

# Evaluation of a Cloud Computing Environment by different Strategies of Sensitivity Analysis

**Rosangela Maria de Melo**

*Federal Institute of Education, Science, and Technology of Pernambuco (IFPE), Belo Jardim Campus, Brazil and Informatics Center, Federal University of Pernambuco, Recife, Brazil*  
rmm3@cin.ufpe.br  
rosangela.melo@belojardim.ifpe.edu.br

**Maria Clara Bezerra**

*Informatics Center, Federal University of Pernambuco, Recife, Brazil*  
mcsb@cin.ufpe.br

**Rubens Matos**

*Informatics Center, Federal University of Pernambuco, Recife, Brazil*  
rsmj@cin.ufpe.br

**Ivanildo Melo**

*Federal Institute of Education, Science, and Technology of Pernambuco (IFPE), Belo Jardim Campus, Brazil and Informatics Center, Federal University of Pernambuco, Recife, Brazil*  
ijmf@cin.ufpe.br  
ivanildo.melo@belojardim.ifpe.edu.br

**Jamilson Dantas**

*Informatics Center, Federal University of Pernambuco, Recife, Brazil*  
jrd@cin.ufpe.br

**Paulo Maciel**

*Informatics Center, Federal University of Pernambuco, Recife, Brazil*  
prmm@cin.ufpe.br

*Abstract—For several years, cloud computing systems have been generating debate and interest within IT corporations. These cloud computing environments provide storage and processing systems that are adaptable, efficient and simple, thus allowing for rapid infrastructure modifications to be made, according to constantly changing workloads. Organizations of any size and type are migrating to the cloud supporting solutions based on Web. Due to the benefits of the pay-per-use model and scalability factors, services such as Video Streaming. Recent failure events in Video Streaming services have demonstrated the critical importance of maintaining high availability in cloud computing infrastructures. One of the methods used to identify the trends of occurrences of failures in computational systems, occurs through the application of strategies of sensitivity analysis. This paper uses hierarchical modeling and different sensitivity analysis techniques to determine the parameters that cause the greatest impact on the availability of a Video on Demand. The results show that distinct approaches provide similar results regarding the sensitivity ranking, with specific exceptions. A combined evaluation indicates that system availability may be improved effectively by focusing on a reduced set of factors that produce large variation on the measure of interest.*

*Keywords-Availability, Cloud computing, Sensitivity Analysis, Video on Demand.*

## I. INTRODUCTION

Cloud Computing may be defined as a model for enabling on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. This resource set is typically utilized as a service model where in the client only pays for what is consumed. Video on Demand (VoD) streaming services are run on cloud infrastructure platforms in order to offer cost savings, easy scalability, and high availability to users.

In recent years, the streaming service has grown dramatically, with a perception that in 2010 video consumption was responsible for 40% of all internet traffic, To 50% in 2012 [2] expecting to reach 80% to 90% by the end of 2019 [3]. Cloud infrastructures have many interconnected components, which makes managing and administering your re-

sources a complex activity, especially when done manually by your administrators [4]. It is important to note that these are not free from flaws. Recently there have been failures in Amazon AWS [5] where the service rendering of several major companies, including one of the leading companies in the VoD market, has been interrupted, Netflix [5].

To evaluate the availability of the service, hierarchical analytical models are employed to represent the architecture [6]. In this paper, we propose an availability model of a VoD service based on Eucalyptus cloud environment, for evaluating the sensitivity of VoD service components. For guiding the implementation of system improvements, Design of Experiments (DoE), Percentage Difference and Discrete Averaged Sensitivity Index (DASI) were used for identifying availability bottlenecks of the VoD service.

The remainder of the paper is organized as follows: Section II introduces related works on system availability and sensitivity analysis, Section III introduces basic concepts of cloud computing technologies, video streaming, dependability models and analysis and sensitivity analysis. Section IV presents the architecture of the system analyzed in this paper. Section V presents the availability model designed for architecture. Section VI conducts a case study about architecture analysis and presents sensitivity analysis. Finally, Section VII shows the conclusions of the study and suggests possible future work.

## II. RELATED WORKS

Recent research have employed hierarchical modeling to represent cloud computing architectures, it is possible to compare the different solutions and estimations of dependability measures [6],[7]. Moreover, some works have also employed sensitivity analysis to identify the critical system components, and thereby propose infrastructure improvements [8]. In [9], Khzaei *et al.* integrated an availability model in overall analytical sub-models of cloud system. Each sub-model captures a specific aspect of cloud centers. The key performance metrics such as task blocking probability and total delay incurred on user tasks are obtained.

In [10] Choub *et al.* proposed a private cloud, with a

modeling of the cloud computing based upon the Eucalyptus architecture. To understand the behavior of Eucalyptus, it was considered Ubuntu Enterprise Cloud (UEC) as reference architecture for our Cloud Test bed environment in Lab. With UEC architecture, it was addressed the availability of each component of the cloud base on Markov chain through the level analysis of hierarchical available model (HAM). [11] Saif *et al.*, provided a formal analysis modeling and verification of open source state-of-art VM-based cloud management platforms to model and analyze the structural and behavioral properties of the systems have used high-Level Petri Nets. In [12] Araujo *et al.*, an availability model of a digital cloud library through a OpenNebula cloud manager is proposed, using a hierarchical approach to model and evaluate the digital library environment. Measurements were performed to obtain the availability parameters of the library service deployed in a private cloud.

