SCALENDAR: Predictive Autoscaling Engine for Microservice Applications in Cloud

1st Vladimir Podolskiy
Technical University of Munich

2nd Anshul Jindal
Technical University of Munich

3rd Yesika Ramirez
Technical University of Munich

4th Atakan Yenel
Technical University of Munich

5th Saifeldin Ahmed
Technical University of Munich

6th Michael Gerndt
Technical University of Munich

SCALENDAR is a distributed predictive autoscaling engine for microservice applications in cloud. The name is the overlap of two words - *scale* and *calendar*, which underlines that the scaling actions are scheduled to be performed in the future according to the ”calendar”.

The aim of SCALENDAR is to match the estimated future demand for the microservice web-application as measured in requests per second (RPS) with the capacity of the application so that the number of violations of service level objectives (SLOs) is mitigated and the resource usage cost is minimized. Thus, in comparison to existing predictive autoscaling solutions which are resource-centric, SCALENDAR is a service-centric system as it bases the scaling actions on service-level metrics such as RPS.

SCALENDAR operates both on the level of virtual machines (VM) and on the level of application containers. It incorporates the following functional blocks:

- **RPS Forecasting** block is responsible for the derivation of the future demand model in terms of RPS based on the historical monitoring data. This block supports multiple forecasting approaches for time series, such as ARIMA/GARCH, Support Vector Regression, Exponential Smoothing, Singular Spectrum Analysis. The derived models are estimated using interval accuracy score; best models are used to estimate the future demand.
- **Capacity Modeling** block is responsible for determining the capacity of single microservice instance on a given VM type in terms of RPS. Maximal number of RPS that microservice can handle with a) successful response rate staying above some threshold and b) successful response time staying below some threshold is considered as *microservice capacity (MSC)*. MSC for each individual microservice is determined via sandboxing. A model that relates MSC for a given microservice to the deployment parameters (e.g. VM parameters) is built only for the limited set of options. MSC for other deployments are derived using the regression.
- **Structural Modeling & Capacity Balancing** block is responsible for constructing the structural model of the microservice application in form of oriented graph and its further analysis. The nodes of the graph represent microservices with their capacities, whereas edges represent logical connections between them. The analysis of the application graph determines smallest groups of connected services that could be balanced in terms of RPS and balances the application by replicating these groups according to the predicted demand. The goal of the balancing is to increase the throughput of the application in terms of RPS.
- **Scaling Schedule Derivation** block is responsible for combining the results of the modeling to get the schedule of scaling actions that would allow to match the application capacity with the forecasted demand. Schedule of future scaling actions, such as addition/removal of VMs and pods, is generated according to one of the policies: Naive, Best resource pair, Only-Delta-load, Always-resize, and Resize when beneficial. These strategies differ by the allowed scaling actions. For example, Only-Delta-load allows to increase or reduce the number of pods and VMs without changing their type which allows it to closely follow the load pattern but leads to high VM clusters fragmentation and to poor long-term manageability.
- **Scaling Execution** block is responsible for timely execution of scaling actions according to the derived scaling policy. Reactive autoscaling action invokes schedule invalidation; this results in adjustment of models.

The implementation of such a system is possible only under some limitations. Current design of SCALENDAR is limited to microservice web applications with homogeneous workload. Models employed in SCALENDAR are limited by the capacity of the application expressed in terms of RPS.