Assessing the Potential Energy Savings of a Fluidified Infrastructure

Assessing the Potential Energy Savings of a Fluidified Infrastructure

Stefano Galantino (stefano.galantino@polito.it)
Politecnico di Torino - Dept. of Computer and Control Engineering

Fulvio Risso (fulvio.risso@polito.it)
Politecnico di Torino - Dept. of Computer and Control Engineering

Vlad C. Coroamă (coroama@tu-berlin.de)
Technische Universität Berlin - Institute of Environmental Technology

Antonio Manzalini (antonio.manzalini@telecomitalia.it)
Telecom Italia Mobile - Innovation Labs

***Abstract*—With the *computing continuum*, highly geographically-dispersed computing resources are aggregated into logical resource pools, spanning from the cloud to the edge of the network. This paper presents and analyzes a concrete use case for the continuum, a University lab where end-user devices are aggregated in a continuum substrate, in which applications can be transparently executed either on the local machine of the user or on the continuum.**
This paper (*i*) proposes a suitable methodology to transform current desktop applications into proof-of-concept fluid software, then (*ii*) evaluates the benefits of the distribution of the fluid workload across heterogeneous computing devices, focusing on the power consumption of the continuum, showing how this paradigm can bring valuable improvements in terms of energy consumption of the entire computing infrastructure.

1. INTRODUCTION

The widespread adoption of cloud computing technologies enabled a notable consolidation of computational resources, which resulted in unprecedented service agility and massive cost and energy savings [1] [2]. In recent years, the IT landscape has evolved further, with myriads of sensors and Internet of Things (IoT) devices being installed in homes, industrial premises, and public spaces [3]. The edge computing paradigm has been proposed as an approach to reduce computational latency (as processing happens closer to data sources) and bandwidth usage, while improving privacy and reliability.

The *computing continuum* has been introduced as a viable solution to integrate cloud, edge, and IoT sensors, while providing guarantees in Quality of Experience (QoE) even for computationally-intensive latency-sensitive applications [4] [5] [6] [7]. However, as of today, most edge computing installations consist of servers deployed on premises that process data, which is further elaborated in remote data centers. This top-down approach, in line with the original cloud computing paradigm, simply extends the cloud data center approach to a local point of presence, failing to integrate with the edge substrate and to leverage the unused processing capacity.

This paper, based upon novel technologies that enable an initial implementation of the com-

Published by the IEEE Computer Society

puting continuum [8], extensively evaluates the possible benefits of such a paradigm shift in our university production and educational environment. These initial results can be easily generalized to cover a real enterprise department, in which each layer (i.e., Cloud, Fog, Edge, etc.) can leverage the resource continuum to actively execute the desired applications. However, such an approach requires an extension of both the concept of cloud and edge. First, the cloud is no longer perceived as the centralized computing power, located in the core of the network, but it rather generalizes to any set of servers that run cloud orchestration frameworks (i.e., including also *private* cloud). Second, the edge is further extended to include also end-user devices (e.g., laptops, mobile phones, etc.). Therefore, from the end-user perspective, applications can either be executed locally on end-user devices or in the continuum, based on the application QoE requirements and the specific goal of the infrastructure (e.g., reduce latency, power consumption, etc.).

This innovative approach to edge-to-cloud management and orchestration raises many additional challenges to cope with the dynamicity and heterogeneity of computing devices. Still, it has the potential to bring the same benefits of the cloud also to the edge of the network, e.g., with respect to power consumption. In fact, this paper presents a preliminary assessment of the potential energy savings when the *fluid* technology is used. This is achieved by analyzing a real use case in the computing continuum, namely a University lab that can benefit from the increased flexibility in running user applications on all the available devices. To achieve this goal, two additional minor contributions can be also envisioned: (*i*) we introduce a new definition of performance requirement, to describe the application QoE within the heterogeneous devices of the continuum, and (*ii*) we define a possible methodology that enables the shift from current desktop applications to the fluid containerized environment.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 introduces enhanced metrics, able to describe respectively the performance of a given platform, and the execution requirements of an application. Section 4 details the additional requirements of a continuum application. Section 5 evaluates the model presented in the previous sections to assess the possible benefits of the cloudified approach in terms of power consumption. Finally, Section 6 concludes the paper.

