Dependability Models for Designing Disaster Tolerant Cloud Computing Systems

Bruno Silva, Paulo Maciel, Eduardo Tavares
Center for Informatics (CIn)
Federal University of Pernambuco (UFPE)
Recife, Pernambuco
{*bs,prmm,eagt*}*@cin.ufpe.br*

Armin Zimmermann
System & Software Engineering
Ilmenau University of Technology
Ilmenau, Germany
Armin.Zimmermann@tu-ilmenau.de

***Abstract*—Hundreds of natural disasters occur in many parts of the world every year, causing billions of dollars in damages. This fact contrasts with the high availability requirement of cloud computing systems, and, to protect such systems from unforeseen catastrophe, a recovery plan requires the utilization of different data centers located far enough apart. However, the time to migrate a VM from a data center to another increases due to distance. This work presents dependability models for evaluating distributed cloud computing systems deployed into multiple data centers considering disaster occurrence. Additionally, we present a case study which evaluates several scenarios with different VM migration times and distances between data centers.**

***Keywords*-cloud computing; dependability evaluation; stochastic Petri nets;**

I. Introduction

Cloud computing has driven the new wave of Internet-based applications by providing computing as a service [1]. Nowadays, usual business applications (e.g., spreadsheets, text editors) are provided as cloud computing services, in the sense that they are often accessed using a web browser, and, their respective software/data reside on remote servers. This approach has affected all fields of the computational system, from users to hardware manufacturers [2].

Such paradigm is attractive for a number of reasons: (i) it frees users from installing, configuring and updating the software applications; (ii) it offers advantages in terms of mobility as well as collaboration; and (iii) updates and bug fixes can be deployed in minutes, simultaneously affecting all users around the globe [3]. Just like traditional utilities such as telephone, water, electricity and gas, cloud computing service can be adopted in accordance with customer needs, such that he does not have to worry about how and where the service is located. Moreover, the adoption of cloud services enables the adoption of computing resources in a scalable fashion, i.e., as new services and resources are needed, the infrastructure is available on demand [3].

An important type of cloud service is the Infrastructure as a Service (IaaS), such as Amazon EC2 [4] and IBM Smart Business Cloud [5]. IaaS delivers, on-demand, computing resources in the form of virtual machines (VMs) deployed into the cloud provider's data center (i.e., IaaS provider), satisfying user needs [6].

In this context, availability is a prominent metric to assess provider's quality-of-service (QoS). For prominent IaaS providers, the availability level is regulated by adopting a Service Level Agreement (SLA), which specifies, for instance, the maximum downtime per year. Penalties may be applied if the defined availability level is not satisfied. Thus, to meet SLA requirements, IaaS providers need to evaluate the dependability level of its environment, contemplating, also, the possibility of disasters.

A disaster recovery plan requires the utilization of different data centers located far enough apart to mitigate the effects of unforeseen disasters (e.g., earthquakes) [7]. If multiple data centers are located in different geographical locations, it is expected an availability improvement for the whole system. On the other hand, VM migration time increases due to distance between data centers. Consequently, dependability evaluation considering VM migration time is of utmost importance when considering the analysis of distributed cloud systems.

This work presents an approach to evaluate dependability metrics in cloud computing systems deployed into geographically distributed data centers as well as taking into account disaster occurrence. The proposed approach contemplates combinatorial (RBD - Reliability Block Diagrams) and state-based models (SPN - Stochastic Petri Nets) to allow dependability evaluation using a hierarchical modeling [8]. Using the proposed approach, IaaS providers can evaluate the system distributed in different data centers and the impact of VM migration on dependability metrics.

The paper is organized as follows. Section II highlights the related works. Section III describes the cloud computing system considered. Then, the formal dependability models are introduced in Section IV, and Section V presents a case study. Finally, Section VI concludes this paper and introduces future works.

II. Related Works

Over the last years, some authors have been devoting efforts to study dependability issues on cloud computing systems. Longo et al. [6] proposed an approach for availability analysis of cloud computing systems taking into account Petri nets and Markov chains. The authors also developed

closed-form equations and demonstrated their approach can scale for large systems.

