Pertanika J. Sci. & Technol. 26 (3): 1289 - 1306 (2018)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

# OPH-LB: Optimal Physical Host for Load Balancing in Cloud Environment

**Sakshi Chhabra* and Ashutosh Kumar Singh**

*Department of Computer Applications, National Institute of Technology, Kurukshetra, Haryana, India*

## ABSTRACT

Cloud computing has set a trend on a worldwide stage along with the rapid growth to enhance global technology standard and market scale in recent years. For the cloud users, load balancing in data center networks initiates the necessity of reducing the downtime for migrating overloaded virtual machines. To achieve better during-task deployment, optimal physical host must be selected efficiently. Nowadays, cloud customers are facing security risks in the context of load balancing of Virtual Machines (VM) which is infrequently addressed. This research addresses this pertinent issue and provides a different perspective of studying ways to develop VM deployment strategy by reducing the probability of VM co-tenancy with their targets. This will in turn make it difficult for attackers to evaluate the strategy. A model called Optimal Physical Host for Load Balancing (OPH-LB) is proposed to find the probability with probabilistic estimation in the form of its computing capability and performance in secure multi-tenant cloud. The proposed solution is evaluated via Cloudsim 3.0.3 and compared with two existing well-known algorithms. The reported results indicate that OPH-LB outperforms in improving the makespan, throughput, performance and reduces the failure number of task deployment. The results show that OPH-LB can effectively reduce the risks and security score and upgrades the utilisation of resources, with an improvement of 42.13% in all types of analyses for the experimental data.

*Keywords:* Cloud computing, deployment, optimality, performance, probability, security

## INTRODUCTION

The paradigm of cloud computing reshapes all prospective users and is emerging as the fastest technology worldwide. It is an eventual and promising way of managing and boosting the utilisation of resources and delivering various computing, IT services (Zissis & Lekkas, 2012). In recent years, the number of network users has been growing linearly, but traffic has been increasing exponentially at

ARTICLE INFO

*Article history:*
Received: 29 December 2017
Accepted: 30 March 2018

*E-mail addresses:*
sakshichhabra555@gmail.com (Sakshi Chhabra)
ashutosh@nitkkr.ac.in (Ashutosh Kumar Singh)
*Corresponding Author

ISSN: 0128-7680 

cloud data centers (Diaz, Martin, & Rubio, 2016). For this reason, load balancers are essential to balance the traffic and that is the reason why overloaded servers are able to process their backlog successfully. Virtual machine migrations and balancing the upcoming workload based on performance requirements are made possible by virtualisation. This is particularly useful when the workload is unpredictable or varies significantly. This XaaS (X-as-a service) model in cloud as infrastructure provides service which is parallel to hardware resources for its clients, such as Amazon EC2 and Amazon S3 (Ang, Por, & Liew, 2017). Because of the inadequate hardware possessions in a cloud system, it has become a big concern as to how to allocate the resources securely and choose the best machine for task deployment effectively. This will result in a strategy for load-balancing and obtaining reliable performance (Kavousi-Fard, Niknam, Taherpoor, & Abbasi, 2014). Many companies propose new techniques for fast and efficient application deliveries for the deployment of load, as KEMP technologies release vRealize Plugins for fast balancer deployments. VMs is also a commonly used resource in the cloud computing environment and acts like an isolated computing unit.

For cloud users, it facilitates the sharing of resources, on-demand resource scaling with high flexibility, controllability and predictable performance. However, apart from all these benefits, it also brings a new security threat as a number of concerns are emerging regarding the issue of multi-tenancy attacks where multiple tenants are residing on the same server. It is evident that if attackers and clients reside on the same server, there are more chances of co-resident attacks (Han, Chan, Alpcan, & Leckie, 2017). At the time of balancing the load, security issues are rarely addressed (Chhabra & Singh, 2018). So, a model for VM's security during load balancing and deployment of job requests is proposed. When virtual machines migrate, common tenants are concerned about VM placements from one host to strange hosts. Hence in this study, the researchers tried to analyse possible threats during load balancing. The general view of the cloud data centers is demonstrated in Figure 1. Here, cloud consumers send the workload requests for the deployment to cloud service providers (CSPs). The resource manager subsequently handles the heavy workload and monitors the whole scheduling of upcoming load in the cloud management portal. Then, CSPs divide the resources into fully managed resource pool chunks which deliver excellent flexibility and controls. To make changes in the resource pool itself is very simple and allocations can be scaled up or down as per requirement.

In facilitating effective usage of computing resources and reduced waiting time, two aspects, which are security and load balancing need to be considered (Lin, Chin, & Deng, 2014). In this paper, the safety of a VM allocation policy in its ability to defend against multi-tenant cloud attacks is measured. These metrics are modeled under basic resource provisioning VM allocation policy such as time-shared, and extensive experiments on the widely used simulation platform Cloudsim to validate the model. Basically, this new proposed secure load balancer policy OPH-LB significantly decreases the co-tenancy of virtual machines on the same physical host, but also satisfies the constraints in workload balance and security (Calheiros, Ranjan, Beloglazov, De Rose, & Buyya, 2011).

