Survivability Evaluation of Disaster Tolerant Cloud Computing Systems

Bruno Silva[a*], Paulo Romero Martins Maciel[a], Armin Zimmermann[b] and Jonathan Brilhante[a]

[a]Federal University of Pernambuco, Recife, Brasil

[b]Ilmenau University of Technology, Ilmenau, Germany

Abstract: A prominent type of cloud service is the Infrastructure-as-a-Service (IaaS), which delivers, on-demand, computing resources in the form of virtual machines (VMs) satisfying user needs. In such systems, penalties may be applied if the defined quality level of service level agreement (SLA) is not satisfied. Therefore, high availability is a critical requirement of these systems. A strategy to protect such systems from natural or manmade disasters corresponds to the utilization of multiple data centers located into different geographical locations to provide the service. Considering such systems, redundancy mechanisms can be adopted to receive copies of VM images during data center operation. Hence, whenever a disaster makes one data center unavailable, the VMs can be re-instantiated in other operational data center. Modeling techniques, with a strong mathematical foundation, such as Stochastic Petri Nets (SPN) can be adopted to evaluate survivability in these complex infrastructures. This work presents SPN models to evaluate survivability metrics in IaaS systems deployed into geographically distributed data centers taking into account disaster occurrences. Using the proposed models, IaaS providers can evaluate the impact of VM transmission time and the VM backup period on survivability metrics. A case study is provided to illustrate the effectiveness of the proposed work.

Keywords: survivability, IaaS systems, disaster recovery plan, stochastic Petri nets.

1. INTRODUCTION

Cloud computing has driven the new wave of Internet-based applications by providing computing as a service [1]. Nowadays, common 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 research, 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]. 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) running on the cloud provider's data center, satisfying user needs. User requests are provisioned depending on the data center capacity in terms of physical machines.

For prominent IaaS providers, the quality 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 quality level is not satisfied. Thus, to meet SLA requirements, IaaS providers need to evaluate their environment, considering, also, the possibility of disasters. Therefore, 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) [6]. Considering such systems, redundant data centers can be adopted to receive copies of VM images during data center operation. Hence, whenever a disaster

* Corresponding author, bs@cin.ufpe.br

makes one data center unavailable, the VMs can be re-instantiated in other operational data center. Unfortunately, some data between the last VM backup and the disaster may be lost and it is necessary some time to restart the operation after a failure. However, this may be traded-off changing the time between backups and the distance between data centers. Two metrics can be utilized to evaluate system survivability: (i) Recovery Point Objective (RPO), which corresponds to the maximum age of the most recent backup prior to disaster and (ii) Recovery Time Objective (RTO) that specifies the maximum time to repair the service after a disaster occurrence. Modeling techniques, with a strong mathematical foundation, such as Stochastic Petri Nets (SPN) [7] can be adopted to evaluate survivability in complex infrastructures.

This work presents an approach to evaluate survivability in IaaS systems deployed into geographically distributed data centers as well as taking into account disaster occurrence. The proposed approach contemplates state-based models (SPN - Stochastic Petri Nets) to determinate the probability of IaaS systems meet their survivability objectives. Using the proposed approach, IaaS providers can evaluate the system distributed in different data centers and the impact of VM backup time on these metrics. The paper is organized as follows. Section 2 highlights the related works. Section 3 describes the cloud computing system considered. Then, basic concepts about SPN models are introduced in Section 4. Section 5 presents the survivability parameters adopted in this work. In Section 6, the SPN models adopted to evaluate IaaS survivability is presented. Finally, Section 8 shows the adopted case study and Section 8 concludes this paper.

2. RELATED WORK

Over the last years, some authors have been devoting efforts to study dependability issues on cloud computing systems. Longo et al. [8] proposed an approach for availability analysis of cloud computing systems based on Petri nets and Markov chains. The authors also developed closed-form equations and demonstrated that 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 the Eucalyptus framework [11], and they also propose a strategy to mitigate such issues during system execution.

