VoD in Eucalyptus Platform: Availability Modeling and Sensibility Analysis

Rosangela Melo*†, Maria Clara Bezerra*, Jamilson Dantas*, Rubens Matos*, Ivanildo Melo Filho*† and Paulo Maciel*

*Informatics Center, Federal University of Pernambuco (UFPE), Recife, Brazil

†Federal Institute of Education, Science, and Technology of Pernambuco (IFPE), Belo Jardim, Campus - Brazil

{rmm3, mcsb, jrd, rsmj,ijmf, prmm}@cin.ufpe.br*, {rosangela.melo,ivanildo.melo}@belojardim.ifpe.edu.br†

Abstract—Due to advantages such as the pay-per-use model and a highly scalable environment, cloud infrastructures are commonly employed by Video on Demand (VoD) streaming services to supply an extensive range of multimedia content to users. However, recent events have demonstrated to commercial VoD service providers the fundamental importance of ensuring highly available cloud computing infrastructures. This paper examines the effectiveness of applying availability modeling to a VoD cloud environment which has no redundancy mechanisms included in its architecture. The analysis employs a combinatorial modeling strategy consisting of RDB models and Markov chains. Sensitivity analysis is also applied to identify the critical points in the system with respect to availability. The outcome of the research clearly shows that the modeling strategy, together with the sensitivity analysis, has a key role to play in the identification of those components on which resources should be focused in order to achieve increased system availability

Keywords-Markov Chains; Availability; Sensitivity Analysis; RDB models; Video Streaming Service.

I. INTRODUCTION

Most popular online services, such as social networks and video streaming providers, employ cloud computing in order to ensure availability and proper service delivery to users. Current Video on Demand (VoD) streaming services rely heavily on benefits offered by cloud computing, such as the pay-per-use model and high scalability [1]. Video streaming is a key element in an increasing number of telecommunications services such as video conferencing, video on demand, and live Internet TV. The ability to measure service availability and identify the most important components of the service is highly advantageous when initiatives are taken to increase system reliability [2].

Analytical modeling techniques and sensibility analysis may be employed to predict several metrics useful to the decision making process concerning system infrastructure and the attainment of required availability levels [3], [4]. Various works in recent years have explored the evaluation of availability in cloud systems. Matos *et al.* [3] investigated the availability of data networks including redundancy mechanisms. Certain scenarios were evaluated through the analytic-numeric solution of Markov chains. Furthermore, the impact of different component parameters on overall system availability was evaluated by means of differential sensitivity analysis. In another study, Khazaei *et al.* [2] integrated an availability model into the overall analytical submodels of a cloud system. Each submodel captured a specific aspect of cloud centers. The study was able to calculate key performance metrics such as task blocking probability and total delay incurred for user tasks.

This paper proposes an analytical availability model to support the evaluation of a VoD streaming service running in a private cloud. The modeling strategy considers a combinatorial approach comprising Reliability Block Diagrams (RBDs) and Continuous Time Markov Chains (CTMCs). A parametric sensitivity analysis of the proposed model is also suggested, which allows for the identification of availability bottlenecks, thereby providing a guide to the implementation of system improvements. The remainder of the paper is organized as follows: Section II introduces basic concepts, Section III presents the architecture of the system analyzed in this paper. Section IV presents the availability models designed for architectures. Section V discusses a case study conducted, which highlights the applicability of the proposed model. Finally, Section VI shows the conclusions of the study and suggests possible future work.

II. PRELIMINARIES

This section presents the background concepts to this paper, including: cloud computing technologies, video streaming, dependability modeling 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. 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). Eucalyptus is a software architecture based on Linux that leverage the implementation of private and hybrid IaaS clouds. The Eucalyptus software framework is modular [5], 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 [5].

