Journal of Universal Computer Science, vol. 27, no. 4 (2021), 341-363
submitted: 13/5/2020, accepted: 9/3/2021, appeared: 28/4/2021

Formal Verification of Cloud and Fog Systems: A Review and Research Challenges

Fairouz Fakhfakh
(ReDCAD, University of Sfax, Sfax, Tunisia
https://orcid.org/0000-0002-4196-0986, fairouz.fakhfakh@redcad.tn)

Slim Kallel
(ReDCAD, University of Sfax, Sfax, Tunisia
https://orcid.org/0000-0002-2824-167X, slim.kallel@redcad.tn)

Saoussen Cheikhrouhou
(ReDCAD, University of Sfax, Sfax, Tunisia
https://orcid.org/0000-0003-4607-7452, saoussen.cheikhrouhou@redcad.tn)

Abstract: Cloud and Fog computing have been widely recognized as attractive solutions in both academic and industrial sectors. Despite their benefits, the adoption of Cloud and Fog computing still have considerable challenges to be handled due to the increase of client requirements. A crucial issue, in this context, is how to verify the correctness of Cloud and Fog systems. The use of formal methods is an efficient mean which provides a real help for the designer to evaluate the behaviour of a system and prevent errors before its implementation. In this paper, we present a systematic literature review (SLR) on the current state of the art in this field. We collect the existing studies on the use of formal methods for proving the correctness of Cloud and Fog systems. The proposed approaches are compared based on some technical properties such as the verification methods, the verification tools, the considered properties, and the application domains. In addition, future directions which need more investigations are presented. We believe that our paper will be useful for industry and academic researchers to understand the existing contributions that deal with the correctness of Cloud and Fog systems. Moreover, it helps them to address several gaps in the literature.

Keywords: Formal verification, Cloud computing, Fog computing, Systematic literature review, Future directions
Categories: A.1, D.2.1, D.2.4
DOI: 10.3897/jucs.66455

1 Introduction

Cloud computing paradigm has known a significant interest in the past decades in both academic and industrial sectors. It offers on-demand access to different types of virtualized resources (virtual machines, servers, etc.) [JoSEP, 2010]. Despite the several features that make it very attractive, the centralized nature of this new paradigm leads to some shortages related basically to the considerable distance between the Cloud servers and the users'devices. Thus, the cost of Cloud resources as well as the bandwidth use are getting more and more expensive. Recently, the adoption of Fog computing as an extension of the Cloud has become an efficient solution which can overcome several problems of Cloud computing based systems. In fact, Fog computing is located between the Internet of Things (IoT) devices and the Cloud servers. In addition, it has weak performance devices which are closer to the end users terminals. Consequently, it can reduce transmission latency and monetary cost and support of user's mobility [Lin, 2016]. A Cloud-Fog computing system consists of three layers in a hierarchical network. The bottom layer includes IoT devices, which are used as user interfaces that transmit requests from users. The middle layer is formed by a set of Fog nodes which are distinguished by their proximity to end-users and the support of mobility. It receives the users' requests. Fog nodes need to connect to the upper layer (Cloud layer) which hosts a set of heterogeneous Cloud resources of a Cloud service provider. Both Cloud and Fog resources can have computing, storage or network capabilities.

One of the key challenges in this context is how to validate the features of Cloud and Fog systems. These systems can be validated using several methods such as simulation, formal verification, real experiments, testing and runtime check. In the literature, simulation and formal verification are the most used strategies. Simulation is an efficient way to evaluate several performance parameters such as execution time, monetary cost, and energy consumption. It is relatively inexpensive in terms of execution time. However, it does not guarantee that systems behave correctly in all possible situations. Given the complexity of the studied applications, their modeling is more prone to errors on the part of the designers. The deployment of erroneous systems can cause a disruption of work procedures and consequently a waste of time, a poor Quality of Service (QoS) level, and client complaints. Thus, it is essential to detect possible errors in the design phase of systems.

The use of a formal method to describe Cloud and Fog systems has become very useful for checking their correctness. Indeed, formal verification provides an effective way for the designer to evaluate the behaviour of a system and prevent errors before the implementation. It consists in describing the properties to be proven without worrying about the possible scenarios.

In the literature, numerous research studies [Sahli, 2017] [BenHalima, 2018] [Amato, 2018] [Fan, 2018] [Zahra, 2017] [Cheikhrouhou, 2019] have been proposed to address the correctness problem for Cloud and Fog systems. They relied on different formal methods and tools to verify functional and non functional properties. In [Souri, 2018], the authors present a survey in which they presented a state of the art of the formal verification approaches in the Cloud environment. Additionally, they provide an insight about the formal methods adopted in each work. However, they do not consider existing works related to Fog computing. To the best of our knowledge, our paper is the first systematic review that deals with the correctness problem in Cloud and Fog environments. The main purpose of this paper is to comprehensively categorize and examine current research studies on the formal verification in the Cloud and Fog computing. To this aim, we review the current state of the art in this topic. We focus on software-oriented correctness verification.

The reviewed papers aim to verify behavioral and non-functional properties, which are necessary for the well-functioning of Cloud and Fog systems. Most of these papers verify the allocation of cloud and/or fog resources. Such property allows to optimize the use of resources which reflects on the cost and the execution time of these systems. Other papers propose different approaches to verify safety and security properties, which also considered as main challenges of Cloud and Fog systems. Other properties like mobility, elasticity and temporal properties are also verified.

The remainder of this paper is organized as follows: Section 2 provides the research methodology applied to our survey paper. In Section 3, we give an overview of the different formal verification methods. Besides, we review the approaches of the selected papers while classifying them according to the adopted verification methods. Moreover, a comparison of these approaches is presented based on some criteria. In Section 4, a discussion and a rich evaluation of the existing approaches are presented. Finally, the last section sums up the paper and identifies some relevant challenges to address in the future.

2 Research selection method

The analysis process of this survey paper is based on the rigorous methodology proposed by Kitchenham [Kitchenham, 2009] to guarantee the reliability of the selected articles. This methodology consists of three main phases: planning the review, conducting the review, and reporting the review. The detailed process of this SLR is shown in Figure 1. The following subsections present an overview description of these phases.

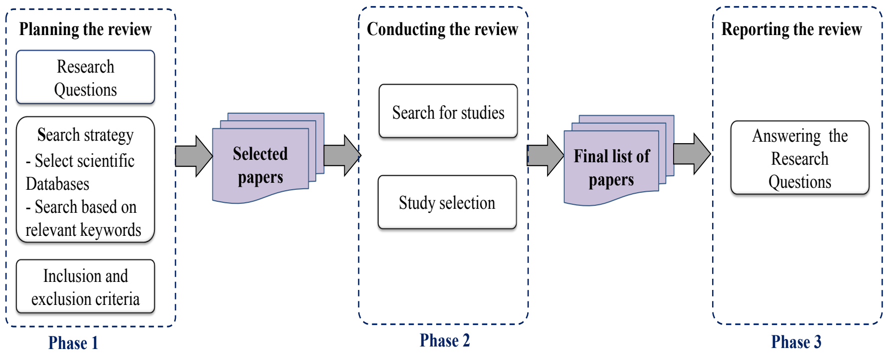

Figure 1: Description of the SLR process

2.1 Planning the review

The first phase of a SLR consists in defining the objectives of this survey by using a set of research questions (RQs). The present paper aims at exploring the contribution of formal verification approaches for Cloud and Fog systems. To do so, we are going to study the reviewed papers through five RQs aiming at having a holistic view about the existing works.

- **RQ1:** Which formal verification methods are typically used for Cloud and Fog approaches ?
- **RQ2:** Which formal modeling languages are adopted to specify Cloud and Fog systems ?
- **RQ3:** Which verification tools are used by each approach ?
- **RQ4:** Which correctness properties are verified in the Cloud and Fog computing approaches ?

– **RQ5:** What are the studied application domains addressed in the existing works ?

Obviously, various other questions can be posed, but we believe that these are sufficient to expose a vast variety of research in the studied field. In order to select relevant publications, we have conducted an extensive search using relevant keywords. Our search query is shown in Figure 2. Its uses synonyms and various words of the main features.

