Ph.D. thesis: Distributed troubleshooting of edge-compute functions

Technicolor, Research and Innovation, Rennes.
INRIA/IRISA, ASAP team, Rennes.
contacts: christoph.neumann@technicolor.com, francois.taiani@irisa.fr

Most of today's Internet applications and services rely on the cloud, i.e. they execute within datacenters that concentrate the associated compute functions and data storage. This cloud-based approach brings strong operational benefits, but also comes with important limitations, in particular in terms of loss of privacy, cost, availability (for end-devices with intermittent connectivity) and latency.

To address the challenges posed by pure cloud-based deployments, alternative distributed architectures are emerging that emphasize *decentralized* and *loosely coupled* interactions. This evolution can be observed at multiple levels of an application's distributed stack: the growing interest, both practical and theoretical, for weak consistency models is one such example.

At a higher, more architectural level, similar motivations explain the push for *micro-services* and recent efforts to offload computing resources into the network (such as *edge computing*) [1] and embedded client-side devices (such as *fog computing*). Micro services help development teams decompose complex applications into a set of simpler and loosely-connected distributed services. They thus encourage a strong decentralization of cluster and cloud-hosted applications, away from traditional monolithic strategies.

The rise of micro-services, fog-, and edge-computing are prompting a fresh rethink of the typical distribution of capabilities between servers and clients in a distributed applications. This is likely to lead to more services and computations being offloaded to geo-distributed devices, in particular within hybrid cloud/edge architectures.

In this context, this PhD thesis focuses on the opportunities that recent generations of end-user gateways (or more generally end-user devices) will offer to implement an edge-compute paradigm powered by user-side micro-services (as illustrated by Amazon's recent announcement of its *Greengrass* platform[1]). We suppose that a set of devices are distributed among the homes of end-users, and each contribute to the distributed system its computing power and the ability to quickly deploy compute functions in an execution environment. In order for service and application providers to actually use the system and deploy applications, the system must ensure an appropriate level of reliability, while simultaneously requiring a very low level of maintenance in order to address the typical size and economics of gateway deployments (at least a few tens of million units).

Providing a good level of reliability in such a large system at a reasonable cost is unfortunately difficult. To address this challenge, we aim in this thesis to exploit the *natural distribution* of such large-scale user-side device deployments to quickly pinpoint problems and troubleshoot applications experiencing performance degradations. Because of the distributed nature of the systems, the root causes of these degradations may be multiple and also distributed. We would like in particular to pinpoint the likely faulty component(s) or subsystem(s): are the edge node(s) (particular ones or all?) responsible? the network (or some part of it)? the end-devices or home networks accessing the service?

The problem of pinpointing faulty components is related to that of anomaly detection and bottleneck identification [2]. Some related works have attacked this problem by applying machine learning on network-level TCP traffic features [4], [5]. In this PhD thesis, we adopt a higher level approach and willingly do not consider low level traffic information (which may results in models and algorithms that are too specific for a given application). One important research challenge is therefore to define the set of generic metrics (e.g. latency, throughput) that allow to troubleshoot widely distributed and loosely coupled applications. Further, the thesis should define where to collect, how to collect, to aggregate and to share these metrics. Here communication overhead, compute and communication cost as well as privacy constraints must be taken into consideration. Instead of local approaches having a single or only a few vantage points (as in [4], [5]) we favor distributed approaches. Finally, the PhD thesis should design and evaluate distributed machine learning and algorithms for pinpointing the faulty component.

The PhD thesis aims to build and evaluate models, algorithms and data-structures with a sufficient level of abstraction, in order to deliver results that remain generally applicable beyond any particular use case or application. However, we also plan to implement and evaluate on edge-devices some of the primitives designed within the course of the PhD thesis, in order to validate that the proposed primitives are able to execute in a resource-constrained environment.

Applicants profile

We are looking for applicants with a strong background and interest in distributed systems and networks. Theory and algorithms as well as design and implementation considerations are of importance in this thesis, and therefore a good theoretical background but also the ability to prototype and validate results in practice are of importance.

[1] `https://aws.amazon.com/greengrass/`

REFERENCES

[1] Valancius, Vytautas, et al. ”Greening the internet with nano data centers.” ACM CoNext 2009.

[2] Ibidunmoye, Olumuyiwa, Francisco Hernndez-Rodriguez, and Erik Elmroth. ”Performance anomaly detection and bottleneck identification.” ACM Computing Surveys (CSUR) 48.1 (2015): 4.

[3] Ahmed, Tarem, Boris Oreshkin, and Mark Coates. ”Machine learning approaches to network anomaly detection.” USENIX SysML, 2007.

[4] Arzani, Behnaz, et al. ”Taking the blame game out of data centers operations with NetPoirot.” ACM SIGCOMM 2016.

[5] Dimopoulos, Giorgos, et al. ”Identifying the root cause of video streaming issues on mobile devices.” ACM CoNext 2015.