

# Two-Phase Virtual Machine Placement in Cloud Computing Data Centers Using Extreme Learning Machine Prediction-Based Triggering Method

Nafiu Musa Muhaammad<sup>1</sup>, Faruk Umar Ambursa<sup>2</sup> and Nasir Faruk<sup>3</sup>

<sup>1</sup>Directorate of Information and Communication Technology, Sule Lamido University Kafin Hausa, Jigawa, Nigeria

<sup>2</sup>Department of Information Technology, Bayero University, Kano, Nigeria

<sup>3</sup>Department of Information Technology Sule Lamido University Kafin Hausa, Jigawa, Nigeria  
nmmuhammad@slu.edu.ng<sup>1</sup>, fuambursa.it@buk.edu.ng<sup>2</sup>, faruk.n@slu.edu.ng<sup>3</sup>

## Abstract

*Background: Two-phase Optimization of Virtual Machine Placement (VMP) Problem considers both the Online Incremental VMP (iVMP) phase in which the new arrival of dynamic requests of Virtual Machines VMs are attended to and the Offline VMP reconfiguration (VMPr) phase that performs placement recalculation. In the two-phase scheme, the first part of the two-phase approach is the iVMP, where virtual machines (VMs) can be built, changed, or destroyed at runtime. While the second phase focuses on raising the standard of solutions produced by the iVMP, several studies have been done in different literature to solve the VMP problem. However, the methods used tend to be over-forecast and have long runs of a linear trend. This affects the prediction and produces a less optimal solution. Objective: The following four objective functions are optimized using the proposed Extreme Learning Machine Prediction-Based Triggering Method for Virtual Machine Placement in Cloud Computing Datacenters in Two-Phases, which combines the advantages of both online (dynamic) and static (offline) VMP formulations: the length of the reconfiguration process, the amount of energy used, the way resources are used, and the financial expenses. This study suggests a brand-new strategy for deciding when to start the VMP reconfiguration phase. Results: The Method provides more accuracy to the predicted requests as well as reduces the total economic penalties for Service Level Agreement (SLA) violations. An experimental comparison with the existing approach is conducted utilizing 400 cases. Conclusion: The results demonstrated that, in comparison to the benchmark approach, the proposed work obtained a minimum cost function with a 10.5% improvement.*

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

## 1. Introduction

A platform for network access and sharing computer resources that may be instantly distributed with little maintenance work or service provider contact is known as cloud computing. (Paliwal). Users of cloud computing can access hosted services under a pay-as-you-go arrangement. The following advantages have been made available to corporations and end consumers, namely: (i) Self-service provisioning which makes it possible for end users to easily access computer resources. (ii) Pay per usage: The user only has to pay for the services they utilize. (iii) Elasticity: the capacity to expand or contract services as needed. (iv) Migration flexibility: Clients can relocate certain workloads to another cloud platform as required. (v) Workload resilience: To ensure durable storage and load reliability, cloud service providers typically use superfluous resources.

The fundamental tenet of cloud computing is virtualization. Virtualization is a technique that involves the abstraction of physical resources from users (Angeles, 2014). According to National Institute of Standards and Technology (NIST), "Virtualization software allows multiple operating systems and applications to run on a single server at the same time" (Mell & Grance, 2009). However, despite the enormous benefits of cloud computing, several issues must be fixed because they increase adoption costs and dangers. They consist of; Security: Security is one of the main issues with cloud computing. The customer is unable to learn where, how, or by whom their data is processed. Identification and forecasting of quantities and costs are challenging. (iii) Resources are insufficient: As more customers use cloud services, fewer resources are available to the clients. (iv) Performance: The volume of consumers is causing a

decline in cloud performance. (v) Scheduling: Virtual Machines (VMs) are designed to run as needed thanks to the adaptability of cloud computing.

Optimization and resource distribution, which involves deciding where VMs should be put in PMs, are the two most challenging scheduling problems in cloud architecture (Rochwerger, et al., 2009). A lot of material has been written about the Virtual Machine Placement (VMP) issue. Energy efficiency, network traffic, and other objective functions are some of the topics that are addressed in the research (Hong et al., 2013; Prevost et al., 2012; Biran, et al., 2012; Li et al., 2013). Because of the nature of the infrastructure requirement, the online formulation is typically explored for the VMP problem (Pires & Baran, 2014). For selecting the appropriate PMs to host VMs with a little bit of temporal management, online formulations are crucial. An offline version of the VMP problem is being researched to provide an additional realistic outcome (Pires, & Baran., 2015). Nevertheless, a two-phase optimization that combines offline and online formulations can also be used to formulate the VMP issue (Pires et al., 2017).

To optimize four objective functions, a two-phase optimization technique was presented by (Pires et al., 2017), which considers the IaaS environment to be the most difficult (Ortigoza et al., 2016). (Wang et al., 2013) (ii) Economic profit (Xu & Fortes, 2010) Use of resources (iii) (Li et al., 2013) Reconfiguration time (iv) (Chamas et al., 2017). Methods for determining when to begin a placement reconfiguration and what to do with requested VMs during the placement recalculation time are included in the proposed system. This is accomplished by shifting virtual machines (VMs) from one physical machine to another (PMs). These two phases make up the two-stage optimization method: In the incremental VMP (iVMP) phase, the lists of needed virtual machines are ordered by revenue using the First-Fit Decreasing algorithm, and the necessary virtual machines are then assigned to the first PM with the available resources. (ii) During the VMP reconfiguration (VMPr) phase, a crucial decision was made using the Memetic algorithm to rearrange the position of VMs.

Additionally, by examining several online algorithms, the overbooking protection factor, and objective function secularization techniques, (Amarilla et al., 2017) provided an investigational evaluation of a two-phase optimization technique for VMP scenarios. Results obtained suggested that the protection factor should be adjusted to 0.0 in low CPU load scenarios and 0.75 in high load scenarios while using an offline Memetic algorithm in the VMPr phase and the BF and BFD heuristics for the online iVMP phase. The research proposed an improved 2-phase optimization VMP technique for both online and offline formulations of the VMP problem to obtain improved solutions based on the complex cloud infrastructural elasticity (Breitgand & Epstein, 2011) and the overbooking of services resources (Tomas & Tordsson, 2014). This is done in consideration of the benefits of formulating the VMP challenge as a 2-phase optimization technique (Pires & Baran, 2014). The topic demands the employment of the most direct method, even though it has been handled by several authors (Bologlov et al., 2012; Calcavecchia et al., 2012; Chamas et al., 2017; Pires et al., 2017). Important considerations here include selecting the proper PM for each VM in the cloud computing architecture as well as the timing and circumstances under which the VMPr phase should be started. Energy efficiency, economic revenue, resource utilization, and reconfiguration time are a few of these factors to take into account. In other words, how and when the VMPr is activated determines how effective the whole optimization plan will be. As a result, several tactics have been employed. The literature has looked at several trigger mechanisms for the VMPr phase, taking into consideration both the offline VMPr phase and the live iVMP phase, where dynamic VM requests are handled. Even with the donation made, there are still numerous problems that have not been resolved, including in a two-phase optimization strategy, the VMPr Triggering approach establishes when the VMPr phase is activated (Amarilla et al., 2017; Pires, et al., 2017). The approach, though, reveals a steadfast trend.