(“formal verification” **OR** “formal model” **OR** “formal specification” **OR** “formal reasoning” **OR** “formal method” **OR** “model checking” **OR** “property checking” **OR** “proof” **OR** “proving” **OR** “prove”) **AND** (“Cloud computing” **OR** “Fog computing”)

Figure 2: Search query

Our search consists in using the known scientific databases: ACM Digital Library[1], IEEE Xplore Digital Library[2], ScienceDirect (Elsevier)[3], Springer[4] and Wiley[5]. We have selected only long papers which are written in English and published in international conferences, workshops, and journals. Based on this phase, we select the studies published between 2015 and February 2020 were found.

2.2 Conducting the review

In the second phase, we have examined all the papers to decide which one respects the inclusion and exclusion criteria. These criteria consist in omitting low-quality papers which do not have pertinent scientific explanations. To do this, we have considered some quality criteria such as the clarity of the research purpose, the feasibility of the proposed solution, the result clarity, and the quality of writing. We have also discarded other publications such as presented slides, summaries, theses, short studies, book chapters, and studies presented survey papers. Based on this phase, we have identified a set of articles related to the formal verification in the context of Cloud and Fog computing.

2.3 Reporting the review

In this phase, we use the five research questions already presented. The first question RQ1 is so useful as it aims to classify the selected papers into four main categories according to the verification method: handwritten proofs [Mendes, 2018], model checking [Baier & Katoen, 2008], and theorem proving [Cook, 1971]. To respond to RQ2, we identify the formal modeling languages used to specify Cloud and Fog systems such as Petri Nets [Billington, 2003], Coloured Petri Nets [Jensen, 1991] , and Event-B. The answer to RQ3 shows the verification tools (such as Rodin [Abrial, 2010a], CPN[6], etc.) that have been used by the researchers to ensure the correctness of the studied systems. For RQ4, we need

[1] http://dl.acm.org
[2] https://ieeexplore.ieee.org
[3] http://sciencedirect.com
[4] http://www.springer.com
[5] http://onlinelibrary.wiley.com
[6] http://cpntools.org/

to identify the correctness properties in the existing approaches. The supported Cloud and Fog requirements can be categorized into functional or non-functional properties. These requirements deal with several characteristics of Cloud and Fog systems. The functional properties aim to evaluate the reaction of a system to given inputs. They are related to problems in Cloud layers such as safety, deadlock, soundness, and reachability. The non-functional properties describe the behaviour of the system like time-related constraints. Finally, in the final research question RQ5, we categorize the application domains considered by the researchers. We notice that the existing approaches are related to different aspects which are linked to business processes, data storage, healthcare, etc.

3 Formal verification approaches

In this section, we present a review of the most relevant formal verification approaches in the Cloud and Fog computing for the selected papers according to the SLR method. As shown in Table 1, we classify these approaches into three types based on the adopted verification methods including handwritten proofs, model checking, and theorem proving. In addition, we embed the references of the studied papers in each method.

The surveyed papers verify different properties which are :

- Resource properties: They consist in checking the matching between tasks and Cloud resources according to some constraints such as shareability [Mohamed, 2015] [Keshanchi, 2017] [Zahra, 2017] [Etchevers, 2017] [Sahli, 2017] [Khebbeb, 2018] [Kochovski, 2019] [Bouanaka, 2019] [Latreche, 2019] [Khebbeb, 2020b] [Boubaker, 2016] [Graiet, 2017] [Fakhfakh, 2018].
- Elasticity properties: They consist in taking into account the vertical and horizontal elasticity which refer respectively to adding/reducing resources capacity assigned to a task and to adding additional resources or removing them when necessary [Moudjari, 2018] [Khebbeb, 2018] [Bouanaka, 2018] [Bouanaka, 2019] [Khebbeb, 2020a].
- Temporal properties: Temporal properties of resources are generally related to pricing strategy. They can be classified into two types: relative and absolute constraints. Relative temporal constraints specify a time interval in which a resource is available at a certain price. Absolute temporal constraints specify the start and finish times of a resource availability at a certain price [BenHalima, 2018] [Latreche, 2019] [Cheikhrouhou, 2019] [Khebbeb, 2020b].
- Security properties: They often aim to guarantee data's confidentiality and users' privacy for Cloud and Fog systems [Zeng, 2016] [Berrima, 2017] [Zahra, 2017] [Puthal, 2018] [Ouchani, 2018] [Bouheroum, 2019] [Alam, 2017].
- Safety properties: They represent requirements that should be continuously maintained by Cloud and Fog systems [Ouchani, 2018] [Toor, 2019].
- Deadlock: It represents a situation in which different processes block each other. For example, a deadlock might occur when a resource is waiting for acknowledgement [Jung, 2015] [Zitouni, 2019].
- Network connectivity: It aims to ensure that each resource is connected with a sink resource [Jung, 2015].

Formal verification approaches		
Handwritten	Model checking	Theorem proving
[Du, 2017] [Kumari, 2017] [Huang, 2016] [Shah-Mansouri, 2018] [Zhang, 2017]	[Mohamed, 2015] [Jung, 2015] [BenHalima, 2016] [Zeng, 2016] [Berrima, 2017] [Keshanchi, 2017] [BenHalima, 2018] [Puthal, 2018] [Zahra, 2017] [Kochovski, 2019] [Toor, 2019] [Cheikhrouhou, 2019] [Etchevers, 2017] [Sahli, 2017] [Ouchani, 2018] [Khebbeb, 2020a] [Moudjari, 2018] [Khebbeb, 2018] [Bouanaka, 2018] [Kochovski, 2019] [Zitouni, 2019] [Bouanaka, 2019] [Latreche, 2019] [Bouheroum, 2019] [Khebbeb, 2020a] [Khebbeb, 2020b]	[Fakhfakh, 2018] [Graiet, 2017] [Lahouij, 2018] [Boubaker, 2016] [Alam, 2017]

Table 1: Taxonomy of the formal verification approaches

– Mobility properties: They aim at providing and keeping resources close to where the data is generated [Khebbeb, 2020b].

3.1 Handwritten proofs-based verification approaches

Handwritten proofs represent a verification method done based on mathematical analysis using paper and pencil. In this subsection, we detail the selected handwritten proofs-based approaches in the Cloud and Fog computing. Next, a comparison of the reviewed approaches is presented.

The approach presented in [Du, 2017] addresses the problem of resource allocation for soft real time applications which are subject to QoS constraints on timely task completion. To this aim, the authors introduce a framework for scheduling tasks on multiple resources. Additionally, they study the efficiency of two existing policies of resources allocation. In order to evaluate the performance of these policies in terms of meeting QoS requirements, they present formal proofs. Their goal is to analyze the performance of scheduling under deterministic workloads, the number of cores needed to satisfy the user's requirements, and the time wasted on each resource core.

In [Shah-Mansouri, 2018], the authors deal with the allocation of Fog and Cloud resources to IoT users while maximizing their quality of experience. Each user can offload its tasks to various Fog or remote Cloud nodes. To model the competition among users, the authors adopt the game approach with the goal of reducing delay and energy. In addition, they present some manual proofs which formulate and verify several properties related to the game.

In another proposal, Kumari et al. [Kumari, 2017] design an authentication scheme for a multi-Cloud environment. This scheme is based on a biometric mechanism to ensure security. In order to show the utility of the proposed scheme, the authors introduce formal proofs which verify the correctness of some security attributes.

The work presented in [Huang, 2016] tackles the verification problem of Vehicular Cloud Computing systems. It presents an algorithm which aims to select reliable vehicles in order to set up the temporary vehicular Cloud. Based on this algorithm, the Cloud user can effectively locate certain vehicles that respond to their requirements while reducing the privacy revelation of location. He can also check the correctness of computation while ensuring the privacy of the outsourced data.

Zhang et al. [Zhang, 2017] introduce a framework for the optimal allocation of resources among Fog nodes to IoT users . To do so, they develop a stakelberg game algorithm to ensure the interaction between data service subscribers and data service operator. In addition, they present hand-written proofs to verify some properties related to this interaction.

According to the surveyed approaches based on handwritten proofs, we illustrate in Table 2 a comparison of the selected studies based on some criteria, including the verified properties, and the application domain. In addition, we indicate whether the authors have considered only Cloud environment or a collaboration between Cloud and Fog computing.

