse (an open-source columnar db for high-performance analytical processing). To reconstruct full-link traces, we correlate individual samples using spatial, temporal, and content-based data. To reduce computational overhead, we use sketching techniques to compress large datasets and capture key statistical and structural information about traces.

Results. (expected) When HTTP user requests are executed, the tool will automatically reconstruct distributed traces. The computational overhead of the tool should be <1.0% and obtain accurate tracing information. The evaluation will be conducted using Tempest, a testing framework used in the OpenStack community to validate the functionality, performance, and behavior of OpenStack services.