In [9], a performability analysis for cloud systems is presented. The authors quantify the effects of variations in workload, failure rate and system capacity on service quality. In [10], the authors investigate the aging effects on Eucalyptus framework, and they also propose a strategy to mitigate such issues during system execution. [11] describes a system design approach for supporting transparent migration of virtual machines that adopt local storage for their persistent state. The approach is transparent to the migrated VM, and it does not interrupt open network connections during VM migration.

In [12], the authors present a case study that quantifies the effect of VM live migrations in the performance of an Internet application. Such study helps data center designers to plan environments in which metrics, such as service availability and responsiveness, are driven by Service Level Agreements. Dantas et al. [13] presents a study of warm-standby mechanisms in Eucalyptus framework. Their results demonstrate that replacing machines by more reliable counterparts would not produce improvements in system availability, whereas some fault-tolerant techniques can indeed increase dependability levels.

Unlike previous works, this paper proposes dependability models for evaluating cloud computing systems deployed into geographically distributed data centers, considering VM migration and the occurrence of disasters.

III. System Architecture

This section presents an overview of the cloud computing system considered in this work, which contemplates a set of components, distributed over distinct data centers (Figure 1).

The system is composed of d data centers, each with two set of machines, namely, hot and warm pools. The hot pool is composed of n physical machines (PM), which are active and running virtual machines (VM). The warm pool takes into account m PMs that are active, but without running VMs. Thus, the number of PMs in a data center is $t = m+n$.

Depending on the capacity of each PM, it is possible to run multiple VMs in the same host. In this study, we assume all physical machines are identical, in the sense that they adopt the same services and hardware/software components. PMs may share a common network attached storage (NAS) or they may adopt a storage area network (SAN) to provide distributed storage and, also, to allow the migration of a virtual machine from one server to another in the same data center [14]. In case of failure, a VM must be instantiated in another physical machine. If there is no available PM, the VM image is migrated to another data center.

Furthermore, a Backup Server (BS) is considered to provide backup of VM data. This component receives a copy of each VM image during data center operation. Hence, whenever a disaster makes one data center unavailable, BS sends VM copies to an operational data center. In this work, the number of running VMs (w) is compared with a threshold (k) to evaluate the availability of cloud computing system. Hence, if $w \geq k$ the system is operational.

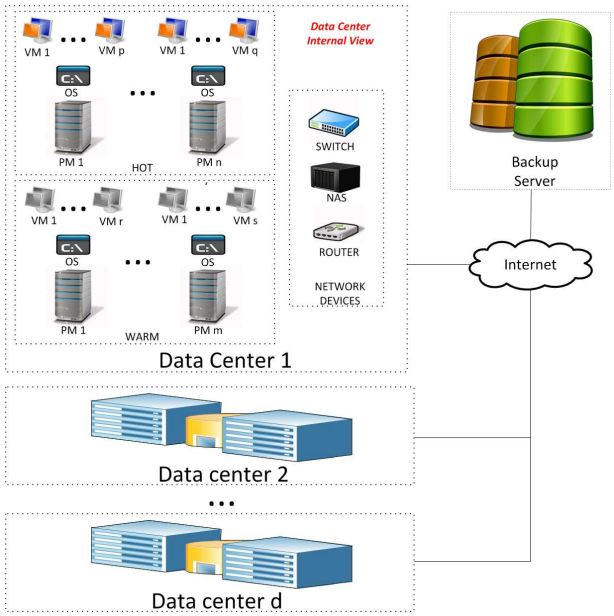


Figure 1. Distributed Cloud System Example

IV. Modeling

This section presents the adopted hierarchical modeling to evaluate system dependability. Firstly, the basic models are presented, and, then, the modeling approach is detailed. Lastly, the approach is demonstrated for representing a cloud computing configuration.

Henceforth, the following operators are adopted for assessing dependability metrics: $P\{exp\}$ estimates the probability of the inner expression (exp); and $\#p$ denotes the number of tokens in place p.

A. SPN block: SIMPLE_COMPONENT

The first component to be represented is named as "*SIMPLE_COMPONENT*". This component is characterized by the absence of redundancy, that is, the component might be in two states, either functioning or failed. In order to compute its availability, mean time to failure (*MTTF*) and mean time to repair (*MTTR*) are the only parameters needed for computing its availability.

