Noname manuscript No.
(will be inserted by the editor)

A Survey on Reliability and Availability Modeling of Edge, Fog, and Cloud Computing

Paulo Maciel · Jamilson Dantas · Carlos Melo · Paulo Pereira · Felipe Oliveira · Jean Araujo · Rubens Matos

Received: date / Accepted: date

Abstract During the past years, sending data to the cloud servers was a prominent trend, making the cloud computing paradigm dominate the technology landscape. However, the Internet of Things (IoT) is becoming part of our daily environments, and it generates a large volume of data, which is creating uncontrollable delays. For the delay-sensitive and context-aware services, these uncontrollable delays may cause low reliability and availability for applications. To overcome these challenges, computing paradigms are moving from centralized cloud environments to the Edge of the networks. Several new computing paradigms, such as Edge and Fog computing, emerged to support delay-sensitive and context-aware services. By combining edge devices, fog servers, and cloud computing, companies can build a hierarchical IoT infrastructure, using Edge-Fog-Cloud orchestrated architecture to improve IoT environments' performance, reliability, and availability. This paper presents a comprehensive survey on reliability and availability of Edge, Fog, and Cloud computing architectures. We first introduce and compare some related works about these paradigms and compare them to define the differences between Edge and Fog environments, since there is still some confusion about these terms. We also describe their taxonomy and how they link to each other. Finally, we draw some potential research directions that may help foster research efforts in this area.

Keywords Reliability · Availability · Cloud Computing · Edge Computing · Fog Computing · Modeling

Paulo Maciel, Jamilson Dantas, Carlos Melo, Paulo Pereira, Felipe Oliveira
Centro de Informática - Universidade Federal de Pernambuco, Brazil
www.cin.ufpe.br
www.modcs.org
E-mail: {prmm, jrd, casm, prps, fdo}@cin.ufpe.br

Jean Araujo
Universidade Federal do Agreste de Pernambuco, Brazil
E-mail: jean.teixeira@ufape.edu.br

Rubens Matos
Instituto Federal de Sergipe, Brazil
E-mail: rubens.junior@ifs.edu.br

1 Introduction

Over the centuries, mankind developed tools to automate and improve the most diverse tasks that were once considered tedious or too time-consuming. Such tools were disposable, and when they fulfilled their purpose, they would be devoided of their functions. As their costs increased, studies aimed to estimate such equipment's lifetime or at least to point out to when it would stop working became necessary, giving rise to reliability studies [76].

Despite failures estimation, the reduction of financial costs about that tool or equipment became possible only with the maintenance process, mainly those of corrective type, after all the logic inherent to the time was that we do not need to repair something that is not failed, so why should we have to have the luxury of performing preventive maintenance? Availability [27] studies originated from this point; later, the term dependability [37] was coined and managed to encompass several concepts that until then were considered disconnected or were treated individually as independent attributes inherent to systems, which are availability, reliability, maintainability, security, integrity, and confidentiality.

The study of dependability [6, 37] attributes has become of greater importance to the most diverse systems and services that emerged after the advent of the Internet. Those services provided by third parties, such as virtualization environments and the ever-emerging Cloud Computing [31]. Such services are subject to a series of threats and meet the most diverse service provision requirements. Service Level Agreement (SLA) [65], in cloud computing, it is signed between the provider and the contractor and tries to guarantee minimum acceptable levels for dependability attributes. However, the need for continuous improvements and recurring technological advances and charges for immediate responses by users of cloud-hosted services raised another issue: the distance between the access point and the data center hosting the service, which can be continents away from the user.

From this conjuncture emerged both Fog [8], and Edge [72] computing, which seeks to bring the user closer to the service offered without disfiguring advances obtained in the study of dependability attributes. This survey tries to show by introducing the main studies and concepts related to two of the main dependability attributes: availability and reliability of the cloud, fog, and edge computing environments. As the main contributions of this survey, we can list:

– Differentiation between Edge and Fog through an in-depth taxonomic analysis of both terms, which remain intertwined;
– Comparison between the main state-of-the-art works related to the assessment of availability and reliability of the Cloud, Edge, and Fog computing that made use of models, whether analytical or simulation;
– Challenges that still open and research opportunities for beginners and researchers interested in applying new efforts to the evaluated areas.

The remains of this survey are organized as follows. Section 2 presents the basic components about modeling and evaluation methods. Later in Section 3, we present an overview of the Cloud, Edge, and Fog computing, by showing a comparison between them. Already Section 4 presents the state-of-the-art and comparison between the evaluated papers. Section 5 shows how this survey was performed and how the obtained results can be replicated. Some directives and current challenges for fog, edge, and cloud researchers are presented in Section 6. Finally, Section 7 presents the conclusions and final remarks.

2 Background

This section presents basic concepts about modeling and evaluation methods, including the mathematical models characterized as either state-based or non-state-based. These models aim to facilitate understanding Edge's typical evaluation, fog, or cloud environments, using dependability attributes such as reliability and availability.

2.1 Dependability Evaluation for Cloud, Edge, and Fog Computing

System dependability is usually defined as delivering a specified functionality that can be justifiably trusted [7]. Dependability measures deserve great attention to determine the quality of the service provided by a system. Dependability studies address reliability, availability, and other metrics [49].

The most basic dependability aspects of a system are the failure and repair events, which may bring the system to different configurations and operational modes. The steady-state availability is a common measure extracted from dependability models. Reliability, downtime, uptime, and mean time to system failure are other metrics usually aimed as output from a dependability analysis in computer systems. In other words, the system reliability at t is the probability that the system performs its functions without failure up to time instant t. The steady-state availability can be expressed as the ratio of the expected system uptime to the expected system up and downtimes, as seen below:

$$A = \frac{E[Uptime]}{E[Downtime] + E[Uptime]} \tag{1}$$

It may also be represented by:

$$A = \frac{MTTF}{MTTF + MTTR}, \tag{2}$$

where MTTF is the mean time to failure and MTTR is the mean time to recovery.

In many circumstances, modeling is the method of choice due to the system's complexity or because it may not exist yet. There are formal techniques that may be used for modeling computer systems and estimating measures related to system availability and reliability. The state-based and non-state-based models may capture the system behavior and allow the description and prediction of dependability metrics.

2.2 Formalisms

These formalisms may be broadly classified into combinatorial and state-space models. State-space models may also be mentioned as non-combinatorial, and combinatorial can be identified as non-state space models [47].

Combinatorial models describe conditions that make a system failure (or functional) concerning structural relations between the system elements. These relations comply with the system's set of components that should be either properly operating or faulty. State-space models describe the system operation (failures and repair activities) by states and events denoted as labeled state transitions. These formalisms embody more complex relations between the system parts, such as dependencies concerning subsystems and device restrictions, and intricate maintenance procedures. Some state-space models may also be assessed by discrete event simulation in unmanageable huge state spaces or when an aggregation of non-exponential distributions prevents an analytic (closed-form) or numerical solution.

The most noticeable combinatorial model types are Reliability Block Diagrams (RBD) and Fault Trees (FT). Markov Chains, Stochastic Petri nets, and Stochastic Processes algebras are the most widely adopted state-space models.

– State-Based Models

Markov models represent the interactions between various system components, for both descriptive and predictive purposes [58]. Markov process has been intensively adopted in performance and dependability modeling since around the fifties [47]. Manufacturing, logistics, communication, computer systems are some examples of fields that can rely on stochastic modeling as an interesting approach to address various problems.

A stochastic process is defined as a set of random variables ($\{X_i(t) \ : \ t \in T\}$) indexed through some parameter (t). Each random variable ($X_i(t)$) is defined in a probability space. The parameter t usually represents time, so $X_i(t)$ denotes the value assumed by the random variable at time t. T is called the parameter space and is a subset of $\mathbb{R}$.

Petri Nets [61] is a family of formalisms very well suited for modeling several system types due to their capability for representing concurrency, synchronization, communication mechanisms, as well as deterministic and probabilistic delays. The first stochastic Petri net extensions were proposed independently by Symons, Natkin and Molloy [23, 51, 59, 60, 63, 78]. These models formed what was then named Stochastic Petri nets (SPN).

Let SPN=(P,T,I,O,H,M_0,Atts) be a Stochastic Petri Net (SPN), where P,T,I,O, and M_0 are defined as for Place-Transition nets, that is, P is the set of places, T is the set of transitions, I in input matrix, O is the output matrix, and M_0 is the initial marking. The set of transition, T, is, however, divided into immediate transitions (T_{im}), timed exponentially distributed transitions (T_{exp}), deterministic timed transitions (T_{det}), and timed generically distributed transitions (T_g):

$$T = T_{im} \cup T_{exp} \cup T_{det} \cup T_g. \tag{3}$$

– Non-State-Based Models

Reliability Block Diagrams (RBD) and Fault Trees (FT) are non-state-based models and the most widely adopted reliability evaluation models. RBD is probably the oldest combinatorial technique for reliability analysis. Fault Tree Analysis (FTA) was initially developed in 1962 at Bell Laboratories by H. A. Watson to analyze the Minuteman I Intercontinental Ballistic Missile Launch Control System. Afterward, in 1962, Boeing and AVCO expanded the use of FTA to the entire Minuteman II [19]. In 1965, W. H. Pierce unified Shannon, Von Neumann, and Moore theories of masking and redundancy as the concept of failure tolerance ([68]. In 1967, A. Avizienis combined masking methods with error detection, fault diagnosis, and recovery into the concept of fault-tolerant systems [6].

