

Highly Available Clouds: System Modeling, Evaluations, and Open Challenges

Patricia Takako Endo, Glauco Estácio Gonçalves, Daniel Rosendo, Demis Gomes, Guto Leoni Santos, André Luis Cavalcanti Moreira, Judith Kelner, Djamel Sadok and Mozhgan Mahloo

Abstract Cloud-based solution adoption is becoming an indispensable strategy for enterprises, since it brings many advantages, such as low cost. On the other hand, to attend this demand, cloud providers are facing a great challenge regarding their resource management: how to provide services with high availability relying on finite computational resources and limited physical infrastructure? Understanding the components and operations of cloud data center is a key point to manage resources in an optimal way and to estimate how physical and logical failures can impact on users' perception. This book chapter aims to explore computational modeling theories in order to represent a cloud infrastructure focusing on how to estimate and model cloud availability.

P.T. Endo (✉)
University of Pernambuco, Recife, Brazil
e-mail: patricia@gprt.br; patricia.endo@upe.br

G.E. Gonçalves
Rural Federal University of Pernambuco, Recife, Brazil
e-mail: glauco@gprt.ufpe.br

D. Rosendo · D. Gomes · G.L. Santos · A.L.C. Moreira · J. Kelner · D. Sadok
Federal University of Pernambuco, Recife, Brazil
e-mail: daniel.rosendo@gprt.ufpe.br

D. Gomes
e-mail: demis.gomes@gprt.ufpe.br

G.L. Santos
e-mail: guto.leoni@gprt.ufpe.br

A.L.C. Moreira
e-mail: andre@gprt.ufpe.br

J. Kelner
e-mail: jk@gprt.ufpe.br

D. Sadok
e-mail: jamel@gprt.ufpe.br

M. Mahloo
Ericsson Research, Recife, Brazil
e-mail: mozhgan.mahloo@ericsson.com

1 Introduction

Cloud providers have gained popularity because they have changed the current and traditional business models, replacing a huge initial investment by a pay-as-you-go model, in which users can deploy their applications with guarantees of high availability, scalability, and security. Currently, one of the biggest challenges cloud providers have facing is to guarantee increasingly stringent availability terms on SLAs (Service-Level Agreements), which is tightly connected to the strategies adopted for the full-stack failure management (from hardware to software) at their data centers. Unforeseen data center failures are expensive (for both sides, providers, and users) and require special attention. The costs of these failures stem from business disruption, lost revenue, diminished end-user productivity in addition to business reputation. To mitigate this issue, a deep understanding of the possible failures as well as the amount of effort and money to spend to fix them is of high interest.

There are some terms commonly used to describe systems performance regarding to their failures handling, such as availability, reliability, MTTF (Mean Time To Failure) and MTBF (Mean Time Between Failures). However, these concepts are frequently used without a precise definition and, in several cases, some of these concepts are used interchangeably. A clear definition of these concepts and how they are related to each other is necessary to understand how cloud services might be impacted in case of failure. Furthermore, a comprehensive modeling of the cloud infrastructure and its performance during runtime is vital towards detecting possible single points of the failure, and calculating the end-to-end availability of the services offered to the customers. A good starting point for reliability assessment of cloud services would be to model the hardware system in the data center in order to estimate the availability of each service, end-to-end, which can be done by defining the involved components related to each service. This helps to have a reliable architecture while modeling the data center infrastructures from the beginning to fulfill service requirements.

Considering aforementioned factors, this chapter will introduce essential concepts about high availability for cloud computing, as well as highlighting some open research questions, via presenting a survey about available modeling theories and how they can be used by cloud providers to handle service reliability issues.

The chapter aims to equip readers with a good insight about high availability challenges and modeling of cloud data centers. Specific competencies to be achieved by the reader at the end of this chapter are:

- To identify the basic principles used for designing and modeling high availability in cloud data centers;
- To identify what research topics should be relevant in high availability in cloud computing area in the coming years. This item is important to guide the future research in this area;
- To know the main available methods proposed by the scientific community on the cloud mechanisms for high availability, as well as understand how they relate to each other; and

- To comprehend the importance of cloud modeling and analysis in high availability area, identifying how one can develop solutions for this aspect.

This chapter is organized as follows: Sect. 2 describes the basic concepts regarding cloud high availability, such as definitions of Service Availability Forum (SAF) models, discussion about reliability *versus* availability, and some techniques for modeling high availability in clouds; Sect. 3 presents an overview of possible impacts of outages in cloud data centers, as well as an overview about data centers infrastructure, and some mechanisms to provide high availability; Sect. 4 shows a systematic literature review about modeling high availability clouds; Sect. 5 presents some open challenges; and finally the Sect. 6 describes our final considerations and future trends.

2 High Availability Concepts

High availability in data centers is of the high importance due to the impact of the failures on service continuity which can be translated into high operational costs for cloud providers and costumers. This Section describes some basic concepts and modeling techniques related to cloud availability.

2.1 SAF Concepts

With emergence of new technologies, several challenges arise for ICT (Information and Communications Technology) companies in regard to fulfilling the quality of their services towards customers. To attend the expectations of users, service providers need to provide high availability and reliability to their customers. On the other hand, they must reduce deployment costs of their services. To achieve these goals, many companies adhere to the open specifications. At this point, the SAF (Service Availability Forum) [43] standardizes interfaces to provide high availability to carrier-grade systems with off-the-shelf hardware platforms, middleware, and applications. The SAF is a consortium of many ICT companies that provides standardized solutions for building high availability services [45]. Using the standard solutions can reduce costs of deployment, human training, and software development, by using compatible and inter-operable solutions from different vendors.

SAF has developed a set of software specifications interfaces to middleware applications and carrier-grade platforms [43]. These specifications are divided into: AIS (Application Interface Specifications) and Hardware Interface Interface Specifications (named HPI (Hardware Platform Interface)). AIS defines standard interfaces to developers for building high availability programs that are portable in multiple platforms. On the other hand, the HPI allows ISVs (independent software vendors) provide COTS (commercial off-the-shelf) components, providing hardware

management platform across multiple heterogeneous platforms. Since this paper is treating of high availability on data centers, we described only the AIS components, because its focus is interfaced for providing high availability for services.

2.1.1 Application Interface Specifications

These specifications are formed by 12 services and two frameworks. The services are classified into three functional groups, and frameworks form another functional group [43].

The three services functional groups are: Platform Services, Management Services and Utility services. Platform Services provide abstraction of the hardware and operating system from other applications and services. Their components allow monitoring hardware and software components required for the nodes' operation. These abstractions facilitate the infrastructure management of clusters and can ensure their smooth operation. Management Services provide basic and standard management interfaces that can be used for the implementation and execution of applications and services. They also offer security, log, and notifications services that facilitate the management of applications and services. Finally, Utility Services provide common interfaces in distributed systems with high availability, such as event distribution and checkpointing messages. The implementation of these services are important for the system to provide high availability, due to mechanisms of detection and disaster recovery [15, 16]. Figure 1 shows a general architecture of framework components proposed by SAF.

SAF also standardizes two frameworks: AMF (Availability Management Framework) and SMF (Software Management Framework). AMF provides functions for managing availability of applications and middleware, while monitoring other softwares running on a system. In addition, AMF includes functions for error reports, life cycle management and health monitoring, which provide important information about services availability. The AFM setting allows prioritization of resources and provides many redundancy models [43]. On the other hand, SMF is used to manage the middleware and applications during upgrades. This framework maintains information about the availability and deployment of softwares and allows the system evolution and orchestrating the migration from one configuration to another. The SMF complements AMF providing a reliable and consistent framework that delivers and update the software in a system [15, 16].

2.2 Reliability Versus Availability

The reliability term is frequently used without a precise definition and, in several cases, this concept is used interchangeably with availability. However, these two terms are not conveying the same message [23]. Reliability can be defined as the ability of an item to perform its required functions for a stated time and under

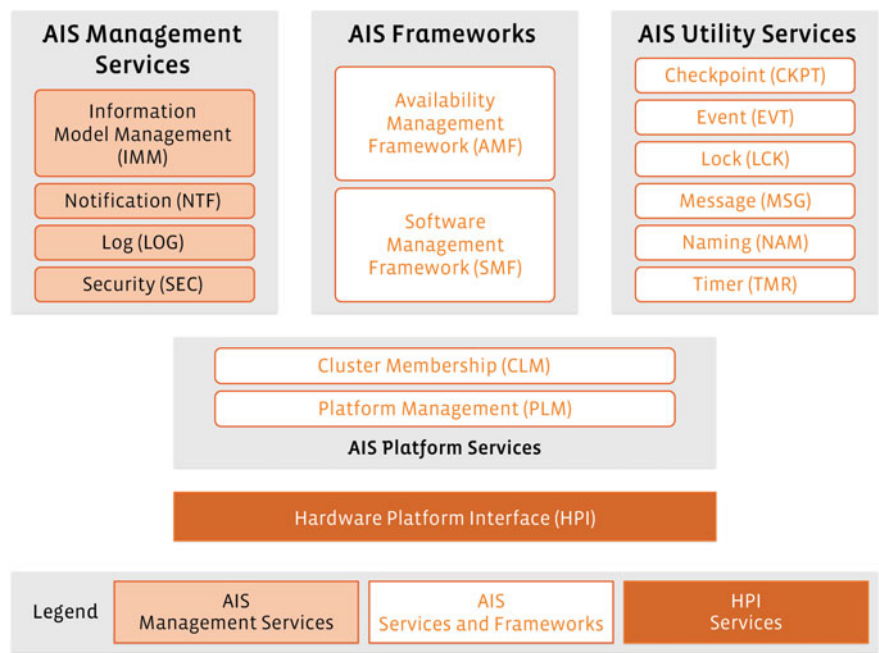


Fig. 1 Overview of SAF Framework components [15]

operational conditions. An item is any component or system, while required functions are the combinations of necessary actions to provide a service.