The respective SPN model of the "*SIMPLE_COMPONENT*" is shown in Figure 2. Both transitions are exponentially distributed (exp) and have single server (ss) semantic [15]. Table I depicts the attributes related to transitions of the *SIMPLE_COMPONENT* model.

Places *X_ON* and *X_OFF* are the model component's activity and inactivity states, respectively. Label "X" is

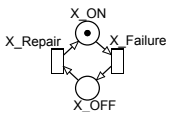


Figure 2. *SIMPLE_COMPONENT* model

Table I
SIMPLE_COMPONENT TRANSITION ATTRIBUTES.

Transition	Type	Delay	Markup	Concurrency
X_Failure	exp	*MTTF*	constant	*ss*
X_Repair	exp	*MTTR*	constant	*ss*

instantiated according to the component name, for instance, *DC_UP* and *DC_DOWN* (Figure 6). A component is operational only if the number of tokens (#) in place *X_ON* is greater than 0. $P\{\#X\_ON>0\}$ means the component's availability (steady state evaluation).

B. SPN block: VM_BEHAVIOR Component

VM_BEHAVIOR component represents the behavior of N VMs running on a physical machine. This basic block interacts with 3 *SIMPLE_COMPONENT* models: (i) one representing the occurrence of disasters (*DC*); (ii) the physical machine (*OSPM*); and (iii) the network (*NAS_NET*). Figure 3 presents the model, in which places *VM_UP1*, *VM_DOWN1*, *VM_RDY1* and *VM_STRTD1* denote the amount of VMs in states operational, failure, repairing, and starting, respectively. Place *FailedVMS* represents VMs that are failed and can be started in another PMs.

Transitions *VM_F1*, *VM_R1* and *VM_STRT1* represent the failure, repair and starting activities associated with the virtual machines. The connections with the *SIMPLE_COMPONENT* models are carried out by immediate transitions *FPM_UP1*, *FPM_DW1*, *FPM_ST1*, *FPM_Subs1* and the respective guard conditions (Table II).

More specifically, these immediate transitions verify disaster occurrence as well as the failures on the physical machine and network devices. As the reader should note, a VM fails whenever the respective infrastructure is not capable to provide the service. Transition *VM_Subs1* denotes the opposite idea, in the sense that, virtual machines start only if the required infrastructure is operational.

Table II
GUARD EXPRESSIONS FOR *VM_BEHAVIOR* COMPONENT OF FIGURE 3.

Transition	Condition	Description
FPM_UP1	*(#OSPM_UPx=0) OR (#NAS_NET_UPy=0) OR (#DC_UPz=0)*	Failure of physical machine or infrastructure
FPM_DW1	*(#OSPM_UPx=0) OR (#NAS_NET_UPy=0) OR (#DC_UPz=0)*	Failure of physical machine or infrastructure
FPM_ST1	*(#OSPM_UPx=0) OR (#NAS_NET_UPy=0) OR (#DC_UPz=0)*	Failure of physical machine or infrastructure
VM_Subs1	*(#OSPM_UPx>0) AND (#NAS_NET_UPy>0) AND (#DC_UPz>0)*	Physical machine and infrastructure working

Table III depicts the attributes related to *VM_BEHAVIOR* component transitions, in which *X_VM_MTTF*, *X_VM_MTTR* and *X_VM_STRT_TIME* denote the mean time to failure, repair and start a VM, respectively.

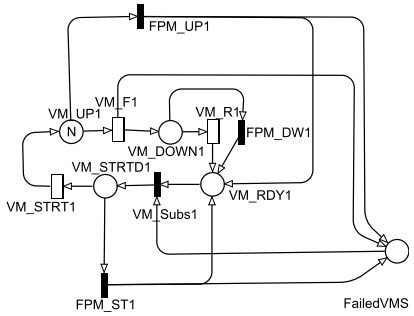


Figure 3. *VM_BEHAVIOR* Component

Table III
VM_BEHAVIOR COMPONENT TRANSITION ATTRIBUTES.

Transition	Type	Delay	Markup	Concurrency
VM_F1	exp	*X_VM_MTTF*	constant	*is*
VM_R1	exp	*X_VM_MTTR*	constant	*is*
VM_STRT1	exp	*X_VM_STRT_TIME*	constant	*ss*

Availability is estimated using the expression $P\{\#VM\_UP1 \geq j\}$, in which j represents the number of virtual machines required to provide the service.