References	**Verified properties**	**Application domains**	**Cloud/Fog**
[Huang, 2016]	Security	Transportation	Cloud
[Kumari, 2017]	Security	Biometrics-based authentication scheme	Cloud
[Du, 2017]	Resource allocation	Real-time applica-tions	Cloud
[Zhang, 2017]	Resource allocation	IoT	Cloud +Fog
[Shah-Mansouri, 2018]	Resource allocation	IoT	Cloud +Fog

Table 2: Comparison of handwritten proofs-based approaches

3.2 Model checking-based verification approaches

Model checking is one of the most promising verification methods which provides a platform for evaluating and proving the correctness of hardware and software systems. It relies on automated techniques that can perform fast evaluation. Figure 3 illustrates the main idea of the model checking method. The latter exhaustively explores every possible system behavior, to check automatically whether the specification is satisfied. In addition, this method consists in producing a counter-example for each not satisfied property.

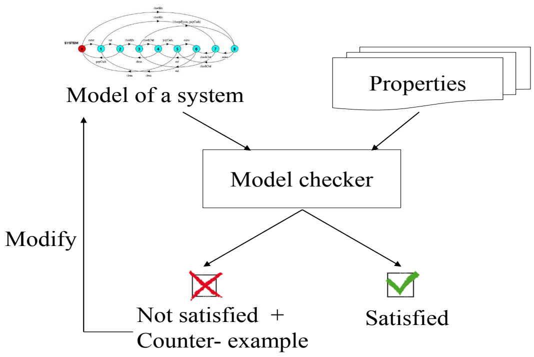

Figure 3: The principle of model checking

We distinguish different model checking based languages such as Timed automata [Alur & Dill, 1994] and Petri Nets [Billington, 2003]. This subsection reviews the model checking approaches focusing on the verification of Cloud and Fog computing systems. After that, a comparison of these approaches based on some criteria is presented.

The work presented in [BenHalima, 2016] introduces a formal approach which describes the pricing models and time dimension of Cloud resources. This approach also ensures a consistent allocation of Cloud resources while taking into account the temporal constraints of activities. The authors identify four classes of temporal properties: (1) duration constraints of each activity, (2) temporal dependency between two activities, (3) absolute temporal constraints of activities, and (4) temporal availability of Cloud resources based on the pricing models. They present a mapping step which consists in assigning the strategies of Cloud-pricing into Timed automata. In order to validate the allocation and the defined requirements, they use the UPPAAL [Behrmann, 2004] model checker to verify whether the temporal constraints of resources satisfy those of activities.

In [BenHalima, 2018], the same authors extend BPMN[7] modeling language to support the specification of Cloud resources and the pricing strategies used by Cloud providers. Next, they implement some mapping rules which generate automatically timed automata from BPMN models. The generated models are the input of UPPAAL to verify the assignment between the temporal constraints of activities and Cloud resources. The contribution of this work is added as an Eclipse[8] plugin.

[7] http://www.bpmn.org/
[8] https://www.eclipse.org/

Another recent approach is proposed by Cheikhrouhou et al. [Cheikhrouhou, 2019]. It consists in formally specifying, verifying, and deploying time-aware business processes in a mono-Cloud and multi-Fog context. The studied business processes are limited by time constraints whose satisfaction requires placing their activities and data in the adequate hosts, whether Cloud, Fog, or either.

In another proposal, Keshanchi et al. [Keshanchi, 2017] introduce an improved genetic algorithm for task scheduling in Cloud environment. This algorithm consists in initializing the population. Next, the assignment of tasks to resources is performed and the fitness value of each chromosome is determined. The elitism selection policy is used to select the best chromosome at each iteration. To analyze the correctness of the algorithm, a behavioural model is proposed and verified using PAT[9] and NuSMV [Cimatti, 2000] model checkers. The verification results of theses tools are compared in terms of the verified properties.

In [Mohamed, 2015], the authors propose an automatic model for the management of Cloud resources. To do so, they extend the Open Cloud Computing Interface [Metsch, 2010] for describing this model. In addition, a formal approach for the elasticity of business processes is presented. In order to demonstrate the efficiency of this approach, realistic scenarios of Cloud environment are introduced.

Etchevers et al. [Etchevers, 2017] introduce a self-deployment protocol for running various virtual machines (VMs) in the Cloud. This protocol includes a platform which facilitates the interaction between users ans IaaS based applications. It can also detect network and VMs failures and handle them. In order to evaluate the correctness of the proposed protocol, the authors use LOTOS [Champelovier, 2011] specification language. They are interested in verifying some properties which must be satisfied in the design phase and during the protocol execution.

Another trend [Puthal, 2018] aims at proposing a new load balancing architecture which consists of two main parts. The first one aims to ensure secure authentication of Fog datacenters before task allocation. The second part focuses on getting load information. In order to ensure the correctness of the proposed solution, they present a formal security model proved using Scyther tool[10] which adopts Security Protocol Description Language (SPDL). In this model, they define two roles which are intended to the authentication initiator and the authentication destination.

In another proposal [Ouchani, 2018], the author proposes a security analysis methodology to verify security and functional properties of IoT systems. To prove the functional correctness of these systems, five steps are developed. These steps consists in defining the IoT components and formalizing the system architecture. After that, the IoT requirements are expressed in probabilistic computation tree logic (PCTL) and the IoT model is given as input of PRISM language. Finally, PRISM verifies the correctness of the requirements on the IoT model.

Furthermore, in [Zahra, 2017], the authors deal with the security issues encountered when outsourcing data from the Fog client to the Fog node. They propose a control algorithm which ensures a secure communication between client and node. This algorithm is based on Shibboleth protocol which guarantees user's privacy and authentication. To prove the correctness of the proposed algorithm, the authors formally verify some basic security properties using High Level Petri Nets (HLPN) [Jensen, 1983] and Z3 solver [de Moura, 2008]. They adopt the deductive verification which is one of the most rigorous techniques to ensure software satisfies its requirements.

---

[9] http://pat.sce.ntu.edu.sg/

[10] https://people.cispa.io/cas.cremers/scyther/

In [Toor, 2019], an energy-aware scheme is introduced for Fog-IoT environments. It consists in modifying the devices speed based on dynamic frequency scaling to overcome energy consumption. The correctness of the proposed scheme is formally verified through UPPAAL tool while adopting Blockchain technology to secure transactions. Some safety properties related to the weather and battery behaviours are considered.

Zitouni et al. [Zitouni, 2019] introduce an architecture to control the urban traffic light. Their main purpose is to use IoT platform to interconnect traffic lights and street'infrastructures. In order to prevent inconsistent cases, the authors specify and verify the traffic light states using timed automata. They guarantee that the change of lights' colors is based on the arrival of new priority vehicle.

In [Jung, 2015], the authors propose an integration platform which consists of a wireless body area network and Cloud Computing. This platform relies on a transmission control protocol which ensures the communication between network devices. A formal model based on timed automata is developed to ensure the construction of efficient networks. In fact, it enables the verification of some specification properties (such as the timing parameters of nodes, network connectivity, and the absence of deadlock) and the localization of any fault in the protocol design.

Authors in [Kochovski, 2019] present a decision making method which ensures the placement of databases deployed in a Cloud-Fog environment. This method uses stochastic Markov models (MDP) to provide a formal guarantee to software engineers. A new orchestration approach is introduced to automate the whole process using PRISM tool[11].

Sahli et al. [Sahli, 2017] address the behavioural and structural aspects of elastic systems based on Cloud Computing. These aspects increase the difficulty of modeling and implementing such systems. In order to handle this problem, the authors propose a formal model based on Bigraphical Reactive Systems (BRS) [Milner, 2001] and using BigMC tool [Perrone, 2012]. Particularly, they use bigraphical reaction rules to describe the behaviour of Cloud-based elastic systems in terms of client/application interactions and elasticity strategies. An example running on the top of Amazon Elastic Compute Cloud [Varia, 2014] infrastructure is presented to demonstrate the performance of the proposed approach.

In [Bouanaka, 2019], Bouanaka et al. propose a formal model for specifying cloud systems structure and its dynamics in terms of quality driven elasticity strategies. This model aims at resolving the non-determinism problem that can take place while selecting the elasticity policy. This is achieved by quantifying the elasticity strategies cost and resource usage to have a fully-probabilistic model that associates a cost attribute to each resource.