[12] 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 [13], 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. [14] present 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 techniques of fault-tolerance can indeed increase dependability levels.

In [15], the authors adopted model checking algorithms to decide if a given system is survivable. The logic CSL was adopted to represent and estimate survivability metrics in GOOD (given-occurrence-of-disaster) models. Unlike previous works, this paper proposes performability models for evaluating cloud computing systems deployed into geographically distributed data centers, considering VM transfer data and disasters occurrence.

3. SYSTEM ARCHITECTURE OF RELIABLE DISTRIBUTED DATA CENTERS

This section presents an overview of the cloud computing system considered in this work, which consists of a set of distributed data centers (Figure 1). The system is composed of D data centers. A Backup Server (BS) is assumed to provide backup of VM data, which periodically 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 operational data centers. 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 assumed operational.

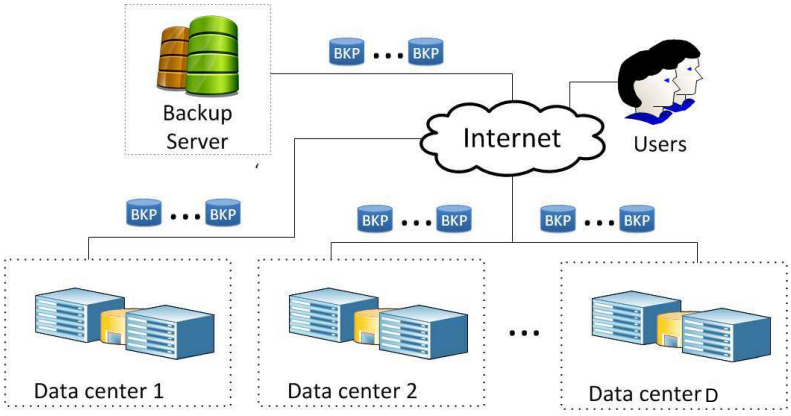

Figure 1: Distributed Cloud System Example

4. SPN MODELS

Petri nets (PN) [16] are a family of formalisms very well suited for modeling several system types, since concurrency, synchronization, communication mechanisms as well as deterministic and probabilistic delays are naturally represented. This work adopts a particular extension, namely, Stochastic Petri Nets (SPN) [17], which allows the association of stochastic delays to timed transitions using the exponential distribution, and the respective state space can be converted into continuous-time Markov chains (CTMC) [18]. Figure 2 depicts an example of a SPN model. Places are represented by circles, whereas transitions are depicted as filled rectangles (immediate transitions) or hollow rectangles (timed transitions).

Arcs (directed edges) connect places to transitions and vice-versa. Tokens (small filled circles) may reside in places, which denote the state (i.e., marking) of a SPN. An inhibitor arc is a special arc type that depicts a small white circle at one edge, instead of an arrow, and they usually are used to disable transitions if there are tokens present in a place. The behaviour of a SPN is defined in terms of a token flow, in the sense that tokens are created and destroyed according to the transition firings [19]. Immediate transitions represent instantaneous activities, and they have higher firing priority than timed transitions. Besides, such transitions may contain a guard condition, and a user may specificy a different firing priority among other immediate transitions. SPNs also allow the adoption of simulation techniques for obtaining dependability metrics, as an alternative to the generation of a CTMC. Regarding SPN formal definitions and semantic, the reader is referred to [17].

