Autoscaling Performance Measurement Tool

Anshul Jindal
Technical University of Munich
Garching (near Munich), Bavaria
Germany
anshul.jindal@tum.de

Vladimir Podolskiy
Technical University of Munich
Garching (near Munich), Bavaria
Germany
v.podolskiy@tum.de

Michael Gerndt
Technical University of Munich
Garching (near Munich), Bavaria
Germany
gerndt@in.tum.de

ABSTRACT

More companies are shifting focus to adding more layers of virtualization for their cloud applications thus increasing the flexibility in development, deployment and management of applications. Increase in the number of layers can result in additional overhead during autoscaling and also in coordination issues while layers may use the same resources while managed by different software. In order to capture these multilayered autoscaling performance issues, an Autoscaling Performance Measurement Tool (APMT) was developed. This tool evaluates the performance of cloud autoscaling solutions and combinations thereof for varying types of load patterns. In the paper, we highlight the architecture of the tool and its configuration. An autoscaling behavior for major IaaS providers with Kubernetes pods as the second layer of virtualization is illustrated using the data collected by APMT.

CCS CONCEPTS

• **Computer systems organization → Cloud computing**;

KEYWORDS

Autoscaling, Autoscaling performance analysis, Cloud, IaaS

ACM Reference Format:
Anshul Jindal, Vladimir Podolskiy, and Michael Gerndt. 2018. Autoscaling Performance Measurement Tool. In *ICPE '18: ACM/SPEC International Conference on Performance Engineering Companion , April 9–13, 2018, Berlin, Germany*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3185768.3186293

1 INTRODUCTION

The dynamic change of the user demand on cloud applications combined with high Quality of Service (QoS) requirements yields the necessity for applications to be highly scalable. All major IaaS providers deliver the facilities for scaling the resources of cloud applications. Such facilities are known under the name of autoscaling solutions. They add or remove virtual machine (VM) instances in response to the change in some monitored metric. The administrator defines a set of rules to add or remove VM instances with appropriate thresholds. In general, autoscaling solutions allow to make cloud applications scalable. However, autoscaling solutions could reasonably differ in terms of performance.

While there exist metrics and techniques to determine the performance of autoscaling solutions [1, 2], there is no tool to compare the performance of different autoscaling solutions for various load patterns. In this paper, such a tool is presented. APMT allows to evaluate the performance of multilayered autoscaling solutions combining Amazon, Microsoft and Google infrastructure autoscaling with Kubernetes horizontal scaling of pods on top. In this work, the use of Kubernetes is reasoned by its introduction of the additional virtualization level of pods and its growing use in the industry. The user of APMT can specify different deployment and scaling parameters as well as set up specific load patterns. Request latency and number of errors are the metrics used for the performance evaluation. The flexibility provided by APMT allows the user to extensively study the performance of a specific autoscaling solution and compare several solutions under the same conditions.

2 TOOL ARCHITECTURE

APMT is a microservice architecture-based web-application used to test different autoscaling solutions and their combinations under different load patterns. The architecture of APMT is shown in Fig. 1.

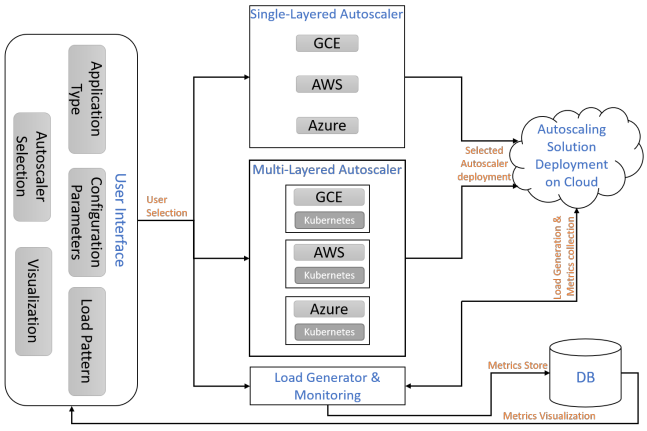

Figure 1: High-level architecture of the Autoscaling Performance Measurement Tool.

APMT provides an automatic deployment procedure for different cloud services providers (CSPs) native autoscaling solutions referred to as the single-layered autoscaling solutions as well as for the combination of the CSPs native autoscalers with the Kubernetes PaaS horizontal pod autoscaling referred as the multi-layered

