Systematic Mapping on Orchestration of Container-based Applications in Fog Computing

Walter do Espírito Santo[2], Rúbens de Souza Matos Júnior[1], Admilson de Ribamar Lima Ribeiro[2], Danilo Souza Silva[2], Reneilson Santos[2]

[1]*Computer Department - Federal Institute of Sergipe, IFS - Lagarto, Brazil*

[2]*Computer Department - Federal University of Sergipe, UFS - São Cristóvão, Brazil*

walter.santo@ifs.edu.br, rubens.junior@ifs.edu.br, admilson@ufs.br, danilo.silva@dcomp.ufs.br, reneilson1@gmail.com

***Abstract*—There is an increasing number of Internet of Things (IoT) devices in the border of computer networks, requiring local processing and lightweight virtualization to deal with issues such as heterogeneity, Quality of Service (QoS) management, scalability, mobility, federation, and interoperability. Fog computing can provide the computational resources required by IoT devices to process their data. Low energy consumption and total cost of ownership are among the desirable properties for auxiliar infrastructures such as those deployed for fog computing, which do not require large computational power though. There is a noteworthy trend of undergoing research efforts towards the definition of software and hardware architectures for fog computing in this context. In this sense, this paper presents a Systematic Literature Mapping with the purpose of understanding and identifying metrics and gaps in current literature about orchestration of container-based applications, especially those hosted in clusters of Single Board Computer (SBC) platforms, such as Raspberry Pi, which have been used for deploying fog computing environments.**

***Index Terms*—systematic mapping, fog computing, cluster, orchestration, containers, SBC**

I. Introduction

Cloud computing has been migrating from large and centralized data centers for distributed cloud environments, comprising a network of virtualized infrastructures. In this direction, virtualized architectures allow the integration of IoT infrastructure to the cloud, by means of the so-called edge or fog computing environments [1]. Although cloud infrastructures can deal with a large amount of data from IoT clusters, limited bandwidth, high latency, or communication costs are important challenges. Consequently, comes up the necessity of processing data near to the source. Fog computing is a trending solution for such an issue [2], [3].

Fog computing pushes applications, services, data, computing power, and decision-making from centralized nodes to the borders of the network, reducing significantly the amount of data transferred to the cloud. The dense geographic distribution of fog environments may help to obtain better localization precision for many applications in comparison with cloud-based counterparts [2]. Nodes in a fog can maintain a collaborative execution environment, creating clusters among them. Clusters can be formed based on fog nodes homogeneity [4], according to their localization [5], load balancing [6], or still development of functional subsystem [7].

A challenge in this new scenario is the development or adoption of lightweight virtualization technologies, such as containers, which also have become recognized as more efficient for applications packaging and execution in an isolated environment [8], [9]. Orchestration of virtualized resources is another important requirement to automate service deployment and compliance to expected QoS metrics. Platforms for container orchestration simplify the resources management and provide a structure not just to define the initial implantation of a container, but also to manage many containers as an entity for purposes of availability, sizing, and networking [10].

This paper presents a Systematic Mapping study to understand and identify metrics and gaps in the current literature regarding containers orchestration in fog computing environments. This study investigates how researches are being conducted in this field and approaches the following research questions of general purpose: "Which containerisation technologies and orchestration techniques can be used as micro-Platform as a Service (micro-PaaS) in a fog computing environment using clusters of SBC platform, and that meet QoS requirements in the IoT context?". Other complementary research questions are derived from the main question.

The remaining of the paper is organized as follows: Section II describes the methodology of systematic mapping applied in this study. Section III shows related works. Section IV discusses about the main results of this paper and analyze them. Finally, Section V shows conclusions and final remarks, as well as some directions for future works.

II. Research Methodology

Systematic mapping method aims at answering a set of raised research questions based on pieces of evidence found in the literature, by means of a broad review of the relevant primary studies for a specific research problem [11].

More specifically, this study followed the methodology of the systematic literature mapping defined in [12], which consists of defining research questions, searching and selecting the relevant studies, extracting, data and mapping the results. The mapping presented in this paper aims to identify researches about orchestration of applications for IoT, using clusters composed by SBC platforms, respecting QoS criteria.

