

An efficient algorithm for monitoring virtual machines in clouds

Sultan Alshamrani

Faculty of College of Computers and Information Technology, Taif University, Kingdom of Saudi Arabia

Article Info

Article history:

Received Dec 1, 2018

Revised Jan 19, 2019

Accepted Feb 16, 2019

Keywords:

Cloud computing

Cloud monitoring

Clustering

Hierarchical structure

Virtual machines

ABSTRACT

Cloud computing systems consist of a pool of Virtual Machines (VMs), which are installed physically on the provider's set up. The main aim of the VMs is to offer the service to the end users. With the current increasing demand for the cloud VMs, there is always a huge requirement to secure the cloud systems. To keep these cloud systems secured, they need a continuous and a proper monitoring. For the purpose of monitoring, several algorithms are available with FVMs. FVM is a forensic virtual machine which monitors the threats among the VMs. Our formulated algorithm runs on FVM. In this paper, we formulate the Random-Start-Round-Robin algorithm for monitoring inside FVM.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Sultan Alshamrani,
College of Computers and Information Technology,
Taif University. P.O. Box 888 Al-Hawiya, Taif,
Zip Code: 21974, Kingdom of Saudi Arabia.
Email: susamash@tu.edu.sa

1. INTRODUCTION

The concept of cloud computing can be understood as a combination of two related concepts. First, the applications which are transported as services over the Internet. Second, the hardware and the software components that are supporting these respective services in the data centers [1]. Cloud systems could be designed according to cloud providers and user needs [2]. However, other different reasons could control the design architecture of cloud computing. For example, security aspects should be considered in designing the cloud systems. [3, 4].

Some of the previous studies related to designing an architecture for the homogeneous computing systems include [5-10] used homogeneous structure. However, there are different architecture as in [11], where the authors used the hierarchical structure called as HASBE (hierarchical attribute-set-based encryption). The main reason for using this type of structure is to enhance the security of Cloud systems [11]. Furthermore, Ganglia [12] uses a hierarchical designed as a distributed monitoring system for cluster and other high-performance computing systems. This paper proposes an algorithm using the same technique.

The hierarchical structure for this paper includes three different levels of architecture. First, the virtual gateways that are defined as the software run by the cloud operator to make cloud resources available to the cloud customers [13]. It secures the user data and allows cloud providers to have better monitoring of their data efficiently [14]. Virtual gateway could have several virtual clusters. Second, the virtual clusters which are comprised of virtual machines (VMs) Those virtual machines are installed on distributed servers. The VMs in a cluster are interconnected by a virtual network. The target is to have a different number of clusters. FVMs and VMs will be distributed among clusters. A virtual cluster comprises several Virtual Machines (VM). Third, the VM that configured to be consistent in one cluster to share physical resources [15].

The evaluation of the problem in this paper for an efficient monitoring of cloud systems is to trace out the symptoms of malicious behavior in the VMs. For this purpose, the simplified Virtual Machines known as Forensic Virtual Machines (FVM), for performing cloud monitoring. The monitoring is done by visiting or inspecting VMs (regularly) to see whether they have the found the symptoms or not? The main contribution of this paper is to study the impact of the new proposed algorithm in the hierarchical structure of a cloud system that influences the efficiency of the monitoring of FVMs, and how to implement it in an efficient way. We start with a short review of previous work and related material, followed by a description of the new model including ways of discovering symptoms in a hierarchical structure. Finally, we present the results of simulations and provide conclusions to our findings.

2. PREVIOUS WORK

The main idea of cloud monitoring is to be able to frequently inspect VMs by several FVMs. FVMs are typically autonomous simplified VMs with limited computability and communication. This concept was more formally defined in the context of Cloud security in [16, 17], where a mobility algorithm was defined for the general purpose of searching for symptoms and discovering their dangerous configurations. The mobility algorithm has become an important tool for cloud monitoring providing light way security, though it is difficult to analyze due to its complexity including security aspects. The simplified version of it, called Reduce-Max algorithm, was formally specified and analyzed in more depth in [18]. The aim of the Reduce-Max algorithm was only to monitor the system by assuring frequent visits to VMs for identification of symptoms, without taking into consideration additional criteria imposed by security objectives. Next, we provide more detail explanation of Mobility and Reduce-Max algorithms.