In [Khebbeb, 2020b], Khebbed et al. introduce a new formal model to verify Cloud-Fog self-adaptation that aims et achieving a trade-off between low latency and resources quantity. This model consists in modeling Cloud and Fog layers to identify the required adaptation actions. Also, an orchestrator is used to decide which action must be triggered to adapt Cloud and/or Fog layers.

The recent work of Khebbeb et al. [Khebbeb, 2020a] consists in proposing a formal approach based on BRS to specify the elastic behaviours of Cloud systems. It focuses on application and infrastructure Cloud layers to handle resources provisioning and deprovisioning. In addition, various strategies that enable horizontal and vertical scale are introduced and verified.

According to the reviewed approaches based on the model checking method, we

[11] www.prismmodelchecker.org

present in Table 3 their comparison in terms of some metrics. These metrics include the formal modeling language, verification tool, verified properties, application domain, and execution environment.

3.3 Theorem proving-based verification approaches

Theorem proving is a powerful correctness method that is based on mathematical logic. It can deal with complex formalisms and ensure the correctness of properties which contain assumptions and fundamental theory. In Figure 4, we present the main idea of the theorem proving method. The latter consists in expressing the system behavior as mathematical formulas and generating a set of proof obligations using a proof assistant. These proof obligations can be proved automatically or semi-automatically with human interaction. If a proof obligation cannot be verified, it is essential to modify the formal specification.

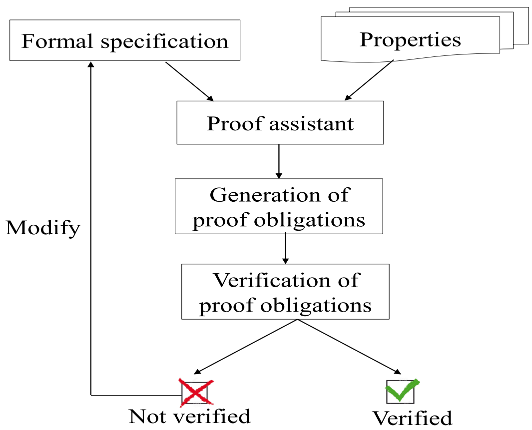

Figure 4: The principle of theorem proving

The most popular formal methods based on the theorem proving are Coq[13], Z [Spivey, 1992], and Event-B [Abrial, 2010]. In this subsection, we review the selected approaches which have adopted theorem proving for ensuring the correctness of Cloud and Fog computing systems. These papers are based on deductive verification technique that consists in carrying out derivations and establishing a rigorous mathematical proof that a given program meets its specification. A comparison of the presented approaches is illustrated in Table 4.

Boubakar el al. [Boubaker, 2016] introduce a formal model for verifying the correctness of Cloud resources assignment using the Event-B method. This model specifies some properties related to the resource requirements both at design time and at runtime. In addition, the authors define the different states of a resource instance during its life cycle. They use a real scenario of a business process from France Telecom/Orange labs to illustrate their approach. The presented scenario is a supervision process which deals

[12] http://www.tcs.hut.fi/Software/maria/index.en.html
[13] https://coq.inria.fr/

References	Modeling languages	Verification Tools	Verified properties	Application domains	Cloud Fog
[Mohamed, 2015]	HLPN	SNAKES	Resource allocation	Business process	Cloud
[Jung, 2015]	Timed automata	UPPAAL	Deadlock+ Network connectivity	Healthcare	Cloud
[Zeng, 2016]	Petri nets	Maria[12]	Security	Business process	Cloud
[Berrima, 2017]	Applied Pi Calculus	ProVerif	Security	Data storage	Cloud
[Keshanchi, 2017]	Petri nets +LTL	NuSMV +PAT	Resource allocation	Business Process	Cloud
[Zahra, 2017]	HLPN	Z3 solver	Security	IoT	Cloud +Fog
[Etchevers, 2017]	LOTOS	CADP	Safety	Data storage	Cloud
[Sahli, 2017]	BRS	BigMC	Resource allocation	Business process	Cloud
[BenHalima, 2018]	Timed automata	UPPAAL	Temporal properties +Resource allocation	Business process	Cloud
[Puthal, 2018]	SPDL	Scyther	Security	IoT	Cloud +Fog
[Ouchani, 2018]	PCTL	PRISM	Security +Safety	IoT	Cloud +Fog
[Moudjari, 2018]	BRS	BigMC	Elasticity	Transportation	Cloud
[Khebbeb, 2018]	BRS +Maude	BigMC	Elasticity +Resource allocation	-	Cloud
[Bouanaka, 2018]	PSMaude	PCTL	Elasticity	Booking system	Cloud
[Kochovski, 2019]	MDP	PRISM	Resource allocation	IoT	Cloud +Fog
[Zitouni, 2019]	Timed automata	UPPAAL	Deadlock	Transportation	Cloud +Fog
[Toor, 2019]	Timed automata	UPPAAL	Safety	Smart cities	Cloud +Fog
[Bouanaka, 2019]	PSMaude	PSMaude	Elasticity +Resource allocation +Mobility	Booking system	Cloud
[Latreche, 2019]	Maude	Timed Computational Tree Logic	Resource allocation +Temporal properties	Real time applications	Cloud
[Bouheroum, 2019]	BRS	BigraphER	Security	Oil and Gas Refinery Plant	Cloud +Fog
[Cheikhrouhou, 2019]	Timed Petri Net	TINA	Temporal properties	Business process	Cloud +Fog
[Khebbeb, 2020a]	Maude	BRS +LTL	Elasticity	The Steam digital library	Cloud
[Khebbeb, 2020b]	Maude +LTL	Maude	Temporal properties +Resource allocation +Mobility	Smart cities	Cloud +Fog

Table 3: Comparison of model checking approaches

with complaints signals raised by a customer. The activities of this process need three types of Cloud resources (compute, network, and storage).
In the same context, Graiet et al. [Graiet, 2017] extend the formal model already proposed to check the behavioural inconsistencies that may occur during execution thanks to the concept of events in Event-B. In fact, events can specify runtime behaviour of the activity execution while consuming a Cloud resource. A case study from an industrial partner is used to demonstrate the feasibility of the the proposed approach.

The contribution of the approach cited in [Lahouij, 2018] ensures the correctness of Cloud composite services. It aims to avoid incorrect composition behaviour and unnecessary execution for an erroneous composition. The formalization of the proposed model is developed using the Event-B formal method. Two abstraction levels are presented. The first one specifies the behaviour of Cloud composite services and the second level aims to verify the resource allocation. The correctness of the proposed formalization is verified by means of proof obligations and Prob tool [Leuschel, 2008] which is integrated in the Rodin platform [Abrial, 2010a].

Another attempt to guarantee the correctness of Cloud resource allocation is described by Fakhfakh et al. [Fakhfakh, 2018]. It introduces a formal model using the Event-B method. This model is based on the refinement technique to check the correctness of resources behaviour for dynamic Workflow applications. This model takes into account the shareability property which consists in assigning a resource to more than one task instance. In order to respond to the requirements of dynamic Workflows at runtime, they specify the elasticity property of resources and the life cycle of a resource instance. The authors consider three adaptation actions which aim at inserting, deleting, and substituting a task. In Listing 1, we present an example of four invariants specified using the Event-B formal method. The first one (inv1) introduces the variable $WF\_Resources$ that represents the resources used to run a given workflow. This variable is a subset of RES which defines all the available resources. In addition, the allocation dependency between a task and its resources is formally defined through the variable AllocDep (inv2). inv3 specifies that a cloud resource can be shareable or non-shareable. Then, a new variable, named "Shareable" is defined as a total function which determines whether the resource is shareable or not. According to inv4, each resource assigned to many tasks is necessarily shareable.

inv1 : $WF\_Resources \subseteq RES$
inv2 : $AllocDep \in WF\_Resources \rightarrow \mathbb{P}(Tasks)$
inv3 : $Shareable \in WF\_Resources \rightarrow BOOL$
inv4 : $\forall res \cdot res \in WF\_Resources \wedge (card(AllocDep[\{res\}]) > 1 \Rightarrow Shareable(res) = TRUE)$

Listing 1: An example of Event-B code [Fakhfakh, 2018]

The contribution of Alam et al. [Alam, 2017] introduces a model which facilitates the resource sharing amongst different tenants. It also present four algorithms which aim at capturing the manner in which a user activates, delegates or revokes a permission. In order to formalize these algorithms, the authors use Z language [Bowen, 1995]. The correctness and the security of these algorithms are demonstrated thought Z3 solver[14].