978-3-903176-24-9 © 2019 IFIP

A. Research Question

We define the following main question which guides this study: "Which containerisation technologies and techniques of container orchestration can be used as micro-PaaS in a fog computing environment using clusters of SBC platforms, and that meet QoS requirements in the IoT context?". In order to answer this question, we extracted another three specific questions, described as follows: **RQ1.** Which are the main containerisation technologies that can be applied to the fog computing and which are the main related metrics, in the IoT context? **RQ2.** Which container orchestration techniques have been focused in fog computing research works and that can be applied in a cluster of SBC platforms? **RQ3.** Which are the main evaluation criteria and metrics for an orchestration environment in fog computing?

B. Search and Selection Strategy

In order to execute the mapping we used three steps: planning, conduction, and study report [11]. We employed the *Parsival* (*Perform Systematic Literature Reviews*) tool (available at https://parsif.al/) to support the mapping process.

Planning step aims to define the objectives to be achieved for the research and the mapping protocol, specifying the research method and allowing reproducibility, besides reducing bias risks on results [13]. The developed protocol comprises objective, research questions, *PICOC* strategy (Population, Intervention, Control, Outcome, Context), search string, inclusion and exclusion criteria of works found in literature, and selection criteria of published research bases. Keywords definition, for search in the papers databases, was made according to PICOC strategy [14], as shown in Table I.

Table I
KEYWORDS BASED ON PICOC STRATEGY

PICOC	KEYWORDS
Population	Fog computing, cloudlets, edge computing
Intervention	Orchestration, cluster, clustering, security, embedded system, single board computer
Control	Not defined
Outcome	Methodology, methodologies, technique, method, prototype, protocol, architecture
Context	Internet of Things

Through the identified keywords was possible to derive a generic search string, which was used as structure to generate specific strings, contemplating the syntactic particularities of each research base (e.g., ACM, IEEEXplore, ScienceDirect, Scopus, and Web of Science) chosen for this mapping: ((*fog computing OR cloudlets*) *AND* (*edge AND computing*)) *AND* (*orchestration OR cluster OR clustering OR security*) *OR* (*embedded AND system*) *OR* (*single AND board AND computer*)) *AND* (*methodology OR methodologies OR technique OR method OR prototype OR protocol OR architecture*) *AND* (*internet of things OR IoT*).

For the selection of results, we defined inclusion and exclusion criteria of papers to be analyzed, as a means to the evaluate their relevance for our study.

Inclusion Criteria. The study must be a complete paper published in journals or conferences (shows mature approaches and, probably, more established studies). It must explore fog computing and bring some orchestration technique or method; and, should also discuss questions related to clusters and containers technologies.

Exclusion Criteria. Studies that do not approach fog computing and cloud computing domain were excluded. Secondary researches and studies were removed since they report third-party approaches. We also excluded short papers, extended abstracts, publications that were not found in full, book chapters, and papers without peer-review (white papers).

Afterwards, it was performed the selection of relevant research bases, using as an example some bases shown in [15]: ACM Digital Library, IEEEXplore, ScienceDirect, Scopus, and Web of Science. Those are relevant research bases in the Computer Science area. Another selection criterion was the ability to use advanced options for applying the search string. Moreover, selected bases must be accessible in the Federal University of Sergipe (Universidade Federal de Sergipe - UFS), through the "Periodicos CAPES" [16] portal. Finally, all selected bases must be compatible with the support research tool. Considering the discussed criteria, the search would be held only in Scopus due to its ability to index all other digital libraries, however, we observed that Scopus was not fully updated at the moment we needed it. Therefore, all mentioned bases were used: ACM Digital Library, IEEEXplore, ScienceDirect, Scopus, and Web of Science.

C. Conduction

First activity executed in the mapping conduction was retrieving the studies. For that, the search string was executed in each base, as shown in Table II. Obtained results were exported for a file of BibTeX format and imported for the support tool *Parsifal*. Searches were executed in October 2018, achieving an amount of 511 studies.

Parsifal tool was chosen to manage this step due to many of its features. It is a web application which supported a collaborative environment among the authors of this study. Usability is another strong factor, since it does not require installation of dependencies, such as the Java Virtual Machine (JVM), used in other tools.

In order to summarize the initial results, Figure 1 shows the distribution of found papers per each selected base and per publication year. We found 75 duplicated studies resulting in 436 potentially relevant studies. Then we read paper titles and abstracts and applied inclusion and exclusion criteria. 367 studies were excluded, remaining 69 relevant studies for the application of quality criteria according to [12].