2.1. Mobility algorithm

The authors in [17] proposed a formula (1) to assess the urgency of visiting a Virtual Machine v :

$$f(v) = \sum_{i=0}^k \frac{Disc(c_i, v)}{size(c_i)} val(c_i) + \lambda(v)T$$

where $Disc(c_i; v)$ is the number of symptoms inside one configuration c_i discovered at VM v , $size(c_i)$ is the number of potential symptoms in configuration c_i , $val(c_i)$ is an assigned value saying how dangerous the symptoms are for this configuration, (v) is a value associated with delay at VM v (also called a weight of v) and T is the current time minus the last time when VM v was visited by an FVM. In the proposed mobility algorithm, each FVM chooses randomly the next VM to visit from some range of VMs v with the highest value $f(v)$.

2.2. Reduce-max algorithm

The author in [18] introduced a simplified version of the Mobility algorithm, called Reduce-Max algorithm. In this algorithm, a decision about the next visited VM v is made by considering the (highest) value of a simplified function $f(v)=T(v)$. In [18], this algorithm was analyzed for five different weighting configurations of VMs: (uniformly) random, uniform, arithmetic, harmonic, and exponential. Some of these sets of weights will be considered in this paper.

2.3. Two ways of implementing reduce-max for many FVMs

In [19], there was a further development of Reduce-Max algorithm supported by a deeper analysis. The paper distinguished two ways of implementing the Reduce-Max algorithm in case of many FVMs: deterministic coordinated reduction and randomized local reduction. Deterministic coordinated reduction means that each i -th FVM inspects the exact i -th highest weighted VMs v (the value $f(v)$ of this VM will be reduced to zero after the visit). Randomized local reduction, on the other hand, allows each FVM to choose randomly VM(v) from a given range of highest values of $f(v)$, say, from the k FVMs v with the highest values $f(v)$, for some parameter. In [19], the tested ranges of highest VMs (chosen randomly by a FVM) were set up from 1 to 16. The same five configurations of weights as in [18] were tested for the Reduce-Max algorithm in both the considered versions (deterministic and randomized).

2.4. Hierarchical model

The simulations in this paper are performed in a hierarchical tree system, which is one of the real structures of the Cloud. This structure firstly used for the work of monitoring VMs in [20]. Two sets of weighting VMs are used here in this paper and were already studied in the context of a non-hierarchical homogeneous system in [19]: uniform and random. The reason of studying them here is that these sets of

weights have better results and the monitoring when using them was most efficient inhomogeneous systems, therefore it is natural to study them in the hierarchical system.

Poisson distribution has also been used and considered in [21] in the context of homogeneous systems, in addition to these two sets of weights. Poisson distribution will be used here too. The findings in [20] show how the performance of monitoring depends on these different configurations. The performance of monitoring measured as the maximum waiting time (or, alternatively, as the maximum accumulated weight since the last visit of an FVM), depends on the number of clusters.

3. THE CONTRIBUTION OF THIS WORK

After previous versions of the Reduce-Max algorithm, an efficient algorithm for the work of monitoring cloud systems is used in this paper. The algorithm is named a Random-Start-Round-Robin algorithm. For the Reduce-Max algorithm, the reduction was from the highest weighted VMs (even with its two versions, the reduction either of the exact highest weighted VMs or from a range of highest values).

However, with the Random-Start-Round-Robin algorithm, the way of reduction is different. One of VMs is selected randomly by one of FVMs. After that, this specific FVM inspects the selected VM after visiting it, wherever its position and not necessary it has the highest weight. then, this FVM inspects all VMs in round robin fashion. All other FVMs follow the same procedures.

4. MODEL DESCRIPTION

The new algorithm in this paper is considered in the cloud model of the tree structure. This structure includes three levels: gateway level clusters level and nodes level.

4.1. Methodology

In this work, Random-Start-Round-Robin algorithm is different from the previously described ones. Here, the results of each two versions of the Reduce-Max algorithm has been used to have a good comparison with the new algorithm results. What distinguishes this algorithm from the previous ones is that this time the reduction is as follows. FVMs choose one of VMs randomly in the first round then the visits to all VMs are in a round robin fashion.

The assumption is fixed from previous works. There is one physical host. The host has 1024 VMs in total, organized into the tree structure. The sets also are fixed, and they are Uniform, Random distribution for two different ranges (3 and 15) and Poisson distribution of weights for two different parameters (2 and 8). For each data set, the results will be for 1, 2, 4, 8 and 16 Clusters. More explanations about datasets are down in this paper.

