PRACTITIONERS' DIGEST

Editor: **Jeffrey C. Carver**
University of Alabama
carver@cs.ua.edu

(Research) Insights for Serverless Application Engineering

Jeffrey C. Carver, Birgit Penzenstadler, Joel Scheuner, and Miroslaw Staron

FOLLOWING ALONG WITH the theme of this special issue of *IEEE Software*, the "Practitioners' Digest" department reports on papers about serverless application engineering from *Journal of Systems and Software*, the 2020 European Conference on Software Architecture, and the 19th International Conference on Middleware. Feedback or suggestions are welcome. In addition, if you try or adopt any of the practices included in the column, please send me and the authors of the paper(s) a note about your experiences.

Evaluating the Performance of Function as a Service

"Function-as-a-Service Performance Evaluation: A Multivocal Literature Review" by Joel Scheuner and Philipp Leitner[1] describes 112 studies that report the empirical evaluation of the performance of function-as-a-service (FaaS) platforms. FaaS is a form of serverless cloud computing and is defined as FaaS platforms [e.g., Amazon Web Services (AWS) Lambda] executing event-triggered code snippets (i.e., functions). Individual studies identify some performance challenges with FaaS platforms, such as cold starts that cause multiple-second execution delays. This paper consolidates the results from 61 industrial and 51 academic performance studies and provides actionable recommendations on reproducible FaaS experimentation. It also offers an interactive online appendix to allow readers to systematically discover relevant performance studies, including pointers to source code and data sets.

The key findings show that existing studies primarily evaluate the AWS Lambda FaaS platform and focus on microbenchmarks using simple functions to measure CPU speed and platform overhead (i.e., cold starts). The industrial studies focus primarily on testing the runtime of different languages, e.g., comparing Java to Node.js. The main results are that 1) existing performance results and tools can guide language choice and 2) there are no cross-platform application benchmarks with realistic workloads, so developers must themselves test applications in different clouds to make an optimal decision. In addition, a reproducibility analysis found that one-third of the studies are practically impossible or difficult to replicate under reasonably similar conditions. These studies neither provide an appropriate description of their experimental setup nor any technical artifacts. Future FaaS performance studies need to address such flaws threatening their reproducibility. The main takeaways from this paper are 1) publish the performance data in addition to the benchmark code to foster discussion and comparability with related performance testing efforts and 2) stop reporting mean values exclusively, but use appropriate statistical tools for nonnormally distributed performance data, such as cumulative distribution functions or percentiles. This paper appears in *Journal of Systems and Software*. Access it at http://bit.ly/PD-2021-Jan-01.

Brokers for Internet of Things Edge Computing

"A Comparison of MQTT Brokers for Distributed IoT Edge Computing"

Digital Object Identifier 10.1109/MS.2020.3028659
Date of current version: 22 December 2020

0740-7459/21©2021IEEE

by Sten Grüner and Julius Rückert[2] describes a study comparing three Message Queuing Telemetry Transport (MQTT) brokers. Industrial devices use the MQTT protocol to enable data collection for connected devices, known as *Internet of Things* (*IoT*) devices. In the IoT, clustered MQTT brokers may reside on the edge of the cloud computing clusters and help address scalability and performance problems. The study compared three clustered MQTT brokers: EMQX, HiveMQ, and VerneMQ. The key results from the study are that 1) the EMQX broker showed the highest stable throughput, 2) the HiveMQ broker achieved the best availability/resilience and is the most extensible, and 3) the VerneMQ broker created new instances the fastest (i.e., has high horizontal scalability). Because all three brokers are used in industrial applications (e.g., Bavarian Motor Works uses HiveMQ), the results from this study, although conducted via an experiment, are relevant and interesting for software engineers who are integrating IoT applications. In addition to these findings, the paper provides guidance for architecture decisions to help software architects in choosing which broker and which MQTT configuration should be adopted. This paper appears in the 2020 European Conference on Software Architecture. Access the paper at http://bit.ly/PD-2021-Jan-02.

Industrial devices use the MQTT protocol to enable data collection for connected devices, known as *Internet of Things* devices.