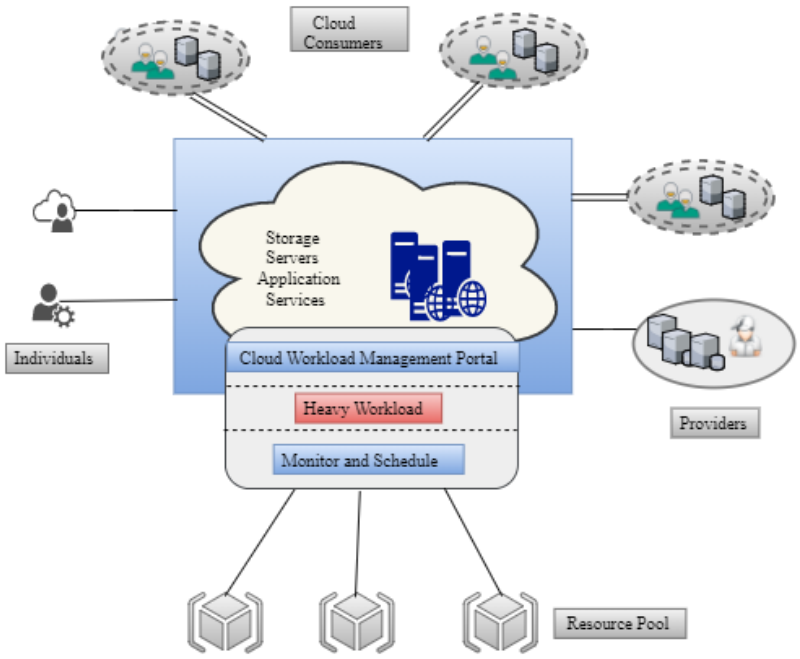


*Figure 1.* General view of cloud data center networks

## Significance of the Research

This idea provides a generic formulation which distributes the load effectively to the respective servers in cloud data center networks. This can only be achieved by considering the current status of requested tasks for the cloud data centers intelligently, which helps to improve the efficiency of computing the resources and managing the incoming requests among the physical machines securely. This secure load balancer not only significantly decreases the co-tenancy of virtual machines on the same physical host but also satisfies the workload balance.

The contributions are threefold:

- To consider the dynamic simulated scenario which helps structured and flexible evaluation of the proposed model during secure load balancing. A reliable VMs for task deployment is able to diminish the risks from malicious hypervisor and ensures the original performance of the load balance.
- To guarantee the scalability of algorithm, no modifications are required in the guest OS, hypervisors or hardware platforms. It basically modifies the structure of the model to a very less extent by converting a simple load balancer to secure load balancer.
- OPH-LB achieves relatively accurate estimation with less communication overheads of the upcoming load and selects optimal physical host for processing the tasks.

The subsequent sections of this paper is organised as follows: In section 2, a brief review of the related work which achieved secure load balancing in cloud datacenters is offered. Section 3 introduces the Optimal Physical Host for Load Balancing (OPH-LB). The performance

evaluation which was obtained from the practical implementations is discussed in section 4. Finally, section 5 concludes the paper and gives future direction of the work.

## RELATED WORK

Load balancing is the mechanism which decreases the possibility of VMs to be overloaded or underloaded. This leads to improvement in the concurrent user capacity and overall reliability of applications and also helps to achieve the best response time and good utilisation of resources. The methods of load balancing can be broadly classified into two categories: static and dynamic. In the static type of balancing, the prior knowledge of the system is needed and cannot change the requirement of resources at the run time because need is fixed before the programme execution. In the case of dynamic algorithm, it is based on the current usage and gives the facility of re-mapping according to their respective task requirement during run time. Dynamic algorithm deals with real-time network and finds the closest server in the whole system and prefers the deployment (Cho, Tsai, Tsai, & Yang, 2015). Here the present state of the system is used to make decisions to handle the load with the benefits of resource utilisation (Li, Qian, Lu, & Wu, 2013). With the aim of improving the performance, redistribution policy is applied in which they transfer the tasks from heavily loaded processors to lightly loaded processors. Literature search on previous work on optimal task deployment and its possible countermeasures has yielded various results. Zhao et al. (2016) proposed a model which focused on the selection problem to place the virtual machines optimally for deploying the requested tasks to achieve the immediate load balancing effect. The authors applied clustering approach with Bayes theorem to choose the optimal set of physical host. It was found that this approach improves the throughput, reduces the number of failures and optimises the load balancing effect. Domanal and Reddy (2014) implemented the method which ensures the usage of resources in an intelligent manner, so efficient utilisation of resources can be done. They also compared with previous Active-VM algorithms. Deng, Wu, Shen and He (2016) presented the model which combines both static and dynamic provisioning and makes the online power management system which helps to save power and reduce the operating cost of carbon emmissions of the data centers. Its algorithm is to adapt the usage of green data centers powered by renewable energy called EcoPower which performs better load balancing and eco-aware power management simultaneously (Deng, Lu, Lai, Luan, & Liang, 2016). Its main objective is to diminish the average time and cost without any compromise on quality. So, this system's performance proves that this model achieves a good balance between power savings, cost and quality. The cost can also be reduced by 20% and can use solar or wind complementary strategies for further purposes. An optimisation of virtual machine placement for energy efficiency algorithm was also proposed by Li et al. (2013) based on evolutionary population initialisation strategy which helps to minimise energy consumption and maximise load balance. This model is evaluated with many-objective optimisation problems (MaOPs) with energy consumption reduced to 23.01 kWh with a percentage of 0.00029 and minimises the 770 number of VM migrations.

The cross VM side channel attacks introduced by Martin (2010) proposed the fine-grained information extraction between VMs. They mapped the internal cloud infrastructure, where

a particular target VM and attacker VM was likely to reside in one physical machine. This approach is mainly focused on mitigating the side channel risks and employs the blinding techniques to minimise the information that can be leaked. An approach proposed by Zhang, Juels, Reiter and Ristenpart (2012) basically deals with extraction of fine-grained information from a victim VM running on the same server. These types of attacks happen on a symmetric micro processing virtualised system. Their algorithm is able to overcome a few problems: filter out numerous sources of noise, core migrations and extract the victim's key. There is a technique which helps to mitigate the arbitrary cloud side channel attacks which is recommended by Moon, Sekar and Reiter (2015). They presented "Nomad", a system that suggests vector-agnostic defense against known and future side-channels. It captures information leakage model by channels and required migration heuristics between VMs in shared cloud deployments. Sun, Shen, Li and Wu (2016) presented the model for analysis related to security threats and generated an information leakage model to secure the load balancing risks. This SeLance technique estimates and predicts the information leakage in the whole process during VM migrations and VM placement. Some researchers only focus on a particular threat suggested by using VM allocation policies to defend against co-resident attacks in cloud computing (Han et al., 2017). They basically follow three metrics: efficiency, coverage and VMmin. Their work satisfies these objectives such as security, workload balance and power consumption. Duan and Yang (2017) implemented the method which generates multi-tenancy oriented private clouds and allow multiple VMs to communicate with others under physical hosts. They have achieved global load balancing on the underlying physical networks. To attain efficient resource provisioning by optimal workload allocation is also studied in light of the max-min algorithm by Cao, Li and Stojmenovic (2014). Other solutions for secured load balancing with many techniques are by Papagianni et al. (2013), Ramezani, Lu, and Hussain (2014), and Zhao, Hu, Ding, Xu, and Hu (2014).