In reliability block diagrams, the components are represented by combined blocks in serial or parallel. The representation of serial blocks requires that each component is working for the system to be operational. A diagram with connected parallel components requires that only one block is working for the system to be operational [80]. Therefore, the system may be described by a set of interconnected functional blocks able to represent the availability and reliability of the system.

The transient availability and reliability of serial blocks may be obtained through Equation 4 and Equation 5.

$$P_s(t) = \prod_{i=1}^{n} P_i(t), \tag{4}$$

and

$$A_s = \prod_{i=1}^{n} A_i, \tag{5}$$

where, $P_i(t)$ may denote the reliability $R_i(t)$, the instantaneous availability $A_i(t)$, and A_i the steady-state of the block B_i.

The reliability and availability of connected parallel blocks may be obtained through Equation 6 and Equation 7.

$$P_p(t) = 1 - \prod_{i=1}^{n} (1 - P_i(t)), \tag{6}$$

and

$$A_p = 1 - \prod_{i=1}^{n} (1 - A_i). \tag{7}$$

The RBD models are utilized in modular systems with many independent modules, and a block may easily represent each one. More complex structures such as *K-out-of-N* constructions may also be represented [47].

3 Edge, Fog, and Cloud Computing at a Glance

This section focuses on comparing Edge, Fog, and Cloud computing, and related computing models to show the importance of these computing paradigms in various scenarios. Furthermore, this section presents a better perception of how these computing paradigms may aggregate advantages to connected devices' current and future landscape. We start our discussion with the cloud paradigm; then, we discuss fog infrastructure and edge environments. To finish, we point out the main difference between them.

3.1 Cloud Computing

Cloud computing is a model that promotes ubiquitous, on-demand network access to shared computing resources [54]. This paradigm was a prominent trend over the past years because it enables companies to count on external providers to save and process their data. Cloud computing supports an agile service-based technology market by granting easy access to computing resources; it also facilitates rapidly deploy new services with full access to the World-Wide-Web [66].

– Types of Cloud Services

The Cloud computing paradigm offers three types of services: infrastructure, platform, and software (IaaS, PaaS, SaaS). Companies and application developers may choose and use several varieties of these services, depending on their necessity. Infrastructure as a service (IaaS) permits customers to use hardware related services using cloud computing principles. These related services clouds include storage (e.g., database or disk storage), processing power, virtual servers, networking resources, and so on [54]. For instance, suppose that someone wants to set up some security system in a building using facial recognition, so this person contacts some cloud provider and acquires an infrastructure as a service. Therefore, this person can configure the hardware in terms of quantity of CPUs cores, amount of RAM capacity, and so forth [87].

Platform as a service (PaaS) is related to offering an environment to develop software and support the software lifecycle. In this kind of service, the customer does not choose the hardware configurations and operating systems; the whole developing environment is already set up [2]. On the other hand, software as a service (SaaS) involves a complete software package on the cloud, where the customer does not need to worry about database scalability, software errors and socket management, for example. In this type of service, the customer does not need to install anything and pays per use [87].

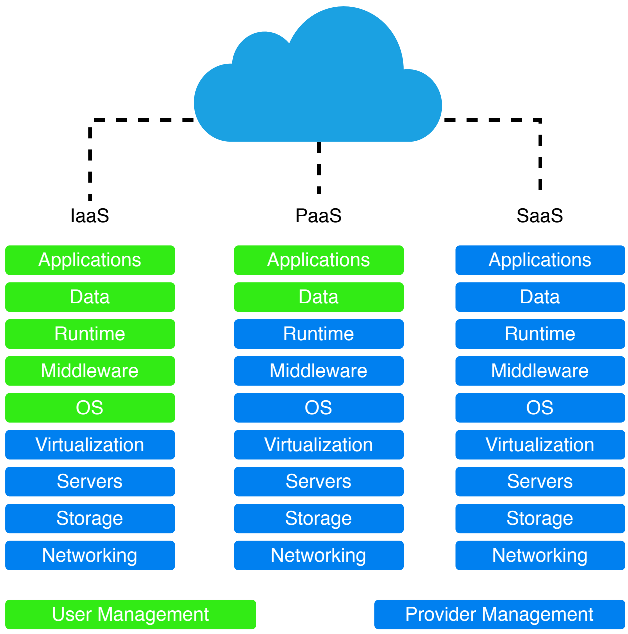


Fig. 1: Cloud service models.

3.2 Fog Computing

Fog computing is a term proposed by researchers from Cisco System [8], and it is related to the processing power at the Edge of the network. There is also a similar concept called cloudlets. However, cloudlets are related to the mobile network, but fog computing is commonly related to IoT infrastructure [62]. This paradigm can be composed of virtualized and non-virtualized resources, and it provides storage, processing power, and networking services for end-devices and cloud servers [15].

The main focus of Fog computing aims at supporting low-latency services [40], but it also supports non-latency services; bringing computational power closer to the end devices improves the overall performance of the applications, and it decreases the networking bandwidth consumption. There are a vast number of heterogeneous nodes (e.g., sensors, actuators, and devices) that are connected to the fog environments, and they generate a high volume of data to be processed, which may be more than the capacity of the fog environment;

therefore, the fog sends these data to the cloud [8]. Fog computing also supports computing resources, cloud integration, communication protocols, mobility, interface heterogeneity, and distributed data analytics [81].

It is worth point out that any device that has processing power, networking, and storage capabilities may act as a fog device [62]. These devices can act as fog servers, and fog gateways, where fog servers are responsible for managing several fog devices, and fog gateways manage and translate the data from heterogeneous devices to the fog servers [62].

The main advantages associated with Fog computing environments are:

- **reduction of network traffic** - fog environments drastically reduces the traffic being sent to the cloud because most of the data is processed closer to the end devices.
- **suitable for IoT tasks and queries** - by using fog environments, IoT tasks may be processed closer to the physical location of the sensors/actuators, which makes the application closer to the geographical context of the sensor or actuator.
- **low latency requirement** - the data processing is performed closer to the nodes; thus, it makes real-time response possible.

Nowadays, we have more than 50 billion devices worldwide connected [62]. This was the estimation that Cisco published two years ago. Hence, fog computing environments become a powerful alternative to process the massive volume of data generated per second, since such environments work as a filter for the cloud servers.

Moreover, fog computing is sometimes harshly recognized as an aggregation of Mobile Cloud Computing (MCC) and Mobile Edge Computing (MEC). That occurs because of the computing and data storing capacities, where the end devices are moved to non-virtualized or virtualized fog servers; therefore, the end devices can obtain the highest processing efficiency and latency reduction in real-time services of end devices [64].

To be adopted by the market, fog computing must be standardized. There is currently no available standard architecture [62]; nevertheless, there are many efforts to propose well-defined fog computing architectures, as discussed in the following sections.

3.3 Edge Computing

Edge devices or servers provide processing, networking, and storage facilities in edge computing environments [62]. Edge computing is related to the technologies that allows computation to be performed at the Edge of the network [75]. Edge computing paradigm processes the data locally at the edge nodes; then the selected data or insights flow to the cloud server. This is the opposite of the traditional cloud paradigm, which allows all the raw data generated from the edge devices to flow to the cloud server [35].

Usually, edge computing is not spontaneously associated with any types of cloud-based services (i.e., IaaS, PaaS, SaaS); it concentrates on the end device side [62]. Any smart device that can temporarily store and process data can be considered an edge node [35]. For instance, if a smartphone receives and processes data from a sensor, then the smartphone is the application's edge node. The main goal of edge computing is that the computation should be done as close as possible to the data sources. This paradigm also overcomes many issues such as privacy, latency, and connectivity; it occurs because latency in edge computing is typically lower than cloud computing due to its localization. It also has a higher service availability because connected devices do not wait for a highly centralized platform to provide a service, and also there are typically many physical nodes [87].

The devices in the edge paradigm not only are data consumers but also are data producers. The devices may demand services from the Edge servers, but they also need to perform the computing tasks. Edge devices may perform computing, data storage, offloading, caching, and processing; they may also process requests and deliver service to other end devices. Therefore, the Edge itself needs to be well-designed to cope with the reliability, security, and privacy concerns of the users [67, 75].

By using edge computing environments, we might achieve several benefits compared to the traditional cloud computing paradigm. For instance, [85] proved that they were able to decrease from 900 to 169 ms the response time of a facial recognition application. Already, [29] proved that by using edge computing environments to offload computing tasks for wearable cognitive assistance, they improved response time to 80 ms; they also proved that the energy consumption was reduced by 40%.

Therefore, this paradigm may be assumed as the method of data retrieval, processing, and analysis closer to the data source, using devices like IoT gateways or network switches or even some other custom-built device with sufficient computing power (e.g., RaspberryPi). This paradigm allows analyzing in real-time and on-site the data collected from various sensors and IoT devices.

