International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.3, No.3, June 2014, pp. 174~182
ISSN: 2089-3337

Cluster as a Service for Disaster Recovery in Intercloud Systems: Design and Modeling

Mohammad Ali Khoshkholghi, Azizol Abdullah, Rohaya Latip, Shamala Subramaniam, Mohamed Othman
Faculty of computer science and information technology
University Putra Malaysia

Article Info

Article history:

Received Apr 30th, 2014
Revised May 25th, 2014
Accepted June 11th, 2014

Keyword:

InterCloud Systems
Cluster as a Service
Disaster Recovery
Markov Chain Model
Availability

ABSTRACT

Nowadays, all modern IT technologies aim to create dynamic and flexible environments. For this reason, InterCloud has been designed to provide a vast and flexible virtualized environment in which many clouds can interact with one another in a dynamic way. Disaster recovery is one of the main applications of InterCloud which can be supported by Cluster as a Service. However, the previous studies addressed disaster recovery and Cluster as a Service separately. In addition, system backup and disaster recovery methods are not sufficiently effective in InterCloud. In this paper, we propose an InterCloud system which integrates both Cluster as a Service and disaster recovery in a harmonious manner. Also, we present a heuristic approach to select the best locations for system backup and disaster recovery in InterCloud systems. Finally, the proposed system is modeled and analyzed using Continuous-time Markov chains.

Corresponding Author:

Mohammad Ali Khoshkholghi
Faculty of computer science and information technology,
University Putra Malaysia
43400 UPM Serdang, Selangor Darul Ehsan, Malaysia
Email: khosh.kholghi63@gmail.com

1. INTRODUCTION

Cloud computing is an on-demand model in which virtualized and scalable resources are provided as services. Computational resources can be allocated to a variety of users over the Internet. On the other hand, based on the requirements, many distributed applications use cluster machines for their execution environments, such as: web services, data analysis and simulation [1]. These applications can be classified into three categories [2]:

1. Context in application such as the tools which generate on-demand environments for a continuous integration test.
2. Amount of transactions to applications such as auto scaling of web services.
3. Geographically constraint on resource for applications such as disaster recovery.

Cluster as a Service (CaaS) is a hybrid model which is created by combining cluster computing and cloud computing to gain benefits of both models. Clustering is an approach that can bring a high level of availability and increased performance [3]. Furthermore, cloud computing is a good fit for scalability, resiliency and also disaster recovery. This combination, produce a system that meets the needs of real cloud environments. In order to improve power computation, there is a need to add more nodes to each cluster. However, increasing the number of clusters - as availability zones- enhances the availability of the cloud.

Because of high scalability of clouds, by increasing the number of consumers, a cloud may not serve all the requests. In contrast, sometimes clouds do not use all their own resources in a period of time. In this case, hybrid approaches including public & private clouds or multiple different public clouds allow Cloud Service Providers (CSPs) to offer higher reliability and ability to deliver better service to their customers. In addition, researchers need to combine different clouds to create a bigger virtualized cloud environment in order to do scientific research projects. For these reasons, recently InterCloud systems have received much attraction. The InterCloud is sets of distributed resources of different organizations or individuals which collectively collaborate together to provide services. InterCloud consists of thousands of nodes dispersed around the world. Due to dynamic and changeable nature of InterCloud, nodes can be introduced or removed from the system any time.

Disasters, either man-made or natural, can lead to expensive service disruption. As a famous proverb says: "Don't put all your eggs in one basket" using one cloud site is not a right approach to the cloud systems . In fact, any single platform may lead to interruption for whole system. One of the goals of disaster recovery planning is to omit as many of single point of failure as possible. So, each cloud needs to have an extra location for risk avoidance which is geographically separated from the original site.

As a main application, InterCloud system can be used for system backup and disaster recovery. Using these services, data protection and service continuity are guaranteed for customers at different levels. In this paper, we design an InterCloud system which is able to provide cluster as a service, as well as able to guarantee data protection and service continuity in the event of a disaster. Furthermore, selecting most suitable sites for both backup and disaster recovery is addressed in this system.

The rest of this paper is organized as follows: Section 2 gives existing related work done in the area. Section 3 details the design of our proposed system including a heuristic approach to find the best locations for the backup site. In section 4, we model and analyze the proposed system using Continuous-time Markov chain. Finally, section 5 concludes this contribution and points out future work.

