Model-Based Sensitivity of a Disaster Tolerant Active-Active GENESIS Cloud System

Tuan Anh Nguyen[1], Xuhua Rui[1], Damsub Lim[1], Jun Oh[1], Dugki Min[1](✉),
Eunmi Choi[2], Tran Duc Thang[3], and Nguyen Nhu Son[3]

[1] School of Computer Science and Engineering, Konkuk University,
Seoul, South Korea
`{anhnt2407,abealasd,miaow89,ohkakaru,dkmin}@konkuk.ac.kr`
[2] School of Management Information Systems, Kookmin University,
Seoul, South Korea
`emchoi@kookmin.ac.kr`
[3] Institute of Information Technology,
Vietnam Academy of Science and Technology, Hanoi, Vietnam
`{thang,nnson}@ioit.ac.vn`

Abstract. Modern cloud computing systems are prone to disasters. And the true cost due to service outages is reportedly huge. Some of previous works presented the use of hierarchical models: fault tree (FT), reliability block diagram (RBD) along with state-space models: continuous time Markov chain (CTMC) or stochastic petri nets (SPN) to assess the reliability/availability of cloud systems, but with much simplification. In this paper, we attempt to propose a combinatorial monolithic model using reliability graph (RG) for a real-world cloud system called general purpose integrated cloud system (GENESIS). The system is designed in active-active high availability configuration with two geographically distributed cloud sites for the sake of disaster tolerance (DT). We then present the model-based comprehensive analysis of system reliability/availability and their sensitivity. The results pinpoint different findings in which the architecture of active-active and geographically dispersed sites with appropriate interconnections of the cloud apparently enhance the system reliability/availability and assure disaster tolerance for the cloud.

Keywords: Disaster tolerance · High availability · GENESIS
Reliability graph

1 Introduction

Disaster tolerance for cloud computing system is of paramount importance for business continuity in many of internet enterprises nowadays. The physical computing system may suffer different failures leading a long period of outage because of component malfunction, man-made faults or even in more severe manner, system cascading failures and natural-cause disasters [1]. According to

Y. Chen and T. Q. Duong (Eds.): INISCOM 2017, LNICST 221, pp. 228–241, 2018.
https://doi.org/10.1007/978-3-319-74176-5_20

a research report [2], a data center suffers in average a rising of downtime cost to \$740,357 from \$500,000 (increased 38%). In addition, within the clusters of 1800 servers used as the building block of Google's IT infrastructure, about 1000 failures of individual machines happen in the first year of each cluster; also thousands of hard drives fails and it takes almost \$300 to repair each of these failures [3]. In a statistics in 2013 of 13 major cloud providers [4], the cloud service of these enterprises suffered 7.5 h outage per year in the period 2007–2012 with a huge amount of downtime cost at about 70 millions USDs estimated under hourly costs accepted in industry. Individually, Amazon as one of the largest cloud providers suffered 49 min downtime for its electronic commerce online shopping website (at https://www.amazon.com/) on Jan. 31, 2013 and the cost amounted more than \$4 million in lost sales [5]. Recently, Facebook, a completely online-service company, has annouced its revenue of first-quarter 2017 at about \$8,032 millions [6], which roughly implies that for every hour of online-service outage, the Facebook is incurred almost \$3 millions in sales. Thus, understanding the true cost of cloud outages is crucial for online-service enterprises. In this concern, designing cloud system for continuous and disaster tolerance (DT) is one of critical business strategies to reform their information infrastructure. Comprehending the reliability/availability of a cloud system is thus, essential for internet enterprises to either not overestimate/underestimate the cost of service downtimes and to determine correctly the amount of investment to be made in cloud services in order to create effective Service Level Agreement (SLA) with cloud providers. The ICT enterprises are demanding to perform comprehensive studies on modeling and evaluation of different scales and architectures of cloud systems.

Since 2012, a general purpose integrated cloud system (GENESIS) has been developed in a cooperative project between Konkuk University, Seoul, South Korea and Busan University, Busan, South Korea. The aim of GENESIS cloud is to build up a distributed/parallel platform for streaming big data processing for logistics and transportation applications. For the sake of constant big-data processing, the cloud system architecture needs DT to eliminate various types of severe failures (including man-made, system-originated or natural ones) and thus to enhance the system's reliability/availability. The cloud is architected based on active-active high availability (HA) configuration [7] associated with disaster tolerant design, in which the cloud architecture comprises of two geographically dispersed and identical sites. The cloud is furthermore, designed to be capable of provisioning and scheduling the cloud resources in response to the amount of users' requests coming over time. In this paper, we take GENESIS as a typical active-active cloud system for DT in modeling and analysis to evaluate the system's reliability/availability.

