Sampling Strategies in Distributed Tracing

Background
Distributed tracing systems are becoming the solution of choice in Microservices to analyze performance and improve debugging. A trace represents an end-to-end view on a set of correlated requests, as they happen from a client-based call of an API to finally accessing a database. As tracing systems are designed to impose low overhead, they rely on sampling to reduce the amount of traces generated.

Recommended read:

Problem
Sampling impacts the quality of analyses based on the number of collected traces. A low sampling rate may hide the occurrence and relevance of issues, while a high sampling rate may be too costly in terms of performance, especially when downstream services “fan out” requests. The choice of an appropriate sampling strategy is a complex problem and typically done based on experience by the involved personnel.

Goals
The student is to analyze sampling strategies and mechanisms for the runtime adaptation of latter in state of the art tracing systems. Either a systematic categorization and practical evaluation or the exploration and prototypical development of new self-adapting strategies, based on ML approaches, can be pursued, depending on the personal interest of the student.

Required Skills:
- Knowledge in distributed systems, Microservices
- Good ability to read and understand code
- Fundamental knowledge of statistical methods
- (Optional): Knowledge in Machine Learning approaches

Contact: Dominik Ernst
de@ise.tu-berlin.de

Our Mission:
Our lectures cover fundamental methods and techniques in the areas of service computing, cloud computing, and enterprise computing. We like to engage students in hands-on building of distributed information systems and to take an interdisciplinary approach to evaluating such systems. Through a close mentoring of students, especially in our seminars, we aim to introduce students to our ongoing research and to excite them to do future studies and research with us.