2. RELATED WORKS

The computing continuum, as an emerging concept, has been intensely discussed in recent literature as a promising solution to allow further evolutions on many "smart" systems (e.g., Smart-cities, Smart-grid, and Smart-industry) [4] [9] [10]. In fact, the Cloud-to-Edge resource continuum guarantees low-latency communications with sensors and actuators, and high-performance computing (HPC) for further data analysis and forecasting. Function-as-a-Service (FaaS) showed some promising results, leveraging the heterogeneous resource continuum: function calls can be executed in the different sections of the resource continuum depending on the individual requirements. Although this paper focuses on lightweight virtualization (i.e., containers), FaaS experiences similar requirements, because, as it happens with containers, computing resources must be granted to function call execution. Pilot-Edge [7] introduced a FaaS interface for application-level tasks on a resource continuum with heterogeneous devices. Delta [5] further extended the problem representation, introducing dynamically evolving estimates of function execution times to determine the most appropriate location. Still, both solutions focused on maximizing the performance for function call execution, neglecting possible implications in the infrastructure power consumption, which, instead, has been intensively investigated in [11] [12] [13]. However, they primarily focused on reducing the power consumption for battery-constrained end-user devices, not accounting for the additional energy requirements on servers. The work presented in this paper, instead, overcomes such narrow evaluation and encompasses all the different devices of the continuum in the evaluation of the energy-aware application placement.

The Cloud-to-Edge continuum is a very dense and complex scenario. At the core level (i.e., cloud) resources can be considered unlimited, whilst they become scarce at the edge. Thus, simulations may be used to perform early-stage evaluations before proceeding to real-world (and

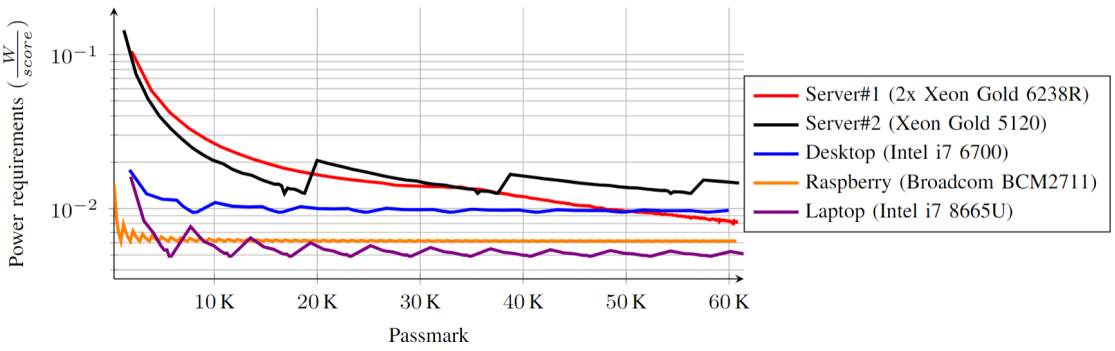

Figure 1. Power requirements of different devices, computed as the ratio between the consumed energy and the measured performance.

thus more costly and complex) testbeds. Abreu, David Perez, et al. [14] proposed a comparative analysis of three of the most promising simulators for the Cloud-to-Fog continuum. Specifically, iFogSim [15], CloudSimSDN [16] (both extensions of the well-known CloudSim [17]) and YAFS [18] guarantee enhanced flexibility in the definition of both the infrastructure and the workloads, while providing empirical results also in terms of infrastructure power consumption. Still, none of the above simulators can provide an experimental evaluation to correctly model the heterogeneity of devices that may be part of the computing continuum, as the only perceived difference between Edge and Cloud is strictly related to the amount of available computing power, and not to the performance of a specific device. Our evaluation is – to the best of our knowledge – the first to properly account and simulate the heterogeneity of computing resources in the continuum while providing insights on the infrastructure power consumption.