According to the reviewed approaches based on theorem proving method, we depict in Table 4 a comparison of the existing studies in terms of some criteria.

[14] https://github.com/Z3Prover/z3/wiki

References	Modeling languages	Verification Tools	Verified Properties	Application domains	Cloud/Fog
[Boubaker, 2016]	Event-B	Rodin	Resources allocation	Customer signalization process	Cloud
[Graiet, 2017]	Event-B	Rodin	Resource allocation	Business process	
[Alam, 2017]	Z language	Z3 solver	Security	Process of delegating a permission	
[Fakhfakh, 2018]	Event-B	Rodin	Resource allocation	Business process	
[Lahouij, 2018]	Event-B	Rodin	Deadlock +Resource allocation	Service composition	

Table 4: Comparison of theorem proving approaches

4 Analytical Discussion

In this section, we present an analytical discussion related to the studied approaches of formal verification in Cloud and Fog computing. Our discussion responds to the research questions introduced in Section 2.

- RQ1: *Which formal verification methods are typically used for Cloud and Fog approaches ?*

  In order to answer RQ1, we study the distribution of formal verification methods that are applied for Cloud and Fog approaches. We can see that most of the reviewed articles (61%) have used the model checking method to ensure the correctness of Cloud and Fog systems. Indeed, this method is easy for developers to understand and it relies on automated techniques that can achieve a faster evaluation. Some papers have adopted handwritten proofs-based method which does not require time and effort to master a formal language. Nevertheless, it is error-prone especially in the case of complex systems. So, it is acceptable that it has 25% usage to evaluate Cloud and Fog approaches. In addition, we notice that only few papers (14%) have adopted theorem proving despite the effective correctness of this method that can deal with complex formalisms. However, it consists in using hard-proof mechanisms which require the user's interaction such as using proof tactics and adding invariants.

- RQ2: *Which formal modeling languages are adopted to specify Cloud and Fog systems ?*

  To answer to RQ2, we study the percentage of the modeling languages used to specify Cloud and Fog computing systems. We observe that Timed automata is the most used (15%) in model checking based approaches. It is a graph based notation which formalizes the behavior of time-constrained systems. In addition, Event-B is a popular modeling language that has 15% usage for theorem proving based approaches. It makes the proofs easier through the use of the refinement technique.

- RQ3: *Which verification tools are used by each solution ?* To answer to RQ3, we study the statistical percentage of the verification tools applied in this literature review based on the studied papers. We notice that most of the existing approaches have adopted UPPAAL (14%) and Rodin (14%) tools for the verification of Cloud and Fog systems.

- RQ4: *Which correctness properties are verified in the Cloud and Fog Computing approaches ?*

  With regard to RQ4, we analyze the repartition of the supported properties based on the surveyed papers. We notice that the largest rate of the existing works (33%) is interested in verifying properties related to the resource allocation. Besides, a great attention (27%) has been given to the correctness of safety properties. Moreover, multiple research efforts (25%) have attempted to verify security-related properties. Also, a little consideration (8%) has been given to the temporal properties of Cloud and Fog systems. Finally, there are only 7% of the studied publications that have considered other properties such as deadlock and mobility.

- RQ5: *What are the studied application domains addressed in the existing works ?*

  In order to answer RQ5, we study the distribution of the application domains in the studied papers. We notice that Cloud and Fog Computing are increasingly used (36%) for deploying and executing business processes. In fact, they allow companies to optimize their processes by offering virtualized resources on demand. Then, the execution can be performed with low operating cost and at a high level of performance. Besides, we can see that 23% of the reviewed articles are interested in IoT based systems which have become very useful in everyday life. Moreover, transportation systems have been exhaustively verified (20%) in Cloud and Fog environment. Indeed, these systems have improved the efficiency of road safety as well as the comfort of both passengers and drivers. Some research efforts attempt to verify data storage systems (5%) and real time applications (2%) using formal methods. Finally, the healthcare and smart city domains are not well addressed in the literature despite their importance for the human life.

5 Open issues

Studies show that Fog computing has been more efficient than Cloud computing in the deployment of smart devices and the scheduling of IoT systems which require real-time services. Indeed, Fog computing acts as an intermediate layer between Cloud and client layers to decrease the delay of processing and communication times, as well as the financial cost. As mentioned in [Mahmud, 2018], Fog computing has become one of the major research domain in academia and business perspectives. It has the potential to be adopted in any application that is latency sensitive, such as healthcare, Vehicular Ad-hoc NETworks (VANETs), and smart cities. In this section, we identify the main formal verification subjects in Fog computing that have not been well addressed in the literature to pave the way for future studies as new challenges:

- Resource allocation is one of the major challenge to the adoption of Cloud and Fog Computing. Unlike Cloud resources, Fog devices are highly dynamic and geographically distributed which makes the problem quite difficult. In the literature, several researches have attempted to address the resource allocation problem in Cloud environments. Different assumptions have been considered and various optimisation strategies have been proposed to have an efficient service allocation. These strategies intended for Cloud computing have to be adapted to Fog computing. In addition, several pricing strategies have been adopted by Cloud providers. However, there is a lack of pricing strategies for Fog based services and users encounter difficulty in identifying the appropriate providers. So, the adoption of a proper pricing strategy will be a promising contribution in the domain of Fog computing. In this context, it is important to notice that without verification, the performance of resource allocation algorithms can not be ensured and can not be adopted with confidence especially for real time systems.

- IoT-based healthcare systems provide remote monitoring services and enhance the quality of care. The IoT generates huge amounts of data that require massive storage and computing resources. Cloud computing has been widely adopted to handle these challenges since it provides a low cost deployment, scalable resource allocation, and huge storage capacities. However, it cannot be considered as an efficient solution for healthcare systems which need real-time response in emergency situations. In this context, Fog computing has been introduced as an efficient technology to gather information very quickly and send response in time. Nowadays, there is a significant number of papers that use Fog computing in healthcare systems [Kraemer, 2017] [Kumari, 2018]. Little effort is spent to verify correctness of healthcare systems in Cloud environment. Nevertheless, the formal verification of Fog based solutions for healthcare is an essential issue that has to be considered before their implementation.

- Smart city is a new trend that is developed due the rapid growth of IoT. It aims to make life more comfortable by providing various services enabling the monitoring and the management of remote devices. The Fog Computing architecture has several benefits over the Cloud architecture for the control of smart city and the support of real time interactions. The evaluation of smart cities is a challengeable task for ensuring the correctness of their behaviours using formal methods. Several attempts have adopted Cloud and Fog computing for smart cities. However, only few ones addressed the verification of functional and non-functional properties related to these applications.

- Similar to Cloud computing but with different characteristics, Fog computing devices face serious security problems that have drawn the attention of the research community [Al-Noman Patwary, 2020][Zhang, 2018]. In fact, they are compromised by several attacks that can perform malicious tasks. Security solutions for Cloud cannot directly applied to Fog due to its heterogeneity, mobility, and distribution. Blockchain has been introduced as a promising technology to build a secure infrastructure in Fog [Islam, 2019]. It can be used to ensure the authentication and reputation of IoT devices and Fog nodes. The integration of Blockchain and Fog Computing has been recently explored in the literature. However, the literature still lacks an effective solution to verify the correctness of Blockchain based solutions.

