Egyptian Informatics Journal 22 (2021) 277–284

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Security-aware dynamic VM consolidation

Mohamed A. Elshabka [a,*], Hanan A. Hassan [a], Walaa M. Sheta [a], Hany M. Harb [b,c]

[a] Informatics Research Institute, City of Scientific Research and Technological Applications (SRTA-CITY), Alexandria, Egypt
[b] Information Technology College, Misr University for Science and Technology (MUST), 6th of October, Giza, Egypt
[c] Computers and System Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

ARTICLE INFO

Article history:
Received 21 June 2020
Revised 5 October 2020
Accepted 7 October 2020
Available online 1 November 2020

Keywords:
VM allocation
Dynamic VM consolidation
Security aware VM consolidation
Security aware VM placement
Cloud security

ABSTRACT

The explosive growth of cloud usage encourages several challenges, especially high energy consumption of Cloud Data Centers (CDCs), new security risks to Virtual Machines (VMs) resulting from co-residency with other risky VMs on the same Physical Machine (PM), and the Quality of Service (QoS) degradation due to sharing resources. Many recent studies have proposed Dynamic VM Consolidation (DVMC) to save energy with minimum degradation of the QoS. However, due to the lack of reliable security measurements and consolidating VMs without any awareness of their security risk degrees, the overall security risk of the CDC may be increased. To tackle these challenges, this study presents a Security-aware DVMC (SDVMC) that consists of a Security Monitoring Module (SMM) and a SDVMC module. The SMM utilizes a three-dimensional security assessment model, while in the SDVMC module we propose a novel VM placement algorithm called Minimum Risk Increase (MRI) with Risk Increase Threshold (RITH). The proposed MRI with RITH VM placement algorithm selects the host that leads to minimum risk increase to the overall security risk while maintaining the risk increase for each VM does not exceed the value of the proposed RITH constraint; which is set according to the aims of the cloud provider. Simulation results show that using our approach with RITH 0.8 results in security improvement, overall risk was decreased by 2% to 5%, without negative impact on energy consumption or QoS. Moreover, using our approach with RITH less than 0.8 enables the tradeoff between energy consumption and the overall security risk. The maximum overall risk decrease ranged from 10% up to 40%, according to the intensive of the communication overhead between the VMs, while the used energy in its maximum was less than half if we used a non-power-ware VM allocation policy.

1. Introduction

Cloud computing solutions have received much attention in recent years [1] as it offers cost-effective on-demand computing resources based on the pay-as-you-go model. The explosive growth of cloud usage results in high energy usage by the CDCs, it is estimated that data centers use around 1% of worldwide electricity [2]. This high energy consumption creats serious challenges for energy management in CDCs.

Many studies in recent years have focused on DVMC as a good way to overcome energy consumption challenges in CDCs without violating service level agreement [3], whereas, little attention has been given to consider the negative impact of DVMC on security. As demonstrated in [4] DVMC saves energy through migrating running VMs from underloaded hosts to other active hosts without overloading them and switching the underloaded hosts to low power state mode. Hosts in low power state mode consume a negligible amount of power while it can be activated in negligible time, the transition delay is 300 ms. While, DVMC saves QoS through keeping running hosts not overloaded by monitoring resource utilization migrating out some VMs from overloaded hosts to other hosts. However, consolidating multiple VMs belonging to different users on the same PM sharing its resources leads to new kinds of security risks. Luigi et al. [5] have shown that risky VMs can be stepping stones for attacking the host hypervisor and the other co-resident VMs. So, how the running VMs are distributed among PMs in the CDC has a significant impact on cloud

* Corresponding author at: Informatics Research Institute, City of Scientific Research and Technological Applications (SRTA-CITY), Alexandria, Egypt.
E-mail addresses: melshabka@srtacity.sci.eg (M.A. Elshabka), hali@srtacity.sci.eg (H.A. Hassan), wsheta@srtacity.sci.eg (W.M. Sheta), Hany.harb@must.edu.eg (H.M. Harb).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

https://doi.org/10.1016/j.eij.2020.10.002
1110-8665/

security as well as the energy consumption and the QoS produced to cloud consumers. Hence, there is a need for more research that considers security as well as energy and QoS. Furthermore, security concern demands a security assessment model to distinguish between VMs with different risk degrees.

To this end, this paper uses a three-dimensional security assessment model and proposes the SDVMC approach. For any VM the security assessment model considers the security risks experienced by the VM itself, risks experienced by its direct communicated VMs, and risks experienced by its collocated VMs to provide an overall metric that represents the overall security score of that VM. While SDVMC consists of two main modules SMM and SDVMC module. The SMM utilizes the three-dimensional security assessment model to continuously monitor and update the security scores for all VMs in the CDC. While the SDVMC module divides the problem of DVMC into four subproblems host underload detection, host overload detection, VM selection, and VM placement. For VM placement, based on the VMs security scores and according to the security assessment model, we proposed a novel MRI with a RITH VM placement algorithm that aims to select the host that leads to the minimum risk increase and avoid breaking the RITH constraint for any of its VMs.

We used CloudSim [6] to evaluate the proposed approach. Although CloudSim is a well-known and dependable cloud simulator, we have found a need to make little effective modifications to its default Power-Aware Best Fit Decreasing order (PABFD) VM placement and the power model for the used PMs. These modifications have enabled us to get more accurate insights when comparing our approach with other approaches that utilize the PABFD VM placement. Simulation results showed that the proposed approach gives high-level security improvements through trading off between energy consumption and the overall security. Moreover, we can get low-level security improvement without any negative impact on energy consumption and QoS.

