Auto-Scaling with Apprenticeship Learning

Kamal Hakimzadeh
KTH - Royal Institute of Technology
Stockholm, Sweden
mahh@kth.se

Patrick K. Nicholson
Nokia Bell Labs
Dublin, Ireland
pat.nicholson@nokia-bell-labs.com

Diego Lugones
Nokia Bell Labs
Dublin, Ireland
diego.lugones@nokia-bell-labs.com

ACM Reference Format:
Kamal Hakimzadeh, Patrick K. Nicholson, and Diego Lugones. 2018. Auto-Scaling with Apprenticeship Learning. In *SoCC '18: ACM Symposium on Cloud Computing, October 11–13, 2018, Carlsbad, CA, USA.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3267809.3275454

Current practices for provisioning and autoscaling applications in cloud are limited and require expert knowledge to deliver acceptable quality of service (QoS). The complexity of the cloud stack requires an iterative process to tune several parameters with robustness to varying workloads, software or hardware upgrades, failures, resource interference, etc. Usually, this process is performed by humans, which limits the agility to adapt to stack changes. The need for human expertise relates to the fact that different applications can have different and highly specific *key performance indicators (KPI's)*. Experts can configure provisioning and autoscaling actions based on application KPIs but that process needs to be repeated for the various combinations of applications, services, and platforms. That is, the KPI-based rules for one platform do not necessarily apply to other environments given the multiplicity of variables that can affect such rules.

Recently, several proposals have included machine learning to remove some of the complexity and human intervention. Still, these techniques require significant amounts of data (e.g., in case of techniques based on deep learning), or a considerable number of iterations to converge to the appropriate rules (e.g., in case of techniques based on reinforcement learning). These limitations can be prohibitive in cloud because 1) the amount of data required can be difficult to obtain, particularly in multi-vendor environments with management policies and specific infrastructure conditions that limit data access, and 2) the number of parameters (or dimensions) to explore can be be unfeasible, even for simple applications.

To overcome these problems, we propose a solution that differentiate from others in two major aspects. First, our proposal is generic and portable to different environments, and robust to changes in the stack. We achieve this by training a model that (indirectly) requires application specific KPIs **one time**: during the training phase. However, when deployed, the model only requires *platform utilization metrics* that are common to all infrastructures (e.g., CPU, Memory, IO, etc.). Second, we reduce the data requirements and exploration space by using *apprenticeship learning* (also called learning from demonstration) which makes use of an expert to demonstrate the task the model needs to perform. Thus, our approach is to learn an autoscaling function, called the *trainee*, based on training data collected by observing an autoscaling expert in a well-known setting. The expert uses KPIs combined with online saturation detection techniques to precisely avoid performance degradation. The trainee, however, learns the expert's behaviour from platform utilization metrics only, with the aim of capturing complex resource bottlenecks as the expert takes autoscaling actions. We find that, even with a small number of *trajectories*, i.e., specific causal action/decision activities of the expert, the trainee is highly accurate at mimicing the expert behaviour.

Our approach leverages the expert's behaviour on a given application to create a generic trainee that can be shipped to other platforms and also applied to applications with similar resource bottlenecks (e.g., CPU-bound applications, or memory-bound).

We have implemented our proposal in a container-based environment. Our scalable testing application is Apache Cassandra, a widely used distributed and column-based database, and Docker Swarm for orchestrating containers. As proof of concept, our system collects KPIs and platform utilization metrics using Google's cAdvisor and a Prometheus time-series database. We stress the application with the Yahoo Cloud Serving Benchmark (YCSB) that generates synthetic traffic profiles, as well as realistic workloads from traces.

In our evaluation, we use an expert that can cope with various workload scenarios. The usage of the expert is based on performance characterizations that are looped-back to the expert for future auto-scaling decisions. Prior to learning, we use techniques as oversampling or grid search cross-validation to improve the accuracy of the trainee. We use two evaluation methodologies: (i) machine learning validation techniques in order to measure the accuracy of the knowledge transferred from the expert to the trainee, and; (ii) available elasticity benchmarks to evaluate both the expert and the trainee ability to scale without KPI degradation. Our initial results, in small scale, CPU-bound scenarios, with read only Cassandra workloads and single container sizes, show that the knowledge transfer has an accuracy between 80% and 90%. In the short term, we plan to test the trainee against several different unseen workload, applications, and platforms.

We summarize our contribution as follows:

- A novel autoscaling solution based on apprenticeship learning that is agnostic to infrastructure and workloads.
- The autoscaling function is based on platform utilization metrics, but is able to yield similar performance to KPI-based autoscaling and is generic and portable.
- A recursive process for improving the knowledge-base of a learner by adding new dimensions in the trajectories.

SoCC '18, October 11–13, 2018, Carlsbad, CA, USA

ACM ISBN 978-1-4503-6011-1/18/10.
https://doi.org/10.1145/3267809.3275454