Decentralized Architecture to Support Sustainability

"Decentralized Architecture for Energy-Aware Service Assembly" by Mauro Caporuscio, Mirko D'Angelo, Vincenzo Grassi, and Raffaela Mirandola[3] shows that sustainability of computer systems is important and has gained increased attention in the last decade. The paper reports on how to design decentralized software architecture to support sustainability. The contribution of the paper is its description of a distributed architecture, which combines the collective fulfillment of quality of service and energy requirements. As the authors state:

> *On the one side, we measure the achievement of some average global system 'quality,' thanks to the contribution of all services. On the other side, we measure whether there is an unbalanced distribution among services of this global quality.*

This model computes energy consumption by summing computational energy and communication energy, providing an advance over other energy-blind models. Furthermore, the model takes into account nonfunctional properties of the communication networks, such as bandwidth and latency (which allow the determination of the cost of computational offloading). Laboratory studies show that it is possible to balance performance, energy consumption, and the distribution/decentralization of the system. This model is important in practice because it allows developers to construct systems that balance energy and quality, which is important for the IoT and distributed, service-based software. This paper appears in the 2020 European Conference on Software Architecture. Access the paper at http://bit.ly/PD-2021-Jan-03.

Serverless Data Analytics

"Serverless Data Analytics in the IBM Cloud" by Josep Sampé, Gil Vernik, Marc Sánchez-Artigas, and Pedro García-López[4] describes IBM-PyWren, a framework for implementing FaaS (function as a service) in the IBM cloud environment. One of the core principles for IBM-PyWren is simplicity, which helps both expert and novice cloud programmers. In comparison to the standard PyWren implementation, the framework provides functionality, such as dynamic composition of functions, automatic data partitioning, and broader support for map-reduce algorithms. The framework is based on Docker containers for higher flexibility and is compatible with Apache OpenWhisk. The core concept of IBM-PyWren is its executor, which gives programmers the ability to execute functions in the environment—both built-in and custom functions. The novel features enable IBM-PyWren to compose functions programmatically rather than declaratively. The paper also presents the results of a laboratory evaluation of IBM-PyWren. The results show that IBM-PyWren is sufficiently elastic to support workloads that are similar to those seen in practice. The experiments also showed a large improvement when using a parallel FaaS approach over a sequential one [with speedups ranging from 10× to 135× (when using 47 and 923 parallel functions)]. Since the publication of this article, the developers of IBM-PyWren have rebranded it Lithops [(Lightweight Optimized Processing) https://github.com/lithops-cloud/lithops] and included support for multiple providers (e.g., AWS, Google Cloud, and Azure) and environments (e.g., Knative). This paper appears in the 19th International Middleware Conference (2018). Access the paper at http://bit.ly/PD-2021-Jan-04.

References

1. J. Scheuner and P. Leitner, "Function-as-a-service performance evaluation: A multivocal literature review," *J. Syst. Softw.*, vol. 170, p. 110708, Dec. 2020. [Online]. Available: http://bit.ly/PD-2021-Jan-01
2. H. Koziolek, S. Grüner, and J. Rückert. (2020). "A comparison of MQTT brokers for distributed IoT edge computing." Presented at Eur. Conf. Softw. Archit. [Online]. Available: http://bit.ly/PD-2021-Jan-02
3. M. Caporuscio, M. D'Angelo, V. Grassi, and R. Mirandola. (2020). "Decentralized architecture for energy-aware service assembly." Presented at Eur. Conf. Softw. Archit. [Online]. Available: http://bit.ly/PD-2021-Jan-03
4. J. Sampé, G. Vernik, M. Sánchez-Artigas, and P. García-López. (2018). "Serverless data analytics in the IBM Cloud." Presented at Middleware '18: Proc. 19th Int. Middleware Conf. Ind. [Online]. Available: http://bit.ly/PD-2021-Jan-04

ABOUT THE AUTHORS

JEFFREY C. CARVER is a professor in the University of Alabama's Department of Computer Science, Tuscaloosa, Alabama, USA. Further information about him can be found at http://carver.cs.ua.edu. Contact him at carver@cs.ua.edu.

BIRGIT PENZENSTADLER is an assistant professor at Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden. Further information about her can be found at https://www.chalmers.se/en/Staff/Pages/birgit.aspx. Contact her at birgitp@chalmers.se.

JOEL SCHEUNER is a Ph.D. student in the joint division of software engineering at Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden. Further information about him can be found at https://joelscheuner.com/. Contact him at scheuner@chalmers.se.

MIROSLAW STARON is a professor in the Software Engineering Division, Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden. Contact him at miroslaw.staron@cse.gu.se.