Many of previous work demonstrated the evaluation of system reliability/availability using stochastic models. A variety of previous works [8–12] typically present comprehensive studies on availability modeling and analysis of virtualized server systems (as basic computing blocks) in cloud/data centers with detailed incorporation of failure modes and different recovery strategies.

Trivedi *et al.* [13] shows various availability assessment using stochastic models of real-world systems such as from Cisco, Sun Microsystem. Most of the above-mentioned works demonstrate the use of state-space models like continuous time Markov chain (CTMC) or stochastic reward net (SRN) to capture detailed system behaviors and operational processes. Some other works attempt to compose hierarchical models using combinatorial models and state-space models to resolve the state-space explosion problem in modeling of complex systems. Smith *et al.* [14] presented a thorough study using hierarchical modeling of fault tree (FT) and CTMC for reliability evaluation of blade server systems in data centers. Maciel *et al.* in [15–19] proposed various hierarchical models using reliability block diagram (RBD) and CTMC or stochastic petri net (SPN) for the evaluation of different measures of interest including reliability, availability, survivability and performability of simplified cloud computing systems. Nguyen *et al.* [20] used reliability graph (RG) and SRN to model and evaluate the availability of a typical software defined network (SDN). Preliminary solutions and architecture designs to achieve HA and DT were presented and discussed in a number of works [1,7,21–25]. But very few previous works attempted to model cloud system for DT. Nguyen *et al.* [1] proposed a monolithic SRN for a comprehensive availability assessment of data center for DT. Silva *et al.* [26,27] also used SRN to evaluate the dependability and performability of IaaS designed for disaster tolerant cloud systems. Andrade *et al.* [28] recently presented a detailed modeling of a disaster-recovery-as-a-service solution using the combination of RBD and SRN models. We find a demanding and stimulating fact to conduct a comprehensive study to model a real-world cloud system for DT using RG. The RG is much capable of capturing the overall architecture and networking inside a complex cloud system. Therefore, in this work we attemp to propose a monolithic RG model for the thorough reliability/availability assessment of a real-world disaster tolerant cloud system.

The main contributions of this paper are summarized as follows:

- Propose a stochastic model for the reliability/availability evaluation of a typical cloud system in practice using RG
- Assess thoroughly the active-active cloud for DT with regards to important measures of interest including reliability, Steady State Availability (SSA) and sensitivity of SSA
- Elaborate several findings to help guide the design and development of disaster tolerant cloud systems in practice

The rest of this paper is organized as follows. Section 1 presents a preliminary introduction of cloud-based system for DT. Section 2 introduces a typical architecture of the GENESIS cloud system in consideration. Section 3 presents the modeling of the system. Section 4 presents the numerical results and system assessment. We discuss limitations and feasible directions for the extension of this work in Sect. 5. Section 6 concludes this paper.

2 A Disaster Tolerant Active-Active Cloud System

Figure 1 presents the overall architecture design of the disaster tolerant active-active GENESIS cloud system. The physical system consists of two identical computing sites which are both connected to distant customers. In order to perform DT for the system, the two sides are geographically distributed in distant areas which are far enough to avoid any type of disasters on both sides simultaneously. In addition to fault/disaster tolerance of the both sites, the system is designed in active-active HA configuration in which both sites operate and serve customers independently in order to proactively respond to the varying load of the changing number of users connected to the cloud, and also to probably enhance the system's availability/reliability for cloud users.