Reliability denotes the ability of an item to work properly until a failure occurs, independent of downtime and repair time. The ability of an item/system to be restored, using prescribed procedures and resources, to a state that it can perform its required functions is called maintainability [23]. Reliability is not influenced by maintainability and vice-versa because the former one is measured until a failure occur and the later one denotes the recovery rate of an item when it fails. A system can be less reliable and high maintainable or high reliable and less maintainable.

These two concepts (maintainability and reliability) are jointly defining the availability. According to Toeroe and Tam [47], availability is the percentage of time in which the service is up during a given interval. The QuEST Forum¹ describes availability as the probability that a system is running when it is required. More precisely, [23] introduces availability of an item/system as the combination of its reliability and maintainability to perform its required function at a stated instant of time or period.

In other words, availability is a probability of an item is functioning in a time t . This way, an item is more available when it is hard to fail (reliable) and has a high recovery rate (maintainable). The relation between reliability and maintainability to improve availability is shown in Table 1.

¹<http://tl9000.org/about/tl9000/overview.html>.

Table 1 Dependency of availability in relation to reliability and maintainability [4]

Reliability	Maintainability	Availability
Constant	Decreases	Decreases
Constant	Increases	Increases
Increases	Constant	Increases
Decreases	Constant	Decreases

The availability also can be calculated by service availability, as enlightened in Eq. 1. During service uptime, the service is operational. The service total time denotes the period in which a system is evaluated, being operational or not. Therefore, the service total time is the sum of operational time and the service downtime, as the Eq. 2 shows. In the downtime, the service is not operational, staying in the repair process until it is concluded.

$$serviceAvailability = \frac{serviceUptime}{serviceTotalTime} \quad (1)$$

$$serviceTotalTime = serviceUpTime + serviceDownTime \quad (2)$$

2.3 MTBF and MTTR

As discussed in Sect. 2.2, the availability can be defined as service uptime over total service time, where total time is described as the sum of service uptime and service downtime. These concepts can be associated with the average behavior of the system for the purpose of availability calculation. In the following formula, the availability is calculated by division of the MTTF (Mean Time To Failure) and the MTBF (Mean Time Between Failures). The MTBF also is defined as the sum of MTTF and MTTR (Mean Time to Repair), indicating the time between the detection of a failure and the detection of next failure, as showed in Eq. 3.

$$availability = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR} \quad (3)$$

Figure 2 illustrates the lifecycle of a hypothetical service. The variables d , u , and tt denote service downtime, uptime, and total time, respectively. A service can be either in downtime, defined by variables d_1 , d_2 , and d_3 , or in uptime, with variables u_1 , u_2 , and u_3 . Note that these times can have different values. Denoting n as the number of failures of system, 3 in our example, the MTTF can be calculated by Eq. 4, as well as MTTR is defined by Eq. 5. These equations state MTTF and MTTR as the

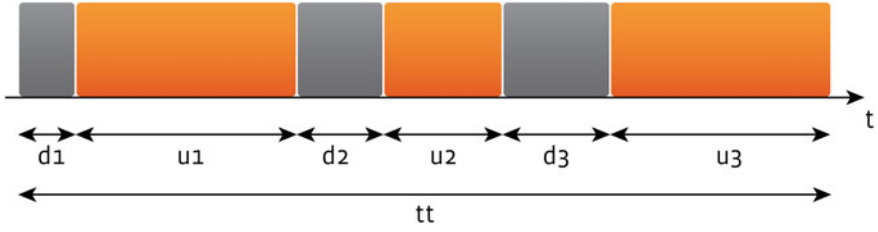


Fig. 2 MTTF, MTTR and MTBF related to service uptime, outage time and total time

averages of uptime and downtime, respectively. MTBF corresponds to mean service total time.

$$MTTF = \frac{\sum_{i=1}^i u_i}{n} \quad (4)$$

$$MTTR = \frac{\sum_{i=1}^i d_i}{n} \quad (5)$$

Some studies such as [13] and [8] do not employ MTTF metric. They replace the MTTF definition by MTBF, i.e., the authors state MTBF as the mean service uptime, denoted by time after failure recovery until the next failure. In the aforementioned studies, availability is calculated using Eq. 6. The result of this calculus is similar of Eq. 3 because MTBF assumes the same meaning that MTTF in these studies. We recommend the use of Eq. 3.

$$availability = \frac{MTBF}{MTBF + MTTR} \quad (6)$$

2.4 Techniques for High Availability Modeling

In order to avoid service outage and the consequential financial losses in cloud data centers, companies are interested in defining and applying formal models to verify and ensure correctness of hardware and software components to achieve highly available systems [10].

In this section, we describe various models used to solve real-life cloud high availability problems. We will mainly focus on the following three topics: (1) what are the advantages and drawbacks of each approach; (2) which model is better fitted to which scenarios/systems; and (3) how these models can be combined to extract the best of each one when modeling and analyzing cloud systems.

We briefly discuss the different modeling approaches that can be used to evaluate the dependability, availability, reliability, and fault tolerance of systems. These approaches differ according to their modeling power, ease of use, and system complexity [31]. In this way, they can be classified into three categories: Non-combinatorial or State-space models; Combinatorial or Non-state-space models; and Hierarchical and Fixed-Point Iterative models.

The Non-Combinatorial or State-Space Models are good for analyzing and verifying system behaviors. These models can be built using approaches like Markov Chains, Semi-Markov processes, Markov Regenerative Processes, Stochastic Petri Nets, or Stochastic Reward Nets. Those strategies make a state space structure that models all states and transitions that a system can reach (e.g., failures and repair operations). They permit the representation of complex systems with resource constraints and dependencies between subsystems [12]. However, those models face the state explosion problem, a problem related to the huge number of states in a system, making the built model difficult to be solved through analytical tools [10].

The Combinatorial or Non-State-Space Models enable a high level and concise representation of the relationship between components of systems and subsystems [48]. In this class, we can find Reliability Block Diagrams, Fault Trees, or Reliability Graphs. Differently from the state space methods, these methods are free of the state-space explosion problem. The main disadvantage of using the Combinatorial models is that they do not enable the representation of system activities and processes, such as rejuvenation, repairs, and failures.

The Hierarchical and Fixed-Point Iterative Models come to mitigate the weaknesses and put together the advantages of the Non-combinatorial and Combinatorial methods to leverage the analysis and modeling of many kinds of systems. This hybrid approach is commonly used to model systems with multiple components. In order to model such systems, it is recommended to use the combination of various simple methods to build multiple simple models, rather than using a single sophisticated model [40].

Some examples of models will be presented in Sect. 4, in which we present the results of a systematic review of the literature on the usage of models to assess high availability on cloud computing data centers.

3 How Do Clouds Achieve High Availability?

According to [8], hardware and software failures are inevitable. Highly available systems are designed so that no single failure causes unacceptable service disruption. Cloud providers have a great challenge to manage the data center infrastructure, considering the continuous need to optimize the resource usage and provide redundancy at the same time. This section presents some examples of how outages in data centers affected the service continuity and business reputation of several large companies, as well as an overview about mechanisms to provide high availability in data centers.

3.1 Outages Examples

The growing number of companies using cloud services brought several new challenges to the cloud providers. Maintaining high availability to meet the demands of these customers is a difficult task for providers. This demand is dynamic, and cloud services must be always available as the service interruption may represent high financial losses for cloud users [20].

Often the root cause of a cloud service unavailability is not completely clear to its customers, but one can point several recent cases which was made public. For instance, on March 13, 2009, during an upgrade operational system, the Deployment Service of Windows Azure began to fail due to network interruption problems. Many applications that running only one instance stopped when the corresponding server went down. On October 3, 2009, BitBucket was unavailable for 16 h. The reason were two DDoS attacks targeted at the network interfaces on Amazon EBS (Elastic Block Store) service for storage used with EC2 instances [41].

On 2012, Salesforce, a cloud company that provides on-demand software service, including CRM (customer relationship management), faced an outage that lasted 7 h. The root cause was a failure in the data center power system which affected several CRM customers. On May 6, 2014, the cloud service provider, Internap, faced three interruptions of services at its New York data centers. The reason was a fault in the power supply system that affected 20 companies, including the online video streaming platform Livestream. Another example of outage occurred in the Joyent data center, a company that provides high performance cloud infrastructures services. The event took place on May 27, 2014, and it was related to a human error that restarted the whole system. On September 3, 2014, the social network Facebook was down during 10 min. Many users realized the unavailability and demonstrated discontent on other social networks [5].

The outages' cost may be high for companies that provide or use cloud services. The Ponemon Institute conducted studies about costs of outages in data centers [2]. The recent study performed in 2016, evaluated 63 data centers of 49 companies in the United States. The survey showed an 7% increase in the average cost of data center outage of \$690,204 in 2013 to \$740,357 in 2015. Among the types of costs associated with outages, which generate more expenses for companies are business interruptions, something around \$256,000. The study also showed that the average outages times, in 12 months, is 95 min, an increase with respect to 2013 average, that was 86 min. The study also showed the maximum cost per minute of an outage is \$17,244.

The financial impacts and losses related to service outages can prove the importance of the high availability mechanism in cloud computing area. The best way to avoid breach of contract due the unavailability of service is to assess and measure the availability that data centers are able to deliver.

3.2 *Datacenter Overview*

In this section, we present the main components of a data center infrastructure that is composed of IT equipment (servers, storage, and network), electrical, and mechanical subsystems. We also present some data center standards that define best practices and recommendations regarding data center design and infrastructure.