The rest of the paper is organized as follows. Section 2, presents the related works. Next, in Section 3, our methodology is produced. Consequently, in Section 4, the experimental design for the simulation is presented in detail. In Section 5, the simulation results are analyzed and discussed, and in section 6 the concluding remarks and future work directions are presented.

2. Related work

Despite the massive research on dynamic VM consolidation, VM placement, and VM allocation techniques, also the massive research related to cloud security, few of the public research considered mitigating cloud security risks when applying DVMC.

VM consolidation algorithms are designed to reduce the power consumption like in [4,7,8]. A. Beloglazov et al. [4] split the problem of VM consolidation into four parts. Host Underload Detection algorithm, Host Overload Detection algorithm, VM Selection algorithm, and VM Placement algorithm. A. Abdelsamea et al. [7] proposed a multiple regression host overload algorithm that uses hybrid resources to enhance VM consolidation. Aya I. Maiyza et al. [8] aimed to balance the overall number of live migrations between VMs. However, all of [4,7,8] did not consider security risks.

Authors in [9–13] aimed to mitigate the risks that come from the multi-user environment based on VM placement. Zaina et. al. [9] proposed to construct a list with incompatible users based on users' adversaries that should not be allocated together on the same host. Their strategy requires knowing users and their adversaries which is difficult in public clouds. Min Li et al. [10] developed a VM migration technique based on Discrete-Time Markov Chain analysis, which aims to improve the survivability of the whole cloud and minimize the security risks that come from mounting side-channel attacks by considering the VMs vulnerabilities and the connections among virtual machines, However, the problem of initial VM placement is not considered in their effort. Yuchi et al. [11] extended the work of Min Li et al. [10] to include the initialization problem of VM placement. Both [10,11] did not reflect the co-resident risk caused by co-resident VMs. Miao et. al. [12] proposed a placement approach that mitigates Prime + Probe, Flush + Reload, and Memory-Bus-Lock attacks based on the characteristics of these attacks while allocating and migrating VMs, whereas, other co-resistant attacks were not considered. T. Kong et al. [13] proposed to migrate containers belonging to users with co-resident problems. However, their solution is for CDC that uses containers technology which is less isolated than VMs, also users without co-resident problems may have different risk degrees. Moreover, none of [9–13] presented a security assessment model that uses an overall metric to represent the risk of a candidate VM, the impact of the VMs communicates with it, and the impact of its co-located VMs.

Jin Han et. al. [14] extended previous work [10] and proposed a security assessment model that considers the vulnerabilities of the hypervisor, network connections between VMs, and the effect of co-resident VMs. And proposed a multi-objective VM Placement approach using Genetic Algorithms (GA) to provide better intrusion resilience, resource utilization, and network performance. However, their work only considers static VM placement. Moreover, DVMC is an online-time sensitive problem, and using multi-objective GA for large solution-space problems is highly time-consumed.

Farhad Ahamed et. al. [15] developed a security profile for VMs based on vulnerability analysis, intrusion detection, and trust-based analysis and introduced a security-aware VM consolidation [15] where VMs are categorized and isolated into security groups based on their security profiles. They do not take VMs dependencies and co-resistance into considerations and do not evaluate the impact of using their consolidation technique on cloud security and do not consider the gaps between VMs' security scores with the same security group.

3. Methodology

This paper uses a security assessment model based on [14] and proposes an SDVMC approach that consolidates the running VMs into the minimum number of hosts that keep the absolute value of the risk increase for each VM over its ideal overall risk score without colocation with other VMs does not exceed a maximum value defined by a RITH. Also, our SDVMC aims to result in a minimum overall mean risk with the same RITH constraint. The used security assessment model and the proposed SDVMC are presented in Sections 3.1 and 3.2.

3.1. Security assessment

Referring to [14], Risky VMs can be attacked directly or utilized as stepping stones to attack the host hypervisor, co-resident VMs, or VMs on other hosts and have a communication channel with it. Here it is assumed that all hosts use the same hypervisor with the same configuration, so this study neglects the impact of the hypervisor risk and uses a three-dimensional security assessment model with the following three parameters:

- *VM Base Risk Score* $\boldsymbol{R}_1$: The risk experienced by the VM itself, it can be measured based on the Common Vulnerability Scoring System CVSS, where each VM vulnerability is assigned a CVSS base score depending on its characteristics [17] and consider the CVSS base score for the highest vulnerability as the VM $\boldsymbol{R}_1$.
- *Network Communication Risk Score* R_2: The risk caused by the set of VMs that have direct network communication with the candidate VM. For a VM V^i, R_2^i is calculated as in Eq. (1).

$$\mathbf{R_2^i} = 1 - \prod_{\mathbf{j=1}}^{\mathbf{n}} \left(1 - \mathbf{R_1^j} * \boldsymbol{N_{ji}}\right) \quad (1)$$

where N_{ji} = 1 if V^j has a direct network connection with V^i and is placed on a different host otherwise N_{ji} = 0, and R_1^j is the base risk score for V^j.

When VM V^x with base risk R_1^x and has direct network communication with VM (V^i) enter the system or be migrated from the host that allocate V^i to another host, the network risk R2 for both machine will be affected. Suppose that the old network risk $R_{2\_old}^i$ for VM (V^i) before it is affected by $V^x$was:

$$R_{2\_old}^i = 1 - \prod_{j=1}^{n} \left(1 - R_1^j * N_{ji}\right)$$

Then we have $\prod_{j=1}^{n} \left(1 - R_1^j * N_{ji}\right) = (1 - R_{2\_old}^i)$

From Eq. (1), the new network risk $R_{2\_new}^i$ for V^iwill be

$$R_{2\_new}^i = 1 - ((\prod_{j=1}^{n} \left(1 - R_1^j * N_{ji}\right))(1 - R_1^x))$$

So we get

$$\mathbf{R_{2\_new}^i} = 1 - (\left(1 - \mathbf{R_{2\_old}^i}\right)(1 - \mathbf{R_1^x})) \quad (2)$$

where $R_{2\_old}^i$ is the network risk score for V^i before it is affected by the impact of V^x, and R_1^x is the base risk score for V^x, and $R_{2\_new}^i$ is the new network risk score for V^i after it is affected by the impact of V^x.