C. SPN block: TRANSMISSION_COMPONENT

Figure 4 presents *TRANSMISSION_COMPONENT*, which represents the transmission of a virtual machine from one data center to another. A VM should migrate to another data center whenever the number of operational physical machines in the data center is less than a given number l. The constant l depends on the service provided and the capacity of each environment. Moreover, Backup Server is responsible for migrating the VM image in case of disaster or network error.

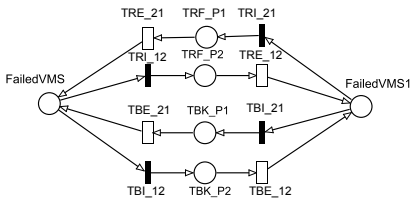


Figure 4. *TRANSMISSION_COMPONENT* SPN model

TRANSMISSION_COMPONENT has eight transitions: four exponentially distributed transitions that represent the VM data transfer and four immediate transitions that depict the enabling of the VM migration. *TRE_21* represents the data transfer from Data Center 2 to Data Center 1; *TRE_12* characterizes the migration from Data Center 1 to Data Center 2; *TBK_21* corresponds to data transfer from Backup Server to Data Center 1, and *TBK_12* characterizes the data

transfer from Backup Server to Data Center 2. Table IV presents the guard expressions of transmission *TRANSMISSION_COMPONENT*.

Table IV
GUARD EXPRESSIONS FOR *TRANSMISSION_COMPONENT*.

Transition	Condition
TRI_12	*((#OSPM_UP1+#OSPM_UP2)=0) AND NOT ((#OSPM_UP3 + #OSPM_UP4)=0 OR #NAS_NET_UP2=0 OR #DC_UP2=0)*
TRI_21	*((#OSPM_UP3+#OSPM_UP4)=0) AND NOT ((#OSPM_UP1 + #OSPM_UP2)=0 OR #NAS_NET_UP1=0 OR #DC_UP2=1)*
TBI_12	*#BKP_UP=1 AND (#NAS_NET_UP1=0 OR #DC_UP1=0) AND NOT((#OSPM_UP3+#OSPM_UP4)=0 OR #NAS_NET_UP2=0 OR #DC_UP2=0)*
TBI_21	*#BKP_UP=1 AND (#NAS_NET_UP2=0 OR #DC_UP2=0) AND NOT((#OSPM_UP1+#OSPM_UP1)=0 OR #NAS_NET_UP1=0 OR #DC_UP1=0)*

The mean time to transmit *(MTT)* symbolizes the mean time to transmit one virtual machine from one location to another. The *MTT* depends on the physical link speed, the distance between the data centers and the VM size. In this block, there are three *MTTs*: mean time to transmit a VM from the data center to another (*MTT_DCS*) and the mean times to transfer the VM image from Backup Server to Data Centers 1 and 2 (*MTT_BK1 and MTT_BK2*). Table V depicts the attributes related to *TRANSMISSION_COMPONENT* exponential transitions.

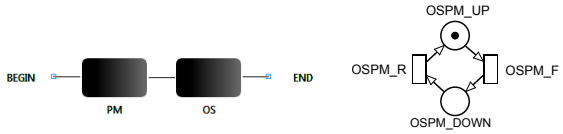
Table V
TRANSMISSION_COMPONENT TRANSITION ATTRIBUTES.

Transition	Type	Delay	Markup	Concurrency
TRE_21	exp	*MTT_DCS*	constant	*ss*
TRE_12	exp	*MTT_DCS*	constant	*ss*
TBE_21	exp	*MTT_BK1*	constant	*ss*
TBE_12	exp	*MTT_BK2*	constant	*ss*

D. Hierarchical modeling

The adopted modeling process considers first the evaluation of lower-level submodels, then the respective results are applied to higher-level models. For instance, Figure 5 depicts a RBD model, such that the operating system (OS) and the physical machine (PM) are in series arrangement. *MTTR* and *MTTF* results estimated from the RBD [16] are associated to transitions *OSPM_R* and *OSPM_F*, respectively, of the SPN model depicted in Figure 5(b).

