ICIC Express Letters
Part B: Applications
Volume **15**, Number **4**, April **2024**

ICIC International ©2024 ISSN 2185-2766
pp. 345–355

ENHANCING CLOUD SECURITY AND PERFORMANCE THROUGH VM-LOGGER ASSIGNMENT OPTIMIZATION

Surapong Wiriya[1], Winai Wongthai[1,2,*], Thanathorn Phoka[1,2]
Nattapon Kumyaito[1], Kittiphop Mahawan[3] and Pornpimon Boriwan[4]

[1]Department of Computer Science and Information Technology
[2]Research Center for Academic Excellence in Nonlinear Analysis and Optimization
Faculty of Science
Naresuan University
99 Moo 9, Thapo Sub-District, Muang District, Phitsanulok 65000, Thailand
{ surapongw58; thanathornp; nattaponk }@nu.ac.th
*Corresponding author: winaiw@nu.ac.th

[3]Department of Computer Education
Faculty of Education
Phranakhon Si Ayutthaya Rajabhat University
Phranakhon Si Ayutthaya 13000, Thailand
mkittiphop@aru.ac.th

[4]Department of Mathematics
Faculty of Science
Khon Kaen University
Khon Kaen 40002, Thailand
pornpimon@kku.ac.th

Received July 2023; accepted September 2023

Abstract. *In this work, we address the challenging "VM-logger assignment problem" in the context of large-scale computer systems that require the efficient assignment of loggers to monitor virtual machines (VMs). To tackle this problem, we propose a mathematical model formulated as a constrained optimization problem with the primary goal of maximizing the number of workload-free loggers. The constraints involve each VM's security and performance level requirements, as well as the workload of the associated logger. Our model assigns each VM to a single logger, while a logger can oversee any number of VMs, as long as the constraints are met. Initially, we implement a random approach to solve the optimization problem. However, this approach proves to be inadequate in finding an assignment that satisfies the constraints when they are too restrictive. To overcome this challenge, we propose relaxing these restrictions by transforming the constraints of the original optimization problem into penalties on the objective function of a new unconstrained optimization problem. The solutions derived from the unconstrained optimization problem may not satisfy the original constraints. However, in our VM-logger assignment problem, these solutions have minimal impact on the overall VM-logger system or customer satisfaction. This research article presents a comprehensive overview of the problem, the proposed mathematical model, the relaxation of constraints, and the implications of our approach on the VM-logger assignment system and customer satisfaction.*

Keywords: Optimization, Virtual machine, Logging system, Cloud computing

1. **Introduction.** Cloud computing has emerged as a transformative technology in the field of information technology (IT), leveraging virtualization technology for efficient data storage and processing [1, 2]. As a type of Infrastructure as a Service (IaaS), cloud computing provides virtual machines (VMs) to customers using services such as Amazon

DOI: 10.24507/icicelb.15.04.345

Elastic Compute Cloud [3]. With applications in various sectors, including government, education, and medical trials, the global public cloud service market is projected to grow by approximately 18.8% in 2022, valued at roughly $490 billion [4]. Consequently, organizations aim to enhance efficiency, security, and service availability while reducing costs.

One of the critical concerns in IaaS cloud computing is security [5]. The Cloud Security Alliance (CSA) has published numerous reports on IaaS cloud security threats, emphasizing the need for robust security measures [6]. Logging systems play a crucial role in mitigating security concerns by monitoring incidents occurring within a customer's IaaS VM, such as unauthorized access or changes to a customer's file in a disk of the VM [7]. As the logging system becomes more active, CPU and RAM usage estimates increase, necessitating efficient resource allocation. To prevent compromising system or application performance, it is advisable to maintain CPU traffic demand at 80% or higher [8].

