ADDRESSING PERFORMANCE CHALLENGES IN SERVERLESS COMPUTING

M PUJITHA 1, BUCHI REDDY CHINTAKINDI 2, PAMBALA NAGESWARA RAO 3
4.J.SHARATH, 5.K.MOUNIKA
Assistant Professor, Department of Computer Engineering,
Ellenki college of Engineering and Technonlogy, patelguda (vi), near BHEL ameenpur (m),
Sangareddy Dist. Telangana 502319

.

ABSTRACT

As a key part of the emerging serverless computing paradigm, Function-as-a-Service (FaaS) platforms allow users to run arbitrary functions without being concerned about operational issues. How- ever, there are several (performance-related) challenges surround- ing the state-of-the-art FaaS platforms that can hinder widespread adoption of FaaS, including size-able overheads, unreliable perfor- mance, and the absence of benchmarks. In this work provide an overview of our ongoing work towards addressing these challenges. Thus far, we have been building a workflow engine, Fission Work- flows, to provide efficient function composition within the FaaS model. Furthermore, together with the SPEC CLOUD RG, we have been working on solving more high-level challenges; defining a common terminology and reference architecture, as well as devel- opinga industry-wide benchmark of these FaaS platforms. Besides presenting our work so far, we also propose a roadmap towards solving these challenges to invite researchers to join this effort.

KEYWORDS

Function-as-a-Service, Serverless Computing, Performance evalua- tion, Workflows

1 INTRODUCTION

Software-development paradigms are increasingly transitioning from monolithic applications to compositions of smaller services that are more granular and distributed (e.g., through Software Ori- ented Architectures and, recently, microservice architectures) [1, 3]. Correspondingly, cloud vendors have started offering serverless com- puting services, hosting on granularly-billed resources the emerg- ing application services. The notion of Function-as-a-Service (FaaS) can be seen as a combination of both developments, allowing users to provide cloud platforms with arbitrary functions leaving the operational logic to be managed by the cloud provider.

This model offers new benefits but also raises new challenges. [7] On a conceptual level, the field lacks consistent terminology, along with a clear reference architecture, inhibiting objective compari- son of different FaaS platforms. On a more technical level, there are various challenges, in domains such as software engineering, system operations, and performance engineering, that potentially hinder further adoption of the paradigm.

Towards addressing these challenges we provide an overview of our ongoing work, in which we address both the conceptual ascollaboration with the SPEC CLOUD RG[2], we provide an analy- sis of the challenges and perspectives in this field (section 2). To address the technical challenge of serverless function composition, we present our work on a prototype of a serverless workflow man- agement system (section 3).

2 PERSPECTIVES AND CHALLENGES

In our prior work [7] we analyzed and proposed four perspectives that aim to capture the direction of the serverless field, and in particular of FaaS architectures. First, the stateless, short-lived na- ture of these cloud functions makes FaaS an ideal candidate for the dynamic models of hybrid clouds and fog computing. Second, as serverless leads to the near-absence of CAPEX, companies will focus on managing the OPEX costs of running applications, rais- ing interesting performance engineering challenges, along with a reevaluation of the existing Dev and Ops roles. Third, we forecast that for individual cloud functions absolute performance will start to matter less, whereas the application-specific performance/cost ratio will start to matter more. Finally, FaaS architectures could be- come the basis for a programming model that is native to the cloud, that is, it considers services as naturally as any other programming concepts, to be developed, composed, and shared with ease.

Towards these perspectives, the current challenges in this do- main can be divided into 3 categories[7]:

(1) Software engineering challenges have been identified as some of the most pressing issues with the serverlesspara- digm, such as developer experience[5]. To overcome these challenges posed by the FaaS model, advances need to be made in testing, tooling, functionality, and training and edu- cation.

(2) System (operational) challenges arise due to the highly dynamic nature of cloud functions, calling for advances in security, cost predictability, and cloud function lifecycle man- agement.

(3) Performance engineering challenges in this emerging paradigm, arising due to the increased delegation of concerns to the cloud platform, which include size-able overheads, un- reliable performance, and new forms of the cost-performance trade-off.

Leading the effort of the SPEC RG CLOUD group in serverless computing, we specifically target the performance engineering challenges, which we expand upon in [6]: Scheduling policies that specifically target the constraints and trade-offs of FaaS, such as caching function source code efficiently.[4] There are various ofpossibilities to reduce, the currently significant, overhead of FaaS platforms, including reusing deployed functions efficiently. Per- formance isolation could be improved to obtain a more resource-efficient placement of functions, avoiding a bottleneck of particular resource dimension on a machine. With the level of introspection that FaaS platforms have in scheduled functions, platforms could be improve to better profile the characteristics of functions and provide better performance predictions. The control of FaaS plat- forms that have on function life-cycles, also allows for further exploration of the cost-performance trade-off, where we can make distinctions between high-priority functions and background func- tions. Finally, the field of serverless computing currently lacks a systematic, objective benchmark of FaaS platforms, along with a consistent terminology for the common components.