## OPH-LB

### Proposed Scenario and Assumptions

In IaaS cloud data centers to improve utilisation of resources, CSPs places the VMs which relates to different tenants in one physical host. The respective VMs share the host's resources and cloud service providers make sure the isolation between each VMs. In this multi-tenant cloud, there are chances of coinciding of VMs attacks. So, the objective of this study is to choose the optimal physical host for deploying the tasks in this secure cloud environment. When clients submit their job requests in resource pool of the cloud data centers, it generally chooses the physical host randomly for deploying the tasks as illustrated in Figure 2. However, it can become optimal if we can decide the most favorable machine for a particular task and will create a better load balancing effect. This idea will surely improve resource utilisation and provide high throughput. At the time of allocating tasks, firstly we need to check if the amount of resource requested by a task is greater than the available resources, then the physical host cannot deploy the task. When available memory is greater or close to the requested ones, then it can only deploy the tasks effectively. Whether it is possible for several physical machines to deploy the same task, we still need the physical host which is optimal among all

physical machines. As we know, parallel computing systems have many processors which run simultaneously and are scattered in multiple locations. For reducing the communication overheads between processors and to obtain the best scheduling performance, a hybrid system topology is pursued. At the time of finding a physical host which is the most favorable, we don't need to go and check for each and every processor because there is an index table named as DUIDX which preserves all information and estimations, and helps the load manager to maintain the records and reflect about any changes in the deployment time. When new jobs are coming in the system, then the load manager chooses the most optimal host for deployment so that communication overheads are reduced substantially.

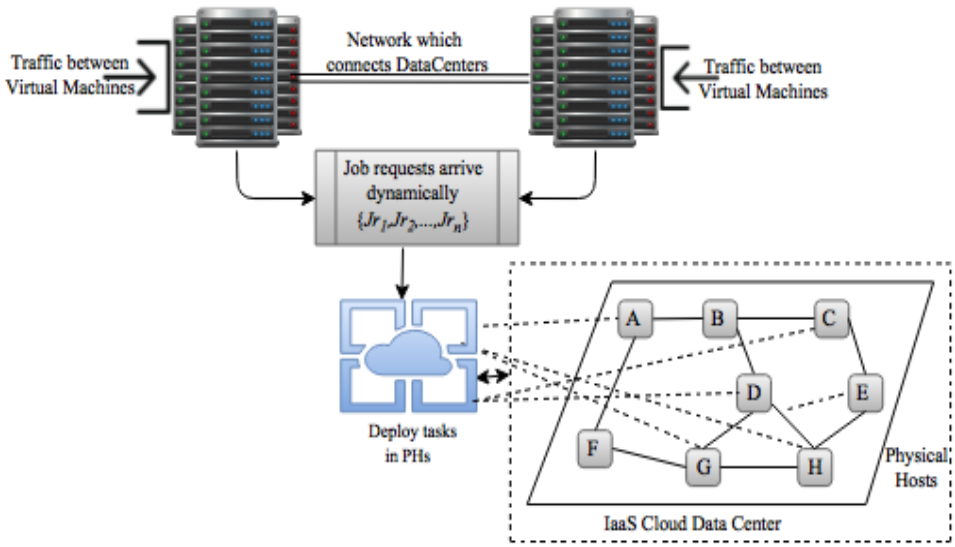


*Figure 2.* Requesting jobs into physical hosts for deployment in IaaS

This strategy is effective in high throughput with favourable scalability and less communication overheads. In multitenant cloud, attackers are always trying to achieve economic gain by utilising virtualisation and allow resource sharing. It occurs when two or more customers are using the same physical machine's services given by cloud service providers at one time. These risks come when both the attacker and clients are in the same cloud and are sharing the same server. In order for multi-tenancy, both virtualisation and resource sharing must be allowed by cloud service providers as shown in Eq. (1).

$$MultiTenancy = Virtualisation + Resource\ Sharing \quad (1)$$

Here, resources are being shared between attackers and customers. Although the difficulty of deploying co-tenant VMs is actually reduced by load balancing, multi-tenancy threat occurs when the attacker and the target victims are in the same cloud and are sharing the same physical host. Given this better and secure load balancing scenario, the new strategy should satisfy these three main objectives:

*Workload Balance* - The importance of balancing the workload is twofold: For cloud providers, distributing VMs over multiple processors have to improve the whole parallel performance in the cloud. In the research policy, all the requesting tasks are not allocated together on the same server at one time. These providers adjust the workload periodically and dynamically so that all running tasks of different locations will complete their execution on time, so that idle time is minimised and resource utilisation can be maximised.

$$WB : \max \frac{1}{N} \sum_{C_n \in £} [\,|\, \{s \mid s \in \mathrm{SR}, €_{\mathrm{C,Jr,PH}} = 1, \forall PH \in ℘\} \,|\,] \tag{2}$$

*Computing Efficiency* - How to effectively use the computing resources is a critical issue for cloud providers. There are many different techniques that have been widely discussed in various research papers (Zhang, Chong, Pezeshki, Moran, & Howard, 2017). However, there is yet to be a 100% foolproof solution. Therefore, here, the most straightforward approach is considered to pick out the most optimal hosts set and minimise the extra running servers.

$$CE : \min[\,|\, \{s \mid s \in \mathrm{SR}, \exists C_n \in £, Jr \in ¥, €_{\mathrm{C,Jr,PH}} = 1\} \,|\,] \tag{3}$$

*Secure Physical Host* - The ability to optimise and secure the cloud service performance is of great importance and selects the most optimal physical host to reduce unnecessary computing resources. Based on minimum resource utilisation and less communication overhead between hosts in secure environments, a way to dynamically produce the best load balancing with low computation complexity is allocated as legal and attackers can't deploy the tasks to the same physical host. Based on such idea, he average number of clients per physical host is minimised.

$$SPH : \min \frac{1}{℘} \sum_{s \in \mathrm{SR}} [\,|\, \{c \mid c \in £, €_{\mathrm{C,Jr,PH}} = 1, \forall PH \in ℘\} \,|\,] \tag{4}$$

These targets have determined and accomplished in the analysis and simulation output of this paper. In addition, the following assumptions are made:

- The available $CP_{avl}$, $MeM_{avl}$ and requested $CP_{rqs}$, $MeM_{rqs}$ capacity of the physical hosts should be known by the load balancer manager and stored in DUIDX index table. However, when a new request is being processed, only the favorable physical hosts with adequate resources left are considered. In other words, designing an algorithm to find the optimal host with less communication overhead is the focus of this study.