2. RELATED WORKS

Recently, InterCloud paradigm has attained attention of researchers. The overall goal of InterCloud is to create a computing environment that facilitates scalable provisioning of different services while achieving expected QoS under the variable workload, resource and network conditions [4]. Different aspects of InterCloud have been addressed in some studies [5-10]. However, this area is quite new so many challenges and open issues have to be addressed to reach a satisfactory level of InterCloud.

Disaster management is one of the major problems which organizations face. In huge companies, between 2% and 4% of IT budget is spent for disaster recovery planning every year. Wood et al. [11] has presented a pricing analysis to compare cloud-based disaster recovery compared to privately owned resources. The results have shown that in Small and Medium Enterprises (SMEs) using disaster recovery as a service provided by clouds is more economical.

Cloud-based disaster recovery has been discussed in some papers [12-20]. One of the initial studies in distributed disaster recovery has been presented in [21]. The authors introduced an architecture with two geographically separated sites. In the case of a disaster, the workload is redirected to the redundant site. Therefore, secondary site can deliver services to the customers as a seamless and transparent way. In [22], the authors have proposed a disaster recovery approach which guarantees the independence between cloud service provider and customers. Using this technique, taking control of data and migration between different clouds are possible.

In [23] the authors tried to improve disaster recovery mechanism with developing the replication technique combined with live VMs migrations. In this technique whole system checkpoint, are stored in a backup site with a high frequency. In [24, 25], the authors have proposed an accurate algorithm to make backup between primary and backup site. Using these algorithms, the amount of data loss is decreased dramatically in the case of disaster. PipeCloud [26], proposed a combined replication technique. This technique aims to gain both the performance of async replication and consistency of sync replication. For this purpose, both data replication between sites and processing operations are performed in parallel. Jian-Hua et al. [27] has been proposed an inter-private cloud which is shared between different private clouds. Inter-private cloud manages data backup in ordinary operations as well as system recovery in the event of a disaster. Huge amounts of disaster-related data have been generated by different organizations, government and even social media. In [28], the authors have provided a Knowledge as a Service (KaaS) framework for better disaster cloud data management.

Because of the diverse nature of cloud services, cloud providers need to provide different level of availability for different services based on their service level agreement (SLA). For this, CaaS has been introduced [29, 30]. In [2] the authors have proposed a CaaS which can be used for self-deployable applications such as disaster recovery.

Based on existing literature, there is a lack of research to investigate CaaS in InterCloud paradigm. For this reason, this paper proposes an InterCloud system which provides CaaS for different cloud customers. In this system, cluster machines can be used for both data and service recovery in the event of a disaster.

3. THE PROPOSED SYSTEM DESIGN

In this section, we present our proposed model for data backup and disaster recovery. We first describe the InterCloud system. Then, the proposed disaster recovery model and action of each part are described. At the end, we describe an approach for selecting the best disaster recovery locations.

3.1. System Description

As shown in Figure 1, the InterCloud system consists of different public and private clouds. These clouds are connected with each other to establish a big virtualized cloud environment. In addition, every single cloud consists of multiple cluster machines. These cluster machines create many VMs that serve CaaS to customers. For this reason, a two-layer CaaS design is used [2]. Using machine images, first layer manages the operating system of the physical nodes forming the clusters. Second layer, deploys the software components to install on the nodes. The cluster of each application must be securely separated in this system.

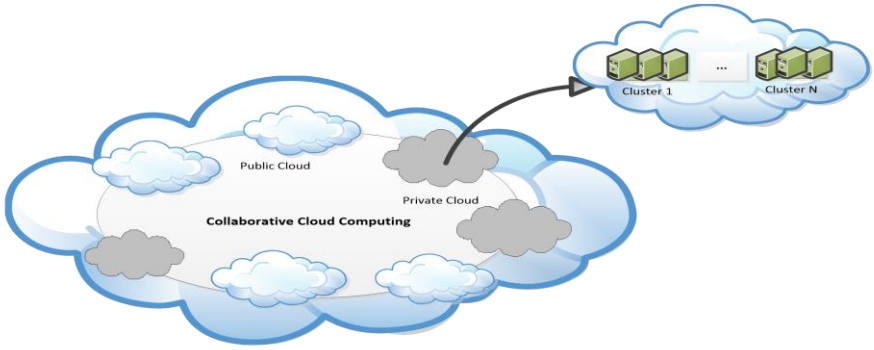

Figure 1. An InterCloud system which is supported by CaaS

3.2. Disaster Recovery Model

With the system described above, our disaster recovery model is presented in Figure 2. We use the Eucalyptus platform because it is able to increase cloud size easily. To enhance compute power, we only add node servers to existing clusters in each cloud. On the other hand, for improving availability, we can add more clusters. The maximum number of clusters in any cloud is 8 clusters with capacity to handle over 1000 VMs.