On the other hand, [13] proposed a based parametric sensitivity analysis method applied to the Markov string to determine possible congestion in the availability of a virtualized system. This method was applied to analyze the availability measures of virtual machines associated with failure, recovery and migration of their applications. In [14] investigated hierarchical modeling techniques to evaluate a VoD service in a basic architecture, in addition they used parametric differential sensitivity analysis to identify the critical points of this architecture.

In this paper, the authors proposed availability models applied to a cloud environment for VoD streaming service. Furthermore, Design of Experiments (DoE), Percentage Difference and Discrete Averaged Sensitivity Index (DASI) were applied in order to find the bottlenecks of system availability.

### III. BACKGROUND

This section presents the concepts which provide a background for this paper, including: cloud computing technologies, video streaming, dependability modeling and analysis and sensitivity analysis techniques.

#### A. Cloud Computing and the Eucalyptus Platform

A cloud computing system is comprised of a bundle of resources, such as hardware, software, development platforms and services, readily usable and accessible through the Internet [15]. Services can be provisioned on different levels by cloud computing providers, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In a cloud computing environment, virtual resources can also be dynamically allocated and resized to deal with a varying workload, enabling optimal use of physical resources. Cloud computing services are typically accessed through a pay-per-use model [16] and the provider offers is determined and guaranteed by means of service level agreements.

Eucalyptus is a software architecture based on Linux that leverage the implementation of private and hybrid IaaS clouds. This means that users can utilize their own collections of resources (hardware, storage and network) through a self-service interface according to their needs. The Eucalyptus software framework is modular [17], and consists of five high level components, each with its own Web Service: Cloud Controller (CLC), Cluster Controller (CC), Node Controller (NC), Storage Controller (SC), e Walrus [17].

#### B. Video Streaming

Video streaming is a technology used in the transmission of digital multimedia content on the Internet [18]. Streaming enables

data to be sent and flow without the need to wait for content to load completely. This requires smaller network bandwidth and less storage space. As the multimedia data arrives, it is stored in a fast buffer before execution starts. Encoding, protocols and buffering mechanism are factors that might affect transmission of a video streaming [19]. The video streaming service works with the protocol Real Time Streaming Protocol (RTSP) which allows the control in the data transfer with real time properties. The RTSP makes the transfer possible on demand of real time data with audio and video [20].

#### C. Models for Dependability Analysis

In early 1980s Laprie coined the term dependability for encompassing concepts such reliability, availability, safety, confidentiality, maintainability, security and integrity etc [21], [22], whereas reliability of system at  $t$  is the probability that the system performs its functions without failing up to time instant  $t$ . Availability can be expressed as the ratio of the expected system uptime to the expected system up and downtime:  $A = E[Uptime] / (E[Uptime] + E[Downtime])$  It may also be represented by:  $A = MTTF / (MTTF + MTTR)$  where MTTF and MTTR is the mean time to failure and to recovery, respectively.

There are several types of models that can be used for analytical evaluation of dependability. Reliability Block Diagrams (RBD), Fault Trees, Stochastic Petri Nets (SPN) and Continuous Time Markov Chains (CTMC) have been used to model fault-tolerant systems and evaluate various dependability measures.

#### D. Sensitivity Analysis

The Design of Experiments, known as DoE, is a method used to perform sensitivity analysis. Through this method it is possible to assess the importance of each parameter of the system and furthermore it can be used to simultaneously determine the individual and interactive effects of many factors that may affect the output measures. In the DoE method, parameters are called factors and the value that is assigned to each factor is called a level. There are numerous types of DoE and among the most commonly used are: full factorial design, fractional factorial design and simple design [23]. Sometimes the number of experiments required for a full factorial design is too large. This may happen if either the number of factors or their levels is large. It may not be possible to use a full factorial design due to the cost or the time required. In such cases, one can only use a fraction of the full factorial design [24]. Fractional factorial designs are commonly used to reduce the number of runs required to build an experiment [24] In this study we adopted the fractional factorial design, because there are more than five variables that we need to analyze. For 2k designs with five or more variables consider using the fractional factorial designs to reduce the number of runs required by the experiment.

Other method to determine parameter sensitivity is calculating Discrete Averaged Sensitivity Index (DASI) is the average value of the partial derivatives computed throughout a discrete range of values for the parameter of interest. DASI proposes to reduce dependency associated to a single parameter in the partial derivative method enabling to build unified ranking sensitivity accurately. This sensitivity index can be obtained through Equation 1:

$$DASI_{\theta}(Y) = \frac{\sum_{i=1}^n \frac{\partial Y}{\partial \theta} |_{\theta=\theta_i}}{n} \quad (1)$$

Specific methods for performing the differential sensitivity analysis in analytic models are needed when there is no direct closed-form equations for computing the measure of interest. When dealing with hierarchical or composite models, the analysis needs to consider all parameters from each model, determining their impact to the global measure of interest [25],[26], [27]. Partial derivatives are an important for performing sensitivity analysis, which enables conclusions focused on the single point that corresponds to the configuration of parameter values used. The DASI strategy calculates the partial derivative at various points, eliminating dependence on analyzing the behavior of the system only at a single configuration of parameters values.