The modeling approach contemplates RBD models for representing the physical machine (*OS_PM*) as well as the data center network infrastructure (*NAS_NET*), such that the respective MTTFs and MTTRs are estimated and utilized in *SIMPLE_COMPONENT* models (Section IV-A). Considering *OS_PM*, the components are OS and physical machine and a series relation is assumed. Similarly, *NAS_NET* contemplates switch, router and distributed storage considering a series arrangement. Furthermore, this section assumes the adoption of some composition rules (e.g., net union), and the reader refers to [17] for detailed information.



(a) RBD model (b) SPN model

Figure 5. RBD model and the respective SPN model for operating system and physical machine.

E. SPN Model - Cloud system with multiple data centers.

This section assumes a system based on Section III deployed into two data centers each with two PMs and up to two VMs per machine (N=4). Figure 6 presents the model, which is composed of *VM_BEHAVIOR* submodels as well simple components. *OSPM_1* and *OSPM_2* represent the physical machines of Data Center 1, and *OSPM_3* as well as *OSPM_4* are the models related to PMs of Data Center 2. *DISASTER1* and *DISASTER2* models disasters in Data Centers 1 and 2, respectively. *NAS_NET_1* and *NAS_NET_2* corresponds to network devices of Data Center 1 and 2.

In this model, the dynamic behavior of the virtual machines is modelled by a transmission component (*TRANSMISSION_COMPONENT*) and *VM_BEHAVIOR* components. The expression $P\{(\#VM\_UP1 + \#VM\_UP2 + \#VM\_UP3 + \#VM\_UP4)=j\}$ is adopted to estimate availability, in which j represents the amount of virtual machines that are required to provide the service.

V. CASE STUDIES.

To illustrate the feasibility of the proposed approach, we present a case study considering a set of cloud system scenarios in which the systems are deployed into two different data centers. We have conducted an availability evaluation considering (i) distance between data centers, (ii) network speeds and (iii) disaster mean time.

The data centers are located in the following pairs of cities: Rio de Janeiro (Brazil)-Brasilia (Brazil), Rio de Janeiro-Recife (Brazil), Rio de Janeiro-NewYork (USA), Rio de Janeiro-Calcutta (India) and Rio de Janeiro-Tokio (Japan). We assume that the Backup Server is located in São Paulo (Brazil).

To estimate the *MTT* value, we considered the approach presented in [18] that assess the network throughput based on the distance between the communication nodes. The equation associates a constant α with the network speed, which can vary from 0 (no connection) up to 1.0 (fastest connection). We have considered the following values for α: 0.35, 0.40 and 0,45. We assume that it is necessary at least two running VMs to consider the system operational and the size of VMs is 4GB.

The disaster mean time values utilized are 100, 200 and 300 years and a data center takes one year to be recovered. Moreover, a VM takes five minutes to start. Table VI presents the dependability parameters associated