3.4 Edge vs Fog vs Cloud Computing

Edge and Fog computing are frequently related to the same architecture; nevertheless, it is essential to distinguish them. This subsection aims to highlight the differences between these two architectures and compare them with the cloud paradigm.

Although Edge computing and Fog computing bring the computation and storage to the Edge of the network and closer to the end devices, these paradigms are not identical. According to the OpenFog Consortium, Edge computing is often erroneously called Fog computing [11]. Fog computing is a hierarchical infrastructure that provides computing, networking, storage, and control anywhere from the cloud to the end devices [87]. On the other hand, Edge computing is usually limited in processing power at the network's Edge. According to [10], Fog computing tries to create a consistent infrastructure of computing services from the cloud to the end devices rather than assuming that the network edges are separated computing architectures. In other words, Fog computing brings intelligence down to the local area network (LAN) level of the network infrastructure, where the data or services are processed in fog nodes or IoT gateways. On the other hand, Edge computing delivers the intelligence, processing, and communication capabilities of edge gateways straight into smart devices such as programmable automation controllers (PAC) [48].

The fog layer is inserted between the Edge and the cloud layer; fog is usually geographically distant from the businesses operating premises [35]. This paradigm generally relates to network infrastructure maintained by a service provider, which leases its infrastructure to the companies; in the fog layer, fog nodes are the processing and communication elements. Based on the use case's nature, the service provider decides which data will be performed at the Edge and uploaded to the fog nodes [35].

According to [48], fog environments include the cloud; on the other hand, edge environments are defined by excluding the cloud. The Fog computing paradigm has a hierarchical and horizontal structure, but it also has several layers forming a network, whereas the edge paradigm is usually restricted to divide nodes into the ones that are not part of the network and the ones that are. The fog environments have extensive peer-to-peer interconnect capability between nodes, where Edge executes its nodes in isolation from others, needing data transport back through the cloud for a peer-to-peer behavior [30, 48].

Fog and Edge computing are a distributed computing paradigm that extends Cloud computing to the Edge of the network; they serve as a complement to the cloud solution, focusing on the emerging IoT architecture [35]. These two new paradigms are responsible for facilitating storage, networking, and computing services between end devices and cloud data centers. Fog computing architecture usually includes segments of an application running both in the cloud and in the fog nodes (i.e., in smart gateways, routers, or dedicated fog devices) [48].

In summary, when we opt for using fog and edge environments, the amount of bandwidth used is reduced considerably; that occurs because clouds demand much more bandwidth than fog and edge paradigms. It is worth highlighting that the Internet is inherently unreliable, and wireless networks have limitations, so reducing the bandwidth consumption is an appealing benefit [87]. These paradigms also allow transmitted data to avoid the Internet, maintaining it as local as possible on smart devices. It is also worth stressing that many data may still be transmitted to the cloud, and the sensitive data could be kept local, letting more bandwidth for the ones using the cloud.

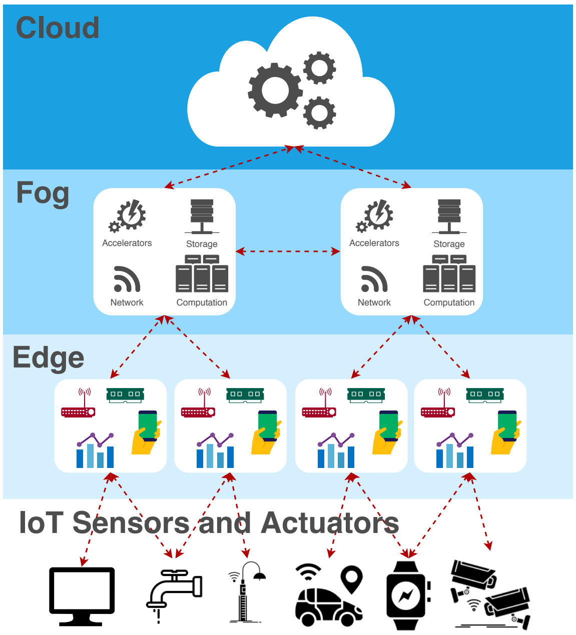


Fig. 2: Edge-fog-cloud stack.

Fog and Edge computing have more closeness to end-users and more geographical distribution. Compared to the Cloud paradigm, Fog and Edge environments highlight the closeness to end devices and client goals, local resource pooling, latency, and the backbone bandwidth. Consequently, it achieves a better quality of service (QoS), resulting in superior user experience [48].

Figure 2 depicts the edge-fog-cloud computing stack and how these paradigms relate to each other.

4 Review on State of the Art

The interest in Edge/Fog/Cloud computing to support IoT infrastructures has been snowballing in academia and industry. In academia, the concept of edge-fog-cloud interoperability in IoT infrastructures for the assurance of various QoS measures (e.g., availability and reliability) has been proposed in several recent works. This section will present a review of state-of-the-art reliability and availability studies that used modeling to evaluate Edge, Fog, and Cloud computing environments. Table 1 shows a summary with the evaluated papers and infrastructure type, evaluation method, and the formalism used.

4.1 Availability

This section presents some important papers that study availability analysis on Edge, Fog, and Cloud computing.

– Edge Computing

Facchinetti et al. [21] presented their expertise with Mobile Cloud Computing (MCC) to develop a lightly-pervasive well-being service, which they called IPSOS Assistant; this service is to manage emergencies in indoor spaces. Their proposal combines indoor positioning, context monitoring, visualization, and social collaboration to assist users in indoor workplaces in case of personal and environmental emergencies. The authors also address the availability and reliability of their proposal.

Jia et al. [36] introduced an edge computing environment that supports resource sharing according to several scenarios. The authors also proposed a trust property, where they subdivided the system based on the heterogeneity and dynamics for increasing availability of the services hosted.

Boukerche and Soto [9] introduced a data retrieval approach for computation offloading in Vehicular Edge Computing (VEC). Their strategy begins with the premise that when the vehicle finishes the computation, it will send the processed data to the sender Road-Side Units (RSU). The authors also state that depending on the speed and the processing time of the task, the vehicle's location may be different from when the task was received; therefore, they proposed a model of the computation offloading system, which analyzes the capacity to simulate the components that enhance its performance and availability in a vehicular Edge computing environment.

Angin et al. [4] proposed a mobile-cloud methodology based on autonomous agent-based application modules. Their main goals are to introduce an operative and easy-to-adopt MCC model. They also introduced a dynamic availability-performance estimation model that permits the tracking and relocation of offloaded computation; this dynamic model tries to achieve optimal availability and performance under varying mobile-cloud contexts.

d'Oro et al. [18] proposed a modeling strategy to measure the performance issues considering Cloud and Edge Computing relying on queuing networks. The authors take into account the coordination of fire brigade teams equipped with sensors and augmented reality devices. Although the authors use a modeling strategy, the authors do not contemplate the availability measures and Fog issues. This work supplementary the missing questions.

Gamatié et al. [22] introduced an original asymmetric multicore infrastructure connected to a programming model and workload control to increase the availability and performance of edge environments. This infrastructure combines ultra-low power cores dedicated to parallel workloads for high throughput and a high-performance core. According to the authors, even though their design resembles a CPU-GPU heterogeneous combination, their proposal is more flexible because the GPU is only used for regular parallel workloads.

Gorbenko et al. [26] introduced analytical models and practical guidance to developers of distributed fault-tolerant systems allowing them to predict the availability of IoT environments. Their work states that the proposed models will help developers analyze the trade-off between consistency, availability, and latency during system design and operation. Although the authors provided an availability model, they did not consider the performance aspects of such environments.

– Fog computing

Zhou et al. [88] proposes a concept in DDoS mitigation in Fog Computing using allocating traffic monitoring and analysis in Industrial Internet of Thing (IIoT). The authors proposed a schema based on real-time traffic filtering via field firewall devices. The proposed schema is tested in an industrial control system testbed, and the experiments evaluate the detection time and rate considering DDoS attacks. The authors concentrate on security issues in their study and measurements. The main difference between our survey and their work is that this study takes into account the modeling strategies such as RBD, CTMC, and SPN technics. This

Table 1: Overview of availability and reliability works in Cloud, Fog, and Edge computing.