To estimate the network risk for any VM that has direct network communication with a set of VMs we can suppose an initial value for R2 equal 0 and iterate over the VMs set that have direct network communication with it to update R2 by the impact of each VM in the network set using Eq. (2).

- *Co-residency Risk Score* $\boldsymbol{R_3}$: The risk caused by the set of VMs that are collocated with the candidate VM on the same host. For a VM V^i that is allocated on host k, R_3^i is calculated as in Eq. (3).

$$\mathbf{R_3^i} = 1 - \prod_{\mathbf{j=1}}^{\mathbf{n}} \left(1 - \mathbf{R_1^j} * \boldsymbol{P_{jk}}\right) \quad (3)$$

where P_{jk} = 1 if V^j is placed on the same host k that allocate V^i otherwise P_{jk} = 0, and R_1^j is the base risk score for V^j.

In a similar way to estimating network risk score, we can estimate the impact of any coming VM to collocate with VM V^ion its co-resident risk. Also, we can estimate the co-resident risk score R3 for any VM V^i way by iterating over all its collocated VMs using Eq. (4).

$$R_{3\_new}^i = 1 - (\left(1 - R_{3\_old}^i\right)(1 - R_1^x)) \quad (4)$$

where $R_{3\_old}^i$ is the co-resident risk score for V^i before it is affected by the impact of V^x, and R_1^x is the base risk score for V^x, and $R_{3\_new}^i$ is the new co-resident risk for V^i after it is affected by the impact of V^x.

At any time instance, the overall security risk score R^i for VM_i is calculated as stated in Eq. (5).

$$R^i = 1 - \left(1 - R_1^i\right)\left(1 - R_2^i\right)\left(1 - R_3^i\right) \quad (5)$$

3.2. Security-aware dynamic VM consolidation (SDVMC)

The proposed SDVMC consists of two main modules, SMM and SDVMC module. The SMM periodically measures and calculates the security risk scores for all VMs according to the proposed security assessment model. Additionally, after each VM migration, new VM placement, or after any changes to the direct communication network of any VM, SMM updates the security risk scores for all VMs that can be affected by these changes. Also, SMM calculates and saves the ideal overall risk score (idealR) for each VM. For VM (V^i), where 1 < i < N and N is the number of VMs, the $\mathbf{idealR^i}$is the overall risk score $\mathbf{R^i}$ when it is not affected negatively by any VM outside its direct communication network and is calculated using Eq.6.

$$\mathbf{idealR^i} = 1 - (\left(1 - \mathbf{R_1^i}\right)\left(1 - \mathbf{R_c^i}\right)) \quad (6)$$

where $\mathbf{R_c^i}$ is calculated as $\mathbf{R_2^i}$ when V^i is not located with any of its direct communicated VMs on the same host.

SDVMC module divides the DVMC into 4 sub-problems as in [6], host overload detection, VM selection, host underload detection, and VM placement. We used Local Regression (LR) for overload detection and Minimum Migration Time (MMT) for VM selection. For under load detection we iterate all active and non-overloaded hosts. For each host, if it is possible to migrate out all of its allocated VMs to other active hosts, it will be switched to inactive. For VM placement we proposed a novel MRI with RITH VM placement algorithm.

Our proposed MRI with RITH VM placement algorithm seeks to select a host to allocate the coming vm with four main objectives. First, the selected host should have enough resources to allocate the coming VM. Second, at each VM placement, the absolute risk increase for the coming VM and for each vm in the host that will allocate it should not exceed the value of the RITH. Where the absolute risk increase for each VM (V^i) is the difference between its new overall risk after allocation ($\mathbf{newR^i}$) and its $\mathbf{idealR^i}$. Third, the proposed VM placement algorithm seeks to select the host that will result in a minimum total risk increase (totalRiskIncrease). Where totalRiskIncrease is the sum of differences between $\mathbf{newR^i}$ and the current overall risk ($\mathbf{R^i}$) for all affected VMs. Finally, if there is more than one suitable host can allocate the coming VM with the same totalRiskIncrease, the host with minimum power increase is selected.

The algorithm refers to the active and non-overloaded hosts as (activeNonOverloadedHostList), the inactive hosts as (inActiveHostList), the coming VM that needs a new placement as (vm), the VMs that has direct communication with vm as (vm. VMsCommunicationList), each VM member of vm.VMsCommunicationList as (commVM), the VM list located at a host that is probable to allocate the coming vm as (hostVMList) and each VM member of the hostVMList as (hostVM). The proposed algorithm deals with each vm as if it is a new VM with defined direct network communications with the running VMs and defined base risk score (vm.R1). So, for vm, the algorithm estimates the network risk score (vm.estimatedR2) that is affected by all vm.VMsCommunicationList and estimates co-residency risk score (vm.estimatedR3). Later, these estimated risk scores will be used in estimating the new network risk score (vm. newR2) and the new co-residency risk score (vm.newR3) for vm . vm.estimatedR2 and vm.estimatedR3 along with vm.R1 contribute to estimate the new overall risk score for (vm.newR).