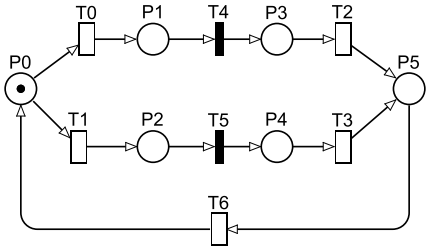

Figure 2: SPN model example

4.1. Distribution Moment Matching

A well-established method that considers *expolynomial distribution* random variables is based on distribution moment-matching . The moment matching process presented in [20] and considers that Hypoexponential and Erlangian distributions have the average delay (μ) greater than the standard-deviation (σ) -$\mu > \sigma$-, and Hyperexponential distributions have $\mu < \sigma$, in order to represent an activity with a generally distributed delay as an Erlangian or a Hyperexponential subnet referred to as s-transition. One should note that in cases where these distributions have $\mu = \sigma$, they are, indeed, equivalent to an exponential distribution with parameter equal to $\frac{1}{\mu}$. Therefore, according to the coefficient of variation associated with an activity's delay, an appropriate s-transition implementation model could be chosen. For each s-transition implementation model (see Figure 3), a set of parameters should be configured for matching their first and second moments. In other words, an associated delay distribution (it might have been obtained by a measuring process) of the original activity is matched with the first and second moments of s-transition (*expolynomial distribution*). According to the aforementioned method, one activity with $\mu < \sigma$ is approximated by a two-phase Hyperexponential distribution with parameters

$$r_1 = \frac{2\mu^2}{(\mu^2 + \sigma^2)}, \tag{1}$$

$$r_2 = 1 - r_1 \tag{2}$$

and

$$\lambda = \frac{2\mu}{(\mu^2 + \sigma^2)}. \tag{3}$$

where λ is the rate associated to phase 1, r_1 is the probability of related to this phase, and r_2 is the probability assigned to phase 2. In this particular model, the rate assigned to phase 2 is assumed to be infinity, that is, the related average delay is zero.

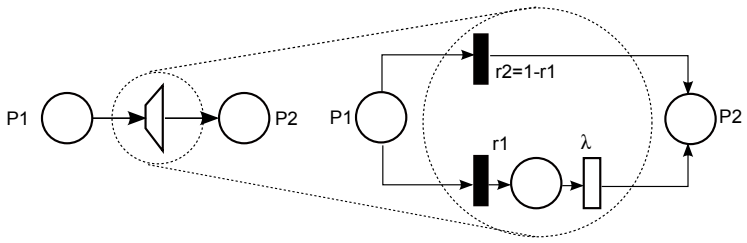

Figure 3: Hyperexponential Model

Activities with coefficients of variation less than one might be mapped either to Hypoexponential or Erlangian s-transitions. If $\frac{\mu}{\sigma} \notin \mathbb{N}, \frac{\mu}{\sigma} \neq 1, (\mu, \sigma \neq 0)$, the respective activity is represented by a Hypoexponential distribution with parameters λ_1, λ_2(exponential rates); and γ, the integer representing the number of phases with rate equal to λ_2, whereas the number of phases with rate equal to λ_1 is

one. In other words, the s-transition is represented by a subnet composed of two exponential and one immediate transitions. The average delay assigned to the exponential transition t_1 is equal to μ_1 ($\lambda_1 = 1/\mu_1$), and the respective average delay assigned to the exponential transition t_2 is $\mu_2(\lambda_2 = 1/\mu_2)$. γ is the integer value considered as the weight assigned to the output arc of transition t_1 as well as the input arc weight value of the immediate transition t_3 (see Figure 4). These parameters are calculated by the following expressions:

$$(\frac{\mu}{\sigma})^2 - 1 \leq \gamma < (\frac{\mu}{\sigma})^2, \tag{4}$$

$$\lambda_1 = \frac{1}{\mu_1} \; and \; _2 = \frac{1}{\mu_2}, \tag{5}$$

where

$$\mu_1 = \frac{\mu \pm \sqrt{\gamma(\gamma+1)\sigma^2 - \gamma\mu^2}}{\gamma+1}, \tag{6}$$

$$\mu_2 = \frac{\gamma\mu \mp \sqrt{\gamma(\gamma+1)\sigma^2 - \gamma\mu^2}}{\gamma+1} \tag{7}$$

If $\frac{\mu}{\sigma} \in \mathbb{N}, \frac{\mu}{\sigma} \neq 1, (\mu, \sigma \neq 0)$, an Erlangian s-transition with two parameters, $\gamma = (\frac{\mu}{\sigma})^2$ is an integer representing the number of phases of this distribution; and $\mu_1 = \mu/\gamma$, where $\mu_1(1/\lambda_1)$ is the average delay value of each phase. The Erlangian model is a particular case of a Hypoexponential model, in which each individual phase rate has the same value. The reader should refer to [20] for details regarding the representation of expolinomial distributions using SPN.

