Ilorin Journal of Computer Science and Information Technology (ILJCSIT)
https://www.iljcsit.com.ng
ISSN: 2550–7214 (*print*)
Vol. 3, No. 1 (2020)

A Two-Phase Optimization Scheme with Efficient Prediction-based Triggering for Virtual Machine Placement in Cloud Datacenters

Rahimatu Hayatu Yahaya[1], Faruku Umar Ambursa[2]*, Bashir Shehu Galadanci[3]

1. *Department of Computer Science, Bayero University Kano, Kano, Nigeria,* raheemahabdulazeez@gmail.com
2. *Department of Information Technology, Bayero University Kano, Kano, Nigeria,* fuambursa.it@buk.edu.ng
3. *Department of Software Engineering, Bayero University Kano, Kano, Nigeria,* bashirgaladanci@yahoo.com

Abstract

Context: Cloud services have become one of the most critical IT infrastructures of today offering easy access to multiple services. Virtualization enables multiple clients and enterprises to leverage such cloud services simultaneously in order to optimize the use of resources and reduce the physical systems available. Virtual Machine Placement (VMP) is concerned with selecting appropriate physical machines for each virtual machine request. For solving the VMP problem, two-phase optimization approach is used recently. In the two-phase scheme, the first phase, also called the incremental VMP (iVMP) phase, involves dynamic handling of arriving VM requests from users, where VMs could be created, modified or destroyed at runtime. The second phase, referred to as the VMP reconfiguration (VMPr) phase, deals with improving the quality of solutions obtained in the iVMP phase. The execution of the VMPr phase is triggered from the iVMP phase. However, an important issue is when or under which circumstances the VMPr phase should be triggered. **Objective:** This work aims to propose an improved two-phase optimization scheme for the VMP problem using a new triggering method. **Methods:** The new triggering method is based on prediction approach using a Damped Trend Exponential Smoothing technique. The proposed method evaluates and more precisely specifies when to activate the VMPr phase for re-calculation and reconfiguration of the placement. Taking into account 400 different scenarios, experimental evaluation was performed against the benchmark research. **Results:** The results show that the proposed VMPr triggering technique recorded better aggregate scores of power consumption, economic revenue, resource usage and the algorithm reconfiguration time, than the benchmark with more than 12% improvement. **Conclusion:** The proposed method is therefore a novel approach to trigger the VMPr phase to achieve better optimization results from the reconfiguration phase of the two-phase optimization scheme.

Keywords: Virtual machine placement, Cloud computing, Incremental VMP, VMP reconfiguration

1. Introduction

Due to the increasing elasticity, reliability, information sharing capability and low cost of a single processor, distributed computing models such as cluster, grid, and cloud have gained significant popularity over the years (Khanchi & Tyagi, 2016). In the current situation, however, cloud computing is the most popular distributed computing paradigm out of all others. It has now become one of the most critical IT infrastructures allowing ubiquitous, viable on-demand network access to multiple services that are easily supported and delivered by the provider with very minimal managerial effort (Paliwal, 2014). One of the major facilitators of the cloud computing paradigm in this respect is virtualization technology, which includes the abstraction of users' physical resources (Angeles, 2014). Virtualization enables multiple clients and enterprises to leverage cloud services simultaneously to optimize the use of resources and reduce the physical systems available. According to the National Institute of Standards and Technology (NIST), virtualization software allows the simultaneous running of different operating systems and applications on the same server. This enables companies to lower their costs while increasing the efficiency, use and reliability of their current computer hardware (Mell & Grance, 2009).

While cloud computing provides the above-mentioned benefits, it presents many challenges that need to be addressed. One of these challenges is resource scheduling, which is assigning and optimizing cloud resources with the main goal of optimizing the use of the various resources

(CPU, storage, network bandwidth, etc.) for purposes such as energy conservation and load balancing to improve the profitability of providers (Rochwerger et al., 2009). One of the challenges cloud providers face in resource allocation is the virtual machine placement problem (VMP) (Vijaypal, Pateriya, & Rajveer, 2015). VMP is concerned with the process of selecting an appropriate physical machine for each virtual machine request. As shown in Figure 1, a suitable physical machine (PM) is selected for each virtual machine (VM) request based on a VMP technique. The VMP problem is an NP-Hard issue, so there are no suitable placement algorithms for the problem; hence the time to reach a solution significantly grows with the problem size (Speitkamp & Bichler, 2010).

The VMP has been extensively researched in a variety of research works, many of which concentrate on improving objective functions like energy usage (Goudarzi & Pedram, 2012; Prevost, Nagothu, Kelly, & Jamshidi, 2013), optimizing network congestion (Biran et al., 2012; Hong, Chen, Huang, Chen, & Hsu, 2013), performance maximization (Gupta, Milojicic, & Kal'e, 2012), economic cost optimization (Mark, Niyato, & Chen-Khong, 2011; Xu & Fortes, 2010) as well as resource usage (Li, Zheng, Wu, & Du, 2015). In contrast, some of the work done puts emphasis on objective optimizations (Pires & Baran, 2015) such as mono objective (Wang, Liu, Zheng, Sun, & Yang, 2013), multiple solved as mono objective (Amarilla, Benıtez, Zalimben, Pires, & Baran, 2017; Chamas, Lopez, & Baran, 2017; Monil & Rahman, 2016) and pure multiple objectives.

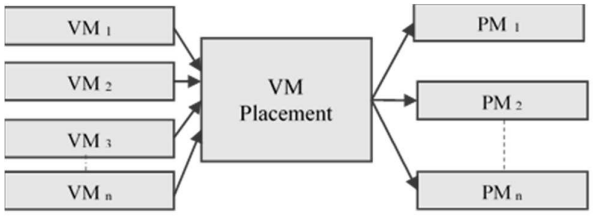

Figure 1. Virtual Machine Placement Technique.

The formulation of the VMP can be online, offline, or two-phase. An online heuristics method is used by VMP online formulation to perform dynamic placement of VMs to suitable PMs at runtime. Such online formulations generally lead to an improved sub-optimal solution (Pires & Baran, 2015). On the contrary, by considering a static environment where VMs are put on more suitable PMs, an offline metaheuristic approach is used to formulate VMP problems. While the offline formulation offers better alternatives than the online formulations, it is not ideal for the random nature of the cloud computing environment. A two-phase formulation was proposed in this regard to provide more efficient solutions to the VMP problem, combining the advantages and positive features of online and offline approaches where a heuristic process is used to receive VMs for placement at runtime (online), and a metaheuristic process is used to perform placement optimization to accomplish more optimal (offline) solutions (Pires & Baran, 2014).