Given the objectives of our proposed VM placement, for each vm the algorithm first calculates vm.estimatedR2 and set vm.estimatedR3 to 0, then, it iterates over activeNonOverloadedHostList and examines the hosts that have enough resources to allocate the coming VM. For each suitable host, it iterates over hostVMList and checks the mutual impact of each hostVM and vm on the other to update their co-resident and network risk scores. Hence the proposed placement algorithm calculates hostVM.newR. Depending on hostVM.newR, if the absolute risk increase for hostVM is greater than the RITH, the host is assumed to be unsuitable to allocate the coming VM. If not, the proposed algorithm updates totalRiskIncrease by adding the difference between hostVM.newR and its current overall risk hostVM.R. After iterating over all hostVMList, vm.newR is calculated as now vm.newR2 and vm.newR3 are completely estimated if vm is allocated to the underlying host. Consequently, if the absolute risk increase for vm is greater than the RITH, the host is assumed to be unsuitable. If not, the totalRiskIncrease is updated by adding the difference between vm.newR and vm.idealR, because vm is assumed to be not affected by any other VM outside its direct communication network. Finally, the suitable host with the minimum total risk increase is selected. If there is more than one suitable host gives the same total risk increase, the one with minimum power increase is preferred. If there is no suitable active host, the algorithm selects from inActiveHostList the one with the minimum power increase.

The target system for our approach is an (IaaS) Cloud, that contains M heterogeneous physical hosts and N heterogeneous VMs. Algorithm 1 shows the proposed MRI using RITH VM placement algorithm. If N is bigger than M and the maximum number of allowable communications for any VM is constant, the total complexity of the algorithm is $F(x) = O\left(N^2\right)$.

4. Experimental design

Simulation using CloudSim [6], was chosen to evaluate our approach on two stages. First, we evaluated the impact of SDVMC using MRI VM placement without RITH, simulation results are in 5.1. Second, we evaluated the impact of the RITH, simulation results are in 5.2. However, to get more accurate results CloudSim was examined before we evaluate our approach and there was a need to perform little modifications to its power estimation process and to its default PABFD VM placement which is utilized for comparison in our evaluation. These modifications are discussed in Section 4.3.

To evaluate the impact of SDVMC using MRI VM placement without RITH we applied the concept of using different isolated security groups which has been proposed by [16] where all VMs are categorized into a different number of security levels according to their risk degrees and all VMs with the same security level are isolated and consolidated into the minimum number of hosts. We compared our SDVMC using MRI VM placement without RITH to DVMC using the modified PABFD VM placement against different numbers of security groups.

To evaluate the impact of using the RITH, we applied the RITH to the modified PABFD and then evaluated the impact of MRI with RITH VM placement compared to the modified PABFD with the RITH against different RITH values. The reason behind this is to evaluate the impact of using our proposed RITH and also compare its performance with both MRI and PABFD VM placement algorithms.

Algorithm 1: MRI with RITH VM Placement

```
Input  : hostList, vmList, RITH
Output: allocationMap of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList do
    minRisk, minPower ← MAX
    alloocatedHost ← NULL
    vm.estimatedR2 ← 0
    foreach commVM in vm.VMsCommuniocationList do
        vm.estimatedR2 ←
        1 − ((1 − vm.estimatedR2) • (1 − commVM.R1))
    vm.estimatedR3 ← 0
    foreach host in activeNonOverloadedHostList do
        if host has enough resources for vm then
            hostSuitable ← True
            totalRiskIncrease ← 0
            vm.newR2 ← vm.estimatedR2
            vm.newR3 ← vm.estimatedR3
            foreach hostVM in hostVMList do
                hostVM.newR3 ← 1 − (1 − hostVM.R3)*(1 - vm.R1)
                vm.newR3 ← 1 − (1 − vm.newR3) • (1 − hostVM.R1)
                foreach commVM in vm.VMsCommuniocationList do
                    if hostVM is the commVM then
                        hostVM.newR2 ←
                        1 − ((1 − hostVM.R2)/(1 − vm.R1))
                        vm.newR2 ←
                        1 − ((1 − vm.newR2)/(1 − hostVM.R1))
                hostVM.newR ← (1 − hostVM.R1)•
                (1 − hostVM.newR2) • (1 − hostVM.newR3)
                if hostVM.newR − hostVM.idealR >= RITH then
                    hostSuitable ← False
                    break;
                totalRiskIncrease ←
                totalRiskIncrease + hostVM.newR − hostVM.R
            vm.newR ← (1 − vm.R1) • (1 − vm.newR2)•
            (1 − vm.newR3)
            if vm.newR − vm.idealR >= RITH then
                hostSuitable ← False
                break;
            totalRiskIncrease ←
            totalRiskIncrease + vm.newR − vm.idealR
            if hostSuitable == True then
                Power ← estimatePower(host, vm)
                if (totalRiskIncrease < minRisk) OR ((totalRiskIncrease
                == minRisk) AND (Power < minPower)) then
                    allocatedHost ← host
                    minRisk ← totalRiskIncrease
                    minPower ← Power
    if allocatedHost == NULL then
        foreach host in inActivehostList do
            Power ← estimatePower(host, vm)
            if (Power < minPower) then
                allocatedHost ← host
                minPower ← Power
    if allocatedHost != NULL then
        allocationMap.add(vm, allocatedHost)
Output: allocationMap
```

4.1. Simulation setup

CloudSim version 4.0 was used to simulate a cloud data center that contains 800 heterogeneous physical hosts, half of them are of type HP ProLiant ML110 G4 and the other half HP ProLiant ML110 G5. The specification characteristics of the used physical hosts and VM are in Tables 1 and 2, respectively.

4.2. Performance metrics

We studied the impact of our solution on security energy and QoS. The metrics used for each parameter are as follows:

4.2.1. Security measure

For any VM^i the time average for its $\mathbf{R^i}$, as presented in 3.1, overall its lifetime will represent its average risk score, and the mean of the average risk scores **MeanR** for all VMs running in the cloud datacenter will represent the risk score of the cloud.

