IoT platforms and services configuration through parameter sweep: a simulation-based approach

Alessandro Barbieri[1], Fabrizio Marozzo[1] and Claudio Savaglio[2]

Abstract— **Due to their inherent cyber-physical features and high interactivity, IoT services exhibit performances which are simultaneously impacted by different orthogonal factors. Indeed, deployment settings (e.g., Cloud- or Edge-based scenarios, network bandwidth, hardware resource availability), algorithmic aspects (e.g., the specific algorithm used to solve a problem) and data features (e.g., packet size and rate) deeply affect the overall functioning of an IoT service and its compliance with specific requirements such as reactivity, reliability and efficiency. An accurate parameter sweep based on realistic IoT simulations is a viable, yet still unexplored, solution to obtain a full-fledged overview and specific evaluations about the performance of an IoT system under development. In such a direction, in this paper we present an approach for assessing Edge analytic in complex IoT scenarios through a parameter sweep analysis conducted through a simulation-based process, enabling a fine-grained modeling of hybrid IoT systems (both Cloud and Edge) of different scales (small, medium and large). Four typical IoT use cases (autonomous vehicles, smart healthcare, gaming, and industrial IoT) are presented to show the benefits of our approach in finding the right settings for configuring and running them. Indeed, the obtained results show that our approach concretely helps IoT developers in the challenging task of tuning the parameters' set so as to meet the given requirements, even in the case of large solution spaces and before the actual deployment phase.**

Index Terms— **Parameter sweep, IoT platforms, IoT services, Simulation, Edge Analytic.**

I. INTRODUCTION

In the last years the ability to produce and gather data has increased exponentially. Huge amounts of digital data are generated by and collected from a plethora of Internet of Things (IoT) devices, such as sensors, cams, in-vehicle infotainment and wearable devices [1], [2]. The huge amount of data generated, the speed at which it is produced, and its heterogeneity in terms of format (e.g., video, text, XML, JSON), represent a challenge to the current storage, process and analysis capabilities [3].

Specialized platforms are needed in the collection, integration and analysis of IoT data capable of offering advanced services that improve the quality of life of human beings. The existing IoT platforms are highly centralized and rely on Cloud solutions for data collection, integration and analysis. The data management and processing approach in the Cloud, however, can be ineffective in terms of response time, network traffic management and power consumption [4]. Moreover, in many medical and security applications it is crucial to offer low-latency and privacy-preserving services, as the delay caused by the transfer of data from an application to the Cloud (and vice versa) or malicious manipulations can cause strong disservices and even loss of life. For this reason, researchers and large IT companies have proposed in recent years the adoption of Edge Computing paradigm and the use of novel IoT solutions for processing data closer to where it is generated, thus reducing the network traffic, energy consumption, privacy risks and service delays [5].

Even the analysis of data generated by devices in IoT platforms and the extraction of valuable insight requires novel tools [6]. Advanced machine learning and data mining algorithms are continuously used for this purpose, i.e., they are able to discover patterns, correlations and trends that occur in the collected data [7]. In the IoT environment such algorithms are often performed on devices with limited resources in terms of memory, computing power, bandwidth and energy [8]. For this reason, it is necessary to find a good trade-off between the performance of the algorithms and the amount of resources needed for their execution [9].

In addition, due to the large scale, heterogeneity and complexity of IoT systems and networks, designing and testing IoT applications are still open issues. The Modeling and Simulation approach (M&S) results to be a powerful and flexible tool for reproducing and testing IoT systems and networks. This opportunity becomes essential when real tests are too expensive due to the size of the infrastructure to be managed and they may also take a long time due to the high combinations of configurations to be tested. In fact, factors such as the deployment settings (e.g., Cloud- or Edge-based scenarios, network bandwidth, hardware resource availability), algorithmic aspects (e.g., the specific algorithm used to solve a problem) and data features (e.g., packet size and rate) deeply influence the overall functioning of an IoT application and its compliance with specific requirements such as reactivity, reliability and efficiency. An accurate parameter sweep leveraging on a realistic IoT simulator is a viable solution to obtain a full-fledged overview and specific quantitative results about the expected operations and performance of an IoT system under development. However, so far, parameter sweep is almost exclusively used over numerical models for AI algorithms hyper-parameters tuning and scheduling tasks in grid environment.