	Publication	Infra. Type	Evaluation Method	Formalism
Availability	Facchinetti et al. [21]	Edge	Analytical Models and Simulation	-
	Jia et al. [36]	Edge	Analytical Models	Markov chain
	Boukerche and Soto [9]	Cloud and Edge	Analytical Models	Markov chain
	Angin et al. [4]	Fog and Edge	Analytical Models	Markov chain
	d'Oro et al. [18]	Cloud and Edge	Analytical Models	Queuing Networks
	Gamatié et al. [22]	Edge	Analytical Models	Markov chain
	Gorbenko et al. [26]	Cloud	Analytical Models and Measurement	Failure model
	Zhou et al. [88]	Fog	Measurement	-
	Yakubu et al. [84]	Fog	Analytical Models	-
	Sanyal and Zhang [71]	Fog and Edge	Analytical Models	Markov chain
	Sharma et al. [74]	Cloud and Fog	Analytical Models	Markov chain
	Sharkh and Kalil [73]	Cloud and Fog	Analytical Models	Markov chain
	Guan et al. [28]	Fog and Edge	Analytical Models	Markov chain
	Andrade et al. [3]	Cloud and Fog	Analytical Models	DSPN
	Ever et al. [20]	Edge	Closed-form equation	Markov chain
	Longo et al. [45]	Cloud	Analytical Models and Simulation	Petri Net, Markov chain
	Ghosh et al. [24]	Cloud	Analytical Models and Simulation	Petri Net, Markov chain
	Ghosh et al. [25]	Cloud	Analytical Models and Simulation	Petri Net, Markov chain
	Machida et al. [46]	Cloud	Analytical Models and Simulation	Petri Net
	Matos et al. [52]	Cloud	Analytical Models	Markov chain
	Matos et al. [53]	Cloud	Analytical Models	Petri nets, RBD, Markov chain
	Melo et al. [55]	Cloud	Analytical Models	DRBD
	Melo et al. [57]	Cloud	Analytical Models	DRBD
	Melo et al. [56]	Cloud	Analytical Models	Petri Net, RBD
	Liu et al. [43]	Cloud	Analytical Models and Simulation	Stochastic Reward Net
	Di Mauro et al. [17]	Cloud	Analytical Models	Petri Net, RBD
	Dantas et al. [12]	Cloud	Analytical Models	RBD, Markov chain
	Dantas et al. [13]	Cloud	Analytical Models	Equation, RBD, Markov chain
	Dantas et al. [14]	Cloud	Analytical Models	Equation, RBD, Markov chain
	Qiu et al. [69]	Cloud	Analytical Models	Markov chain
	Li et al. [39]	Cloud and Edge	Analytical Models and Measurement	Gray-Markov chain
	Ataie et al. [5]	Cloud	Analytical Models and Simulation	Petri nets
	Mao et al. [50]	Cloud	Analytical Models and Simulation	Markov chain
	Wang et al. [83]	Fog	Simulation	-
	Liu et al. [44]	Cloud	Analytical Models and Simulation	Petri Net
	Addis et al. [1]	Cloud	Artificial Intelligence	-
	Jammal et al. [34]	Cloud	Analytical Models	Markov chain
	Santos et al. [70]	Cloud, Fog and Edge	Analytical Models	Petri Net, RBD
	Wang et al. [82]	Cloud	Analytical Models	Probability dist. models
Reliability	Sun and Liu [77]	Edge	Analytical Models	Markov chain
	Huang et al. [33]	Cloud and Edge	Analytical Models and Simulation	Petri Net, RBD
	Li and Huang [41]	Fog and Edge	Analytical Models	Petri Net
	Zilic et al. [89]	Edge	Analytical Models and Simulation	Markov chain
	Liang et al. [42]	Cloud and Edge	Combinatorial Optimization	-
	Lee et al. [38]	Cloud	Analytical Models and Simulation	Reliability Expression
	Tian et al. [79]	Cloud	Analytical Models and Simulation	Reliability Expression
	Dehnavi et al. [16]	Cloud, Fog and Edge	Analytical Models and Simulation	Reliability Expression
	Huang et al. [32]	Cloud and Edge	Modeling	Graph
	Yousefpour et al. [86]	Cloud, Fog, and Edge	Artificial Intelligence	-

study also aggregates the availability issues and Edge and Cloud Computing paradigms.

Yakubu et al. [84] introduced a systematic review of security challenges in Fog computing. The authors discussed several procedures utilized to solve security dilemmas within the fog-computing environment. In their study, the authors show that most of the security methods used by different researchers are not dynamic sufficient to overcome all the fog security issues, including availability and reliability problems. The main difference between our survey and their work is that this study is more general in the sense of showing several types of research regarding availability and reliability. This study also aggregates the Edge and Cloud computing paradigms, and not only the Fog computing.

Sanyal and Zhang [71] introduced a data analytics framework and data aggregation structure to enhance the efficiency of decision making regarding the availability of Fog environments. The authors also showed an uncertainty model for IoT sensor data based on Shannon's entropy. According to the authors, the data aggregation and preprocessing scheme proposed may adequately eliminate the IoT data uncertainties while maintaining the data dynamics. They also stated that data restoration from the partial sets of raw data before the subspace tracking significantly benefits similar methods. The approach presented does not demand any previous knowledge about the outliers or the missing values and may work on a fully and partially collected dataset.

Sharma et al. [74] introduced an efficient and flexible architecture that uses SDN, Fog computing, and a blockchain paradigm to capture and analyze IoT income data of end devices. Although their proposal focuses on performance evaluation, it also considerably increased the provided services' availability and reliability. Their distributed Fog environment used SDN and blockchain paradigm to bring computing resources to the edge devices, so the traffic has a minimal end-to-end delay between IoT sensors and actuators and computing power.

Sharkh and Kalil [73] introduced a novel optimization model to investigate and minimize the total Cloud cost to serve requests to increase the service's availability and reliability. According to the authors, in cases where IoT service providers give more importance to the cost, and the providers are limited by network capacity, they can use Fog environments to serve all requests that arrive in the service. Their results show that in scenarios where the requests have a high percentage of high-value requests, the minimal cost may be achieved by giving more importance to the Cloud nodes; in other words, the request must be sent to the cloud nodes. They also state that in cases where a high percentage of requests have high value, it is preferable to change some of the Fog Node work.

Guan et al. [28] proposed a layered platform-centric model from the cloud's perspective to provide a full picture of handling the offloading issue in the multi-user multicore mobile cloud environment. According to the authors, a platform-centric offloading scheme was formulated and evaluated in this model, aiming to improve task execution efficiency and energy reduction during offloading and increase the service's availability.

Andrade et al. [3] proposed a Deterministic and Stochastic Petri Net (DSPN) approach for evaluating Fog–Cloud IoT environments composed of hundreds of physical Things. The author's approach evaluates the trade-offs of many performability metrics such as utilization, response time, throughput, and availability and may help system designers choose the most suitable Fog–Cloud IoT environment. The results show that adopting a Fog node would improve availability and performance in some instances.

Ever et al. [20] introduced a stochastic performance and an energy availability evaluation model for WSNs, where they consider node and link failures. Their proposal contemplates an integrated performance and availability approach. In their study, several duty cycling schemes and the medium-access control of the WSNs were also examined in order to include the consequence of sleeping and idle states. Their results reveal the consequences of failures, several queue capacities, and system scalability.

– Cloud computing

Longo et al. [45] introduced an approach for availability analysis of IaaS cloud systems, which can generate a quick and scalable response and considers multiple classes of server pools. Their approach is based on modeling the system as a joined interacting Markov chain using sub-models. As a result, the authors elaborated closed-form equations for the sub-models; and regarding the dependencies among the sub-models, the authors solved it by using fixed-point iteration. According to the authors, their proposal decreases the complexity and solution time for analyzing IaaS clouds (e.g., IaaS Cloud systems' availability composed of thousands of physical servers may be analyzed in seconds).

Ghosh et al. [24] introduced a stochastic model-driven mechanism to optimize the cost and capacity in an IaaS cloud infrastructure. Their approach is an optimization framework built throughout the performance and availability models of an IaaS Cloud because it helps the computation of different cost components. The authors also present simulated stochastic algorithms to solve the non-linearity and non-convexity of the overall optimization. According to their results, their proposal is quicker than a naive search algorithm while maintaining the solution's optimality.

Ghosh et al. [25] proposed a stochastic modeling mechanism to evaluate large IaaS cloud systems' availability. Their work demonstrates a way to overcome scalability problems for a monolithic model through interacting sub-models or simulation. Using the sub-model interactions, the authors produced fast model solutions facilitating scalability without jeopardizing the accuracy. Through simulation, the authors were able to generate results that closely match the monolithic and interacting sub-models technique. The authors also derived closed-form solutions to solve large cloud models faster.

Machida et al. [46] proposed a component-based availability modeling framework called Candy. Accordinging to the authors, this framework composes a comprehensive availability technique to model cloud services from the perspective of system specifications described using SysML diagrams. The authors also state that their proposal semi-automatically converts the elements of SysML diagrams into model components. To validate their framework, the authors applied their framework in a web application system on an IaaS cloud, and the generated model is used to evaluate the effectiveness of the automatic scale-up mechanism and failure-isolation zones.

Matos et al. [52] introduced a sensitivity analysis of mobile cloud availability based on hierarchical analytical models. Their results improved the system availability considerably by focusing on a decreased set of components that create a considerable variation on steady-state availability. Their sensitivity analysis ranked the system's components. At the highest positions of sensitivity rankings, it is the battery discharge or the cloud infrastructure components named Infrastructure and Storage Managers; it means that these components must obtain the highest priority to produce improvements in system availability.

Matos et al. [53] introduced a sensitivity analysis of hierarchical heterogeneous models for Eucalyptus private cloud architectures, which may be used for guiding further improvements to the system availability. They also present closed-form equations that enable the solution of desired metrics and computation of sensitivity indices without the necessity for the RBD and MRM models' numerical solution.

In Melo et al. [55], the authors evaluated both availability and reliability issues regarding service provisioning over cloud computing environments. They used the Mercury tool's script language to represent and evaluate dynamic reliability block diagrams (DRBD) to obtain the related metrics and represent some dependencies between the system's components. Already in Melo et al. [57], the authors extended those DRBDs to evaluate the availability of hyper-converged cloud computing environments. It is relevant to mention that DRBDs is an extension of common RBDs. However, with the possibility of evaluating and establishing a dependency between the system's components, in both papers, the evaluated environments were managed by OpenStack cloud computing platform, and deployment costs related to service provisioning were considered.

