PhD Position - On-Demand/Programmable VIMs' Collaborations to Operate and Use Geo-Distributed Infrastructures

Mission and activities

The PhD position will focus on a new composition model aiming at delivering on demand/programmable collaborations between several Virtual Infrastructure Managers (VIMs), each operating one site (i.e., a medium/micro data center). As opposed to the state-of-the-art approaches that have been focusing on designing "glue" components in the form of a hierarchical broker or orchestration services, the innovative part is to automatically recompose the workflow of control plane operations to be able to perform inter and multi sites requests without requiring intrusive changes at the code level.

While it is clear that edge infrastructures are required for emerging use-cases related to IoT, Virtual Reality, Network Functions Virtualization and others latency critical applications, there is currently no resource management system able to deliver all the features that make the success of cloud computing for smaller data centers deployed at the edge. Production solutions like Akamai Cloudlets [1] or AWS Lambda [2] do not allow for instance to run stateful workloads in isolated environments (e.g., containers, virtual machines) due to the numerous constraints that come with edge infrastructures [3]. For instance, edge sites are potentially unmanned and therefore must be administered remotely; management systems should be designed to cope with intermittent network access to sites; distinct operators might be interested in interconnecting their infrastructures (like network peering), etc.

In 2016, the ETSI Mobile Edge Computing Industry Specification Group defined an architecture to orchestrate distinct independent cloud systems, a.k.a.Virtual Infrastructure Managers (VIM) [4]. The idea consists in federating VIMs of the different Data Centers (DCs) that compose the edge infrastructure. By reusing VIMs, ETSI targets an edge computing resource management that behaves in a same fashion as traditional ones, while mitigating development requirements. Although there is no implementation available, the idea of federating VIMs seems promising as several projects have been built on similar concepts. ONAP [5], an industry-driven solution, enables the orchestration and automation of virtual network functions across distinct VIMs. From the academic side, FogBow [6] aims to support federations of Infrastructure-as-a- Service (IaaS) providers. More recently, NIST initiated a collaborative effort with IEEE to advance Federated cloud through the development of a conceptual architecture and a vocabulary[1]. These projects provide valuable contributions, however, they all have been designed by only considering the DevOps' perspective : they deliver abstractions to orchestrate the life cycle of geo-distributed applications across several VIMs. But, they do not allow operators to manage edge resources overall.

STACK members have been dealing with the aforementioned challenges since 2016. In July 2018, they alerted the community of the importance of delivering such a resource management system for the advent of the edge paradigm, underlining the main challenges to tackle [7]. In December 2018, they presented a new research direction to enable on demand collaboration between distinct OpenStack instances without intrusive change. Considering that the federation of several

[1] https://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/FederatedCloudPWGFC (Valid on March 2019).

independent but similar VIMs can be seen as a single software composed of several instances of the same sub-service (*i.e.,* one per site), we proposed to extend the OpenStack API to be able to define, dynamically, the composition of the OpenStack services. Concretely, we propose to add a dedicated option, called `--os-scope`, that specifies which services (e.g., compute, image, etc.) of which OpenStack instances (e.g., zone 1, zone 2) an OpenStack is made of to perform the request. For instance, the provisioning of a VM in zone 1 with an image in zone 2 will be invoked as follows :

```
openstack server create vm --flavor tiny --image cirros \
    --os-scope '{"Compute": "Zone1", "Image": "Zone2"}'
```

A first proof-of-concept will be presented in May 2018 during the OpenInfrastructure summit[2].

The first objective of this PhD is to formally define the model on which this idea has been built, with the goal of ensuring the correct composition of the control plane services. The second objective will aim at integrating into this model the set theory (or relational algebra) to enable multi sites operations. For instance, populating a new VM image in a subset of the available edge zones will correspond to the following request :

```
openstack image create \
--scope '{"Identity": "Zone1", "Image": "Zone1 && Zone2 && Zone3"}' my-new-image
```

One can envision other operators such as OR and XOR that will enable finer-grained collaborations. For instance, provisioning a VM either in edge zone 1 or zone 2 could be requested as follows:

```
openstack server create --flavor m1.tiny --image cirros \
--scope '{"Identity": "Zone1", "Compute": "Zone1||Zone2", "Image": "Zone1"}' my-vm
```

The last objective will be to implement this model into a domain specific language that will enable administrators/DevOps to perform cross and inter-site operations between independent VIMs. More generally, the DSL we envision will deliver common set operators to allow the execution of one request on a large number of edge zones (possibility of executing one operation on every edge zone except one, composition of multiple edge zones into a new one, etc.).

By leveraging programming software compositional proposals [8], this approach breaks with the state of the art that of federated approaches that have been mainly focused on broker and orchestrator approaches. Among the different questions this PhD should deal with, studying whether the model can be generalized to different software stacks such as kubernetes or more application oriented systems is probably the most important one we would like to tackle with the ultimate goal of delivering key design principles for edge services, where network disconnections/offline mode is the norm.

References

[1] Akamai Cloudlets. http://cloudlets.akamai.com. (Accessed: 2018-03-08).

[2] Amazon Lambda@Edge. https://aws.amazon.com/lambda/edge/. (Accessed: 2018-03-08).

[2] https://www.openstack.org/summit/denver-2019/summit-schedule/events/23352/implementing-localization-into-openstack-cli-for-a-free-collaboration-of-edge-openstack-clouds (Valid on March 2019)

[3] The OpenStack Foundation. Cloud Edge Computing: Beyond the Data Center (White Paper). https://www.openstack.org/assets/edge/OpenStack-EdgeWhitepaper-v3-online. pdf, Jan 2018. (Accessed: 2018-03-08).

[4] Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., and Giust, F. Mobile- Edge Computing architecture: The role of MEC in the Internet of Things. IEEE Consumer Electronics Magazine 5, 4 (Oct 2016), 84–91.

[5] Open Network Automation Platform. https://www.onap.org. (Accessed: 2018-03-08).

[6] Brasileiro, F., Silva, G., Araújo, F., Nóbrega, M., Silva, I., and Rocha, G. Fogbow: A middleware for the federation of IaaS Clouds. In The 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) (2016), IEEE, pp. 531–534.

[7] R.-A. Cherrueau, A. Lebre, D. Pertin, F. Wuhib et J. M. Soares - Edge Computing Resource Management System: a Critical Building Block ! USENIX Workshop on Hot Topics in Edge Computing (HotEdge), Boston, USA, July 2018

[8] Fujii K, Suda T. Dynamic service composition using semantic information. InProceedings of the 2nd international conference on Service oriented computing 2004 Nov 15 (pp. 39-48). ACM.

Skills and profiles

Knowledge/experience on programming languages and software engineering
Knowledge/experience on distributed systems (OpenStack/Kubernetes would be a plus)
Experimentation skills
Autonomy / Curiosity
English mandatory

Additional information

The candidates are invited to contact Adrien Lebre (firstname.name@imt-atlantique.fr).

Duration: 36 months
Location: Nantes, France