That said, in this paper we present a novel approach

*This work was partially supported by the Italian MIUR, PRIN 2017 Project "Fluidware" (CUP H24I17000070001).

[1]Alessandro Barbieri and Fabrizio Marozzo are with the Department of Computer Science, Modeling, Electronics and Systems Engineering (DIMES) of the University of Calabria, Italy. abarbieri@dimes.unical.it; fmarozzo@dimes.unical.it

[2] Claudio Savaglio is with the Institute for High Performance Computing and Networking (ICAR) of National Research Council (CNR) of Italy claudio.savaglio@icar.cnr.it

for assessing the performance of Edge analytic in complex IoT scenarios through a parameter sweep analysis conducted through a simulation-based process. At the best of our knowledge, indeed, there is no similar work enabling the configuration of IoT platforms and services through a comprehensive and systematic evaluation of the numerous, possible, design choices (only in [10] this research line is outlined as possible future work). As well as original, the presented approach is also highly versatile, being domain- and simulator-neutral: nevertheless, we advice *EdgeCloudSim* simulator [11] for its fine-grained modeling of IoT systems deployed according to different scale (small, medium and large) and to different infrastructures (Cloud-only, Edge-only and Cloud/Edge). Four typical IoT use cases (industrial IoT, smart healthcare, autonomous vehicles and mobile gaming) are are outlined to show how our approach works and its benefits in supporting design choices and configuration of execution environments. The results obtained *(i)* highlight the suitability of Cloud-based deployments for high resource-demanding IoT services and of Edge-based ones for large-scale scenarios; and *(ii)* allow managing a good compromise among responsiveness, reliability and efficiency on the basis of service needs. By providing such insights, hence, our approach concretely helps IoT developers to determine the system configuration better matching with the given requirements, even in the case of large solutions spaces, before its actual deployment.

The structure of the paper is as follows. Section II discusses background and related work. Section III describes the proposed approach. Section IV presents the case studies and the experimental evaluation through a large set of simulations. Finally, Section V concludes the paper.

II. Background and Related Work

With the increasing popularity of IoT technologies, several novel research projects have been carried out for obtaining valuable insight from large data generated by IoT devices. Big data analytics in IoT requires technologies and tools that can transform large amounts of structured, unstructured, or semi-structured data into more understandable data ready to be analyzed. Advanced machine learning and data mining algorithms are used to discover patterns, trends, and correlations over a variety of time horizons in the data [6]; however, these tasks are executed by devices characterized by limited resources, such as memory, processing, bandwidth and energy [8]. Thus, it is necessary to find the right compromises between performance (e.g., accuracy) and amount of resources required for computation. There are many papers published in this area, i.e. propose data mining/machine learning solutions in the IoT area, which are described below.

For example, Shanthamallu et al. [12] covered supervised and unsupervised learning algorithms in the IoT area, and outlined several applications including pattern recognition, anomaly detection, computer vision, speech processing.

Alam et al. [13] examined the applicability of eight data mining/machine learning algorithms for IoT data. These include, among others, the deep learning artificial neural networks (DLANNs), which build a feed forward multi-layer artificial neural network (ANN) for modelling high-level data abstractions. The achieved results on real IoT datasets show that decision tree algorithms have better accuracy, are memory efficient and have relatively higher processing speeds, while ANNs and DLANNs can provide highly accurate results but are computationally expensive.

Hussain et al. [14] highlighted how the security and privacy solutions of the IoT platforms suffer from a series of problems related to the dynamic nature of networks. The paper highlights how some of the gaps in current IoT platform solutions can be overcome by using advanced machine learning algorithms. Machine learning techniques can be used to enable the IoT devices to adapt to their dynamic environment, by supporting self-organizing operation and also optimize the overall system performance by learning and processing statistical information from the environment.

