



Review

Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics

Majid Ashouri *, Fabian Lorig , Paul Davidsson and Romina Spalazzese

Department of Computer Science and Media Technology, Malmö University, 20506 Malmö, Sweden; fabian.lorig@mau.se (F.L.); paul.davidsson@mau.se (P.D.); romina.spalazzese@mau.se (R.S.)

* Correspondence: majid.ashouri@mau.se

Received: 30 September 2019; Accepted: 6 November 2019; Published: 8 November 2019



Abstract: The deployment of Internet of Things (IoT) applications is complex since many quality characteristics should be taken into account, for example, performance, reliability, and security. In this study, we investigate to what extent the current edge computing simulators support the analysis of qualities that are relevant to IoT architects who are designing an IoT system. We first identify the quality characteristics and metrics that can be evaluated through simulation. Then, we study the available simulators in order to assess which of the identified qualities they support. The results show that while several simulation tools for edge computing have been proposed, they focus on a few qualities, such as time behavior and resource utilization. Most of the identified qualities are not considered and we suggest future directions for further investigation to provide appropriate support for IoT architects.

Keywords: Internet of Things; edge computing; simulation tools; quality characteristics; metrics; ISO/IEC 25023

1. Introduction

The rapid development of the Internet of Things (IoT) affects nearly all aspects of society, including industrial automation; building, as well as HVAC systems; smart metering; and health care [1]. An IoT system often generates huge amounts of data, which requires heavy computational processing [2]. Cloud computing offers attractive computational and storage solutions to cope with these issues, however, cloud-based solutions are often accompanied by drawbacks and limitations, for example, latency, energy consumption, privacy, and bandwidth [3]. Edge computing, in which computation and storage are done closer to where the data is generated, could help to address these challenges by meeting specific requirements, such as low latency or reduced energy consumption [4].

The design of an IoT system entails several different aspects, in particular the hardware infrastructure (including servers, gateways, communication networks, sensors, and actuators), the application software, and the decision of where to deploy the application software components within the hardware infrastructure. Currently, the most common IoT system approach is to do as much computing and storage as possible in the cloud, however, the combination of edge and cloud computing in edge computing architectures provides new opportunities for the design of efficient IoT systems [5]. Because an IoT system often has to meet many different and possibly conflicting requirements, it is a challenge to design its architecture including identifying the optimal deployment for each component within the resulting edge-cloud continuum [6,7]. Moreover, the optimality of a deployment is often related to the system's scalability and a tradeoff must be made. Figure 1 illustrates the software component deployment problem, which concerns the placement of the computation and storage tasks the system should execute.

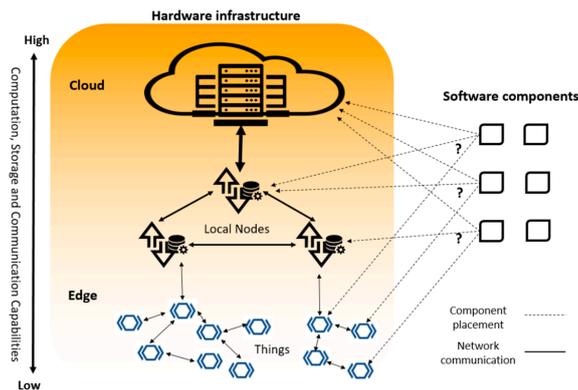


Figure 1. The software component deployment problem in Internet of Things (IoT) system design. Each component may be deployed either in the cloud, in local nodes (e.g., local servers or gateways), or on the things.

In case the hardware infrastructure is not readily available, the IoT architect not only needs to solve the software component deployment problem, but also needs to determine the architecture, as well as the entities (things, local nodes, and sometimes even central servers) of the hardware infrastructure. The architect must also ensure that the resulting IoT system is capable of efficiently fulfilling its intended purpose while achieving the desired quality. The quality of an IoT system is typically seen as the degree to which it satisfies the needs of its various stakeholders. The stakeholders' needs can be categorized into a set of quality characteristics (QCs), such as functional suitability, performance, usability, reliability, security, and maintainability as listed by ISO/IEC 25010 [8]. Here, objective quality metrics are also required, from which the quality of the system can be measured.

To address the challenges of modeling and investigating different aspects of complex systems, computer simulation has proven to be a suitable technique [9]. A large number of simulators exist that enable and facilitate the modeling and assessment of different aspects of IoT systems, for example, load balancing or network utilization. However, they only provide basic predefined attributes for the investigation of the IoT system's quality [10] and the integration and use of more sophisticated QCs is often not supported. In addition, simulation has been broadly used to analyze the quality of an IoT system in edge computing before its implementation [11]. Nevertheless, limitations in the extendibility of simulators apply and, as well, a fragmentation of the research regarding the relation between edge computing simulators and QCs can be observed.