As part of our ongoing work we are focusing on this last chal- lenge of being able to objectively compare the performance of common components in different FaaS platforms. Our analysis of multiple FaaS platforms so far are shown in the form of a reference architecture as shown in figure 1. It shows how this stack of abstract layers that comprise a FaaS platform abstract away the operational concerns, like resource management, and autoscaling, the higher up you get in the stack.

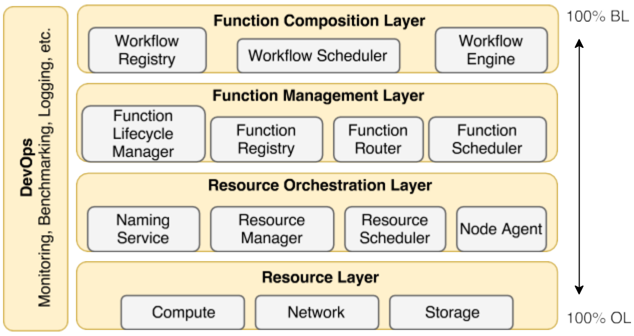

Figure 1: FaaS Reference Architecture (Work-in-Progress) ordered from business logic (BL) to operational logic (OL).

3 FISSION WORKFLOWS

Separate from the SPEC RG CLOUD effort, we are specifically look- ing into how to introduce composition in the FaaS model. Function composition in FaaS should allow the developer to avoid operational concerns related to function composition, such as communication logic and state management. Additionally, as these functions, simi- lar to current functional programming languages, can get composed into increasingly complex functions, it is vital that the overhead of composition should be minimized.

For this purpose we developed Fission Workflows1, which is

a workflow management system (WMS) specifically targeted at serverless workflows. Similar to existing WMSs, it provides a re- liable, traceable executions of function dependency graphs, trans- ferring data between functions where needed. In contrast to tradi- tional WMSs, it focuses on minimizing the overhead of the WMS and provides a workflow definition language focused on promoting composition of functions and workflows.

Within the Fission Workflows prototype, the workflow scheduler contains scheduling policies that optimize the serverless workflow execution. One of the implemented scheduling policies aims to reduce the execution time by booting functions ahead of the ac- tual invocation, reducing the provisioning delay of new functions. Additionally, using the data collected in the workflow engine the scheduler is able to profile the individual functions and thereby improve its runtime predictions.

We aim to evaluate the performance of this workflow engine on four aspects, namely fault tolerance, throughput, performance, and resource consumption. Additionally, experiments will be conducted to compare the performance of Fission Workflows with alterna- tive approaches to function composition, including coordinator functions, hosted cloud coordinators, such as AWS Step Functions, and traditional workflow management systems, such as Luigi or Apache Airflow. Finally, we want to evaluate the effectiveness of different scheduler policies; their impact on the workflow execution compared to their effect on the overhead of the WMS.

4 CONCLUSION & ROADMAP

The emerging serverless and FaaS models hold good promise for future cloud applications, but also raise new (performance-related) challenges that can hamper its adoption. In this work, we pre- sented the approaches how we aim to contribute to solving these challenges on different levels. Specifically for function composi- tion and the reduction of FaaS performance overhead, we propose a workflow-based approach to the orchestration of dependency graphs of functions with the Fission Workflows prototype. On a higher level, we address community-wide problems, such as ter- minology and benchmarking with a joint industry and academic effort with the SPEC CLOUD RG.

In our envisioned roadmap we continue with this wide approach to the various challenges in serverless computing. We see the Fis- sion Workflows prototype as the foundation of further research into function composition, with challenges such as data transfer, hybrid cloud deployments, and advanced scheduling policies. Furthermore, with the collaborative SPEC CLOUD RG effort we will continue to encourage researchers to join the effort of developing an com- prehensive reference architecture and industry-wide benchmark of FaaS platforms.

REFERENCES

[1] Nick Gottlieb. 2016. State of the Serverless Community Survey Results. https://serverless.com/blog/state-of-serverless-community/. (Nov 2016).

[2] Cloud Group. 2017. SPEC Research Group’s Cloud chapter. https://research.spec. org/working-groups/rg-cloud.html. (2017).

[3] Eric Jonas, ShivaramVenkataraman, Ion Stoica, and Benjamin Recht. 2017. Occupy the Cloud: Distributed Computing for the 99%. In SoCC. 1:1. (In print, available as CoRR article http://arxiv.org/abs/1702.04024).

[4] Edward Oakes, Leon Yang, Kevin Houck, Tyler Harter, Andrea C Arpaci-Dusseau, and Remzi H Arpaci-Dusseau. 2017. Pipsqueak: Lean Lambdas with Large Li- braries. In Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on. IEEE, 395–400.

[5] Mike Roberts. 2016. Serverless Architectures. https://martinfowler.com/articles/ serverless.html. (2016).

[6] E. van Eyk, A. Iosup, C. Abad, Eismann S., and Grohmann J. 2018. A SPEC RG Cloud Group’s Vision on the Performance Challenges of FaaS Cloud Architectures (submitted). In ICPE.

[7] E. van Eyk, A. Iosup, S. Seif, and M. Thömmes. 2017. The SPEC cloud group’s research vision on FaaS and serverless architectures. In WoSC held with ACM/IFIP/Usenix Middleware.