For a correct evaluation of machine learning algorithms, realistic tests are needed on an IoT infrastructure consisting of a large number of entities/devices connected to each other and running different types of software. Designing and validating IoT infrastructures is still a complex issue due to the even-increasing complexity and technological improvement that makes them difficult to study and evaluate. M&S plays an essential role in managing this complexity since it allows to imitate the structure and behaviour of a complex system during its lifecycle.

D'Angelo et al. [15] introduced main issues on the simulation of IoT infrastructure, and discussed a new combination of M&S techniques to enhance scalability and permit the real-time execution of massively populated IoT environments (e.g., large-scale smart cities).

Bosmans et al. [16] presented an overview of the challenges that arise when testing large IoT applications at the system level. To overcome these challenges, the author proposed a hybrid simulation-based testing technique that allows to evaluate IoT infrastructure by orchestrating a real-time interaction between real-life and virtual local IoT entities so as to detect emergent behaviors.

Concerning the tools and software, there are many simulators that have been proposed in recent years in the IoT field [17]. Among them, EdgeCloudSim [11] is a CloudSim extension [18] that provides a modular architecture for supporting a variety of crucial functionalities such as network modeling specific to WLAN and WAN, device mobility model, realistic and tunable load generator to support Edge/Cloud continuum. The rich set of built-in models can be customized, extended as well as integrated with external tools, for being realistically applied also in vertical domains or for introducing new performance metrics (e.g., energy consumption). The default simulations output, such as service time, service failure rate,network/server utilization, allow easily disclosing the particular impact of computation- and communication-related factors.

III. Proposed Approach

Functional but also not functional requirements (e.g., responsiveness, reliability, efficiency) deeply impact on the operation of IoT services and, therefore, they have to be simultaneously considered to make wise design choices. A surveillance camera correctly detecting car accidents but only after 1 hour from the event is useless; likewise, a fast but inaccurate accident detection would trigger too many alerts, thus forcing a human operator to continuously monitor the video. A lot of similar examples might be considered since we daily experience a plethora of smart objects failing in providing services which, instead, might be smoothly provided by fully-equipped machines (e.g., lagged multimedia streaming, imprecise GPS localization). It happens because the functioning of IoT services is a matter of balance between technical and business goals [9] and the management of this trade-off before the slow, error prone, and costly deployment phase, is a challenging development task: indeed, due to the high number of factors to be considered, there exists a plethora of possible configurations, few of them are eligible and these need to be accurately compared and evaluated. In this direction, we propose an approach that, according to the requirements of the specific scenario and its unrestricted parameters, allows outlining the most suitable configuration for a given IoT service by means of a parameter sweep activity conducted over the EdgeCloudSim simulator.

Parameter sweep allows to test a scenario numerous times with different parameter configurations, defined over specified ranges. Through an iterative process, the parameter sweep enables fine-tuning parameter values, exploring a wide solutions space, and calibrating simulations to data. In such a way, before the actual system deployment, its behaviors can be assessed in a variety of settings and the design choices can be systematically supported. Parameter sweep is mostly exploited to set the hyper-parameters of AI algorithms [19] and scheduling tasks in grid environments [20] by running a set of simulations over numerical models. However, for IoT services, the simulator upon which the parameter sweep is performed should allow a fine-grained modeling of algorithmic, data, and infrastructural aspects so as to provide realistic results and effective insights to the IoT developer. We therefore advice EdgeCloudSim which, better than other simulators currently available at the state-of-the-art, allows comprehensively analyzing IoT systems of different scales and provided with hybrid Edge- Cloud-based offloading strategies.