A previous study by Yahaya et al. (Yahaya et al., 2020) presented a novel triggering methodology based on prediction utilizing a Triggering method called Damped Trend Exponential Smoothing (DTES) to anticipate when to activate the VMPr phase in the 2-phase optimization technique. Nevertheless, the DTES technique has lengthy runs of a linear trend with identical gradients, owing to this reason, only the weak linear trend will be considered (McKenzie & Gardner, 2010). The result of the reconfiguration step is thus not optimal. The outcome of the forecast is hence less precise. The performance of the algorithm reconfiguration phase, economic output, resource utilization, and cloud power consumption are all impacted as a result of the reconfiguration phase producing a less ideal solution. As a result, a novel

optimization technique for the VMP based on better prediction systems is required. In this study, a fresh prediction-based triggering method based on an ELM is proposed. The recommended method examines and more precisely selects when to activate the VMPr phase to recalculate and change the position. The new strategy was evaluated and compared to the previous strategy. The technique decreases total financial penalties for SLA violations and makes requests more predictable. 400 alternatives are considered in an experimental comparison to the earlier approach. The experimental findings show that the suggested technique when compared to the benchmark methodology, produced a minimal cost function of 10.5% with better overall outcomes for economic revenue, algorithm reconfiguration time, resource usage, and power consumption.

The remainder of this article is organized as follows. The related works are provided in section 2. Section 3 discussed the proposed technique in detail. Section 4 presents an example of a performance evaluation of the proposed solution. Section 5 summarizes the findings of the study.

## 2. Related Works

This section will address an earlier study on the VMP problem, which is where the problems for this research were found.

Fang, et al., (2013) developed a VMP and Migration technique employing a fat tree topology. For IVM placement (iVMP), they use the heuristic algorithms First Fit (FF), Best Fit (BF), and Worst Fit (WF). For the incremental phase, they use Random Placement Mechanism and Power-efficient Placement Mechanism, where Random Placement Mechanism looks for a functional PM that is about to be used before taking a new PM. The algorithms used by the power-efficient placement mechanism are FF, BF, and WF. In FF, the regulator identifies the differences between them and allocates the VM to the PM with the smallest weight in BF, while in BF and WF, the regulator finds the differences between them and allocates the VM to the PM with the biggest weight in WF. (ii) The VM migration will halt once you become a flow star. Although in FF the controller looks over every VM and migrates the one with sufficient resources, BF and WF look for weight disparities and then move the VM to the PM.

Moreover, (Calcavecchia et al., 2012) created a new Backward Speculative Placement (BSP) method that considers the type of VM requests. It has two algorithms (i) that employ the Best Fit Decreasing technique to manage the flow of requests (iVMP). (ii) Periodic optimization that moves the busiest hosts. (VMPr). When there is a workload, the optimization starts at a specific period and stops if a new request comes in.

A VM dynamic consolidation approach for the distribution of resources with consideration for energy (Baloglazov et al., 2012). The VM assignment process has two steps: Virtual machine accepts requests and assign hosts to receive them. The virtual machines (VMs) were distributed to a host with lower power consumption after being sorted in order of greatest to lowest CPU utilization. This was accomplished by allocating (iVMP) and (ii) Optimization of the existing VM Allocation (VMPr). Two sections make up this optimization: the virtual machines (VMs) that required moving, used three double-threshold VM assignment methods to set the higher and lower thresholds for the hosts, and recorded the total CPU consumption from all of the VMs allocated to the host.

Pires, et al., (2017) also suggested a two-phase optimization approach. (i) Incremental VMP phase: They use the First-Fit Decreasing heuristic to quickly allocate the requested VM to the first PM with enough resources after arranging the necessary VMs while considering revenue. (ii) The VMP reconfiguration phase: At this step, the placement of the VMs was fundamentally reconfigured using the Memetic algorithm. The VMPr phase consists of the following elements: use a VMPr Triggering technique to choose the appropriate time to launch the VMPr phase. The VMPr phase provided the following to distinguish the prerequisites for beginning the phase: (i) have a method for VMPr Triggering that is prediction-based. (ii) VMPr Recovery method outlined what to do in the event of a new request while the recalculation time was being calculated. The VMPr Recovery approach proposed update-based progress for recovery by updating the calculations from the VMPr phase. VMPr Triggering and VMPr Recovery consume resources while using less energy as a consequence.

Similarly, (Lopez-Pires et al., 2019) investigated 36 potential optimization algorithms for 2-phase virtual machine placement algorithms in green cloud data centers to determine the best technique for decreasing power usage. 400 situations and four dynamic factors were considered. The two-phase method with prediction-based triggering and update-based recovery procedures is most effective in lowering power consumption, according to experimental findings.

According to a recent suggestion, the VMPr phase employs a Memetic algorithm to make a unified decision to reposition VMs globally (Yahaya et al., 2020). In this method, the VMP phase is further divided into two strategies: The VMPr recovery strategy specifies how to handle new incoming VM requests while the VMPr is being recalculated. The iVMP triggering mechanism indicates when the VMPr phase should be initiated. The suggested VMPr triggering model estimates values of the global objective functions  $F(x, t)$  using a statistical method known as damped trend exponential smoothing (DTES) to decide when to activate the VMPr phase in the two-phase optimization scheme. When activated, the VMPr recalculates the placement of VMs at a discrete-time  $t$  over discrete time slots (i.e., recalculation time). Because the VMPr phase is static, the position may not be correct after recalculation. Before performing a full reconfiguration, the recalculated position must be precisely obtained using a VMPr recovery procedure since, while the VMPr is computing, the iVMP may still accept and handle incoming requests, making the VMPr computed solution irrelevant.

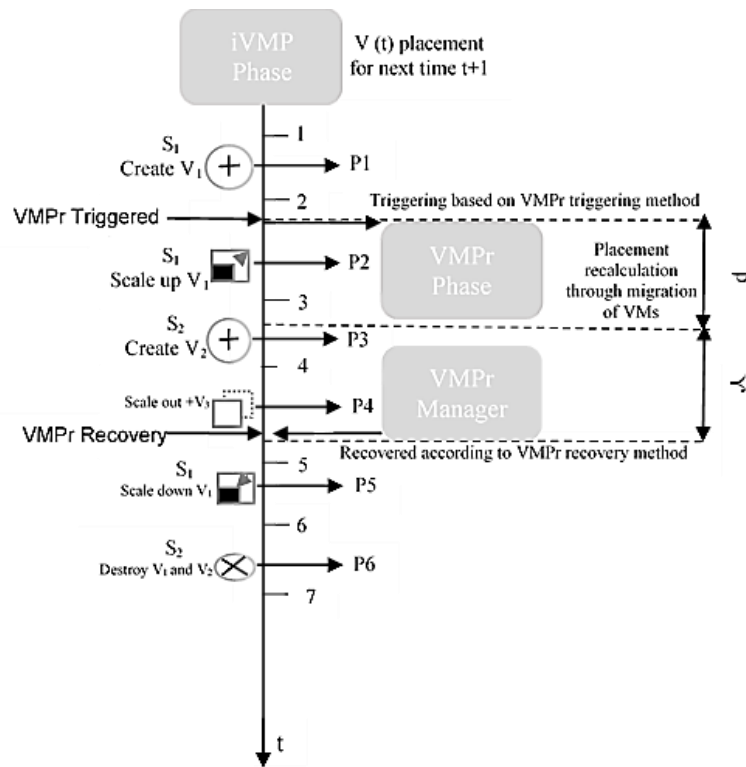


Figure 1. Two-Phase Allocation Scheme (Yahaya & Ambursa, 2019)