Optimization techniques have been employed in cloud research to enhance efficiency. Numerous studies have explored various aspects of cloud optimization, such as maximize resource utilization, minimize energy consumption, communication cost and security [9], optimal solutions for cloud resource usage [10], virtual machine allocation to the task using an optimization method [11], the use of optimization strategies to address issues such as unbalanced load or low VM resource utilization [12], and to achieve optimal resource allocation and minimize the total runtime of requested services in cloud computing [13]. It can be observed that optimization in the cloud primarily focuses on resource allocation, such as CPU, RAM, hard disk, and tasks. However, one area that remains unexplored is the VM-logger assignment problem.

In this work, we aim to bridge this gap by introducing a method for addressing the VM-logger assignment problem to benefit both service providers and cloud service users. By examining the appropriate assignment of VMs to loggers, we can further optimize cloud computing resources, enhance security, and improve overall system performance.

The VM-logger assignment problem involves efficiently allocating loggers to VMs while considering the constraints and requirements of each VM and logger. Our proposed method seeks to maximize the number of workload-free loggers while ensuring the constraints involving each VM's security and performance level requirements, as well as the workload of the associated logger, are met. Addressing this problem effectively will lead to better resource utilization, reduced costs, and improved customer satisfaction.

To tackle the VM-logger assignment problem, we propose a mathematical model formulated as a constrained optimization problem. Our model assigns each VM to a single logger, while a logger can oversee any number of VMs, as long as the constraints are met. The solutions derived from this optimization problem may not satisfy the original constraints, leading to suboptimal results. To overcome this challenge, we propose relaxing these restrictions by transforming the constraints of the original optimization problem into penalties on the objective function of a new unconstrained optimization problem.

By addressing the VM-logger assignment problem, we contribute to the growing body of research on cloud optimization and security. Our findings have implications for cloud service providers, who can use the proposed method to optimize their resource allocation, resulting in enhanced security, improved system performance, and reduced costs. Furthermore, our research will benefit cloud service users, who can expect better performance and security from their VMs in an optimized environment.

Research gaps and contributions: The main contribution of this paper lies in addressing the VM-logger assignment problem. A shortcoming in earlier works (such as [7, 14, 15, 16, 17]) is that their logging systems were limited to detecting processes within a single virtual machine, leading to reduced efficiency. And the logging system increases CPU and RAM usage when using multiple logging systems for VMs. To enhance efficiency, a logging system ought to be capable of monitoring processes across multiple virtual machines, which would, in turn, decrease CPU and RAM usage for the provider, allowing the system to operate at full capacity. To tackle the VM-logger assignment problem and achieve optimized resource allocation for both the provider and the customer, an appropriate process for determining the optimal assignment is utilized, ensuring that the computer can work at maximum efficiency. We need to find a solution to maximize the number of workload-free loggers while ensuring the constraints involving each VM's security and performance level requirements. We propose a mathematical model for VM-logger assignment optimization, along with an example.

2. **Background.**

2.1. **Infrastructure as a Service cloud architecture.**

Figure 1 illustrates the architectures of both IaaS cloud and logging systems. These architectures are adapted from our previous work [18, 19]. In Figure 1, the white boxes represent the primary elements of the IaaS architecture, namely the hypervisor, dom0, hw0, domU, hwU, disk0, diskU, and memU. Components ending with '0' indicate physical ownership and management by the IaaS provider, while those ending with 'U' denote virtual ownership and management by the cloud customer.

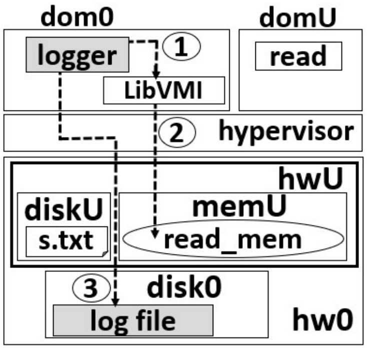


FIGURE 1. Infrastructure as a Service cloud architecture and logging system architecture, adapted from [18, 19]

The hypervisor, depicted by box number 2 in Figure 1, is software that enables a physical computer to accommodate multiple VMs. The top-left box, dom0 or domain 0, serves as the manager for all VMs created by customers. During system booting, the hypervisor launches dom0, which is also a VM and has exclusive access and control over hw0 and all customer-created VMs, or domUs. The bottom box in Figure 1, hw0, represents all the physical hardware managed by dom0.