IV. SYSTEM ARCHITECTURE

The system architecture is based on Eucalyptus cloud computing platform. A broad view of the components of the VoD service architecture is seen in Figure 1 [28].

The VoD architecture is divided into two sides: client and the server. The physical structure is composed of three machines. One machine is used for frontend and two machines for the nodes. The client connects to the video streaming server through the Internet. A storage volume is allocated in the frontend for storing the collection of videos. A Virtual Machine (VM), running the Apache and VLC applications, is instantiated in the nodes. VLC provides the video streaming features, whereas Apache is responsible for hosting the service on a dedicated Web page. The user issues a request for displaying a video hosted on a specific web page. VLC, in turn, grabs the requested video from the remote storage volume and transmits the stream to the user.

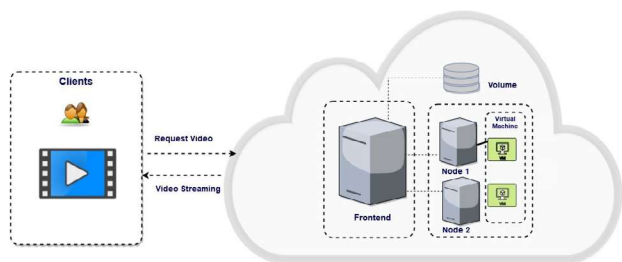


Figure 1: VoD Achitecture

V. AVAILABILITY MODELS

This section discusses the availability models employed to represent the redundant architecture evaluated by this re- search, from which the availability values were calculated. Such values were obtained by the hierarchical combinatorial method, which combines the system state representation of Markov chains with RBDs [29], and is the method commonly employed to evaluate complex IT systems. The systems were modeled and evaluated with the SHARPE [30] and Mercury [31] tools, which were specifically designed for the analysis of such models.

A. Model for Architecture

RBD and CTMC models were used to represent the subsystems of the architectures presented in Figure 1. These models are then combined, constituting a hierarchical model [28]. The architecture can be divided into three parts: Frontend, Node and Service, which are represented by the RBD model depicted in Figure 2. In the Figure 2, we made a union the service and nodes subsystem to represent the components of the architecture of the Figure 1. This junction was performed because the dependece which exists between the service and nodes subsystem.



Figure 2: RBD Architecture

Volume subsystem is allocated in the frontend for storing the collection of videos. The Service subsystem is further refined by a CTMC (see Figure 3), which allows to compute availability values to be considered top-level in the RBD. A CTMC was proposed due to the interdependency between the system's components.

In the top-level model the service as well as the node subsystem infrastructure is represented by the Service RBD block. However the availability of such a sub-system (service + node subsystem infrastructure) cannot be properly represented by an RBD since the node subsystem implements an active redundant mechanism.

Therefore, the Service RBD block is refined by the CTMC depicted in Figure 3, which represents the service availability of the node subsystem infrastructure. The CTMC comprises the states UUU, UDU, UUD, and UUU (service available), and the states DDW, DUW, DDD, DWU, DDU, DWD, and DUD (service unavailable).

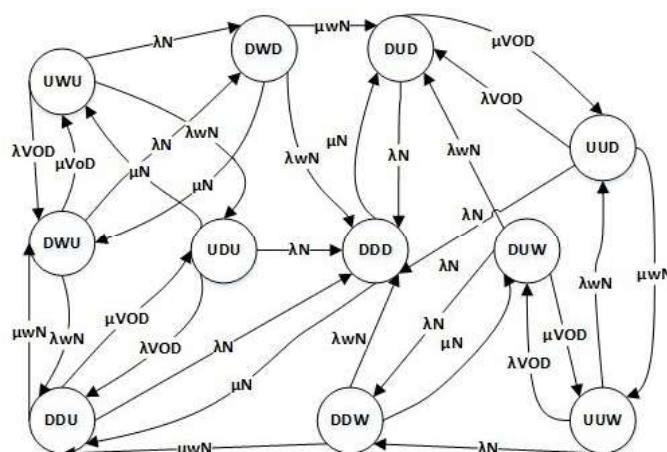


Figure 3: Service module CTMC.

The notation for the states is based on the current condition of each component. The three letters represent initialisms of the operating condition of the three components, respectively, the service, the first node, and the second node. The service may be up (U) or down (D). The NCs work by being alternately in warm standby mode, and only one of them should be up (U) at any one time, whilst the other is either in warm standby (W) or down (D). In this model the initial service is represented by UUU, where the service is available, the first node in warm standby, and the second node is running. From this state it is possible to move to DUU (service failure), DUU (second node failure), or UDU (first node failure). From the DUU state (service down, first node in warm standby and second node up), may be reached UUU (representing service repair), DUU (second node failure), or DDU (first node failure).

From state DUU three outcomes are possible; either the failure of all system components (DDD), the initialization of the first node (DDU) or the repair of the second node (DUU). From state UDU (service and second node running), the possible outcomes are DDD (failure of all components), DDU, or UUU. The state UDU can lead to either the failure



of all system components (DDD), the repair to waiting state of the first node (DWU), or the instantiation of a new virtual machine with all system applications, making the service available again (UDU).