### 3. Proposed Two-Phase Optimization Scheme VM Placement

The first step is to research the available literature. The current study towards the online (dynamic) formulation of the VMP problem involves selecting the best PMs to host VMs in real-time. (Pires & Baran, 2014), and offline (static) formulations that concentrate on a static environment where there is no altering when choosing appropriate PMs to host VMs. (Pires & Baran, 2015). The goal of this effort is to create a reliable technique for starting the VMPr phase. The two phases' algorithms were taken from earlier works. As a result, before introducing our suggested approach for starting the VMPr phase from the iVMP phase, we briefly discuss the iVMP and VMPr phases and their specifics.

The solution obtained offline must be in a static environment (when the system is at rest), which is not applicable when solving a dynamic IaaS environment. For these reasons, VMP problems can use two-phase formulations that combine both online and offline approaches. Additionally, there is no guarantee that online placement will result in an optimal solution. (Pires et al., 2017).

The two-phase formulation has an iVMP phase (Dynamic) that stores VM placement requests and a VMPr phase (Static) that performs recalculation and replacement. The research question and the parameters of the study were established at this point. Details of the traits mentioned are provided in Section 1. The research project considers structuring the VMP problem as a two-phase optimization to optimize the four main objective functions as follows: (i) power consumption, (ii) economic revenue, (iii) resource usage, and (iv) reconfiguration time. This is similar to the work suggested by Yahaya et al., (2020).

### 3.1 Incremental VMP (iVMP) Phase

The iVMP discussed in previous work accepts a dynamic request for placement at runtime (pires, Baran, Benitez, Zalinmben, & Amarilla, 2017). They use FFD heuristic to address the VMP problem at every discrete time  $t$  where VM requests change from time to time. According to the FFD heuristic, to destroy cloud services, more virtual machines are created and their resources are increased first (steps 1–3 of Algorithm 1). VMs resources and number are handled earlier (steps 5–8), and elastic cloud service is more essential than non-elastic at step 4, where requests from  $V(t)$  are arranged with the criterion of revenue  $(R_j(t))$  from highest to lowest. Next, requests that are merely cloud service creations and are not handled by  $V_j(t)$  are ranked from highest to lowest (steps 7–18). The first  $H_i$  with appropriate resources gets the assignment from a  $V_j$ . If none of the  $H_i$  have enough resources to host  $V_j$ , it will choose a different federated provider. The location  $x(t+1)$  is then changed and returned (steps 19–20).

### 3.2 VMP Reconfiguration (VMPr) Phase

Researchers, like Yahaya et al., (2020), developed an offline Memetic algorithm (MA), the method is centralized when taking a choice. The placement recalculation was done here based on the static environment. This is how it functions:

Because it will be used in the following stage, step 1 will involve creating the population. The method uses local search to try to enhance the third-step result. The best option will be chosen after comparison. Initialization occurs at step 5, evolution begins (steps 6–12), and then crossover and mutation happen. (Step 8), and the hidden remedy is fixed in (step 9). Using local search will then enhance the solution at step 10. The procedure will continue till the completion condition is satisfied using the best practicable at step 11, the counter will update at step 12, and so on.

It must be stressed that this work's primary addition to the two-phase optimization strategy is the triggering mechanism of VMPr, which decides when to start the VMPr phase. This is covered in the subsection that follows.

### 3.3 Proposed VMPr Triggering Method

The specifics of the prior VMP problem solution scheme, how it was split into two phases, and the technique used to solve this problem is discussed in detail. It also details how the proposed algorithm for the VMP problem works. The cutting-edge two-optimization with its two stages, iVMP and VMPr.

#### 3.3.1. Previous VMPr Triggering Method

The prior VMPr triggering solution is described in this subsection, along with a discussion of its disadvantages.

The earlier method makes use of the Damped Trend Exponential Smoothing (DTES) technique (Venigella, 2010). After assessing the optimized objective function  $F(x, t)$ , the technique decides when to initiate the VMPr phase for recalculation and position reconfiguration

In a 2-phase optimization approach for VMP issues, the VMPr triggering mechanism determines when to start the VMPr phase. Nevertheless, the Damped Trend Exponential Smoothing (DTES) VMPr triggering strategy was put out in earlier research articles. Moreover, the long-term tendency in the projections generated by statistical approaches

suggests that they over predict (Hyndman & Athanasopoulos, 2018). As a result, the prediction's outcomes are less accurate.

DTES has smoothing parameters  $\alpha$  and  $\beta$ , and both values are in the range of 0 and 1 just like Holt's method with another parameter for damping  $\phi$ . Therefore, it is in the equation below. (Eq. 1 to 3).

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

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

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

where:

$E_t$ : represent the expected value of  $f(x, t)$  at time  $t$

$T_t$ : represent the trend of  $f(x, t)$  at time  $t$

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

$\alpha$ : smoothing parameter for level,  $0 < \alpha < 1$

$\beta$ : smoothing parameter for trend,  $0 < \beta < 1$

$\phi$ : denotes the damping parameter,  $0 < \phi < 1$

The previous triggering method first determines whether there are values of  $x(t)$  (placement currently received from iVMP phase); if so, the values are then obtained, used to compute the expected value ( $E_t$ ) and trend value ( $T_t$ ) of the global objective function  $F(x, t)$  using damped trend exponential smoothing (DTES), and finally used to compute the forecast of  $N$  values  $f(x, t + 1)$ . A reconfiguration is initiated and the VMPr phase is returned if the predicted values of  $F_1(x) < F_2(x) < \dots < F_n(x)$  are true; otherwise, the algorithm quits and the placement in the iVMP phase continues.

### Listening 1: Damped Trend Triggering Algorithm

INPUT: values of  $x(t)$

OUTPUT: reconfiguration phase (VMPr phase)

Begin:

Step 1: Check values of  $x(t)$

Step 2: **If**  $x(t)$  exists then

Step 3: Get  $n$  values of  $x(t)$

Step 4: Compute  $E_t$  using DTES technique

Step 5: Estimate  $F_{t+h/t}$  for  $n$  values of  $F(x, t + 1)$

Step 6: **If** values of  $F_1(x) < F_2(x) \dots < F_n(x)$

Step 7: Activate reconfiguration then

Step 8: Return VMPr phase

Step 9: **End**

Step 10: **Else**, return the iVMP phase

Step 11: **End**

Step 12: **End**

However, it was noted that over the long term, the statistical method's forecasts had a consistent pattern, and it was found that the method looked to over-predict (Hyndman & Athanasopoulos, 2018). As a result, the prediction's outcomes are less accurate.