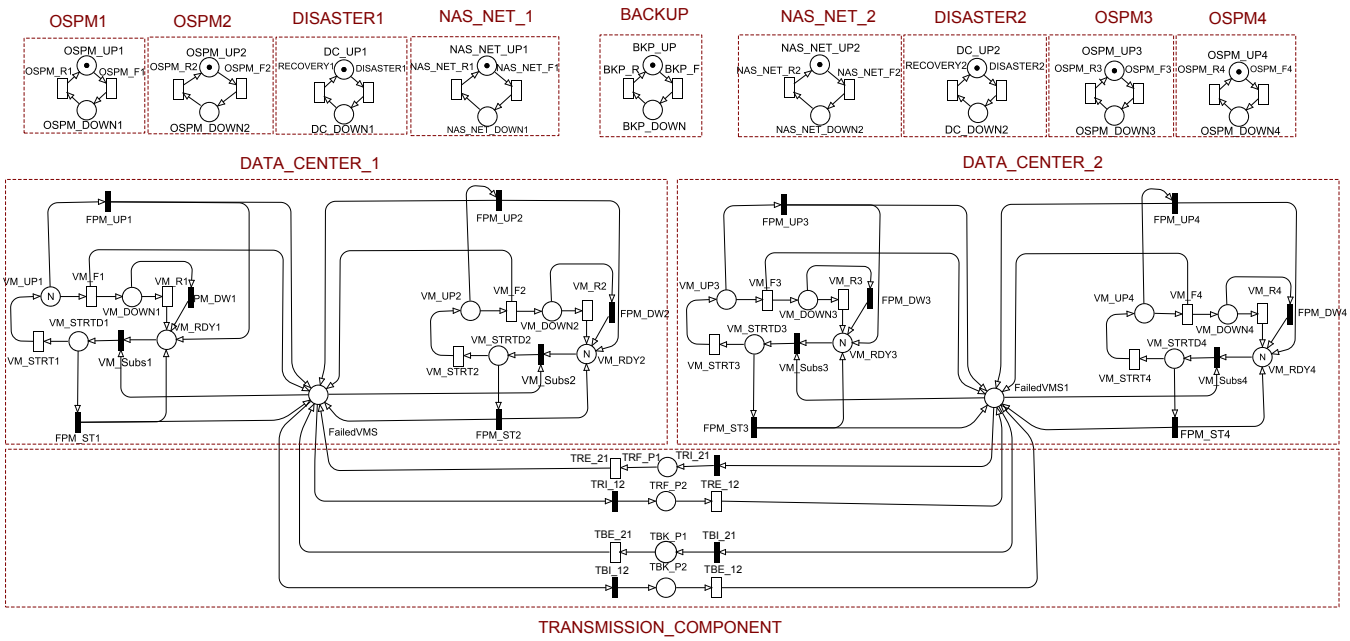


Figure 6. SPN model representing four physical machines in two different data centers

with the devices, which were taken from [19], [20], [21], [22]. Mercury [23] and TimeNET [24] tools have been adopted to perform the evaluation.

Table VI
DEPENDABILITY PARAMETERS FOR COMPONENTS OF FIGURE 1.

Component	MTTF(h)	MTTR(h)
Operating System (OS)	4000	1
Hardware of Physical Machine (PM)	1000	12
Switch	430000	4
Router	14077473	4
NAS	20000000	2
VM	2880	0.5
Backup Server	50000	0.5

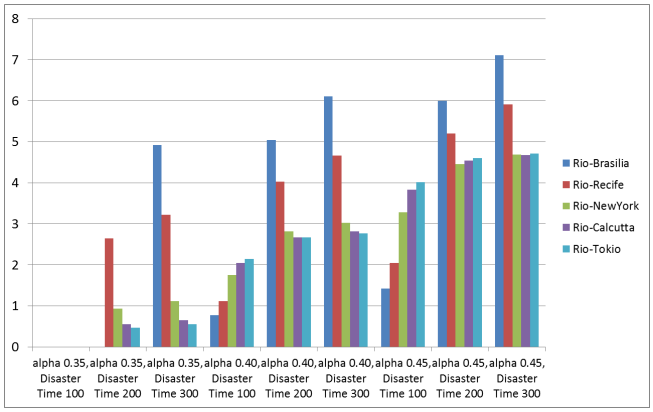


Figure 7. Availability increase of different ditributed cloud configurations

Figure 7 shows availability results for each different configuration. The baseline architectures are the systems with α = 0.35 and disaster mean time = 100 years. The results are presented in terms of improvement of number of nines, which is calculated by expression $nines = -log[1 - A]$ (A corresponds to availability). The results show that the higher availability scenario corresponds to a system with data centers in Rio de Janeiro - Brasilia, alpha = 0.45 and disaster mean time = 300 years. We can also observe smaller distances and disaster mean time significantly affects the availability. If larger distances are considered, the availability is mostly impacted by the network speed.

Table VII
AVAILABILITY VALUES FOR THE BASELINE ARCHITECTURES.

Architecture	Availability	Number of nines
Cloud system with one machine	0.9842914	1.80
Cloud system with two machines in one data center	0.9899101	1.99
Cloud system with four machines in one data center	0.9900631	2.00
Baseline architecture: Rio de janeiro - Brasilia	0.9997317	3.57
Baseline architecture: Rio de janeiro - Recife	0.9995968	3.39
Baseline architecture: Rio de janeiro - NewYork	0.9987753	2.91
Baseline architecture: Rio de janeiro - Calcutta	0.9977486	2.64
Baseline architecture: Rio de janeiro - Tokio	0.9972643	2.56

Table VII compares availability values of baseline architectures with non geographically distributed cloud systems with the same basic components. In this table, the system with more distant data centers has better availability than the scenario with the same number of machines in a single data center.