Each cloud (of N clusters) divides its own clusters in two categories: operational clusters (OC) and disaster recovery clusters (DR) where: DR= N - OC. To reduce the complexity of the system and provide a better description and analysis of the system, we assume that each cloud is composed of 2 clusters (N=2). Based on our assumption, there is one DR cluster which is shown in Figure 2. If any failure happens for OC, its workload will be assigned to DR. In this case, DR delivers the expected service to the customers until the failed cluster is recovered.

In addition, every cloud in the InterCloud system needs a backup cloud to make a system state replication, data backup and also disaster recovery. To this end, each cloud (primary site) selects another cloud as a backup site.

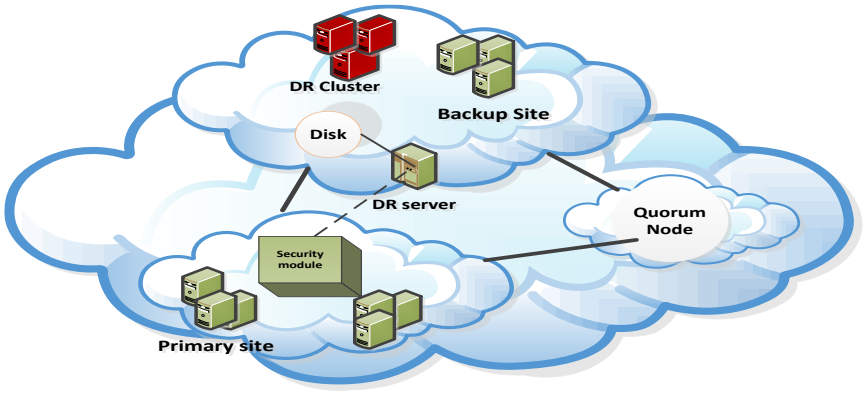

Figure 2. The proposed disaster recovery model in the InterCloud system

In a normal operation before any disaster, the cloud only needs a disk to store data and a single disaster recovery server to maintain the VMs' states. In this stage, primary site replicates only changed data and VMs' states to the backup site. However, after a disaster happens and primary site becomes inaccessible, the failed cloud traffic is redirected to the backup cloud; and DR cluster undertakes to deliver the application services. After a failed primary cloud comes back online, new data, which was created during the disaster, have to be resynchronized from the secondary site to the original site.

Failure detection is another issue which has to be taken into account in the disaster recovery plan. For this purpose, one cloud is selected as a quorum node to detect a real failure and to differentiate between link failure and host failure. Quorum node monitors and controls both primary and backup clouds by receiving frequent heartbeat messages. The primary site sends data and system checkpoints to the backup site in specific periods of time as well as receive an Ack from the backup site. However, if the backup site does not receive any checkpoint during this period, will send a heartbeat to the quorum site to know whether the primary site is safe. In this case, if the primary site has crashed, so the quorum site sends an activated message to the backup site. Instead, when either the network link failed or delay happened because of network congestion, backup site waits for a while to receive the checkpoints. In another situation, if primary site does not receive any Ack, it will send a message to the quorum node. In the case that the backup site has crashed, primary site chooses another site for making backups.

It is notable that, security and confidentiality are two major challenges in the system, because backups are distributed in the distinct organizations. The security module is designed to protect data and secure transmission over the network. The security module encrypts, scrambles, fragments and duplicates the data based on the expected level of recovery [31]. Then, data are sent to the backup site.

3.3. Disaster Recovery Site Selection

In the proposed system, each cloud needs to choose one cloud from different candidates as a disaster recovery site and also to replicate data and VMs' checkpoints. We leverage harmony resource selection [10] to propose a selection technique for our disaster recovery system. As shown in Figure 3, we define four factors to create an overall QoS:

1. Most distance reachable site: Natural disasters, lead to severe damage in a vast area. So, the backup site has to be geographically far from the primary site.

2. Highest bandwidth: In the event of a disaster - in both failover and failback procedure- massive amount of data must be transferred between the original and backup sites in a short time. Therefore, the bandwidth has an undeniable impact on successful deployment of disaster recovery strategies.