B. Video Streaming

Video streaming is the technology employed for the transmission of digital multimedia content instantaneously on the Internet [1]. Streaming enables content to be sent and viewed

without requiring the data to be completely downloaded locally before it can be consumed. Hence, less network bandwidth is taken up and less storage space is required. As the multimedia data arrives, it is stored in a fast buffer prior to execution [1]. The video streaming service is run with the Real Time Streaming Protocol (RTSP) which manages the data transfer in real time.

C. Models for Dependability Analysis

Dependability is related to disciplines such as fault tolerance and reliability. The concept of dependable computing first appeared in 1820s when Charles Babbage undertook the enterprise to conceive and construct a mechanical calculating engine to eliminate the risk of human errors, [6]. In early 1980s Laprie coined the term dependability for encompassing concepts such reliability, availability, safety, confidentiality, maintainability, security and integrity etc [7], 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: $A = E[Uptime]/(E[Uptime] + E[Downtime])$.

D. Sensitivity Analysis

Sensitivity analysis is a technique used to determine the factors that are most relevant regarding the measures or output of a model [8]. There are several ways to carry out a sensitivity analysis. The differential analysis was chosen for this work because it can be efficiently performed in analytical models usually employed in availability and performance studies. The differential analysis is accomplished by calculating the partial derivatives for the measures of interest the respective parameters of interest. For example, considering a metric Y which depends on a parameter (λ), the sensitivity of Y with respect to λ is computed with Equation $S_\lambda(Y) = \frac{\partial Y}{\partial \lambda}$, or $S_\lambda^*(Y) = \frac{\partial Y}{\partial \lambda} \left(\frac{\lambda}{Y}\right)$ when adopting scaled sensitivity [9]. Other scaling methods can be used, depending on the nature of the parameters, measure of interest and the need for removing the effects of units [3]. $S\lambda(Y)$ and $S_\lambda^*(Y)$ are also referred to as sensitivity coefficients [8], whose ordered values produce the ranking that is used to compare the degree of influence among all parameters. The SHARPE tool [10] was used to aid the the analysis presented in this work, providing the sensitivity indices of CTMCs. These indices are used in sensitivity functions for top-level RBD models, derived in the Mercury tool [11].

III. SYSTEM ARCHITECTURE

The system architecture is based on the Eucalyptus cloud computing platform. Figure 1 illustrates a general view of the main system components. On the server side the physical structure is composed of two machines; one machine is the Frontend and the other is the Node. The client connects to the video streaming server through the Internet, and a storage volume is allocated in the Frontend for storing the videos. A Virtual Machine (VM), running the Apache and VLC
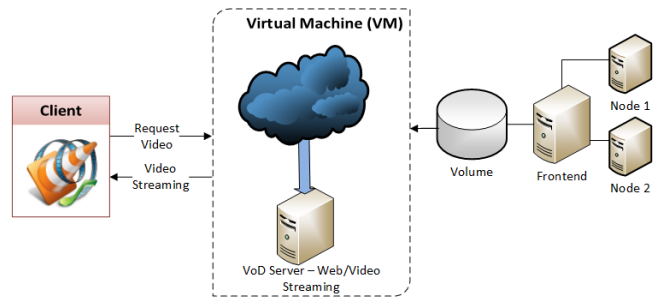

Figure 1: Architecture of Video Streaming Service

applications, is instantiated in the cloud 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 to display a video hosted on a specific web page. VLC, in turn, captures the requested video from the remote storage volume and relays the stream to the user. Further explanation of the infrastructure employed in each model is required to better understand the current research.

IV. AVAILABILITY MODELS

This section describes the availability models designed to represent the non-redundant systems which are the subject of this work. The quantification of availability for complex IT systems is achieved by the representation of system states through the hierarchical modeling of RBDs and Markov chains [12].