Table 1
Physical Hosts Specification.

Host Type	Type 1	Type 2
Total MIPS	2660	1860
Total processor units	2	2
Total RAM	8 GB	8 GB
Network bandwidth	1 GB/s	1 GB/s
Total storage size	80 GB	80 GB

$$R^i_{avg} = \frac{1}{t - t_0} * \int_{t_0}^{t} R^i(t)dt \tag{7}$$

$$MeanR = \frac{1}{N} * \sum_{i}^{N} R^i_{avg} \tag{8}$$

where t_0 represents the beginning time VM^i in the system, t represents the end time of VM^i, R^i presents the instance security risk of VM^i and R^i_{avg} represents the time average of security risk for VM^i.

4.2.2. Energy measure

To estimate the power consumption rate for a host at any point in time, a power model is used based on CPU utilization for each hot type as in [4] which utilizes the data provided by [18]. Energy Ep consumption for a host along time tp will be.

$$Ep = (fromPower + (toPower - fromPower)/2) * tp \tag{9}$$

where fromPower and toPower is the power consumption rate at the beginning and the end of the time. The total energy consumption E is the aggregated energy consumption for all hosts all over their lifetime. The used host types and their power model are presented in Section 4.3.

4.2.3. QoS measure

A workload independent metric is used to represent the QoS Violations. This is Service Level Agreement violation SLA which is combined from both Performance Degradation due Migration PDM and Aggregated Overload Time Fraction (AOTF).

PDM: From [4] the migration time and performance degradation experienced by migrating a VM j are estimated as:

$$T_{m_j} = \frac{M_j}{B_j} \tag{10}$$

$$U_{d_j} = 0.1. \int_{t_0}^{t_0 + T_{m_j}} u_j(t)dt \tag{11}$$

where U_{d_j} represents the performance degradation by VM_j, t_0 is the beginning time of migration, T_{m_j} is the ending time of migration, $u_j(t)$ and M_j are the CPU utilization and used memory experienced by VM_j, and B_j is the available network bandwidth.

$$PDM = \frac{1}{N} \sum_{j=1}^{N} \frac{C_{d_j}}{C_{r_j}} \tag{12}$$

where N is the number of VMs, C_{d_j} is the estimate of the performance degradation of the VM_j which is the summation of all performance degradation caused by all migrations to this VM. In this work, C_{d_j} is estimated to be 10% of the CPU utilization in MIPS during all migrations of VM_j, C_{r_j} is the total CPU capacity requested by the VM_j during its lifetime.

AOTF: Represents the aggregated fraction of time during where active hosts have experienced the CPU utilization of 100%.

$$AOTF = \frac{1}{M} \sum_{i=1}^{M} \frac{T_{s_i}}{T_{a_i}} \tag{13}$$

where M is the number of hosts, T_{s_i} is the total time during which the host i has experienced the utilization of 100% leading to an SLA violation, T_{a_i} is the total active time of the host.

$$SLA = PDM * AOTF \tag{14}$$

4.3. CloudSim modification

CloudSim uses a power model for each host type to estimate the energy consumed by its hosts for a specific period, also this power model is used at each VM placement to calculate the power increase if a host is selected to allocate the underlying VM. However, when calculating the power increase at VM placement, hosts with zero utilization are assumed to be active and consumes power as if it is in the idle state. This way in estimating power increase for hosts with zero utilization in some cases leads to selecting hosts with zero utilization while there are other active hosts with non-zero utilization that can allocate the candidate VM. In contrast, when measuring the energy consumed by all hosts for a specific period, CloudSim assumes that hosts with zero utilization are in an inactive state with zero power consumption, however, these inactive hosts consume a small amount of power as stated in [19] for a typical blade server. To be more accurate, in this study hosts with zero utilization should always be transferred to inactive low power state mode and consumes a small amount of power which is assumed here to be 10 w for each inactive host. Table 3 show the modified power model for the used physical servers. Moreover, when searching for a host to allocate a VM migrated

Table 2
VM Specification.

VM Type	Type 1	Type 2	Type 3	Type 4
Total MIPS	2500	2000	1000	500
Total processor units	1	1	1	1
Total RAM	1 GB	1 GB	1 GB	1 GB
Network bandwidth	100 Mbit/s	100 Mbit/s	100 Mbit/s	100 Mbit/s
Total storage size	2.5 GB	2.5 GB	2.5 GB	2.5 GB

Table 3
Power consumption model for the used physical hosts.

Server	inactive	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant ML110 G4	10	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant ML110 G5	10	93.7	97	101	105	110	116	121	125	129	133	135

from an overloaded host, PABFD VM placement searchers through all non-overloaded hosts, active and inactive hosts. Here we modified it to search only through the active and non-overloaded hosts first if there is no suitable host we select from the inactive hosts if any. The reason for these modifications is that in our MRI VM placement algorithm we search first through active hosts before selecting a host from the inactive set.

To evaluate the impact of these modifications, DVMC was conducted before and after our modifications using the workload traces of randomly selected 10 days from PlanetLab [20] during March and April 2011. DVMC was conducted using Local Regression LR for overload Detection, Minimum Migration Time MMT for VM selection, and PABFD for VM placement. Table 4 presents the impact of our modification on Energy and QoS metrics. It is showed that our modifications provide better results, the main reason for this is using less number of hosts due to avoiding activating any inactive host for VM migration where we can select from the active set.

4.4. Security assumptions for simulation purpose

For simplicity and the purpose of simulation, each VM was assigned a fixed random score between 0 and 1, using a uniform distribution, to represent its base security risk score R_1, also we assumed that the communication network for all VMs does not change along with the simulation.