- This multi-modal optimisation considers two objectives: CPU utilisation and memory. It is done for every incoming task requests when it arrives in the cloud data center, so that only current system capacity state and task requests are taken into consideration.
- A dynamic environment for implementing this approach is considered in order to reduce unnecessary computation complexity and assure optimal balancing effect.
- Cloud Service Providers ($CSP_i$) don't have foregoing knowledge of the attacker's capability and requests are considered identically. VM live migration is taken into consideration because when attack launches, intrusion detection system notices automatically. It is known when the current working VM is in secure or insecure state.

## METHODS

### Architecture

The proposed problem of task deployment can be formulated with the following scenario: In a cloud paradigm of *N*clients £= $\{C_1, C_2, ..., C_n\}$, *R* job requests ¥ = $\{Jr_1, Jr_2, ..., Jr_r\}$ requesting to *p* physical hosts □ = $\{PH_1, PH_2, ..., PH_p\}$. A mapping $\epsilon$: £ ×¥⟶□ allocates each physical host from each user with specific job requests, $\epsilon_{£\times¥\times□} = \{\epsilon_{C,Jr,PH} | \epsilon_{C,Jr,PH} = 1$ if job request *Jr* of client *C* is allocated to optimal physical host$\}$. This obtained solution vector mapping $\epsilon$ is the deployment strategy and used to conclude which task will be deployed into which physical host ($PH_p$), and should be in secure state. The nomenclature of these notations is described in Table 1. These hosts are assumed to be heterogeneous and implemented in dynamic environment. When cloud data centers receive the request for deploying the tasks, then the OPH-LB problem is formulated in a stochastic framework to find out the final deployment strategy by utilising its algorithm mechanism. This problem for load balancing can be solved through optimising task deployment problem in every Δt time from a long perspective. The tuples are defined in $□ = \{Jr, PH, C_n, CP_{avl}, MeM_{avl}, CP_{rqs}, MeM_{rqs}, E\}$. These $Jr_r$ defines the job requests which comes for deploying their tasks from $\{i=1,2,...,r\}$. PH illustrates the set of available physical hosts $PH(p,tm) = \{PH_1, PH_2, ..., PH_p\}$, where *tm* represents the starting time for deploying the tasks. These $CP_{avl}$, $MeM_{avl}$ parameters discuss the current available CPU and memory resource amount of the p physical host in the set $CP^1_{avl}, CP^2_{avl}, \ldots, CP^n_{avl}$ and $MeM^1_{avl}, MeM^2_{avl}, \ldots, MeM^n_{avl}$. $CP_{req}$ and $MeM_{rqs}$ is the requested resource amount by clients for deploying the r tasks in *Jr*.

Table 1
*Description of notations to be used*

| Notation | Description |
|---|---|
| $C_n$ | Number of Clients |
| £ | Set Set of clients $\{1,2,...,n\}$ |
| ¥ | Set of Job Requests $\{1,2,...,r\}$ |
| € | Specifies mapping between hosts, job requests and clients |
| □ | Octatuple |
| $PH_p$ | Number of Physical hosts |
| $CP_{avl}$,$MeM_{avl}$ | Available resource amount of CPU and Memory |
| $CP_{rqs}$,$MeM_{rqs}$ | Requested resource amount of CPU and Memory |
| SR | Set of servers $\{1,2,...,s\}$ |
| $VM_{i(t)}$ | VM *i* which is located by tenant *t* |
| $VM_{i'(t')}$ | VM *i'* which is located by tenant *t'* malicious one |
| WB | Workload Balance |
| SPH | Secure Physical Host |
| CE | Computing Efficiency |
| RD | Random Deployment |

The goal is to protect the multi-tenant cloud in this way so that co-resident attacks never harm any physical host. A binary variable $\perp_p^i = 1$ indicates $VM_i$ will be placed on $ph_p$ and *0* otherwise. The load placement matrix is defined as *T*. Let $T_{i,j}$ be an element of matrix *T*, then $T_{i,j}$ defines the load placement between $VM_i$ and $VM_j$. It should be noted that a feasible VM placement decision should satisfy the following resource constraints:

$$\sum_{i=1}^{\wp} \perp_{\mathrm{p}}^{\mathrm{i}} CP_{rqq} \leq CP_{avl} i \in \{1,2,..,n\} \tag{5}$$

$$\sum_{i=1}^{\wp} \perp_{\mathrm{p}}^{\mathrm{i}} MeM_{rqq} \leq MeM_{avl} i \in \{1,2,..,n\} \tag{6}$$

Eq. (5) and (6) ensures that the total required consumption of processors and memory resource amount should not exceed its total capacity. For avoiding the overflow condition of the node servers, we have the following load constraint:

$$\frac{1}{2} \sum_{i=1}^{V} \sum_{p=1, p \neq i}^{\wp} T_{i,j} \perp_p^i \perp_j^p \leq B_s \forall s_t, s_q \in \delta, t \neq q \tag{7}$$

where $B_s$ denotes the bandwidth of servers st. The coefficient (1/2) defines the VM placement pair of respective servers, the load is calculated two times. It guarantees that each task is allocated in one VM as respective physical host with specific requests. This $VM_{i(t)}$ is defined

as VM *i* which is located by tenant *t* and $VM_{i'(t')}$ as $VM_{i'}$ located by tenant *t'* which is called malicious one. This $Co\text{-}Tenant_{t,i,t',i'}(\xi t)$ represents a Boolean value which defines whether any $VM_{i(t)}$ and $VM_{i'(t')}$ are Co-Tenant at time ξt. As the main goal is to reduce the leakage introduced by Chhabra and Singh (2016) or insecure states which occur during this proposed process, a secure load balancing policy is designed where resource manager allocates the tasks effectively so that optimal VM is selected and placed successfully. For each strategy, when the upcoming load comes for deployment, the secure or insecure states are calculated and predicted for every decision. When clients ($Cl_i$) send their workload job requests to the resource manager, OPH-LB model identifies the reliable or unreliable states. It maintains a collection of all possible VM($VM_i$) as in Eq. (8) to be selected and placement at first. Then, it calculates the secure or insecure states for all feasible migration paths. As the system has less insecure states, it should have more security advantages during the deployment of requested jobs. In case of safe states, the qualified hosts as required and available amount of resources are observed. The safe VMs are returned to the resource manager who will make the final decision, in appraisal of both security and load balancing, as illustrated in Figure 3.