As the name implies, the two-phase VMP optimization is carried out in two phases. The first phase (online phase), also called the incremental VMP (iVMP) phase, involves dynamic handling of arriving VM requests from users, where VMs could be created, modified or destroyed at runtime. The second phase (offline phase), also referred to as the VMP reconfiguration (VMPr) phase, deals with improving the quality of solutions obtained in the iVMP phase. This requires reconfiguring the placement through VM migration considering static environments. The execution of the VMPr phase is triggered in the iVMP phase. However, an important issue here is when or under which circumstances the VMPr phase should be triggered (Lopez-pires, Baran, Benitez, Zalinmben, & Amarilla, 2017). In other words, when and how the VMPr is triggered is critical to the success of the overall optimization solution. For this reason, numerous techniques have been applied.

Lopez-pires et al., 2017 proposed a prediction-based VMPr triggering method that statistically analyzes and proactively detects situations where VMPr triggering is potentially required for a placement reconfiguration. The proposed method used the double exponential smoothing (DES) technique for the prediction. However, for an indefinite period, the forecasts formed by DES show a persistent trend, and it was discovered that this method appears to over-predict, especially for long forecast periods (Hyndman & Athanasopoulos, 2018). Therefore, the result of the prediction becomes less precise. As a consequence, the reconfiguration phase results in a less ideal solution, which affects performances in terms of cloud power consumption, economic revenue, resource usage, and the algorithm reconfiguration period. As such, it is necessary to develop a better optimization scheme for the VMP based on a more precise prediction process. In this work, a new prediction-based triggering is proposed based on Damped Trend Exponential Smoothing. The proposed method evaluates and more precisely specifies when to activate the VMPr phase for re-calculation and reconfiguration of the placement. The new method was evaluated and compared with the previous method. Results showed that the proposed strategy recorded better aggregate scores of power consumption, economic revenue, resource usage, and the algorithm reconfiguration time, than the previous method with more than 12% improvement.

The remainder of this paper is organized as follows. The next section (Section 2) discusses the related works. Section 3 explains, in detail, the proposed method. The performance evaluation of the proposed solution is demonstrated in Section 4. Section 5 presents the conclusion of the study.

2. Related Works

This section presents some of the previous studies related to our work. Numerous studies exist for the problem under study based on the periodical, threshold-based, and prediction-based techniques.

Periodical triggering is generally the simplest model where the VMPr cycle is triggered occasionally after some fixed time instance. Several slightly different methods have been proposed. One of the earliest of these methods was the Backward Speculative Placement (BSP) technique, proposed by (Calcavecchia, Biran, Hadad, Moatti, & Y., 2012) that uses the Best Fit Decreasing (BFD) approach for managing the request stream (iVMP) and the periodic optimization (VMPr) to migrate the most commonly loaded hosts. The periodic optimization cycle is triggered periodically over a period of time (e.g., 10 seconds instant) to boost the current placement but it is terminated when a new request arrives. In another approach presented by (Farahnakian, Bahsoon, Liljeberg, & Pahikkala, 2016), a Self-Adaptive Resource Management System (SARMS) is proposed where, using the Adaptive Utilization Threshold (AUT) method, PMs were categorized as overweight or underweight. The BFD heuristic is employed in the iVMP process, from which VMPr is periodically activated. The VMPr process, however, first migrates VMs from overweight PMs and then consolidates VMs to release underweight PM resources, switching them into hibernation mode. Other techniques include the VMP Biogeography-based Optimisation (VMPBBO) method particularly geared towards green cloud computing, proposed by (Zheng et al., 2016), to minimize the waste of resources and power usage during the consolidation process and the live migration algorithm based on the basic set (LMABBS), as explained by (Yue & Chen, 2014), which conducts server consolidation by grouping VMs to as few PMs as possible and shutting down or placing idle PMS in a minimal-energy state to reduce energy consumption. Similarly, the work presented by (Azizi, Zandsalimi, & Li, 2020) proposed an efficient heuristic to model power consumption and resource wastage minimizations by first sorting the PMs in non-ascending order of their power efficiency where the first PM is selected and then requested VMs are placed on available PMs based on a Resource Usage Factor model using a reward and penalty mechanism for the iVMP phase. Afterward, resource utilization is optimized across multiple dimensions to reduce resource wastage and the number of active PMs for the VMPr phase.

In general, although the periodic triggering of VMP reconfiguration appears to be a simple approach, it presents some important drawbacks when defining fixed periods. Specifically, a migration of VMs may be required before the appointed period, where optimization opportunities could be wasted or even economical penalties could impact the cloud datacenter operation. Conversely, in some scenarios, the reconfiguration may not be necessary, and therefore initiating the VMPr would be akin to a waste of time and resources.

The threshold-based approach partially addresses these problems by establishing thresholds for the use of PM resources (e.g., CPU). Thresholds indicate when a PM (Hi) is perceived to be under-loaded or full and will activate a VMPr phase as a result. One of the several versions of this approach is the decentralized threshold-based decision process, proposed by (Baloglazov, Abawajy, & Buyya, 2012), which uses a Modified Best-Fit Decreasing (MBFD) algorithm to sort a list of VM requests and allocate each VM to a PM which provides the lowest possible increase in power consumption. Another approach, advanced in (Shi, Furlong, & Wang, 2013) has the assignment optimization (VMPr) phase triggered based on definite utilization thresholds of PM resources to allocate the characterized VMs to PMs of the allocated type. However, the allocation of incoming VM requests is not considered during placement optimization. There is also a technique presented by (Tighe & Bauer, 2014) which proposed an auto-scaling algorithm as a fragment of the Dynamic VM Allocation (DVMA) algorithm that is triggered to perform placement optimization whenever an under-loaded or overloaded PM is identified based on some specified utilization thresholds. However, the work does not consider the allocation of new VM requests when the optimization phase is triggered. In another approach presented by (Braiki & Youssef, 2019), VM requests are sorted in non-decreasing order by CPU utilization and are allocated to PMs for iVMP, while the current placement is periodically optimized based on a fuzzy-based multi-objective BFD method (VMPr). Similarly, the work by (Haghighi, Maeen, & Haghparast, 2018) proposed a technique that used a k-means clustering approach to cluster VMs and independent tasks based on their CPU processing power (MIPS); each cluster of tasks is then assigned to the appropriate cluster of VMs using the first come first serve (FCFS) algorithm for iVMP and placement optimization by migration is performed when an under-loaded or overloaded host is detected for VMPr. Other techniques include the dynamic programming technique for resource waste and power consumption optimization presented by (Chang, Gu, & Luo, 2016) that used knapsack approach for selecting PMs, sorting VMs where resources are assigned to VMs with higher resource requests and dynamic placement where the VM request will be assigned to the most appropriate PM in the iVMP phase. However, when a PM is not suitable for a VM request, the requests are sorted in non-decreasing order and the first VM request from the list is assigned to the PM for VMPr phase.