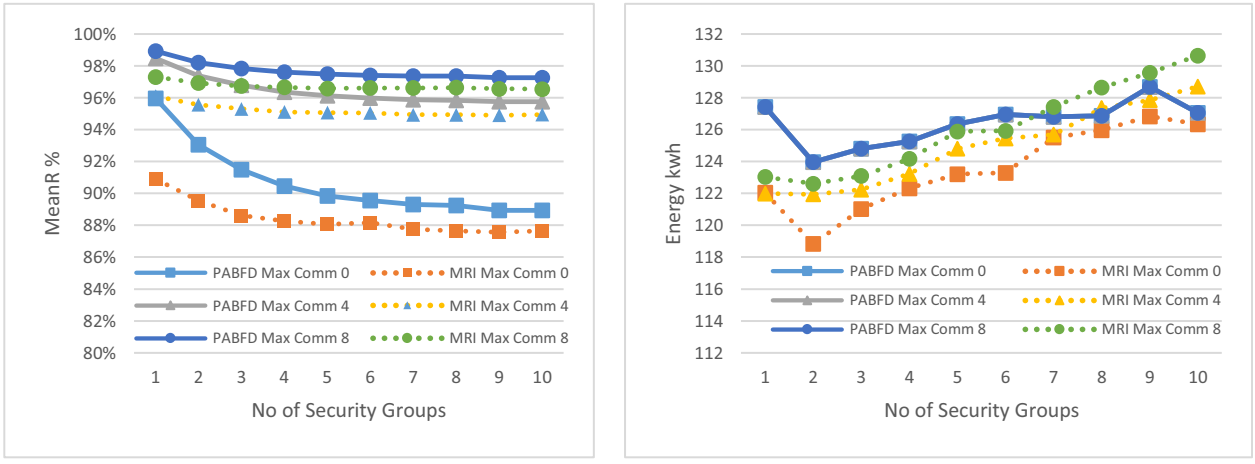
4.5. Workload preparation

Workload traces of day 05/03/2011 were selected to conduct our experiments. to achieve randomness, each experiment was repeated ten times using different values of R_1 and the mean of the ten runs was considered as the result.

To study the impact of different cases of communication-intensive between VMs that directly affect R_2, we considered three different cases. First, there is no communication between the VMs (0). Second, medium intensive communications ranging from 0 to

Table 4
Impact of CloudSim modifications.

Metric	Energy		No of VM migrations		PDM		AOTF		SLA		ESV	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Mean	162	133	28,175	13,910	0.08	0.04	6.21	6.43	4.97	2.42	7.95	3.17
Median	158	132	27,418	14,263	0.08	0.04	6.12	6.17	4.62	2.33	8.11	3.12



(a) Mean R

(b) Used Energy

(c) SLA

Fig. 1. impact of using MRI and PABFD with different number of security groups on different metrics (No TH).

4 as a maximum number of allowable communications for each VM. Finally, highly intensive communications, ranging from 0 to 8 as a maximum number of allowable communications. For all scenarios, each VMs was assigned a random number of connections with other VMs, between 0 and the maximum number of allowable communications. The VMs communication network for each VM is randomly selected according to the number of its connections. All random configuration settings are saved in configuration files.

5. Simulation results and discussion

Using the workload data as explained in Section 4.5, and given the experimental design mentioned in Section 4, we conducted extensive experiments to evaluate the performance of the proposed solution. Our results are presented and discussed in the following subsections.

5.1. Impact of using MRI without RITH

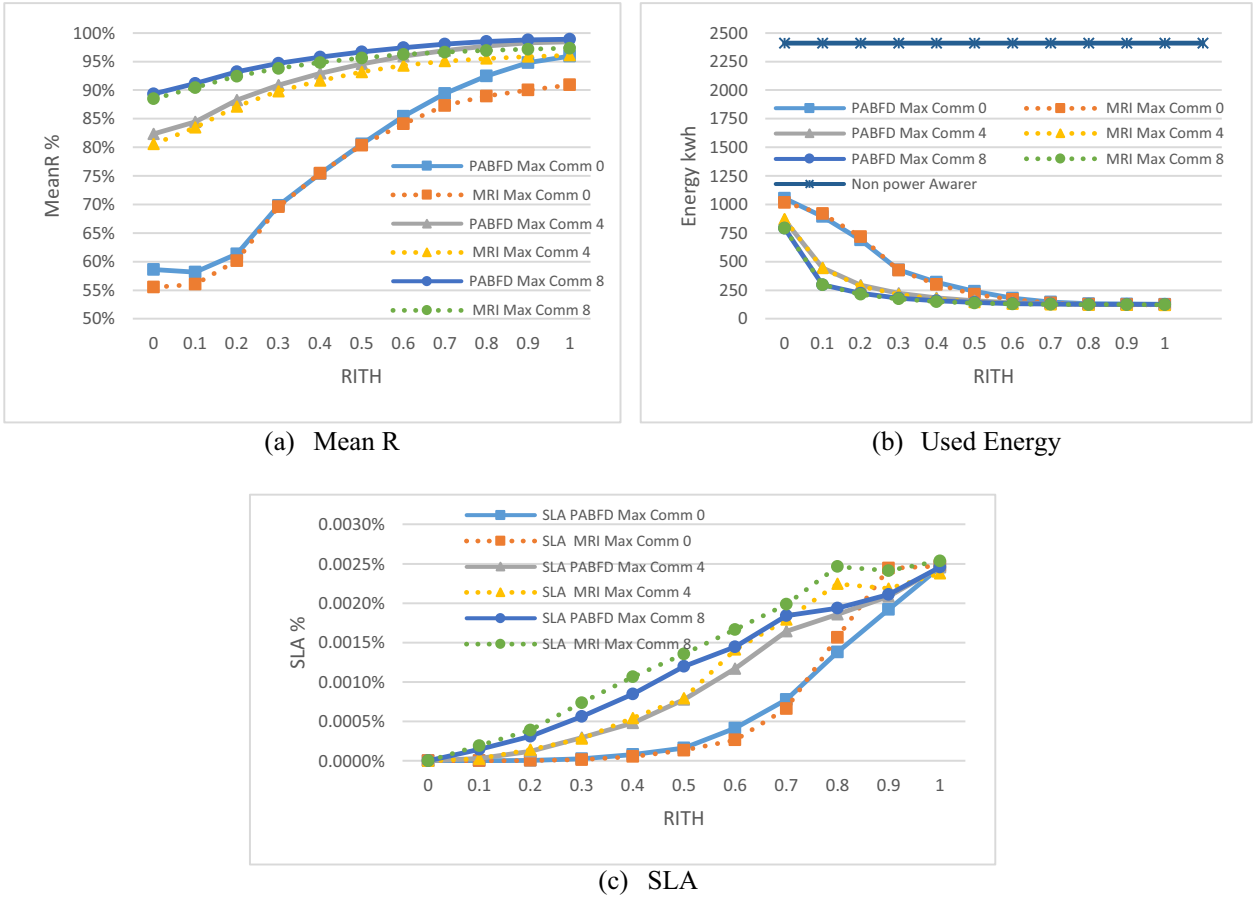
Fig. 1 illustrates the impact of using MRI VM Placement without RITH compared to using the modified PABFD VM placement under different number of isolated security groups ranging from one to ten, for different scenarios of communications defined in 4.5. All scenarios are evaluated by measuring MeanR, Used Energy, and SLA.