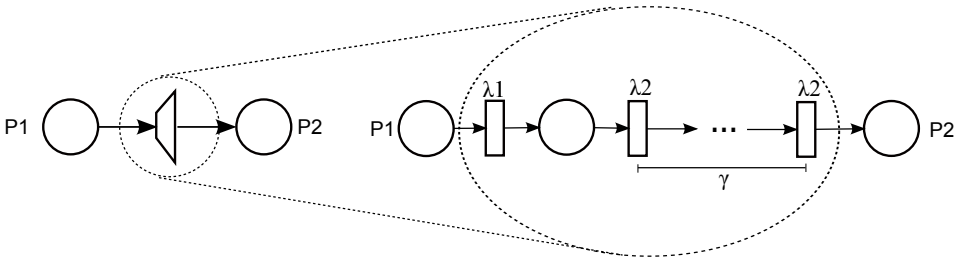

Figure 4: Hypoexponential Model

5. BUSINESS CONTINUITY OBJECTIVES

Survivability can be defined as the ability of a system to recover a predefined service level in a timely manner after the occurrence of disasters [15]. In this context, companies are adopting Business Continuity Management (BCM) [21] to support the ability to operate in spite of unforeseen events and recover in a short time frame. The main BCM's outcome is the Business Continuity Plan (BCP) or Disaster Recovery Plan (DRP), which is a document that describes the business continuity process in order to reduces or minimizes impact of events that disrupt critical services and their supporting resources [21]. Two indexes are utilized to define survivability objectives: (i) Recovery Point Objective (RPO), which corresponds to the time limit of the most recent backup prior to disaster and (ii) Recovery Time Objective (RTO) that specifies the maximum time to repair the service after a disaster occurs.

These objectives are based on business decisions that contemplate costs of inactivity periods and data loss. Additionally, technological factors (e.g., system performance) must be considered to stablish these parameters [22, 23].

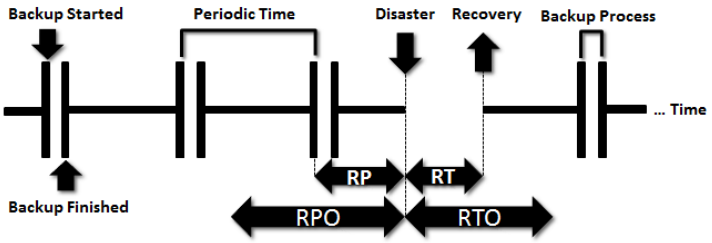

Figure 5: RPO and RTO requirements.

Figure 5 illustrates the backup operation along the time for a general system. During the system operation, a backup is periodically performed and whenever a disaster happens, the last backup should be recovered. If the age of the last backup (Recovery Point - RP) is higher than the RPO or the time to recover the system (Recovery Time - RT) is higher than the RTO, then the survivability requirements are not satisfied. In this case (Figure 5), the system meet the requirements. It is important to state that the amount of data that should be restored or backed up is not fixed for some applications. Consequently, the time to perform backup and restore operations is stochastic and depends on the amount of data involved and the technology utilized [23]. Additionally, for some applications (e.g., collaboration websites) the RPO and RTO should not be higher than a few minutes. On the other hand, for other applications (e.g., static websites) these requirements are not so critical.