3. Low price: InterCloud consists a variety of clouds with different attributes. They have different budget for their recovery plan based on the cloud size and their applications. So, individual clouds seek cheaper places to gain disaster recovery solutions.

4. Most available disaster recovery resources: As mentioned before, each cloud allocates some clusters for disaster recovery plan (DR clusters). The number of DRs is not the same in all clouds. Furthermore, these clusters can be leased to other clouds in the absence of any disaster. Based on the expected disaster recovery plan, primary clouds can select those backup clouds which have enough resources.

Different requests need different level of QoS, so the priority of factors depends on the client's requirements. To this end, each cloud defines its considered priority. A neural network model is used to discover the optimal weight of each of the factors. Combining different factors, an overall QoS will be generated which shows the best influence of the selection factors. To select the quorum site, the only selection factor is most distance site which is reachable for both the primary and backup sites.

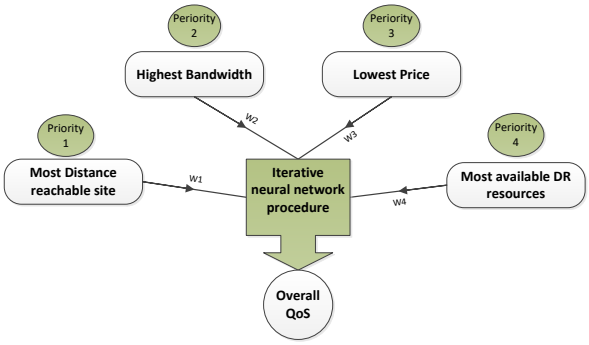

Figure 3. Using a neural network to select the best suitable sites for disaster recovery

4. MODELING AND ANALYSIS

In this section, we analyze the proposed disaster recovery model using Continuous-time Markov chain [32]. The state chart diagram is shown in Figure 4. As mentioned in section 3, we use the Eucalyptus platform in which every cloud can utilize maximum 8 clusters. However, to reduce the complexity of analysis, we consider two clusters in each cloud. One of the clusters is OC and the other cluster is DR.

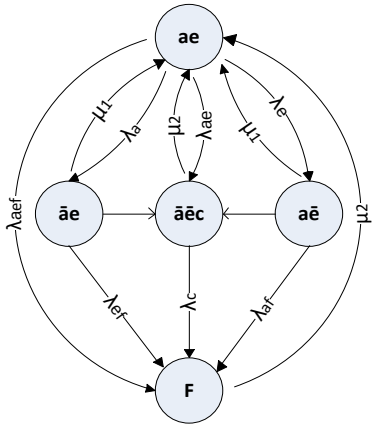

Figure 4. State Transition Model

ae : Both Clusters in healthy state
$a\bar{e}$: Cluster 2 in failed state
$\bar{a}e$: Cluster 1 in failed state
$\bar{a}\bar{e}c$: The primary cloud is crashed, backup cloud is activated
F : System is down
π: The probability of each state
λ: Failure rate
μ : Replacement / recovery rate

The transition model has five states. During the lifetime of the system, it goes through these states. In the initial state, both clusters of the primary site are healthy and safe for work. In case if cluster 1 fails with failure rate λ_a , then it reaches the state of āe. It means cluster 1 fails and its workload is redirected to another cluster. In this situation, cluster 1 can be repaired or replaced with rate μ_1 . In the same way, if cluster 2 fails with failure rate λ_e , it goes to aē and then can be repaired or replaced with a rateμ_1 . After a disaster happens, both primary and backup clusters may become inaccessible. So, the system reaches the state āēc with failure rate λ_{ae} . In another scenario, where states are aē and āe, if the second cluster fails, the

system reaches to āēc . Therefore, disaster recovery plan has to be handled to change the control to the backup site. To this end, failover procedure must be performed to activate the backup site. Then, the DR cluster in the backup site performs the redirected tasks. The primary site can be recovered with the recovery rate of μ_2 .

The state of F denotes that system is down and cannot deliver any service to their customers. In three situations, the system goes to Failure state (F):

1. The system is in the state of āēc , and the DR cluster fails (āēc ⟶ F).
2. Two clusters of the primary site suffer a failure. The data should be redirected to the backup site, however DR cluster cannot handle moved workload due to certain reasons (aē and āe ⟶ F).
3. The system is in the healthy state. After a disaster happens, the backup site should be activated, but it does not happen. (ae ⟶ F).