The authors in Melo et al. [56] evaluated the availability of blockchain-as-a-service platform based on Hyperledger Fabric in a cloud computing environment. The authors used RBDs and SPN models created on Mercury Tool to obtain the environment's annual downtime and detect the impact and the bottlenecks among the input parameters over the system's availability by applying difference sensitivity analysis.

In Liu et al. [43], the authors proposed monolithic availability models based on Stochastic Reward Net (SRN) formalism in order to evaluate the availability of a cloud computing infrastructure-as-a-service (IaaS) provisioning. The authors also had performed a parametric sensitivity analysis evaluation to identify the components that impact the most on the service provisioning availability considering a cloud data center infrastructure. Two repair routines were also considered on the sensitivity analysis evaluation and pointed out that shared maintenance can have lower costs and keep a similar availability value for individual maintenance applied to each server or site in a cloud data center.

Di Mauro et al. [17] introduced a characterization by an availability perspective of the container-based implementation of an IMS system (cIMS). The authors are interested in recognizing the optimal settings of a cIMS deployment that can provide five-nine availability requirements, meaning that a maximum downtime of 5 minutes and 26 seconds per year is allowed.

The work developed by Dantas et al. Dantas et al. [12] proposes hierarchical modeling to represent redundant cloud infrastructure, comparing their availability. The authors consider a high-level model based on RBD representing the Eucalyptus platform subsystems and a low-level model based on Continous Time Markov Chain representing the respective subsystems employing a warm-standby replication mechanism. in Dantas et al. [13], Dantas still considers as an extension of the paper. In this way, the authors include the acquisition cost and compare then to the public cloud.

In another work developed by Dantas et al. Dantas et al. [14], the authors estimate the availability and capacity-oriented availability through closed-form equations. The authors compare the approach to using models such as Continuous Time Markov Chains and SPN simulation model, considering execution time and values of metrics obtained with both approaches.

Qiu et al. [69] proposed a systematic study of correlations amongst availability, reliability, performance, and energy consumption metrics. Their models incorporate Markov models, Laplace-Stieltjes transform, a Bayesian approach, and a recursive method. In their work, the significant R-P-E relationships are examined using their models for estimating expected service time and energy consumption of a cloud service; their models estimate the resource usage under two typical recovery approaches, retrying and check-pointing.

The authors in Li et al. [39] used gray-Markov chains as an analytical model and measurement to enhance the data availability of edge computing environment connected to a traditional cloud system. They had worked out with replicas managed by their proposed algorithm and could improve both performance and data availability by bringing the data closer to the user. The authors also presented some scenarios that could be applied, some as manufacturing, smart cities, and augmented reality.

Ataie et al. [5] proposed a modeling approach to evaluate the performance and power consumption of an IaaS cloud while taking into account the availability. They presented a series of SRN models for computing the availability, performance, and power consumption of cloud environments and they also presented two estimated models that are flexible enough to evaluate the metrics of interest in cloud systems. The principal benefit of the fixed-point estimated model proposed by them is its capacity in modeling large cloud environments without state space explosion.

Mao et al. [50] proposed a visual model-based framework for cloud-based PHM using block definition diagram (BDD) and internal block diagram (IBD). The authors used structure diagrams and behavior diagrams to describe the static structure and dynamic states of PHM. They introduced a framework that permits several algorithm implementations of function modules. The authors performed a performance and availability evaluation and proposed a measurement method for their PHM framework.

Wang et al. [83] investigated the problem of wireless connections because, for fast-moving users (e.g., those on trains and buses), they are often unstable and unreliable. The authors carried real experiments on fast-moving trains to examine the quality of 3G connections. According to their results, the authors were able to see that the 3G connections were not stable and suffered from frequent disruptions of connectivity, and the connections that were placed in separate train compartments were mostly independent. Consequently, the authors proposed a brand-new fog computing structure, which serves as a middle layer between the end-users and the 3G infrastructure. Their results suggest that their proposal improves the reliability of wireless connections on fast-moving trains.

Liu et al. [44] designed an SRN model to understand the method of VM deployment and migration; they proposed a stochastic Petri net model, where they use different guard functions of specific transitions to represent several strategies. The authors assumed that all inter-event times are exponentially distributed; they also designed three metrics and computation techniques to analyze their algorithms, including energy usage, QoS, and Cost of Migration.

Addis et al. [1] proposed a utility-based optimization approach to considerably increase the availability and performance of cloud servers running virtual machines (VMs). The authors provided a theoretical framework to investigate the circumstances in which a hierarchical method may be applied.

Jammal et al. [34] proposed a high-availability component called CHASE for cloud environments. According to the authors, using CHASE, the high availability of services is achieved while considering performance and delay demands and redundancies; the authors also consider different failure ranges. An analysis is conducted to provide higher scheduling priorities for critical components than standard ones. The HA-aware scheduler estimates the availability of components using its mean time to failure (MTTF), mean time to repair (MTTR), and recovery time.

Santos et al. [70], proposed stochastic availability models to understand how failures in edge devices, fog devices, and cloud infrastructure impacts e-health monitoring system availability. The models are also used to perform sensitivity analysis to understand which components significantly impact e-health monitoring systems' availability. The authors also proposed stochastic performance models integrated with availability models and investigate the impact on performance metrics, such as service time and throughput.

Wang et al. [82] states that in the dynamic and volatile cloud environment, the performance and availability of services might vary due to several factors. Usually, QoS values are constant, and it does not reflect the consistency of the service in dynamic and volatile environments. Therefore, the authors proposed FT4MTS-D (fault-tolerant strategies for multi-tenant service-based systems with dynamic quality), which is a method that estimates the criticality of dynamic cost-effective fault tolerance strategies for multi-tenant service-based systems.

4.2 Reliability

This section discusses some papers that focus on reliability analysis on Cloud, Edge, and fog computing.

– Edge computing

Sun and Liu [77] proposed an uplink transmission approach in IoT powered by energy harvesting to permit each node to transmit data through heterogeneous networks, where the uplink transmission approach's primary goal is to improve communication reliability and availability. The authors also proposed performance metrics to analyze outage probability and ergodic capacity and analytical models.

Huang et al. [33] introduced a simulation-based optimization approach for REliability-Aware Service compositiON (REASON) for edge computing paradigm. The authors employ Stochastic Petri Net (SPN) to build the atomic service model for reliability-aware performance evaluation, which may model the dynamics of task arrivals, service procedures, failures, and recoveries. They also introduced a model combination scheme to express the complex service composition process, where scheduling between multiple layers and service collaboration inside or beyond layers can be dynamically modeled.

Li and Huang [41] introduced a theoretical approach to reliability-aware performance evaluation of IoT services. The authors proposed a generalized stochastic Petri net (GSPN) to express the IoT environments' variability, including task scheduling, queueing, request arrivals, failures, repairs, and recoveries; the authors modeled atomic services and comprehensive systems, and they also presented corresponding quantitative analyses. [41] conducted empirical experiments based on real-world data to provide a predictive methodology of performance evaluation of IoT services, taking into account the reliability of these environments.

Zilic et al. [89] propose the Energy Efficient and Failure Predictive Edge Offloading (EFPO) framework to determine the offloading decision policy's feasibility. They model the edge environment with Markov Decision Process (MDP). The results present performance and energy efficiency metrics, besides adopting failure rates to identify that Edge and computational server are less reliable than Edge regular and cloud.

Liang et al. [42] formulated a new reliability problem and, adopting combinatorial optimization techniques, developed a randomized algorithm and a heuristic to evaluate the proposed solution. The experimental results showed that the proposed solution is promising, and its empirical results are superior to the analytical ones.

– Cloud computing

In Lee et al. [38], the authors proposed a simulator to optimize the speed and reliability of a mobile cloud computing environment running over a smartphone cluster. The Mobile MapReduce Task Allocation (MTA) simulator had improved both associated metrics significantly by performing a better allocation process.

The authors in Tian et al. [79] used a data set from Google Cloud to identify, predict, and mitigate the impact of failures over a public cloud environment reliability and efficiency. They had analyzed the operational data and diagnosed the tasks that can impact the most on service provisioning. A predictive mathematical model and a set of simulations showed to be enough to mitigate issues and improve the system's reliability and efficiency.

In Dehnavi et al. [16] the authors evaluated the reliability of both hybrid cloud computing, edge and fog environments using the reliability expression previously assembled in the background section, as well as a set of simulation experiments over eight different setups. It is significant to mention that this research focused on the reliability requirements of real-time applications to smart factory environments. Once again, fog and Edge's use proved to be a sound alternative to improving cloud-based applications.

Huang et al. [32] investigated the reliability of modern distributed networks comprising computers, the internet of things (IoT), edge servers, and cloud servers for data transmission. The authors also state that a distributed network can be modeled as a graph with nodes and arcs, in which each arc represents a transmission line, and each node represents an IoT device, an edge server, or a cloud server. Therefore, the authors first constructed a model of a network to illustrate the flow relationship among the IoT devices, edge servers, and cloud servers and subsequently developed an algorithm to evaluate network reliability.