In the case of software deployment problems for IoT systems (see Figure 1), an initial step towards understanding how simulation can support IoT system designers is to analyze the available simulators with respect to supported qualities that can be assessed. Thus, in this study, we investigate currently available edge computing simulators and study the support they provide for modeling and analyzing qualities that might be relevant for IoT architects while designing IoT systems. To achieve this, first, we systematically investigated state of the art papers to derive the key qualities and related metrics for IoT systems using edge computing. Accordingly, we extracted and categorized the metrics to form a list of evaluated qualities by other researchers. Finally, by studying current edge computing simulators, we investigated which of the extracted qualities are addressed by the simulators. In summary, the main contributions of this study are:

- A set of the relevant qualities for edge computing simulation based on a systematic review of the literature;
- An analysis of the Edge computing simulators in terms of which of the identified qualities and the related metrics they support.

The remainder of this paper is organized as follows: In Section 2, related work is discussed; in Section 3, the search method, the results of the literature review, and the identified key qualities

are described; in Section 4, an analysis of edge computing simulators is presented with respect to the identified qualities; in Section 5, we discuss the findings and shortcomings of existing simulators; and in Section 6, we conclude and summarize this work.

2. Related Work

The idea of using processing and storage resources at the “edge” of an IoT system and providing the required capabilities closer to the source of data has been proposed recently [4]. In some works, the term “fog computing” is also used to refer to systems where edge devices carry out a substantial amount of computation and storage. In this paper, we use the term “edge computing” to include any computing and storage resources (including fog computing) along the path between sensors or actuators and cloud servers. As mentioned, edge computing performs well as compared with cloud computing with respect to several IoT system QCs such as latency, energy consumption, privacy, and bandwidth. The benefits of edge computing have been investigated in different applications such as deep learning [12] and multimedia delivery [13]. However, cloud computing has many strengths due to its huge processing and storage capacity. To combine the strengths of edge and cloud computing, hybrid edge-cloud approaches have been proposed [14]. The software component deployment problem in a hybrid edge-cloud approach has been investigated in a number of studies. For instance, Sarkar et al. [15] focused on a comparison between edge and cloud computing by considering generalized models of delay, power consumption, and cost to evaluate performance of IoT applications with and without the existence of edge computing. On the basis of a simple simulation setup, it was concluded that edge computing reduced energy consumption, latency, and cost. Similarly, Li et al. [5] studied the performance of hybrid edge-cloud architectures based on latency and energy consumption. However, the tradeoffs with conflicting QCs were not considered in these studies.

Computer simulation has been proven to be a suitable technique to use for investigating different aspects of complex systems, such as service configuration, resource placement, or management strategies in IoT systems. To this end, several simulators (simulation frameworks) were proposed that address specific requirements for the application in cloud computing, for example, modelling of physical networks and virtual topologies [16] or emulating IoT environments [17]. Additionally, there are a few simulators designed for the simulation of edge and cloud computing, for example, iFogSim [18], EdgeCloudSim [19], and IOTSim [20]. Many edge computing simulators are extensions of the comprehensive CloudSim simulator [21] which focuses on the simulation of cloud computing infrastructures. Svorobej et al. [11] conducted a survey study to identify and analyze edge computing simulators in terms of implemented features such as mobility, scalability, as well as infrastructure and network level modeling, however, they did not study the support of QCs in the simulators. To investigate and compare the performance of different IoT architectures by means of simulation, and in terms of QCs relevant for the application at hand, quantitative metrics must be defined. As outputs, most frameworks focus on general network theoretical metrics, such as latency (end-to-end response time), network congestion, or energy consumption [11,17]. To investigate and assess edge-cloud architectures, as well as provide design decision support, the application of such standard metrics is not sufficient. For many IoT systems, the QCs required to adequately support and assess design decisions include, for example, accuracy, availability, and privacy. Existing frameworks, however, neither allow for capturing these QCs nor provide functionalities for specifying their own metrics.

3. Identification of Relevant Qualities

The distributed deployment of IoT applications in an edge computing-based architecture is a challenging decision. From a non-functional point of view, several quality attributes should be taken into consideration for the decision. Simulators can support architects for choosing the best option in the continuum of physical resources from the edge to the cloud by providing various quality measurements, however, the question is “What are the main qualities that should be considered by simulators to provide a better overview of the problem?”. As stated by Blas et al. [22] (p.10), “an attribute of a

software product needs to be described by a quality sub-characteristic and has to be measured by a metric”, which shows the relationship between metrics and qualities. Accordingly, to answer the question, we rely on the metrics considered in the literature, and, then, derive the relevant qualities from the extracted metrics. In the following, first, we explain the literature study for extracting the metrics, and, then, describe the derived qualities from the metrics.