For each set of parameters, each execution runs for 500000 rounds and compute the total max. Total-Max is the highest weighted latency $((v)T(v))$ between two visits to a VM that can be observed from the whole system. The problem is to schedule the visit of FVMs to VMs. The main functional objective is to minimize the latencies of visiting VMs. This means minimizing Total-Max. After that, for each total max, the execution is repeated 100 times and take the average; we do it to avoid random fluctuation of the results.

As explained above, five sets of data for weighting VMs are used, which have been used also in papers [19-21]. Total-Max in this paper is the highest weighted latency $\lambda(v)*T(v)$ between two visits to a VM that can be observed from the whole system. The problem is to schedule the visit of FVMs to VMs. The main functional objective is to minimize the latencies of visiting VMs. This means minimizing Total-Max.

5. RESULTS

5.1. Uniform weights

We start with the results of Uniform weights studied for the Random-Start-Round-Robin algorithm. Uniform inputs in this paper mean that all VMs are equal in regard to weights; without loss of generality, it could be assumed that all of the weights of VMs are ones [8]. The experiments were done using 16 FVMs and for 1, 2, 4, 8 and 16 Clusters as following.

In Figure 1, it can be seen clearly that the results of the total max of the Random-Start-Round-Robin algorithm are much better of the best result of the Randomized-Local version of the Reduce-Max algorithm, especially for smaller numbers of clusters. The result in Figure 1, for one cluster in the system, is around 0.23, which is better than the best result of the Randomized-Local version that is above 0.6. However, the Deterministic-Coordinated version of Reduce-Max with the same settings shows the result of 0.062 that is not only better than the Randomized-Local version but also better than Random-Start-Round-Robin algorithm's result.

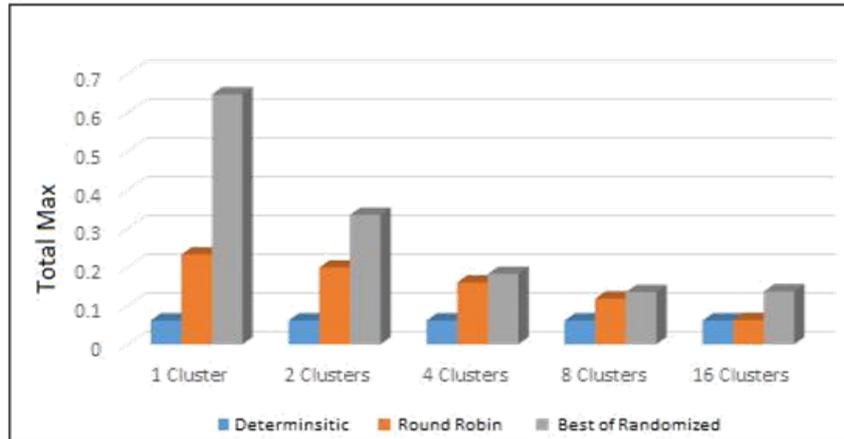


Figure 1. Total max of random-start-round-robin algorithm for uniform weights

Then, the results of a Random-Start-Round-Robin go down towards 0.063 when 16 clusters applied, which is really close to Deterministic-Coordinated results. Moreover, the best results of Randomized-Local version of Reduce-Max are de-creasing too with the increasing number of clusters. The best results of Randomized-Local reduction, when 16 clusters are in the system, is around 0.13.

5.2. Random [1,3] weights

Random inputs for weighting VMs in this work means that the values of weights are selected randomly and independently from integer values from a given range. We use two different ranges for Random weights. The first one is Random [1,3] weights. The range of it includes the values of 1, 2 and 3. The second one is Random [1, 15] weights, which includes the random selection of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15.

The reason of choosing these two different random ranges are first to have small values of random weights to compare the total-max of these weights with both total-max with bigger random range [1-15] and with Poisson (2). Secondly, we will also compare the range [1, 15] with Poisson (8). The results for Random [1, 3] weights, when the selection is from the range from 1 to 3, in Figure 2 are better than the best results of a Randomized-Local version of the Reduce-Max algorithm for most of the numbers of clusters and worse than Deterministic-Coordinated results in all cases.