3. DEVICE CHARACTERIZATION

University campuses may include a wide variety of devices, such as servers, laptops, desktop computers, and, lastly, low-end devices like Raspberry PIs. They differ not only in terms of the number of available computing resources (i.e., number of CPU cores) but also in terms of performance per core, as different CPU architectures may provide very different computing power (e.g., one Raspberry ARM Cortex core is significantly less powerful than a high-end Intel i7 core). Therefore, traditional performance metrics, usually expressed in terms of number of CPU cores reserved, need to be replaced with a CPU-independent mechanism, which can guarantee that the same amount of work is done despite any difference in CPU architectures.

We used the Passmark score[1] to compare the performance among different CPU models, which represents a free alternative to the costly SPEC[2] suite. At the same time, it avoids the known limitations of MIPS benchmarks, which focus on the number of instructions per second and do not capture the complex set of conditions affecting the CPU performance (e.g., core turbo boost, thermal throttling, etc.). In fact, the Passmark score mimics the behavior of several common applications, including complex mathematical calculations involving compression, encryption, and physics simulations, to effectively evaluate the performance of a given platform. It is widely adopted by the computing community and has shown consistent results even across different OS [19] [20]. For our evaluation, the Passmark tool has been containerized using Docker for two reasons: (*i*) Docker is widely used as virtualization technology in cloud solutions, allowing us to replicate the execution environment for a generic workload, and (*ii*) it offers the possibility to restrain the Passmark execution on a subset of computing resources (e.g., on N CPU cores), hence allowing us to map the "traditional" representation of computing resources into the new Passmark metric. Specifically, the Passmark container has been executed by linearly increasing

[1]https://www.passmark.com/.
[2]https://www.spec.org/benchmarks.html.

the number of CPU cores assigned and collecting the related value of Passmark score. The proposed performance metric enables the characterization of the desired workload in terms of CPU resources (e.g., N vCPU) required on any given platform to deliver the same workload (e.g., M Passmark score).

In addition to performance, different CPU architectures heavily differ in power consumption, as some chips are designed for power efficiency (e.g., Raspberry PIs), whereas others privilege processing performance (e.g., servers). As a consequence, this paper needs first to establish a common ground to compare performance and power consumption across different CPUs. Power consumption metrics are then evaluated to correlate device power requirements with delivered performance. Specifically, these metrics are collected using a smart plug, connected directly to the wall outlet, monitoring the total power consumption of the device, while progressively increasing the number of CPU cores assigned to the Passmark application.

Results in Fig. 1 correlate the performance of different devices (hence, different CPUs) in terms of energy required to deliver a single Passmark (the lower, the better). Due to the different levels of performance achievable by each CPU, a new CPU was added to the plot when the previous ones become 100% loaded, hence resulting in the 'saw tooth' shape of the figure. As shown in the figure, it is not always possible to identify the most efficient device, as such efficiency may be extremely variable, depending on the applied workload.

4. APPLICATION CHARACTERIZATION AND CONTAINERIZATION

This paper relies upon the novel concept of fluid workload distribution, which fundamentally changes the way applications are perceived and executed, in particular when referring to most of the current desktop-oriented applications. This new paradigm defines a computing continuum of resources, in which applications are no longer executed solely on the particular device where they have been requested by the user. Instead, applications can be started on the most convenient device within the infrastructure, as it happens with the cloud computing paradigm. This paradigm shift requires not only a proper infrastructure, able to provide benefits in the given scenario, but it could potentially involve a substantial re-design of the application to allow seamless interaction for the end user.