3.1. Literature Study and Metric Extraction

To understand what metrics related to each quality have been covered by the literature, we conducted a systematic search to find relevant papers. For this purpose, we followed a guideline proposed by Petersen et al. [23], and used the following search string to capture the papers: (“fog computing” OR “edge computing”) AND (“IoT” OR “Internet of Things”) AND (“metrics” OR “attributes” OR “criteria”). For our search, we applied the search string for well-known databases, i.e., Scopus, IEEE explore, Science Direct, and Web of Science. After removing duplicates and book chapters, we found 92 papers, from which 44 relevant papers were identified for final studying.

The next step was to extract the metrics by considering the quantitative measurement of qualities in the papers. In this study, to rely on a solid base for initial classification of the metrics, we utilized the ISO/IEC 25010 quality model [8]. It consists of eight main quality characteristics, as shown in Figure 2. The definition of each quality (based on the ISO/IEC) is explained as follows:

- **Functional suitability:** The degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.
- **Performance efficiency:** The performance relative to the amount of resources used under stated conditions.
- **Compatibility:** The degree to which a product, system, or component can exchange information with other products, systems, or components, and/or perform its required functions while sharing the same hardware or software environment.
- **Usability:** The degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.
- **Reliability:** The degree to which a system, product, or component performs specified functions under specified conditions for a specified period of time.
- **Security:** The degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.
- **Maintainability:** The degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in the environment, and in requirements.
- **Portability:** The degree of effectiveness and efficiency with which a system, product, or component can be transferred from one hardware, software, or other operational or usage environment to another.

We classified the extracted metrics for each quality based on the ISO/IEC quality model.

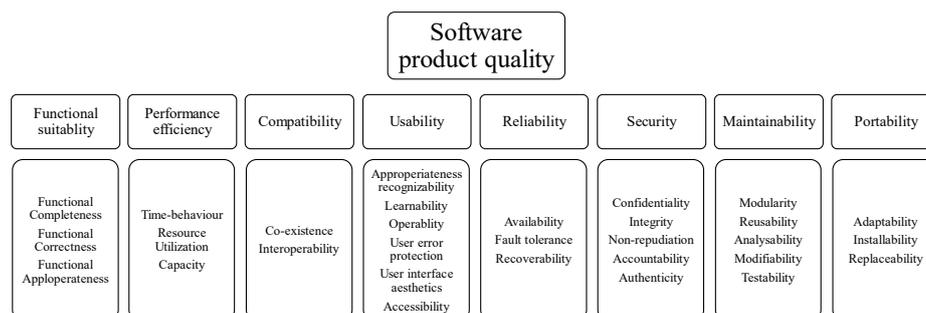


Figure 2. ISO/IEC 25010 quality model.

3.2. Deriving Qualities and Metrics

Although the ISO/IEC 25010 quality model provides a base to identify the relevant qualities, the model is quite abstract for the domain of edge computing, and a finer-grained level that is related metrics for each quality is required. For this purpose, we used the ISO/IEC 25023 standard [24] that provides a set of measures (metrics) for each quality, however, not all of the metrics in the standard are applicable to edge computing. Thus, to identify the relevant metrics from ISO/IEC 25023, we relied on the extracted metrics from the literature. Before deriving the final set of qualities and the related metrics, we found that some extracted metrics from the literature were not appropriate to be used on generic simulators for edge computing, for example, as they were application-specific or could not be measured by simulators. Accordingly, the following criteria were used to exclude the metrics:

- Application-specific metrics, such as decision time of an application, ECG waveform, or wireless signal level;
- Not possible to measure by simulation, for example, user experience can be measured by survey;
- Can be calculated by other metrics, such as the average slowdown of an application, that is average system waiting time divided by its service time, can be calculated by combining other metrics;
- Abstract metrics without detailed information regarding how to measure it, such as degree of trust;
- Metrics that are input for the simulation, such as processing capacity of a device.

The list of the extracted metrics, after applying the filtering, is available in Appendix A, Table A1.