In particular, the approach we propose consists in modeling and simulating an IoT service over EdgeCloudSim, performing a parameter sweep analysis over the obtained results, and finally outlining the best configurations matching with the requirements initially identified. The approach is domain- and simulator-neutral, being easily adopted to any IoT service whose complexity and variety of design choices demand for a systematic analysis, i.e., the parameter sweep, rather than naive or approximate evaluations. In detail, our approach consists in:

1) a *preliminary analysis* of the scenario, aimed at ranking its main requirements according to their priorities;
2) a *modeling phase* or setup, defining the allowable ranges for the ”sweep variables”, namely, for the exchanged data (e.g., packet size), for the computational complexity of the algorithms (expressed in MI obtained by multiplying the execution time and the frequency in MIPS of the considered processing units) and for the infrastructural resources (by specifying the number and hardware features of the devices, the network bandwidth in Mbps, etc).
3) the *parameter sweep*, by considering the highly-ranked requirement as pivot around which let vary the other ones in the defined ranges. EdgeCloudSim allows considering non functional requirements like responsiveness, reliability and efficiency while a separate analysis might be performed to assess the fulfilment of functional requirements, e.g., accuracy or precision of a given algorithm.
4) gathering the obtained data, display them through charts or heat maps for simplifying their analysis and outlining the *better system configuration*; reiterate to the previous point in case of not satisfactory results.

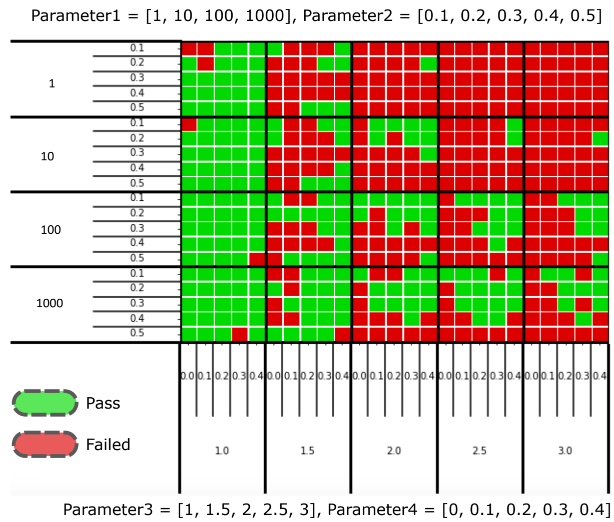

Fig. 1. Example of an heat-map gathering the results of a parameter sweep analysis

Fig. 1 reports an example of an heat map filled with the results of a parameter sweep, conducted over 4 parameters and colored according to two requirements which determine the IoT service failure (red boxes) or success (green boxes). Developers can exploit it or analyze the charts, as we have done next, for taking their design choices.

IV. USE CASES AND SIMULATIONS

To showcase of our approach, we consider four use cases (UCs) with constraints and parameters that can easily be traced back to typical application scenarios in the IoT field: *Industrial IoT* (UC1), *smart healthcare* (UC2), *autonomous vehicles* (UC3) and “mobile gaming” (UC4). The goal is to evaluate the reactivity, reliability and use of resources in different configurations for disclosing the eligible ones and obtaining meaningful insights.

The main parameters used for configure the simulations of each use case are reported in Table I. For all of the UCs

TABLE I

MAIN PARAMETERS OF THE PRESENTED USE CASES.

Parameter	UC1	UC2	UC3	UC4
Avg. packet size (KB)	{20-2000}	{20-2000}	Fixed 200	{20-2000}
Task length (MI)	{250-8000}	Fixed 1000	{250-8000}	{250-8000}
Device population	Fixed 100	{100-500}	{100-500}	Fixed 100
Off-loading strategy	Cloud OR Edge	Cloud OR Edge	Cloud OR Edge	Cloud AND Edge

we have considered a realistic setting comprising an *AWS r5a.xlarge* instance (3.9GHz quadcore- 106926 MIPS) as Cloud Server and five *Raspberry Pi 3+* (1.2 GHz quadcore- 2451 MIPS) as Edge Servers. For the network *bandwidth*, instead, we have set a value of 100 Mbps, both for WLAN (i.e., device-Edge Server connection) and WAN (i.e., device-Cloud Server connection) as used in other related works [11]. The *average packet size* considers both uploaded and downloaded data and in a range (20-2000 KB) wide enough to model heterogeneous payload, from text to low quality images, audio recording and video. Likewise, the *task length*, namely the algorithm complexity, varies from 250 to 8000 MI for modelling both lightweight or heavyweight tasks. The *device population* has been set for considering small, medium and large scenario, up to 500 devices (threshold for smooth simulations) while the *offloading strategy* contemplates that task are executed exclusively on the Cloud or on the Edge servers or on both with different percentages according to a stochastic distribution.