The architecture design and operations of both sites are identical. The cloud system in a site consists of four composite divisions including (i) cloud security, (ii) cloud management system, (iii) cloud computing system, and (iv) cloud storage system. When customers access to GENESIS cloud, the requests are delivered to a cloud gateway. A traffic monitor (server) is attached to the gateway to examine and control the user traffics coming into the cloud. The user requests could be directed to the remaining cloud site through the backbone network if the current cloud site undergoes a downtime period or the cloud site does not respond properly to the user requests. After passing the security examination through the firewall, the user requests come to the cloud frontend server. At this time, the user requests have passed the cloud security part and thus the customers can interact with the cloud. The customers now can define their own configuration of the computing system in demand through the cloud management system. Based on the cloud configurations and the number of requests coming to the cloud frontend server, a virtual cloud (VC) manager communicates with the servers of VC provisioning system and VC scheduling system in order to provision/schedule Virtual Machine (VM)s running on a provisioned physical server system. All the requests then are delivered to a cloud management system. This server manages the cloud computing system which comprises of GENESIS cloud system (a cluster of physical computing nodes) and GENESIS cloud database system (a cluster of database nodes). This farm of cloud servers connect to the cloud management server via a central switch with very HA. A load balancer is attached to the cloud farm in order to balance the processing load among the servers. Every computing nodes and database nodes are also interconnected to each other via the central switch in order to process computing jobs with high performance. The system is designed so that an application either for computing or To assure and enhance HA for the GENESIS cloud system, an automated cloud computing node creator is provided to create new computing node (physical server) and install all pre-defined software and environment as initial. There is also an automated cloud storage creator with the same purposes attached to the GENESIS cloud database system. With these automatic creators, we expect to enhance the cloud system's availability and readiness. At the border of the cloud system architecture, the cloud storage system is designed for HA consisting of a storage area network (SAN) and a storage system. For the sake of DT for physical cloud system, two specific networking connections between the two sites are established, one connects the two cloud computing systems via the two central switches and the other connects the two storage systems via the two SANs. The two connections use different technologies, (i) overlay transport virtualization (OTV) [29] for business resiliency to deploy multiple geographically disparate cloud systems; and (ii) switched virtual circuit (SVC) for temporary data transmission between the two storage systems to get the data most updated and synchronized. We will use the above-described cloud system architecture for the modeling and analysis using RG model. For the ease of stochastic modeling in this paper, we assume to limit the number of computing servers and database servers in the cloud computing system at a small number. The detailed description of the system components is summarized in Table 1. In the following sections, we present the overall model of the whole cloud system.

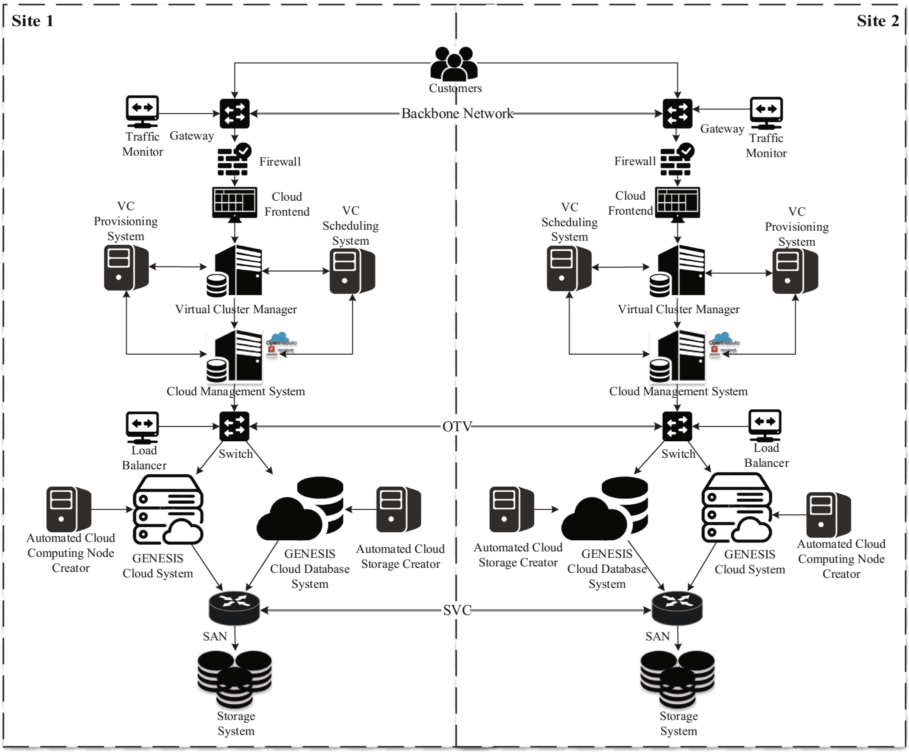


Fig. 1. A disaster tolerant active-active cloud system

3 Reliability Graph Models

Figure 2 illustrates the RG model of the overall two-sites disaster tolerant active-active GENESIS cloud system. The RG model basically consists of two types of elements which are nodes and edges connecting the two different nodes. A node can have a certain number of input/output edges but only two special nodes are different which are sink node (S) with no input edges and destination node (D) with no output edges. An edge in the RG model represents for a certain component of the system. A failure/recovery of a certain component in the system corresponds to the breakage/continuity of the respective edge in the RG model. The information of the system component such as mean time to failure (MTTF) and mean time to repair (MTTR) thus, can be attached to the edges in order to compute the measures of interest like reliability/availability. The whole cloud system is considered reliable/available if the user requests can be dilivered from the customers to the cloud computing division which in turn frequently accesses the storage systems. In term of modeling, the there exists at least one uninterrepted path connecting the sink node and the destination node throughout the edges and normal nodes.