An innovative prediction-based triggering strategy based on Extreme Learning Machine (ELM) techniques is suggested as a solution to the issue. (See Equations (4) and (5)). This method has been demonstrated to be extremely effective for many predictions (Huang et al., 2010).

The ELM is a learning technique that uses feed-forward in artificial neural networks and is used for classification and regression. It was first proposed by (Huang et al., 2006). The algorithm is simpler than other neural network techniques because it just has one hidden layer. In general, neural network algorithms outperform most other methods, including the DTES utilized in the benchmark work, in terms of prediction. The suggested approach determines when to initiate the VMPr phase by evaluating the optimized objective function  $f(x, t)$ .

How it works: the ELM starts with training data, inputs are multiplied by a weight, added with bias, and then the activation function is applied. The phases will be repeated a certain number of times to obtain the outcome.

The model of ELM is given in equation (4)

$$FL(x) = \sum_{i=1}^L \beta_i g_i(x) = \sum_{i=1}^L \beta_i g_i(w_i * x_j + b_i), j = 1, \dots, N \quad (4)$$

$$(x_i, t_i) \in R_n \times R_m (i = 1, 2, \dots, N) \quad (5)$$

where:

- $L$ : represent the number of hidden units
- $N$  represent the number of training samples
- $\beta$ : represent the weight vector between the hidden layer and output
- $w$ : represent a weight vector between the input and hidden layer
- $g$ : represent an activation function
- $b$ : represent a bias vector
- $x$ : represent an input vector
- $m$ : represent the number of outputs
- $H$ : represent hidden layer output matrix
- $T$ : represent a training data target matrix

(1) In accordance with  $\beta$ , the method can achieve the lowest training error; (2) it can achieve the best generalization performance of the output connection weights and network; and (3) it is special. The local optimum solution may not be produced as a result. In conclusion, given a hidden node number  $N$ , an activation function  $f(x)$ , and a training set  $(x_i, t_i) \in R_n \times R_m (i = 1, 2, \dots, N)$ .

The three stages that make up the ELM algorithm are as follows:

In step 1, input weights  $a_i$  and hidden layer biases  $b_i$  is assigned at random (where  $i = 1, 2, \dots, N$ ).

Step 2: Determine the  $H$  matrix for the hidden layer output.

Step 3: Determine the output weight based on the aforementioned equation

The Extreme Learning Machine (ELM) is used to compute the expected values  $E_t$  of the  $f(x, t)$  global objective function before computing the forecast of  $N$  values  $f(x, t + 1)$ . This is the proposed triggering technique in Algorithm 2. It first checks to see if there are values of  $x(t)$  (placement currently received from iVMP phase); if the value of  $x(t)$  exists; otherwise, it gets the values. The method ultimately determines if the forecast values of  $F_1(x), F_2(x), \dots, F_n(x)$  are accurate; if so, reconfiguration is triggered and the VMPr phase is returned; otherwise, the algorithm leaves and the placement in the iVMP phase continues.

## Listening 2: Proposed Extreme Learning Machine Triggering Algorithm

INPUT: values of  $x(t)$

OUTPUT: reconfiguration phase (VMPr phase)

Begin:

Step 1: Check values of  $x(t)$

Step 2: **If**  $x(t)$  exists then

Step 3: Get  $n$  values of  $x(t)$

Step 4: Compute  $E_t$  using ELM technique

- Step 5: Estimate  $F_{t+h/t}$  for n values of  $f(x, t + 1)$
- Step 6: **If** values of  $F_1(x) < F_2(x) \dots < F_n(x)$
- Step 7: Activate reconfiguration then
- Step 8: Return VMPr phase
- Step 9: **End**
- Step 10: **Else**, return the iVMP phase
- Step 11: **End**
- Step 12: **End**

#### 4. Performance Evaluation

This section provides the experimental setting for this study, including computing resources. These tools are used to assess the success of the research effort. Windows 10 Pro, an Intel(R) Core TM i5-4300U processor operating at 2.50 GHz, and 8 GB of Memory were used to complete the experiments. Python programming language was used to implement the algorithms that were evaluated.

##### 4.1 Experimental Settings

The experimental setup's specifications were modified from those in a benchmark publication (Yahaya et al., 2020). For the proposed virtual machines, the Elastic Compute Cloud (EC2) that will be utilized was adopted from Amazon EC2 described in (Yahaya et al., 2019) It's critical to remember that the following hazy criteria would be considered: (I) the horizontal elasticity of the number of VMs that make up cloud services, (II) the vertical elasticity of virtual resource capabilities, (III) the usage of CPU and RAM virtual resources, (IV) the utilization of networking virtual resources (both relevant for overbooking). Thus, it would be considered that each parameter's behavior is represented by two different Probability Distribution Functions (PDFs) (i.e., Uniform and Poisson). To build workload traces of cloud service requests for provider-focused VMP problems, a Cloud Workload Trace Generator (CWTG) was employed (Ortigoza et al., 2016).

Table 1: PM types considered for simulations

PM type	Pr1, i [ECU]	Pr2, i [GB]	Pr3, i [Mbps]	P maxi	ci
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	50	10	8	10	8



Table 3: VM types from Amazon EC2 are 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	12	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

## 4.2 Objective Functions

The four primary objective functions provided by Pires and Baran (2015) in their study, A Virtual Machine Placement Taxonomy, will be taken into consideration in this work: power consumption, economic revenue, resource utilization, and reconfiguration time. The four-goal functions were integrated into a single optimization context while accounting for the cost of each objective function. This section outlines the processes that were followed to arrive at the goal function. These acts are modified versions of prior work.

### 4.2.1 Power Consumption

The total power consumption of each Physical Machine (PM)  $H_i$  (i.e., resources not given to any Virtual Machine (VM)  $V_j$  that makes up the complex IaaS environment, as indicated in equation (6), may be used to illustrate the power consumption minimization proposed by (Baloglazov et al., 2012).

$$f_1(x,t) = \sum_{i=0}^n \left( (pmax_i - pmin_i) \times Ur_{1,i}(t) + pmin_i \right) \times Y_i(t) \tag{6}$$

where;

$x$ : Evaluated solution of the problem;

$f_1(x,t)$ : Total power consumption of PMs at instant  $t$ ;

$pmax_i$ : Maximum power consumption of a PM  $H_i$ .

$pmin_i$ : Minimum power consumption of a PM  $H_i$ .