The various problems related to the periodical and threshold approaches are addressed by prediction-based VMPr triggering methods which attempt to proactively determine situations where a VMPr triggering is potentially required for a placement reconfiguration. This is generally achieved through statistically analysing and optimizing some objective functions. One approach, as explained by (Chamas et al., 2017), is to model the relevant uncertain parameters in the IaaS (Internet as a Service) environment considering four objective functions which, in the placement optimization phase (VMPr), the Ant Colony Optimization (ACO) algorithm was used to globally

reconfigure the current placements. Advanced by (López-Pires, Barán, Pereira, Velázquez, & González, 2018), a first implementation of the previously proposed work by (Lopez-pires et al., 2017) was performed in OpenStack and the filter scheduler was used to perform the proposed experiment. However, the recalculation time is not considered as a function of time t for real-world operations. Similarly, (López-Pires, Barán, Pereira, Velázquez, & González, 2019) presented an experimental evaluation of 36 different optimization algorithms for the two phase virtual machine placement algorithms in green cloud datacenters to identify the best algorithm for minimizing power consumption where four dynamic parameters and 400 scenarios were considered. Experimental results show that the two-phase algorithm with the prediction-based triggering and update-based recovery methods are best suited for power consumption minimization. In another approach described by (Lopez-pires et al., 2017), a predictive triggering and recovery method was proposed where the VMP reconfiguration (VMPr) phase applies a Memetic algorithm to make a centralized decision to reconfigure the placement of VMs globally. In this approach, the VMPr phase is further divided into two techniques; (i) the VMPr triggering technique which is a technique that determines when the VMPr phase is to be activated and (ii) the VMPr recovery technique to determine what to do with new incoming VM requests during the recalculation time of VMPr. The proposed VMPr triggering model uses a statistical method known as double exponential smoothing (DES) to forecast values of the global objective function F (x, t) to decide when to activate the VMPr phase in the two-phase optimization scheme. As shown in the example in Fig. 2, taken from the work presented by (Yahaya & Ambursa, 2019), Once the VMPr is activated, it recalculates the location of VMs at a discrete-time t within β discrete time slots (i.e. recalculation time). Given the static nature of the VMPr phase, the recalculated placement may be out of date. In addition, while the VMPr is calculating, the iVMP could still receive and serve arriving requests, making the VMPr calculated solution obsolete; thus, the recalculated placement must be retrieved correctly using a VMPr recovery process, before full reconfiguration is performed.

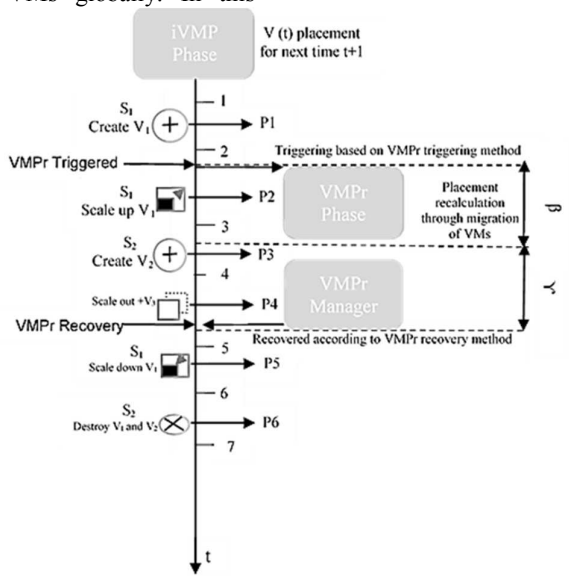

Figure 2. Two-Phase Allocation Scheme (Yahaya & Ambursa, 2019).

3. Proposed Two-Phase Optimization Scheme for VM Placement

In this section, we explain, in detail, our proposed two phase VMP scheme with a new prediction-based triggering method. As in previous works, the proposed optimization scheme is split into two phases; the incremental VMP (iVMP) phase in which VMs are collected and positioned at runtime (online) and the VMP reconfiguration (VMPr) phase in which VMs are reconfigured to provide a more effective solution (offline) (Pires & Baran, 2015). The focus of this work is on developing an efficient method for triggering the VMPr phase. The algorithms for the two phases are adopted from previous works. Therefore, we first briefly explain the iVMP and VMPr phases along with the adopted algorithm in each and then demonstrate, in detail, our proposed method for triggering the VMPr from the iVMP phase.

3.1. Incremental VMP (iVMP) phase

At runtime, the incremental iVMP phase receives online placement requests. In the iVMP process, where VM requests change over time, a First Fit Decreasing (FFD) heuristic (Lopez-pires et al., 2017) is adopted to solve the online VMP problem. The FFD heuristic begins with the elimination of cloud service, then the size of resources and the scale of cloud service is regulated. After this, requests for VMs are grouped by revenue in decreasing order, and then cloud service scale-out, as well as VMs resource scale-up, are managed. It subsequently assigns the request to the first PM with sufficient resources to fulfill its demands for each unprocessed VM request; else, it allocates to another federated provider if no PM has sufficient resources to fulfill it. Finally, it updates and returns the placement. A detailed demonstration of FFD for

the iVMP phase can be obtained in Lopez-pires et al. (2017).

3.2. VMP Reconfiguration (VMPr) phase

The VMPr phase performs placement re-evaluation and reconfiguration for a static environment. In this phase, the Memetic Algorithm (MA) heuristic is used and it functions as follows: A specified population of candidate solutions is generated randomly. These candidate solutions are then remedied to validate that the population has only viable solutions that meet the constraints. Next, using local search, the algorithm attempts to enhance candidate solutions. It then tests the solutions obtained and selects the first best option from the population collection based on the given criterion factor. Then an iteration is counter initialized and an evolution process is applied by selecting solutions from the populations and the best-known option. Following this, crossover and mutation operators are performed and the solutions obtained are repaired to contain only feasible solutions. The solutions might be improved using local search and the best-known option (if necessary), as well as the iteration counter, are updated. The process of evolution is repeated until it meets the algorithm's stopping condition. Finally, the best-known placement reconfiguration solution is returned.

It is important to emphasize that this work's main contribution centres on the triggering technique of VMPr that specifies when to activate the VMPr phase in the two-phase optimization scheme. This is addressed in the next subsection.