The RG model is constructed strictly based on the overall system architecture. Thus, as shown in Fig. 2, the model also comprises of two branches from sink node to destination node. The connections between the sink node (S) and the nodes (1) and (2) as well as the connection between the destination nodes (D) and the nodes (19) and (48) represent for the connections of users to the two sites which are supposed to be always available. The notation on each edge in RG (summarized in Table 1) depicts the corresponding component of the system residing either in site 1 or site 2. In the cloud system, we assume that there are n_{cS} number of cloud servers hosting n_{App} number of cloud applications in overall. Also correspondingly, there are n_{DB} number of database applications running on respective n_{cDB} cloud database servers. We assume that a certain app can run with multiple instances on different servers. Thus there is no dependences between the edges in the RG models. It is noted to consider the notations in the RG model. The edges with their notations as follows, cS_{ij}, App_{ik}, DB_{ig} and cDB_{il} (where, $i = 1, 2$; $j = 1, n_{cS}$; $k = 1, n_{App}$; $g = 1, n_{DB}$ and $l = 1, n_{cDB}$) respectively represent for the cloud server j, cloud application k, database application g and cloud database server l in the corresponding cloud site i. As shown in the RG model, the model is symmetrical because the two cloud sites are identical. Thus the upper part of the RG model represents for the modeling of the cloud site 1 and the lower part of the RG model is for the cloud site 2.

4 Numerical Results

In this section, we present the evaluation results of the cloud system. The RG model in Fig. 2 is implemented and analyzed using Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) [30,31] developed by DUKE University, USA. The default values of input parameters for the model

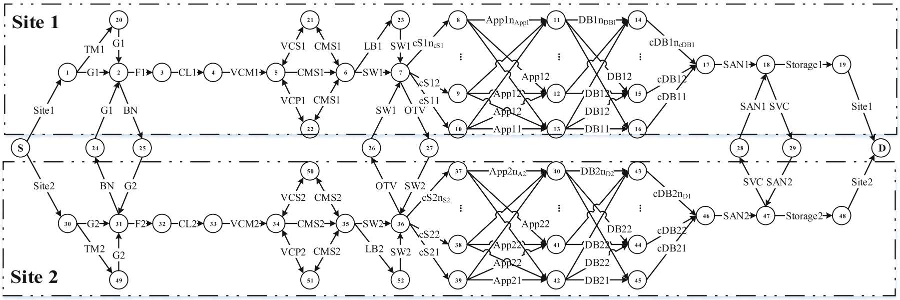


Fig. 2. Reliability Graph Model of the cloud system

are given in Table 2. To realize the pros and cons of the proposed cloud architecture for DT, in this paper, we attempt to evaluate the two cloud systems as follows:

- *Case I*: the cloud system has only one cloud site with no other redundant sites. This system is modeled by either the upper/lower part of the RG model in Fig. 2.
- *Case II*: the cloud system comprises of two redundant cloud sites as proposed in Fig. 1. This system is modeled exactly by the overall RG model in Fig. 2.

Our measures of interest are:

- Reliability of the two cases over time, as shown in Fig. 3
- SSA of the two cases under default values of input parameters, as shown in Table 3
- Sensitivity of SSA with respect to major impacting parameters including λ_{cS} and λ_{cDB}, as shown in Fig. 4a; and λ_{BN}, λ_{OTV} and λ_{SVC}, as shown in Fig. 4b.

Relaibility Analysis: Fig. 3 depicts the reliability analysis results of the cloud system in two cases (i) 1 site and (ii) 2 sites. The gap between the two graphs clearly shows that the system with two redundant sites is much higher reliable over time in comparision with the cloud with only one site. Particularly, the reliability of the system with one site exposes a much faster decaying rate in accordance with the high slope of the respective reliability graph (which is not desired in reliability engineering).

Steady State Analaysis: Under the default parameters, we perform SSA analysis for the cloud systems in the two cases as shown in Table 3. Apparently, the cloud with two sites gains much higher availability (four nines after decimal point) than the one with only one site does (only two nines after decimal point). As we compute downtime minutes for the cloud systems, the two-sites cloud undergoes only about 38 min in a year whereas the another one suffers from 5129 min of outage per year.