A. Model for Architecture

RBD and CTMC models were employed to represent the subsystems of the architectures presented in Figure 1. The combining of these models to perform the analysis constitutes a hierarchical model. The architecture is divided into four parts: Frontend, Node, Volume and Service, as represented by the RDB model depicted in Figure 2. The Volume subsystem,
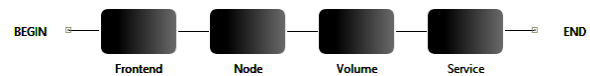

Figure 2: RBD model for architecture

which stores the collected videos, is allocated in the Frontend. Due to the interdependency between system components, the Service subsystem is further refined by a CTMC (see Figure 3). The availability value is calculated for the subsystem which is then fed into the top level RBD model for the computation of overall system availability. The CTMC model enables the acquisition of a closed-form equation of steady-state availability. Such an equation is useful for sensitivity analysis purposes, as well as for computing the desired metrics without incurring a numerical solution of the model. The CTMC depicted in Figure 3 is an availability model of the system composed of the following software components: Apache, VLC and VM. The CTMC has five possible states: UP, Fap, Fapvlc, Fvlc
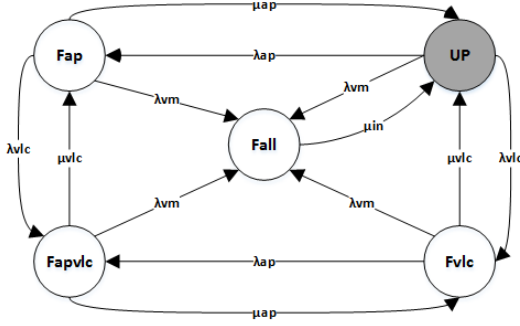
Figure 3: CTMC model for the Service module.

and Fall. The white circles indicate the down states (when the service is unavailable due to failure in at least one component) and the gray circle signifies the operational state. The state Fapvlc implies that both Apache and VLC have failed. The state Fall represents system down due to a VM failure. The rates λ_{ap}, λ_{vlc}, and λ_{vm} denote the failure rates of the Apache, VLC, and VM, respectively. The failure and repair rates are given as $MTTF_i = \frac{1}{\lambda_i}$, and $MTTR_i = \frac{1}{\mu_i}$, assuming that λ_i and μ_i are constant. The repair rate for the Apache is μ_{ap}, and for VLC is μ_{vlc}. μ_{in} is the rate of instantiation of a new VM with the service; i.e., the reciprocal of the mean time to initiate a VM after it has been requested. The closed-form Equation (I) was obtained from the CTMC, and with it the availability of the Service block (see Figure 2) is calculated.

$$A_s = \frac{(\mu in(\lambda ap\lambda vm(\beta) + \lambda ap(\beta_1)\mu vlc + (\beta_1)(\beta_2)(\beta + \mu vlc)))}{((\lambda ap + \beta_1)(\lambda vm + \mu in)(\beta)(\lambda ap + \beta + \mu vlc))},$$

(I)

where: $\beta = \lambda vlc + \lambda vm + \mu ap$, $\beta_1 = \lambda vm + \mu ap$, and $\beta_2 = \lambda vm + \mu vlc$. The result can then be used in the equation $A = Af \times An \times Av \times As$, with which the availability (A) of the entire system may be calculated. In this equation, Af, An, Av, and As correspond to the availability of Frontend, Node, Volume, and Service respectively.

V. CASE STUDY

The case study focused on the analysis of system availability and the identification of components which most affect the streaming service. Initially, the availability of a non-redundant architecture (Figure 1) was evaluated. Following this, the sensitivity of the service model was calculated. Finally, a sensitivity analysis of all the system components (Figure 2) was performed. This created a sensitivity ranking from which it was possible to identify the most critical elements in the streaming service.

A. Availability for Architecture

Figure 1 depicts the architecture, consisting of two machines, one acting as Frontend and the other as Node. Table I presents the RBD model block parameters for the structure illustrated in Figure 2. The MTTF and MTTR values for Frontend, Node, and Volume modules are derived from analyses performed in [4] and [12]. The availability of the

Service module is computed from the CTMC depicted in Figure 3. Table II lists all the parameter values employed in the computation of Service module availability [12]. The rate μ_in represents the mean time to initiate a VM. With this parameter

TABLE I: Parameters for RDB of the system

Component	MTTF	MTTR
Frontend	180.72 h	0.96999 h
Node	481.83 h	0.91000 h
Volume	100000 h	1 h
Service	217.77 h	0.92633 h

TABLE II: Input parameters for the CTMC model of Service module [12] and [4].