3.3. Proposed VMPr Triggering Method

In a two-phase optimization scheme for VMP problems, the VMPr triggering method defines when to activate the VMPr phase. Previous works proposed a statistical VMPr triggering method based on double exponential smoothing (DES). However, for an indefinite period, the forecasts formed by DES show a persistent trend and it was discovered that these methods appear to over-predict, especially for long forecast periods (Hyndman & Athanasopoulos, 2018). Therefore, the result of the prediction becomes less precise. As a consequence, the reconfiguration phase results in a less ideal solution. As such, it is necessary to develop an improved two-phase optimization scheme based on a more precise prediction process.

To address the above issue, we propose a new prediction-based triggering method based on damped trend exponential smoothing technique (see Eqs. (1) - (3)). This technique has been proven to be very successful for numerous statistical analyses (Venigella, 2010). The proposed method evaluates the optimized objective function F (x, t) and specifies when to activate the VMPr phase for re-calculation and reconfiguration of the placement. The damped trend model consists of smoothing parameters α and β as well as a damping parameter ϕ, with values between 0 and 1.

$$F_{t+h/t} = E_t + \phi^h T_t \quad (1)$$

$$E_t = \alpha N_t + (1-\alpha)(E_{t-1} + \phi T_{t-1}) \quad (2)$$

$$T^t = \beta(E_t - E_{t-1}) + (1-\beta)\phi T_{t-1} \quad (3)$$

where,

E_t indicates the expected value of f (x,t) at time t.

T_t signifies the trend of f (x,t) at time t.

$F_{t+h|t}$ is the value of f (x, t + 1) predicted at time t.

α shows the smoothing parameter for level, 0< α <1

β shows the smoothing parameter for trend, 0< β <1 and

ϕ denotes the damping parameter, 0< ϕ <1.

The proposed triggering technique, as shown in Algorithm 1, initially checks if there are values of x (t) (existing placement from iVMP phase); if values of x (t) exist, then get the values, compute the expected value (E_t) and the trend value (T_t) of F (x, t) (global objective function) using the damped trend exponential smoothing (DTES) and next compute the forecast of N values of F (x, t+1). Afterward, the algorithm checks if the forecasted values of F1 <F2 <Fn are true; if correct, reconfiguration is triggered and the VMPr phase is returned; else, the algorithm exits and continues placement in the iVMP phase.

INPUT: values of x (t)
OUTPUT: reconfiguration phase (VMPr phase)

Begin:
Step1: Check values of x (t)
Step 2: If x (t) exists then
Step 3: Get n values of x (t)
Step 4: Compute E_t and T_t using DTES technique
Step 5: Estimate $F_{t+h/t}$ for n values of f (x, t)
Step 6: If values of $f_1(x) < f_2(x) < F_n(x)$
Step 7: Activate reconfiguration then
Step 8: Return VMPr phase
Step 9: End
Step 10: Else, return iVMP phase
Step11: End

End

Listing 1. Proposed Damped Trend Triggering Algorithm.

4. Performance Evaluation

This section presents the experimental environment that is considered for this work, such as computer resources and experimental set-up. These resources are used to examine the performance of this research work. The experiments were conducted on a GNU Linux Operating System with an Intel(R) dual Core TM i7-4600U CPU@ 2.10 GHZ and 16 GB of RAM. The evaluated algorithms were implemented in the Java programming language using NetBeans IDE.

Experimental Settings

In this work, the assumptions and experimental settings, parameters, and scenarios were adopted from the benchmark work (Lopez-pires et al., 2017). The IaaS provider owns a set of PMs, where each PM i is represented by r different physical resources. This work considers r = 3 physical resources ($Pr_{1,i}$ to $Pr_{3,i}$): CPU [EC2 Compute Unit (ECU)], RAM [GB] and network capacity [Mbps]. The maximum power consumption ($pmax_i$) [W] is also considered as shown in Table 1. Other considered parameters for the physical resources include: recalculation time $\beta = 1$; protection factor for each resource k: $\lambda k = 0.5$; and Penalty factor for each resource k: $\varphi k = 1$.

In addition, there were 4 different types of PMs (H.small, H.medium, H.large, H.xlarge) considered as input data as depicted in Table 1. Given the considered PM types, five IaaS data centers with a different number of PMs (DC1 to DC5) were considered as shown in Table 2. Furthermore, at each specified time t, 80 distinct traces of the workload of requested cloud services (V(t)), their specifications were considered as input data as well as their utilization U(t) of resources. The workload traces were generated using a Cloud Workload Trace Generator (CWTG) for provider-oriented VMP problems (Tchernykh, Schwiegelsohn, Alexandrov, & Talbi, 2015). The spec for the virtual resource capacities (i.e., Vr1,j-Vr3,j) of the requested VM instances in terms of CPU [EC2 Compute Unit (ECU)], RAM [GB] and network capacity [Mbps] as well as the associated economical revenue (Rj($)), is given in Table 3. This is in line with the Amazon Elastic Compute Cloud (EC2) instance forms.

It is important to bear in mind, however, that the following unpredictable parameters were considered (i) virtual resource capability (vertical elasticity), (ii) number of VMs comprising cloud services (horizontal elasticity), (iii) use of virtual resource CPU and RAM, and (iv) use of virtual resource networking (both applicable for overbooking). Therefore, two distinct Probability Distribution Functions (PDFs) (i.e. Uniform and Poisson) represent each parameter behavior.

Table 1. PM types considered for simulations.

PM type	$Pr_{1,i}$ [ECU]	$Pr_{2,i}$ [GB]	$Pr_{3,i}$ [Mbps]	$P\,max_i$	c_i
H.small	32	128	1000	800	1
H.medium	64	256	1000	1000	1
H.large	256	512	1000	3000	1
H.xlarge	512	1024	20000	5000	1

Table 2. Number of PMs per each considered IaaS data center.

PM type	DC1	DC2	DC3	DC4	DC5
H.small	50	30	20	15	10
H.medium	50	30	20	10	10
H.large	50	30	15	10	9
H.xlarge	30	10	8	10	8

Table 3. VM types from Amazon EC2 considered for cloud service requests in simulations.

ID Instance	Type	Vr1,j [ECU]	Vr2,j [GB]	Vr3,j [Mbps]	Rj [$]
0	m4.large	6.5	8	450	0.12
1	c4.large	8	3.75	500	0.105
2	m4.xlarge	13	16	750	0.239
3	c4.xlarge	16	7.5	750	0.209
4	m4.2xlarge	26	32	1000	0.479
5	c4.2xlarge	31	15	1000	0.419
6	m4.4xlarge	53.5	64	2000	0.958
7	c4.4xlarge	62	30	2000	0.838
8	m4.10xlarge	124.5	160	4000	2.394
9	c4.8xlarge	132	60	4000	1.675
10	x1.32xlarge	349	1952	10000	13.338

4.1. Objective Functions

This work considers the optimization of four objective functions, which defines the evaluation metrics for the study. They are (i) power usage, (ii) economic revenue, (iii) resource usage, and (iv) reconfiguration period (Lopez-pires et al., 2017). Each objective function cost, however, is normalized and the four functions were merged as a single context of optimization, considering the minimization approach. In this section, the steps in arriving at the objective function are provided. These steps are adopted and modified from the previous works.