In state DDD all system components are down; the service is unavailable and the two nodes are unavailable. From this state it is possible to reach two other states; repair of the first node (DUD) and the repair of the second node (DDU). State DUD represents service unavailability, where service and second node are down, but the first node is up. From this state, three other states can be achieved; failure of all components of the system (DDD), repair of the service (UUD), or repair of the warm standby mode of the second node (DUW). Conversely, state UUD indicates system availability, where the service and the first node are up, but the second node is faulty. From here, the following three states can be reached; DUD (service failure), UUD (repair of the node to warm standby mode), and DDD (since failure of the only functional mode will automatically cause service failure too).

In state DUW the system is unavailable due to service application failure, although the first node is up and the second node is in warm standby mode. From DUW the following states can be reached; DUD (failure of the warm standby node), DDW (first node failure), or UUD (service repair). With service and first node operational, and the second node in warm standby, UUD indicates system availability. From this position in the model, the possibilities are warm standby failure (UUD), service failure (DUW) or first node failure, which would cause the service to become unavailable (DDW). The state DDW indicates system unavailability, with service and first node down, and second node in warm standby. From DDW, it is possible to reach three other states; failure of all the components of the system (DDD), initialization of the second node (DDU), or repair of the first node (DUW).

System failure is an event that occurs when the provided service deviates from the intended service [31], [32]. The failure rates of the two nodes are represented by  $\lambda_N$ , whilst  $\mu_N$  represents the rate of node repair. A node in warm standby has the failure rate of  $\lambda_{WN}$ , and the repair rate to return it to standby is  $\mu_N$ . A warm standby node is transformed to available mode at the rate of  $\mu_{WN}$ . The failure rate of the service application is  $\lambda_{VoD}$ , while the repair rate is  $\mu_{VoD}$ . The  $\lambda_{VoD}$  was obtained from the inverse of the time to failure of the service module. To calculate this result, we used the CTMC model of Figure 3 and the Mercury tool [31], [32]. The repair rate of the service is considered as the instantiation of a new virtual machine, including all the applications necessary to its operation (Apache and VLC).

Through the CTMC of figure 3 it is possible to obtain the closed-form equation for calculating the availability of the redundant model ( $A_{VoD}$ ) as shown below in equation 2. The equation for calculating the availability of the whole infrastructure of the service ( $A_{VoD}$ ) can also be obtained from its corresponding RBD model [6].

A closed-form equation for computing the availability of the complete service ( $A_{Service}$ ) can also be obtained, as demonstrated by Equation 3.  $A_f$  (availability of the frontend) and  $A_v$  (availability of the volume) can be computed from the RBD of Figure 2, whilst  $A_{VoD}$  is calculated from equation 2. In this equation,  $A_f$ ,  $A_v$ , and  $A_{VoD}$  correspond to the availability of the frontend, volume, and service, respectively.

$$A_{VoD} = \frac{\alpha(2\beta^2\alpha_1 + \beta\alpha_2\beta_1 + \alpha_3)\beta_1^2 + \alpha_4\beta_1^3}{\alpha_5(\beta^2\alpha_1(\lambda_N + 2\mu_N) + \beta\varphi\beta_1 + \varphi_1\beta_1^2 + \beta_7\beta^3)}, \quad (2)$$

Where

$$\begin{aligned} \beta &= \lambda_{WN}, \beta_1 = \mu_{WN}, \alpha = \mu_N\mu_{VoD}, \\ \alpha_1 &= (\lambda_N + \beta + \mu_N)^2(\lambda_N + \beta + \mu_N), \\ \alpha_2 &= 4\lambda_{N^2} + 17\lambda_N\beta + 13\beta + 5\lambda\mu_N + 12\beta\mu_N + 2\mu_{N^2}, \\ \alpha_3 &= (8\beta(2\beta + \mu_N) + \lambda_N(11\beta + \mu_N)), \\ \alpha_4 &= (7\beta + 2\mu_N)(\mu_{WN^3}), \alpha_5 = \lambda_N + \lambda_{VoD} + \mu_{VoD}, \\ \alpha_6 &= 2\lambda_N(\lambda_N + \beta)(\lambda_N + 3\beta), \\ \beta_2 &= (8\lambda_{N^2} + 24\lambda_N\lambda_{WN} + 13\beta^2)\mu_N, \\ \beta_3 &= (7\lambda_N + 12\beta)\mu_{N^2} + 2\mu_{N^3}, \\ \beta_4 &= 2\lambda_N\beta(2\lambda_N + 3\beta), \\ \beta_5 &= (\lambda_N + \beta)(\lambda_N + 16\beta)\mu_N, \\ \beta_6 &= (\lambda_N + 8\beta)\mu_{N^2}, \\ \beta_7 &= (\lambda_N(\beta + 2\mu_N) + \mu_N(7\beta + 2\mu_N)) \\ \varphi &= \alpha_6 + \beta_2 + \beta_3, \text{ and} \\ \varphi_1 &= \beta_4 + \beta_5 + \beta_6. \end{aligned}$$

$$A_{Service} = A_f \times A_v \times A_{VoD} \quad (3)$$

## VI. CASE STUDIES

Four case studies were designed to analyze the system availability. The case studies can be summarized as follows:

- i) Case Study I: availability analysis of architecture.
- ii) Case Study II: sensitivity analysis of all the system components (see Figure 2) to establish a ranking of the most important parameters in the video streaming service.
- iii) Case Study III: analysis the behavior of the system using other technique of sensitivity analysis, by means of a percentage difference.
- iv) Case Study IV: analysis the behavior of the system using technique of sensitivity analysis, DASI.

