

Optimizing Latency in Function-as-a-Service with Distributed Promises

Soam Vasani
Platform9 Systems Inc.
Sunnyvale, California
soam@platform9.com

Erwin van Eyk
Platform9 Systems Inc.
Sunnyvale, California
TU Delft, The Netherlands
erwin@platform9.com

ABSTRACT

Functions-as-a-Service (FaaS) or serverless functions have emerged to be particularly well-suited for stateless, short-lived computation. They make the power of complex distributed systems, such as Kubernetes, accessible to many programmers without the steep learning curve. At the same time, these FaaS functions provide cloud providers with unprecedented control over the underlying infrastructure and operational logic, opening up new options and insights that could be used to improve performance beyond that of existing cloud models.

However, dealing with asynchronous, stateful functions in FaaS is a largely unsolved problem - especially when it comes to communication with third-party applications. Although, the introduction of workflow management systems (e.g. AWS Step Functions, Fission Workflows) in this field has provided it with a way to manage state through composition of FaaS functions into more complex functions or workflows, these do not capture well the asynchronous interactions of functions with external systems.

In this talk we explore the problem of FaaS functions and workflows interacting asynchronously with stateful systems (e.g. AWS MechanicalTurk) external to the FaaS system. We explore several different approaches, including polling, callbacks, channels, etc.

We propose the use of distributed *Promises* to elegantly solve stateful coordination problems between FaaS functions, workflows, and external systems. For the application developers the familiar notion of promises allows them express asynchronous tasks more intuitively using the FaaS model. On the other hand, for cloud providers and researchers promises provide yet more control and options to improve the performance of the FaaS platform. One interesting example of this is *Promise Pipelining*, which would allow for reductions in latencies by eliminating network round-trips in certain cases.

Using a prototype implementation of distributed Promises on top of the Kubernetes orchestration platform, we demonstrate integration with the Fission FaaS and Fission Workflows systems. With this prototype, and an overview of potential research directions, we hope to encourage researchers to also explore this direction of decoupling data from computation in FaaS.

CCS CONCEPTS

• **Computer systems organization** → **Cloud computing**; • **Applied computing** → **Service-oriented architectures**; **Event-driven architectures**;

KEYWORDS

Function-as-a-Service, Serverless Computing, Promises, Execution Model

ACM Reference Format:

Soam Vasani and Erwin van Eyk. 2018. Optimizing Latency in Function-as-a-Service with Distributed Promises. In *Proceedings of First Workshop on Hot Topics in Cloud Computing Performance (HotCloudPerf 2018)*. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

HotCloudPerf 2018, April 2018, Berlin, Germany

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>