A. UC1: Real-time asset monitoring for Industrial IoT

A fixed number of devices (e.g., sensors embedded within a conveyor belt) might forward data of different sizes (e.g., vibration data with variable resolution or in batch) for feeding algorithms of different complexities (from lightweight classifiers to heavyweight long-short-term-memory networks).

Results reported in Fig. 2(a) shows that, for a population of 100 sensors deployed on a conveyor belt, the service time exhibits markedly different trends between the Edge- and the Cloud-based scenarios. In detail, the service time is independent of the packet size but just of task length in the Edge-based scenario; conversely, all the Cloud-based configurations exhibit a clear correlation with the packet size, independently from the task length. The same trends characterize the service fails, as reported in Fig. 2(b). The underlying motivation is that, as shown in Fig. 2(c), the network is not overloaded (indeed, the contribution of network to the service time is negligible) even with a packet size of 2,000 KB, but the Edge Servers, due to their limited CPUs, suffer heavyweight tasks. Conversely, in the centralized scenario, a bottleneck effect slows down the network and leads to service failures, independently from the task length: indeed, Cloud servers' networking time of Fig. 2(c) exactly overlaps the service time and exhibits a very similar trend to service failure. The insights from this simulation are that both responsiveness and reliability are exclusively impacted by the task length in the Edge-based scenario and by the packet size in the Cloud-based one. If we consider, for example, a threshold value of 5 seconds for the service time and 20% for the service fails, the only suitable configurations are (250, 500, 1,000 MI) for Edge and (20, 200, 500 KB) for the Cloud. Therefore, if we aim at better performances, we should increase, respectively, the CPU powers and the network bandwidth.

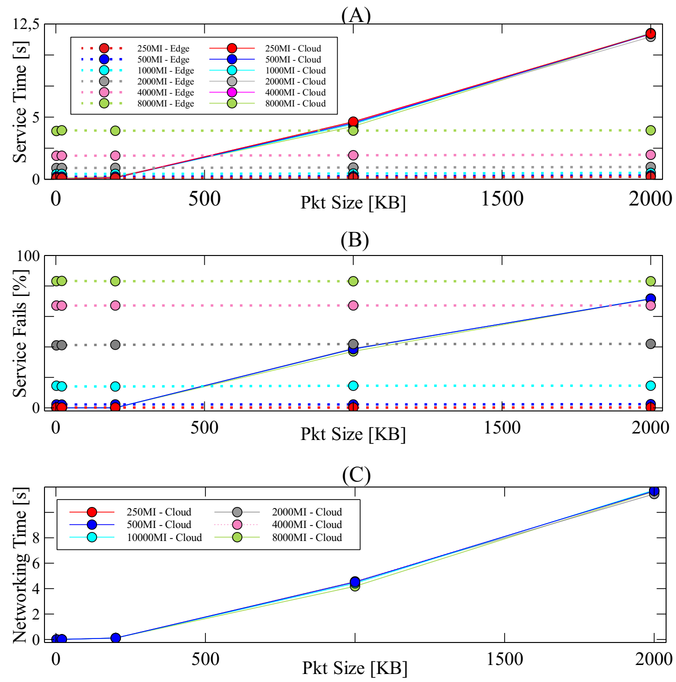

Fig. 2. Use Case 1: Industrial IIoT

B. UC2: Stroke detection for smart healthcare

Given a legacy system for stroke detection running a particular algorithm, we aim at performing a scalability test (i.e., how many users can be simultaneously monitored) with varying input sizes (e.g., heartbeat and/or blood pressure and/or oxygenation).