### A. Case Study I

Figure 1 depicts the architecture, which has the dedicated frontend machine and two machines for the nodes. The Table I shows the values of mean time to failure (MTTF) and mean time to repair (MTTR) used the model of the Frontend. Those values were obtained from [29], [6] and were used to compute the dependability metrics for the frontend, and subsequently for the whole system. This subsystem has a MTTF of 180.72 h and a MTTR of 0.96999 h. The values used of MTTF and MTTR for Volume subsystem are described in the Table I. Those values were obtained from [29]. Table I presents the parameters for the blocks of the RBD model for the structure shown in the Figure 2. The values of MTTF and MTTR of the Frontend and volume modules are based on [6] and [29]. The availability of the Service module is computed from the CTMC depicted in Figure 3. Table II presents all values of the parameters used for computing the availability of the Service module. The values are based on the analyses shown in [6] and [29].

Component	MTTF	MTTR
Frontend	180.72 h	0.96999 h
Volume	100000 h	1 h
Service	217.77 h	0.92633 h

Parameter	Description	Value ( $h^{-1}$ )
$\lambda_N$	Mean Time to Node Failure	1/481.83
$\lambda_{wN}$	Mean Time to Standby Node Failure	1/578.196
$\mu_N$	Mean Time to Node Repair	1/0.91
$\mu_{wN}$	Mean Time to Standby Node Repair	1/0.0333
$\lambda_{VoD}$	Mean Time to Service Failure	1/217.779
$\mu_{VoD}$	Mean Time to Service Instantiate	1/0.0275

For the presented configuration of parameters, we find a value of 0.994401 for the availability of the video streaming system, without redundancy. This availability corresponds to about 49,05 hours of downtime in a year, and therefore highlights the importance of searching for effective solutions to improve this system.

**B. Case Study II**

For 2k designs with five or more variables consider using the fractional factorial designs to reduce the number of runs required by the experiment [33]. This way, we performed the analysis of a fractional factorial design of experiments to provide point of view on the sensitivity of Video Stream Service availability with respect to each parameter. This analysis was performed on the 10 parameters shown in the ranking based on partial derivatives. Two levels were considered for each parameter: the minimum and maximum values used in the graphical analysis. This 2k factorial experiment was evaluated according to the individual effects for the system availability, and these values are shown in Table IV. The measures of interest were calculated with the values given in Table I and III, and the sensitivity ranking of availability for all parameters of the streaming service were given in Table IV. The tools Minitab was used for calculated sensitivity analysis [34].

Metrics	Parameters	Minimum Value (h)	Maximum Value (h)
MTTF	$\lambda_f$	180.72	4380
	$\lambda_{VoD}$	217.779	4380
	$\lambda_{Vol}$	10000	110000
	$\lambda_N$	481.83	4380
	$\lambda_{wN}$	578.196	4380
	MTTR	$\mu_{Vol}$	1
$\mu_f$		0.96999	2
$\mu_{VoD}$		0.0275	0.083333
$\mu_{wN}$		0.0333	0.083333
$\mu_N$		0.091	2

The results are ordered according to absolute values. Negative values indicate that there is an inverse relationship between parameters and system availability. For example, the sensitivity with respect to failure rates is negative due to the fact that as failure rate increases (that is, the MTTF decreases) availability decreases. The index in Table IV indicates that the parameters  $\lambda_f$ ,  $\lambda_{wN}$  and  $\mu_f$  have the largest effect values. Based on the results it can be concluded that the Frontend is the critical point of the video streaming system in terms of availability, and should therefore receive priority when improvements to the system are considered. Table IV also demonstrates that the rate  $\lambda_N$ , which is the rate of repair the node, is shown to have the second least impact on system availability, with only  $\mu_N$  having less. The ranking obtained from design of experiments sensitivity analysis provides a direct view of the order of importance of all parameters.

Figure 4 is a graphic representation of the system availability where the MTTF parameters for Frontend were altered one at a time, whilst fixing all other parameters to the values given in Tables I and II. The plot confirms that increasing the failure times of the Frontend modules results in increased availability.

The need to implement improvements in the Frontend modules is confirmed by the ranking given in Table IV. An alternative method for reaching this conclusion would be to implement failover mechanisms in one or both of the components and evaluate on impact on system availability.

TABLE IV: SENSITIVITY RANKING FROM 2K EXPERIMENT ANALYSIS.