Finally, by considering the similarities among the extracted metrics from the literature, we identified relevant qualities based on ISO/IEC 25010, and a set of metrics based on the ISO/IEC 25023 standard. We also added four more refined metrics. Three of the refined metrics were identified under mean turnaround time (processing delay, network delay, and storage read/write delay). We found them important metrics, since they are frequently used by the literature. Moreover, energy consumption was another metric frequently used by the literature, but it was not among the metrics in the ISO/IEC 25023 standard. The identified qualities and the related metrics are shown in Figure 3. In the following, a brief description of each metric is provided:

- **Mean response time:** The mean time taken by the system to respond to a user or system task.
- **Mean turnaround time:** The mean time taken for completion of a job or an asynchronous process.
 - **Processing delay:** The amount of time needed for processing a job.
 - **Network delay:** The amount of time needed for transmission of a unit of data between the devices.
 - **Storage read and write delay:** The amount of time needed to read/write a unit of data from disk or long-term memories.
- **Mean throughput:** The mean number of jobs completed per unit time.
- **Bandwidth utilization:** The proportion of the available bandwidth utilized to perform a given set of tasks.
- **Mean processor utilization:** The amount of processor time used to execute a given set of tasks as compared to the operation time.
- **Mean memory utilization:** The amount of memory used to execute a given set of tasks as compared to the available memory.
- **Mean I/O devices utilization:** The amount of an I/O device busy time that used to perform a given set of tasks as compared to the I/O operation time.
- **Energy consumption:** The amount of energy used to perform a specific operation like data processing, storage or transfer.
- **Transaction processing capacity:** The number of transactions that can be processed per unit time.
- **System availability:** The proportion of the scheduled system operational time that the system is actually available.

- Fault avoidance:** The proportion of fault patterns that has been brought under control to avoid critical and serious failures.

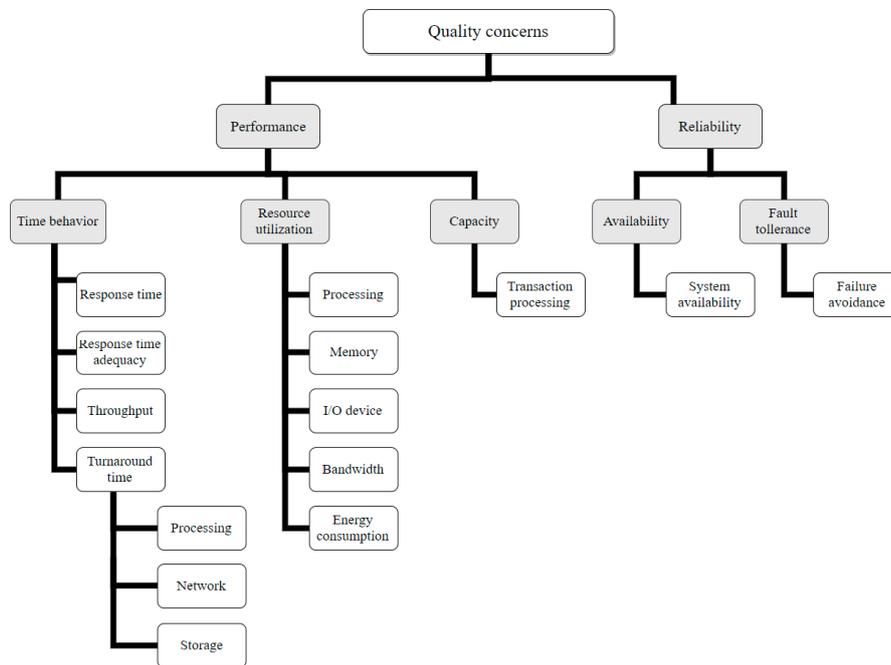


Figure 3. Qualities and related metrics identified in the literature.

4. Supported Qualities by Edge Computing Simulators

The use of simulation as a method for analyzing and optimizing IoT systems has been established, for example, for designing efficient networks or for investigating the scalability of a specific system [25]. To this end, a large number of simulators exists which can be used for investigating different aspects of IoT systems. To analyze the distribution and processing of tasks, there are many simulators that focus on aspects of cloud computing [26]. These simulators support, for instance, the design of architectures, virtualization management, or investigations of the system’s security.

Identifying available simulators for edge computing has been recently investigated by Svorobej et al. [10], who extracted and explained features of each simulator. Here, we use the same list of simulators (shown in Figure 4) for extracting the supported qualities and metrics. Having features and supporting qualities help IoT systems designers select the appropriate simulator. The supported qualities and metrics by simulators are shown in the Table 1.

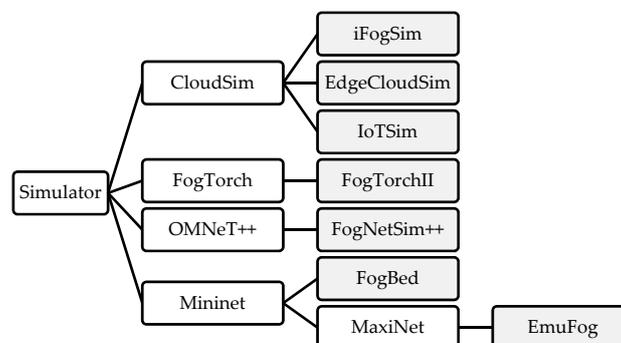


Figure 4. Relations of edge computing simulators.