3.2.1 **Information Technology Infrastructure**

Data center IT equipment may be classified as servers, storage, and networking devices. Servers are mounted within racks and consist of hardware resources (such as CPUs, NICs, and RAMs) hosting applications like video streaming and big data. All the data generated by these applications are stored in storage systems.

Data center storage consists of high capacity (around Terabytes) disk drives or flash devices. The storage tiers are connected to the servers either directly or through networking devices (managed by a global distributed file system), forming a NAS (Network Attached Storage) [7]. The RAID (Redundant Array of Independent Disks) storage technology can be used to provide high availability, redundancy, and increase fault tolerance.

Networking equipment manages the internal communication between servers and storage systems as well as all the input/output data flow from/to the data center. Typically, a data center network is designed based on three hierarchical levels: core, distribution, and edge. It is through the core level that all data center traffic (ingress/egress) to the outside world will be managed. Distribution is a level between the edge and core levels which aggregates the edge switches. Its main goal is simply network management and cabling reduction. Finally, the edge level passes data from devices (generating or consuming data) to the distribution or core levels [46].

Manage all these components is a great challenge because hardware clusters have a deep and complex memory/storage hierarchy, the system is composed of heterogeneous components, and there are many failure-prone components. It is necessary to use an additional software layer to provide an abstraction of this variety of components. According to [7], this software layer has three levels: platform level, cluster level, and application level. The platform-level software is composed of firmware, operational system, and kernel that provide an abstraction of hardware of a single machine in the cluster and provide server-level services. The cluster-level software is related to any software that managed resources at cluster level, such as distributed file systems, management resource systems, and monitoring systems. The application-level software is composed of software that implements a particular service. The set of all software used to monitor, measure, manage and control the data center behavior is named DCIM (Data Center Infrastructure Management).

3.2.2 Power Infrastructure

IT infrastructure needs power facilities with enough capacities (generators and UPSs) to operate properly. Hence, faults in power system components directly affect the overall data center availability. The CENELEC EN 50600-2-2 standard defines requirements and recommendations for planning and designing data center power supply facilities. The standard introduces Availability Classes from I to IV for the power supply and distribution systems to address various level of data center availability by including layers of redundancy.

A typical data center power system architecture includes an utility substation, an alternate power supply, a transfer switchgear or an ATS (Automatic Transfer Switch), an UPS (Uninterruptible Power Supply) system, and a PDU (Power Distribution Unit). The main power supply of a data center is the utility substation. Data centers may also contain an alternative power feed like fuel cell and renewable energy sources (such as solar, wind, bioenergy, hydroelectric, and wave) [25]. Both primary and secondary power sources are connected to an ATS. The ATS provides input for the cooling and UPS systems. The UPS system routes power to the PDU (rack socket for cabinets). Finally, a PDU distributes electrical energy to the IT equipment. Figure 5 depicts those components.

3.2.3 Cooling Infrastructure

As the power infrastructure, the cooling (or mechanical) system reliability and maintainability are fundamental to a proper data center operation. The heat dissipation of IT equipment requires the deployment of cooling system design strategies [22, 27]. The ASHRAE TC9.9 standard defines thermal design recommendations regarding data center cooling technologies, air flow rack level design (hot aisle and cold aisle), IT equipment (network, storage, and server), and energy-saving techniques (reducing cooling fans speed).

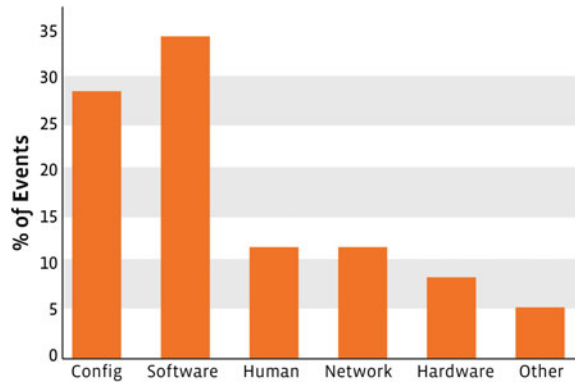
Data center cooling system equipment can be based on different technologies, such as central cooling, water-cooled, air-cooled, direct expansion, evaporative cooling, water economization, direct economization, indirect economization, and economization options. Each one differs according to the operation mode, energy efficiency, costs, heat exchanger technology, and cooling technology [27].

A typical data center cooling system relies on some cooling components, which includes a CRAC (Computer Room Air Conditioning), chillers, cooling towers, piping, pumps, heat exchanger, and water treatment systems.

3.3 High Availability Mechanisms

Providing redundancy in data centers comes with a price. There is always a trade-off between increasing the availability and the amount of required cost investments to

Fig. 3 Distribution of service disruption events by most likely cause at one of Google's main services, collected over a period of 6 weeks (adapted from [7])



achieve desired level of reliability [7]. Without protection mechanisms, failures can make the data center inaccessible to the users, which might lead to SLA violations and financial losses for the cloud provider. However, having preventive strategies such as planned maintenance of some components in predefined time intervals can reduce service downtime when a failure occur, preventing users to be affected by the failure.

Moreover, it is necessary to discover the main sources and severity of failures which can impact on the data center performance, to offer an uninterrupted service to the users or meet the SLA requirements. Among the data center components, as shown in Fig. 3, the major failures are related to the software (almost 35%) and misconfiguration errors (almost 30%). Hardware components represent between 5% and 8% of failures, whereas network contributes to around 10% of total data center failures. Human errors also cause 10% of failures in data centers. Therefore, high availability mechanisms in a data center must be focused on software failures, so that if a tolerant mechanism at the software level is implemented robustly, it can maintain the service up even if certain hardware failure occur [7].

However, it should be noted that reliable hardware architecture planning and infrastructure design is the basis of any service, meaning that without availability of redundant network paths and hardware resources, it is not possible to implement high availability on the software level. Besides, a failure in a hardware component causes a very high downtime: in network devices, a downtime caused by failures in a device correspond to 78% of total downtime caused by all probable errors [19]. Data centers must implement some mechanisms to mitigate unplanned outages. Distributed storage, health monitoring, and disaster recovery mechanisms are some examples of available methods to achieve highly available cloud services. These strategies are complementary: a disaster recovery requires a monitoring health service to check components and a distributed storage to recovery data in another entity. Distributed storage focuses on errors on application level and must have an easy configuration to avoid misconfiguration problems; health monitoring and disaster recovery provide redundancy in software, but trigger when a failure occurs in the hardware or network.

In a similar way, HDFS (Hadoop Distributed File System) [44] also stores data in geo-diverse nodes. Its architecture is divided into three main nodes: DataNodes, which stores data blocks and can rebalance data distribution among them; NameNode, which manages files information such as namespace, permissions and mapping files to DataNodes; and CheckpointNode, that provides fault tolerance and increases availability saving files and merging it with NameNode.

Regarding to health monitoring, that is related to checking the resources consumption of the application instances running on cloud, we can cite Google System Health, which offers high availability [7]. This solution monitors the configuration, activity, and error data from each server, storing this information in a repository that allows some analytic engine to diagnose and suggests the best approach for repairing or preventing the failure. VMware vSphere² also provides solutions such as the automatic restart of VMs in the servers, instantaneous live migration, and automatic remediation by monitoring at application level. These solutions can be integrated with distributed storage to provide a fast recovery in case of failure.

Disaster recovery has a high adoption rate in data centers. The use of redundant data centers mitigate the service downtime when an active data center turns inaccessible by events such as human errors, fire, terrorist attacks or natural disasters. Huawei provides a disaster recovery solution [3] in three different levels: application level, data level, and media level. These levels have different recovery procedures and times: a failure in application level must be recovered in minutes by replication of virtual machines; in media level, switches and data are mirrored in another data center, with the recovery is established in hours; and media level makes a backup of main data center in others data centers, and its protocol decides which data center will replace the failed one. The recovery is made in one or more days.

VMware offers a Disaster Recovery-as-a-Service called VMware vCloud Air Disaster Recovery. This service avoids the data center to implement your own disaster recovery mechanism using a more consolidated service. The service leverages vSphere Replication, as discussed beforehand, to provide robust, asynchronous replication capabilities at the hypervisor layer. This solution is limited to vSphere environments.

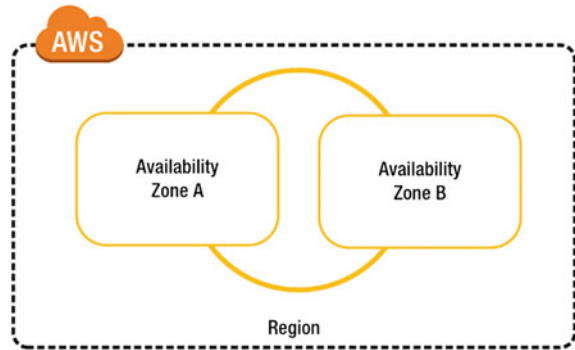
More information about mechanisms used to achieve high availability in clouds can be found in [15].

3.4 Commercial Solutions

This section describes briefly some commercial solutions and shows how they provide high availability.

²VMware: Business Continuity and Disaster Recovery— <http://www.vmware.com/solutions/business-continuity.html>.

Fig. 4 Amazon Availability Zones (AZ) can be considered as separate data centers, with separate power, cooling, and Internet access. The elastic load balancing can be used to balance traffic between them [9]



3.4.1 AWS Cloud

The AWS (Amazon Web Services) Cloud provides an elastic load balancing that automatically distributes incoming application traffic across multiple Amazon EC2 instances, spreaded around the world into multiple regions. Each region is independent of the others in order to design highly available applications with low latency response time to the customers (Fig. 4).

According to AWS site,³ the AWS Cloud also supports disaster recovery (DR) architectures for small customer workload data center failures and for hot standby environments that enable rapid failover at scale. With data centers in several regions around the world, AWS provides a set of cloud-based DR services that enable rapid recovery of the IT infrastructure and data.