– Cloud, Fog and Edge computing

In Yousefpour et al. [86], the authors propose the *deepFogGuard*, a deep neural network (DNN) augmentation scheme for making the distributed DNN inference task failure-resilient. The simulation considered mainly different reliability settings for cloud, fog, and edge environments. The results presented demonstrated that the *deepFogGuard* is more efficient than a DNN that is not trained for failure resiliency.

5 Discussion on State of the Art

This section shows the results and discussion about the state of the art. This also includes remarks on the reliability and availability evaluation modeling approaches in Fog, Edge, and Cloud computing.

The relevant papers on our topics of interest have been explored in the well-known academic databases IEEE Xplore, Springer Link, ACM Digital Library, Science Direct and Scopus, as depicted in Table 2.

The research criteria took into account throughout this study are described as follows:

- Papers that were published between 2010 and 2020;
- The papers must have been published in English;
- Employment of technical metrics such as reliability and availability in their evaluation;

Database	Related URL
IEEE explorer	`https://ieeexplore.ieee.org/`
Springer Link	`https://link.springer.com/`
ACM	`https://dl.acm.org/`
Science Direct	`https://www.sciencedirect.com/`
Scopus	`https://www.scopus.com/`

Table 2: Academic databases searched in tho study

– Papers must be directly related to stochastic modeling formalisms such as CTMC, SPN, RBD, etc.;
– The papers should deal with infrastructure such as cloud, edge, or fog computing;

The initial search was executed in October 2020, considering the time range from 2010 to 2020. The results give 111 papers tha wee cut to 49 articles related to dependability measures (reliability or availability) and modeling techniques in the Edge, Fog and Cloud computing.

Figure 3 depicts the detailed distribution of 49 selected papers. Most papers (87.7%) were published between 2015 and 2020 in the related research direction, and the highest portion of the published papers was in 2019 with 15 papers, followed by 2020 with 12 papers. The year 2017 was the third with the highest production, with 7 paper published.

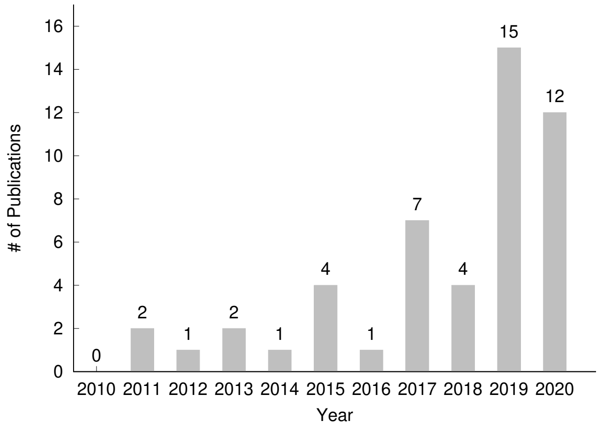


Fig. 3: Number of Publications per Year

Another result is depicted in Figure 4, which describes the most published papers summarized by geographical distribution. As it is shown, China holds the most published papers. According to the map, other countries have been working in the cloud, fog, and edge computing area, considering the reliability and availability. China has the most significant number of publications, with 15 published papers out of 49, followed by Brazil with nine published articles.

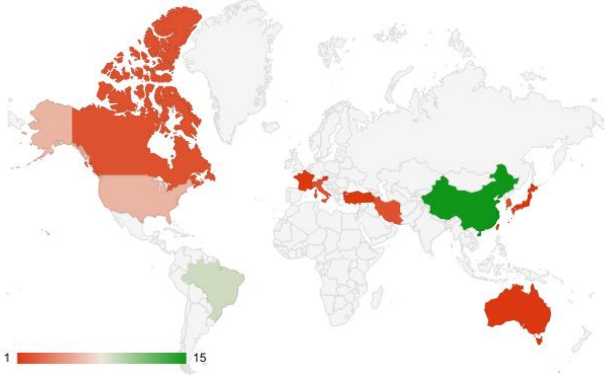


Fig. 4: Distribution of research articles per countries

Figure 5 depicted a comparison between the number of published papers and the infrastructure under study. According to the analyzed period from 2010 to 2020 and the search criteria in reliability and availability modeling, three infrastructures are considered: fog, edge, and cloud computing. As illustrated in the chart, most of the reviewed papers have as main focus the cloud computing infrastructure with coverage of 24 papers in the literature review. The cloud/fog and fog/edge's mixed infrastructure is covered by up four papers each one. Cloud/edge comes with five papers, and Cloud/Fog/Edge with three.

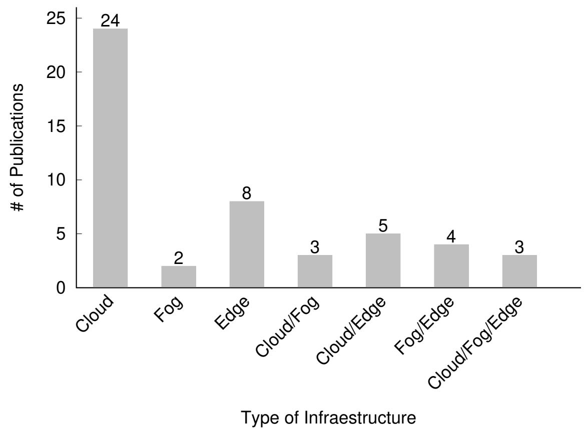


Fig. 5: Publication versus the type of infrastructure

Figure 6 shows a comparative study between the evaluation approach and formalism type. According to the formalism type, we made a preliminary study that may be seen in Figure 6a. The related field on the chart considers state-based, combinatorial, and closed-form equation models. Considering the appropriate keywords, most related papers related used the state-based type of formalisms. Also, it is shown in Figure 6c an overview of the evaluation approach analyzed. How we can see, most of the papers have been published during 2010 and October 2020, applying analytical models as the most common evaluation strategy. The combination of analytical solution and simulation is another relevant choice that have been analyzed.

Regarding the specific formalism, an analysis can be made by means of Figure 6b. We can see the Markov Chain and extensions as the most usual specific formalism considering that during these interval times, other techniques such as artificial intelligence, combinatorial optimization, and simulation have been used to solve availability and reliability problems. It is important to mention that the hierarchical modeling realizes a combination of some formalisms like SPN+CTMC+RBD or mathematical expression yet.

6 Challenges and Future Directions

For Edge, Fog, and Cloud computing to evolve, several challenges need to be overcome regarding both availability and reliability. In this section, we point out some of these challenges and key research directions to researchers who aim at surpassing them.

The Cloud is still an emerging paradigm, everyday new services and providers pop up, and the requirements to keep these services running are going up. The main alternatives to improve both reliability and availability attributes rely on redundancy techniques to mitigate the impact of failures over the services. When this redundancy is applied to the hardware stack, the energy consumption goes up, which may cause a bigger environmental impact than it usually should. Deployment, maintenance, and provisioning costs are important metrics that must be evaluated and considered in the cloud infrastructure planning process.

The Fog computing, besides being a recent advance on service provisioning, if not well modeled and planned, can drown due to a huge amount of data generated and sent by users from every place and device possible. The fog servers are not as powerful as a cloud data center and can succumb through legally performed tasks (Big Data) or distributed denial-of services attacks (DDoS) performed through IoT devices. The latter one, unfortunately, became more and more frequent in recent years. These attacks may affect the reliability and availability of both Fog and the Edge. However, if properly implemented, it can be an excellent resource to improve the quality of services offered, such as availability, reliability and performance.

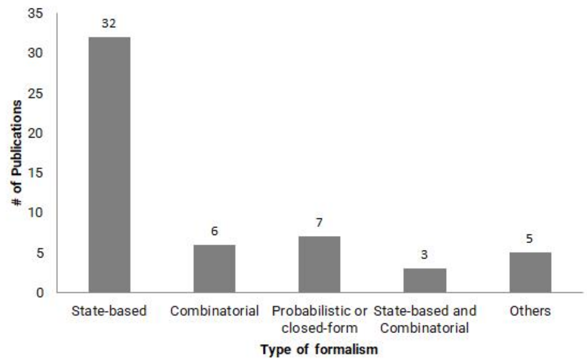
Edge computing is more susceptible to security attacks, due to the presence of vulnerabilities in some devices with lower processing power. If the proper care is not taken, it can be an easy source for an army of zombie bots. In this case, this same infrastructure could be remotely accessed and used in DDoS attacks to some third-party computing infrastructures.

Another *threat* to common service providers model is the distributed service provisioning that relies on the peer-to-peer network and distributed ledger technologies, such as blockchain-based applications. This new model presents a client that can be both service provider and user and can perform this task on their devices, an edge device. This can affect the performance of the provider's infrastructure, which may not have been designed for this type of application.