Table 1. Supported qualities and metrics by edge computing simulators.

		iFogSim	FogNetSim++	EdgeCloudSim	IoTSim	FogTorch II	EmuFog	FogBed
Time behavior	Response time	Y	-	Y	-	Y	Y	Y
	Processing delay	Y	Y	Y	Y	-	-	-
	Network delay	-	Y	Y	-	Y	Y	-
	Storage read/write delay	-	-	-	-	-	-	-
	Throughput	-	-	-	-	-	-	-
Resource Utilization	Bandwidth utilization	Y	-	Y	Y	Y	-	-
	Processing utilization	Y	-	-	Y	-	-	-
	Memory utilization	-	-	-	-	Y	-	-
	I/O devices utilization	-	-	-	-	Y	-	-
	Energy consumption	Y	Y	-	-	-	-	-
Capacity	Transaction processing capacity	-	Y	Y	-	-	-	-
Availability	System availability	-	-	-	-	-	-	-
Fault tolerance	Failure avoidance	-	-	-	-	-	-	-

The iFogSim [18] simulator has been commonly used among the literature for simulating edge computing and mainly aims to provide an evaluation platform for resource management policies. This simulator is based on the CloudSim [21], which has been broadly used for simulating cloud computing. iFogSim provides users with quantitative measurements of energy consumption, latency, and network congestion to quantitatively evaluate the qualities. For the latency, a control loop which contains the processing components can be specified to calculate the processing latency from sending a request to receive a response. For energy consumption, the iFogSim simulator calculates the amount of consumed energy by each device in the system. It is also possible to measure the amount of network used between different devices. Moreover, this simulator measures resource usage of edge and cloud devices to be used in the internal resource allocation algorithm.

By focusing on simulation of the network characteristics of distributed edge computing devices, the FogNetSim++ [27] simulator aims to simulate a large network of devices. This simulator is based on the well-known OMNET++ [28] network simulator, which includes extensive network libraries and supports many network protocols. The network model they used is based on the publish-subscribe mechanism and a broker device is responsible to manage other devices. In this model, the measured delay is the time taken from the end device to the broker node. For the energy consumption, residual energy of end devices and also energy consumption for each task is calculated. Execution time for each task, number of requests that can be processed based on the capacity of Edge devices, and the queue waiting time for each request have also been considered by this simulator. Moreover, the packet error rate of the wireless channel is used for internal measurements which was not among the identified qualities.

The EdgeCloudSim simulator, similar to the iFogSim simulator, is also an extension of the CloudSim simulator with a special focus on demands from edge computing [19]. In contrast to iFogSim, its focus lies on the more dynamic and realistic investigation of service usage, as well as the consideration of mobility. For this purpose, the EdgeCloudSim simulator implements mobility models which enable the simulation of the movement of mobile devices, and thus the investigation of the rate

of failed tasks. To overcome mobility-related issues, virtual machine (VM) migration methods can be used, which are not yet supported by EdgeCloudSim. Another special feature of this simulator is the consideration of the effect of network load and transmission delays on the delay of service requests. Finally, a load generator is provided, which allows for the dynamic generation of tasks with individual data size and length.

The IoTsim simulator is also built on CloudSim and focuses on the investigation of big data processing [20]. It extends CloudSim's capabilities for modeling IoT applications and facilitates the model building process. Moreover, IoTsim makes use of MapReduce and other big data systems to allow for the processing of large amounts of sensor data. The IoTsim simulator places special emphasis on the simulation of network communication and delays between data storage and processing VMs. This makes IoTsim a well-suited simulator for the investigation of the performance of large IoT applications that generate and process huge amounts of data. Accordingly, the provided measurements are related to MapReduce and VM functions such as execution time of Mapreduce jobs, discrepancy between start of reduce and map tasks, network cost of Mapreduce, and CPU computation cost incurred by VMs.

Brogi et al. [7] proposed a simulation tool called FogTorchII with the aim of helping IT designers regarding where to deploy each component of an IoT application in a cloud-fog model. They considered two metrics, latency and bandwidth usage, where latency is the service latency and the communication link latency and bandwidth is measured by the link upload and download bandwidth. To calculate the final results of appropriate solutions, they also considered fog resource consumption through the average percentage of consumed RAM and storage.

To support repeatable and controllable experiments with real applications, EmuFog [29] proposed an emulation framework for emulating network topologies. This tool was developed on top of MaxiNet [30] network emulator. For quality measurement, they measured latency that is the latency between client and fog node. They also discussed the possibility of using network bandwidth, which they did not consider in the framework.