4.1.1. Power consumption

The minimization of power consumption, as suggested by (Baloglazov et al., 2012), is expressed by the sum of the power consumption of each PM comprising the IaaS environment, as given in Eq. (4).

$$f_i\,(x,t) = \sum_{i=0}^{n}\Big((pmax_i - pmin_i) \times Ur_{1,j}(t) + pmin_i\Big) \times Y_i(t) \quad (4)$$

where,

x denotes the evaluated solution of the problem, $f_1(x, t)$ shows the overall power consumption of PMs at instant t, $pmax_i$ gives the maximum power consumption of a PM H_i, $pmin_i$ shows the minimum power consumption of a PM H_i.

$Y_i(t) \in \{0, 1\}$ indicates if H_i is turned on ($Yi(t) = 1$) or not ($Yi(t) = 0$) at instant t.

4.1.2. Economic revenue

The maximization of the total economic revenue earned by an IaaS provider is accomplished by reducing the total revenue from leasing resources from the cloud federation's alternate data centers as well as the total economic penalties for SLA violations. This is defined as in Eq. (5).

$$f_2\,(x,t) = LC(t) + EP(t) \quad (5)$$

where,
f_2 (x, t) shows the overall economical revenue of the main IaaS provider at instant t.

4.1.3. Resources utilization

Maximizing resource utilization is accomplished by reducing the average ratio of resources wasted on each PM Hi (i.e. resources not allocated to any VM) as represented in Eq. (6).

$$f_3(x,t) = \frac{\sum_{i=1}^{n}\left[1-\left(\frac{\sum_{k=1}^{r} Ur_{k,j}(t)}{r}\right)\right]\times Y_i(t)}{\sum_{i=1}^{n} Y_i(t)} \quad (6)$$

where,

$f_3(x,t)$ is the average ratio of wasted resources at instant t, $Ur_{k,i}(t)$ denotes the utilization ratio of resource k of PM H_i at instant t and r depicts the number of considered resources, in this case, r=3, i.e CPU, RAM, and network capacity).

4.1.4. Reconfiguration time

VM's migration time for placement reconfiguration is expected to be as low as possible, so minimizing the (total) reconfiguration time could be accomplished by minimizing the maximum amount of memory to be migrated from one PM Hi to another H_i' as defined in Eq. (7).

$$f_4\,(x,t) = \left(MT_{i,i}\right)\ \forall i, i \in \{1, \dots, n\} \quad (7)$$

where,

$f_4(x, t)$ denotes the network traffic overhead for VM migrations at instant t $MT_{i,i}'$ gives the overall amount of RAM to be migrated from PM H_i to H_i.

4.1.5. Aggregate Cost Function (ACF)

For optimization purposes, the four metric functions, defined above, are combined into a single optimization metric. To combine the metric functions, the cost of each objective function is first normalized by computing $\hat{f}_i(x,t) \in R$ i.e $0 \le \hat{f}_i(x,t) \le 1$, as in Eq. (8);

$$\widehat{f_i}\,(x,t) = \frac{f_i\,(x,t) - f_i\,(x,t)_{min}}{f_i\,(x,t)_{max} - f_i\,(x,t)_{min}} \quad (8)$$

The functions are combined into a single objective taking into consideration a minimum Euclidean distance to the origin as defined in Eq. (9). Let F (x, t) represent the aggregate function cost. The cost is defined as;

$$F\,(x,t) = \sqrt{\sum_{i=1}^{q} \widehat{f_i}\,(x,t)^2} \quad (9)$$

where,

F (x, t) denotes a single objective function combining each $\hat{f}_i(x,t)$ at instant t, $\hat{f}_i(x,t)$ represents the normalized cost of the objective function $f_i(x, t)$ at instant t and q indicates the number of objective functions (in this case q = 4).

Furthermore, like in previous works, this work also considers the uncertainty of the evaluation parameters (i.e. (i) virtual resource capacities (ii) number of VMs that compose cloud services (iii) utilization of CPU and RAM virtual resources, and (iv) utilization of networking virtual resources) due to randomness of customer requests and generally the dynamic nature of cloud environments. Uncertainty is usually modelled through a set of scenarios S, where the uncertain parameters are considered. To model the uncertainty, a time-based average value of the aggregate cost function F (x, t) is provided for each scenario $s \in S$ as shown in Eq. (10).

$$\overline{\mathrm{F}_s(\mathrm{x},\mathrm{t})} = \frac{\sum_{t=1}^{tmax} \mathrm{F}\,(\mathrm{x},\mathrm{t})}{\mathrm{tmax}} \quad (10)$$

Where

$\overline{\mathrm{fs}(\mathrm{x},\mathrm{t})}$: Time-based average of the combined objective function for all discrete-time instants t in scenario $s \in S$.

t_{max} is the scenario duration.

Finally, in this work, a solution from a set of candidate solutions is selected using the average criteria (Aloulou & Croce, 2008), which is defined as the average $\overline{\mathrm{F}_s(\mathrm{x},\mathrm{t})}$ for all scenarios $s \in S$, as shown in Eq. (11).

$$A_{criteria} = \overline{F\,(x,t)} = \frac{\sum_{s=1}^{|S|} \overline{F_s(x,t)}}{|S|} \quad (11)$$

Where

$A_{criteria}$: denotes the average cost function criteria.

The overall objective of the optimization is to minimize the average cost function

4.2. Experimental Results

This subsection presents the results obtained for the proposed triggering technique and analyses them as compared to the benchmark work in the conducted experiments.

4.2.1. Results Presentation

The simulation carried out considered five IaaS data centers (DC1 to DC5) and used 80 distinct workload traces. Therefore, for the experiments carried out, 400 different scenarios were considered. The average values of the results of the simulation are shown in Figure 4 to Figure 8.

The numerical summary of the mean results of the simulation obtained from the five data centers (i.e. DC1 to DC5) is shown in Table 4. The values in the table represents the costs of the considered evaluation criteria defined in Eq. (11).

In order to dissect the results in Table 4, Figure 3 shows the values of the time-based average cost of DC1 to DC5, i.e. the Fs (x,t) in DC1 to DC5 per scenario s ∈ S. The values were obtained using Eq. (10).