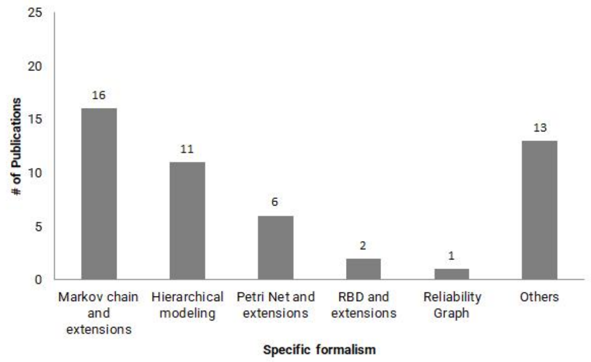
In any case, analytical models can be used to mitigate the impact of threats over the availability and reliability of services at any level of the edge-fog-cloud stack. Through models, one can know beforehand how much their computational infrastructure impacts the environment, how much it will spend with energy and maintenance issues, the impact of a DDoS attack, and the possibility to change from an old to an entirely new paradigm without spending a single cent to check its viability.

7 Conclusions

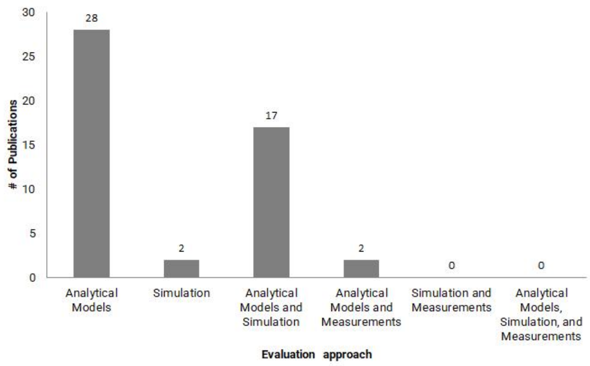
This survey presented the state of the art of availability and reliability modeling evaluation of Cloud, Edge, and Fog computing environments, applications, and services. We have performed a literature review and extracted scientific research such as journals and conference papers that result from the work of many people, universities, and research groups around the world. This survey identified that 2019 was the most productive year, being closely followed by 2020, which can surpass 2019, considering that the survey took into account only papers published until October 2020. It also identified that China is by far the country with the most published papers on the research topic, almost half of the total papers. It also found that the vast majority of papers adopted analytical models and state-based formalisms, more specifically Markov models and Petri nets. Finally, the most researched infrastructure by papers is the Cloud computing, but there is also a growing interest in Edge environments. Therefore, we were able to identify the most prominent research papers classified mainly by the dependability attributes (availability and reliability) and the adopted infrastructure, which allows pointing out possible gaps that can still be filled.



(a) Publication versus type of formalism



(b) Publication versus specific formalism



(c) Publication versus evaluation approach

Fig. 6: Comparing the published evaluation approach and the formalism type

Finally, we point out some challenges and future directions in these areas.

Acknowledgments

We would like to thank CAPES (the Brazilian Coordination of Improvement of Higher Education Personnel), CNPq (the Brazilian National Council for Scientific and Technological Development), FACEPE (Foundation Science and Technology Support of the State of Pernambuco) and MoDCS Research Group for their support.

References

1. Addis B, Ardagna D, Panicucci B, Squillante MS, Zhang L (2013) A hierarchical approach for the resource management of very large cloud platforms. IEEE Transactions on Dependable and Secure Computing 10(5):253–272
2. Ahmed M, Chowdhury ASMR, Ahme M, Rafee MMH (2012) An advanced survey on cloud computing and state-of-the-art research issues. International Journal of Computer Science Issues (IJCSI) 9(1):201
3. Andrade E, Nogueira B, de Farias Júnior I, Araújo D (2020) Performance and availability trade-offs in fog–cloud iot environments. Journal of Network and Systems Management 29(1):1–27
4. Angin P, Bhargava B, Jin Z (2015) A self-cloning agents based model for high-performance mobile-cloud computing. In: 2015 IEEE 8th International Conference on Cloud Computing, IEEE, pp 301–308
5. Ataie E, Entezari-Maleki R, Rashidi L, Trivedi KS, Ardagna D, Movaghar A (2017) Hierarchical stochastic models for performance, availability, and power consumption analysis of iaas clouds. IEEE Transactions on Cloud Computing

6. Avizienis A, Laprie JC, Randell B (2001) Fundamental concepts of computer system dependability. In: Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments, Citeseer, pp 1–16
7. Avizienis A, Laprie JC, Randell B, Landwehr CE (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Dependable Sec Comput 1(1):11–33
8. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp 13–16
9. Boukerche A, Soto V (2020) An efficient mobility-oriented retrieval protocol for computation offloading in vehicular edge multi-access network. IEEE Transactions on Intelligent Transportation Systems
10. Chiang M, Ha S, Chih-Lin I, Risso F, Zhang T (2017) Clarifying fog computing and networking: 10 questions and answers. IEEE Communications Magazine 55(4):18–20
11. Consortium O, et al. (2017) Openfog reference architecture for fog computing. Architecture Working Group pp 1–162
12. Dantas J, Matos R, Araujo J, Maciel P (2012) An availability model for eucalyptus platform: An analysis of warm-standy replication mechanism. In: 2012 IEEE international conference on Systems, man, and cybernetics (SMC), IEEE, pp 1664–1669
13. Dantas J, Matos R, Araujo J, Maciel P (2015) Eucalyptus-based private clouds: availability modeling and comparison to the cost of a public cloud. Computing 97(11):1121–1140
14. Dantas J, Araujo E, Maciel P, Matos R, Teixeira J (2020) Estimating capacity-oriented availability in cloud systems. International Journal of Computational Science and Engineering 22(4):466–476
15. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: Principles, architectures, and applications. In: Internet of things, Elsevier, pp 61–75
16. Dehnavi S, Faragardi HR, Kargahi M, Fahringer T (2019) A reliability-aware resource provisioning scheme for real-time industrial applications in a fog-integrated smart factory. Microprocessors and Microsystems 70:1–14
17. Di Mauro M, Galatro G, Longo M, Postiglione F, Tambasco M (2019) Ip multimedia subsystem in a containerized environment: availability and sensitivity evaluation. In: 2019 IEEE Conference on Network Softwarization (NetSoft), IEEE, pp 42–47
18. d'Oro EC, Colombo S, Gribaudo M, Iacono M, Manca D, Piazzolla P (2019) Modeling and evaluating a complex edge computing based systems: an emergency management support system case study. Internet of Things 6:100054
19. Ericson CA, Ll C (1999) Fault tree analysis. In: System Safety Conference, Orlando, Florida, vol 1, pp 1–9
20. Ever E, Shah P, Mostarda L, Omondi F, Gemikonakli O (2019) On the performance, availability and energy consumption modelling of clustered iot systems. Computing 101(12):1935–1970
21. Facchinetti D, Psaila G, Scandurra P (2019) Mobile cloud computing for indoor emergency response: the ipsos assistant case study. Journal of Reliable Intelligent Environments 5(3):173–191
22. Gamatié A, Devic G, Sassatelli G, Bernabovi S, Naudin P, Chapman M (2019) Towards energy-efficient heterogeneous multicore architectures for edge computing. IEEE Access 7:49474–49491
23. German R (2000) Performance analysis of communication systems - modelling with non-Markovian stochastic Petri nets. Wiley-Interscience series in systems and optimization, Wiley
24. Ghosh R, Longo F, Xia R, Naik VK, Trivedi KS (2013) Stochastic model driven capacity planning for an infrastructure-as-a-service cloud. IEEE Transactions on Services Computing 7(4):667–680
25. Ghosh R, Longo F, Frattini F, Russo S, Trivedi KS (2014) Scalable analytics for iaas cloud availability. IEEE Transactions on Cloud Computing 2(1):57–70
26. Gorbenko A, Romanovsky A, Tarasyuk O (2019) Fault tolerant internet computing: Benchmarking and modelling trade-offs between availability, latency and consistency. Journal of Network and Computer Applications 146:102412
27. Goyal A, Lavenberg SS (1987) Modeling and analysis of computer system availability. IBM Journal of Research and Development 31(6):651–664
28. Guan S, De Grande RE, Boukerche A (2016) A novel energy efficient platform based model to enable mobile cloud applications. In: 2016 IEEE Symposium on Computers and Communication (ISCC), IEEE, pp 914–919
29. Ha K, Chen Z, Hu W, Richter W, Pillai P, Satyanarayanan M (2014) Towards wearable cognitive assistance. In: Proceedings of the 12th annual international conference on Mobile systems, applications, and services, pp 68–81
30. Hardesty L (2017) Fog computing group publishes reference architecture
31. Hayes B (2008) Cloud computing
32. Huang CF, Huang DH, Lin YK (2020) Network reliability evaluation for a distributed network with