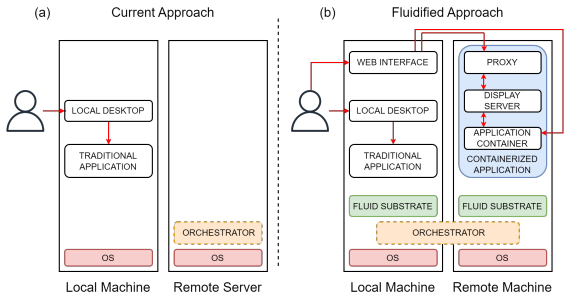

Figure 2. Components involved when running a user application with the traditional approach vs using a fluidified approach.

Compared to the current monolithic approach (Fig. 2a), in which the end user directly interacts either with the application through the *local desktop* or with the remote server, such as running remote virtual desktops (i.e., Virtual Desktop Infrastructure – VDI) through a generic orchestrator, the fluidified approach heavily relies on containerization to run the same workloads, while maintaining the same interaction with the end-user. Specifically, our fluid applications are composed of three layers:

1) *Application container*: the traditional, monolithic process now running in a containerized environment.
2) *Display server*: it captures all the graphical input/output of the application and makes them available to the remote user, without any modification to the running application. This is currently implemented as a VNC server.
3) *Web proxy*: it enables the interactions between a client and the display server through the HTTP/HTTPS protocols, hence through a web browser, which simplifies the user experience. Technically, it transforms WebSockets (RFC 6455), and the subsequent TCP bi-directional traffic, into the VNC protocol spoken by the Display server.

Although this layered approach enables a seamless shift of current applications towards the fluid approach, it could add some performance drawbacks, caused by the graphical processing performed both on the end-user and application side. A more recent trend in application development foresees the strict decoupling of application logic and content visualization (i.e., in a cloud-native client-server fashion). The fluidified approach would massively benefit from such a design choice, removing the unnecessary overhead caused by the proxy and display server layers and, additionally, with the possibility to share the same back-end replica between multiple front-ends.

Recently, our university has adopted a similar solution to deliver the Programming Fundamentals exam to the enrolled students.[3] Multiple containerized replicas of the PyCharm IDE are executed on private servers, and students were able to interact with the IDE using only a *web interface*, directly connected to the remote instance of the application through a noVNC server. This paper considers a further evolution of such approach, leveraging the computing continuum to distribute the applications across the entire infrastructure, according to specific execution constraints (i.e., power consumption).

In addition to the *web interface*, such distributed system requires additional computing resources also on the server side for the control plane logic to ensure the correct execution. Specifically, the *orchestrator* logic can be shared among multiple devices (i.e, the control plane can be executed on a small subset of machines), whereas an additional layer of *fluid substrate* must be introduced to guarantee the dynamic infrastructure re-configuration.

5. EXPERIMENTAL EVALUATION

This section presents a preliminary assessment of the potential energy savings of the cloudified approach when used in a realistic use case. Specifically, devices are modeled, based on the performance and power consumption metrics described in Section 3, to better assess the heterogeneity of computing resources.

The purpose of this evaluation is to investigate the possible benefits of the cloudified approach, enabled by the consolidation of the computing resources in the continuum. In this respect, we implemented a scheduling algorithm based on a brute-force exhaustive search, which guarantees the capability to find the best job placement with respect to the overall power consumption.

[3]https://medium.com/the-liqo-blog/liqo-in-production-at-turin-polytechnic-20ed71dca475.