The results of Figs. 3(a) and 3(b) show, with a fixed task length of 1000 MI, Edge-based configurations scale better than Cloud-based ones, exhibiting horizontal lines instead of piece-wise linear curves. In particular, the service time is constant for all the configurations in Edge-based scenario, despite both the number of users and packet size change; the service time raises, instead, in all the Cloud-based configurations, mostly due to the packet size and its bottleneck effect on the network, also causing the majority

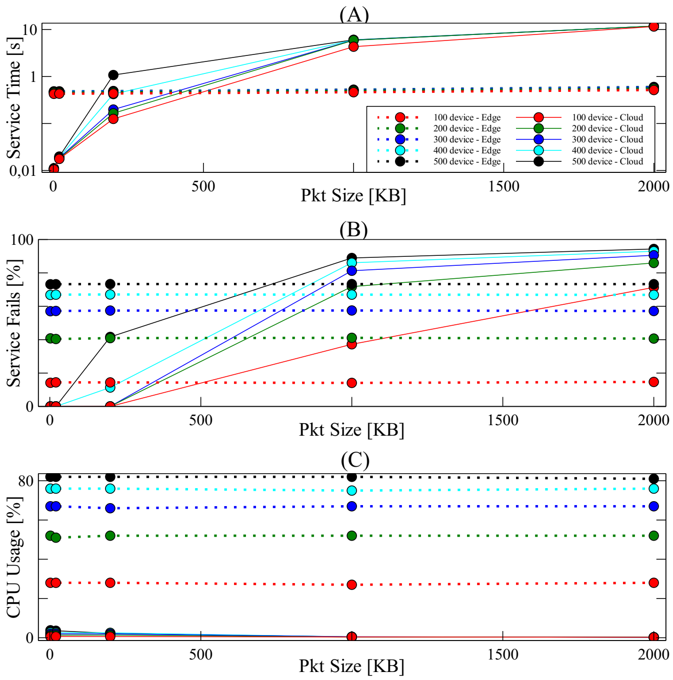

Fig. 3. Use Case 2: Smart Healthcare

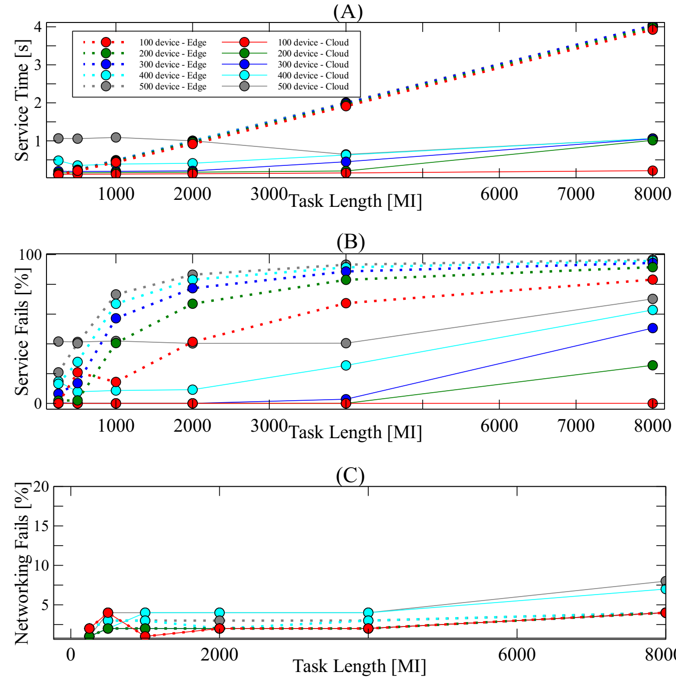

Fig. 4. Use Case 3: Autonomous Vehicles

of service fails. This hypothesis is confirmed by Fig. 3(c) which highlights that the CPU usage of Cloud Servers is negligible while it is relevant for Edge Servers: due to such high utilization, the Edge-based scenario can hardly support the simultaneous deployment of other CPU-intensive IoT services. For example, given the critical nature of this use case, we might tolerate up to 1 sec of service time and below 15% of service failures: such constraints would limit the suitable settings to 100 devices for the Edge-based scenario and 400 device - 200 KB for the Cloud-based one.