Parameter	Effect
$\lambda_f$	0.009044
$\lambda_{wN}$	0.003580
$\mu_f$	-0.002920
$\mu_{wN}$	-0.001342
$\mu_{Vol}$	-0.000905
$\lambda_{Vol}$	0.000896
$\mu_{VoD}$	0.000895
$\lambda_{VoD}$	-0.000447
$\lambda_N$	-0.000447
$\mu_N$	-0.000001

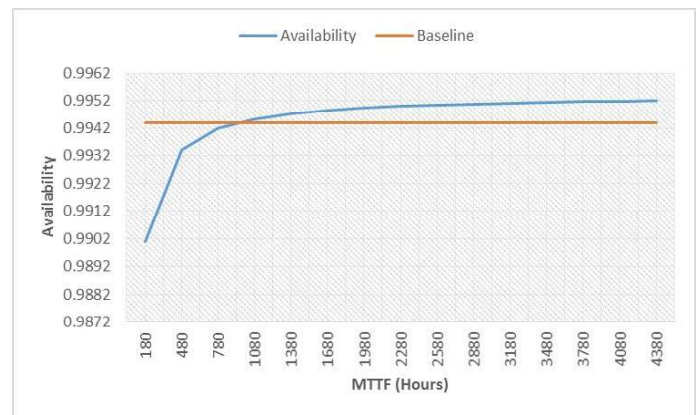


Figure 4: Effect of MTTF on system availability.

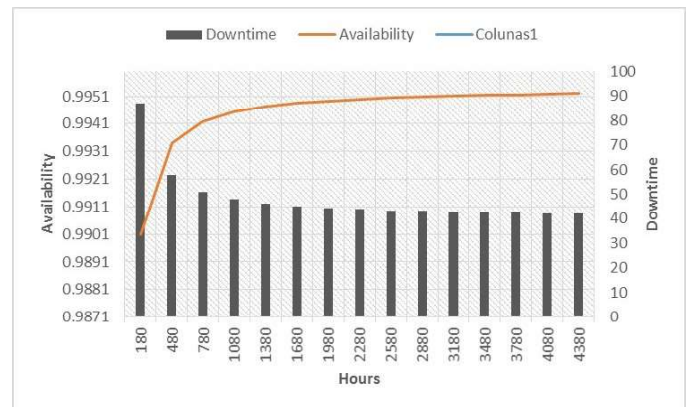


Figure 5: Availability versus Downtime

Figure 5 illustrates the system availability as a function of the MTTF for the Frontend module, and clearly demonstrates the impact that MTTF increases have on system availability. Increasing the MTTF of the Frontend results in a reduction in downtime of 44.80 hours in the year.

**C. Case Study III**

In this case study analysis the behavior of the system using another technique of sensitivity analysis. In this case we used the percentage difference with this method we can also identify components that affect the availability of the systems.

The percentage difference is calculated using a Equation 4. This equation shows the expression for this approach, where  $\max \{Y(\theta_i)\}$  and  $\min \{Y(\theta_i)\}$  are the maximum and minimum output values, respectively, computed when varying the parameter  $\theta$  over the range of its n possible values of



interest. If  $Y(\theta_i)$  is known to vary monotonically, so only the extreme values of  $\theta$  (i.e.,  $\theta_i$  and  $\theta_n$ ) may be used to compute  $\max\{Y(\theta_i)\}$  and  $\min\{Y(\theta_i)\}$  subsequently  $S_{\theta_i}\{Y\}$  [23].

$$S_{\theta_i}\{Y\} = \frac{\max\{Y(\theta_i)\} - \min\{Y(\theta_i)\}}{\max\{Y(\theta_i)\}} \quad (4)$$

Table V presents the ranking of the sensitivity analysis in descending order. In order to generate the results of this ranking we used the input data from Table III and applied in Equation (4). Table V gives the percentage difference of all system components calculated from the input parameter values given in Table III.

TABLE V: SENSITIVITY RANKING FROM PERCENTAGE DIFFE

Parameter	SI (A)
$\mu_f$	0.00563709501
$\lambda_f$	-0.00514476562
$\mu_{VoD}$	0.00037202503
$\mu_{Vol}$	0.00018996201
$\lambda_{VoD}$	-0.00011998807
$\lambda_N$	-0.00009585142
$\mu_{wN}$	0.00003967628
$\lambda_N$	0.00002953291
$\lambda_{Vol}$	-0.00000090908
$\mu_{wN}$	-0.0000001800

The Table V indicates that the repair and rate Frontend ( $\mu_f$ ) and ( $\lambda_f$ ) module are the most important component with respect to availability.

A refined analysis combining the two rankings— $2^k$  DoE and percentage difference indices may provide a reduced list of parameters which deserve the highest priority to improve the system availability. We perform such a combined analysis by checking the parameters which appear among the first three positions of the rankings. The parameters which match such a criterion are:  $\lambda_f$ ,  $\mu_f$ ,  $\mu_{VoD}$  and  $\mu_{Vol}$ .

#### D. Case Study IV

In this case study, applied another technique of sensitivity analysis to evaluate the behavior of the system. In this case we used the DASI with this method we can also identify components that affect the availability of the systems.