3.4.2 Google Cloud Platform

Google Cloud Platform⁴ is a platform that provides services such as compute, storage, machine learning, big data, and more. In general, Google uses its own tools to keep the services available, such as GFS (Google File System). GFS is one of the most well-known distributed storage solutions [18], comprised of disk redundancy and efficient resource allocation. It is designed to provide efficiency, availability, and scalability to large volumes of data in data centers. GFS distributes files into chunks of 64 MB that are managed by a master node.

Google Cloud has a service focused on providing high availability in SQL instances. The developer configures an instance to failover in a failure case or replicates automatically the data in other zones. When a zone comes unavailable, Google Cloud failover SQL instance in another zone is available.⁵

³aws.amazon.com.

⁴<https://cloud.google.com/>.

⁵<https://cloud.google.com/sql/faq>.

3.4.3 Microsoft Azure

Azure⁶ is the Microsoft platform of cloud computing that offers several services for host applications. This platform enables to use other Microsoft solutions for processing data, such as big data and machine learning. Azure has many mechanisms for providing high availability and DR, such as load balancing, replication of data and redundancy of components. Some mechanisms are executed automatically, but in some cases the application developer should have an additional work to configure these mechanisms. For instance, Azure load balancing mechanism uses round-robin to evenly distribute jobs across instances. However, if complexity of jobs varies greatly, it is possible that some instances assigned with a number of complex jobs while other instances remain idle [6].

The structure of Azure is also divided logical and physically into regions, that are composed of data centers. Under some circumstances, it is possible that an entire region become unavailable, for instance due network failures or natural disasters. Azure offers redundancy approach and backup of VMs. Azure provides mechanism that is referred to Geo-Redundant Storage (GRS); GRS replicates storage to a paired data center hundreds of miles apart within a specific geographic region [15].

3.4.4 IBM Cloud

IBM Cloud comprises a set of cloud computing services including IaaS, PaaS and SaaS through public, private, and hybrid delivery models. IBM cloud offers specific products which provide cloud solutions for compute, storage, network, security, management, data and analytics, that can be combined together with other open or third-party solutions. For instance, IBM Bluemix is platform that combines PaaS and IaaS available in local, dedicated, and public and also offers a suite of instant-on services, including Watson,⁷ Data Analytics, and Mobile Services.⁸ It can deploy a cloud infrastructure using, for instance, OpenStack or CloudFoundry, without losing integration with other cloud solutions like, for instance, IBM Cloud orchestrator,⁹ a cloud management platform for automating provisioning of cloud services.

High availability is mainly provided by IBM cloud services themselves but also by the specific solutions. Also, high availability can be enabled for the cloud service providers and the applications running in the cloud. IBM Bluemix offers a catalog of availability monitoring services that can run synthetic tests to proactively detect and fix performance issues before they impact users. It also has multi-region architectures and supports different services configuration and data replication.

⁶<https://azure.microsoft.com/pt-br/overview/what-is-azure/>.

⁷<https://www.ibm.com/watson/>.

⁸<https://www.ibm.com/cloud-computing/bluemix>.

⁹<http://www.ibm.com/software/products/en/ibm-cloud-orchestrator>.

3.5 Infrastructure Standards and Tiers

Standards such as TIA-942, Uptime Institute, ANSI/BICSI 002, ASHRAE TC9.9, and CENELEC EN 50600-x define fundamental aspects, best practices, and recommendations regarding data center design and infrastructure. The TIA (Telecommunications Industry Association) covers key topics related to the site space planning, cabling infrastructure, environmental considerations, and tiered reliability. Based on the service requirement and the criticality of applications running on the clouds, data centers can be divided into four tiers with different availability levels.

Uptime Institute consortium provides different recommendations related to the level of redundant components, points of failure, watts per square foot, and availability for each tier. Tier specification goes from Tier 1 to 4, where higher tiers provide greater availability and inherit requirements of lower tiers. Though, increase in the reliability comes in the price of higher costs and operational complexities. The tier classification standards help to compare data centers reliability and design strategies. Table 2 presents some of these recommendations according to tier classification.

The redundant components refer to the number of IT equipment, cooling, and power components that comprises the data center infrastructure. In Tier 1, N means no redundancy indicating that system failures will result in outages. While in Tiers 2, 3, and 4, $N + 1$ means that there is some level of component redundancy. The number of delivery paths refers to the number of distribution paths of the power and

Table 2 Uptime Institutes Tier Classification System for data centers [28]

Tiers	Description	Redundant components	Number of delivery paths	Availability level
Tier 1 Basic	Planned and unplanned activity may cause system disruption	N	Only 1	99.671%
Tier 2 Redundant	Less susceptible to system disruption from planned and unplanned activity	$N+1$	Only 1	99.741%
Tier 3 Concurrently Maintainable	Equipment replacement and maintenance do not require disrupting computer hardware operation	$N+1$	1 active and 1 passive	99.982%
Tier 4 Fault Tolerant	Adds fault tolerance to the infrastructure. Sustains a worst case, unplanned event with no critical load impact	$2(N+1)$	2 simultaneously active	99.995%

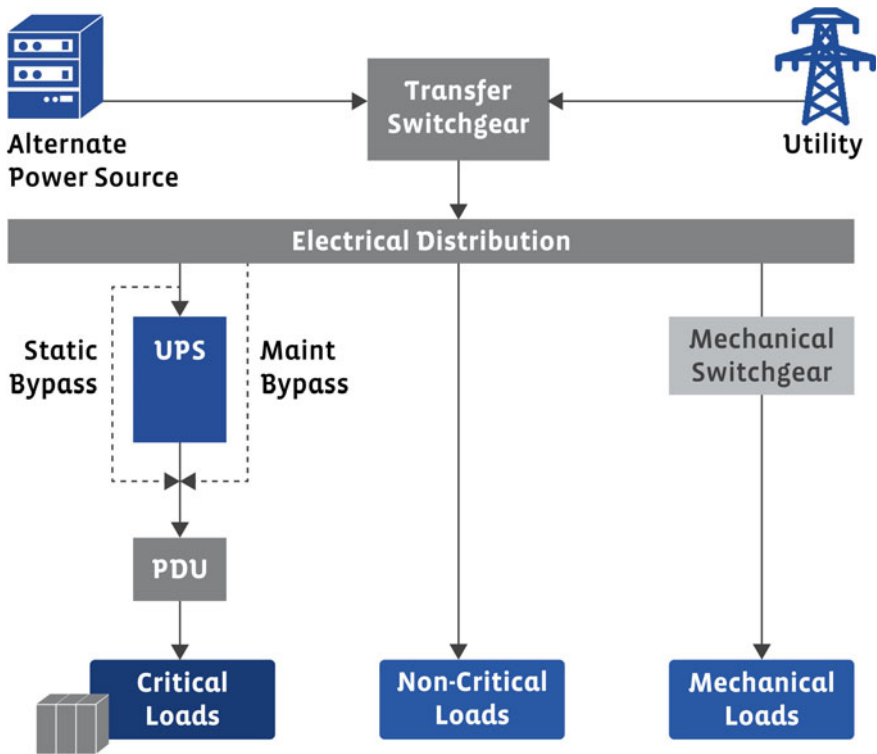


Fig. 5 Example of a power infrastructure—Tier 1 classification (adapted from BICSI [1])

cooling systems serving the IT equipment. That way, more distribution paths result in higher overall system availability [17]. Figures 5 and 6 show examples of a power infrastructure with Tier 1 and 4, respectively.

It is important to highlight that the tier classification relies on the all data center segments (IT equipment, electrical infrastructure, mechanical infrastructure, and facility operations) to deliver the overall availability. Therefore, a data center with a Tier 2 mechanical system and a Tier 4 electrical system will result in a Tier 2 data center availability rating.

Another important note is that the tier selection depends on the business requirements (such as availability, employment costs, and downtime financial consequences), meaning that a Tier 4 selection may not be the best option for a data center running non-critical workloads [49].

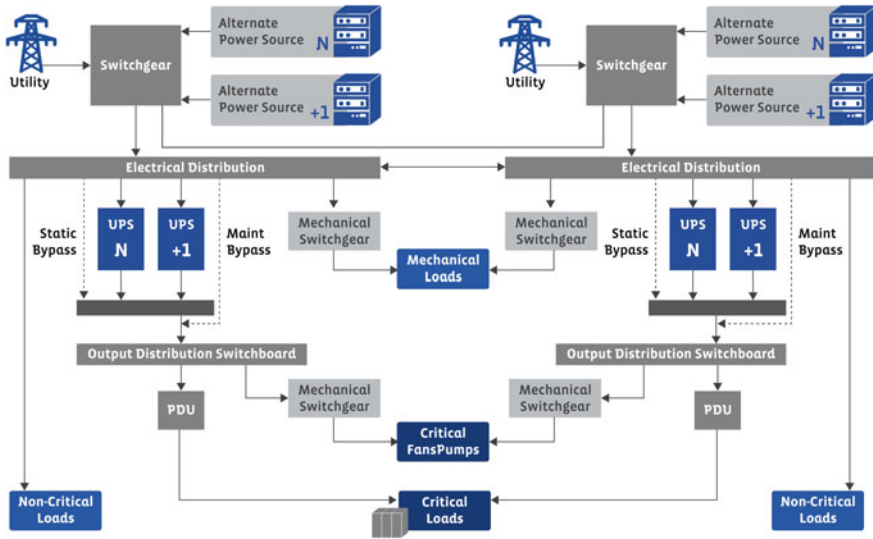


Fig. 6 Example of a power infrastructure—Tier 4 classification (adapted from BICSI [1])

4 Modeling High Availability Clouds

In order to clarify what have been done in the scientific literature about modeling high availability clouds, we performed a systematic review. Using this method, we intend to provide reader with a broad view of the field. The following subsections describe our methodology, results, and some discussions.