Quality criteria were applied by formulation and application of qualification questions: (**QQ1.**) Is the validation process structured and defined? (**QQ2.**) Are the variables considered by the study measured properly? (**QQ3.**) Are the techniques or methods described clearly and is their selection justified? (**QQ4.**) Was the study designed to achieve these objectives? (**QQ5.**) Are the research objectives clearly specified?

Table II
SEARCH STRING USED IN EACH DIGITAL LIBRARY

Digital Library	Search String
ACM	*(+("fog computing" cloudlets "edge computing")+(orchestration cluster clustering security "embedded system" "single board computer")+(methodology OR methodologies OR technique OR method OR prototype OR protocol OR architecture)+(internet of things OR IoT))*
IEEEXplore	*((fog computing OR cloudlets) AND (edge AND computing)) AND ((orchestration cluster OR clustering OR security) OR (embedded AND system) OR (single AND board AND computer)) AND ((methodology OR methodologies OR technique OR method OR prototype OR protocol OR architecture)) AND ((internet of things OR IoT))*
Science Direct	*((fog computing OR cloudlets) AND {edge computing}) AND ((orchestration OR cluster OR clustering OR security) AND {embedded system} AND {single board computer}) AND (methodology OR methodologies OR technique OR method OR prototype OR protocol OR architecture) AND (internet of things OR IoT))*
Scopus	*TITLE-ABS-KEY (((fog AND computing OR cloudlets) AND (edge AND computing)) AND ((orchestration OR cluster OR clustering OR security) OR (embedded AND system) OR (single AND board AND computer)) AND ((methodology OR methodologies OR technique OR method OR prototype OR protocol OR architecture))AND ((internet of things OR IoT))) AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (SRCTYPE , "j") OR LIMIT-TO (SRCTYPE , "p")) AND (LIMIT-TO (SUBJAREA , "COMP"))*
Web of Science	*(((fog computing OR cloudlets) AND (edge AND computing)) AND ((orchestration OR cluster OR clustering OR security) OR (embedded AND system) OR (single AND board AND computer)) AND ((methodology OR methodologies OR technique OR method OR prototype OR protocol OR architecture)) AND ((internet of things OR IoT)))*

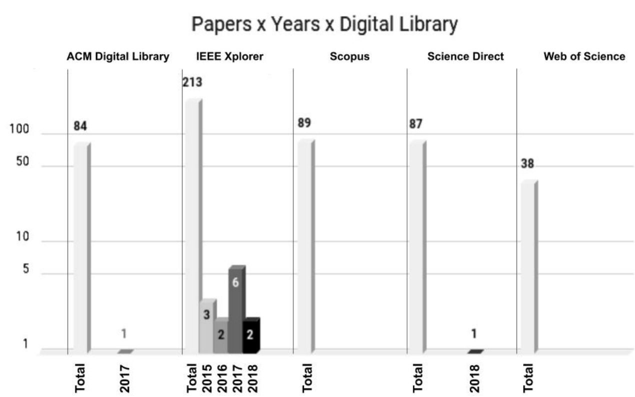

Figure 1. Amount of papers vs. Publication Years vs. Digital Library

For each question, there is a score assigned to the related study, which depends on the relevance of the question: 2 (relevant), 1 (partially relevant), or 0 (not relevant). After applying quality questions, studies with score below 50% were discarded. 54 studies did not reach the minimum score and were removed, so 15 studies remained.

Selection Bias. At the process beginning, the inclusion and exclusion criteria were applied with first author's judgment. This means some studies could have been categorized incorrectly. In order to mitigate that, the protocol was discussed among the authors to ensure a common understanding. Afterwards, two authors read each paper and another one was called to discuss when needed.

Data Extraction. Extraction bias or problem could affect the classifications of facets and the analysis of selected studies. In order to reduce this bias, the definitions of each data, initial taxonomy, and extraction protocol were discussed deeply. Then, data extraction was made and the doubts were discussed and solved, in order to make the mean clearance of data items for other researches. If there was no consensus about the classification of extracted information, a third opinion was required to resolve conflicts and, consequently, come to a common view to ensure extracted data validity.