A steady state analysis of the diagram with the balance equation is performed as follows:

$$\pi_{ae} = \frac{\mu_1\ \pi_{\bar{a}e}+\mu_1\ \pi_{a\bar{e}}+\mu_2\ \pi_{\bar{a}\bar{e}c}+\mu_2\ \pi_F}{\lambda_a+\lambda_{ae}+\lambda_{ae}+\lambda_{aef}} \quad (1)$$

$$\pi_{\bar{a}e} = \frac{\lambda_a}{\mu_1\ +\lambda_e+\ \lambda_{ef}} \times \pi_{ae} \quad (2)$$

$$\pi_{a\bar{e}} = \frac{\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} \times \pi_{ae} \quad (3)$$

$$\pi_{\bar{a}\bar{e}c} = [[\frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + \quad \lambda_{ae}] \times \frac{1}{\mu_2\ +\lambda_c}] \times \pi_{ae} \quad (4)$$

$$\pi_F = [[\frac{\lambda_{af}\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \frac{\lambda_{ef}\lambda_a}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + \quad [[\ \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + \lambda_{ae}\ ] \times \quad \frac{\lambda_c}{\mu_2\ +\lambda_c}] + \lambda_{aef}] \times \frac{1}{\mu_2}\ ] \times \pi_{ae} \quad (5)$$

The conservation equation can be obtained as follows:

$$\pi_{ae} + \pi_{\bar{a}e} + \pi_{a\bar{e}} + \pi_{\bar{a}\bar{e}c} + \pi_F = 1 \quad (6)$$

Combining the above-mentioned balance equations with the conservation equations, we get:

$$\pi_{ae} = \left[\begin{array}{l} 1 + \frac{\lambda_a}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + \frac{\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \\ \left[\left[\ \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + \lambda_{ae}\right] \times \frac{1}{\mu_2\ +\lambda_c}\right] + \\ [[\frac{\lambda_{af}\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \frac{\lambda_{ef}\lambda_a}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + [[\ \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_a+\ \lambda_{af}} + \\ \frac{\lambda_a\lambda_e}{\mu_1\ +\lambda_e+\ \lambda_{ef}} + \lambda_{ae}\ ] \times \frac{\lambda_c}{\mu_2\ +\lambda_c}] + \lambda_{aef}] \times \frac{1}{\mu_2}\ ] \times \pi_{ae} \end{array}\right]^{-1} \quad (7)$$

We assume that Mean Time To Failure (MTTF) is 6 months and Mean Time To Disaster (MTTD) is 1 year for the system in the healthy state. According to operating parameters which is listed in Table 1, we can calculate the probability of each state as shown in Table 2.

Table1. Operating parameters for the analyzed model

Parameters	Values
λ_a , λ_e and $\lambda_c = \frac{1}{MTTF} = \frac{1}{6 \times 30 \times 24} = \frac{1}{4320}$ hours	2.31×10^{-4} hours
$\lambda_{ae} = \frac{1}{MTTD} = \frac{1}{12 \times 30 \times 24} = \frac{1}{8640}$ hours	4.62×10^{-4} hours
λ_{aef}	10.672×10^{-8} hours
λ_{af} and λ_{ef}	5.336×10^{-8} hours
$\mu_1 = \mu_2 = \frac{1}{3}$ hours	3.33×10^{-1} hours

Table 2. Probability value of each system state

System state	Probability value
π_{ae}	9.9714×10^{-1}
$\pi_{\bar{a}e}$, $\pi_{a\bar{e}}$	1.3824×10^{-3}
$\pi_{\bar{a}\bar{e}c}$	1.4775×10^{-3}
π_F	1.9×10^{-7}

The system does not provide service when it is in the failure state (F). So, the availability of the system is given by:

$$\text{Availability} = 1 - \pi_F \quad (8)$$

Downtime is another metric which denotes the time period in which the system stop providing the service. It is given by:

$$\text{Downtime} = \pi_F \times L \quad (9)$$

Where L is the time interval = 1 year

As per our calculation, the obtained values of availability and downtime are given in Table 3.