Parameters	Description	Values (h^{-1})
λap	Apache failure rate	1/788.4
λvlc	VLC failure rate	1/336
λvm	VM failure rate	1/2880
μap	Apache repair rate	1
μvlc	VLC repair rate	1
μin	Instantiation rate for a new VM	1/0.019166

configuration, an availability value of 0.9885713 was achieved for the non-redundant VoD service.

B. Sensibility Analysis for Service Module

Initially, a parametric sensitivity analysis of the model presented in Section IV-A was performed with the dependability CTMC models proposed for the Service module. The sensitivity indices $S_\theta^*(A)$ were computed, where A is the system steady-state availability and θ is the system parameters (the failure and repair rates of each component).The values for $S_\theta^*(A)$ were obtained with the sensitivity analysis features developed for the SHARPE [3],[10] package. In this case the Equation $S_\theta(As) = \frac{\partial As}{\partial \theta}$ was used for the Service module. The measures of interest relevant to the sensitivity analysis are listed in Table II. The scaled sensitivity of the availability with respect to the CTMC parameters is shown in Table III. The results are ordered according to the absolute values of

TABLE III: Sensitivity ranking for Service module

Parameter	$SS(A_s)$
λ_{vlc}	−0.00296633
μ_{vlc}	0.00296530
λ_{ap}	−0.00126634
μ_{ap}	0.00126590
μ_{in}	0.00005786
λ_{vm}	−0.00005639

the sensitivity index: negative values indicate that there is an inverse relationship between the parameters and the system availability. The sensitivity ranking of Table III shows that the failure rate of VLC (λ_{vlc}) is the most important parameter with respect to availability. The result also shows that the repair rate of VLC (μ_{vlc}) is the second most important parameter, and the first of those directly related to system recovery. It may therefore be stated that VLC is the critical point in the Service module when considering improvements to the availability of this subsystem.

C. Sensitivity Analysis of the Architecture

Sensitivity indices may also be computed by means of Equation II for all the architecture components (Frontend, Node, Volume, and Service).

$$S_\theta(A) = \frac{\partial A\text{f}}{\partial \theta} \times A\text{n} \times A\text{v} \times A\text{s} + A\text{f} \times \frac{\partial A\text{n}}{\partial \theta} \times A\text{v} \times A\text{s}+ \quad \text{(II)}$$

$$A\text{f} \times A\text{n} \times \frac{\partial A\text{v}}{\partial \theta} \times A\text{s} + A\text{f} \times A\text{n} \times A\text{v} \times \frac{\partial A\text{s}}{\partial \theta}$$

With the failure and repair rates of Frontend denoted as λ_f and μ_f, respectively, the corresponding derivative expressions are shown in $\frac{\partial A\text{f}}{\partial \lambda_f} = -\mu_f/(\mu_f + \lambda_f)^2$ and $\frac{\partial A\text{f}}{\partial \mu_f} = -\mu_f/(\lambda_f + \mu_f)^2 + 1/(\lambda_f + \mu_f)$. Since these subsystems are represented with RBDs, the derivative expressions for An and Av are similar to those of Af. The measures of interest were computed with the values given in Tables I and II. Table IV lists the scaled sensitivity of availability for all the video streaming service parameters.