Finally, Fogbed [31] is an extension of the Mininet network emulator, which enables the dynamic adding, connecting, and removing of virtual nodes via Docker containers. By this means, the network topology can be dynamically changed, which allows for the investigation of real-world fog infrastructures, where cloud services are provided closer to the edge of the network. Characteristics of the containers that are considered and can be changed are, for instance, available memory or CPU time, however, others that are of relevance to fog and edge computing such as security, fault tolerance, and scalability are not considered by Fogbed [11]. Although this emulator is mentioned by Sovobej et al. [11], its primary purpose is not edge computing.

5. Discussion and Research Gaps

In Section 3, we presented the process of deriving qualities and related metrics for simulating the deployment of IoT systems in edge computing. In order to reduce the risk of human judgment, for the identification of the qualities, we conducted a systematic search to extract the metrics used by other researchers. Then, to rely on a solid base, the ISO/IEC standards were used to classify the identified metrics, however, we found that some of the important identified metrics, i.e., processing, network, and storage delay were not explicitly mentioned by the standards. Nevertheless, they are considered to be finer-grained metrics under mean turnaround time. Moreover, energy consumption was used frequently in the literature, but there is no related metric in the ISO/IEC 25023 standard, which is why we added energy consumption as a metric, however, we do not claim that the derived qualities and metrics are a complete list. Nevertheless, we believe it is an appropriate set and a suitable reference to evaluate the current edge computing simulators.

Regarding the qualities and metrics supported by the edge computing simulators (as represented in Table 1), our analysis revealed that none of the simulators cover more than five of the 13 identified metrics. With the rapidly increasing amount of research on edge computing, there is still a need for

simulators that can assess a wider set of metrics. Among the metrics, delay is the most popular one that is measured by all the existing simulators, but in different ways. Mean response time, processing delay, and network delay are the most common time-behavior metrics supported by the simulators. Among resource utilization metrics, network usage is the most common. Processing utilization and energy consumption were also supported by two simulators, however, we expected more measurements of resource utilization from the current simulators, since efficient resource usage is considered one of the main motivations for the emergence of edge computing. Moreover, some of the identified metrics such as throughput, system availability, and fault avoidance were not supported by any of the simulators.

It is also worth mentioning that, in this study, we analyzed edge computing simulators in terms of the supported qualities and not regarding features such as mobility and infrastructure modeling. For an analysis of these features, we refer to the survey conducted by Svorobej et al. [10]. Moreover, the scalability of IoT systems is a key factor that should be supported and evaluated by the edge computing simulators, however, scalability is not considered as a quality by ISO/IEC 25010, and as Duboc et al. [32] outlined, a system's ability to scale cannot be analyzed in isolation but has to be assessed with respect to other system qualities, for example, availability or resource usage. To this end, they define scalability analysis as investigating the effect that variations in environmental or design aspects have on qualities of the system. Thus, to facilitate a comprehensive simulation-based scalability analysis of IoT systems and edge computing, different qualities must be supported by the simulator.

We believe our work can help IoT architects understand the current state of the simulators with respect to supported qualities description and measurement. Moreover, in terms of qualities, we can conclude that there still are gaps. Simulators should cover a wider range of qualities' measurements and modeling new measurements is necessary. Accordingly, we argue that important future work should include the extension of existing simulators to support this, as well as an investigation of the tradeoffs among qualities to provide better support to IoT architects for decision making.

6. Conclusions

The consideration of edge computing architectures for the deployment of IoT systems has been a trend recently, however, the decision regarding where to deploy IoT application software components is a complex multifaceted problem, requiring the measurement and consideration of various quality characteristics. Due to the lack of available edge computing testbeds and cost concerns, using simulation can help architects make better decisions. In this study, we investigated what qualities and metrics are supported, in terms of modeling and analysis, by the main edge computing simulators that are currently available. To achieve this, we conducted a systematic search to identify the relevant qualities used by the literature. By studying the edge computing simulators, we assessed which of the identified qualities and the related metrics are supported by each simulator. Finally, based on the results, we identified some research gaps and discussed future work, for example, covering a wider range of qualities' measurements and modeling new measurements for edge computing simulation.

Author Contributions: Conceptualization, M.A., F.L., P.D., and R.S.; methodology, M.A., F.L., P.D., and R.S.; validation, M.A., F.L., P.D., and R.S.; investigation, M.A. and F.L.; Writing—Original draft preparation, M.A. and F.L.; Writing—Review and editing, M.A., F.L., P.D., and R.S.; supervision, P.D. and R.S.

Funding: This research was partially funded by Stiftelsen för Kunskaps- och Kompetensutveckling, grant number 20140035.