Sensitivity analysis of SSA: Fig. 4a shows the variation of the cloud's SSA with respect to major impacting factors in consideration. Figure 4a illustrates the

Table 1. Description of system components and abbreviation in RG system model

Abbreviation	Name	Description
G1, G2	Gateway (switch)	A configured switch specifically receives requests from outer customers
TM1, TM2	Traffic monitor	A server is in charge of monitoring and controlling the amount of traffics flowing into the cloud
BN	Backbone network	A specific network connection between the two sites of the cloud system in order to abstract the physical cloud system with respect to the users
F1, F2	Firewall	A firewall mechanism to examine the coming user requests for security
CL1, CL2	Cloud frontend	A server runs a GUI for user interactions in demanding desired cloud configuration
VCM1, VCM2	Virtual cluster management system	A server runs a cloud management software to configure and request for cluster formation of physical servers
VCS1, VCS2	Virtual cluster scheduling system	A server runs a software to schedule the realistic cluster of servers based on statistic prediction to match with the users' varying load
VCP1, VCP2	Virtual cluster provisioning system	A server runs a prediction software to provision physical server farm to fit with the varying coming requests of customers at a time based on statistic user data
CMS1, CMS2	Cloud management system	A server runs a management software to control and monitor the whole cloud computing system and cloud database system
SW1, SW2	Switch	Cloud central switches connect the two divisions: cloud management system with cloud computing system and also to interconnect the cloud server farm and the cloud database servers
LB1, LB2	Load balancer	A server to balance the processing loads among the computing nodes
OTV	Overlay transport virtualization	A network connection with OTV technology for DT between the two cloud physical infrastructure
cS	Cloud physical server	A server in the cloud farm in charge of processing the user requests
App	Cloud application	A requested application running on a cloud server
DB	Database application	A corresponding database application running along with a certain cloud application
cDB	Cloud database server	A server in the cloud database farm in charge of hosting database applications
SAN1, SAN2	Storage area network	A high-speed network for block-level access to storage system
SVC	Switched virtual circuit	A network connection between the two networks of SANs for data synchronization of storage systems between the two sites
Storage1, storage2	Storage system	A redundant storage system with HA configurations

dependence of the SSA on the MTTFs of cloud servers and cloud database servers (λ_{cS} and λ_{cDB}). As shown in the figure, the two-sites cloud gains clearly higher SSA over time than the one-site cloud does. The distant differences of the graphs are particularly huge in the early range of small values and reduce quickly as the values of the parameters increase. This implies that even for the farm of normal physical servers with low values of MTTF (which may suffer downtime periods more frequently), the architecture of two-sites cloud apparently boosts the system availability compared to the one of one-site cloud. Figure 4b depicts the SSA variation on the MTTFs of the network connection BN, OTV and SVC (λ_{BN}, λ_{OTV} and λ_{SVC}). We observed that, as long as any of the network connection for DT stay available (MTTFs increase), the overall availability of the two-sites cloud is accordingly enhanced. Furthermore, the OTV connection which inter-connects the two central switches of the cloud computing server farm and cloud database server farm is more sensitive than the remaining connections in which a small change of its MTTF (especially in the early period) can cause an apparent variation of the SSA. In comparison, a failure of the BN connection does not pull down the SSA as much as a failure of SVC or OTV connection does. Thus, we realize that the failure of OTV connection causes the most severe consequence to the system's availability. When we compare the two figures Fig. 4a and b, we find that even though the effects of the MTTFs on the system's SSA are similar, but the values of the SSA are much different. In particular, a small change of MTTFs of cloud server and cloud database servers (λ_{cS} and λ_{cDB}) leads to a big difference of the SSA. Whereas an increase or decrease of the MTTFs of the network connection BN, OTV or SVC causes a small change in value of the SSA, in comparison with the previously-mentioned case.