ICPE '18, April 9–13, 2018, Berlin, Germany

ACM ISBN 978-1-4503-5629-9/18/04…$15.00
https://doi.org/10.1145/3185768.3186293

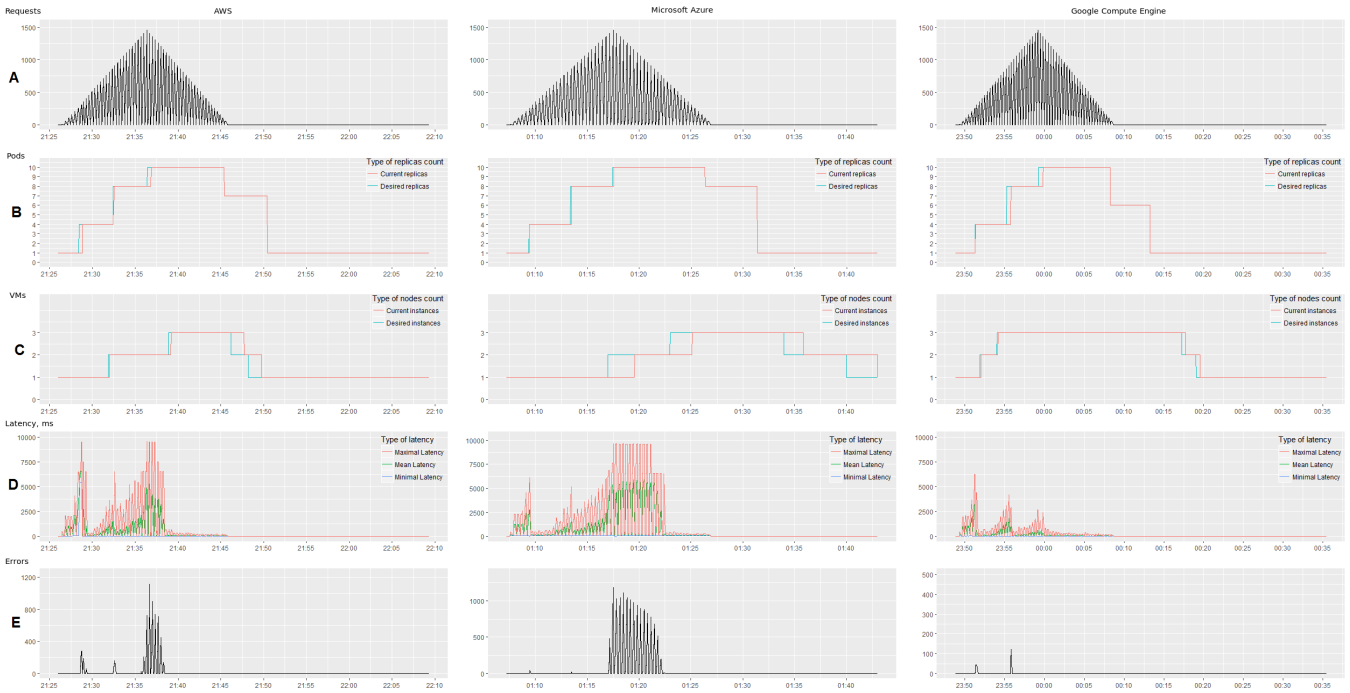

Figure 2: Comparison of IaaS autoscaling solutions (and Kubernetes horizontal scaling of pods). Rows: A) Total number of requests sent; B) Desired amount of instances (DAI) and Current amount of instances (CAI) of Kubernetes pods; C) DAI and CAI of VM instances; D) Minimal, Mean, and Maximal latency; E) Number of errors.

autoscaling solutions. APMT supports autoscaling solutions of 3 CSPs: Google (GCE), Microsoft (Azure), and Amazon (AWS).

The user has the option to deploy the desired type of application - CPU intensive, I/O intensive, high memory usage application on the cloud using any of the autoscaling solutions. Example applications of these types are included into the tool distribution. The autoscaling solutions are configured with the different parameters: type of instance, min. and max. number of instances, scaling decision metric (with its threshold), and autoscaling policy. After the deployment, the load generator which is integrated as part of the tool is used to test the performance of these solutions on different load patterns: linearly increasing, linearly increasing and becoming constant after specific time, triangle, and random. The source code of APMT is accesible via the link: **github.com/ansjin/APMT**

3 APMT USE-CASE

This section presents the test results for each supported multilayered autoscaling solution for the triangle load pattern. The triangle load pattern corresponds to the linearly increasing amount of requests per second which then linearly decreases. The load is generated by 50 concurrently running clients.

The main autoscaling characteristics collected by APMT are depicted in Fig. 2: autoscaling time for IaaS autoscaling and Kubernetes horizontal pods autoscaling (see rows B and C), performance characteristics (latencies in row D and number of error responses in row E). Rows D and E in Fig. 2 clearly highlight the multilayered solution with the best performance - GCE/Kubernetes combination which has the smallest amount of errors and smallest latency. As we can see in the plot C-3, such a performance is achieved through the overprovisioning of VMs. For AWS and Azure we see higher latencies and number of errors, this implies the necessity to adjust the scaling policies. Moreover, rows B and C for Azure demonstrate problems with the autoscaling synchronization between layers: new pods are created faster than new VMs are deployed.

4 CONCLUSIONS AND FUTURE RESEARCH

APMT provides flexibility for experiments settings: the user may choose an own test application, autoscaling solution or combination thereof, load pattern under investigation, etc. The processing and the interpretation of the collected data may differ dependent on the technique used to process these data.

The following APMT extensions are considered: 1) support for real world-like load patterns; 2) introduction of additional metrics; 3) autoscaling performance evaluation for heterogeneous multi-cloud infrastructure; 4) support for pluggable autoscaling policies.

REFERENCES

[1] Alexandros Evangelidis, David Parker, and Rami Bahsoon. 2017. Performance Modelling and Verification of Cloud-based Auto-Scaling Policies. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '17)*. IEEE Press, Piscataway, NJ, USA, 355–364. https://doi.org/10.1109/CCGRID.2017.39

[2] Alexey Ilyushkin, Ahmed Ali-Eldin, Nikolas Herbst, Alessandro V. Papadopoulos, Bogdan Ghit, Dick Epema, and Alexandru Iosup. 2017. An Experimental Performance Evaluation of Autoscaling Policies for Complex Workflows. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering (ICPE '17)*. ACM, New York, NY, USA, 75–86. https://doi.org/10.1145/3030207.3030214