5.1. Recovery Point Evaluation

In this study, we are interested in evaluating the recovery point considering the worst-case scenario (Figure 6). According to [24], this situation happens when the disaster occurs during the backup process. Observe that, in this case, the Recovery Point (R_p) is equal to the sum of the Backup Period (B_p) and the actual Backup Time (B_t). Consequently, the probability of meeting the RPO (P_{RPO}) in the worst-case scenario is given by:

$$P_{RPO} = P\{R_p \leq RPO\} = P\{B_t \leq RPO - B_p\}.$$

As B_p and the RPO are project decision parameters, the cloud designer must evaluate the behavior of the backup process to check the P_{RPO} metric.

5.2. Recovery Time Evaluation

The process of checking the survivability in terms of RTO is similar to the recovery point evaluation. While the last considers the worst-case scenario, the recovery time evaluation is calculated directly. The probability of the recovery time meet the objective (P_{RTO}) is calculated as follows:

$$P_{RTO} = P\{R_t \leq RTO\}.$$

Where R_t denotes the recovery time.

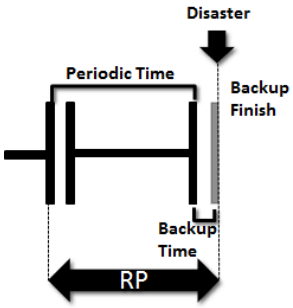

Figure 6: RPO worst case scenario.

6. MODELING APPROACH AND BASIC MODELS

This section presents the adopted hierarchical modeling to evaluate system dependability. The proposed approach (Figure 7) adopts SPN models for estimating survivability parameters in IaaS clouds. Although this work is focused on cloud computing systems, the approach is generic enough to be applied in other disaster recovery systems.

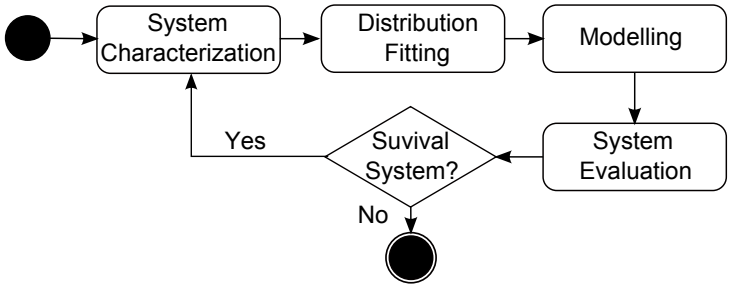

Figure 7: Methodology

The methodology's first step concerns the system characterization. In this phase, the designer must specify the VM characteristics and the number of VMs must be periodically backed up. In this activity, the probable data center locations should be selected and the transfer connection characteristics between the data centers and BS must be collected. Additionally, the backup period should be selected. In the second step, the designer must estimate what is the most suitable expolinomial distribution (Section 4.1) to represent the time to transfer a single VM data. Considering the modeling activity, the model is created with the parameters collected in the previous activities. Finally, the system is evaluated using transient evaluation, which is adopted to observe the behavior of the disaster recovery mechanism along the time [7]. If the evaluated system does not meet the constraints, the disaster recovery mechanisms must be reconfigured (e.g., backup period, data center locations). Henceforth, the following operators are adopted for assessing survivability metrics: $P\{exp\}$ estimates the probability of the inner expression (exp); and $\#p$ denotes the number of tokens in place p.

6.1. RPO Evaluation Model

The RPO evaluation model is presented in Figure 9. It represents the backup process in which a data center transmit VM images to BS. This model is composed of three main sub-models: (i) Data Center Backup, (ii) Transmission and (iii) Store. The first sub-model represents the start of backup operation in a given data center. A token in *BSTR* means that the backup process should start and the firing of *BK_ST* leads to the creation of new tokens (VMs) representing new VM images to be backed up. The

new tokens are stored in *DC_TMT*.