Table 3. Calculated performance metrics for the analyzed system

Metric	Value
Availability	9.9999×10^{-1}
Downtime	1.6416×10^{-3} hours

5. CONCLUSION

In this paper, we propose a disaster recovery solution for CaaS model in InterCloud systems. We model and analyze the proposed system using Continuous-time Markov chain. The result of steady state probability analysis shows that availability of the system is very high and the system is down only a few minutes in a year. The use of the proposed system is not limited to CaaS but also any kind of applications. In that case, DR clusters can be allocated to different clouds to guarantee the service continuity. As security issues are major challenges in InterCloud systems, we will focus on developing the security module in our future work. We will also investigate trust and privacy to enhance the secure deployment of InterCloud.

REFERENCES

[1] CaaS-Openstack. [Online]. Available: https://wiki.openstack.org/wiki/CaaS.
[2] S. Yokoyama and N. Yoshioka, "Cluster as a Service for self-deployable cloud applications," *12th IEEE/ACM International Symposium on Cloud and Grid Computing*, pp. 703-704, 2012.
[3] A. Chilwan, *et al.*, "Effects of Dynamic Cloud Cluster Load on Differentiated Service Availability," *21st International Conference on Computer Communications and Networks*, pp. 1-6, 2012.
[4] R. Buyya, *et al.*, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," *Algorithms and architectures for parallel processing*. vol. 6081, pp. 13-31, 2010.
[5] J. Li, *et al.*, "CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration," *11th ACIS International Conference on.Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing*, pp. 181-186, 2010.
[6] C. Liu, *et al.*, "Cloud resource orchestration: A data-centric approach," *Conference on Innovative Data Systems Research*, pp. 1-8, 2011.
[7] Y. Demchenko, *et al.*, "Intercloud Architecture for interoperability and integration," *IEEE 4th International Conference on Cloud Computing Technology and Science*, pp. 666-674, 2012.
[8] C. Liu, *et al.*, "Declarative automated cloud resource orchestration," *2nd ACM Symposium on Cloud Computing*, 2011.
[9] D. Bernstein, *et al.*, "Blueprint for the intercloud-protocols and formats for cloud computing interoperability," *4th International Conference on IEEE Internet and Web Applications and Services*, pp. 328-336, 2009.
[10] H. Shen and G. Liu, "An Efficient and Trustworthy Resource Sharing Platform for Collaborative Cloud Computing," *IEEE Transaction on Parallel and Distributed Systems*, Issue. 99, pp. 1-11, 2013.
[11] T. Wood, *et al.*, "Disaster recovery as a cloud service: Economic benefits & deployment challenges," *2nd USENIX Workshop on Hot Topics in Cloud Computing*, pp. 1-7, 2010.
[12] T. T. Lwin, and T. Thein, "High Availability Cluster System for Local Disaster Recovery with Markov Modeling Approach," *International Journal of Computer Science Issues, Vol. 6, No. 2*, 2009.
[13] B. Silva, *et al.*, "Dependability models for designing disaster tolerant cloud computing systems," *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 1-6, 2013.
[14] M. Pokharel, *et al.*, "Disaster Recovery for System Architecture using Cloud Computing," *10th IEEE/IPSJ International Symposium on Applications and the Internet*, pp. 304-307, 2010.
[15] S. Rajagopalan, *et al.*, "SecondSite: disaster tolerance as a service," *ACM SIGPLAN Notices*, Vol. 47. No. 7. ACM, pp. 97-108, 2012.
[16] H. Kashiwazaki, "Practical uses of cloud computing services in a Japanese university of the arts against aftermath of the 2011 Tohoku earthquake," *40th ACM SIGUCCS annual conference on Special interest group on university and college computing services*, pp. 49-52, 2012.
[17] T. Nayak, *et al.*, "End-to-end disaster recovery planning: From art to science," *Network Operations and Management Symposium*, pp. 357-364, 2010.
[18] "Virtualizing disaster recovery using cloud computing", *IBM White Paper*, January, 2012.
[19] O. H. Alhazmi and Y. K. Malaiya, "Assessing Disaster Recovery Alternatives: On-site, Colocation or Cloud," *IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, pp. 19-20, 2012.
[20] V. Salapura, "Cloud computing: Virtualization and resiliency for data center computing." , *IEEE 30th International Conference on Computer Design*, pp. 1-2, 2012.
[21] W. Chang, "A resource efficient scheme for network service recovery in a cluster," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 1087-1091, 2001.
[22] V. Javaraiah, "Backup for cloud and disaster recovery for consumers and SMBs," *IEEE 5th International Conference on Advanced Networks and Telecommunication Systems*, pp. 1-3, 2011.
[23] B. Cully, *et al.*, "Remus: High Availability via Asynchronous Virtual Machine Replication", *5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 161-174, 2008.
[24] M. C. Caraman, *et al.*, "Romulus, Disaster Tolerant System based on Kernel Virtual Machines", *20th International DAAAM Symposium :Intelligent Manufacturing & Automation: Theory, Practice & Education*, pp. 1671-78, 2009.
[25] M. C. Caraman, *et al.*, "Continuous Disaster Tolerance in the IaaS clouds," *IEEE 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pp. 1226-32, 2012.
[26] T. Wood, *et al.*, "PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery." *2nd ACM Symposium on Cloud Computing*, 2011.
[27] Z. Jian-hua and Z. Nan, "Cloud Computing-based Data Storage and Disaster Recovery", *IEEE International Conference on Future Computer Science and Education*, pp. 629-632, 2011.
[28] K. Grolinger, *et al.*, "Knowledge as a Service Framework for Disaster Data Management." *IEEE 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 313-318, 2013.
[29] M. Brock, and A. Goscinski, "Offering clusters from clouds using WSDL and stateful web services," *IEEE Asia-Pacific Services Computing Conference*, pp. 233-238, 2009.
[30] F. Doelitzscher, *et al.*, "ViteraaS: Virtual Cluster as a Service." *IEEE Third International Conference on Cloud Computing Technology and Science*, pp. 652-657, 2011.
[31] Y. Ueno, *et al.*, "Performance Evaluation of a Disaster Recovery System and Practical Network System Applications," *IEEE Fifth International Conference on Systems and Networks Communications*, pp. 195-200, 2010.