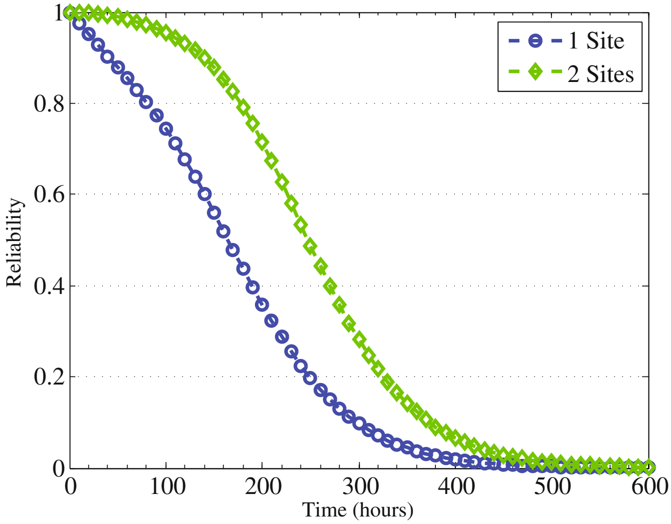


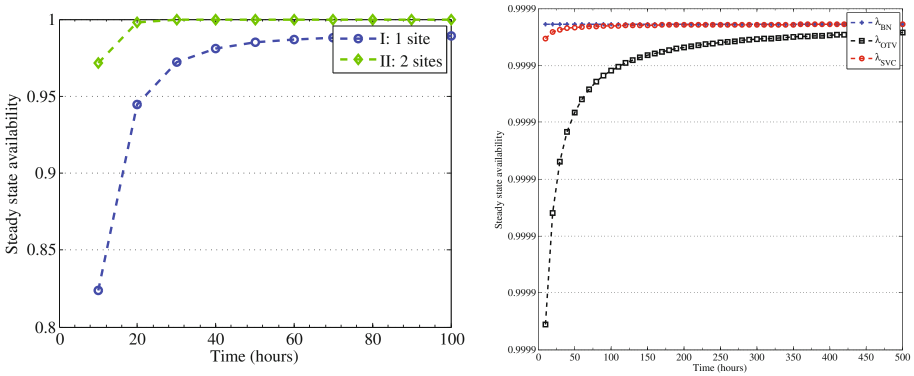
Fig. 3. Reliability analysis

Table 2. Default input parameters

Notation	Description	Value
$1/\lambda_{App}$	Mean time to failure of an application on a server	5 days
$1/\lambda_{BN}$	Mean time to failure of backbone network	3 months
$1/\lambda_{CL}$	Mean time to failure of a cloud frontend server	1.5 months
$1/\lambda_{CMS}$	Mean time to failure of a server of cloud management system	3 weeks
$1/\lambda_{DB}$	Mean time to failure of a database software	3 days
$1/\lambda_{F}$	Mean time to failure of a firewall component	3 months
$1/\lambda_{G}$	Mean time to failure of a gateway component	8 months
$1/\lambda_{LB}$	Mean time to failure of a load-balancer	2 weeks
$1/\lambda_{OTV}$	Mean time to failure of overlay transport virtualization (OTV) connection	5 months
$1/\lambda_{SAN}$	Mean time to failure of a SAN	9 months
$1/\lambda_{SW}$	Mean time to failure of a switch	1 year
$1/\lambda_{Storage}$	Mean time to failure of a storage system	6 months
$1/\lambda_{SVC}$	Mean time to failure of switched virtual circuit (SVC) connection	4 months
$1/\lambda_{TM}$	Mean time to failure of a traffic monitor server	2 weeks
$1/\lambda_{VCM}$	Mean time to failure of a virtual cluster manager server	2 months
$1/\lambda_{VCP}$	Mean time to failure of a virtual cluster provisioning server	3 weeks
$1/\lambda_{VCS}$	Mean time to failure of a virtual cluster scheduling server	3 weeks
$1/\lambda_{cDB}$	Mean time to failure of a cloud database server	2 months
$1/\lambda_{cS}$	Mean time to failure of a cloud server	2 months
$1/\lambda_{DB}$	Mean time to recovery of an application on a server	10 min
$1/\mu_{BN}$	Mean time to recovery of backbone network	5 h
$1/\mu_{CL}$	Mean time to recovery of a cloud frontend server	2 h
$1/\mu_{CMS}$	Mean time to recovery of a server of cloud management system	3 h
$1/\mu_{DB}$	Mean time to recovery of a database software	30 min
$1/\mu_{F}$	Mean time to recovery of a firewall system	2 h
$1/\mu_{G}$	Mean time to recovery of a gateway component	2 h
$1/\mu_{LB}$	Mean time to recovery of a load-balancer	8 h
$1/\mu_{OTV}$	Mean time to recovery of overlay transport virtualization (OTV) connection	7 h
$1/\mu_{SAN}$	Mean time to recovery of a SAN	5 h
$1/\mu_{SW}$	Mean time to recovery of a switch	8 h
$1/\mu_{Storage}$	Mean time to recovery of a storage system	3 h
$1/\mu_{SVC}$	Mean time to recovery of switched virtual circuit (SVC) connection	3 h
$1/\mu_{TM}$	Mean time to recovery of a traffic monitor server	1 h
$1/\mu_{VCM}$	Mean time to recovery of a virtual cluster manager server	8 h
$1/\mu_{VCP}$	Mean time to recovery of a virtual cluster provisioning server	6 h
$1/\mu_{VCS}$	Mean time to recovery of a virtual cluster scheduling server	6 h
$1/\mu_{cDB}$	Mean time to recovery of a cloud database server	8 h
$1/\mu_{cS}$	Mean time to recovery of a cloud server	8 h
n_{cS}, n_{App}, n_{DB}, n_{cDB}	Respectively, numbers of cloud servers, cloud applications, cloud database applications and cloud database servers in each site of the system	3

Table 3. Steady state analysis

Case	Description	SSA	Downtime (minutes/year)
I	Cloud system with 1 site	0.990240932	5129
II	Cloud system with 2 sites	0.999927727	38



(a) λ_{cS} and λ_{cDB}

(b) λ_{BN}, λ_{OTV}, and λ_{SVC}

Fig. 4. Sensitivity analysis of SSA wrt. some major impacting parameters

5 Limitations and Extentions

We find an open future research avenue for the reliability/availability of disaster tolerant cloud based systems. The extensions of this work can go in several directions.
Scalability: This work takes into consideration only a small scale system for cloud-based DT. The number of physical nodes in server farm is limited for the sake of modeling and analysis. Nevertheless as the rapid advancement of the real-world data centers and cloud computing systems, it is required to consider the system with a large number of servers or even a multiple number of redundant sites in the cloud system.