C. UC3: Object recognition for autonomous vehicles

Given a certain data to be exchanged within packet of fixed size (e.g., same video-cameras mounted on each car forwarding video of equal quality) and might be produced by a variable number of devices (e.g., the autonomous cars we consider) and elaborated according to algorithms of different complexities (convolutional networks, histogram of oriented gradients, etc.) based on their goal (from simple road sign detection to advanced event recognition, like car accidents).

As shown also in the previous UCs, Cloud-based scenarios mostly suffer the higher number of users and the subsequent network traffic while the Edge-based ones the increasing task length. In this particular UC where the packet size is fixed and limited to 200 KB, the Cloud-based settings outperform the Edge-based ones in terms of service time and service fails for every configuration with a task length exceeding the 2,000 MI, as shown respectively in Figs. 4(a) and (b). Moreover, in both cases, the packet size weakly impacts on the service reliability, thus the chosen algorithm might be as lightweight as possible. For example, if we limit the service responsiveness to 1s and its reliability to 30%, the only allowable Edge-based settings contemplates algorithms of 250 and 500 MI, while Cloud Servers can handle up to 400 devices and task length of 4000 MI.

D. UC4: Augmented Reality for Mobile Gaming

The task-offloading strategy can be hybrid (pool of service requests shared by Edge or Cloud servers), the data exchanged in different packet sizes (e.g., users with different smartphones might upload videos of different quality and might receive different kind of contents) and the algorithms more or less complex, according to the purchased plane (e.g., levels of a neural network for a higher precision).

In Fig. 5(a) we can clearly see how the task-offloading impacts on the service time in a setting which contemplates 100 devices, a task length of 1000 MI and a variable packet size values; in Fig. 5(b), instead, the service time is evaluated with respect to 100 devices exchanging data packets of 200 KB according to different task length values; finally, in Fig. 5(c), we inspect how stressed the CPU of Cloud and Edge servers are in executing tasks of 2000 MI on behalf of 100 devices. Main findings are manifold: 200 KB is the packet size value representing a reversal point for the service time trends; even with the minimum Cloud offloading percentage of 10% the Edge-based scenario provides service times comparable (less than 1 second of difference) to the Cloud-based ones; the Edge-server utilization drastically decreases only if the 60% are off-loaded to the Cloud server, which conversely results poorly under-loaded. Starting from these insights, developers can precisely manage the trade-off between an Edge- and a Cloud-based deployment and also individuate the most suitable server configuration for the IoT service under development.

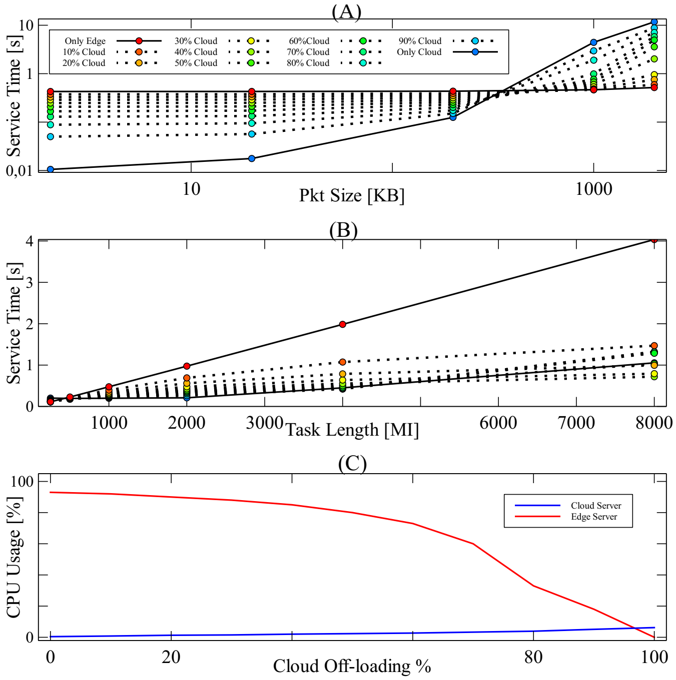

Fig. 5. Use Case 4: Mobile Gaming

V. CONCLUSIONS

IoT developers continuously face many design choices, which can deeply impact on the overall system's operations and performance. An accurate parameter sweep analysis properly conducted over a realistic IoT simulator can represent, therefore, a viable solution to support decision making by preliminary assessing the IoT system under development.