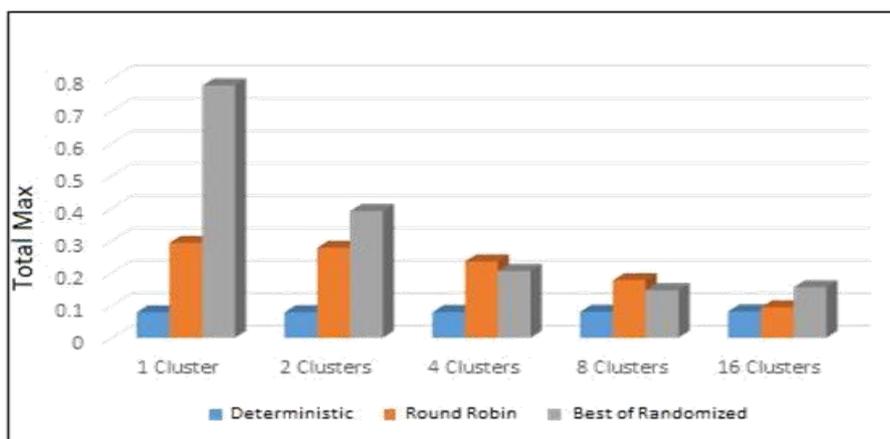


Figure 2. Total max of random-start-round-robin algorithm for random [1,3] weights

The results start at around 0.29 for one cluster and end with 0.094 for 16 clusters. The case with the best result of a Randomized-Local version of Reduce-Max is almost the same. It starts from about 0.78 for

one cluster and ends with around 0.16 for 16 clusters. However, the results of Deterministic-Coordinated version increase from 0.079 for one cluster to 0.081 for 16 clusters.

5.3. Random [1,15] weights

In this subsection, results are shown for a random set when the selection is from 1 to 15 for different numbers of clusters. Overall, in Figure 3 the results of a Random-Start-Round-Robin are similar to those for random [1,3] weights. They are better than the best results of Randomized-Local version of the Reduce-Max algorithm for most of the numbers of clusters and worse than Deterministic-Coordinated results in all cases. They start at about 0.43. This value is close to the corresponding one for Poisson (8) weights. This value becomes 0.12 when 16 FVMs are applied. The result of Deterministic-Coordinated version is about 0.074 for one cluster and goes up to 0.08 for 16 clusters. Moreover, the best results of Randomized-Local version decrease from 0.72 to 0.16.

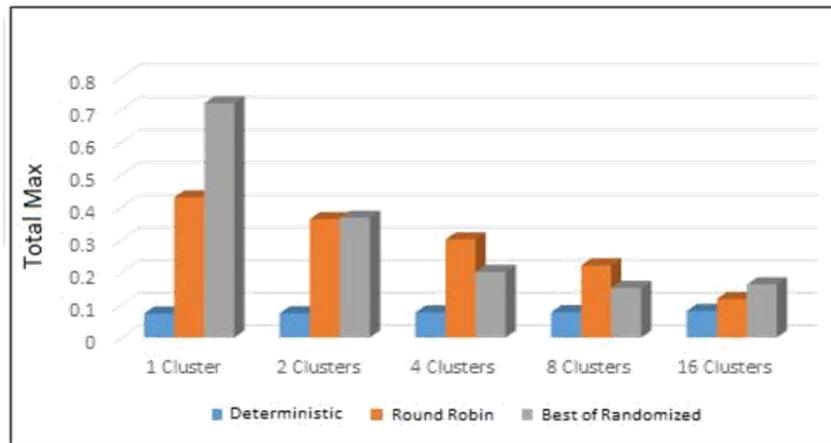


Figure 3. Total max of random-start-round-robin algorithm for random [1,15] weights

5.4. Poisson (2) weights

For Poisson weights with=2, in Figure 4, the result of a Random-Start-Round-Robin algorithm is showing the worst results compared to the results of both Deterministic and Randomized versions of Reduce-Max.

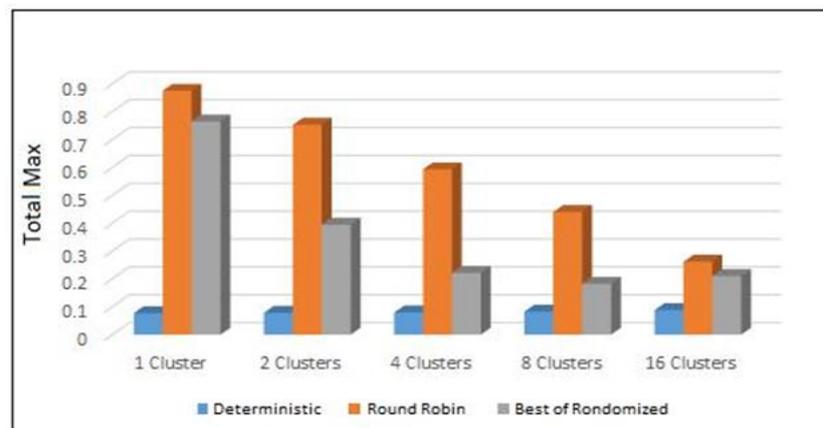


Figure 4. Total max of random-start-round-robin algorithm for poisson (2) weights