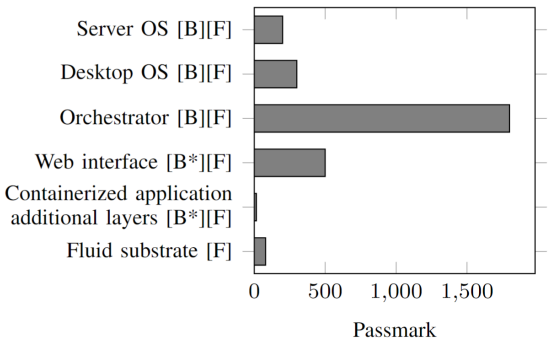

Figure 3. Passmark requirements of the different components of the fluid solution presented in Fig. 2.

5.1. Experiment Setup

The computing continuum is evaluated against the "traditional" workload execution, referred as *baseline*, in which applications are executed directly on the end-user terminals or, in case of lack of computing resources, on the dedicated set of servers that are statically configured on purpose. While in the case of the local execution, the energy consumption is purely due to the reference application (plus the power required to run the physical machine), measurements with the remote execution always account for the web interface to connect to the remote application and the additional requirements of the containerized workload (this applies also in case of "traditional" workload execution on remote servers). Instead, the OS server-side overhead is always present, as well as the requirements of a generic orchestrator (i.e., Kubernetes in our case), whereas the additional fluid substrate is present only in the continuum approach.

The requirements of the fluidified approach (detailed in Section 4) are summarized in Fig. 3 in terms of processing power. Particularly, [B] identifies processing components present in the baseline ([B*] applies only if static server remotization is present), while [F] highlights components that must be considered in the fluid approach. The web interface overhead is evaluated by measuring

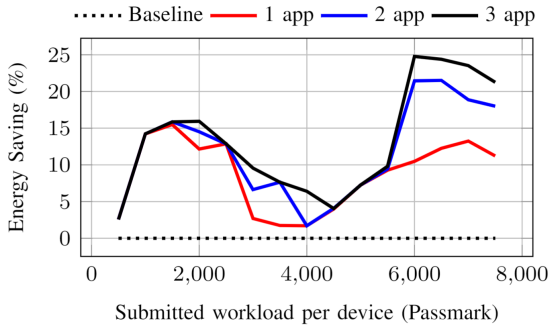

Figure 4. Power consumption saving with respect to the baseline increasing the submitted workload.

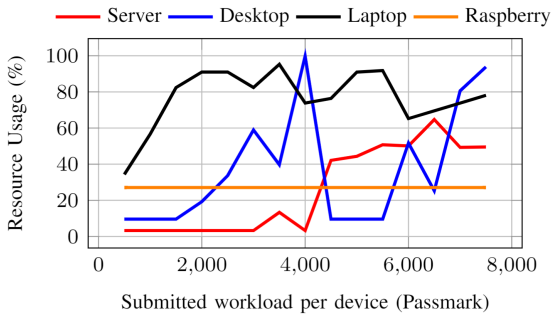

Figure 5. Device resource usage increasing the submitted workload.

the Google Chrome CPU requirements to connect to the remote instance. Specifically, the resource usage of the client front-end never exceeds 500 Passmark, corresponding to the 7% of the CPU in our desktop (as a reference, the desktop OS settles at around 300 Passmark, ≈4% CPU). Instead, the overhead for the orchestrator can be evaluated by averaging the computing requirements on a given period, and results to be about 1800 Passmark (≈1% of CPU usage of the server). Finally, the fluid substrate includes the requirements for the Liqo control plane logic,[4] the selected lightweight framework to create and manage the continuum infrastructure, which requires ≈80 Passmark.