Therefore, in this paper we have presented a simulator-based and domain-neutral approach, first in its kind, to evaluate the simultaneous and mutual impact of different deployment settings, algorithmic aspects and data features on the performance of an IoT service. We have performed a parameter sweep analysis over EdgeCloudSim and we get evidence that the responsiveness, reliability and efficiency of both Cloud- and Edge-based deployments markedly vary according to the specific parameter settings. In particular, Cloud-based settings outperform in resource-intensive IoT applications while Edge-based settings avoid bottleneck effects on the network and provide greater scalability. However, this is just a thumb rule: indeed, the optimal configuration for a given IoT system can be established only case-by-case through a comprehensive simulation-based analysis.

As future work, we aim at inserting the proposed approach within a full-fledged modeling and simulation framework. Doing so, the configuration of an IoT system according to its many requirements and constraints can be treated as an optimization problem and its best solution automatically provided to the IoT developer for a final validation.

REFERENCES

[1] P. Escamilla-Ambrosio, A. Rodríguez-Mota, E. Aguirre-Anaya, R. Acosta-Bermejo, and M. Salinas-Rosales, "Distributing computing in the internet of things: cloud, fog and edge computing overview," in *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop in Tlalnepantla, Mexico*. Springer, 2018, pp. 87–115.

[2] G. Fortino, C. Savaglio, G. Spezzano, and M. Zhou, "Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[3] L. Belcastro, F. Marozzo, and D. Talia, "Programming models and systems for big data analysis," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 34, no. 6, pp. 632–652, 2019.

[4] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of things: Integrating internet of things and cloud computing and the issues involved," in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th-18th January, 2014*. IEEE, 2014, pp. 414–419.

[5] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[6] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big iot data analytics: architecture, opportunities, and open research challenges," *ieee access*, vol. 5, pp. 5247–5261, 2017.

[7] D. Talia, P. Trunfio, and F. Marozzo, *Data Analysis in the Cloud: Models, Techniques and Applications*, 2015.

[8] S. Zahoor and R. N. Mir, "Resource management in pervasive internet of things: A survey," *Journal of King Saud University-Computer and Information Sciences*, 2018.

[9] C. Savaglio and G. Fortino, "A simulation-driven methodology for iot data mining based on edge computing," *ACM Trans. Internet Technol.*, vol. 21, no. 2, Mar. 2021. [Online]. Available: https://doi.org/10.1145/3402444

[10] Y.-W. Lin, Y.-B. Lin, and T.-H. Yen, "Simtalk: Simulation of iot applications," *Sensors (Basel, Switzerland)*, vol. 20, no. 9, 2020.

[11] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018.

[12] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and iot applications," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017, pp. 1–8.

[13] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter internet of things (iot)," *Procedia Computer Science*, vol. 98, pp. 437–442, 2016.

[14] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[15] G. D'Angelo, S. Ferretti, and V. Ghini, "Simulation of the internet of things," in *2016 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, 2016, pp. 1–8.

[16] S. Bosmans, S. Mercelis, J. Denil, and P. Hellinckx, "Testing iot systems using a hybrid simulation based testing approach," *Computing*, vol. 101, no. 7, pp. 857–872, 2019.

[17] A. Markus and A. Kertesz, "A survey and taxonomy of simulation environments modelling fog computing," *Simulation Modelling Practice and Theory*, vol. 101, p. 102042, 2020.

[18] 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.

[19] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning *et al.*, "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1407–1416.

[20] H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman, "Heuristics for scheduling parameter sweep applications in grid environments," in *Proceedings 9th Heterogeneous Computing Workshop (HCW 2000)(Cat. No. PR00556)*. IEEE, 2000, pp. 349–363.