Sub-figure 1-a showed that, for MeanR, SDVMC using the proposed MRI VM placement without RITH always outperformed DVMC using the modified CloudSim default PABFD VM placement algorithm, the MeanR reduction of SDVM using MRI without RITH over DVMC using the modified PABFD ranged from 0.7% to 5.2%. While, sub-figure 1-b showed that SDVMC using MRI VM placement without RITH resulted in less energy consumption for most cases, the reduction in the used energy ranged from 0 kWh to 5 kWh. However, SDVM using MRI without RITH when using security groups greater than 6 for workload with max communication 8 and when using 10 security groups for workload with max communication 4, resulted in a small increase in the used energy than DVMC using the modified PABFD, the used energy increase ranged from 0 kWh to 3 kWh. Finally, sub-figure 1-c showed that for workloads with max communications 4 and both approaches give arbitrary the same SLA, while for workload with 0 communications SDVM using MRI VM placement without RITH gives slightly better results when using security groups less than 7.

Results also showed that by increasing the number of isolated security groups, energy consumption was getting larger where MeanR and SLA were getting smaller. Moreover, SDVMC using MRI VM placement without RITH with three isolated security groups resulted in better MeanR results from DVMC using the modified PABFD VM placement with any number of isolated security groups, up to ten security groups. Additionally, the results showed that the least values for the MeanR were 96.5%, 94.9%, and 87.5% for the workloads with maximum allowable communications of 8, 4, and 0 respectively, which were achieved by our SDVMC approach.

5.2. Impact of combining RITH with MRI

Fig. 2 illustrates the impact of using MRI with RITH VM Placement compared to using the modified PABFD VM placement on both of SDVMC and DVMC respectively. Performance is measured in terms of MeanR, Used Energy, and SLA. values of RITH ranged from 0 to 1 by 0.1 step. Where the value 1 for RITH means the max-



(a) Mean R (b) Used Energy

(c) SLA

Fig. 2. impact of using MRI and PABFD with security risk increase threshold (No isolated security groups).

imum probable risk increase is allowed and the value 0 states that no risk increase is allowed except there is no active or inactive host could be selected without VMs risk increase. Also, sub-figure 2-b compares the used energy for both algorithms to the maximum energy consumption which occurs when using a non-power-aware VM allocation policy, where all hosts are active and running all over the lifetime.

The results showed that by decreasing RITH, MeanR and SLA are getting smaller while Used Energy is getting larger. Moreover, the results showed that SDVMC using MRI with RITH resulted in slightly less MeanR and slightly more SLA than DVMC using the modified PABFD with the same values of the RITH while the used energy consumption pattern is arbitrary the same for both MRI and PABFD. However, it is showed that the effectiveness of the RITH depends on the intensive nature of the communications between the VMs. The maximum reduction that results from SDVMC using MRI with RITH over the DVMC using PABFD without RITH reaches about 40%, 18%, and 10% for workloads with maximum allowable communications 0, 4, and 8, respectively. Also, SDVMC using MRI with RITH at 0.8 for RITH decreases the MeanR by 5%, 3%, 2% over MeanR achieved by DVMC using PABFD without RITH for workloads with maximum allowable communications 0, 4, and 8, respectively with lower or equal energy consumption and SLA.

Moreover, the energy consumption in its maximum, which results from SDVMC using MRI with RITH at value 0, is still around half of the energy consumption when using a non-power aware VM allocation policy. Also, the results showed that using RITH with a value equal to or greater than 0.8 had an insignificant impact on both used energy and MeanR. While, the more reduction to the value of RITH below 0.1, the highly increase we get in the energy consumption with insignificant impact on both MeanR and SLA. Additionally, the fewer values used for RITH, the increase in the resulting energy consumption is getting large compared to the decrease in the resulting MeanR.