6 Conclusion

In this paper, we addressed the correctness problem of Cloud and Fog systems using formal methods. To this aim, we conducted a systematic literature review based on a set of research studies in this field. For each surveyed paper, we extracted the verification method, the modeling language, the verification tool, and the supported properties. In addition, we identified the application domains of the verification approaches. Furthermore, we presented a critical comparison of the reviewed works. Finally, we emphasized a set of research challenges that can significantly improve the existing works.
This paper could be improved by giving some hints for solving the existing problems for each studied research work. Furthermore, these works could be assessed by (i) applying

metric based indicators for evaluation such as the existence, richness, size and diversity of data sets used for evaluation and the satisfaction level of both functional and non-functional properties such as QoS factors and (ii) qualitative metrics such as effectiveness (i.e., simplicity and accuracy), efficiency (i.e., time taken and features coverage). Furthermore, it could be better to enrich this Cloud/Fog landscape with Blockchain technology since these latters can work hand in hand and give promises to innovate decentralised systems.

Acknowledgements

This work was partially supported by the LABEX-TA project: "Méthodes Formelles pour le Génie Logiciel" (MeFoGL).

References

[Abrial, 2010] Abrial, J.-R. (2010). " Modeling in Event-B: system and software engineering". Cambridge University Press.

[Abrial, 2010a] Abrial, J.-R., Butler, M., Hallerstede, S., Hoang, T. S., Mehta, F., & Voisin, L. (2010a). "Rodin: an open toolset for modelling and reasoning in event-b". Journal of Software Tools for Technology Transfer, 12(6), 447–466.

[Al-Noman Patwary, 2020] Al-Noman Patwary, A., Fu, A., Naha, R. K., Battula, S. K., Garg, S., Kaium Anwarul Patwary, M., & Aghasian, E. (2020). "Authentication, access control, privacy, threats and trust management towards securing fog computing environments: A review". arXiv, (pp. arXiv–2003).

[Alam, 2017] Alam, Q., Malik, S. U. R., Akhunzada, A., Choo, K. R., Tabbasum, S., & Alam, M. (2017). "A cross tenant access control (CTAC) model for cloud computing: Formal specification and verification". Journal of IEEE Transactions on Information Forensics and Security, 12(6), 1259–1268.

[Alur & Dill, 1994] Alur, R. & Dill, D. L. (1994). "A theory of timed automata". Journal of Theoretical Computer Science (TCS), 126(2), 183–235.

[Amato, 2018] Amato, F., Moscato, F., Moscato, V., & Colace, F. (2018). "Improving security in cloud by formal modeling of iaas resources". Journal of Future Generation Computer Systems, 87, 754–764.

[Amziani, 2013a] Amziani, M., Melliti, T., & Tata, S. (2013a). "Formal modeling and evaluation of service-based business process elasticity in the cloud". In Proceedings of the 22nd Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 284–291). Hammamet, Tunisia: IEEE Computer Society.

[Babenyshev & Rybakov, 2010] Babenyshev, S. & Rybakov, V. (2010). "Linear temporal logic ltl: basis for admissible rules". Journal of Logic and Computation, 21(2), 157–177.

[Baier & Katoen, 2008] Baier, C. & Katoen, J.-P. (2008). Principles of model checking. MIT press.

[Behrmann, 2004] Behrmann, G., David, A., & Larsen, K. G. (2004). "A tutorial on uppaal". In Proceedings of the International School on the Formal Methods for the Design of Real-Time Systems (SFM-RT) (pp. 200–236). Bertinoro, Italy: Springer.

[Berrima, 2017] Berrima, M., Lafourcade, P., Giraud, M., & Rajeb, N. B. (2017). "Formal analyze of a private access control protocol to a cloud storage". In Proceedings of the 14th Conference on e-Business and Telecommunications (ICETE) (pp. 495–500). Madrid, Spain.

[Bersani, 2014] Bersani, M. M., Bianculli, D., Dustdar, S., Gambi, A., Ghezzi, C., & Krstic, S. (2014). “Towards the formalization of properties of cloud-based elastic systems”. In Proceedings of the 6th International Workshop on Principles of Engineering Service-Oriented and Cloud Systems (PESOS) (pp. 38–47). Hyderabad, India: ACM.

[Berthomieu & Vernadat, 2006] Berthomieu, B. & Vernadat, F. (2006). “Time petri nets analysis with tina”. In In Proceedings of the Third International Conference on the Quantitative Evaluation of Systems (QEST) (pp. 123–124).

[Billington, 2003] Billington, J., Christensen, S., Van Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., & Weber, M. (2003). “ The petri net markup language: concepts, technology, and tools”. In Proceedings of the International Conference on Application and Theory of Petri Nets (pp. 483–505).: Springer.

[Blanchet, 2016] Blanchet, B., Smyth, B., Cheval, V., & Sylvestre, M. (2016). “Proverif 1.94 pl1: Automatic cryptographic protocol verifier, user manual and tutorial”.

[Botta, 2016] Botta, A., de Donato, W., Persico, V., & Pescapè, A. (2016). “Integration of cloud computing and internet of things: A survey”. Journal of Future Generation Computer Systems, 56, 684–700.

[Bouanaka, 2018] Bouanaka, C., Maatougui, E., Belala, F., & Zeghib, N. (2018). “A quality-driven approach for analyzing elastic cloud computing”. In ICAASE (pp. 84–91).

[Bouanaka, 2019] Bouanaka, C., Maatougui, E., Belala, F., & Zeghib, N. (2019). “A formal quantitative analysis of elastic cloud systems based on psmaude”. Journal of King Saud University-Computer and Information Sciences.

[Boubaker, 2016] Boubaker, S., Mammar, A., Graiet, M., & Gaaloul, W. (2016). “Formal verification of cloud resource allocation in business processes using event-b”. In Proceedings of the 31st International Conference on Advanced Information Networking and Applications (AINA) (pp. 746–753).: IEEE.

[Bouchenak, 2013] Bouchenak, S., Chockler, G. V., Chockler, H., Gheorghe, G., Santos, N., & Shraer, A. (2013). “Verifying cloud services: present and future”. Journal of Operating Systems Review, 47(2), 6–19.

[Bouheroum, 2019] Bouheroum, A., Benzadri, Z., & Belala, F. (2019). “Towards a formal approach based on bigraphs for fog security: Case of oil and gas refinery plant”. In 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud) (pp. 64–71).: IEEE.

[Bowen, 1995] Bowen, J. P. (1995). “Glossary of Z notation”. Journal of Information & Software Technology, 37(5-6), 333–334.

[Champelovier, 2011] Champelovier, D., Clerc, X., Garavel, H., Guerte, Y., Powazny, V., Lang, F., Serwe, W., & Smeding, G. (2011). “Reference manual of the lotos nt to lotos translator (version 5.4)”. Inria/Vasy, 149.

[Cheikhrouhou, 2019] Cheikhrouhou, S., Kallel, S., Guidara, I., & Maamar, Z. (2019). “Business process specification, verification, and deployment in a mono-cloud, multi-edge context”. Computer Science and Information Systems, (pp. 1–21).

[Chen & Wang, 2017] Chen, X. & Wang, L. (2017). “Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method-pepa”. IEEE Communications Letters, 21(4), 745–748.

[Cimatti, 2000] Cimatti, A., Clarke, E., Giunchiglia, F., & Roveri, M. (2000). “Nusmv: a new symbolic model checker”. International Journal on Software Tools for Technology Transfer, 2(4), 410–425.

[Cook, 1971] Cook, S. A. (1971). “The complexity of theorem-proving procedures”. In Proceedings of the 3rd annual ACM symposium on Theory of computing (pp. 151–158).: ACM.

[Cremers, 2014] Cremers, C. (2014). “ Scyther user manual”. Department of Computer Science, University of Oxford: Oxford, UK.

[de Moura, 2008] de Moura, L. & Bjørner, N. (2008). "Z3: An efficient smt solver". In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS) (pp. 337–340). Berlin, Heidelberg: Springer.

[Du, 2017] Du, Y. & de Veciana, G. (2017). "Scheduling for cloud-based computing systems to support soft real-time applications". Journal of ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS), 2(3), 13:1–13:30.

[Etchevers, 2017] Etchevers, X., Salaün, G., Boyer, F., Coupaye, T., & De Palma, N. (2017). "Reliable self-deployment of distributed cloud applications". Software: Practice and Experience, 47(1), 3–20.

[Fakhfakh, 2018] Fakhfakh, F., Kacem, H. H., & Kacem, A. H. (2018). "Towards a formal approach for verifying dynamic workflows in the cloud". In Proceedings of the 15th European Mediterranean & Middle Eastern Conference on Information Systems (EMCIS) (pp. 144–157). Limassol, Cyprus: Springer.