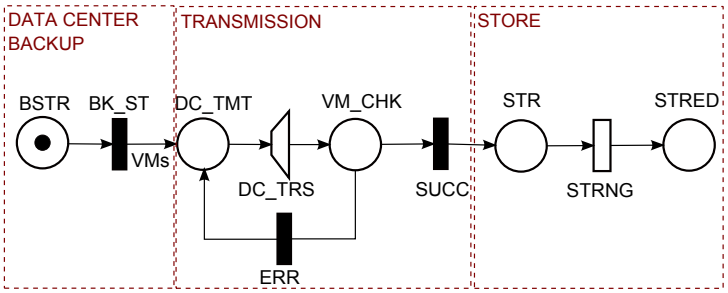

Figure 8: RPO Model

In Transmission sub-model, *DC_TRS* represents the transmission of VMs from the data center to BS. It is important to stress that the moment matching approach (Section 4.1) is adopted to represent expolinomial distributions for this transition. Once the VMs are transmitted, the data integrity of each VM is checked (*VM_CHK*). If the process present errors, the process is restarted (*ERR*). In case of correct transmission, the VMs should be stored/replicated (Store sub-model). Finally, the Store sub-model represents the storing/replication process of the transmitted VMs. It is composed of two places, one that represents the VM images that are about to be stored (*STR*) and another one to model the saved images (*STRED*). The transition *STRNG* models the store/replication process. If the BS has not replication mechanisms, *SUCC* may be connected directly to *STRED*. In this case, STR and STRNG may be discarded. In order to evaluate the system survivability in terms of RPO, a transient evaluation must be performed adopting the metric $P\{\#STRED = VMs\}$in the time $RPO - B_p$. In other words, we are interested in evaluate the probability of finish the backup process in a specific time ($RPO - B_p$). If the assessed probability is less than the user defined level, the system is not survival.

6.2. RTO Evaluation Model

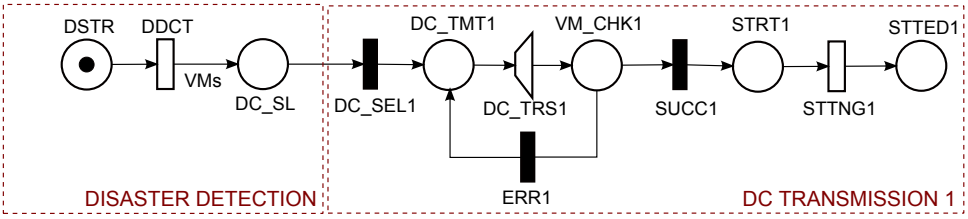

Figure 9: RTO Model

Figure 9 shows the RTO evaluation model which represents the recovery process immediately after the disaster occurrence. The model is composed of two basic sub-models, one representing the disaster detection and other modeling the transmission of VM images to operational data centers. The first sub-model is composed of a place that represents the start of recovery process (*DSTR*), a transition which models the disaster detection (*DDCT*) and a place that denotes the data center selection (*DC_SL*). The number of VM images that will be recovered is represented by VMs (arc multiplicity from *DDCT* to *DC_SL*). The other components of DC Transmission block are analogous to the components of RPO model (Figure 9) and will not be explained in details. The difference is that the three last components (STRT1, STTNG1 and STTED1) represents virtual machine instantiation instead of data store. Similarly to RPO evaluation model, a transient evaluation should be conducted to assess the probability to recover the system respecting the RTO limit. The observed metric is $P\{\#STTED1 = VMs\}$ in time RTO. If the evaluated probability is less than the requirement, the system is not survival.