The main goal of this systematic review was to answer the following research questions (RQ):

- RQ.1: What is the current state of the art in high availability cloud modeling?
- RQ.2: What are the most common metrics used to measure HA in cloud systems models?
- RQ.3: What are the most common data center subsystems modelled to evaluate high availability in clouds?
- RQ.4: What are the most common approaches used to model high availability for clouds?
- RQ.5: What are the main remaining research challenges in this field?

The initial search returned 8, 20, 110, and 186 articles from ACM Digital Library, IEEE Xplore, Springer, and Science Direct, respectively, totaling 324 works. By reading all abstracts and using the criteria for inclusion or exclusion, we selected 15 papers for data extraction and quality evaluation.

Fig. 7 Number of articles per year

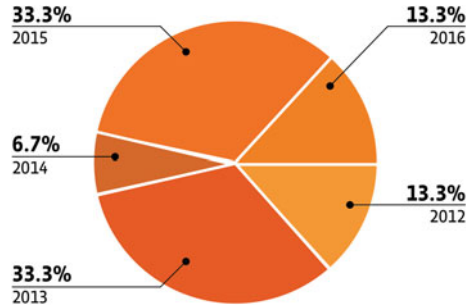
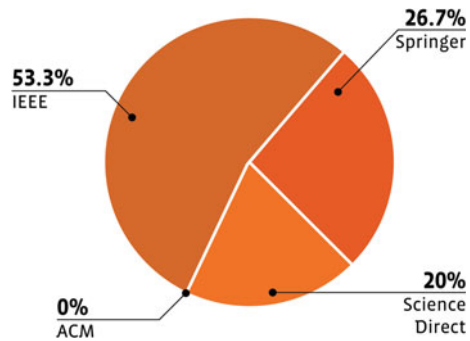


Fig. 8 Number of articles per research source



4.1 Overview of High Availability Modeling for Clouds

Considering the RQ.1, Fig. 7 shows the number of published articles per year; while the Fig. 8 shows the number of articles per research source. As one can note, 2013 and 2015 concentrate most of the works done in this research area; and IEEE is the research source with more articles published in this area.

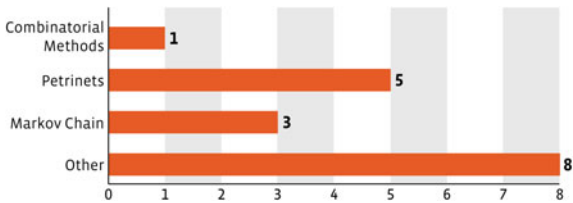
To answer the RQ.2, Table 3 summarizes some metrics and their respective definition presented in various articles addressing highly available cloud solutions. Service availability and SSA (Steady-State Availability) are the most common metrics used by authors [13, 24, 30, 35, 36, 42] and are related directly to the available and operational time of a target service. We also found metrics related to the provider costs due service unavailability, such as downtime cost analysis [38] and number of transactions lost [30]. On the other hand, authors in [37] modeled metrics to a specific application, MMOG (Massively Multiplayer Online Game); these metrics are related to interruptions caused by game unavailability, and two metrics are interesting because analyze the unavailability impact on players, severity of the interruptions and average non-served clients.

Answering the RQ.3, the results showed that computing resources are the most focused subsystems (with 93.3%), which are addressed in the literature, reinforcing the crucial role of the IT infrastructure.

Table 3 Main metrics used to evaluate the cloud availability

Metrics	Definitions	References
Service availability	The percentage of served requests in comparison to the total number of received requests	[24]
	The uptime over a year	[35]
	It can be understood as the probability that the system is found operational during a given period of time or has been restored after the occurrence of a failure event	[13]
Steady-state availability (SSA)	Number of nines	[36]
	Represents the long-term probability that the system is operating correctly	[42]
	and available to perform its functions	[30]
Downtime cost analysis	caused by a disaster in a data center per year in minutes	[38]
Number of transactions lost	It is the number of transactions lost due to VMM rejuvenation	[30]
Instantaneous non- interruption ratio	Ratio between the measured state update frequency of the MMOG within one measurement timestep and the required minimal frequency	[37]
Total non-interruption ratio	The percentage of time the MMOG the state update frequency equal or greater than the required frequency, over a given time interval	[37]
Duration of the interruptions	The start of the failure to the moment when all affected clients recover	[37]
Number of interruptions	Number of interruptions in a time interval	[37]
Severity of the interruptions	The percentage of affected players	[37]
Average non-served clients	Number of clients who were denied service	[37]

Fig. 9 The most common approaches used to model high availability



Finally, considering the RQ.4, Fig. 9 shows the most common approaches used to model high availability in cloud data centers. In order to detail these solutions, we will describe them in next subsections.

4.1.1 Markov Chain Solutions

Authors in [13, 26, 36] used Markov chain approach to model cloud availability. In [26], authors proposed three CTMC (Continuous Time Markov Chain) submodels that capture a specific aspect of a cloud data center. These submodels are RASM (Resource Allocation Submodel), VMPSM (Virtual Machine Provisioning Submodel), and Availability Model; and the integration between them is shown in Fig. 10.

While in [36], authors used an extended DTMC (Discrete Time Markov Chain) to model computing resources of a multi-cloud and a controller (dual layer) to guarantee system availability while minimize costs (Fig. 11). The DTMC has physical nodes (that represent concrete elements in the cloud, such as physical server or a pool of VMs offered by a cloud provider) and logic nodes (that represent the success or the failure state of the application). The DTMC has control variables and measured

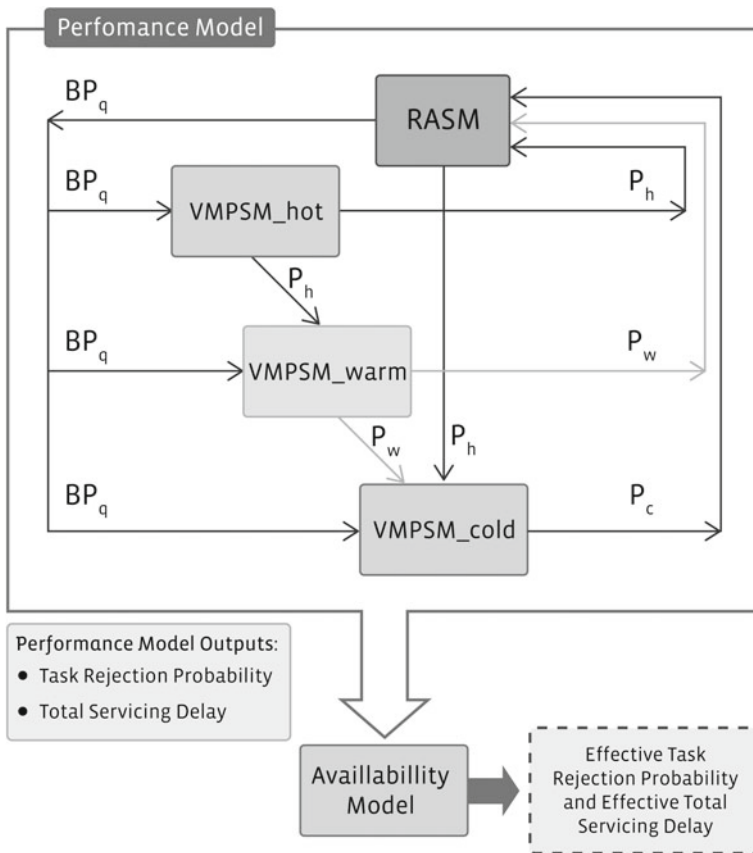


Fig. 10 Submodels integration proposed by [26]

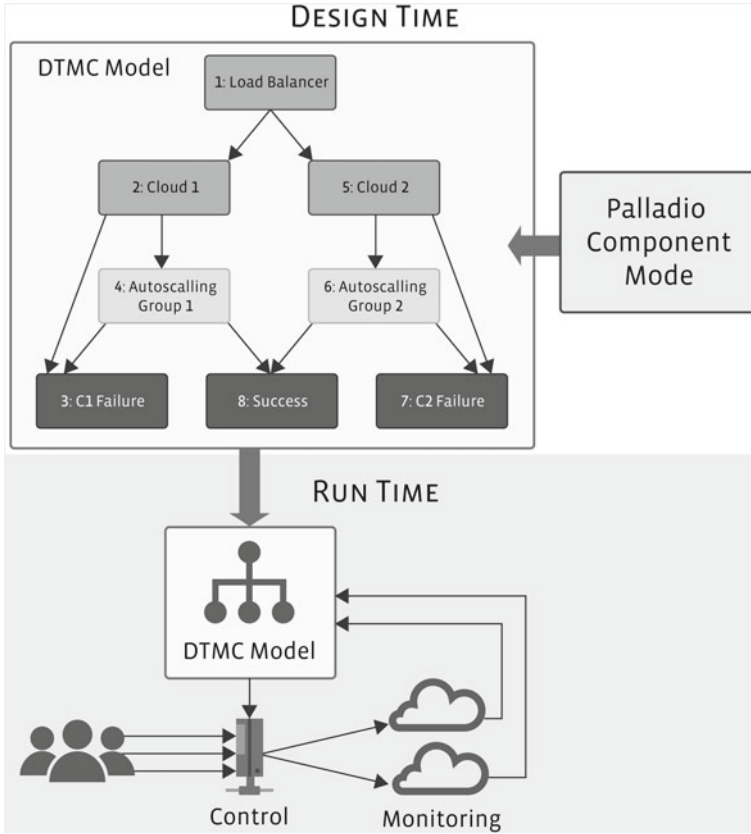


Fig. 11 Approach proposed by [36]

availability as labels to transitions; and rewards or costs can be introduced. The DTMC model is meant to be kept alive at runtime, and the controller is responsible for modifying it, changing the effects on the actual implementation.