edge computing. Computers & Industrial Engineering 147:106492
33. Huang J, Liang J, Ali S (2020) A simulation-based optimization approach for reliability-aware service composition in edge computing. IEEE Access 8:50355–50366
34. Jammal M, Kanso A, Shami A (2015) Chase: Component high availability-aware scheduler in cloud computing environment. In: 2015 IEEE 8th International Conference on Cloud Computing, IEEE, pp 477–484
35. Jayashree L, Selvakumar G (2020) Edge computing in iot. In: Getting Started with Enterprise Internet of Things: Design Approaches and Software Architecture Models, Springer, pp 49–69
36. Jia C, Lin K, Deng J (2020) A multi-property method to evaluate trust of edge computing based on data driven capsule network. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, pp 616–621
37. Laprie JC (1992) Dependability: Basic concepts and terminology. In: Dependability: Basic Concepts and Terminology, Springer, pp 3–245
38. Lee Jw, Jang G, Jung H, Lee JG, Lee U (2019) Maximizing mapreduce job speed and reliability in the mobile cloud by optimizing task allocation. Pervasive and Mobile Computing 60:101082
39. Li C, Wang Y, Tang H, Zhang Y, Xin Y, Luo Y (2019) Flexible replica placement for enhancing the availability in edge computing environment. Computer Communications 146:1–14
40. Li J, Zhang T, Jin J, Yang Y, Yuan D, Gao L (2017) Latency estimation for fog-based internet of things. In: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, pp 1–6
41. Li S, Huang J (2017) Gspn-based reliability-aware performance evaluation of iot services. In: 2017 IEEE International Conference on Services Computing (SCC), IEEE, pp 483–486
42. Liang W, Ma Y, Xu W, Jia X, Chau SCK (2020) Reliability augmentation of requests with service function chain requirements in mobile edge-cloud networks. In: 49th International Conference on Parallel Processing - ICPP, Association for Computing Machinery, New York, NY, USA, ICPP '20
43. Liu B, Chang X, Han Z, Trivedi K, Rodríguez RJ (2018) Model-based sensitivity analysis of iaas cloud availability. Future Generation Computer Systems 83:1–13
44. Liu Y, Wang K, Ge L, Ye L, Cheng J (2019) Adaptive evaluation of virtual machine placement and migration scheduling algorithms using stochastic petri nets. IEEE Access 7:79810–79824
45. Longo F, Ghosh R, Naik VK, Trivedi KS (2011) A scalable availability model for infrastructure-as-a-service cloud. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), IEEE, pp 335–346
46. Machida F, Andrade E, Kim DS, Trivedi KS (2011) Candy: Component-based availability modeling framework for cloud service management using sysml. In: 2011 IEEE 30th International Symposium on Reliable Distributed Systems, IEEE, pp 209–218
47. Maciel P, Trivedi KS, Matias R, Kim DS (2011) Dependability modeling. In: Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions, IGI Global, Hershey
48. Mahmood Z, Ramachandran M (2018) Fog computing: concepts, principles and related paradigms. In: Fog Computing, Springer, pp 3–21
49. Malhotra M, Trivedi KS (1994) Power-hierarchy of dependability-model types. IEEE Trans Reliability 43(3):493–502
50. Mao K, Zhu Y, Chen Z, Tao X, Xue Q, Wu H, Mao Y, Hou J (2017) A visual model-based evaluation framework of cloud-based prognostics and health management. In: 2017 IEEE International Conference on Smart Cloud (SmartCloud), IEEE, pp 33–40
51. Marsan MA, Balbo G, Conte G, Donatelli S, Franceschinis G (1994) Modelling with Generalized Stochastic Petri Nets. John Wiley & Sons, Inc., USA
52. Matos R, Araujo J, Oliveira D, Maciel P, Trivedi K (2015) Sensitivity analysis of a hierarchical model of mobile cloud computing. Simulation Modelling Practice and Theory 50:151–164
53. Matos R, Dantas J, Araujo J, Trivedi KS, Maciel P (2017) Redundant eucalyptus private clouds: Availability modeling and sensitivity analysis. Journal of Grid Computing 15(1):1–22
54. Mell P, Grance T, et al. (2011) The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National ...
55. Melo C, Dantas J, Oliveira D, Fé I, Matos R, Dantas R, Maciel R, Maciel P (2018) Dependability evaluation of a blockchain-as-a-service environment. In: 2018 IEEE Symposium on Computers and Communications (ISCC), IEEE, pp 00909–00914
56. Melo C, Dantas J, Maciel R, Silva P, Maciel P (2019) Models to evaluate service provisioning over

cloud computing environments-a blockchain-as-a-service case study. Revista de Informática Teórica e Aplicada 26(3):65–74
57. Melo C, Dantas J, Maciel P, Oliveira DM, Araujo J, Matos R, Fé I (2020) Models for hyper-converged cloud computing infrastructures planning. International Journal of Grid and Utility Computing 11(2):196–208
58. Menascé DA, Almeida VA, Dowdy LW (2004) Performance by Design: Computer Capacity Planning by Example. Prentice Hall PTR
59. Molloy MK (1982) On the integration of delay and throughput measures in distributed processing models.
60. Molloy MK (1982) Performance analysis using stochastic petri nets. IEEE Trans Comput 31(9):913–917, DOI 10.1109/TC.1982.1676110, URL `https://doi.org/10.1109/TC.1982.1676110`
61. Murata T (1989) Petri nets: Properties, analysis and applications. Proceedings of the IEEE 77(4):541–580
62. Naha RK, Garg S, Georgakopoulos D, Jayaraman PP, Gao L, Xiang Y, Ranjan R (2018) Fog computing: Survey of trends, architectures, requirements, and research directions. IEEE access 6:47980–48009
63. Natkin SO (1980) Les Reseaux de petri stochastiques et leur application de l'evaluation des systemes informatiques
64. Nguyen TA, Min D, Choi E (2020) A hierarchical modeling and analysis framework for availability and security quantification of iot infrastructures. Electronics 9(1):155
65. Patel P, Ranabahu AH, Sheth AP (2009) Service level agreement in cloud computing
66. Pereira P, Araujo J, Maciel P (2019) A hybrid mechanism of horizontal auto-scaling based on thresholds and time series. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE, pp 2065–2070
67. Pereira P, Araujo J, Torquato M, Dantas J, Melo C, Maciel P (2020) Stochastic performance model for web server capacity planning in fog computing. The Journal of Supercomputing pp 1–25
68. Pierce WH (2014) Failure-tolerant computer design. Academic Press
69. Qiu X, Dai Y, Xiang Y, Xing L (2017) Correlation modeling and resource optimization for cloud service with fault recovery. IEEE Transactions on Cloud Computing
70. Santos GL, Endo PT, da Silva Lisboa MFF, da Silva LGF, Sadok D, Kelner J, Lynn T, et al. (2018) Analyzing the availability and performance of an e-health system integrated with edge, fog and cloud infrastructures. Journal of Cloud Computing 7(1):16
71. Sanyal S, Zhang P (2018) Improving quality of data: Iot data aggregation using device to device communications. IEEE Access 6:67830–67840
72. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39
73. Sharkh MA, Kalil M (2018) A quest for optimizing the data processing decision for cloud-fog hybrid environments. In: 2018 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, pp 1–6
74. Sharma PK, Chen MY, Park JH (2017) A software defined fog node based distributed blockchain cloud architecture for iot. Ieee Access 6:115–124
75. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: Vision and challenges. IEEE internet of things journal 3(5):637–646
76. Singh C, Billinton R (1977) System reliability, modelling and evaluation, vol 769. Hutchinson London
77. Sun W, Liu J (2017) Coordinated multipoint-based uplink transmission in internet of things powered by energy harvesting. IEEE Internet of Things Journal 5(4):2585–2595
78. Symons FJW (1989) Modelling and analysis of communication protocols using numerical petri nets.
79. Tian Y, Tian J, Li N (2020) Cloud reliability and efficiency improvement via failure risk based proactive actions. Journal of Systems and Software 163:110524
80. Trivedi KS, Hunter S, Garg S, Fricks R (1996) Reliability analysis techniques explored through a communication network example. Tech. rep., North Carolina State University. Center for Advanced Computing and Communication
81. Vahid Dastjerdi A, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: Principles, architectures, and applications. arXiv pp arXiv–1601
82. Wang F, Wang X, Zhang C, He Q, Yang Y (2020) Fault tolerating multi-tenant service-based systems with dynamic quality. Knowledge-Based Systems p 105715
83. Wang T, Peng Z, Wen S, Lai Y, Jia W, Cai Y, Tian H, Chen Y (2017) Reliable wireless connections for fast-moving rail users based on a chained fog structure. Information Sciences 379:160–176
84. Yakubu J, Christopher HA, Chiroma H, Abdullahi M, et al. (2019) Security challenges in fog-computing environment: a systematic appraisal of current developments. Journal of Reliable Intelligent Environments 5(4):209–233

85. Yi S, Hao Z, Qin Z, Li Q (2015) Fog computing: Platform and applications. In: 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), IEEE, pp 73–78
86. Yousefpour A, Devic S, Nguyen BQ, Kreidieh A, Liao A, Bayen AM, Jue JP (2019) Guardians of the deep fog: Failure-resilient dnn inference from edge to cloud. In: Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things, Association for Computing Machinery, New York, NY, USA, AIChallengeIoT'19, p 25–31
87. Yousefpour A, Fung C, Nguyen T, Kadiyala K, Jalali F, Niakanlahiji A, Kong J, Jue JP (2019) All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture 98:289–330
88. Zhou L, Guo H, Deng G (2019) A fog computing based approach to ddos mitigation in iiot systems. Computers & Security 85:51–62
89. Zilic J, Aral A, Brandic I (2019) Efpo: Energy efficient and failure predictive edge offloading. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing, Association for Computing Machinery, New York, NY, USA, UCC'19, p 165–175