III. RELATED WORK

This systematic mapping pointed out fifteen (15) relevant studies on fog computing adressing also container technologies, container orchestration, SBC clusters, and analysis of main metrics in an IoT context.

Authors in [8] proposed a reference architecture which supports functionalities of integrated computing based on containers and on the orchestration framework *Kubernetes*. The majority of high-level requirements of integrated computing are fulfilled in the implementation of a proposed prototype, with exception of the support to dynamic workload mobility and real-time workload. Still, the implemented platform can be used to increase efficiency, development flexibility, deployment of an industrial application set.

A Dynamic Mobile Cloudlet Cluster Policy (DMCCP) is presented in [17] for fog server offloading. This policy arranges dynamically an optimal cloudlet to meet the different resource demands of local mobile device tasks. All available mobile devices (potential cloudlet resources) must be preconnected to a fog server via wireless network. In order to use efficiently the IT resource pool, a Mobile Resources Monitor (MRM) on fog server is employed to observe the available capacity of each mobile device. The MRM receives data about Central Processing Unit (CPU) utilization, memory usage, and battery level. If a single mobile device requests a task to the fog server, this system will group a cloudlet by DMCCP based on resource demands of this specific task.

In [18], the authors proposed a simple algorithm for task scheduling in an inter-fog environment, or fog federation environment, in which Fog Services Providers (FSPs) can capitalize in the cooperation among fog instances or cloudlets, while achieving load balance among fog nodes.

The research described in [19] designed implemented, and tested an innovative middleware solution for fog computing based in i) scalability extensions of IoT gateway provided by the open-source *Kura* structure, aiming at decentralizing functionalities from the MQTT broker of the *Kura* cloud structure to the wrapped edges; ii) *Docker*-based container on RaspberryPi devices, aims to explore the containerisation to ease interoperability and portability by standardization of node settings in terms of micro services definition.

In [20], a Parallel Genetic Algorithm (GA-Par) was employed in the orchestration of IoT services in a fog environment to determine and select the best devices for dynamic composition of holistic workflows for more complex functions. The main responsibility of the orchestrator is to select resources and implant the general service workflow according to security data requirements, reliability, and system efficiency.

An architecture for fog environment management was defined in [21], taking into account a hybrid approach including orchestration and choreography. Orchestration is used in the north chaining region between fog and cloud components, to have a global vision of the system. On the other hand, the choreography is used in the south chaining region of the fog, allowing automatic quick answers in lower levels.

In [22], the authors showed an architecture for infrastructure and service orchestration management based on major fog computing requirements. In such an architecture, fog nodes can execute applications as virtualized and containerized services, offering access to connected devices over distinct communication technologies to deal with their tasks.

An evaluation of how containers can affect the general performance of applications in fog nodes was proposed in [9]. Besides the evaluation, the authors analyzed different container orchestration tools and how they meet the fog requirements to execute an application. Based on that analysis, they proposed a container orchestration framework for fog computing infrastructures.

Foggy, an architectural structure and software platform for workload orchestration and resources negotiation in multi-layer cloud computing system was proposed in [23]. Moreover, it is tailored for highly distributed, heterogeneous, and decentralized infrastructures. Foggy is able to orchestrate the workload in fog computing environment, acting as a mediator between the infrastructure owner and the renters, improving: i) efficiency and efficacy of the infrastructure provided, and ii) application performance to satisfy the requirements.

The research done in [24] proposed orchestration to implement distributed services of edge computing in a federated cloud environment, by means of a service manifest *Heat Orchestration Template* (HOT), which is analyzed by the *Orchestration Broker*. The broker extracts automatically every element which describes the relation between micro-services, which must be implemented in federated clouds using modules. An important feature is the ability to select target cloud as a function of geographic position of federated clouds, allowing to implement microservices at network edges.

In [25], the authors propose a model to investigate the task completion time as a min-max problem in the Fog Computing Supported Software-Defined Embedded System (FC-SDES), by joint consideration of task image positioning and task scheduling. In order to address high computational complexity, the authors present a heuristic algorithm of task completion time in three steps. Two first steps deal individually with the I/O time and the computation time, transforming the min-max problem of Mixed-Integer Nonlinear Programming (MINLP), in equivalents of max-min MINLP. Third step gathers the I/O processing and task computation together with transmission time. Linear programming relaxation is applied to approximate the solution of MINLP problems in the algorithm.