$Ur_{1,i}(t)$ : Utilization ratio of resource 1 (in this case CPU) by  $H_i$  at instant  $t$ ;

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

### 4.2.2 Economical revenue

By reducing the total expenses of renting resources from other cloud federation data centers as well as the total economical fines for SLA breaches, as shown in equation 7, the goal of this effort is to maximize the entire economic income that an IaaS provider obtains.

$$f_2(x,t) = LC(t) + EP(t) \tag{7}$$

where:

$f_2(x,t)$ : Total economical expenditure of the main IaaS provider at instant  $t$

#### 4.2.3 Resource Utilization

By reducing the average ratio of resources spent on each Physical Machine (PM)  $H_i$  (i.e., resources not given to any Virtual Machine (VM)  $V_j$ ), as stated in equation 8, the process of maximizing resource usage was carried out.

$$f_3(x, t) = \frac{\sum_{i=1}^n \left[ 1 - \left( \frac{\sum_{k=1}^r Ur_{k,i}(t)}{r} \right) \right] \times Yi(t)}{\sum_{i=1}^n Yi(t)} \tag{8}$$

where:

$f_3(x, t)$ : Average ratio of wasted resources at instant  $t$ ;

$Ur_{k,i}(t)$ : Utilization ratio of resource  $k$  of PM  $H_i$  at instant  $t$ ;

$r$ : Number of considered resources, in this work  $r=3$ : CPU, RAM, and network capacity.

#### 4.2.4 Reconfiguration Time

By reducing the maximum amount of RAM that may be transferred from one PM  $H_i$  to another  $H_j$  as specified in equation 9, the maximum reconfiguration time can be reduced.

$$f_4(x, t) = \max(MT_{i,i}) \quad \forall i, i \in \{1, \dots, n\} \tag{9}$$

where:

$f_4(x, t)$ : Network traffic overhead for VM migrations at instant  $t$ ;

$MT_{i,i}$ : Total amount of RAM to be migrated from PM  $H_i$  to  $H_i$ .

#### 4.2.5 Aggregate Cost Function (ACF)

Every objective function that is taken into consideration must be expressed in a separate optimization context, and the costs of each objective function must be normalized to be comparable and integrated as one goal. This work normalizes each objective function cost by computing  $F_1(x, t)R$ , i.e.,  $F_1(x, t)$  for each original objective function  $f_i(x, t)$ , to merge the metric functions according to equation (10)

$$\hat{f}_i(x, t) = \frac{f_i(x, t) - f_{i(x,t)min}}{f_{i(x,t)max} - f_{i(x,t)min}} \tag{10}$$

Where:

$\hat{f}_i(x, t)$ : Normalized cost of the objective function  $f_i(x, t)$  at instant  $t$ .

$f_i(x, t)$ : Cost of the original objective function  $f_i(x, t)$

$f_{i(x,t)min}$ : Minimum possible cost for  $f_i(x, t)$

$f_{i(x,t)max}$ : Maximum possible cost for  $f_i(x, t)$ .

A single objective is created by combining the supplied normalized objective functions with a minimal Euclidean distance to the origin as specified in equation (11)

$$F(x, t) = \sqrt{\sum_{i=1}^q \hat{f}_i(x, t)^2} \tag{11}$$

Where:

$F(x, t)$  denotes a single objective function that combines all of the  $\hat{f}_i(x, t)$  values at instant  $t$ ;  $\hat{f}_i(x, t)$  represents the normalized cost of the  $f_i(x, t)$  values at instant  $t$ ; and  $q$  denotes the number of  $f_i(x, t)$  values (in this example,  $q=4$ ).

Also, much as in earlier studies, the number of virtual machines (VMs) that make up cloud services as well as the uncertainty of the evaluation parameters are taken into account in this study. Due to the erratic nature of client demands and the overall maturity of dynamic cloud settings, (iii) CPU and RAM virtual resources consumption as well as (iv) utilizing virtual networking resources.

Typically, uncertainty is modeled using a collection of scenarios  $S$  that take into account the unknown parameters. Equation 12 defines the temporal average value of the objective function  $F(x, t)$  for each scenario  $s \in S$ .

$$\overline{f_s(x, t)} = \frac{\sum_{t=1}^{t_{max}} F(x, t)}{t_{max}} \tag{12}$$

Where:

$\overline{f_s(x, t)}$  : Temporal average of the combined objective function for all discrete time instant  $t$  in scenario  $s \in S$ ;  
 $t_{max}$  : Duration of a scenario in discrete time instants.

The average criteria (Aloulou & Croce, 2008), which is defined as the average  $F_s(x, t)$  for all situations  $s \in S$  as given in equation 13, is used in this study to choose a solution from a group of candidate solutions.

$$A_{criteria} = \overline{F(x, t)} = \frac{\sum_{s=1}^{|S|} \overline{F_s(x, t)}}{|S|} \tag{13}$$

Where:

$A_{criteria}$ : denotes the average cost function criteria.  
 The optimization's overarching goal is to reduce the average cost function.

### 4.3 Experimental Results

This section offers the experimental findings and a discussion of the experiment results with a comparison to the benchmark study for the suggested triggering technique.

#### 4.3.1 Results Presentation

The outcomes of the simulation experiment are presented in this subsection. The simulation took into account 80 distinct workloads in addition to five IaaS data centers (DC1, DC2, DC3, DC4 & DC5). 400 different possibilities were therefore taken into account. Figures 4 to 8 for the executed experiments indicate the average values of the simulation's outcomes.

Table 4 provides a numerical breakdown of the simulation's average findings from the five data centers (DC1 through DC5). The figures in the table show how much each of the assessment factors that were taken into consideration as defined in equation (13).

The picture displays the values of the time-based average cost of DC1 to DC5, or the  $F_s(x, t)$  in DC1 to DC5 for each scenario  $s \in S$ , to analyze the findings in Table 4. Equation 12 was used to acquire the values.

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

Algorithms	DC 1	DC 2	DC 3	DC 4	DC 5	Ranking
<b>This Work</b>	0.677	0.690	0.682	0.684	0.692	1 <sup>st</sup>
<b>Bench work (Rahina et al, 2020)</b>	0.711	0.780	0.820	0.798	0.841	2 <sup>nd</sup>

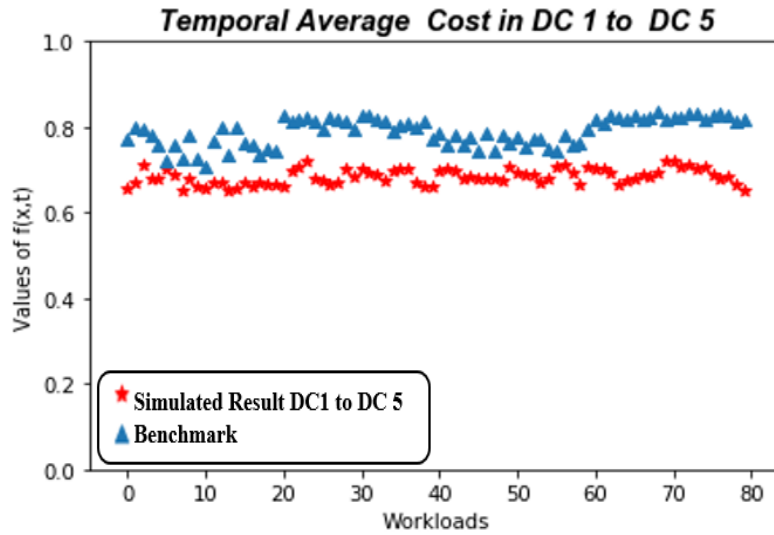


Figure 2. Average values of  $F_s(x, t)$  in DC 1 to DC 5 per each Scenario  $s \in S$

The performance behavior of the comparative works with regard to the various data centers is shown in Figures 3 to 7. It is clear that in each of the five data centers' trials, the work based on the suggested strategy performed better than the benchmark work.