VI. CONCLUSION

This work presented models for dependability evaluation of cloud computing systems deployed into geographically

distributed data centers as well as taking into account disaster occurrence. The approach is based on a hybrid modeling technique, which considers combinatorial and state-based models. The proposed technique allows the impact assessment of disaster occurrence, VM migration and data center distance on system dependability.

Additionally, a case study is provided considering a set data centers located in different places around the world. The results demonstrated the influence of distance, network speed and disaster occurrence on system availability. As future research, we intend to assess performance metrics in the proposed method.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[2] D. A. Menasc and P. Ngo, "Understanding cloud computing: Experimentation and capacity planning," 2009.

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 2010.

[4] Amazon ec2. [Online]. Available: http://aws.amazon.com/ec2

[5] Ibm smart business cloud. [Online]. Available: http://www-935.ibm.com/services/us/igs/cloud-development/

[6] F. Longo, R. Ghosh, V. Naik, and K. Trivedi, "A scalable availability model for infrastructure-as-a-service cloud," in *Dependable Systems Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*, june 2011, pp. 335 –346.

[7] *Hyper-V Live Migration over Distance*. [Online]. Available: http://www.hds.com/assets/pdf/hyper-v-live-migration-over-distance-reference-architecture-guide.pdf

[8] P. Maciel, K. S. Trivedi, R. Matias, and D. S. Kim, *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*, ser. Premier Reference Source. Igi Global, 2011, ch. Dependability Modeling.

[9] R. Ghosh, K. S. Trivedi, V. K. Naik, and D. S. Kim, "End-to-end performability analysis for infrastructure-as-a-service cloud: An interacting stochastic models approach," in *Proceedings of the 2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing*, ser. PRDC '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 125–132.

[10] J. Araujo, R. Matos, P. Maciel, R. Matias, and I. Beicker, "Experimental evaluation of software aging effects on the eucalyptus cloud computing infrastructure," in *Proceedings of the Middleware 2011 Industry Track Workshop*, ser. Middleware '11. New York, NY, USA: ACM, 2011, pp. 4:1–4:7.

[11] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schiöberg, "Live wide-area migration of virtual machines including local persistent state," in *Proceedings of the 3rd international conference on Virtual execution environments*, ser. VEE '07. New York, NY, USA: ACM, 2007, pp. 169–179.

[12] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Proceedings of the 1st International Conference on Cloud Computing*, ser. CloudCom '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 254–265.

[13] 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*, oct. 2012, pp. 1664 –1669.

[14] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286.

[15] R. German, *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[16] C. Ebeling, *An Introduction to Reliability and Maintainability Engineering*. Waveland Press, 1997.

[17] G. de Albuquerque, P. Maciel, R. Lima, and A. Zimmermann, "Automatic modeling for performance evaluation of inventory and outbound distribution," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 5, pp. 1025 –1044, sept. 2010.

[18] W. M. Les Cottrell and C. Logg, "Tutorial on internet monitoring and pinger at slac," Tech. Rep., 1996. [Online]. Available: http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html

[19] D. S. Kim, F. Machida, and K. S. Trivedi, "Availability modeling and analysis of a virtualized system," in *Proceedings of the 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*, ser. PRDC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 365–371.

[20] (2012, Oct.) Cisco systems: Switch dependability parameters. [Online]. Available: http://tinyurl.com/cr9nssu

[21] (2012, Oct.) Cisco systems: Router dependability parameters. [Online]. Available: http://tinyurl.com/d7kcnqo

[22] (2012, Oct.) Service level agreement - megapath business access and value added services. [Online]. Available: http://tinyurl.com/cwdeebt

[23] B. Silva, G. Callou, E. Tavares, P. Maciel, J. Figueiredo, E. Sousa, C. Araujo, F. Magnani, and F. Neves, "Astro: An integrated environment for dependability and sustainability evaluation," *Sustainable Computing: Informatics and Systems*, no. 0, pp. –, 2012.

[24] R. German, C. Kelling, A. Zimmermann, and G. Hommel, "Timenet: a toolkit for evaluating non-markovian stochastic petri nets," *Performance Evaluation*, vol. 24, no. 1-2, pp. 69 – 87.