Hierarchical modeling: The RG model in this work captures the general architecture and configuration of the cloud system. Nevertheless, we also need to capture the operational states of every components and processes. This direction usually encounters with the largeness problem in modeling in which the hierarchical modeling is an efficient approach for the future extension of this work.

Security: Security for cloud currently is of paramount importance. Although the cloud architecture in this work takes into account security elements in design, but the modeling and analysis for security along with availability/reliability needs an appropriate extension.

6 Conclusion

This paper presented a reliability/availability modeling and analysis of an active-active cloud system for DT. The typical cloud physical system consists of two sites which are geographically dispered and interconnected for the sake of DT. The modeling using RG is the main approach to capture the overall system architecture. We carried out a comprehensive evaluation of system reliability, SSA and the sensitivity of SSA with respect to MTTFs of physical servers and interconnecting network connections between the two cloud sites. The results showed that the proposed architecture of active-active two-sites cloud can assure DT and enhance overall reliability/availability. Furthermore, the analysis also figured out the importances of network connection between the two sites. This work is also planned to extend in different manners including scalability of the system, hierarchical modeling for the whole physical system and security modeling and analysis for the cloud in future work.

Acknowledgment. – This research was supported by the Vietnam-Korea cooperation project: VAST.HTQT.HANQUOC.01/17-18 managed by Vietnam Academy of Science and Technology.

– This work was supported under the framework of international cooperation program managed by the National Research Foundation of Korea (project number, FY2016K2A9A1A06925440).

References

1. Nguyen, T.A., Kim, D.S., Park, J.S.: Availability modeling and analysis of a data center for disaster tolerance. Future Gener. Comput. Syst. (2015). http://www.sciencedirect.com/science/article/pii/S0167739X15002824
2. Ponemon Institute: Calculating the Cost of Data Center Outages (Report). Technical report (2013)
3. Miller, R.: Failure rates in google data centers (Report). Technical report, Data Center Knowledge (2008)
4. Gagnaire, M., Diaz, F., Coti, C., Cérin, C., Shiozaki, K., Xu, Y., Delort, P., Smets, J.-P., Lous, J.L., Lubiarz, S., Leclerc, P.: Downtime statistics of current cloud solutions. Technical report, The International Working Group on Cloud Computing Resiliency (2013)
5. Linthicum, D.: Calculating the true cost of cloud outages (2013). http://www.infoworld.com/article/2613537/cloud-computing/calculating-the-true-cost-of-cloud-outages.html
6. Facebook: Facebook Reports First Quarter 2017 Results (2017). https://investor.fb.com/investor-news/press-release-details/2017/Facebook-Reports-First-Quarter-2017-Results/default.aspx
7. Loveland, S., Dow, E.M., LeFevre, F., Beyer, D., Chan, P.F.: Leveraging virtualization to optimize high-availability system configurations. IBM Syst. J. **47**(4), 591–604 (2008). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386515