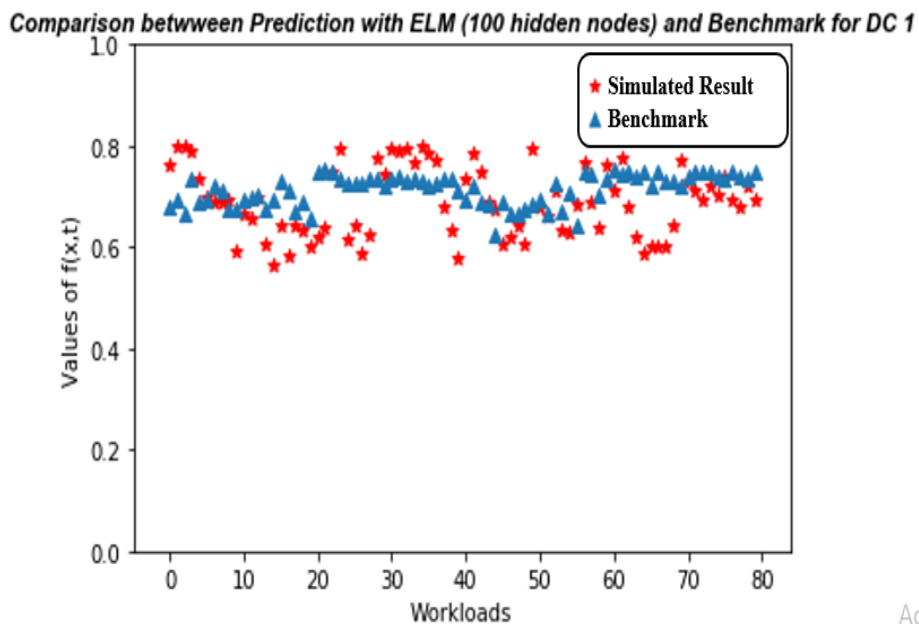


Figure 3. Average values of  $F_s(x, t)$  in DC 1 per each Scenario  $s \in S$

Comparison between Prediction with ELM (100 hidden nodes) and Benchmark for DC 2

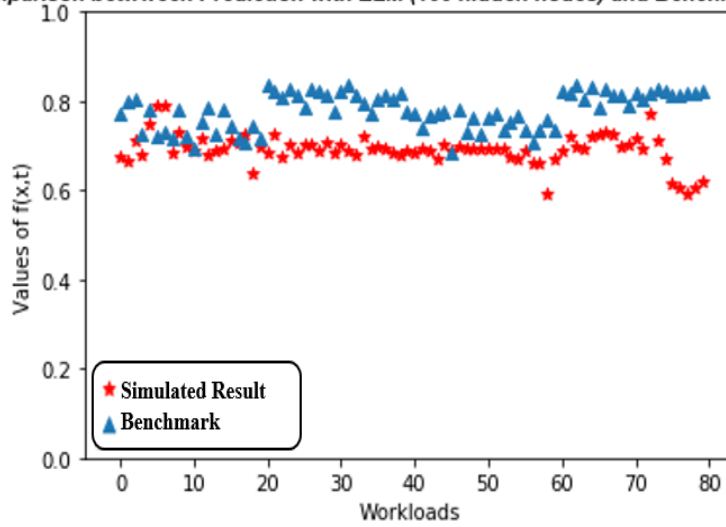


Figure 4. Average values of  $F_s(x, t)$  in DC 2 per each Scenario  $s \in S$

Comparison between Prediction with ELM (100 hidden nodes) and Benchmark for DC 3

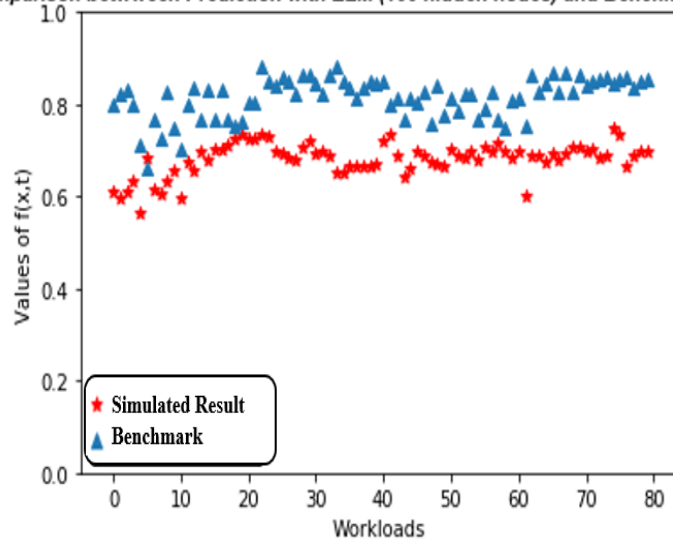


Figure 5. Average values of  $F_s(x, t)$  in DC 3 per each Scenario  $s \in S$

Act  
Got

Comparison between Prediction with ELM (100 hidden nodes) and Benchmark for DC4

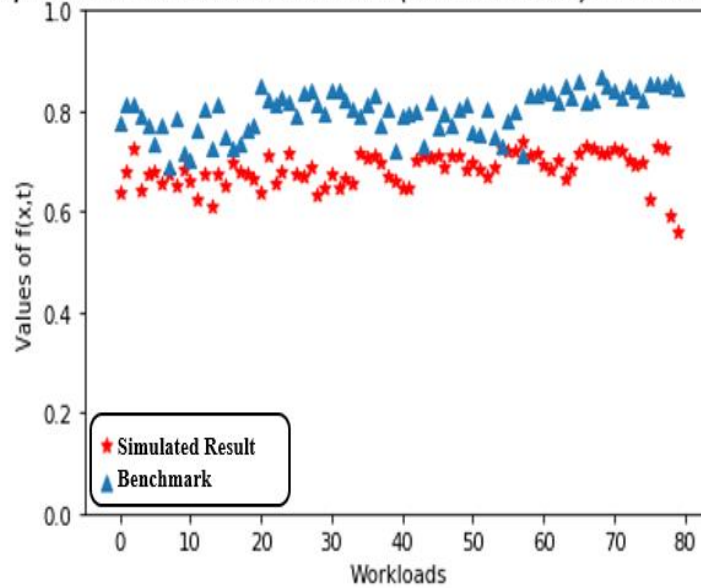


Figure 6. Average values of  $F_s(x, t)$  in DC 4 per each Scenario  $s \in S$

Comparison between Prediction with ELM (100 hidden nodes) and Benchmark for DC5

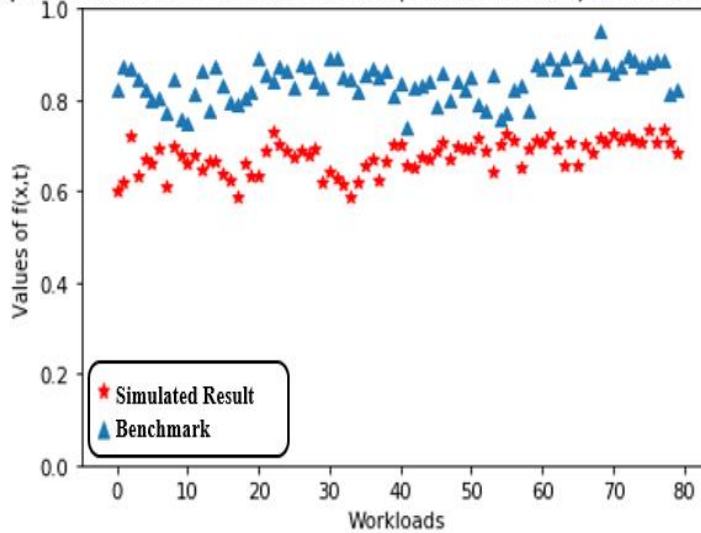


Figure 7. Average values of  $F_s(x, t)$  in DC 5 per each Scenario  $s \in S$

4.3.2 Result Discussion

This study investigated the impact of VMPr triggering approaches on the 2-phase virtual machine placement difficulties. The combined power usage, economic revenue, resource usage, and reconfiguration period functions make up the mean evaluation metric used,  $F(x, t)$  minimization cost function.

The average numerical values of the  $A_{criteria}$  attained by each strategy at the end of each trial serve as a summary of the assessment outcomes in Table 4. The table's analysis of the benchmark work's performance from DC1 to DC5 shows that while the number of PMs dropped from 50 to 10, the aggregate cost climbed dramatically, from 0.711 to 0.841. On the other side, the total cost of this activity across the five data centers climbed somewhat, from 0.67 to 0.69.

The results of Figure 2 clearly show the consequences of Table 4 scenario by scenario. For each scenario, the results from the five data centers were averaged. Figure 2 also illustrates the behavior and performance of the  $F_s(x, t)$  Temporal Average Cost on the Extreme Learning Machine (ELM) and Damped Trend Exponential Smoothing (DTES) in each experiment. As shown in Figure 2, the suggested work outperformed the benchmark work as the workload grew, although both techniques performed equally when the workload was between 0 and 30. This performance behavior of the benchmark work is a result of the prediction model's flaw, which is that over the long term, statistical forecasts exhibit a consistent tendency, and it has been found that these approaches overpredict. During times of light workloads, this behavior did not show, but as the load mounted, the constraint became clear. As a result, the optimization's start at the VMPr phase was impacted, which led to the placement of VMs outside of optimum conditions. On the other hand, the proposed system's successful performance was a result of the influence of its ELM prediction model. Since the ELM model has a higher prediction accuracy than the DTES, optimization during the VMPr phase was effectively triggered, leading to better placement VMs.

Figures 3 to 7 give the breakdown of the result presented in Figure 2 to more clearly understand and study the statistics displayed in Table 4. Even though the benchmark work initially displays competitive behavior in each of the five data centers, it is clear from the numbers that the suggested technique ultimately recorded greater performance in each of the five data centers.

Results from this study are presented in Table 4 above, which shows that the suggested Extreme Learning Machine model for Triggering the VMPr phase outperforms the existing benchmark technique in the experiment and that the results attain a minimal cost function in the circumstances considered. This study result is consequently superior to the benchmark research findings, as determined by the Extreme Learning Machine Approach.