Authors in [13] used a different Markov chain, named MRM (Markov Reward Model), in conjunction with RBD (Reliability Block Diagrams) in order to represent the system behavior, that cannot be captured by only pure RBDs. The RBD model (Fig. 12) is used to describe the high-level components, whereas the MRM is used to model the components involved in the evaluation of availability in [13]. In a similar way, in [12], authors also used RBD and MRM, but they model the components involved in the redundancy mechanism. Figure 13 shows the RMR model proposed to represent a redundant system composed of two nodes.

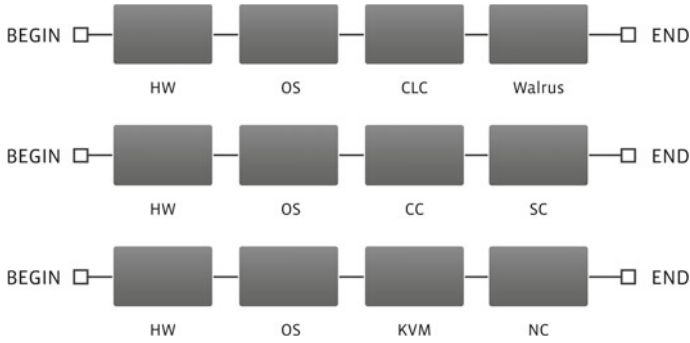


Fig. 12 RBDs that represent cloud, Cluster and Nodes Subsystems (from *top to down*) proposed by [13]

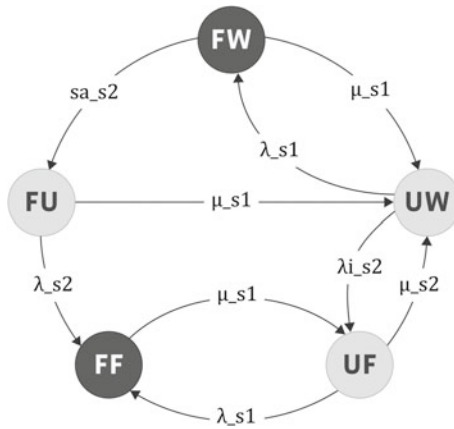


Fig. 13 MRM model that represents a redundant system with two nodes proposed by [12]

4.1.2 Petrinets Solutions

Petrinets are also used to model cloud availability [24, 30, 34, 35, 38, 42]. In [24], authors presented a SCPN (Stochastic Colored Petri Nets) model, which is a class of DSPN (Deterministic Stochastic Petri Nets) models, to evaluate the availability of cloud services and take in consideration the application deployment in geographically distributed data centers. For that, authors propose five different SCPN building blocks, shown in Fig. 14: DC (data center), server, VM (virtual machine), load balancer, and component (multi-tier app).

Petrinets can also be used with others approaches. Still considering the environment with more than one data center, in [38], authors used SRN (Stochastic Reward Nets) to model each component (such as VM, host, storage, data center) as subsystem and the global system is composed of all data centers. Authors consider the disaster

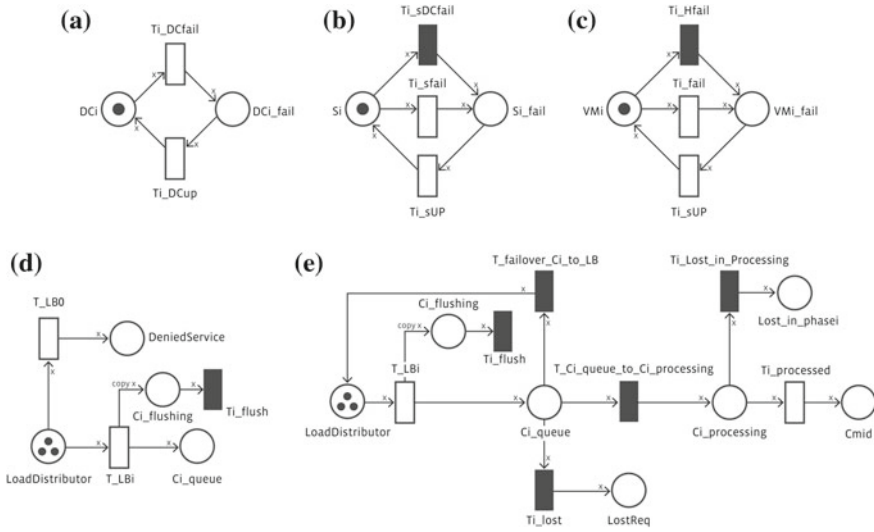


Fig. 14 SCPN models proposed by [24]

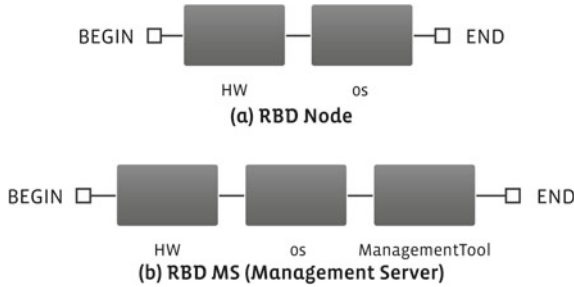


Fig. 15 RBDs that represent Node and Management Server used by [35] and [34]

tolerant scenario, and make a analysis about the trade-off between system availability and downtime cost with infrastructure construction cost.

In [35], SPNs were used to describe low-level modeling and RBD is used again at high-level modeling, forming a hierarchical model to describe system rejuvenation. The RBD models consider only the non-aging failures. Figure 15a shows that a node fails only if both hardware (HW) or operating system (OS) fail; and Fig. 15b shows that a failure in the management tool may provide a fault in the Management Server, independent of hardware and operating system. The proposed SPN model is composed of three submodels: (a) Management Server Model, (b) Clock Model and (c) System Model (Fig. 13).

Following the same combination (including the same RBD models presented in Fig. 15), the same authors evaluated other rejuvenation policies and find steady-state availability and expected annual downtime in [34]. The proposed SPN models were

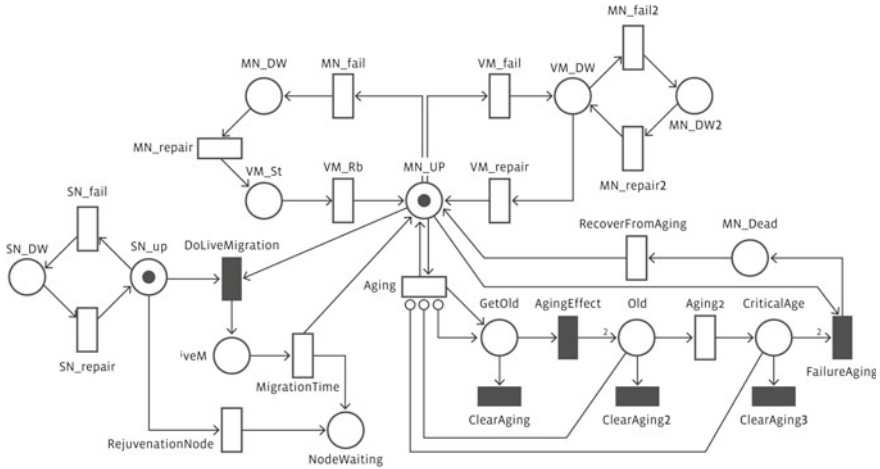


Fig. 16 SPN that represents the system submodel proposed by [34]

based on [35], and they improve the model with a Clock Model with check, that is a enhanced version of Clock Model. The Clock Model with check execute a check on VMM aging status before perform a live migration. Figure 16 shows the SPN that represents the proposed system submodel.

In [30], authors studied the effectiveness of VM migration rejuvenation by modeling it more precisely using CTMC model. However, due to the complexity of construction of CTMC models, the availability models of a server virtualized system is done in an extension of Petrinets, named SRN (Stochastic Reward Net). In [42], authors also used SRN, but they used a SRN hierarchical model for modeling memory virtualization and availability. The proposed SRN is transformed in a MRM to steady state and/or transient analysis.

4.1.3 Other Solutions

Other approaches were used to model cloud availability. In [32], authors used Bayesian networks to allocation of cloud resources in order to maximize the service dependability. Others used mathematical approaches, such as probability distribution [21] to model the accumulated downtime, statistical distribution [37] to model MMOGs (Massively multiplayer online games) as a service-based market, and combinatorial method [39] to define the availability model, considering the operational power and replication mechanism.

4.1.4 Proprietary Tools

There are some commercial tools that simulate a data center infrastructure. 6SigmaDCX¹⁰ is a commercial simulator that uses CFD (Computational Fluid Dynamics) technique to provide levels of productivity for data center design, troubleshooting and operation. 6SigmaDCX models many data center components, such as cooling, power and IT infrastructures. 6SigmaDCX allows 3D model of data center and configuration of several metrics (such as type of customer, data center floor space and critical IT load). The simulation results are represented by graphs and can be integrated with several DCIM (Data Center Infrastructure Management) solutions.

CoolSim¹¹ is a tool for performing occasional simulations to determine the best location for cooling and IT equipment. This simulator also uses CFD for modeling the airflow in data centers. An interesting feature of CoolSim is that it allows users to pay as you go for use of the application through a cloud SaaS delivery model.

However, the main disadvantage of these solutions is the price. For instance, the annual license of CoolSim is about \$15,000. As an alternative, there is other non-commercial simulators, such as the BigHouse [33], that is a tool focused on infrastructure for data center systems. Instead of simulating servers using detailed micro-architectural models, BigHouse raises the level of abstraction using a combination of queuing theory and stochastic modeling. BigHouse leverages statistical simulation techniques to limit simulation turn around time to the minimum runtime needed for a desired accuracy.