[Fan, 2018] Fan, G., Chen, L., Yu, H., & Liu, D. (2018). "Formally modeling and analyzing cost-aware job scheduling for cloud data center". Journal of Software: Practice and Experience, 48(9), 1536–1559.

[Freitas, 2014] Freitas, L. & Watson, P. (2014). "Formalising workflows partitioning over federated clouds: Multi-level security and costs". In Proceedings of the 8th IEEE World Congress on Services (SERVICES) (pp. 219–226). Washington, DC, USA: IEEE Computer Society.

[Gaetani, 2017] Gaetani, E., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., & Sassone, V. (2017). "Blockchain-based database to ensure data integrity in cloud computing environments". In Proceedings of the 1st Italian Conference on Cybersecurity (ITASEC) (pp. 146–155). Venice, Italy.

[Garfatta, 2018a] Garfatta, I., Klai, K., Graiet, M., & Gaaloul, W. (2018a). "Formal modelling and verification of cloud resource allocation in business processes". In Proceedings of the International Conference on the Move to Meaningful Internet Systems (OTM) (pp. 552–567). Valletta, Malta: Springer.

[Garfatta, 2018b] Garfatta, I., Klai, K., Graiet, M., & Gaaloul, W. (2018b). "Formal modelling of IT resource allocation in business processes". In Proceedings of the 11th IEEE Conference on Service-Oriented Computing and Applications (SOCA) (pp. 227–232). Paris, France.

[Graiet, 2017] Graiet, M., Mammar, A., Boubaker, S., & Gaaloul, W. (2017). "Towards correct cloud resource allocation in business processes". Journal of IEEE Transactions on Services Computing, 10(1), 23–36.

[Hack, 1976] Hack, M. (1976). "Petri net language".

[BenHalima, 2016] Halima, R., Kallel, S., Klai, K., Gaaloul, W., & Jmaiel, M. (2016). "Formal verification of time-aware cloud resource allocation in business process". In Proceedings of the International Conference on the Move to Meaningful Internet Systems (OTM) (pp. 400–417). Rhodes, Greece.

[BenHalima, 2018] Halima, R., Zouaghi, I., Kallel, S., Gaaloul, W., & Jmaiel, M. (2018). "Formal verification of temporal constraints and allocated cloud resources in business processes". In Proceedings of the 32nd IEEE International Conference on Advanced Information Networking and Applications (AINA) (pp. 952–959). Krakow, Poland: IEEE Computer Society.

[Hermanns, 2002] Hermanns, H., Herzog, U., & Katoen, J.-P. (2002). "Process algebra for performance evaluation". Theoretical computer science, 274(1-2), 43–87.

[Holland, 1992] Holland, J. H. (1992). "Genetic algorithms". Journal of Scientific american, 267(1), 66–73.

[Huang, 2016] Huang, C., Lu, R., Zhu, H., Hu, H., & Lin, X. (2016). "PTVC: achieving privacy-preserving trust-based verifiable vehicular cloud computing". In Proceedings of the 59th IEEE Global Communications Conference (GLOBECOM) (pp. 1–6). Washington, DC USA: IEEE.

[Islam, 2019] Islam, N., Faheem, Y., Din, I. U., Talha, M., Guizani, M., & Khalil, M. (2019). "A blockchain-based fog computing framework for activity recognition as an application to e-healthcare services". Future Generation Computer Systems, 100, 569–578.

[Jalali, 2016] Jalali, F., Hinton, K., Ayre, R., Alpcan, T., & Tucker, R. S. (2016). "Fog computing may help to save energy in cloud computing". IEEE Journal on Selected Areas in Communications, 34(5), 1728–1739.

[Jensen, 1983] Jensen, K. (1983). "High-level petri nets". In Applications and theory of Petri nets (pp. 166–180). Springer.

[Jensen, 1991] Jensen, K. (1991). "Coloured petri nets: A high level language for system design and analysis". In High-level Petri Nets (pp. 44–119). Springer.

[JoSEP, 2010] JoSEP, A. D., KAtz, R., KonWinSKi, A., Gunho, L., PAttERSon, D., & RABKin, A. (2010). "A view of cloud computing". Journal of Communications of the ACM, 53(4).

[Jung, 2015] Jung, M., Han, K., & Cho, J. (2015). "Advanced verification on wban and cloud computing for u-health environment". Journal of Multimedia Tools and Applications, 74(16), 6151–6168.

[Keshanchi, 2017] Keshanchi, B., Souri, A., & Navimipour, N. J. (2017). "An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing". Journal of Systems and Software, 124, 1–21.

[Khebbeb, 2018] Khebbeb, K., Hameurlain, N., & Belala, F. (2018). "Modeling and evaluating cross-layer elasticity strategies in cloud systems". In International Conference on Model and Data Engineering (pp. 168–183).: Springer.

[Khebbeb, 2020a] Khebbeb, K., Hameurlain, N., & Belala, F. (2020a). "Formalizing and simulating cross-layer elasticity strategies in cloud systems". Cluster Computing.

[Khebbeb, 2020b] Khebbeb, K., Hameurlain, N., & Belala, F. (2020b). "A maude-based rewriting approach to model and verify cloud/fog self-adaptation and orchestration". Journal of Systems Architecture, 110, 101821.

[Khebbeb, 2019] Khebbeb, K., Hameurlain, N., Belala, F., & Sahli, H. (2019). "Formal modelling and verifying elasticity strategies in cloud systems". Journal of IET Software, 13(1), 25–35.

[Kitchenham, 2009] Kitchenham, B. A., Brereton, P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. G. (2009). "Systematic literature reviews in software engineering - A systematic literature review". Journal of Information & Software Technology, 51(1), 7–15.

[Kochovski, 2019] Kochovski, P., Sakellariou, R., Bajec, M., Drobintsev, P., & Stankovski, V. (2019). "An architecture and stochastic method for database container placement in the edge-fog-cloud continuum". In 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS) (pp. 396–405).: IEEE.

[Kraemer, 2017] Kraemer, F. A., Braten, A. E., Tamkittikhun, N., & Palma, D. (2017). Fog computing in healthcare–a review and discussion. IEEE Access, 5, 9206–9222.

[Kumari, 2018] Kumari, A., Tanwar, S., Tyagi, S., & Kumar, N. (2018). "Fog computing for healthcare 4.0 environment: Opportunities and challenges". Computers & Electrical Engineering, 72, 1–13.

[Kumari, 2017] Kumari, S., Li, X., Wu, F., Das, A. K., Choo, K. R., & Shen, J. (2017). "Design of a provably secure biometrics-based multi-cloud-server authentication scheme". Journal of Future Generation Computer Systems, 68, 320–330.

[Lahouij, 2016] Lahouij, A., Hamel, L., Graiet, M., Elkhalfa, A., & Gaaloul, W. (2016). "A global sla-aware approach for aggregating services in the cloud". In Proceedings of the International Conference on the Move to Meaningful Internet Systems (OTM) (pp. 363–380). Rhodes, Greece.

[Lahouij, 2018] Lahouij, A., Hamel, L., Graiet, M., & Malki, M. E. (2018). "A formal approach for cloud composite services verification". In Proceedings of the 11th IEEE Conference on Service-Oriented Computing and Applications (SOCA) (pp. 161–168). Paris, France.

[Lamport, 1994] Lamport, L. (1994). "The temporal logic of actions". Journal of ACM Transactions on Programming Languages and Systems, 16(3), 872–923.

[Latreche, 2019] Latreche, F. & Belala, F. (2019). "Timed ctl checking of time critical cloud applications using timed bigraphs". International Journal of Critical Computer-Based Systems, 9(4), 379–406.

[Leuschel, 2003] Leuschel, M. & Butler, M. (2003). "Prob: A model checker for B". In Proceedings of the 12th International Symposium of Formal Methods Europe (FME) (pp. 855–874). Pisa, Italy: Springer Berlin Heidelberg.

[Leuschel, 2008] Leuschel, M. & Butler, M. (2008). "Prob: an automated analysis toolset for the b method". International Journal on Software Tools for Technology Transfer, 10(2), 185–203.

