[Extended Abstract] Predicting Performance Degradations of Black-box Micro-service Applications

Martin Sträßer
martin.strasser@stud-mail.uni-wuerzburg.de
University of Wuerzburg, Germany

Johannes Grohmann
johannes.grohmann@uni-wuerzburg.de
University of Wuerzburg, Germany

In recent years, a clear trend towards the development of micro-service applications has evolved. This simplifies the development process and offers several benefits in operation and maintenance. Concurrently, new challenges and opportunities arise. As performance degradations of these applications are still a main contributing factor to user dissatisfaction, our goal is to utilize micro-service applications for the prediction of performance degradations.

Most state-of-the-art approaches do not specifically focus on micro-service architectures and therefore are not able to utilize the implicit performance properties (e.g., statelessness) and lag behind in accuracy and explainability. Other approaches are only capable of detecting current performance degradations, without the ability to predict and anticipate future performance degradations.However, approaches that focus on micro-service applications usually require prior knowledge about the application architecture, have a limited application scope, or offer only limited explainability of their results.

In this talk, we present our novel approach for modeling and predicting performance degradations of micro-service applications. We utilize the performance dependencies as well as the statelessness of individual services as two main properties of our approach. Due to the statelessness of any single service, we identify the number and type of incoming requests as the main cause of performance degradations at the service level. Therefore, we use modern time series forecasting techniques to estimate the future of user requests. Based on the incoming user requests, we utilize the Request Propagation Model (RPM) to predict the type and amount of requests for each individual service. The connections and dependencies between several micro-services are extracted using architectural information from monitoring and tracing data at runtime. Therefore, the Request Propagation Model in conjunction with a forecasting engine is able to predict arrival rates for all micro-service endpoints.

At the same time, we gather response times and other performance metrics in response to a specific experienced load of each micro-service at runtime. This data serves as a basis for training the Performance Inference Model, a machine-learning-based regression model, which calculates a performance prediction from estimated load intensities and architectural information. The resulting performance predictions can then be aggregated using a Performance Propagation Model, leading to an efficient and accurate pinpointing of performance problems in complex micro-service applications.

In this talk, we describe the theoretical aspects, present a generic architecture for realization, and present our concrete open-source implementation of the envisioned approach. Additionally, we present preliminary results from two state-of-the-art micro-service test applications (TeaStore and TrainTicket). We define and perform measurements in five test scenarios, in which different application states and degradation sources are simulated. The results show that the models are able to learn the performance behavior and architectural dependencies of the applications quickly and without prior knowledge. Depending on the test scenario, up to 72% of the measured performance degradations are predicted correctly. Moreover, the models are able to keep their prediction quality nearly constant even with higher prediction horizons.

The presented approach is application-agnostic, does not need prior application knowledge, produces comprehensible and explainable results, and can be easily extended. A weakness to be discussed is the current strong dependence on the load forecast. We hope that the discussion will help to mitigate weaknesses or improve the performance of our proof-of-concept prototype.