8. Kim, D.S., Machida, F., Trivedi, K.S.: Availability modeling and analysis of a virtualized system. In: 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing, vol. 1, pp. 365–371. IEEE, November 2009. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5368189
9. Nguyen, T.A., Kim, D.S., Park, J.S.: Availability modeling and analysis of a virtualized servers system (2014). https://doi.org/10.1155/2014/165316
10. Nguyen, T.A., Min, D., Park, J.S.: A comprehensive sensitivity analysis of a data center network with server virtualization for business continuity. Math. Probl. Eng. **2015**, 1–20 (2015). http://www.hindawi.com/journals/mpe/aa/521289/ http://www.hindawi.com/journals/mpe/2015/521289/
11. Kim, D.S., Hong, J.B., Nguyen, T.A., Machida, F., Trivedi, K.S.: Availability modeling and analysis of a virtualized system using stochastic reward nets. In: Proceedings of 16th IEEE International Conference on Computer and Information Technology (IEEE CIT 2016). IEEE, Fiji (2016)
12. Machida, F., Kim, D.S., Trivedi, K.S.: Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration. Perform. Eval. **70**(3), 212–230 (2013)
13. Trivedi, K.S., Kim, D.S., Ghosh, R.: System availability assessment using stochastic models. Appl. Stoch. Models Bus. Ind. **29**(2), 94–109 (2013). http://doi.wiley.com/10.1002/asmb.951
14. Smith, W.E., Trivedi, K.S., Tomek, L.A., Ackaret, J.: Availability analysis of blade server systems. IBM Syst. J. **47**(4), 621–640 (2008). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386524
15. Costa, I., Araujo, J., Dantas, J., Campos, E., Silva, F.A., Maciel, P.: Availability evaluation and sensitivity analysis of a mobile backend-as-a-service platform. Qual. Reliab. Eng. Int. (2015). http://doi.wiley.com/10.1002/qre.1927
16. Matos, R., Dantas, J., Araujo, J., Trivedi, K.S., Maciel, P.: Redundant eucalyptus private clouds: availability modeling and sensitivity analysis. J. Grid Comput. 1–22 (2016). http://link.springer.com/10.1007/s10723-016-9381-z
17. Melo, M., Maciel, P., Araujo, J., Matos, R., Araujo, C.: Availability study on cloud computing environments: live migration as a rejuvenation mechanism. In: 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 1–6. IEEE, June 2013. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6575322
18. Sousa, E., Silva, E., Lins, F., Tavares, E., Maciel, P.: Dependability evaluation of cloud infrastructures. In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1282–1287. IEEE (2014). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6974091
19. Silva, B., Maciel, P.R.M., Zimmermann, A., Brilhante, J.: Survivability evaluation of disaster tolerant cloud computing systems. In: Proceedings of Probabilistic Safety Assessment and Management (PSAM12), p. 453, Hawaii, USA (2014). http://psam12.org/proceedings/paper/paper_453_1.pdf
20. Nguyen, T.A., Eom, T., An, S., Park, J.S., Hong, J.B., Kim, D.S.: Availability modeling and analysis for software defined networks. In: 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC), April, pp. 159–168. IEEE, November 2015. http://ieeexplore.ieee.org/document/7371859/ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7371859
21. Radhakrishnan, R., Mark, K., Powell, B.: IT service management for high availability. IBM Syst. J. **47**(4), 549–561 (2008). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386521

22. Morrill, H., Beard, M., Clitherow, D.: Achieving continuous availability of IBM systems infrastructures. IBM Syst. J. **47**(4), 493–503 (2008). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386519
23. Lumpp, T., Schneider, J., Holtz, J., Mueller, M., Lenz, N., Biazetti, A., Petersen, D.: From high availability and disaster recovery to business continuity solutions. IBM Syst. J. **47**(4), 605–619 (2008)
24. Clitherow, D., Brookbanks, M., Clayton, N., Spear, G.: Combining high availability and disaster recovery solutions for critical IT environments. IBM Syst. J. **47**(4), 563–575 (2008). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386509
25. Adeshiyan, T., Attanasio, C.R., Farr, E.M., Harper, R.E., Pelleg, D., Schulz, C., Spainhower, L.F., Ta-Shma, P., Tomek, L.A.: Using virtualization for high availability and disaster recovery. IBM J. Res. Dev. **53**(4), 8:1–8:11 (2009). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5429062
26. Silva, B., Maciel, P., Zimmermann, A.: Performability models for designing disaster tolerant Infrastructure-as-a-Service cloud computing systems. In: 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), pp. 647–652. Infonomics Society, December 2013. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6750283
27. Silva, B., Maciel, P., Tavares, E., Zimmermann, A.: Dependability models for designing disaster tolerant cloud computing systems. In: 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 1–6. IEEE, June 2013. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6575323
28. Andrade, E., Nogueira, B., Matos, R., Callou, G., Maciel, P.: Availability modeling and analysis of a disaster-recovery-as-a-service solution. Computing **99**, 1–26 (2017). http://link.springer.com/10.1007/s00607-017-0539-8
29. Santana, G.A.A.: Data Center Virtualization Fundamentals: Understanding Techniques and Designs for Highly Efficient Data Centers with Cisco Nexus, UCS, MDS, and Beyond. Cisco Press, Indianapolis (2013)
30. Trivedi, K.S.: SHARPE 2002: symbolic hierarchical automated reliability and performance evaluator. In: Proceedings of the 2002 International Conference on Dependable Systems and Networks, p. 544. IEEE Computer Society (2002). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1028975
31. Trivedi, K.S., Sahner, R.: SHARPE at the age of twenty two. ACM SIGMETRICS Perform. Eval. Rev. **36**(4), 52 (2009). http://dl.acm.org/citation.cfm?id=1530873.1530884