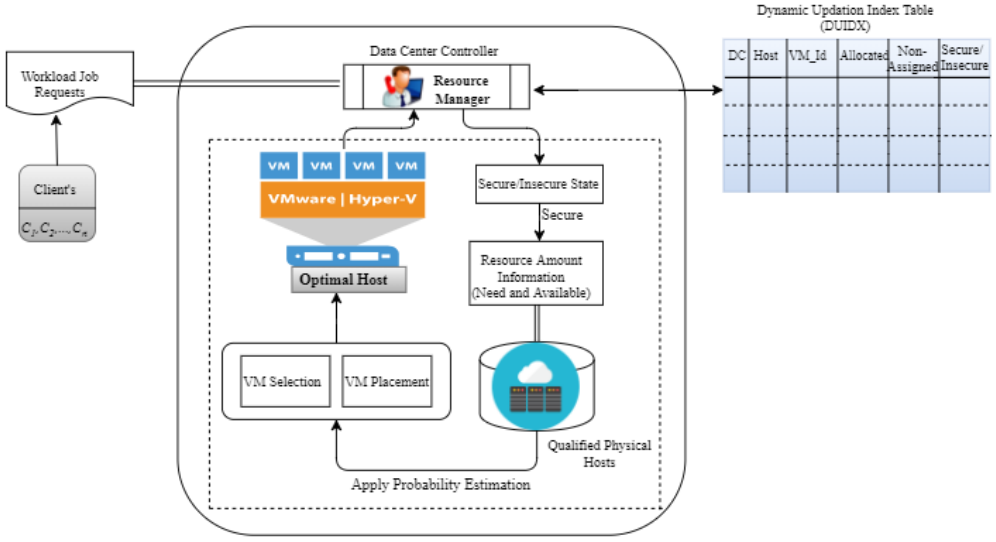


*Figure 3.* OPH-LB

Secure and insecure states were introduced, from which this model is able to show whether upcoming requests are deployed securely or not. For safe states, some conceptions have been initiated accordingly. Firstly, the reliability of particular tenants is estimated with the help of Co-Tenancy between those clients in historical time ξt. $THR_S$ time to find the malicious VMs from all the co-tenant VMs is considered when VM *P* and VM *Q* have been co-tenant in the past ξt times and resides parallel for more than its threshold time $THR_S$. When the intrusion detection system doesn't ring an alarm, then we can generally think that both VMs are favourable to each other and there is no maliciousness between them as shown in Eq. (8) (9); these machines are ready for the deployment of tasks. For the trusted hosts, intrusion detection system needs

$THR_{KH}$ time for finding the position of the hypervisor whether it is able to deploy the VM's tasks. If VM $P$ is migrated to host $R$, in the past $\xi t$ times and $P$ has been co-tenant in host $R$ for less than $THR_{KH}$ then $R$ is unknown host to $P$ and it can be the malicious one, otherwise it is called Known Host as in Eq. (10)

$$\sum_{i=1,t=1}^{n,m} VM_{i(t)} = (VM_{1(t1)}, VM_{2(t2)}, \ldots, VM_{i(ti)}, \ldots, VM_{n(tm)}) \tag{8}$$

$$[Co - Tenant_{i(t)} \times Favorable_{i(t)}](\xi t) > \prod THR_S \tag{9}$$

$$[Co - Tenant_{i(t)} \times Favorable_{i(t)} \times KnwHost_{i(t)}](\xi t) > \prod THR_{KH} \tag{10}$$

These equations help to observe the most favourable physical hosts which help to save the resources for the further deployments productively. By using probabilistic model (Chhabra & Singh, 2018), accurate estimations are calculated which depend on the particular set of data. The filtered ones are called favourable qualified sets (*FQPH*).

$$FQPH\{PH_1, PH_2, ..., PH_n \mid \Xi) = FQPH(PH_1 \mid \Xi) \times FQPH(PH_2 \mid \Xi) \times \ldots \times FQPH(PH_n \mid \Xi) \tag{11}$$

where $\Xi$ are the parameters, that is, CPU and Memory. For every host there should be fixed parameters of this function:

$$\aleph(\Xi; PH_1, PH_2, \ldots, PH_n) = FQPH(PH_1, PH_2, \ldots, PH_n \mid \Xi) = \prod_{i=1}^{n} FQPH(PH_i \mid \Xi) \tag{12}$$

Where ; denotes separation between two types of input. This equation demonstrates and helps to calculate the probability of each host for handling tasks. It assists in finding the qualified sets which meet the performance constraints. Tasks to particular hosts have been filtered out and assigned for the deployment which has maximum probabaility.

$$= \frac{\text{Requested}_i}{Capacity_i} = \frac{R_i^C}{C_i^C} \times \frac{R_i^M}{C_i^M}; \quad i \in \{1,2,\ldots, fq\} \tag{13}$$

where f$q$ is qualified set. $R_i$ and $C_i$ defines the requested and capacity computing power of CPU, Memory. Now, the maximum probability among all favourable qualified sets can be calculated.

$$\Phi_{OPT} = \{\max_{i=1}^{fq} \eta(\Xi; \mathrm{PH}_1, PH_2, \ldots, PH_{fq})\} \tag{14}$$

Finally, the tasks are placed on that host for deployment in cloud data centers for executing tasks. For convenience, the operational summary of the proposed model is mentioned in Algorithm 1.

Algorithm 1 *OPH-LB* ($Cl_n, V_v, PH_p, CP_{Avl}, MeM_{Avl}, CP_{req}, MeM_{req}, FQPH, E$)

Initiation: Requesting the jobs by clients for secure load balancing in Multi-tenant cloud.