The study effort also yielded improved results (i.e., Minimum Cost) for the average values  $F_s(x, t)$  in the overall scenario considered for the five data centers, taking into consideration the results displayed in Figure 2. (DC 1 to DC 5). In conclusion, our work achieves a better result (Minimum cost) with an average value of 0.685, or 68.5%, compared to the benchmark task, which receives an average value of 0.79, or 79%.

The four different objective functions of power use, economic revenue, resource usage, and reconfiguration period were integrated into a single composite objective function to simplify the optimization process and gain better trade-offs between the four separate objective functions. In terms of power consumption, economic income, resource utilization, and reconfiguration time, this implies that the proposed technique may optimize VM location to achieve better trade-offs than the benchmark.

This work outperforms the benchmark work using the minimization approach by 10.5% due to the suggested Extreme Learning Machine Techniques, which provide a better and more accurate Prediction Outcome and lead to a more effective solution during the reconfiguration phase of the two-phase optimization scheme.

A unique technique to begin the VMPr phase of the 2-phase optimization scheme for the uncertain VMP Problem is using the suggested prediction-based VMPr triggering approach on extreme learning machines. It is founded on an analysis of the tests that were carried out.

Overall, the findings demonstrate that, in the instances considered, the strategy proposed in this study to initiate the VMPr phase beats the benchmark method by achieving a minimum cost function. The ELM-based VMPr triggering technique, therefore, outperforms Damped Trend Exponential Smoothing for the two-phase optimization of VMP (DTES).

## 5. Conclusion

This study proposed a novel two-phase triggering mechanism for the Virtual Machine Placement problem, comparing an offline (VMPr) phase that is a static environment for placement recalculation and reconfiguration with an online (iVMP) phase that takes dynamic requests from the client at runtime. Also, the study effort suggested a way for

activating the VMPr phase of the two-phase optimization approach based on an Extreme Learning Machine. As a result, the VMPr phase's performance is improved, which also improved the performance metrics under consideration.

This research paper presented an Extreme Learning Machine Prediction Basis Triggering approach for virtual machine placement in cloud computing data centers to address the difficult VMP challenges in the IaaS environment. As a consequence, resource overbooking (server and network) and service flexibility (vertical and horizontal) are taken into account. It combines the benefits of offline (static) and online (dynamic) VMP formulations. Additionally, some novel prediction-based triggering algorithms were presented to specify the optimum time to start the placement recalculation phase. It is therefore the most complicated.

400 different circumstances were used in an experimental evaluation of a benchmark study. By generating a minimal cost function with an improvement of more than 10.5%, the findings show that the recommended VMPr triggering approach surpasses the benchmark work. The suggested prediction-based triggering techniques based on Extreme Learning Machine, as shown in table 4, offer a superior optimal solution from the reconfiguration phase of the two-phase optimization strategy. Further work includes formulating the VMP problem with a more intricate Cloud Federation that can account for more than five data centers, putting the suggested work into practice, and formulating the VMP problem with the dynamic PM that accounts for both maintenance and the deployment of new generation hardware.