4.2 Discussion

From the review of the literature, we can observe that the cloud availability can be modelled from various perspective based on the goal of the study. However, a holistic view of the all layers are still missing.

We can highlight that each of the presented modeling approaches has its advantages and also disadvantages. For instance, RBDs are commonly used due to their simplicity, but they are not suitable when detail behaviors of systems need to be described. Markov chains can be used with RBDs or separately, but it is not scalable enough for modeling large systems. Hence, to have a comprehensive models and framework addressing all aspects of high availability in data centers, a combination of various models should be considered. Until now, a huge effort was put on modeling the computing resources, while the rest of the subsystems, such as power and cooling systems, are nearly neglected in regarding their availability models.

The RQ.5 that deal with main challenges will be described in next section.

¹⁰<http://www.futurefacilities.com/solutions/data-centers/>.

¹¹<http://www.coolsimsoftware.com/Home.aspx>.

5 Open Research Challenges

The enterprise applications that rely on data center infrastructure are inherently more available, robust, and scalable. Physical resilience is assured by virtual resource redundancy, and it brings a relatively simple way to deal with a hardware component outage, by migrating the virtual application to other available and healthy physical resource. However, according to [17], *while the spreading of risk across virtual systems reduces the risk of physical outage, complexities of running pools of highly integrated systems have its own challenges*.

These challenges include the need to address the deterioration of physical buildings and systems of the data center, and also have to embrace the degradation of virtual servers, storage, and network components. In this section, we highlight some of the most important challenges in cloud data center availability.

5.1 Cloud Data Center Modeling and Simulation

A comprehensive modeling of the data center system during runtime is the first step towards detecting possible single points of the failure, and calculating the end-to-end availability of the services offered to the customers. A good starting point would be to model the hardware system in the data center in order to calculate the availability of each service, end-to-end, by defining the involved components related to each service.

However, according to [11], *“because of heterogeneous software/hardware components and complicated interactions among them, the probability of failures improves. The services reliability arouses more attention”*. Cloud reliability analysis and modeling are very critical but hard because of the complexity and large scale of the system.

Authors in [29] say that *“a monolithic model may suffer from intractability and poor scalability due to vast parameter space”*. In this way, some authors have proposed separate submodels for different subsystems of a complex cloud center, such as [26]. However, one should integrate these subsystems in a coherent way, considering the general aspect of the model.

Beyond that, there are also some tools that make use of these models to simulate data center behavior under different circumstances, such as hardware failures, resource allocation, and even disasters. However, most of them are focused on physical and logical layers.

5.2 High-Level Metrics

PUE (Power Usage Effectiveness) has been used as a good metric to classify data centers' energy performance. According to [17], PUE determines the energy efficiency

of data centers, and has been used worldwide in the technology industry, becoming a mainstream approach to determine data center energy use efficiency. PUE is defined as Eq. 7.

$$PUE = \frac{\sum PowerDeliveredToDataCenter}{\sum ITEquipmentPowerUse} = \frac{\sum P_{mechanical} + P_{electrical} + P_{other}}{\sum P_{IT}} \quad (7)$$

There are many other metrics related to data center performance, such as ERE (Energy Reuse Effectiveness), WUE (Water Usage Effectiveness), Carbon Usage Effectiveness (CUE) and Return Temperature Index (RTI), as shown in Table 4. However, common metrics suffer from lack of relation with user perception. For instance, a system with availability of 0.9999 is down for an of average 52 min during a year. However, if the down time occurs during a peak hour, it has a higher impact than a failure that occurs when the system load is low, and less users will be affected by that failure. It is important to look for other metrics that go beyond the power, mechanical, and IT systems

To address this issue, other high-level metrics can be defined. For instance, we can consider FIF (Failure Impact Factor) proposed by [14]. FIF helps to measure the risk of any single failure for the owner. The definition of this parameter can also help designing a system with improved reliability without investing more than needed. By using FIF it is possible to define the components with high risk in the system, and protect them up to the level that guarantees no more than a given number of

Table 4 Data center performance metrics (from [17])

Metric	Description	Equation
ERE	Recognize that some data centers have the ability to provide energy that can be reused in other parts of the facility or campus	$ERE = \frac{AnnualFacilityEnergyUse - AnnualEnergyReused}{AnnualITEnergyUse}$
WUE	Determine the efficacy of water use in the data center, based on the energy used by the IT equipment	$WUE = \frac{AnnualSiteWaterUsage}{AnnualITEnergyUse}$
CUE	Judge the amount of carbon that is expended as compared to the annual IT energy used in the data center	$CUE = \frac{AnnualCO_2emissionscausedbythedatacenter}{AnnualITEnergyUse}$
RTI	Determine the efficacy of the air management in a data center	$RTI = \frac{ReturnAirTemp - SupplyAirTemp}{RackOutMeanTemp - RackInMeanTemp} * 100$

customers/services will be affected by a single failure in the system at any given time. Risk is defined by combining the probability of the failure to occur and the severity of the incident (scenario) occurring. So, for calculating FIF, first the system availability should be calculated.

For instance, authors in [14] define a resilience parameter, namely the failure impact (FI) in rational and irrational environments.¹² The FI in a rational environment is proportional to the number of customers disconnected by the failure, N , and the unavailability of the component, U . On the other hand, the FI in an irrational environment, all failures are statistically independent and all failures have a binary consequence: connection is fully disconnected (0) or not (1), no intermediate situations are considered [14]. The FI is given by Eq. 8.

$$FI = N^\alpha \times U \quad (8)$$

where $\alpha > 1$ leads to more and more irrationality, and $\alpha = 1$ is the rational environment.

In [37], authors use different metrics in their experiments that reflect the unavailability impact on MMOG players, such as number of interruptions in a time interval, duration of the interruptions (the start of the failure to the moment when all affected players recover), severity of the interruptions (% of affected players), average non-serviced players (# of players who were denied service). These metrics are interesting because relate failures and users interruptions, and were described previously in Table 3.

Another high-level metric is the lost revenue, a simple way of calculating the potential loss in a data center outage [17], as shown in Eq. 9.

$$LostRevenue = \frac{GR}{TH} \times I \times H \quad (9)$$

where GR denotes gross yearly revenue, TH denotes total yearly business hours, I denotes percentage impact, and H denotes number of hours of outage.

According to [17], loss can be viewed in different perspectives, such as monetary loss, reputational loss, employee productivity loss, client loyalty loss, and also combinations of all of these. Every business will suffer different degree of cost and, in the end, must balance against the risk and the cost of a disruption.

According to [17], “it is important to understand that these metrics should be used together, providing a range of data points to help understand the efficiency and effectiveness of a data center; different combinations of these metrics will produce a synergistic outcome”.

¹²According to authors, “an irrational environment is where a network operator is worried more about a big failure disconnecting all clients for 1 h at the same time than for multiple small failures throughout the year disconnecting every client for 1 h on average [14].”

6 Final Considerations

According to Gartner, Inc.,¹³ “by 2020, a corporate “no-cloud” policy will be as rare as a “no-internet” policy is today”, in other words, the migration to a cloud-based solution is practically infeasible. It is interesting to note that it does not mean that everyone will be cloud-based; but the scenario with no-cloud will gradually vanish.

Due to a growing number of companies that use services on cloud, many challenges begin to emerge. Serving the demand of many services is a task complicated, due to a limited resources of the data centers. Understanding the cloud data center is a key point to manage resources and to estimate how physical and logical failures occurred on a data center can impact their users’ perception.

In this work, we presented basic concepts needed to understand the mechanisms to provide high availability, consequences of outages in cloud data centers and a data center infrastructure overview.

Furthermore, we also presented a systematic literature review about cloud high availability, highlighting the main approaches used to model it. Whereas we have found Markov Chain, RBD, and Petrinets as main approaches, some articles have used hybrid approaches, due to limitations of each technique.

The cloud data center is a complex and big system composed of other subsystems, such as power and cooling. Some research questions emerge when we are modeling these subsystems separately: how can they be integrated in order to improve the understanding of overall data center behavior? How does a failure in one of these subsystems affect the overall data center and the cloud users? Which protection strategies can we suggest to mitigate these negative impacts?

Acknowledgements This work was supported by the RLAM Innovation Center, Ericsson Telecomunicações S.A., Brazil.

References

1. Ansi/bicsi 002, data center design and implementation best practices. Retrieved November 2016, from https://www.bicsi.org/uploadedFiles/BICSI_Website/Global_Community/Presentations/CALA/Ciordia_002_Colombia_2016.pdf.
2. Cost of data center outages: Data center performance benchmark series. Retrieved November 2016, from <http://www.emersonnetworkpower.com/en-US/Resources/Market/Data-Center/Latest-Thinking/Ponemon/Documents/2016-Cost-of-Data-Center-Outages-FINAL-2.pdf/>.
3. Data center disaster recovery and backup solution. enterprise. Retrieved November 2016, from enterprise.huawei.com/ilink/enenterprise/download/HW_322364.
4. Relationship Between Availability and Reliability. Retrieved November 2016, from <http://www.weibull.com/hotwire/issue26/relbasics26.htm>.
5. Top 4 data center outages of 2014. Retrieved November 2016, from <http://www.cyrusone.com/blog/top-5-data-center-outages-of-2014/>.
6. Bai, H. (2014). *Zen of cloud: Learning cloud computing by examples on microsoft azure*. CRC Press.

¹³<http://www.gartner.com/newsroom/id/3354117>.