Output: Searching for secure and optimal physical host.

PHList={}, QPHList={}, FQPH={}

For each client's $CL_n$ job request do

$T_{i,j} \perp_p^i \perp_j^p \leq B_s \ \forall s_t, \ s_q \in \delta, \ t \neq q$

end for

if $\perp_p^i CP_{req} \leq CP_{avl} \quad i \in \{1,2,...,n\}$ then

if $\perp_p^i MeM_{req} \leq MeM_{avl} \quad i \in \{1,2,...,n\}$ then

QPHList.add(*PHp*)

else

PHList.add(PHp)

end if

end if

if two VMs are co-tenant in $\xi t$ times then

$[Co-Tenant_{i(t)} \times Favorable_{i(t)}](\xi t) > \prod\prod THR_S$

FQPHList.add(PHp)

$\aleph(\Xi; PH_1, PH_2, .., PH_n) = FQPH(PH_1, PH_2, .., PH_n \mid \Xi) = \prod_{i=1}^{n} FQPH(PH_i \mid \Xi)$

end if

secure states found

for each physical host ϵ FQPHList do

the probability estimation applies only on favorable qualified physical hosts PH

$\varpi = \frac{\text{Requested}_i}{Capacity_i} = \frac{R_i^C}{C_i^C} \times \frac{R_i}{C_i}; \quad i \in \{1,2,.., fq\}$

end for

for each find the maximum oneamong all out of favorable qualified sets do

$\Phi_{OPT} = \{\max_{i=1}^{fq} \eta(\Xi; \text{PH}_1, PH_2, .., PH_{fq})\}$

end for

return secure optimal host

## RESULTS AND DISCUSSION

Experiments are conducted with performance and efficiency of the proposed solution and evaluated by considering dynamic creation via Cloudsim simulation environment. It is extremely difficult to examine these long-term experiments on real infrastructures and compared with the proposed technique. OPH-LB is compared with other two well-known methods applied in the literature against the corresponding performances: Random Deployment (RD) and Dynamic Load Balancing (DLB).

## Experimental Setup

The simulated cloud network is considered for achieving realistic results. The evaluation is conducted on 100 physical machines with different configurations which can efficiently fulfill the requirements of upcoming load simulation conditions. Every physical machine has its own *VMId* which increases/decreases according to this deployment of cloudlets in dynamic nature. In the space-shared, the machines are partitioned into a set of clusters and every cluster is allocated as a single job and shares the memory space. In time-shared, the computing power is divided by many users and each job runs for a quantum of time. The completion time is compared during analysis of the cloudlets in time-shared and space-shared allocation policies. These 100 physical machines have different available computing resource amount of CPU and Memory. There are 50 cloudlets running continuously on these physical machines. The information captured on these machines and tasks on the global blackboard of the OPH-LB is summarised in Table 2. Some parameters are defined in the range because of the different configurations used for every physical machine.

Table 2
*Parameters used in simulation*

| Parameters | Value |
|---|---|
| Host Memory | 204800 |
| Host Storage | 10000000 |
| Host Bandwidth | 100000 |
| System architecture | x86 |
| Operating system | Linux |
| MIPS | 250-350 |
| VM Image Size | 1000-5000 |
| VM Memory (RAM) | 2048 MB |
| VM Bandwidth | 1000-2000 |
| VMM Name | Xen |

## Makespan

Makespan is calculated by the difference between start time and fininsh time for processing the tasks during scheduling. When we assign the workload requests to physical hosts, it is mainly used during context switching. The processing time increases with increase in the number of requested tasks. As can be seen from Figure 4, this method is analysed and compared with the RD and DLB approaches. In Random Deployment (RD) scheme, it processes the tasks randomly in the cloud data centers. If the number of requested tasks increases and is randomly selected, the capacity of handling that volume of tasks will also weaken. This DLB approach is basically based on the prediction model and is built on some repository knowledge or historical experience. The total processing time of these strategies is always greater than OPH-LB method. It is because OPH-LB method can quickly find the optimal host based on the required resource, so that it has a smaller makespan among all under the same conditions. It also saves the cost of utilisation of resources and indirectly saves power consumption for cloud data centers.

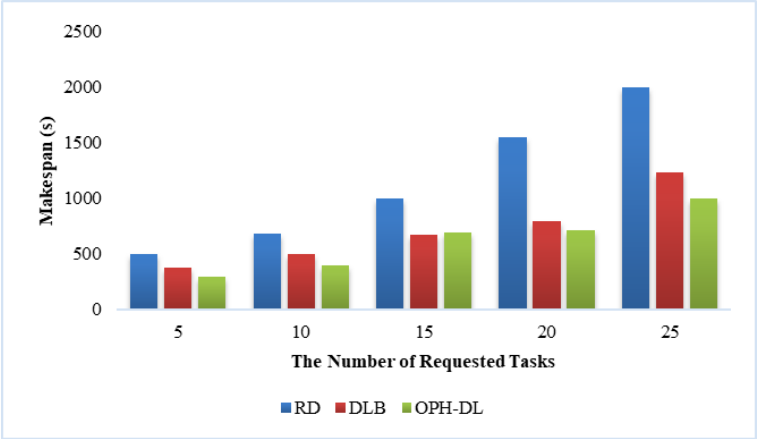


*Figure 4.* Comparison of makespan between random-migration, DLB and OPH-LB

## Failure of Number of Processing Events

In this evaluation, the number of failure tasks is found by the cloudsim simulator during scheduling and deploying the tasks. The finding of these failure nodes in the dynamic environment can only be possible if the chosen physical machine is unable to fulfill some of the demands of requested tasks. When the number of requested tasks increases, chances of the number of failures increases gradually because the ability of handling tasks will weaken slowly. As shown in Figure 5, the framework approach is compared with other existing techniques based on the failed number of tasks during deployment in the simulated cloud network data center. In this figure, RD approach has so many failure sets of tasks because it deploys randomly and in DLB there is less failure than RD because of its knowledge repository experimental values. But these sets of failure cannot manage in the real time. In the case of OPH-LB the analysis shows that the quantity of failure is less and in the plotted experiment, up to 43 requested tasks have no failure node (when tasks = 100, just <15 or few failed tasks). To sum up, the OPH-LB has better solidity and effectiveness for large-scale cloud data centers.