## References

- A. Baloglavov, Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud Computing. *Future Gener. Comp. Syst.*, 755-768.
- Amarilla, A., Benitez, L., Zalimben, S., Lopez-Pires, F., & Baran, B. (2017). Evaluating a Two-Phase Virtual Machine Placement Optimization Scheme for Cloud Computing Datacenters. *MIC/MAEB 2017*, (pp. 4-7). Barcelona.
- Angeles, S. (2014). *Virtualization Vs Cloud Computing: What's the Difference?* Business News Daily.
- Biran, O., Corradi, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., et al. (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)* (pp. 498-506). IEEE Computer Society.
- Calcavecchia, N. M., Biran, O., Hadad, E., & Moatti, Y. (2012). VM placement strategies for cloud scenarios in cloud computing (CLOUD). *IEEE 5th International Conference* (pp. 852-859). IEEE.
- Chamas, N., Lopez-Pires, F., & Baran, B. (2017). Two Phase Virtual Machine Placement Algorithms for Cloud Computing: An Experimental Evaluation under Uncertainty. *IEEE*.
- Ding, Zhao & Ru nie, (2013). Extreme Learning Machine: Algorithm, Theory and Applications *ArtifIntell Rev (2015) 44:103–115 DOI 10.1007/s10462-013-9405-z*
- Dong, J., Wang, H., Jin, X., Li, Y., Zhang, P., & Cheng, S. (2013). Virtual machine placement for improving energy efficiency and network performance in its cloud. in *Distributed Computing Systems Workshops (ICDSW), 2013 IEEE 33rd International Conference* (pp. 238-243). IEEE.
- Fang, S., Kanagavelu, R., Lee, B.-S., Foh, C., & Aung, K. (2013). Power-Efficient virtual machine placement and migration in data centers. *IEEE International Conference* (pp. 1408-1413). in Green Computing and Communications, GreenCom, 2013 IEEE and Internet of Things, IThings/CSPCom.
- Fang, W., Liang, X., Li, S., Chiaraviglio, L., & Xiong, N. (2013). VMPlanner: Optimising virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. *Computer Networks, vol.57* (1), 179-196.
- Ferreto, T., Netto, M., Calheiros, R., & Rose, C. D. (2011). server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems, vol.27*, 1027-1034.
- Goudarzi, H., & Pedram, M. (2012). Energy-efficient virtual machine replication and placement in a cloud computing system in cloud computing (CLOUDS). *IEEE 5th International Conference*, (pp. 750-757).
- Gupta, A., Milojevic, 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, ACM*, (pp. 1-6).
- H. Goudarzi, & M. Pedram. (2012). Energy-efficiency virtual machine replication and placement in a cloud computing system. in *cloud computing (CLOUD), IEEE 5th International Conference*, (pp. 750-757).



- 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*, (pp. 1-2).
- Huang, G. B., Zhu, Q. Y., & Siew, C., K. (2006) Extreme Learning Machine theory and application *Neuro computing 70 (2006) 489–501*
- Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. Melbourne, Australia: oTexts.
- Ihara, D., Pires, F. L., & Baran, B. (2015). Many-objective virtual machine placement for Dynamic Environments. J.J.Prevost, K.Nagothu, B.Kelly, & M.Jamshidi. (2013). Optimal update frequency model for physical machine state change state and virtual machine placement in cloud. *System of Systems Engineering (SoSE), 8th International Conference on IEEE*, (pp. 159-164).
- Kaur, G., & Bhardwaj, V. (2016). A review on VM Placement Strategies. *International Journal of Advanced Research in Computer Science and Software Engineering, Vol.6 (5)*, 521-526.
- Le, K., Bianchini, R., Zhang, J., Jaluria, Y., Meng, J., & Nguyen, T. D. (2011). Reducing Electricity cost through virtual machine placement in high performance computing cloud. in *Proceedings of the 2011 International Conference for High-Performance Computing, Networking, Storage and Analysis* (p. 22). ACM.
- Li, K., Wu, J., & Blaisse, A. (2013). Elasticity-aware virtual machine placement for cloud data centers. in *Cloud Networking (CloudNet), 2013 IEEE 2nd International conference* (pp. 99-107). IEEE.
- Li, K., Zheng, H., Wu, J., & Du, X. (2015). Virtual machine placement in cloud systems through the migration process. *International Journal of Parallel, Emergent and Distributed Systems*, 393-410.
- Masdari, M., Nabavi, S. S., & Ahmadi, V. (2016). An Overview of virtual machine placement schemes In Cloud Computing. *Journal of Network and Computer Applications*.
- McKenzie, Everette & Gardner. (2010). Damped Trend Exponential Smoothing: A modeling viewpoint *International Journal of Forecasting 26 (2010) 661–665*.
- Mell, P., & Grance, T. (2009). The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*.
- Ortigoza, J., Pires, F., & Baran, B. (2016). Workload generation for virtual machine placement in cloud computing environments. *XLII Latin American Computing Conference, CLEI*, (pp. 1-9).
- Paliwal, S. (n.d.). Performance challenges in cloud computing.
- Pires, F. L., & Baran, B. (2015). A Virtual Machine Placement Taxonomy. *International Symposium on Cluster, Cloud and Grid Computing, 15th IEEE/ACM*, 159-168.
- Pires, F. L., & Baran, B. (2013). Multi-objective virtual machine placement with service level agreement: A memetic Algorithm approach .in *proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing* (pp. 203-210). IEEE Computer Society.
- Pires, F. L., & Baran, B. (2014). virtual machine placement literature review (data). *Polytechnic School, National University of Asuncion on, Tech. Rep*.
- Pires, F. L., Baran, B., Benitez, L., Zalinmben, S., & Amarilla, A. (2017). Virtual machine placement for elastic infrastructures in overbooked cloud computing data centers under uncertainty. *Future Generation Computer Systems*.
- Pires, F. L., Baran, B., Benitez, L., Zalinmben, S., & Amarilla, A. (2017). Virtual machine placement for elastic infrastructures in overbooked cloud computing data centers under uncertainty. *Future Generation Computer Systems*.
- Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I., et al. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development, vol. 53, no. 4*, 1-4.
- Speitkamp, B., & Bichler, M. (2010). A mathematical programming approach for server. *IEEE Trans. Serv. Comput*, 266–278.
- Tchernykh, A., Schwiiegelsohn, U., Alexandrov, V., & Talbi, E. (2015). towards understanding uncertainty in cloud computing resource provisioning. *Procedia Comp. Sci.*, 1772-1781.
- Tomas, L., & Tordsson, J. (2014). An autonomic approach to risk-aware data center overbooking. *IEEE Trans. Aloud Comput.* , 292-305.

- Wu, G., Tang, M., Tian, Y.-C., & Li, W. (2012). Energy Efficient virtual machine placement in data centers by Genetic Algorithm. *in Neural Information Processing. Springer*, 315-323.
- Wu, J.-J., Liu, P., & Yang, J.-S. (2012). Workload characteristics-aware virtual machine consolidation algorithms. *in Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 42-49). IEEE Computer Society.
- Yahaya, R., H., Ambursa, F., U., & Galadanci, B., S. (2020). A two-phase Optimization Scheme with Efficient Prediction-based Triggering for Virtual Machine Placement in Cloud Datacenters. *Ilorin Journal of Computer Science and Information Technology (ILJCSIT)*<https://www.iljcsit.com.ng>ISSN: 2550-7214 (print) Vol. 3, No. 1 (2020)