[Li, 2018] Li, Q., Zhu, H., Ying, Z., & Zhang, T. (2018). "Traceable ciphertext-policy attribute-based encryption with verifiable outsourced decryption in ehealth cloud". Journal of Wireless Communications and Mobile Computing.

[Lin, 2016] Lin, Y. & Shen, H. (2016). "Cloudfog: leveraging fog to extend cloud gaming for thin-client mmog with high quality of service". IEEE Transactions on Parallel and Distributed Systems, 28(2), 431–445.

[Mahmud, 2018] Mahmud, R., Kotagiri, R., & Buyya, R. (2018). "Fog computing: A taxonomy, survey and future directions". In Internet of everything (pp. 103–130). Springer.

[Mendes, 2018] Mendes, A. & Ferreira, J. F. (2018). "Towards verified handwritten calculational proofs". In 9th International Conference on Interactive Theorem Proving (ITP) (pp. 432–440).: Springer.

[Metsch, 2010] Metsch, T., Edmonds, A., Nyren, R., & Papaspyrou, A. (2010). "Open cloud computing interface–core". In Open Cloud Computing Interface (OCCI).

[Milner, 2001] Milner, R. (2001). "Bigraphical reactive systems". In International Conference on Concurrency Theory (pp. 16–35).: Springer.

[Milner, 2008] Milner, R. (2008). "Bigraphs and their algebra". Journal of Electronic Notes in Theoretical Computer Science, 209, 5–19.

[Mohamed, 2015] Mohamed, M., Amziani, M., Belaïd, D., Tata, S., & Melliti, T. (2015). "An autonomic approach to manage elasticity of business processes in the cloud". Journal of Future Generation Computer Systems, 50, 49–61.

[Moudjari, 2018] Moudjari, R., Sahnoun, Z., & Belala, F. (2018). "Towards a fuzzy bigraphical multi agent system for cloud of clouds elasticity management". International Journal of Approximate Reasoning, 102, 86–107.

[Ni, 2017] Ni, L., Zhang, J., Jiang, C., Yan, C., & Yu, K. (2017). "Resource allocation strategy in fog computing based on priced timed petri nets". IEEE internet of things journal, 4(5), 1216–1228.

[Ölveczky, 2017] Ölveczky, P. C. (2017). "Design and validation of cloud storage systems using formal methods". In Proceedings of the international conference on Topics in Theoretical Computer Science (TTCS) (pp. 3–8). Tehran, Iran: Springer.

[Ouchani, 2018] Ouchani, S. (2018). "Ensuring the functional correctness of iot through formal modeling and verification". In International Conference on Model and Data Engineering (pp. 401–417).: Springer.

[Perrone, 2012] Perrone, G., Debois, S., & Hildebrandt, T. T. (2012). "A model checker for bigraphs". In Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC) (pp. 1320–1325). Riva, Trento, Italy: ACM.

[Pommereau, 2008] Pommereau, F. (2008). “Quickly prototyping petri nets tools with snakes”. In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (pp.Ĩ7).: ACM.

[Puthal, 2018] Puthal, D., Obaidat, M. S., Nanda, P., Prasad, M., Mohanty, S. P., & Zomaya, A. Y. (2018). “Secure and sustainable load balancing of edge data centers in fog computing”. IEEE Communications Magazine, 56(5), 60–65.

[Ren, 2015] Ren, Z., Wang, L., Wang, Q., & Xu, M. (2015). “Dynamic proofs of retrievability for coded cloud storage systems”. Journal of IEEE Transaction Services Computing, 11(4), 685–698.

[Sahli, 2017] Sahli, H., Hameurlain, N., & Belala, F. (2017). “A bigraphical model for specifying cloud-based elastic systems and their behaviour”. International Journal of Parallel, Emergent and Distributed Systems (IJPEDS), 32(6), 593–616.

[Sahli, 2019] Sahli, H., Ledoux, T., & Rutten, É. (2019). “Modeling Self-Adaptive Fog Systems Using Bigraphs”. In 17th International Workshop on coordination and Self-Adaptativeness of Sotware applications (FOCLASA) (pp. 1–16). Oslo, Norway.

[Sevegnani, 2016] Sevegnani, M. & Calder, M. (2016). “Bigrapher: rewriting and analysis engine for bigraphs”. In International Conference on Computer Aided Verification (pp. 494–501).: Springer.

[Shah-Mansouri, 2018] Shah-Mansouri, H. & Wong, V. W. (2018). “Hierarchical fog-cloud computing for iot systems: A computation offloading game”. IEEE Internet of Things Journal, 5(4), 3246–3257.

[Souri, 2018] Souri, A., Navimipour, N. J., & Rahmani, A. M. (2018). “Formal verification approaches and standards in the cloud computing: A comprehensive and systematic review”. Computer Standards & Interfaces, 58, 1–22.

[Spivey, 1992] Spivey, J. M. & Abrial, J.-R. (1992). “The Z notation”. Prentice Hall Hemel Hempstead.

[Sultan, 2014] Sultan, N. (2014). “Making use of cloud computing for healthcare provision: Opportunities and challenges”. International Journal of Information Management, 34(2), 177 – 184.

[Toor, 2019] Toor, A., ul Islam, S., Sohail, N., Akhunzada, A., Boudjadar, J., Khattak, H. A., Din, I. U., & Rodrigues, J. J. (2019). “Energy and performance aware fog computing: A case of dvfs and green renewable energy”. Future Generation Computer Systems, 101, 1112–1121.

[Uriarte, 2018] Uriarte, R. B. & De Nicola, R. (2018). “Blockchain-based decentralized cloud/fog solutions: challenges, opportunities, and standards”. IEEE Communications Standards Magazine, 2(3), 22–28.

[Varia, 2014] “Varia, J., Mathew, S., (2014). Overview of amazon web services”. Journal of Amazon Web Services, (pp. 1–22).

[Wang, 2009] Wang, Q., Wang, C., Li, J., Ren, K., & Lou, W. (2009). “Enabling public verifiability and data dynamics for storage security in cloud computing”. In Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS) (pp. 355–370). Saint-Malo, France: Springer.

[Wang, 2011] Wang, Q., Wang, C., Ren, K., Lou, W., & Li, J. (2011). “Enabling public auditability and data dynamics for storage security in cloud computing”. Journal of IEEE transactions on parallel and distributed systems, 22(5), 847–859.

[Xiong, 2018] Xiong, Z., Feng, S., Wang, W., Niyato, D., Wang, P., & Han, Z. (2018). “Cloud/fog computing resource management and pricing for blockchain networks”. IEEE Internet of Things Journal, 6(3), 4585–4600.

[Yataghene, 2016] Yataghene, L., Ioualalen, M., Amziani, M., & Tata, S. (2016). "Using formal model for evaluation of business processes elasticity in the cloud". In Proceedings of the 14th International Conference on Service-Oriented Computing (ICSOC) (pp. 33–44). Banff, AB, Canada: Springer.

[Zahra, 2017] Zahra, S., Alam, M., Javaid, Q., Wahid, A., Javaid, N., Malik, S. U. R., & Khan, M. K. (2017). "Fog computing over iot: A secure deployment and formal verification". IEEE Access, 5, 27132–27144.

[Zeng, 2016] Zeng, W., Koutny, M., Watson, P., & Germanos, V. (2016). "Formal verification of secure information flow in cloud computing". Journal of Information Security Applications, 27-28, 103–116.

[Zhang, 2017] Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, F. R., & Han, Z. (2017). "Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching". IEEE Internet of Things Journal, 4(5), 1204–1215.

[Zhang, 2018] Zhang, P., Zhou, M., & Fortino, G. (2018). "Security and trust issues in fog computing: A survey". Future Generation Computer Systems, 88, 16–27.

[Zhou, 2015] Zhou, J., Dong, X., Cao, Z., & Vasilakos, A. V. (2015). "Secure and privacy preserving protocol for cloud-based vehicular dtns". Journal of IEEE Transactions on Information Forensics and Security, 10(6), 1299–1314.

[Zitouni, 2019] Zitouni, R., Petit, J., Djoudi, A., & George, L. (2019). "Iot-based urban traffic-light control: Modelling, prototyping and evaluation of mqtt protocol". In 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 182–189).: IEEE.