The sensitivity ranking was obtained by DASI strategy through Equation 5, for the models of the Figures 2 and 3.

$$DASI_{\theta_i}^*(A_{S2}) = \frac{\sum_{i=1}^n \frac{\partial A_f}{\partial \theta} \left(\frac{\theta}{A_{S2}}\right) |_{\theta=\theta_i}}{n} \times A_v \times A_{VoD2} + \quad (5)$$

$$A_f \times \frac{\sum_{i=1}^n \frac{\partial A_v}{\partial \theta} \left(\frac{\theta}{A_{S2}}\right) |_{\theta=\theta_i}}{n} \times A_{VoD2} +$$

$$A_f \times A_v \times \frac{\sum_{i=1}^n \frac{\partial A_{VoD2}}{\partial \theta} \left(\frac{\theta}{A_{S2}}\right) |_{\theta=\theta_i}}{n}$$

The expressions of the partial derivatives for the subsystem  $A_f$ ,  $A_v$  are listed below.

$$\frac{\partial A_f}{\partial \lambda_f} = -\frac{\mu_f}{(\mu_f + \lambda_f)^2} \quad (6)$$

$$\frac{\partial A_f}{\partial \mu_f} = -\frac{\mu_f}{(\lambda_f + \mu_f)^2} + \frac{1}{\lambda_f + \mu_f} \quad (7)$$

The partial derivatives for the subsystem AVoD2 with respect to each of the parameters ( $\frac{\partial A_{VoD2}}{\partial \mu_{VoD}}$ ,  $\frac{\partial A_{VoD2}}{\partial \lambda_{VoD}}$ ,  $\frac{\partial A_{VoD2}}{\partial \lambda_N}$ ,  $\frac{\partial A_{VoD2}}{\partial \mu_{wN}}$ ,  $\frac{\partial A_{VoD2}}{\partial \lambda_{wN}}$  e  $\frac{\partial A_{VoD2}}{\partial \mu_{wN}}$ ) of the CTMC model does not require detail because it is not the purpose of the analysis. In addition, the equations are complex.

In the Table VI, we present the ranking of the sensitivity analysis in descending order for the DASI strategy, in order to generate the results of this classification.

TABLE VI: RANKING OF THE SENSITIVITY BY DASI ESTRATEGY

Parameter	$DASI_{\theta_i}^*(A_{S2})$
$\mu_f$	0.005861225208
$\lambda_f$	0.005331802213
$\mu_n$	0.002049195663
$\lambda_n$	0.001932307554
$\mu_{VoD}$	0.000412384957
$\lambda_{VoD}$	0.000126247799
$\mu_{vol}$	0.000017099727
$\lambda_{vol}$	0.000009999877
$\lambda_{wN}$	0.000008896625
$\mu_{wN}$	0.000000117839

We identified at the top of the ranking of the sensitivity analysis for the DASI strategy the components  $\mu_f$ ,  $\lambda_f$ ,  $\mu_n$  as the main components candidates for actions of system improvements. On the other hand, the component  $\mu_{wN}$  is of less importance as pointed out by this strategy. This analysis gives us the possibility of not calculating the sensitivity strategy for only one specific configuration parameter, which allows us the flexibility of the value variation. This provides a better response to system availability.

Each strategy of sensitivity analysis can obtain a different ranking, the reason for which we use for evaluation of the computational systems, more than one strategy, in order to obtain high reliability of these systems. That way, as we apply more than one strategy and relate their results, we have the opportunity to capture more events.

## VII. CONCLUSION

This paper presented sensitivity analysis of VoD service availability based on hierarchical analytical models and distinct techniques to assess the impact of each input parameter. The results show that the system availability may be improved effectively by focusing on a reduced set of factors which produce large variation on steady-state availability. Most parameters ranked in the highest positions of sensitivity rankings is related to the frontend component.

This component should receive the highest priority to achieve effective improvements on system availability. We also performed a combined analysis from the three techniques which is useful to reduce the list of parameters which an analyst should focus on, and solve possible conflicting results from distinct approaches. For future work, the authors propose to implement further sensitivity techniques, and also to extend the scope of the models considered in different topologies.

## REFERENCES

- [1] N. I. of Standart and Technology's, "Cloud computing," 2013, available on <http://www.nist.gov>.
- [2] I.-S. Hwang, A. Nikoukar, C.-H. Teng, and K. R. Lai, "Scalable architecture for vod service enhancement based on a cache scheme in an ethernet passive optical network," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 5, no. 4, pp. 271–282, 2013.
- [3] C.V.N.Index, "Forecast and methodology, 2010–2015, June 1, 2011,"