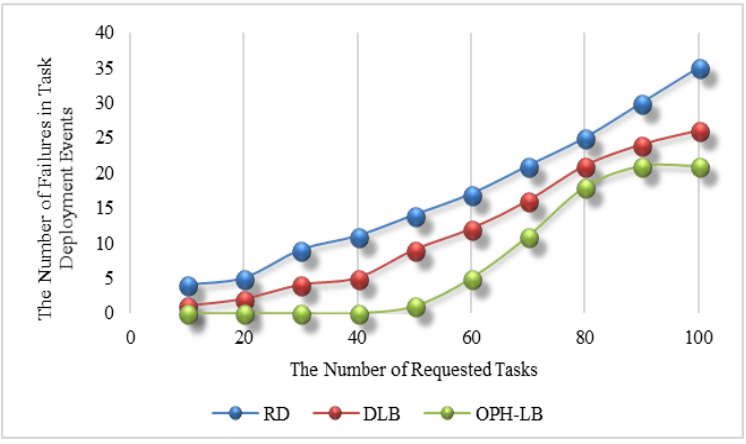


*Figure 5.* Comparison in failure between random-migration, DLB and OPH-LB

## Throughput Performance

This evaluation measure mainly uses an effective measure of load balancing to analyse and evaluate whether it is good or bad in terms of its service performance. These effective measures mainly include the ability of dealing with tasks, the response time to calculate a task request and the number of completed services per unit time. Based on these parameters in cloud system, we can calculate the throughput rate and evaluate the external service performance with respect to increasing time. In this study, the throughput among various numbers of requested tasks was calculated by taking cloudlets as 1000, 2000, 3000 and 4000 as shown in Figure 6.

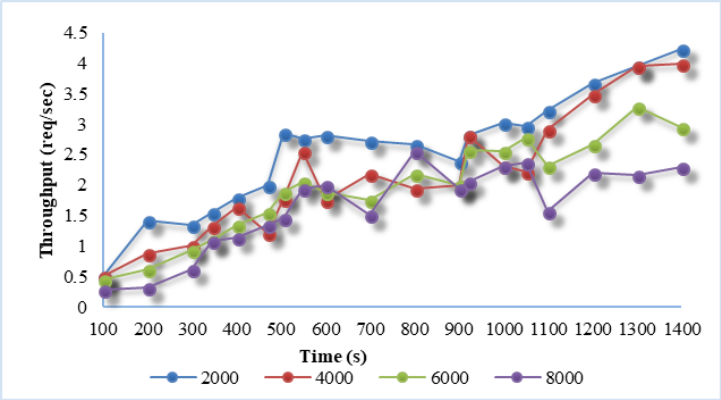


*Figure 6.* Throughput in different number of requested tasks

## Throughput for Succeeded Cloudlets

In this experiment, the succeeded throughput for measuring the performance of requested tasks has been illustrated, except the failure number of nodes. The result in Figure 7 shows that throughput based on succeeded cloudlets gives much better execution than the above result. It also helps to improve the resource utilisation of the cloud data center effectively.

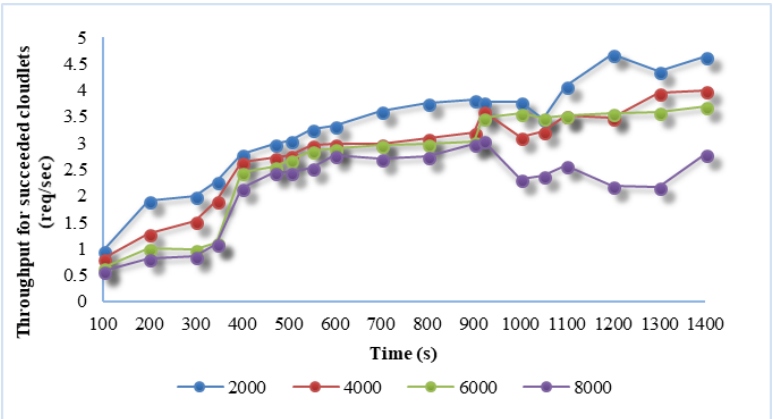


*Figure 7.* Throughput in different number of succeeded requested tasks

## Number of Requested Tasks under Normal State and Secure State

Figure 8 illustrates cloudlet completion time in time-shared scheduler policy for their respective tasks. When tasks increase from 100 to 1000, the time taken by the cloudlets for task completion also increases. At n = 100, 75% of cloudlets complete their processing in time = 160.2 s while at n = 800, only 6% of cloudlets complete their processing in t = 160 s and remaining 94% cloudlets complete their processing in 320.6 or 480.8 s respectively in normal state. Figure 9 shows that some of the cloudlets complete their execution more than 640.8 s but it is quite secure than the previous graph. It shows that even thought it takes more time, it ensures the clients share the resources without any insecurity.