The top-right white box corresponds to domU, or the user domain, which is a customer-created VM that runs on the hypervisor. DomU is an IaaS cloud product provided by the IaaS provider to the customer. HwU, physically located in hw0, represents the virtual hardware of domU. Although it is physically owned and managed by dom0 or the provider, it is virtually owned and managed by the domU owner or the IaaS customer. Disk0 represents the physical disk of dom0, while diskU is the virtual disk of domU. Finally, memU is the virtual main memory of domU.

2.2. **Logging system.**

A logging system operates by utilizing a logger to monitor a domU. In this context, we will refer to a domU as a VM and a dom0 as a host VM. One logger can monitor multiple VMs.

A logging system can log incidents occurring in a customer's IaaS VM, such as who has access to or what transpires with a customer file in a VM's disk [7, 18]. A logging system can consist of a logging process and a log file [20]. In this paper, we will refer to the logging process as a logger. The system architecture of the logging system is derived from our previous work [18, 19] and is also depicted in Figure 1. The box inside the domU in Figure 1 represents the read process. For the purposes of the experiment, we assume that this process could potentially be controlled by an attacker. As a result, the attacker could maliciously read a sensitive file, such as s.txt, belonging to an IaaS customer in diskU, as illustrated by the document shape within the diskU. The read mem in memU represents a reserve memory space for the read process provided by the OS hosting this process. The white box in the dom0 is LibVMI, the new name for XenAccess [21]. It is a C library installed in the dom0 that can access read mem in memU to detect the malicious activity of the read process, which is reading s.txt.

From Figure 1, the three working steps of the logger are represented by the circles numbered 1 to 3. In Step 1, the logger in the dom0 calls LibVMI to access memU to obtain the logging data from read mem (Step 2). This data includes i) the file name of s.txt or the string “s.txt” and ii) the process ID of the read process. Then, LibVMI accesses memU to obtain this data in read mem. Subsequently, it returns the obtained data to the logger. Finally, the logger processes the data and writes (in Step 3) the data into the log file.

2.3. **Optimization in cloud computing.** The optimization process is a critical aspect of developing a system that operates with maximum efficiency [22]. It involves the utilization of an algorithm to identify the optimal solution to an optimization problem. The development of an optimization model necessitates the establishment of a quantifiable and measurable criterion to assess the effectiveness of a decision. This criterion can aim to either maximize a favorable outcome or minimize costs associated with the decision [23].

To create an appropriate optimization model, it is imperative to establish the three fundamental components that constitute an optimization problem: the objective function, the problem constraints, and the decision variables. The definitions of each of these components are expounded in [24]. In the realm of cloud computing research, optimization techniques have been applied to enhancing efficiency, such as resource provisioning cost optimization in [3], optimal resource utilization in cloud computing in [10], and optimization strategies to address unbalanced load, slow convergence speed, and low utilization of VM resources in [12]. The optimization process involves the formulation of a specific optimization problem by incorporating the design variable(s), objective function, and constraints.

2.4. **Random search algorithm.** A random search algorithm is an optimization method that incorporates randomness or probability, usually through a pseudo-random number generator. This type of algorithm is also known as a Monte Carlo method or stochastic algorithm in academic literature [25]. Various random search algorithms have been developed, including simulated annealing, genetic algorithms, evolutionary programming, particle swarm optimization, ant colony optimization, cross-entropy, stochastic approximation, multi-start, clustering algorithms, and other techniques, which are widely used to solve both continuous and discrete global optimization problems [12, 26, 27, 28, 29, 30].

The random search algorithm operates in the following steps.

Step 0: Initialize algorithm parameters Θ_0 initial points $X_0 \subset S$ and iteration index $k = 0$.

Step 1: Generate a collection of candidate points $V_{k+1} \subset S$ according to a specific generator and associated sampling distribution.