Acknowledgments: The authors would like to thank the anonymous reviewers, whose comments and suggestions helped improve this manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A : Extracted Metrics

The extracted metrics after applying the filtering are listed in Table A1. In the table, two types of metrics are presented, the ISO/IEC metrics, and the metrics extracted from the literature. We classified extracted metrics from the literature based on the metrics suggested by ISO/IEC 25023. For increased readability, we removed statistical terms like average, mean, and percentage.

Table A1. Metrics used in the literature, categorized according to the ISO/IEC standards (the numbers between brackets show the number of papers that used the metric).

ISO/IEC 25010 Qualities	ISO/IEC 25023 Measures	Related Metrics Used in the Literature
Time behavior	Mean response time	Service allocation delay, first response time of processing a dataset, service latency (used in 3 papers), system response time (3), service execution delay
	Mean turnaround time (processing, network storage delay)	CPU time, round-trip time (2), processing delay, latency of application, communications latency, latency ratio, delay of transferring datasets, completion time, time to write or read (2), time taken to create container for new images, time taken to create container for existing images, time taken to transfer the application file, time taken to transfer the log file, transmission delay, loading time, delay jitter, execution time (2), delivery latency, response times of computation tasks, activation times, running time, accessing cost of data chunks, handover latency, latency of data synchronization, workload completion time, waiting time, end to end delay, sync time, delay, queuing delay (2), execution time, container activation time
	Throughput	Number of messages processed, throughput of request serviced, successful executed job throughput, queue length in time, bandwidth throughput, disk throughput
Resource utilization	Memory utilization	Queue utilization, allocated slots
	Processor utilization	Processing power, processing cost for each application, CPU utilization (3), computing load, CPU consumption, max system load, fairness by Gini coefficient, computation cost, system load
	I/O devices utilization usage	Storage overhead
	Bandwidth utilization	Communication cost, number of retransmissions (2), bandwidth consumptions, required bandwidth, traffic load, amount of data sent between fog sites, volume of data transmitted, amount of network traffic sent, communication overhead
Capacity	Energy consumption*	Energy consumption (7), transmission energy, power consumption (3), allocated power, residual energy, energy efficiency
	Transaction processing capacity	Number of satisfied requests, system loss rate, cloud requests fulfilled, number of existing jobs in the CPU queues, rate of targets dropped over arrival, cumulative delivery ratio, provisioned capacity, service drop rate, number of finalized jobs, number of executed packets
Availability	System availability	Uptime
Fault tolerance	Failure avoidance	Scheduled requests with crash

*Energy consumption is not among ISO/IEC 25023 measures.

References

1. Zikria, Y.B.; Yu, H.; Afzal, M.K.; Rehmani, M.H.; Hahm, O. Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques. *Future Gener. Comput. Syst.* **2018**, *88*, 699–706. [[CrossRef](#)]
2. Miorandi, D.; Sicari, S.; De Pellegrini, F.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Netw.* **2012**, *10*, 1497–1516. [[CrossRef](#)]
3. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. *IEEE Int. Things* **2016**, *3*, 854–864. [[CrossRef](#)]
4. Shi, W.; Dustdar, S. The promise of edge computing. *Computer* **2016**, *49*, 78–81. [[CrossRef](#)]
5. Li, W.; Santos, I.; Delicato, F.C.; Pires, P.F.; Pirmez, L.; Wei, W. System modelling and performance evaluation of a three-tier Cloud of Things. *Futur. Gener. Comput. Syst.* **2017**, *70*, 104–125. [[CrossRef](#)]
6. Bittencourt, L.; Immich, R.; Sakellariou, R.; Fonseca, N.; Madeira, E.; Curado, M.; Villas, L.; DaSilva, L.; Lee, C.; Rana, O. The Internet of Things, Fog and Cloud continuum: Integration and challenges. *Int. Things* **2018**, *3*, 134–155. [[CrossRef](#)]
7. Brogi, A.; Forti, S.; Ibrahim, A. How to Best Deploy Your Fog Applications, Probably. In Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017.
8. ISO/IEC. *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*; BS ISO/IEC 25010: 2011; BSI Group: Geneva, Switzerland, 31 March 2011.
9. Law, A.M.; Kelton, W.D. *Simulation modeling and analysis*; McGraw-Hill Education: New York City, NY, USA, 2013; Volume 3.
10. Sotiriadis, S.; Bessis, N.; Asimakopoulou, E.; Mustafee, N. Towards Simulating the Internet of Things. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014.
11. Svorobej, S.; Takako Endo, P.; Bendeche, M.; Filelis-Papadopoulos, C.; Giannoutakis, K.; Gravvanis, G.; Tzovaras, D.; Byrne, J.; Lynn, T. Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges. *Futur. Internet* **2019**, *11*, 55. [[CrossRef](#)]
12. Ning, Z.; Dong, P.; Wang, X.; Rodrigues, J.J.P.C.; Xia, F. Deep Reinforcement Learning for Vehicular Edge Computing: An Intelligent Offloading System. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 60. [[CrossRef](#)]
13. Balasubramanian, V.; Wang, M.; Reisslein, M.; Xu, C. Edge-Boost: Enhancing Multimedia Delivery with Mobile Edge Caching in 5G-D2D Networks. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019.
14. Masip-Bruin, X.; Marin-Tordera, E.; Tashakor, G.; Jukan, A.; Ren, G.J. Foggy clouds and cloudy fogs A real need for coordinated management of fog-to-cloud computing systems. *IEEE Wirel. Commun.* **2016**, *23*, 120–128. [[CrossRef](#)]
15. Sarkar, S.; Chatterjee, S.; Misra, S. Assessment of the Suitability of Fog Computing in the Context of Internet of Things. *IEEE Trans. Cloud Comput.* **2015**, *6*, 1–10. [[CrossRef](#)]
16. Byrne, J.; Svorobej, S.; Giannoutakis, K.M.; Tzovaras, D.; Byrne, P.J.; Östberg, P.O.; Gourinovitch, A.; Lynn, T. A Review of Cloud Computing Simulation Platforms and Related Environments. In Proceedings of the 7th International Conference on Cloud Computing and Services Science, Porto, Portugal, 24–26 April 2017.
17. Brady, S.; Hava, A.; Perry, P.; Murphy, J.; Magoni, D.; Portillo-Dominguez, A.O. Towards an emulated IoT test environment for anomaly detection using NEMU. *Glob. Int. Things Summit (GIoTS)* **2017**, 1–6.
18. Gupta, H.; Buyya, R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Softw.: Pract. Exp.* **2017**, *47*, 1275–1296. [[CrossRef](#)]
19. Sonmez, C.; Ozgovde, A. EdgeCloudSim: An environment for performance evaluation of edge computing systems. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, 1–17. [[CrossRef](#)]
20. Zeng, X.; Garg, S.K.; Strazdins, P.; Jayaraman, P.P.; Georgakopoulos, D.; Ranjan, R. IOTSim: A simulator for analysing IoT applications. *J. Syst. Archit.* **2017**, *72*, 93–107. [[CrossRef](#)]