[32] S. Chuob, *et al.*, "Modeling and Analysis of Cloud Computing Availability based on Eucalyptus Platform for E-Government Data Center," *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 289-296, 2011.

BIOGRAPHY OF AUTHORS

Mohammad Ali Khoshkholghi is currently a Ph.D. candidate at the Faculty of Computer Science and Information Technology, University Putra Malaysia. He obtained his master's degree in computer science from Iran in 2010. His current research interests include distributed computing focusing on resource management, green computing , optimization , and security in cloud computing.

Azizol Abdullah obtained his Master of Science in Engineering (Telematics) from the University of Sheffield, UK in 1996 and his PhD in Distributed System from University Putra Malaysia, Malaysia in 2010. He is a Senior Lecturer at Department Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia. Currently, he is Head of Department Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia. He is also has been appointed as Fellow Researcher for ITU-UUM Asia Pacific Centre of Excellence For Rural ICT Development (ITU-UUM). His main research areas include cloud and grid computing, network security, wireless and mobile computing and computer networks.

Rohaya Latip is a Senior Lecturer of Technology Communication and Network Department, Faculty of Computer Science and Information Technology. She received her B.Sc. degree in Computer Science from University Technology Malaysia in 1999. Her M.S degree in Distributed System year 2001 and Ph.D in Distributed Database year 2009 from University Putra Malaysia. She is a member of IEEE Computer Society and also an associate researcher at the Laboratory of Computational Science and Informatics, Institute of Mathematical Science (INSPEM), University Putra Malaysia. Her main research interest includes Data Grid, Distributed Database, Grid Computing, and Network Management.

Shamala Subramaniam received a B.S. degree in Computer Science from University Putra Malaysia (UPM) in 1996, an M.S. (UPM) in 1999, and a Ph.D. (UPM) in 2002. She is a Associate Professor at Department Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia. Her research interests include Computer Networks, Simulation and Modeling, and Scheduling and Real Time Systems. She has published several journal papers.

Mohamed Othman is a Professor in Computer Science at the Dept of Communication Tech and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM) and prior to that he was a Deputy Director of Information Development and Communication Center (iDEC), who involved in monitoring the UMPNet network campus, uSport Wireless Communication project and UPM Server farm. His main research interests are in the fields of parallel and distributed algorithms, high-speed networking, network design and management (network security, wireless and traffic monitoring) and scientific computing. He is a member of IEEE Computer Society, International Association of Computer Science and Information Technology (IACSIT), Malaysian National Computer Confederation and Malaysian Mathematical Society. He already published more than 160 National and International journals and more than 250 proceeding papers. He is also an associate researcher and coordinator of High Speed Machine at the Laboratory of Computational Science and Mathematical Physics, Institute of Mathematical Science (INSPEM), UPM.