Motivated by a study case in modern ski resorts, which operate with high latency infrastructures and collect data from many sensoring devices, the authors in [26] proposed a Platform as a Service (PaaS) middleware for edge computing with container management and Raspberry Pi cluster. The architecture is organized in three layers: the lower level is composed by IoT devices; the intermediate level is responsible for the field area network and IP basic infrastructure; the higher level, therefore, is composed by the virtualized infrastructure for computation and storage in the cloud.

The authors in [27] proposed an optimization method to group small multiuser cells for distributed fog computing. The approach allows adaptive dimensionality and computer clusters resources management. It also establishes simultaneously computer clusters for all active requests for a better exploration of available resources, aiming at better Quality of User Experience (QoE). The proposed method is a joint optimization of workload distribution, computational capacity allocation, and transmission power control.

An experimental evaluation of QoS-aware distributed scheduler for Data Stream Processing (DSP) systems was made in [4], based on *Storm*, which is able to operate in cloud computing distributed environments. The authors used two different sets of applications: the first is based on a reference topology with different requirements; the second is some application well known. Results show that the proposed scheduler exceeds the Storm standard, improving the application performance and the system is improved with adaptation resources.

Finally, [6] defines a customizable algorithm to establish and manage resources of small, low complexity, cells for fog clustering to improve QoE, adressing load balance issues.

IV. Result and Discussion

Considering the studies found and filtered throughout systematic mapping, we found two problems which remain open: The first is to manage dependencies among containers in distributed applications of many layers. Something like an orchestration plane can describe components in a container, their dependencies, and their life cycle. A PaaS cloud can, therefore, approve the orchestration workflow of a plane by a container mechanism. According to [26], the PaaS services support packaging and deployment of containers. The second is: defining, implanting, and operating services in clouds compatible with many lightweight platforms, suitable to an

SBC device, such as Raspberry Pi, results in the need to move implementations from cloud services to fog premises [28].

Moreover, in [29] is defined a concise evaluation criteria set, taken as essentials for fog computing environments. The criteria are: heterogeneity, QoS management, scalability, mobility, federation, and interoperability; which are shown in Table III.

Table III
EVALUATION CRITERIA SUMMARY

Criterion	Definition
Heterogeneity C1	Nodes of the fog and cloud layer are very heterogeneous in computational and storage capacity. The fog must be able to deal with heterogeneity.
QoS Management C2	Fog should benefit real-time applications due to proximity with IoT devices and final users. However, latency varies greatly, depending on application components location, thus demanding QoS management.
Scalability C3	Fog should cover a great number of end-user devices, applications, domains, and nodes. Therefore, it must be operational in great scale and increase and decrease in an elastic way.
Mobility C4	IoT devices, final users, and fog nodes can be mobile. Fog systems must be able to deal with this mobility.
Federation C5	Fog distributes geographically on large scale deployments, in which each domain of the fog can belong to a different provider, as well as different cloud components. Provisioning applications demand the federation of these different providers, which can host different components.
Interoperability C6	As part of a federated system, an application can be executed with its components spread by different providers. Fog computing must be interoperable in the level of providers and architectural modules.

Table IV shows evidence of these criteria on selected studies, and a general vision of gaps which those relevant studies do not approach. Table IV is useful to answer the research questions of this systematic mapping, as presented in Section II-A. Aiming at a more judicious analysis, we also elaborated a bubble graph, depicted in Figure 2, with three dimensions to analyze, together, the facets: criterion, containerisation & orchestration, and metrics. The "criterion" facet brings evaluation criteria considered in the selected studies [29]; the "containerisation & orchestration" facet approaches most relevant container technologies and orchestration techniques in the literature; the "metrics" facet reflects the most relevant metrics found in the studies.

Bubbles in the top left of Figure 2 show that "*Docker Container*" and "scalability" are the most common focus in the analyzed studies. Orchestration technique "*Docker Swarm*" and "scalability" are also a common focus. Moreover, it is possible to conclude that "mobility" and "*Docker Container*" do not have any bubble associated to it, as well as "mobility", "federation", and "heterogeneity" related to "*Docker Swarm*", reporting gaps which may be fulfilled by future works.