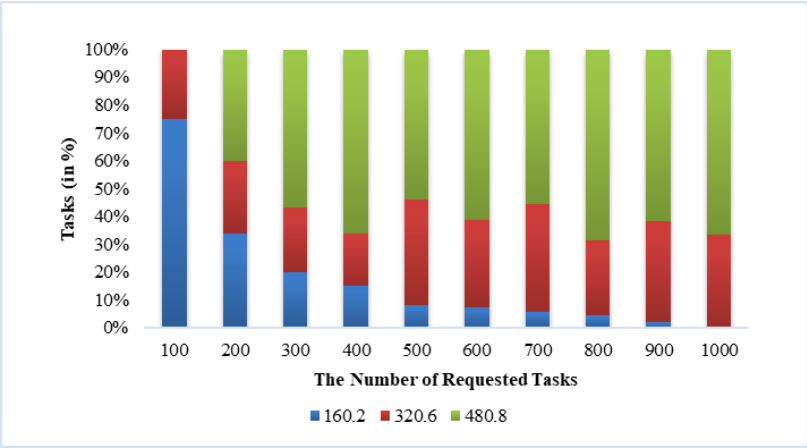


*Figure 8.* Number of requested tasks under normal state

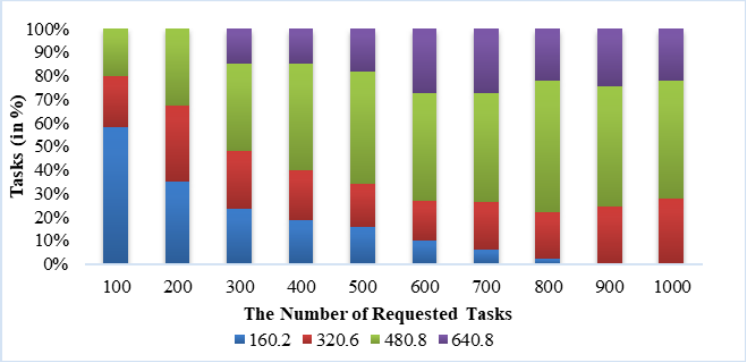


*Figure 9.* Number of requested tasks under secure state

## CONCLUSION

This paper provides a new perspective of task deployment to see the positive effect for secure load balancing. This paper propose an improved solution by using probabilistic model for effective utilisation of resources to provide the service of clients in cloud data centers. Its main purpose is to research and design a novel secure LB policy OPH-LB, to calculate and predict the secure or insecure states for every possible decision of VM selection and VM placement. It certainly reduces information leakage during load balancing and improves security benefit in the cloud. The simulation results show that OPH-LB has many benefits such as it reduces failure rate, shows improvement in throughput, decreases the makespan time, upgrades the utilisation of computing resources and boosts the external service performance too.

## REFERENCES

Ang, T. F., Por, L. Y., & Liew, C. S. (2017). Dynamic pricing scheme for resource allocation in multi-cloud environment. *Malaysian Journal of Computer Science, 30*(1), 1-17.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience, 41*(1), 23-50.

Cao, J., Li, K., & Stojmenovic, I. (2014). Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. *IEEE Transactions on Computers, 63*(1), 45-58.

Chhabra, S., & Singh, A. K. (2016). Dynamic data leakage detection model based approach for MapReduce computational security in cloud. In *2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS)* (pp. 13-19). Bhopal, India: IEEE.

Chhabra, S., & Singh, A. K. (2018). A probabilistic model for finding an optimal host framework and load distribution in cloud environment. *Procedia Computer Science, 125*, 683-690.

Chhabra, S., & Singh, A. K. (2018). Beyond lightning: A systematic review of information security in the age of cloud computing using key management. *International Journal of Computer Engineering and Applications, 11*(12), 299-315. Retrieved from www.ijcea.com ISSN 2321-3469.

Cho, K. M., Tsai, P. W., Tsai, C. W., & Yang, C. S. (2015). A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. *Neural Computing and Applications, 26*(6), 1297-1309.

Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal, 3*(6), 1171-1181.

Deng, X., Wu, D., Shen, J., & He, J. (2016). Eco-aware online power management and load scheduling for green cloud datacenters. *IEEE Systems Journal, 10*(1), 78-87.

Diaz, M., Martín, C., & Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications, 67*, 99-117.

Domanal, S. G., & Reddy, G. R. M. (2014). Optimal load balancing in cloud computing by efficient utilization of virtual machines. In *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)* (pp. 1-4). Bangalore, India: IEEE.

Duan, J., & Yang, Y. (2017). A load balancing and multi-tenancy oriented data center virtualization framework. *IEEE Transactions on Parallel and Distributed Systems, 28*(8), 2131-2144.

Han, Y., Chan, J., Alpcan, T., & Leckie, C. (2017). Using virtual machine allocation policies to defend against co-resident attacks in cloud computing. *IEEE Transactions on Dependable and Secure Computing, 14*(1), 95-108.

Kavousi-Fard, A., Niknam, T., Taherpoor, H., & Abbasi, A. (2014). Multi-objective probabilistic reconfiguration considering uncertainty and multi-level load model. *IET Science, Measurement and Technology, 9*(1), 44-55.

Li, X., Qian, Z., Lu, S., & Wu, J. (2013). Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Mathematical and Computer Modelling, 58*(5-6), 1222-1235.

Lin, C. C., Chin, H. H., & Deng, D. J. (2014). Dynamic multiservice load balancing in cloud-based multimedia system. *IEEE Systems Journal, 8*(1), 225-234.

Martin, T. D. (2010). Hey you - Get off of my cloud: Defining and protecting the metes and bounds of privacy, security, and property in cloud computing. *Journal of Patent and Trademark Office Society, 92*, 283-314.

Moon, S. J., Sekar, V., & Reiter, M. K. (2015). Nomad: Mitigating arbitrary cloud side channels via provider-assisted migration. In *Proceedings of the 22nd Acmsigsac Conference on Computer and Communications Security* (pp. 1595-1606). Denver, USA: ACM.

Papagianni, C., Leivadeas, A., Papavassiliou, S., Maglaris, V., Cervello-Pastor, C., & Monje, A. (2013). On the optimal allocation of virtual resources in cloud computing networks. *IEEE Transactions on Computers, 62*(6), 1060-1071.

Ramezani, F., Lu, J., & Hussain, F. K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International Journal of Parallel Programming, 42*(5), 739-754.

Sun, Q., Shen, Q., Li, C., & Wu, Z. (2016). Selance: Secure load balancing of virtual machines in cloud. *Trustcom/BigDataSE/I SPA, 2016 IEEE* (pp. 662-669). Tianjin, China: IEEE.

Zhang, Y., Juels, A., Reiter, M. K., & Ristenpart, T. (2012). Cross-VM side channels and their use to extract private keys. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (pp. 305-316). Raleigh, USA: ACM.

Zhang, Z., Chong, E. K., Pezeshki, A., Moran, B., & Howard, S. D. (2017). Near-optimal distributed detection in balanced binary relay trees. *IEEE Transactions on Control of Network Systems, 4*(4), 826-837.

Zhao, J., Hu, L., Ding, Y., Xu, G., & Hu, M. (2014). A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment. *PloS One, 9*(9), e108275.

Zhao, J., Yang, K., Wei, X., Ding, Y., Hu, L., & Xu, G. (2016). A heuristic clustering-based task deployment approach for load balancing using bayes theorem in cloud environment. *IEEE Transactions on Parallel and Distributed Systems, 27*(2), 305-316.

Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation Computer Systems, 28*(3), 583-592.