Poster Abstract: Hierarchical Serverless Computing for the Mobile Edge

Eyal de Lara, Carolina S. Gomes, Steve Langridge, S. Hossein Mortazavi, Meysam Roodi
Central Hardware Engineering Division
Huawei Technologies Co.
Toronto, Canada

***Abstract*—EdgeScale is a new platform that leverages serverless cloud computing to enable storage and processing on a hierarchy of data centers positioned over the geographic span of a network between the end-user device and the traditional wide-area cloud datacenter. EdgeScale applications are structured as lightweight stateless handlers that can be rapidely instantiated on demand. EdgeScale provides a scalable and persistent storage service that automatically migrates application state across the data center hierarchy to optimize access latency and reduce bandwidth consumption.**

***Keywords*-serverless computing; edge computing; cloudlets;**

Today most computing applications consist of two main components: an interface that runs on the users device (smartphone, PC) and server logic that is deployed on a cloud datacenter accessed over a wide-area network. Whereas this paradigm currently experiences wide success, we argue that it is about to be greatly changed due to two main factors: (1) wide-area network latencies stand in the way of the crisp interactive response time required by next generation applications, such as augmented reality and real-time translation, and (2) the Internet of Things (IoT) is anticipated to add billions of new devices to the network which risk to overwhelm the already taxed network infrastructure.

To address these challenges voices in the research community have proposed the offloading of computation, storage and network functions to the edge of the network closer to the user. This approaches, variously referred to in the academic literature as cloudlets [1], micro data centers, or fog [2], augment the traditional cloud architecture with an additional layer of servers that are located closer to the end user (typically one-hop away). Whereas certain dimensions (such as power, cooling, bandwidth etc) at the edge are restricted relative to wide area cloud datacenters, with the maturing of micro servers it is possible today to deploy small and medium size (tens to hundreds of cores) datacenters closer to the edge.

We believe that edge computing is a bold first step toward a general hierarchical cloud architecture implemented over the geographic span of a network, that supports scalable data streaming by providing storage and computation along the path between the end-user device (e.g., smartphone, IoT appliance) and the traditional wide-area cloud datacenter. We envision a succession of micro and mini datacenters positioned between end user device and large traditional wide-area cloud datacenter. In this approach, the execution of applications takes place as close to the edge as possible, moving deeper into the network only when needed. For example, a request from a mobile or embedded device would be first handled by a micro datacenter one hop away. If additional computation or state is required, the request could be escalated to a regional mini datacenter, and (if need be) to the wide-area cloud datacenter. This hierarchical approach provides for massive scalability by distributing load geographically and cuts response time by minimizing network latencies. In addition, the network operator benefit from local processing that reduces stress on core resources and backhaul bandwidth. Our vision does not assume that all applications will use all layers of the hierarchy at all times. Instead, we expect applications to opportunistically reconfigure themselves and execute functionality at most appropriate locations given the user needs and the networks dynamic status.

EdgeScale is a new research platform that we are developing to explore the hierarchical cloud computing vision. EdgeScale implements a serverless computation model [3], [4], [5], [6], [7], [8] that supports code and data mobility by enforcing a clear separation between computation and state. EdgeScale applications are composed of a collection of stateless event handlers that are implemented using high level languages, such as Java or Python, and can therefore execute on a variety of architectures (e.g., X86, ARM). Since handlers tend to be small, they can be easily instantiated on any server that runs the EdgeScale framework. In fact, EdgeScale does not give developers any guarantees about the server where a handler may run, and as such application developers cannot make any assumption about the local availability of preexisting state. Instead, EdgeScale provides a scalable and persistent storage service that handlers can use to read and store state through well-defined interfaces. In turn, EdgeScale automatically migrates objects (application code and user data) across the data center hierarchy to optimize access latency and reduce bandwidth consumption. EdgeScale leverages hardware acceleration techniques at the edge to address practical space and power constraints. It is anticipated that micro clusters at the edge will have a limited number of compute elements as well as limitations on their power and thermal envelope. Hardware acceleration is an attractive approach for increasing computation capacity.

978-1-5090-3322-5/16 $31.00 © 2016 IEEE
DOI 10.1109/SEC.2016.37

EdgeScale consists of the following main components:

- EdgeExecute: Implements a serverless cloud container that supports the execution of lightweight stateless application handlers. The current implementation is based on AppScale [7], a serverless cloud computing system,which is an open-source implementation of Google App Engine [8].
- EdgeStore: Provides a scalable and persistent storage service that handlers can use to read and store state through well-defined interfaces. It provides a flexible column-based storage layer that automatically migrates objects across a hierarchy of data centers to optimize access latency and reduce bandwidth consumption.
- EdgeRoute: This component is in charge of routing requests to the appropriate EdgeScale node. Routing decisions take into account the user's location in the network, application preferences, and system state (e.g., application availability, load).
- EdgeDeploy: Dynamically deploys and removes EdgeScale applications and associated user data on EdgeScale nodes, according to fluctuations in application demand by users.
- EdgeStream: Makes it possible for the otherwise stateless EdgeScale applications to implement connection-oriented functionality. EdgeScale applications use the service to create stateful network connections and to register application handlers that execute on data arrival. EdgeStream can be used to implement flexible media streaming applications.
- EdgeAccelerate: Provides hardware acceleration services for common operations such as video encoding and manipulation.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[3] "AWS Lambda," https://aws.amazon.com/lambda/, May 2016.

[4] "IBM OpenWhisk," https://developer.ibm.com/openwhisk/, May 2016.

[5] "Microsoft Azure Functions," https://azure.microsoft.com/enus/services/functions/, May 2016.

[6] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Serverless computation with OpenLambda," in *8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16)*, 2016.

[7] N. Chohan, C. Bunch, S. Pang, C. Krintz, N. Mostafa, S. Soman, and R. Wolski, "Appscale: Scalable and open appengine application development and deployment," in *International Conference on Cloud Computing*. Springer, 2009, pp. 57–70.

[8] D. Sanderson, *Programming Google App Engine: build and run scalable web apps on Google's infrastructure*. O'Reilly Media, Inc., 2009.