Additionally, in the right part of the graph, the "transfer rate" is related just with the criteria "QoS management" and "mobility", and the metric "energy consumption" is related just with the criteria "scalability" and "mobility". This indicates that there are gaps of great relevance which were not treated by previous researchesMoreover, the "latency" has correspondence with all the criterion facets, which shows a great interest of the scientific community on this metric.

RQ1. Which are the main containerisation technologies existing that can be applied to the fog computing and which are the main related metrics, in the IoT context?

Analyzed works describe different containerisation technologies in fog computing environment [8], [9], [19], [22], [23], [26], orchestration techniques, as the discussed in the studies in [4], [8], [9], [17], [19], [21]–[24], [26]. From these, 40% pointed *Docker Container* as containerisation technology [8], [19], [23], [26]. *Docker* appears as highly suitable for packaging and application management in fog computing, more flexible and lighter than VMs. Although some researches consider QoS levels, questions related to transfer rate were observed just in [17]. Network congestion occurs, mainly, because the increase in overdraft, due to the interactions of a large number of IoT devices and sensorswith cloud datacenters, degrading the system performance. In the same way, just 13% of the studies consider energy consumption metric. Due to the decentralized nature and heterogeneity of the fog, to obtain desirable performance is not an easy task in environments without proper management of computational resources. Differently, the latency metric was a research object in 50% of related works, denoting this metric is very relevant and has raised considerable interest on scientific community.

RQ2. Which container orchestration techniques have been focused in fog computing research works and that can be applied in a cluster of SBC platforms?

Works presented in [9], [19], [22], show that the container orchestrations using the *Docker Swarm* technology, besides being compatible with SBC devices, are lighter when compared to other orchestration technologies.

RQ3. Which are the main evaluation criteria and metrics for an orchestration environment in fog computing?

Many important aspects of fog computing have been identified throughout the selected papers, and there are some other directions which may be approached and improved yet. Most studies did not mention explicitly which evaluation criteria are needed for application orchestration in the fog computing environment. Aiming at a more judicious analysis, considering the taxonomy based on [29] and data presented on Table IV and Figure 2, we notice that QoS management and scalability are the most cited criteria in literature. Interoperability is the third most cited, whereas mobility, federation, and heterogeneity are not considered so often as the other mentioned criteria.

V. CONCLUSION AND FUTURE WORK

This work presented a systematic mapping study of computer science literature to analyze how current container technologies, and SBC clusters are being applied, and how suitable they are, for fog computing supporting IoT environments. It was observed, e.g., the lack of evaluation criteria definition for applications orchestration in fog computing. Thus, this study proposed a taxonomy to classify the related studies according to the evaluation criteria (scalability, QoS management, interoperability, mobility, federation, and heterogeneity), based on

Table IV
Main characteristics and requirements for an efficient orchestration in a fog computing environment

Paper	Validation[1]	Evaluated Metrics	Criteria[2]					
			C1	C2	C3	C4	C5	C6
[8]	P	latency; workload; migration time	✓	x	x	x	x	x
[30]	S	computational resources; transfer rate	x	✓	x	✓	x	x
[18]	S	computational resources; latency	x	✓	x	x	✓	x
[19]	P	latency; execution time	x	x	✓	x	x	✓
[20]	S	security; performance	x	✓	✓	x	x	x
[21]	x	latency; resilience; scalability	x	✓	✓	x	x	✓
[22]	P S	computational resources; loss opportunities; escalation, mitigation and orchestration time	x	✓	x	x	x	✓
[9]	x	computational resources; escalation time	x	✓	✓	x	x	x
[23]	P S	workload	✓	x	x	x	✓	✓
[24]	P	analysis and implementation time	x	x	✓	x	✓	✓
[25]	S	latency; computational load	x	✓	x	x	x	x
[26]	P	n/a	x	✓	✓	x	x	x
[27]	S	energy consumption; workload and computational capacity allocation; latency; transmission power control	x	x	✓	✓	x	x
[4]	n/a	network traffic; latency	x	✓	✓	x	x	x
[6]	S	latency; energy consumption	x	x	x	✓	x	x
Proposed Research	E P	latency; energy consumption; transfer rate	✓	✓	✓	✓	✓	✓