5.2. Fluid workload distribution

We modeled an infrastructure based on a realistic university environment, which is composed of 15 user terminals (respectively, 5 desktop computers, 5 laptops, and 5 raspberry PIs, the same depicted in Fig. 1), to replicate the possible interactions of the end-users with the infrastructure, and 2 servers (server#1 in the same figure), to provide additional computing power and to cope with more demanding applications. We simulate 15 users, each one submitting the same workloads to the infrastructure (simulating end-user requests to execute applications), which are modeled based on the Passmark described in the previous section. Specifically, we increased the workload from 500 up to 7500 Passmark.[5]

Results in Fig. 4 compare the overall power consumption of the computing continuum against the baseline. Specifically, the continuum is evaluated in three different configurations (i.e., 1-2-3 app/user) representing the number of applications deployed by each user (hence, requested on each device), while the value on the x axis represents the cumulative workload deployed for every device (e.g., $x = 2000$ accounts for 1 application of 2000 Passmark, 2 applications of 1000 each or 3 applications of 666 each). The proposed simulation raises two considerations: (*i*) The computing continuum is always able to guarantee a non-negligible reduction in power consumption (i.e., approximately between 5% and 25% if we consider the most realistic case of 3 applications per user) (*ii*) The reduction of power consumption depends also on the processing requirements of the deployed applications. In fact, having multiple relatively-small workloads can drastically increase the number of potential locations for their optimal placement.

Fig. 5 depicts how the submitted workloads are spread across the available devices due to the combination of the fluid workload distribution and the heterogeneity of our hardware. The percentage of the utilization on the different devices in the infrastructure strictly depends on the cumulative demands of the deployed applications and is extremely variable, demonstrating the advantages of the rather aggressive optimization opportunities granted by the fluid approach. Overall, when the computing demand is rather low, desktops and laptops appear to guarantee the most efficient placement, whereas in the case of high computing demands servers can provide better scheduling opportunities. As a result, further optimizations can be implemented to make the most out of

[4] https://liqo.io.

[5] As a reference, 7500 Passmark corresponds to 7 CPU cores in our desktop.

the continuum by dynamically turning on/off the devices, depending on the applied workload.

6. CONCLUSIONS

With the introduction of the concept of *computing continuum*, highly geographically-dispersed computing resources can be logically aggregated in the so-called resource continuum, with the possibility to define resource-sharing policies between cloud, fog, and edge and guarantees in QoE for computationally-intensive latency-sensitive applications. As an emerging concept, the computing continuum is still widely unexplored and the focus has only recently shifted from the application QoE to the possible benefits for the distributed infrastructure.

This paper envisions a possible use case for the computing continuum, in which also end-user devices can share the unused processing capacities and the generated workloads may be executed on the different layers of the continuum, minimizing on the overall power consumption of the infrastructure. First, we propose a CPU-independent mechanism for performance characterization, which can describe the application QoE within the heterogeneous devices. Performance metrics are then correlated with power requirements to assess device efficiency; results suggest an extreme variability, depending on the applied workload. Then, we discuss a possible methodology to shift from current desktop applications to a fluid containerized environment, including a detailed analysis of the related requirements. Finally, extensive simulations validate the proposed continuum use case, focusing on the overall power consumption of the computing infrastructure. Our findings suggest that the capability of the continuum to leverage computing resources on all available devices enables a non-negligible reduction of the power consumption, despite the additional requirements of the continuum orchestration components. Indeed, small-medium-sized applications allow for better consolidation and, consequently, are best-suited for the continuum with a 13.3% reduction of power consumption on average, and 25% peaks. In addition, as the percentage of the utilization on the different devices of the infrastructure is extremely dynamic and strictly related to the cumulative demands of the deployed applications, it is possible to further optimize the overall power consumption, leveraging the dynamic powering of the unused devices, which will be investigated in our future works.

Acknowledgment

Stefano Galantino acknowledges the support from TIM S.p.A. through the PhD scholarship.

This work was partly supported by European Union's Horizon Europe research and innovation programme under grant agreement No 101070473, project FLUIDOS (Flexible, scaLable, secUre, and decentralIseD Operating System).