From results, it is observed that SDVMC using MRI without RITH, the used value for RITH is 1, results in limited security improvements without any negative impact on the energy consumption because it still makes the maximum possible VM consolidation. While SDVMC using MRI with RITH with values below 1 for RITH enables us to get significant security improvements at the expense of the energy consumption by controlling the increase in the overall risk for any VM over its overall ideal risk does not exceed the value of the RITH. Also, results showed that using RITH with values equal to or greater than 0.8 provides limited security improvements because most VMs may be risky 100% before the amount of risk increase reaches 0.8. While using RITH with small values provides significant security improvements but leads to weak consolidation and high energy consumption.

6. Conclusion

This paper presents an SDVMC approach that considers energy consumption, QoS, and security. The proposed SDVMC consists of two main modules, SMM that utilizes a three-dimensional assessment model, and SDVMC module which utilizes the proposed MRI with RITH VM placement algorithm. From simulation results, we conclude the following findings. First, SDVMC using MRI with RITH VM placement algorithm with a value 0.8 for RITH gives limited security improvements without affecting the energy consumption or QoS, for our defined workload cases the risk reduction ranged from 2% to 5%. Second, SDVMC using MRI with RITH VM placement algorithm with a value less than 0.8 for RITH enables tradeoff between energy consumption and the overall security risk, for our defined workload cases the maximum risk reduction ranged from 10% to 40% with energy consumption less than half of a non-power-ware VM allocation policy. Finally, for RITH with a value below 0.1, the more decreased value used for RITH the highly increase we get in the energy consumption with insignificant security improvements.

Future research directions may include working on developing a strongest security assessment model that considers Intrusion detection and different attack paths and attributes of different vulnerabilities. Also, it is important to evaluate theproposed approach in a real environment.

Acknowledgment

This work was supported by the Egyptian Academy of Scientific Research and Technology (ASRT), JESOR Grant. This project was entitled 'Resource-effective Cloud Data Center: Developing a dynamic data center management tool'.

References

[1] Vu K, Hartley K, Kankanhalli A. Predictors of cloud computing adoption: a cross-country study. Telematics Inf J 2020;52.
[2] Masanet E, Shehabi A, Lei N, Smith S, Koomey J. Recalibrating global data center energy-use estimates. J Sci 2020;367:984–6.
[3] Hamdi N, Chainbi W. A survey on energy aware VM consolidation strategies. J Sustain Comput: Inf Syst 2019;23:80–7.
[4] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. J Concurrency Comput: Pract Experience 2012;24(13):1397–420.
[5] Coppolino L, D'Antonio S, Mazzeo G, Romano L. Cloud security: emerging threats and current solutions. Comput Electric Eng J 2017;59:126–40.
[6] Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. J Software: Practice Experience 2011;41 (1):23–50.
[7] Abdelsamea A, El-Moursy AA, Hemayed EE, Eldeeb H. Virtual machine consolidation enhancement using hybrid regression algorithms. J Egypt Inf J 2017;18(3):161–70.
[8] Maiyza AI, Hassan HA, Sheta WM, Sadek NM, Mokhtar MA. End-user's sla-aware consolidation in cloud data centers. In: Proc. of 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). p. 196–204.
[9] Afoulki Z, Bousquet A, Rouzaud-Cornabas J, "A security-aware scheduler for virtual machines on iaas clouds", Tech. Rep. RR-2011-08, University of D'orleans, 2011, [Online] https://www.univ-orleans.fr/lifo/rapports.php?lang=en&annee=2011, (accessed 20 June 2020).
[10] Li M, Zhang Y, Bai K, Zang W, Yu M, He X. Improving cloud survivability through dependency based virtual machine placement. In: Proc. of international conference on security and cryptography (SECRYPT-2012). p. 321–6.
[11] Yuchi X, Shetty S. Enabling security-aware virtual machine placement in iaas clouds. In: Proc. of MILCOM 2015–2015 IEEE military communications conference. p. 1554–9.
[12] Miao F, Wang L, Wu Z. A VM placement based approach to proactively mitigate co-resident attacks in cloud. In: Proc of 2018 IEEE symposium on computers and communications (ISCC). p. 285–91.
[13] Kong T, Wang L, Ma D, Xu Z, Yang Q, Chen K. A secure container deployment strategy by genetic algorithm to defend against co-resident attacks in cloud computing. In: Proc of 2019 IEEE 21st international conference on high performance computing and communications; IEEE 17th international conference on smart city; IEEE 5th international conference on data science and systems (HPCC/SmartCity/DSS). p. 1825–32.
[14] Han J, Zang W, Chen S, Yu M. Reducing security risks of clouds through virtual machine placement. In: Proc. of IFIP annual conference on data and applications security and privacy. Springer; 2017. p. 275–92.
[15] Ahamed F, Shahrestani S, Javadi B, Garg S. Developing security profile for virtual machines to ensure secured consolidation: conceptual model no. January. In: Proc. of 13th Australasian symposium on parallel and distributed computing (AusPDC 2015). p. 27–30.
[16] Ahamed F, Shahrestani S, Javadi B. Security aware and energy-efficient virtual machine consolidation in cloud computing systems. In: Proc. of 2016 IEEE Trustcom/BigDataSE/ISPA. p. 1516–23.
[17] Mell P, Scarfone K, Romanosky S. Common vulnerability scoring system. IEEE Security Privacy J 2006;4(6):85–9.
[18] The SPECpower benchmark, [Online] https://www.spec.org/power_ssj2008/, (accessed 20 June 2020).
[19] Meisner D, Gold BT, Wenisch TF. Powernap: eliminating server idle power. J ACM SIGARCH Comput Architecture News 2009;37:205–16.
[20] Park K, Pai VS. Comon: a mostly-scalable monitoring system for planetlab. J ACM SIGOPS Operating Syst Rev 2006;40(1):65–74.