Step 2: Update X_{k+1} based on the candidate points V_{k+1}, previous iterates and algorithmic parameters. Also update algorithm parameters Θ_{k+1}.

Step 3: If a stopping criterion is met, stop. Otherwise increment k and return to Step 1.

2.5. **Security and performance.** An accuracy refers to the precision of a logging system in recording information from volatile memory in a target monitored VM [31]. The accuracy of process detection by the logger affects the security of files on a customer's virtual machine. Hence, the researcher used this accuracy to define a security level.

Based on [31], we can define five security levels, from Level 1 to Level 5.

Level 1 is the highest level. A customer who chooses Level 1 will have one logger to monitor their VM1. This logger will not monitor any other VMs. This 1-to-1 monitoring method allows the logger to have high resource to capture malicious processes in VM1, resulting in high accuracy and security.

Level 2 is the second-highest level. A customer who chooses Level 2 will have one logger to monitor their VM1, but this logger will also monitor VM2. This 1-to-2 monitoring method requires the logger to share its resource between VM1 and VM2. Consequently, logger2 may not provide the same high accuracy and security as logger1.

Level 3 is the middle level. A customer who chooses Level 3 will have one logger to monitor their VM1, but this logger will also monitor VM2 and VM3. This 1-to-3 monitoring method requires the logger to share its resource between VM1, VM2, and VM3. Thus, logger3 may not provide the same level of accuracy and security as logger1.

Level 4 is the lower level. A customer who chooses Level 4 will have one logger to monitor their VM1, but this logger will also monitor VM2, VM3, and VM4. This 1-to-4 monitoring method requires the logger to share its resource between VM1, VM2, VM3, and VM4. Consequently, logger4 may not provide the same level of accuracy and security as logger1.

Level 5 is the lowest level. A customer who chooses Level 5 will have one logger to monitor their VM1, but this logger will also monitor VM2, VM3, VM4, and VM5. This 1-to-5 monitoring method requires the logger to share its resource between VM1, VM2, VM3, VM4, and VM5. As a result, logger5 may not provide the same level of accuracy and security as logger1.

Table 1 and Table 2, which are summarized from [31], present the tradeoffs between security levels and performance levels that will be considered in this work.

Table 1. Security level

Security level	Description	Security of files in VMs
5	Lowest security	86.47%
4	Low security	99.67%
3	Medium security	99.75%
2	High security	99.82%
1	Highest security	99.86%

Table 2. Performance level

Performance level	Description	Performance of VMs
5	Highest performance	83.00%
4	High performance	80.00%
3	Medium performance	73.00%
2	Low performance	66.00%
1	Lowest performance	50.00%

3. Design and Implementation.

3.1. Solutions for allocation of loggers and VMs design.

3.1. Solutions for allocation of loggers and VMs design. Suppose a demonstration host that supports 5 VMs. A solution for allocation can be designed. Figure 2 presents the solutions for the allocation of loggers and VMs. The columns represent logger1 to logger5, and the rows represent the allocated VMs to each logger. The white box represents a logger, and the number inside the white box represents the number of VMs working with the logger. To simulate the service for users, five scenarios are considered for allocation:

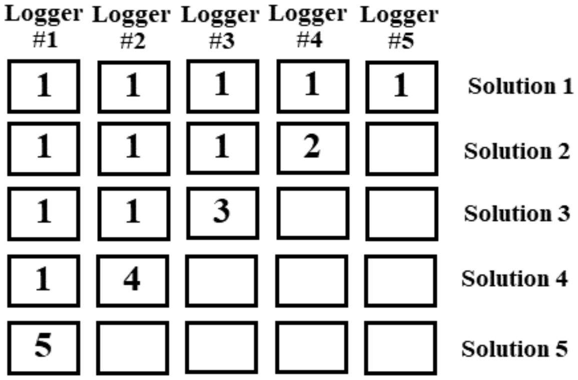
Scenario 1: All five users require security in detecting the logging process, and thus, each person will be working with one logger per VM.