Random-Start-Round-Robin result for one cluster is approximately 0.875. This value of total max is decreasing with the rising number of clusters. It reaches around 0.26 when 16 clusters are in the system. The

same case happens with the best result of a Randomized-Local version of Reduce-Max. The total max starts from about 0.77 for one cluster and ends with around 0.21 for 16 clusters.

On the other hand, the results of a Deterministic-Coordinated version of Reduce-Max start from approximately 0.077. Then, an increase in the results happens until reaching 0.088 for 16 clusters. However, the results for Deterministic-Coordinated are indicating the best total max.

5.5. Poisson (8) weights

In regard to Poisson weights with=8, the results of the Random-Start-Round-Robin algorithm in Figure 5 are the worst results compared to the results of both Deterministic-Coordinated and Randomized-Local versions of Reduce-Max when 2, 4 and 8 clusters are applied. However, when one and 16 clusters are in the system, the results are better than the best results of a Randomized-Local version of the Reduce-Max algorithm and worse than Deterministic-Coordinated results.

Random-Start-Round-Robin result for one Cluster is 0.47. This value of total max is falling to the value of 0.15 when 16 clusters are in the system. The best result of the Randomized-Local version of Reduce-Max starts from about 0.72 for one cluster and ends with around 0.16 for 16 clusters. However, the results of the Deterministic-Coordinated version that keeps giving the best results start from around 0.074. The results rise until reaching 0.08 when 16 clusters are applied.

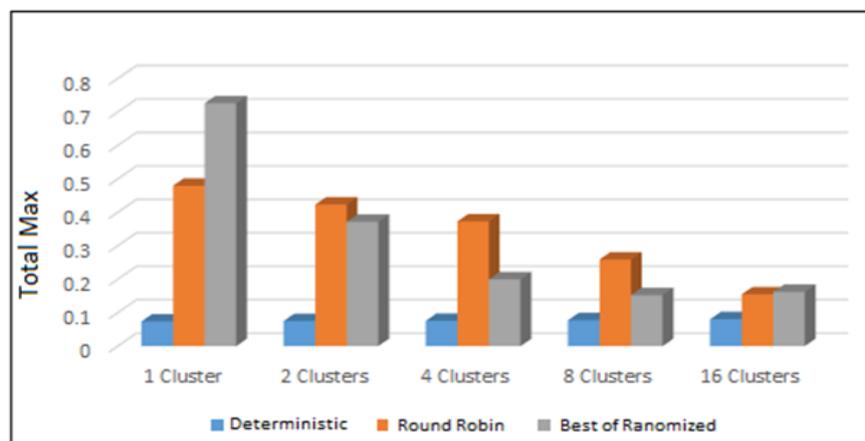


Figure 5. Total max of random-start-round-robin algorithm for poisson (8) weights

6. CONCLUSION AND OPEN PROBLEM

In this paper of Random-start-Round-Robin algorithm, we compare it with other related algorithms like Reduce-Max algorithm by in hierarchical systems. Overall, the results of Random-Start-Round-Robin algorithm shows better than the best results of Randomized-Local reduction technique of Reduce-Max algorithm. However, the Deterministic-Coordinated technique of Reduce-Max, as expected, gives better results but it needs more coordination time for processing. We can say Random-Start-Round-Robin algorithm is a better choice than Randomized-Local reduction and its results are not far from the Deterministic-Coordinated version of Reduce-Max. The future and interesting direction of this research are to test the efficiency of distributed monitoring in both hierarchical and homogeneous structures.

ACKNOWLEDGEMENTS

I would like to thank Taif University, Saudi Arabia for providing the financial support on this project.

REFERENCES

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [2] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities". In *High Performance Computing and Communications, 2008. HPC²08. 10th IEEE International Conference on*, 2008: pages 5–13.