REFERENCES

1. S. Kaxiras and M. Martonosi, "Computer architecture techniques for power-efficiency," *Synthesis Lectures on Computer Architecture*, vol. 3, no. 1, pp. 1–207, 2008.
2. A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing," *The computer journal*, vol. 53, no. 7, pp. 1045–1051, 2010.
3. A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.
4. P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck, "Harnessing the computing continuum for programming our world," *Fog Computing: Theory and Practice*, pp. 215–230, 2020.
5. R. Kumar, M. Baughman, R. Chard, Z. Li, Y. Babuji, I. Foster, and K. Chard, "Coding the computing continuum: Fluid function execution in heterogeneous computing environments," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2021, pp. 66–75.
6. D. Rosendo, P. Silva, M. Simonin, A. Costan, and G. Antoniu, "E2clab: Exploring the computing continuum through repeatable, replicable and reproducible edge-to-cloud experiments," in *2020 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2020, pp. 176–186.
7. A. Luckow, K. Rattan, and S. Jha, "Pilot-edge: Distributed resource management along the edge-to-cloud continuum," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2021, pp. 874–878.
8. M. Iorio, F. Risso, A. Palesandro, L. Camiciotti, and A. Manzalini, "Computing without borders: The way

towards liquid computing," *IEEE Transactions on Cloud Computing*, pp. 1–18, 2022.

9. D. Balouek-Thomert, E. G. Renart, A. R. Zamani, A. Simonet, and M. Parashar, "Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows," *The International Journal of High Performance Computing Applications*, vol. 33, no. 6, pp. 1159–1174, 2019.
10. V. C. Pujol, P. Raith, and S. Dustdar, "Towards a new paradigm for managing computing continuum applications," in *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 2021, pp. 180–188.
11. S. Cao, X. Tao, Y. Hou, and Q. Cui, "An energy-optimal offloading algorithm of mobile computing based on hetnets," in *2015 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2015, pp. 254–258.
12. M. Deng, H. Tian, and B. Fan, "Fine-granularity based application offloading policy in cloud-enhanced small cell networks," in *2016 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2016, pp. 638–643.
13. Y. Zhao, S. Zhou, T. Zhao, and Z. Niu, "Energy-efficient task offloading for multiuser mobile cloud computing," in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2015, pp. 1–5.
14. D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A comparative analysis of simulators for the cloud to fog continuum," *Simulation Modelling Practice and Theory*, vol. 101, p. 102029, 2020.
15. H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.
16. J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon, and R. Buyya, "Cloudsimsdn: Modeling and simulation of software-defined cloud data centers," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 2015, pp. 475–484.
17. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011.
18. I. Lera, C. Guerrero, and C. Juiz, "Yafs: A simulator for iot scenarios in fog computing," *IEEE Access*, vol. 7, pp. 91 745–91 758, 2019.
19. C. Yang, X. Wang, Y. Luo, and G. Stefanek, "Overcome it lab challenge in covid-19 with windows-to-go," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2020, pp. 0791–0796.
20. B. Ryabko and A. Rakitskiy, "An analytic method for estimating the computation capacity of computing devices," *Journal of Circuits, Systems and Computers*, vol. 26, no. 05, p. 1750086, 2017.

Stefano Galantino Stefano Galantino is Ph.D. student at Politecnico di Torino, Italy. His research interests include Cloud-to-Edge continuum, sustainable and energy-efficient computing. Galantino received his M.Sc. degree in computer engineering from Politecnico di Torino, Italy. Contact him at stefano.galantino@polito.it.

Fulvio Risso Fulvio Risso is Associate Professor at Politecnico di Torino, Italy. His research interests include high-speed and flexible network processing, edge/fog computing, network functions virtualization. Risso received his Ph.D. in computer engineering from Politecnico di Torino, Italy. Contact him at fulvio.risso@polito.it.

Vlad Coroamă

Antonio Manzalini Antonio Manzalini is Senior Project Manager in TIM at Turin, 10149, Italy. His research interests include Edge Computing and Quantum Communications. Manzalini received his PhD in Computer Science from Sorbonne University. Contact him at antonio.manzalini@telecomitalia.it