- 2015, disponível em: [http://www.abed.org.br/censoead2014/CensoEAD2014\\_portugues.pdf](http://www.abed.org.br/censoead2014/CensoEAD2014_portugues.pdf). Acessado em: 3 de out 2016.
- [4] C. Taurion, "Cloud computing: Computação em nuvem: Transformando o mundo da tecnologia da informac,a"o," Rio de Janeiro: Brasport, p. 2, 2009.
- [5] Netflix, "Lessons netflix learned from the aws outage," 2014, disponível em <http://techblog.netflix.com/2011/04/lessons-netflix-learned-from-aws-outage.html> Acessado em: 13 dez. 2016.
- [6] J. Dantas, R. Matos, J. Araujo, and P. Maciel, "An availability model for eucalyptus platform: An analysis of warm-standby replication mechanism," in Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on. IEEE, 2012, pp. 1664–1669.
- [7] S. Chuob, M. Pokharel, and J. S. Park, "Modeling and analysis of cloud computing availability based on eucalyptus platform for e-government data center," in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on. IEEE, 2011, pp. 289–296.
- [8] R. Matos, P. R. M. Maciel, F. Machida, D. S. Kim, and K. S. Trivedi, "Sensitivity analysis of server virtualized system availability," Reliability, IEEE Transactions on, vol. 61, no. 4, pp. 994–1006, 2012.
- [9] H. Khazaei, J. Mistic, V. B. Mistic, and N. B. Mohammadi, "Availability analysis of cloud computing centers," in Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE, 2012, pp. 1957–1962.
- [10] S. Chuob, M. Pokharel, and J. S. Park, "Modeling and analysis of cloud computing availability based on eucalyptus platform for e-government data center," in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on. IEEE, 2011, pp. 289–296.
- [11] S. Malik, S. Khan, and S. Srinivasan, "Modeling and analysis of state-of-the-art vm-based cloud management platforms," 2013.
- [12] M. M. G. C. Julian Araujo, Paulo Maciel and E. Andrade, "Availability evaluation of digital library cloud services," The Fourth International Workshop on Dependability of Clouds, Data Centers and Virtual Machine Technology.
- [13] R. Matos, P. R. M. Maciel, F. Machida, D. S. Kim, and K. S. Trivedi, "Sensitivity analysis of server virtualized system availability," Reliability, IEEE Transactions on, vol. 61, no. 4, pp. 994–1006, 2012.
- [14] M. C. Bezerra, R. Melo, J. Dantas, P. Maciel, and F. Vieira, "Availability modeling and analysis of a vod service for eucalyptus platform," in 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2014, pp. 3779–3784.
- [15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
- [16] F. C. Cardoso, "Concepts of virtual private network for secure streaming video," 2010.
- [17] "Eucalyptus cloud computing platform - administrator guide," 2010, technical report, Eucalyptus Systems, Inc., Version 2.0.
- [18] G. D. Delgado, V. C. Frías, and M. A. Igartua, "Video-streaming transmission with qos over cross-layered ad hoc networks," in Software in Telecommunications and Computer Networks, 2006. SoftCOM 2006. International Conference on. IEEE, 2006, pp. 102–106.
- [19] D. Diaz-Sanchez, F. Almenarez, A. Marín, D. Proserpio, and P. Arias Cabarcos, "Media cloud: an open cloud computing middleware for content management," Consumer Electronics, IEEE Transactions on, vol. 57, no. 2, pp. 970–978, 2011.
- [20] A. M. A. Valeriana and R. Marcelo, "Monitoramento do protocolo rtsp (real time streaming protocol) utilizando ntop (network top)," Centro Brasileiro de Pesquisas Físicas, 2008.
- [21] J. Laprie, Dependability: Basic Concepts and Terminology. Springer-Verlag, 1992.
- [22] J.-C. Laprie, "Dependable computing and fault tolerance: Concepts terminology," in Fault-Tolerant Computing, Twenty-Fifth International Symposium on. IEEE, 1995, pp. 2–11.
- [23] R. Jain., "The art of computer systems performance analysis: Techniques for experimental measurement, simulation and modeling," Wiley India Pvt. Ltd., 1991.
- [24] P. G. Mathews., "Design of experiment with minitab," William A. Tony, EUA., 2004.
- [25] D. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," Environmental monitoring and assessment, vol. 32, no. 2, pp. 135–154, 1994.
- [26] R. d. S. Matos, P. R. Maciel, F. Machida, D. S. Kim, and K. S. Trivedi, "Sensitivity analysis of server virtualized system availability," IEEE Transactions on Reliability, vol. 61, no. 4, pp. 994–1006, 2012.
- [27] R. de Souza Matos Júnior, "An automated approach for systems performance and dependability improvement through sensitivity analysis of markov chains," 2011.
- [28] R. Melo, M. C. Bezerra, J. Dantas, R. Matos, I. J. de Melo Filho, A. S. Oliveira, F. D. de Oliveira Feliciano, and P. R. M. Maciel, "Sensitivity analysis techniques applied in cloud computing environments," in Information Systems and Technologies (CISTI), 2017 12th Iberian Conference on. IEEE, 2017, pp. 1–7.
- [29] D. S. Kim, F. Machida, and K. S. Trivedi, "Availability modeling and analysis of a virtualized system," in Dependable Computing, 2009. PRDC'09. 15th IEEE Pacific Rim International Symposium on. IEEE, 2009, pp. 365–371.
- [30] R. A. Sahner and K. S. Trivedi, "Reliability modeling using sharpe," Reliability, IEEE Transactions on, vol. 36, no. 2, pp. 186–193, 1987.
- [31] B. Silva, G. Callou, E. A. G. Tavares, P. Maciel, J. C. de Figueiredo, E. Souza, C. J. M. Araujo, F. Magnani, and F. A. S. Neves, "Astro: An integrated environment for dependability and sustainability evaluation," Sustainable Computing: Informatics and Systems, vol. 2, pp. 1–31, 2012.
- [32] "Mercury tool developed by modcs group. <https://sites.google.com/site/mercurytool/download/>."
- [33] P. G. Mathews, Design of Experiments with MINITAB. ASQ Quality Press, 2005.
- [34] I. Minitab, "Minitab statistical software," Minitab Release, vol. 13, 2000.