TABLE IV: Sensitivity ranking for availability system

Parameter	$SS(A)$
λ_n	$-1,8850725e-03$
λ_f	$-1,5866281e-05$
μ_f	$1,5866281e-05$
λ_{vlc}	$-8,8157368e-06$
μ_{vlc}	$8,8126759e-06$
μ_n	$5,5918029e-06$
λ_{ap}	$-3,7634921e-06$
μ_{ap}	$3,7621848e-06$
μ_{vol}	$2,9719022e-08$
λ_{vol}	$2,9719022e-08$
μ_{in}	$1,9777666e-08$
λ_{vm}	$1,5409453e-08$

According to the sensitivity ranking of Table IV, Node module failure rate (λ_n) is the most important parameter in respect to availability. The ranking also shows that Frontend failure rate ((λ_f)) is the second most important parameter, whilst the repair rate μ_f is the third, and the first related to system recovery. In this new outcome it may be stated that Node now merits priority as the critical point for considering improvements to the availability of the system. Table IV also shows that the rate μ_{in}, which is the instantiation rate of a new VM in the service, has the least impact on system availability, with the exception only of the λ_{vm} rate. Figure 4 shows system availability as a function of MTTF for Node and Frontend modules, illustrating the respective impact that increasing MTTF values for these modules has on system availability. Increasing Frontend MTTF produces a reduction in downtime of 10.58 hours over the year. On the other had, the effect of increasing Node MTTF is a reduction in downtime of 15.45 hours.

VI. CONCLUSION

This paper proposed a combinatorial approach employing Markov Chain Models and Reliability Block Diagrams to evaluate the system availability of a streaming video service. Furthermore, a parametric sensitivity analysis was employed
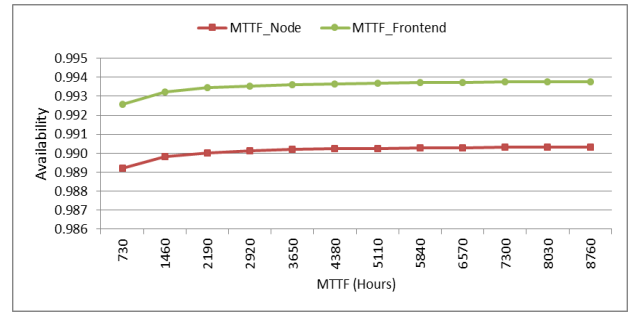

Figure 4: Effect of failure times on system availability.

to identify which parameters had greater influence on system availability, which additionally allowed for weaknesses to be identified, thereby providing a guidance mechanism for the implementation of system improvements. The results of the research showed that the proposed non-redundant architecture achieved an availability of 0.988571. The results also identified Node failure rate as the most important parameter in the architecture with respect to availability, whilst Frontend failure rate is the second most important. For future work, the authors propose to implement further sensitivity analysis techniques and compare their approaches.

REFERENCES

[1] 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.

[2] J. A. Lozano, A. Castro, B. Fuentes, J. M. González, and Á. Rodríguez, "Adaptive qoe measurement on videostreaming ip services," in *Network and Service Management (CNSM), 2011 7th International Conference on*. IEEE, 2011, pp. 1–4.

[3] 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.

[4] J. Dantas, R. Matos, J. Araujo, and P. Maciel, "An availability model for eucalyptus platform: An analysis of warm-standy replication mechanism," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1664–1669.

[5] "Eucalyptus cloud computing platform - administrator guide," 2010, technical report, Eucalyptus Systems, Inc., Version 2.0.

[6] P. Maciel, K. S. Trivedi, R. Matias, and D. S. Kim, "Dependability modeling," in *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*. Hershey: IGI Global, 2011.

[7] J. Laprie, *Dependability: Basic Concepts and Terminology*. Springer-Verlag, 1992.

[8] D. M. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," *Environmental Monitoring and Assessment*, vol. 32, no. 2, pp. 135–154, 1994.

[9] R. Matos Junior, "An automated approach for systems performance and dependability improvement through sensitivity analysis of markov chains," 2011.

[10] R. A. Sahner and K. S. Trivedi, "Reliability modeling using sharpe," *Reliability, IEEE Transactions on*, vol. 36, no. 2, pp. 186–193, 1987.

[11] 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.

[12] 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.