7. Barroso, L. A., Clidaras, J., & Hözlze, U. (2013). The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis Lectures on Computer Architecture*, 8(3), 1–154.
8. Bauer, E., & Adams, R. (2012). *Reliability and availability of cloud computing*. Wiley.
9. Brian Beach. (2014). *Pro powershell for amazon web services: DevOps for the AWS cloud*. A press.
10. Clarke, E. M., Klieber, W., Nováček, M., & Zuliani, P. (2011). Model checking and the state explosion problem. In *LASER Summer School on Software Engineering*, pp. 1–30. Springer.
11. Chen, J., Liu, Y., Cui, H., & Li, Y. (2013). Methods with low complexity for evaluating cloud service reliability. In *Proceedings 16th International Symposium on Wireless Personal Multimedia Communications*, pp. 1–5. IEEE.
12. Dantas, J., Matos, R., Araujo, J., & Maciel, P. (2012). An availability model for eucalyptus platform: An analysis of warm-standby replication mechanism. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1664–1669. IEEE.
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. Dixit, A., Mahloo, M., Lannoo, B., Chen, J., Wosinska, L., Colle, D., & Pickavet, M. (2014). Protection strategies for next generation passive optical networks-2. In *2014 International Conference on Optical Network Design and Modeling*, pp. 13–18. IEEE.
15. Endo, P. T., Rodrigues, M., Gonçalves, G. E., Kelner, J., Sadok, D. H., & Curescu, C. (2016). High availability in clouds: Systematic review and research challenges. *Journal of Cloud Computing*, 5(1), 16.
16. Gailey, G., Taubensee, J., Rabeler, C., Glick, A., & Squillace, R.: Azure resiliency technical guidance: Recovery from a region-wide service disruption. Retrieved December 2016. <https://docs.microsoft.com/en-us/azure/resiliency/resiliency-technical-guidance-recovery-loss-azure-region>.
17. Geng, H. (2014). *Data center handbook*. Wiley.
18. Ghemawat, S., Gobiuff, H., & Leung, S.-T. (2003). The google file system. In *ACM SIGOPS operating systems review*, vol. 37, pp. 29–43. ACM.
19. Gill, P., Jain, N., & Nagappan, N. (2011). Understanding network failures in data centers: Measurement, analysis, and implications. In *ACM SIGCOMM Computer Communication Review*, vol. 41, pp. 350–361. ACM.
20. Gonçalves, G., Endo, P. T., Rodrigues, M., Kelner, J., Sadok, D., & Curescu, C. (2016). Risk-based model for availability estimation of saf redundancy models. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 886–891. IEEE.
21. Gonzalez, A. J., & Helvik, B. E. (2013). Hybrid cloud management to comply efficiently with sla availability guarantees. In *2013 12th IEEE International Symposium on Network Computing and Applications (NCA)*, pp. 127–134. IEEE.
22. Hoelzle, U., & Barroso, L. (2009). The datacenter as a computer. *Morgan and Claypool*.
23. Høyland, A., & Rausand, M. (2009). *System reliability theory: models and statistical methods*, vol. 420. Wiley.
24. Jammal, M., Kanso, A., Heidari, P., & Shami, A. (2016). A formal model for the availability analysis of cloud deployed multi-tiered applications. pp. 82–87. IEEE.
25. Kao, W., & Geng, H. (2015). Renewable and clean energy for data centers. *Data Center Handbook*, pp. 559–576.
26. Khazaei, H., Mišić, J., Mišić, V. B., & Mohammadi, N. B. (2012). Availability analysis of cloud computing centers. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 1957–1962. IEEE.
27. Kosik, W. J., & Geng, H. (2014). Energy and sustainability in data centers. *Data Center Handbook*, pp. 15–45.
28. ADC Krone. (2008). Tia-942: Data center standards overview.
29. Longo, F., Ghosh, R., Naik, V.K., & Trivedi, K.S. (2011). A scalable availability model for infrastructure-as-a-service cloud. In *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*, pp. 335–346. IEEE.

30. Machida, F., Kim, D. S., & Trivedi, K. S. (2013). Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration. *Performance Evaluation*, 70(3), 212–230.
31. Malhotra, M., & Trivedi, K. S. (1994). Power-hierarchy of dependability-model types. *IEEE Transactions on Reliability*, 43(3), 493–502.
32. Marrone, S. (2015). Using bayesian networks for highly available cloud-based web applications. *Journal of Reliable Intelligent Environments*, 1(2–4), 87–100.
33. Meisner, D., Wu, J., & Wenisch, T. F. (2012). Bighouse: A simulation infrastructure for data center systems. In *2012 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 35–45. IEEE.
34. Melo, M., Araujo, J., Matos, R., Menezes, J., & Maciel, P. (2013). Comparative analysis of migration-based rejuvenation schedules on cloud availability. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4110–4115. IEEE.
35. Melo, M., Maciel, P., Araujo, J., Matos, R., & Araújo, C. (2013). Availability study on cloud computing environments: Live migration as a rejuvenation mechanism. In *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 1–6. IEEE.
36. Miglierina, M., Gibilisco, G. P., Ardagna, G. P., & Di Nitto, E. (2013). Model based control for multi-cloud applications. In *2013 5th International Workshop on Modeling in Software Engineering (MiSE)*, pp. 37–43. IEEE.
37. Nae, V., Prodan, R., & Iosup, A. (2014). Sla-based operations of massively multiplayer online games in clouds. *Multimedia Systems*, 20(5), 521–544.
38. Nguyen, T. A., Kim, D. S., & Park, J. S. (2016). Availability modeling and analysis of a data center for disaster tolerance. *Future Generation Computer Systems*, 56, 27–50.
39. Noor, T. H., Sheng, Q. Z., Yao, L., Dustdar, S., & Anne, H. H. (2016). Ngu. CloudArmor: Supporting reputation-based trust management for cloud services. *IEEE Transactions on Parallel and Distributed Systems*, 27(2), 367–380.
40. Pelánek, R. (2008). Fighting state space explosion: Review and evaluation. In *International Workshop on Formal Methods for Industrial Critical Systems*, pp. 37–52. Springer.
41. Pham, C., Cao, P., Kalbarczyk, Z., & Iyer, R. K. (2012). Toward a high availability cloud: Techniques and challenges. In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, pp. 1–6. IEEE.
42. Ro, C. (2015). Modeling and analysis of memory virtualization in cloud computing. *Cluster Computing*, 18(1), 177–185.
43. SAForum. (September, 2011). *Service Availability Forum Service Availability Interface—Overview SAI-Overview-B.05.03*. SAForum.
44. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pp. 1–10. IEEE.
45. Szatmári, Z., Kövi, A., & Reitenspiess, M. (2008). Applying mda approach for the sa forum platform. In *Proceedings of the 2nd Workshop on Middleware-Application Interaction: Affiliated with the DisCoTec Federated Conferences 2008*, pp. 19–24. ACM.
46. ASHRAE Technical Committee. (2011). Thermal guidelines for data processing environments expanded data center classes and usage guidance.
47. Toeroe, M., & Tam, F. (2012). *Service availability: principles and practice*. Wiley.
48. Trivedi, K., Sathaye, A., & Ramani, S. Availability modeling in practice.
49. Turner, W. P., PE, J. H., Seader, P. E., & Brill, K. J. (2006). Tier classification define site infrastructure performance. *Uptime Institute*, 17.

Author Biographies

Patricia Takako Endo received her PhD of Computer Science from the Federal University of Pernambuco (UFPE) in 2014. She is a professor at University of Pernambuco (UPE) since 2010; and a researcher at Research in Networks and Telecommunication Group (GPRT) since 2009. Her current research interests are: cloud Computing, and resource management.

Glauro Estácio Gonçalves is a professor at Rural Federal University of Pernambuco (UFRPE) since 2013. He received his Ph.D. degree in Computer Science from the UFPE, respectively in 2007 and 2012. His research interests include: Performance Evaluation of Networked Systems; Cloud Computing; and Optimization Algorithms for Resource Allocation.

Daniel Rosendo is a Master Student in Computer Science at UFPE. He is taking part of the GPRT since 2014. His areas of interests are Software-Defined Networking (SDN), Network Management, and Internet of Things (IoT).

Demis Gomes is a student of Information Systems at UFRPE and member of GPRT since 2015. His current research interests are Cloud Computing, Performance Evaluation, Internet of Things, and Fog Computing.

Guto Leoni Santos is a student of Information Systems at UPE. He is taking part of GPRT and his research interests include: Distributed Systems, cloud Computing, Performance Evaluation, Internet of Things, Smart Cities, Neural Networks, and Artificial Intelligence.

André Luis Cavalcanti Moreira received his PhD of Computer Science from the UFPE. His research topic is in self-organization of cloud networks and adaptation of CDN provisioning algorithms. Currently, he is involved in a research project in a platform for clouds at GPRT.

Judith Kelner received her PhD from the Computing Laboratory at the University of Kent at Canterbury, UK in 1993. She is a Full Professor at UFPE, since 1979. Currently she leads the GRVM team as well as coordinates a number of research projects in the areas of multimedia systems, design of virtual and augmented reality applications, and smart communication devices.

Djamel Sadok received his PhD of Computer Science at the University of Kent at Canterbury, UK in 1990. He is a member of staff at UFPE since 1993. His research interests include communication systems, access networks, security, cloud computing and traffic classification. Currently he leads the GPRT team as well coordinates a number of research projects.

Mozhgan Mahloo works as the researcher in cloud technology department of Ericsson. She holds a PhD degree in Communication systems from KTH Royal Institute of Technology, Sweden. Her research interests lie in the general area of cloud computing and networking as well as business and economic evaluation of such technologies. She is author/co-author of over 12 publications in international journals and conferences as well as one patent application. She has been involved in several national projects as well as FP7 European projects.

Research Advances in Cloud Computing

Chaudhary, S.; Somani, G.; Buyya, R. (Eds.)

2017, XX, 465 p. 126 illus., 81 illus. in color., Hardcover

ISBN: 978-981-10-5025-1