- [3] QianWang, CongWang, Jin Li, Kui Ren, and Wenjing Lou. "Enabling public verifiability and data dynamics for storage security in cloud computing". In *European symposium on research in computer security*, Springer, 2009: pages 355–370.
- [4] Sara Qaisar and Kausar Fiaz Khawaja. "Cloud computing: network/security threats and countermeasures". *Interdisciplinary Journal of Contemporary Research in Business*, 2012; 3(9): 1323.
- [5] Mohand Mezmas, Nouredine Melab, Yacine Kessaci, Young Choon Lee, E-G Talbi, Albert Y Zomaya, and Daniel Tuytens. "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems". *Journal of Parallel and Distributed Computing*, 2011; 71(11): 1497–1508.
- [6] R. Ge, Xizhou Feng and K. W. Cameron, "Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters," *SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, Seattle, WA, USA, 2005, pp. 34-34.
- [7] D. Zhu, R. Melhem and B. R. Childers, "Scheduling with dynamic voltage/speed adjustment using slack reclamation in multiprocessor real-time systems," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 7, pp. 686-700, July 2003.
- [8] Kyong Hoon Kim, Rajkumar Buyya, and Jong Kim. "Power aware scheduling of bagof-tasks applications with deadline constraints on dvs-enabled clusters". In *CCGRID*, 2007, volume 7, pages 541–548.
- [9] Jian-Jia Chen and Tei-Wei Kuo, "Multiprocessor energy-efficient scheduling for real-time tasks with different power characteristics," *2005 International Conference on Parallel Processing (ICPP'05)*, Oslo, Norway, 2005, pp. 13-20.
- [10] B. Rountree, D. K. Lowenthal, S. Funk, V. W. Freeh, B. R. de Supinski and M. Schulz, "Bounding energy consumption in large-scale MPI programs," *SC '07: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, Reno, NV, USA, 2007, pp. 1-9.
- [11] Z. Wan, J. Liu and R. H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743-754, April 2012.
- [12] Matthew L Massie, Brent N Chun, and David E Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7):817–840, 2004.
- [13] Y. Royon, S. Frenot, and F. Le Mouel, "Virtualization of service gate-ways in multi-provider environments," in *Component-Based Software Engineering*, Springer, 2006, pp. 385–392.
- [14] H. Lohr, A.-R. Sadeghi, and M. Winandy, "Securing the e-health cloud," in *Proceedings of the 1st ACM International Health Informatics Symposium*, ACM, 2010: pp. 220–229.
- [15] X. Zhang, K. Keahey, I. Foster, and T. Freeman, "Virtual cluster workspaces for grid applications," SI: TR-ANL/MCS-P1246-0405, 2005.
- [16] K. Harrison, "Virtual machines," Sept. 30 2010. US Patent App. 13/822,239.
- [17] K. Harrison, B. Bordbar, S. T. T. Ali, C. I. Dalton and A. Norman, "A Framework for Detecting Malware in Cloud by Identifying Symptoms," *2012 IEEE 16th International Enterprise Distributed Object Computing Conference*, Beijing, 2012, pp. 164-172.
- [18] S. Alshamrani, "Discovering malicious behaviour symptoms in cloud systems," in *Proceedings of the Eighth Saudi Students Conference in the UK*, *World Scientific*, 2016, pp. 375–384.
- [19] S. S. Alshamrani, D. R. Kowalski and L. A. Gasieniec, "Efficient Discovery of Malicious Symptoms in Clouds via Monitoring Virtual Machines," *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, 2015, pp. 1703-1710.
- [20] S. Alshamrani, D. Kowalski and L. Gasieniec, "The impact of hierarchical structure on efficiency of Cloud monitoring," *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)*, Marrakech, 2016, pp. 40-46.
- [21] S. Alshamrani, D. Kowalski, L. Gasieniec, and M. Abdulazeez, "Balancing mobility algorithm for monitoring virtual machines in clouds," in *ECCWS2016-Proceedings fo the 15th European Conference on Cyber Warfare and Security*, 2016, p. 9.

BIOGRAPHY OF AUTHOR



DR. Sultan S Alshamrani is currently working as an assistant professor at Taif University in Saudi Arabia and he is the Chairman of the committee of faculty members affairs and the Chairman of the committee of Capstone Projects at the department of Information Technology. DR. Sultan got his PhD from the University of Liverpool in UK and a master's degree in Information Technology (Computer Networks) from the University of Sydney in Sydney, Australia. DR. Sultan finished his bachelor's degree in computer science from Taif University in 2007 with General Grade "Excellent" With first honor and an accumulative GPA of (4,85) out of (5,00) where considered the highest GPA in the collage