1 P = prototype; S = simulation; E = emulation; 2 ✓= attended criterion; x = unattended criterion;

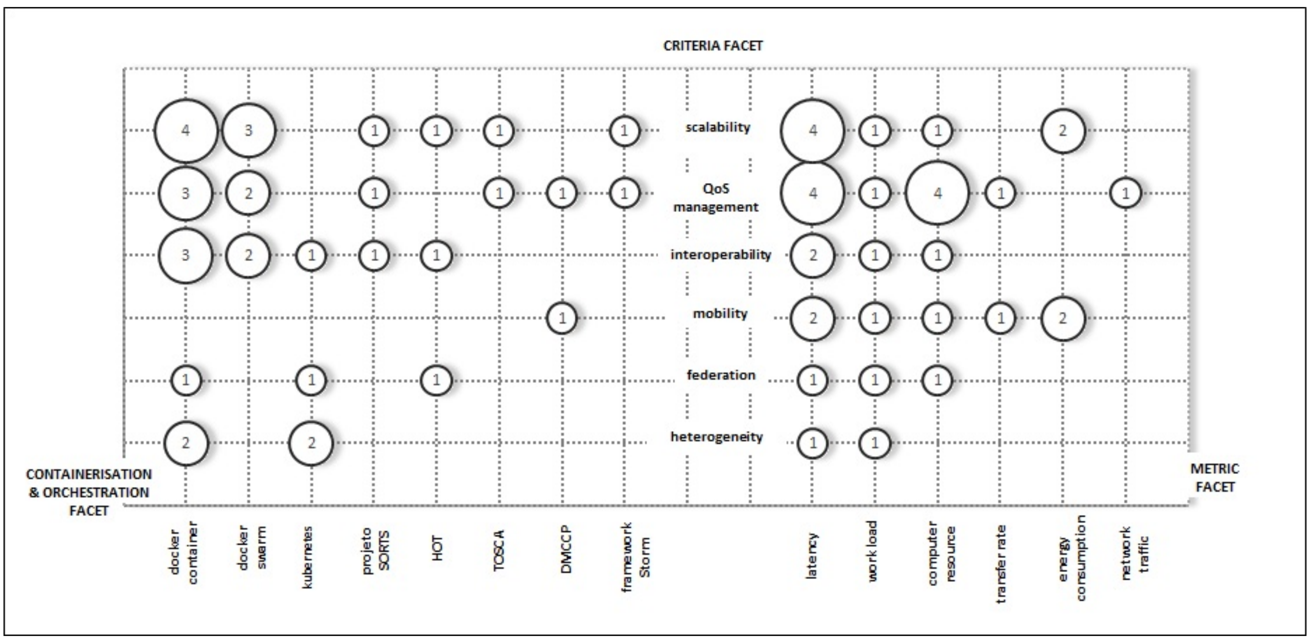

Figure 2. Criteria X Containerisation & Orchestration X Metrics

the classification discussed in [29] and that is essential for efficient application orchestration in fog computing setups.

The containerisation technology most indicated for fog computing environments found in the selected studies was the *Docker Container*. Although widespread, Docker is still in its initial development and has some limitations that need to be surpassed, such as the migration of services among nodes of different platforms, for example one ARM host to other with x86 architecture. *Docker Swarm*, besides being compatible with SBC platforms and natively integrated into Docker engine, is lighter than other orchestration technologies. Orchestration resources become essential in fog computing environments when there is a concern of scaling services horizontally among many nodes. This mapping concludes, therefore, that significant improvements are still needed to deal with the aspects related to container-based orchestration in fog computing environments ,thus, further efforts for its implementation can be relevant for the progress of this area.

As future works, the gaps identified here may be used to propose a wider taxonomy evaluation and apply it in other fog computing domains. Moreover, it is also possible to develop features for extending container technologies for Platform-as-a-Service (PaaS) frameworks tailored for fog environments.

References

[1] C. Pahl and B. Lee, "Containers and clusters for edge cloud architectures–a technology review," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*. IEEE, 2015, pp. 379–386.

[2] A. Munir, P. Kansakar, and S. U. Khan, "Ifciot: Integrated fog cloud iot: A novel architectural paradigm for the future internet of things." *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 74–82, 2017.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[4] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, "On qos-aware scheduling of data stream applications over fog computing infrastructures," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*. IEEE, 2015, pp. 271–276.