Table 4. Mean values of the results obtained in the performed experiment.

Algorithms	DC1	DC2	DC3	DC4	DC5	Ranking
This work	0.711	0.780	0.820	0.798	0.841	1st
Benchmark (lopez-pires et al., 2017)	0.779	0.866	0.926	0.918	0.963	2nd

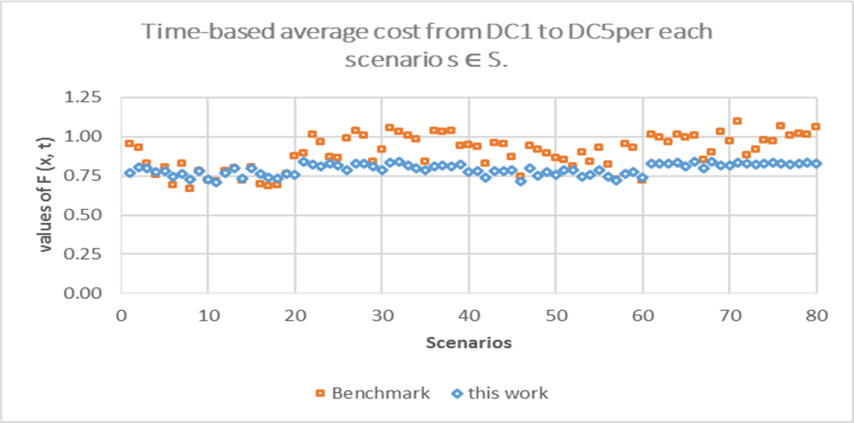

Figure 3. Figure 3. Average values of Fs (x, t) in DC1 to DC5 per each scenario s ∈ S.

Figures 4 to 8 depict the performance behavior of the compared works with respect to the individual data centers. It can be seen that the work based on the proposed method outperformed the benchmark work in all the experiments of the five data centers.

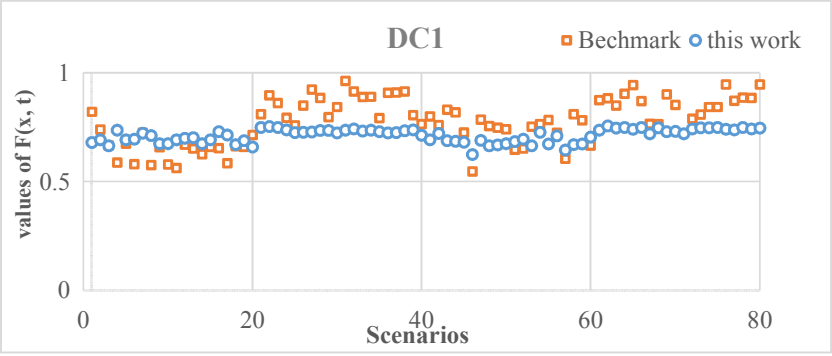

Figure 4. Values of Fs (x, t) in DC1 per each scenario s ∈ S.

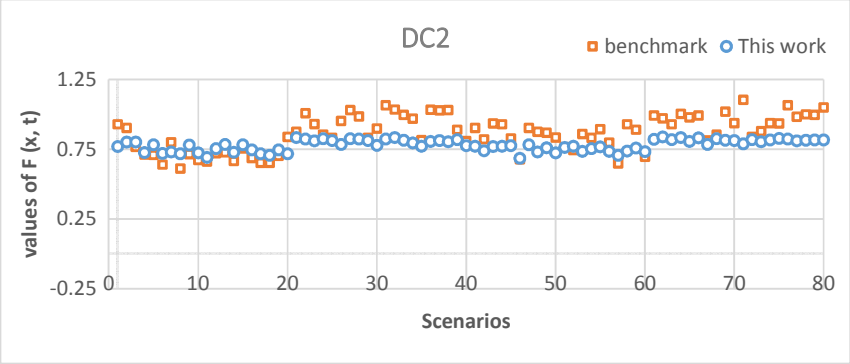

Figure 5. Values of Fs (x, t) in DC2 per each scenario s ∈ S.

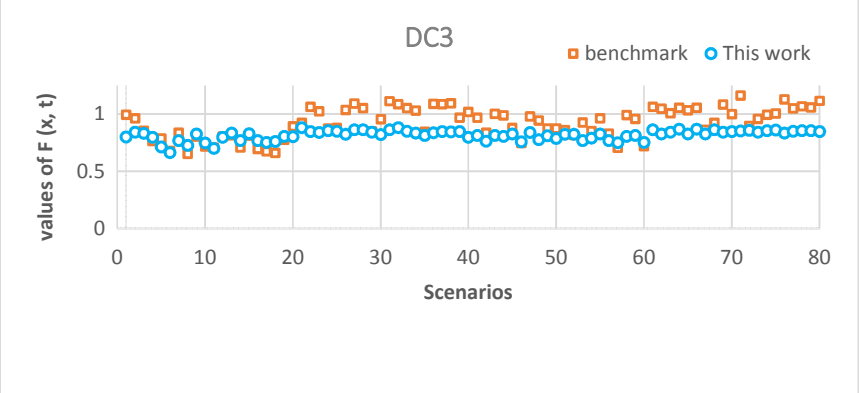

Figure 6. Values of Fs (x, t) in DC3 per each scenario s ∈ S.

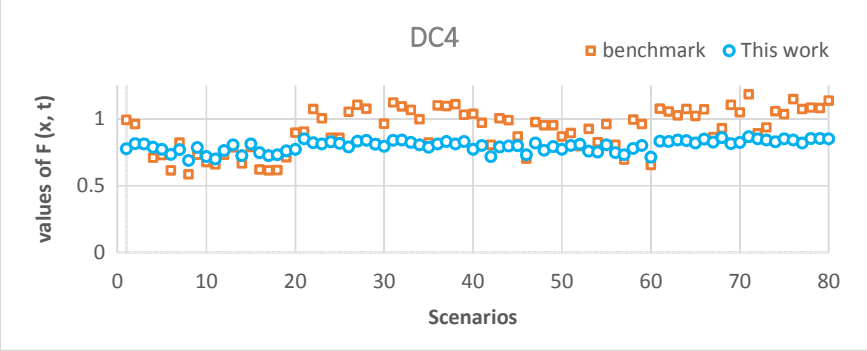

Figure 7. Values of Fs (x, t) in DC4 per each scenario s ∈ S.