Scenario 2: Three users require security in detecting the logging process and efficient performance of two VMs.

Scenario 3: Two users require security in detecting the logging process and efficient performance of three VMs.

Scenario 4: One user requires security in detecting the logging process and efficient performance of four VMs.

Scenario 5: All five users require efficient performance of five VMs.



Logger #1	Logger #2	Logger #3	Logger #4	Logger #5	
1	1	1	1	1	Solution 1
1	1	1	2		Solution 2
1	1	3			Solution 3
1	4				Solution 4
5					Solution 5

Figure 2. Examples of solutions for assignment of loggers and VMs

3.2. Optimization model. This section commences with the development of an optimization model to address the VM-logger assignment problem. The objective of this model is to determine the optimal assignment of loggers to virtual machines while maximizing the number of loggers with no workload. The model takes account of the security and performance requirements of the users to ensure that the assigned loggers meet the desired specifications.

Let X be a decision variable while $X \in I^{m\times n}$, that is,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix},$$

where $x_{ij} \in \{0, 1\}$, for each $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$. Notice that the notation x_{ij} indicates the assignment of the ith VM is whether assigned to the jth logger. That is, $x_{ij} = \begin{cases} 1, & \text{the } i\text{th VM is assigned to the } j\text{th logger} \\ 0, & \text{otherwise} \end{cases}$, and for each i and j, $U_i = \begin{bmatrix} x_{i1} & \cdots & x_{in} \end{bmatrix}$, $L_j = \begin{bmatrix} x_{1j} & \cdots & x_{mj} \end{bmatrix}^T$, and $\sum_{j=1}^{n} x_{ij}$ for each i must be equal to 1. This means that a VM will be assigned to only one logger. For each j, we define the workload of L_j by

$$N(L_j) = \sum_{i=1}^{m} x_{ij}$$

Therefore, the optimization problem for the problem of interest will be represented in the following form.

$$\max f(X),$$

subject to $X \in I^{m\times n}$, $p_i \le h(U_i) = N(L_j) \le s_i$ for each $i \in \{1, 2, \dots, m\}$, where $h : I^{1\times n} \to I$ provides the workload $N(L_j)$ when $x_{ij} = 1$. The objective function $f : I^{m\times n} \to I$ is referred to as the number of workload-free loggers which will be maximized:

$$f(X) = \sum_{j=1}^{n} O(N(L_j)),$$

where $O(N(L_j)) = \begin{cases} 1, & \text{if } N(L_j) = 0 \\ 0, & \text{otherwise} \end{cases}$, p_i is the required performance level of the ith VM, and s_i the required security level of the ith VM.

For instance, assuming that there are five loggers and five VMs, the researcher replaces logger1 with L_1, logger2 with L_2, logger3 with L_3, logger4 with L_4 and logger5 with L_5, respectively. Similarly, the 1st VM is replaced with U_1, the 2nd VM with U_2, the 3rd VM with U_3, the 4th VM with U_4 and the 5th VM with U_5. To demonstrate how Solution 1 from Figure 2 can allocate logger 1:1, the researcher provides a representation in Figure 3.

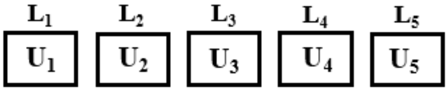


FIGURE 3. The solutions 1 for allocation of loggers and VMs

The proposed mathematical model can be illustrated as follows.