[5] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[6] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*. IEEE, 2015, pp. 1–6.

[7] M. A. Al Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE internet of things journal*, vol. 3, no. 2, pp. 161–169, 2016.

[8] S. V. Gogouvitis, H. Mueller, S. Premnadh, A. Seitz, and B. Bruegge, "Seamless computing in industrial systems using container orchestration," *Future Generation Computer Systems*, 2018.

[9] S. Hoque, M. S. de Brito, A. Willner, O. Keil, and T. Magedanz, "Towards container orchestration in fog computing infrastructures," in *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, vol. 2. IEEE, 2017, pp. 294–299.

[10] A. Khan, "Key characteristics of a container orchestration platform to enable a modern application," *IEEE Cloud Computing*, no. 5, pp. 42–48, 2017.

[11] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[12] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering." in *EASE*, vol. 8, 2008, pp. 68–77.

[13] I. Steinmacher, A. P. Chaves, and M. A. Gerosa, "Awareness support in distributed software development: A systematic review and mapping of the literature," *Computer Supported Cooperative Work (CSCW)*, vol. 22, no. 2-3, pp. 113–158, 2013.

[14] A. Beelmann, *Review of Systematic reviews in the social sciences. A practical guide.* Hogrefe & Huber Publishers, 2006, vol. 11, no. 3.

[15] K. Manikas, "Revisiting software ecosystems research: A longitudinal literature study," *Journal of Systems and Software*, vol. 117, pp. 84–103, 2016.

[16] "Periódico capes," acessed: out. 2018. [Online]. Available: https://www-periodicos-capes-gov-br.ez141.periodicos.capes.gov.br/

[17] Y. Li, N. T. Anh, A. S. Nooh, K. Ra, and M. Jo, "Dynamic mobile cloudlet clustering for fog computing," in *2018 International Conference*

[22] M. S. de Brito, S. Hoque, T. Magedanz, R. Steinke, A. Willner, D. Nehls, O. Keils, and F. Schreiner, "A service orchestration architecture for fog-

on Electronics, Information, and Communication (ICEIC), Jan 2018, pp. 1–4.

[18] D. P. Abreu, K. Velasquez, M. R. M. Assis, L. F. Bittencourt, M. Curado, E. Monteiro, and E. Madeira, "A rank scheduling mechanism for fog environments," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018, pp. 363–369.

[19] P. Bellavista and A. Zanni, "Feasibility of fog computing deployment based on docker containerization over raspberrypi," in *Proceedings of the 18th international conference on distributed computing and networking*. ACM, 2017, p. 16.

[20] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, 2017.

[21] K. Velasquez, D. P. Abreu, D. Gonçalves, L. Bittencourt, M. Curado, E. Monteiro, and E. Madeira, "Service orchestration in fog environments," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2017, pp. 329–336.

enabled infrastructures," in *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*. IEEE, 2017, pp. 127–132.

[23] D. Santoro, D. Zozin, D. Pizzolli, F. De Pellegrini, and S. Cretti, "Foggy: a platform for workload orchestration in a fog computing environment," in *Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on*. IEEE, 2017, pp. 231–234.

[24] M. Villari, A. Celesti, G. Tricomi, A. Galletta, and M. Fazio, "Deployment orchestration of microservices with geographical constraints for edge computing," in *Computers and Communications (ISCC), 2017 IEEE Symposium on*. IEEE, 2017, pp. 633–638.

[25] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, 2016.

[26] C. Pahl, S. Helmer, L. Miori, J. Sanin, and B. Lee, "A container-based edge cloud paas architecture based on raspberry pi clusters," in *Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on*. IEEE, 2016, pp. 117–124.

[27] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, "Small cell clustering for efficient distributed fog computing: A multi-user case," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*. IEEE, 2015, pp. 1–5.

[28] T. H. Noor, Q. Z. Sheng, A. H. Ngu, and S. Dustdar, "Analysis of web-scale cloud services," *IEEE Internet Computing*, vol. 18, no. 4, pp. 55–61, 2014.

[29] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2017.

[30] M. Lu, A. Fan, J. Xu, and W. Shan, "A compact, lightweight and low-cost 8-bit datapath aes circuit for iot applications in 28nm cmos," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 1464–1469.