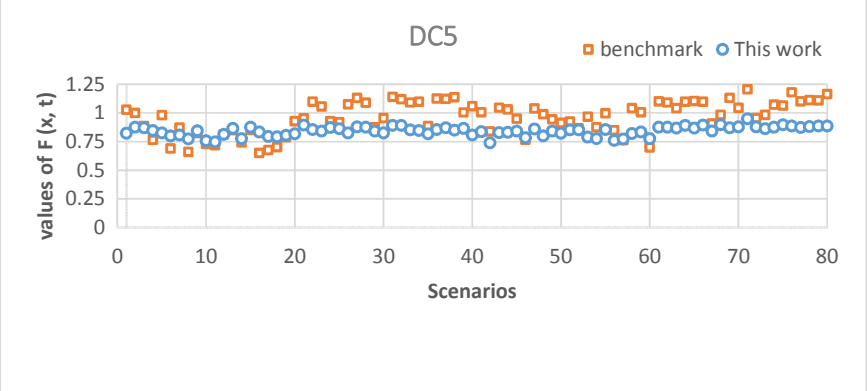

Figure 8. Values of Fs (x, t) in DC5 per each scenario s ∈ S.

4.2.2. Discussion of Results

This study was aimed at investigating the impact of VMPr triggering strategies in the two-phase optimization of virtual machine placement problems. The main evaluation metric used is the F (x, t) minimization cost function, which is a composite of power usage, economic revenue, resource usage and reconfiguration period functions.

Table 4 shows the summary of the evaluation results in terms of numerical average values of $A_{criteria}$ attained by each method at the end of each experiment. From the table, it can be observed that, with respect to performance of the benchmark work, as the number of PMs reduces from 50 to 10 across the DC1 to DC5, the aggregate cost increased significantly, from 0.779 to 0.963. On the other hand, the cost with respect to this work, increased slightly from 0.711 to 0.841 across the five experiments. It is also interesting to notice the effectiveness and robustness of our method in that the percentage improvement recorded, from the simple to a difficult case, increased significantly from more than 8.7%, in the simple case, to more than 12.6%, in the most difficult case.

The results in Figure 3 shows a bigger picture, scenario-by-scenario, of what the results in Table 4 entail. Under each scenario, the results of the five data centers were averaged. The figure demonstrates the behaviour and performance of the F_S (x, t) time-based average cost based on Damped Trend Exponential Smoothing (DTES) as compared to Double Exponential Smoothing (DES) in each experiment. From the figure, it can be observed that in the beginning of each experiment, when the workloads were between 0 and 20, the two schemes showed competitive performance behaviour. However, as the workload increased, the proposed work consistently outperformed the benchmark work. This performance behaviour demonstrated by the benchmark is as a result of the weakness of its prediction model, which tends to over-forecast, especially for long forecast periods. This behaviour did not manifest in the periods of small workloads, but as the load increased the limitation became apparent. In consequence, initiation of the optimization at the VMPr phase was affected and hence the unideal placement of VMs. On the other hand, the good performance of the proposed system was due to the impact of its prediction model, the DTES. The DTES model has better forecast accuracy compared to the DES and hence optimization at the VMPr phase was triggered efficiently, which resulted in better placement of VMs.

Figures 4 - 8 presents the dissection of the result shown in Figure 3, to further understand and analyse the values presented in Table 4. From the figures, it can be observed that although the benchmark work shows competitive behavior in the first few scenarios in each of the five datacenters, the proposed method eventually recorded better performance in all of them.

It is worth remembering that since the four objective functions (power usage, economic revenue, resource usage and reconfiguration period) were combined into a single composite objective function for simplifying the optimization task, achieving optimal values of the function is akin to attaining better trade-offs of the four individual objective functions. By implication, this means that the proposed method is able to optimize VMs placement so that better trade-offs, in terms of power usage, economic revenue, resource usage and reconfiguration time, are achieved compared to the benchmark.

Overall, the finding is that the technique proposed in this work to trigger the VMPr phase outperformed the existing benchmark technique in the experiments conducted by achieving a minimal cost function in the scenarios considered. Therefore, the VMPr triggering approach based on DTES is better than Double Exponential Smoothing (DES) for the two-phase optimization of VMP.

5. Conclusion

This work proposed an improved two-phase optimization scheme for the Virtual Machine Placement problem. The considered two-phase optimization scheme incorporated the features and advantages of both online (dynamic) and offline (static) VMP formulations, where a predictive VMPr triggering technique was proposed to decide when to cause a recalculation phase of the placement. Taking into account 400 different scenarios, an experimental evaluation was performed against the benchmark research and the results show that the proposed VMPr triggering technique outperforms the benchmark work by achieving a minimum cost function with more than 12% improvement. Therefore, the proposed method is a novel approach to trigger the VMPr phase to achieve better optimization results from the reconfiguration phase of the two-phase optimization scheme. For future work, it is recommended to seek for evolutionary approaches with better explorative and exploitative power than MA to improve performance in the VMPr phase. Also, while this research work considered only four objectives, several others should be considered as future work. Furthermore, experiments with Geo-distributed datacenters are still left as areas for potential research. Lastly, more extensive experimental analysis of different parameters could still be considered for the VMP problem.

References

Aloulou, M. A., & Croce, F. D. (2008). Complexity of single machine scheduling problems under scenario based uncertainty. In *Operations Research Letters* (pp. 338–342). Retrieved from https://dl.acm.org/doi/10.1016/j.orl.2007.11.005

Amarilla, A., Benıtez, L., Zalimben, S., Pires, F. L., & Baran, B. (2017). Evaluating a Two-Phase Virtual Machine Placement Optimization Scheme for Cloud Computing Datacenters. *MIC/MAEB 2017*, 4–7. Barcelona.

Angeles, S. (2014). Virtualization Vs Cloud Computing: What's the Difference? *BusinessNewsDaily*.

Azizi, S., Zandsalimi, M., & Li, D. (2020). An energy-efficient algorithm for virtual machine

placement optimization in cloud data centers. *Cluster Computing*. Retrieved from https://doi.org/10.1007/s10586-020-03096-0

Baloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy aware resource allocation heuristics for efficient management of datacenters for cloud Computing. *Future Gener. Comp. Syst.*, 755–768.

Biran, O., Corrad, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., & Silvera, E. (2012). A stable network aware VM placement for Cloud Systems. *In Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)*, 498–506.

Braiki, K., & Youssef, H. (2019). Fuzzy-logic-based multi-objective best-fit-decreasing virtual machine reallocation. *The Journal of Supercomputing*. Retrieved from https://doi.org/10.1007/s11227-019-03029-8

Calcavecchia, N. M., Biran, O., Hadad, E., Moatti, & Y. (2012). VM placement strategies for cloud scenarios in cloud computing (CLOUD). *IEEE 5th International Conference*, 852–859. IEEE.