7. CASE STUDY

In order to demonstrate the feasibility of the proposed work, this work presents a case study which evaluates survivability parameters in a IaaS environment. The environment is composed of five data centers and a BS. The data centers are located in the following cities: (i) New York (USA), (ii) Rio de Janeiro (Brazil), (iii) Zurich (Switzerland), (iv) Vienna (Austria) and Sydney (Australia). The backup server is located in Ilmenau (Germany). This experiment evaluates the recovery and backup process by using the modelling approach presented in Section 6. In this case study, we assume that 512 MB should be transmitted to BS to synchronize the VM data (backup process) and each VM image has 4 GB (recovery process). To estimate the time to transmit the VM data between the data centers and BS, a measurement process has been conducted to characterize the transfer rate between the backup server and the data centers. Mercury-ASTRO [25] and TimeNET [26] tools have been adopted to perform the evaluation. The transfer rates between BS and each data center is presented in Table 1.

Table 1: Transfer rates between BS and each data center.

Data Center Location	Rate (MB/s)	Standard Deviation (MB/s)
Zurich	2.0701	0.4019
Vienna	1.7412	0.3270
New York	1.1253	0.2126
Rio de Janeiro	0.6859	0.1351
Sydney	0.5659	0.1088

To evaluate the recovery process, the behavior of each data center to receive and instantiate five VM images (4 GB each) is considered. The mean time to detect the disaster is 30 minutes and the mean instantiation time is five minutes. The transmission success probability considered is 99.9%. For this particular experiment, the transition *DC_TRS* was converted to Hypoexponential subnets for all data centers (Section 4.1). The evaluation results for each data center are presented in Figure 10 and some important points are summarized in Table 2. For instance, considering that the RTO is two hours, the data center located in Austria can be a good option to recover the service if we assume that probability to recovery should be higher than 93%. On the other hand, if the RTO is four hours and the minimum probability to recovery is 99%, all data centers can be adopted to restart the affected VMs.

Table 2: Probability to recover the service for different data centers

Time(h)	Zurich	Vienna	New York	Rio de Janeiro	Sydney
1.0	0.5539	0.4370	0.2028	0.0027	0.0003
2.0	0.9369	0.9187	0.9183	0.6636	0.5424
3.0	0.9946	0.9909	0.9837	0.9501	0.9326
4.0	0.9998	0.9989	0.9973	0.9913	0.9903

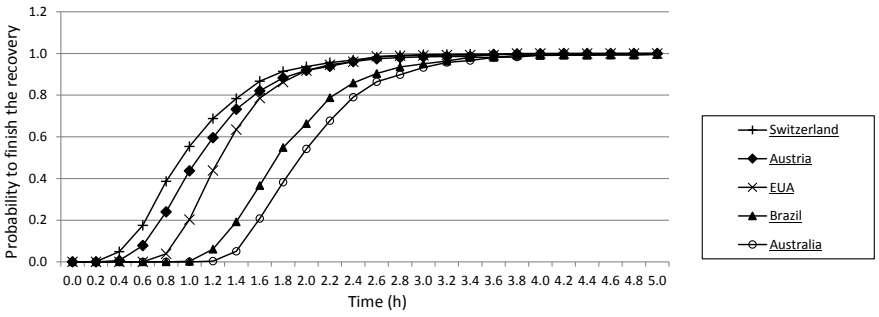

Figure 10: Recovery probability along the time

A similar evaluation was performed considering the backup process. In this case, each data center