21. Calheiros, R.N.; Ranjan, R.; Beloglazov, A.; De Rose, C.A.F.; Buyya, R. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exp.* **2011**, *41*, 23–50. [[CrossRef](#)]
22. Blas, M.J.; Gonnet, S.; Leone, H. An ontology to document a quality scheme specification of a software product. *Expert Syst.* **2017**, *34*, 1–21. [[CrossRef](#)]
23. Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* **2015**, *64*, 1–18. [[CrossRef](#)]
24. ISO/IEC. *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Measurement of System and Software Product Quality*; ISO/IEC 25023: 2016; BSI Group: Geneva, Switzerland, 15 June 2016.
25. D'Angelo, G.; Ferretti, S.; Ghini, V. Simulation of the Internet of Things. In Proceedings of the 2016 International Conference on High Performance Computing & Simulation (HPCS), Innsbruck, Austria, 18–22 July 2016.
26. Lynn, T.; Gourinovitch, A.; Byrne, J.; Byrne, P.; Svorobej, S.; Giannoutakis, K.; Kenny, D.; Morrison, J. A Preliminary Systematic Review of Computer Science Literature on Cloud Computing Research using Open Source Simulation Platforms. In Proceedings of the 7th International Conference on Cloud Computing and Services Science, Porto, Portugal, 24–26 April 2017.
27. Qayyum, T.; Malik, A.W.; Khattak, M.A.K.; Member, S. FogNetSim ++: A Toolkit for Modeling and Simulation of Distributed Fog Environment. *IEEE Access* **2018**, *6*, 63570–63583. [[CrossRef](#)]
28. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008.
29. Mayer, R.; Graser, L.; Gupta, H.; Saurez, E.; Ramachandran, U. EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures. In Proceedings of the 2017 IEEE Fog World Congress (FWC), Santa Clara, CA, USA, 30 October–1 November 2017.
30. Wette, P.; Draxler, M.; Schwabe, A.; Wallaschek, F.; Hassan Zahraee, M.; Karl, H. Maxinet: Distributed emulation of software-defined networks. In Proceedings of the 2014 IFIP Networking Conference, Trondheim, Norway, 2–4 June 2014.
31. Coutinho, A.; Greve, F.; Prazeres, C.; Cardoso, J. Fogbed: A Rapid-Prototyping Emulation Environment for Fog Computing. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018.
32. Duboc, L.; Rosenblum, D.; Wicks, T. A Framework for Characterization and Analysis of Software System Scalability. In Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Dubrovnik, Croatia, 3–7 September 2007.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).