Chamas, N., Lopez, F. P., & Baran, B. (2017). Two Phase Virtual Machine Placement Algorithms for Cloud Computing: An Experimental Evaluation under Uncertainty. *IEEE International Conference*.

Chang, Y., Gu, C., & Luo, F. (2016). A novel energy□aware and resource efficient virtual resource allocation strategy in IaaS cloud. *In Computer and Communications (ICCC), 2016 2nd IEEE International Conference*, 1283□ 1288.

Farahnakian, F., Bahsoon, R., Liljeberg, P., & Pahikkala, T. (2016). Self-adaptive Resource Management System in IaaS clouds. *IEEE 9th International Conference on Cloud Computing*, 553–560.

Goudarzi, H., & Pedram, M. (2012). Energy-efficiency virtual machine replication and placement in a cloud computing system. *In the Cloud Computing (CLOUDS) IEEE 5th International Conference*.

Gupta, A., Milojicic, D., & Kal'e, L. V. (2012). Optimizing VM placement for hpc in cloud. *In Proceedings of the 2012 Workshop on Cloud Services, Federation and 8th Open Cirrus Summit*, 1–6. ACM.

Haghighi, M. A., Maeen, M., & Haghparast, M. (2018). An Energy-Efficient Dynamic Resource Management Approach Based on Clustering and Meta-Heuristic Algorithms in Cloud Computing IaaS Platforms. *Wireless Personal Communications*.

Hong, H.-J., Chen, D.-Y., Huang, C.-Y., Chen, K.-T., & Hsu, C.-H. (2013). Qoe-aware virtual machine placement for cloud games . in Network and Systems Support for Games (NetGames). *2013 12th Annual Workshop on IEEE*, 1–2.

Hyndman, R., & Athanasopoulos, G. (2018). Forecasting: principles and practice. In *OTexts*. Retrieved from OTexts.com/fpp2

Khanchi, M., & Tyagi, S. (2016). VM Scheduling in Cloud Computing using Meta-heuristic Approaches. *International Journal of Scientific & Engineering Research*, *7*(12), 139–144.

Li, K., Zheng, H., Wu, J., & Du, X. (2015). Virtual machine placement in cloud systems through migration process. *International Journal of Parallel, Emergent and Distributed Systems*, 393–410.

Lopez-pires, F., Baran, B., Benitez, L., Zalinmben, S., & Amarilla, A. (2017). Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty. *Future Generation Computer Systems*.

López-Pires, F., Barán, B., Pereira, C., Velázquez, M., & González, O. (2018). Towards Elastic Virtual Machine Placement in Overbooked OpenStack Clouds under Uncertainty. *VI Jornadas de Cloud Computing & Big Data (JCC&BD 2018)*, 146–157.

López-Pires, F., Barán, B., Pereira, C., Velázquez, M., & González, O. (2019). Evaluation of two phase virtual machine placement algorithms for green cloud datacenters. *2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. Retrieved from 10.1109/FAS-W.2019.00028

Mark, C. T., Niyato, D., & Chen-Khong, T. (2011). Evolutionary Optimal virtual machine placement and Demand forecaster for Cloud Computing. *International Conference on Advanced Information Networking and Aplications*, 348–355.

Mell, P., & Grance, T. (2009). *The NIST defination of cloud computing. National Institute of Standards and technology.*

Monil, M. A., & Rahman, R. M. (2016). VM consolidation approach based on heuristics, fuzzy logic and migration control. *Journal of Cloud Computing: Advances, Systems and Applications*.

Paliwal, S. (2014). *performance challenges in Cloud computing*.

Pires, F. L., & Baran, B. (2014). *virtual machine placement literature review (data)*. Retrieved from http://arxiv.org/abs/1506.01509

Pires, F. L., & Baran, B. (2015). A Virtual Machine Placement Taxonomy. *International Sympodium on Cluster, Cloud and Grid Computing, 15th IEEE/ACM*, 159–168.

Prevost, J., Nagothu, K., Kelly, B., & Jamshidi, M. (2013). Optimal update frequency model for physical machine state change state and virtual machine placement in cloud. System of Systems Engineering (SoSE), , (pp.). *8th International Conference on IEEE*, 159–164.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I., & . . . al., J. e. (2009). The reservior model and architechture for open federated cloud computing. *IBM Journal of Research and Development*, *53*, 1–4.

Shi, L., Furlong, J., & Wang, R. (2013). Empirical evaluation of vector bin packing algorithms for energy efficient data centers. *Computers and Communications ISCC*, 000009–000015.

Speitkamp, B., & Bichler, M. (2010). A mathematical programming approach for server. *IEEE Trans. Serv. Comput*, 266–278.

Tchernykh, A., Schwiegelsohn, U., Alexandrov, V., & Talbi, E. (2015). towards understanding uncertainty in cloud computing resource provisioning. *Procedia Comp. Sci*, 1772–1781.

Tighe, M., & Bauer, M. (2014). Integrating cloud application autoscaling with dynamic VM allocation. *In 2014 IEEE Network Operations and Management Symposium, NOMS*, 1–9.

Venigella, S. (2010). *Cloud storage and online bin packing, UNLV. Papers/Capstones*. Retrieved from http://digitalscholarship.unlv.edu/thesesdissertations/894.

Vijaypal, R. S., Pateriya, S. R., & Rajveer, K. G. (2015). An Efficient Virtual Machine Scheduling Techniques in Cloud Computing Environment. *I.J. Mordern Education and Computer Science*, 39–46.

Wang, S., Liu, Z., Zheng, Z., Sun, Q., & Yang, F. (2013). particle swarm otimization for energy-aware virtual machine placement optimization in virtualized data centers. *In Parallel and Distributed Systems (ICPADS) 2013 International Conference on IEEE*, 102–109. IEEE.

Xu, J., & Fortes, J. A. (2010). Multi-objective virtual machine placement in virtualized data center environments. *IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CSPCom) in Green Computing and Comunications (GreenCom)*, 179–188.

Yahaya, R. H., & Ambursa, F. U. (2019). Enhanced Two-Phase Virtual Machine Placement Scheme for Cloud Computing Datacenters. *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, 1–5. https://doi.org/10.1109/ICECCO48375.2019.9043260.

Yue, W., & Chen, Q. (2014). Dynamic Placement of Virtual Machines with Both Deterministic and Stochastic Demands for green cloud computing. *Mathematical Problems in Engineering*, 1–11. Retrieved from http://dx.doi.org/10.1155/2014/613719

Zheng, Q., Li, R., Li, X., Shah, N., Zhang, J., Tian, F., … Li, J. (2016). virtual machine consolidated placement based on multi-objective biogeography-based optimization. *Uture Gener. Comput. Syst.*, 95–122.