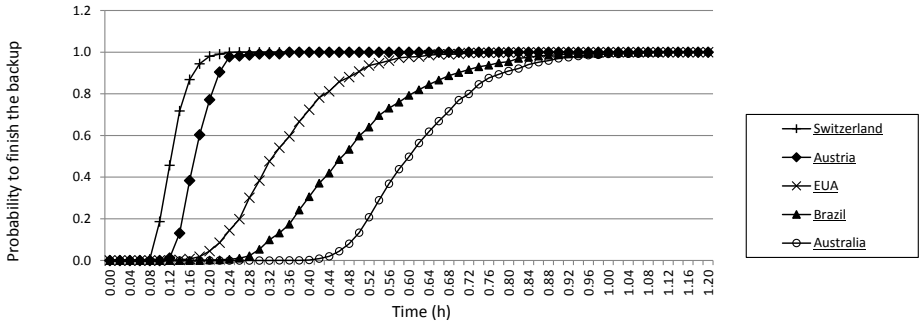

Figure 11: Backup probability along the time

synchronizes the data of five VMs (512 MB each) to BS and the replication process takes one minute. Figure 11 presents the evaluation results. Additionally, Table 3 shows important points considered in this evaluation. For the worst case scenario, if the difference between the RPO and the backup period is 0.2 hours and minimum probability to backup the VMs is equal to 0.98, only the data center of Zurich can be adopted. On the other hand, if the difference between the RPO and the backup period is one hour and minimum probability is equal to 0.99, all data centers respect the requirement.

Table 3: Probability to backup the service for different data centers

Time(h)	Zurich	Vienna	New York	Rio de Janeiro	Sydney
0.2	0.9801	0.7715	0.0449	0.0019	0.0001
0.4	0.9999	0.9998	0.7235	0.3062	0.0019
0.6	∼1.000	∼1.000	0.9771	0.7928	0.4980
0.8	∼1.000	∼1.000	0.9988	0.9545	0.9100
1.0	∼1.000	∼1.000	∼1.000	∼1.000	0.9911

8. CONCLUSION

This work presented models for survivability evaluation of cloud computing systems deployed into geographically distributed data centers. The proposed technique allows the survivability assessment taking into account the distance between data centers, RPO and RTO requirements. 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, backup time and disaster recovery requirements on system survivability. As future research, we intend to create a tool to conduct the methodology steps automatically.

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. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[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. [Online]. Available: http://dx.doi.org/10.1007/s13174-010-0007-6

[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] *Hyper-V Live Migration over Distance*. [Online]. Available: http://goo.gl/GzlkNk

[7] 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.

[8] 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.

[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. [Online]. Available: http://dx.doi.org/10.1109/PRDC.2010.30

[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. [Online]. Available: http://doi.acm.org/10.1145/2090181.2090185

[11] "Open source private and hybrid clouds from Eucalyptus," http://www.eucalyptus.com.

[12] 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. [Online]. Available: http://doi.acm.org/10.1145/1254810.1254834

[13] 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.

[14] 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.

[15] L. Cloth and B. R. Haverkort, "Model checking for survivability!" in *Quantitative Evaluation of Systems, 2005. Second International Conference on the*. IEEE, 2005, pp. 145–154.

[16] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, April 1989.

[17] A. Marsan, *Modelling with generalized stochastic Petri nets*, ser. Wiley series in parallel computing. Wiley, 1995.

[18] K. Trivedi, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*, 2nd ed. Wiley Interscience Publication, 2002.

[19] R. German, *Performance Analysis of Communication Systems with Non-Markovian Stochastic*

Petri Nets. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[20] A. A. Desrochers and R. Y. Al-Jaar, *Applications of Petri nets in manufacturing systems: modeling, control, and performance analysis*. IEEE press Piscataway^ eNJ NJ, 1995, vol. 70.

[21] "Business continuity management: Bs25999-1," British Stansdard, pp. 2–3, 2006. [Online]. Available: https://www.bsigroup.com/

[22] T. Wood, E. Cecchet, K. Ramakrishnan, P. Shenoy, J. Van der Merwe, and A. Venkataramani, "Disaster recovery as a cloud service: Economic benefits & deployment challenges," in *2nd USENIX Workshop on Hot Topics in Cloud Computing*, 2010.

[23] K. Keeton, C. A. Santos, D. Beyer, J. S. Chase, and J. Wilkes, "Designing for disasters." in *FAST*, vol. 4, 2004, pp. 59–62.

[24] Q. Yang, W. Xiao, and J. Ren, "Trap-array: A disk array architecture providing timely recovery to any point-in-time," in *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2. IEEE Computer Society, 2006, pp. 289–301.

[25] 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*, 2012.

[26] 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.