Step 1: Let

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{5\times 5}$$

Therefore,

$$\begin{aligned} U_1 &= [1 \quad 0 \quad 0 \quad 0 \quad 0] \\ U_2 &= [0 \quad 1 \quad 0 \quad 0 \quad 0] \\ U_3 &= [0 \quad 0 \quad 1 \quad 0 \quad 0] \\ U_4 &= [0 \quad 0 \quad 0 \quad 1 \quad 0] \\ U_5 &= [0 \quad 0 \quad 0 \quad 0 \quad 1] \end{aligned}$$

$$\begin{aligned} L_1 &= [1 \quad 0 \quad 0 \quad 0 \quad 0]^T \\ L_2 &= [0 \quad 1 \quad 0 \quad 0 \quad 0]^T \\ L_3 &= [0 \quad 0 \quad 1 \quad 0 \quad 0]^T \\ L_4 &= [0 \quad 0 \quad 0 \quad 1 \quad 0]^T \\ L_5 &= [0 \quad 0 \quad 0 \quad 0 \quad 1]^T \end{aligned}$$

Step 2: Count the number of VMs assigned to each logger.

$$\begin{aligned} N(L_1) &= x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 + 0 + 0 + 0 + 0 = 1 \\ N(L_2) &= x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 0 + 1 + 0 + 0 + 0 = 1 \\ N(L_3) &= x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 0 + 0 + 1 + 0 + 0 = 1 \end{aligned}$$

$$N(L_4) = x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 0 + 0 + 0 + 1 + 0 = 1$$
$$N(L_5) = x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 0 + 0 + 0 + 0 + 1 = 1$$

Step 3: Once the number of VMs assigned to each logger has been determined, set the value to ‘0’ if the logger is in use (i.e., the $N(L_j)$ value is greater than 0), and set the value to ‘1’ if the logger is inactive (i.e., the $N(L_j)$ value is equal to 0). This process is carried out to determine the number of loggers that are not assigned any workload, i.e., the number of workload-free loggers.

Based on Step 2, it can be observed that the summation of workload-free loggers is

$$O(N(L_1)) + O(N(L_2)) + O(N(L_3)) + O(N(L_4)) + O(N(L_5)) = 0 + 0 + 0 + 0 + 0 = 0$$

Step 4: The constraints for creating the optimization model were established based on the security and performance requirements of the virtual machines which is referred to Section 2.5. The allocation conditions for the logging system were designed by taking account of the needs of the service customer. As per the approach in [31], the security and performance levels were classified into five levels, numbered 1 to 5. The researcher aimed to maximize the number of virtual machines assigned to each logger while ensuring higher security levels. Additionally, a higher number of virtual machines assigned to the logger should correspond to a lower security level and a higher performance level.

If it is not possible to find an allocation solution that meets the constraints, the researcher has developed a penalty function model. This model converts a constrained optimization problem into an unconstrained one that can be used to find a solution.

We now consider an unconstrained optimization problem of the original constrained problem. The following notations will be used to define the problem:

$$cost(U_i) = \begin{cases} g_p \times (h(U_i) - p_i), & \text{if } p_i > h(U_i) \\ g_s \times (s_i - h(U_i)), & \text{if } s_i < h(U_i) \\ 0, & \text{otherwise} \end{cases}$$

denoting the negative penalty value of U_i. The weight of the penalty for a security limitation adjustment is denoted as g_s. As the main purpose is to ensure higher security levels, a weight of 2 is assigned to g_s. On the other hand, the weight of the penalty for a performance limitation adjustment is denoted as g_p and a weight of 1 is assigned to it.

Therefore, the unconstrained optimization problem is represented as follows:

$$\max f(X) + \sum_{i=1}^{m} cost(U_i),$$

subject to $X \in I^{m \times n}$.

3.3. **Flowchart of optimal logging system model.** A flowchart demonstrating an optimal logging system model for virtual machines (VMs) on cloud computing is shown in Figure 4. The following steps will be explained in detail to describe the process involved in this logging system model.

1) Get input data are number of loggers, number of VMs, security level, and performance level.

2) Generate random candidate solution for the parameters of the purpose function.

3) Calculate the value of the objective function for the randomized parameters.

4) Check whether the obtained objective function value is the best value. If so, store the value and its associated solution. Increase the number of duty cycles.

5) Check whether the duty cycle conditions have met the number of cycles. For this operation we set the number of cycles to 10000. If correct, terminate the operation and return the parameter value. However, if the work does not complete the number of cycles, repeat Steps 2-5.

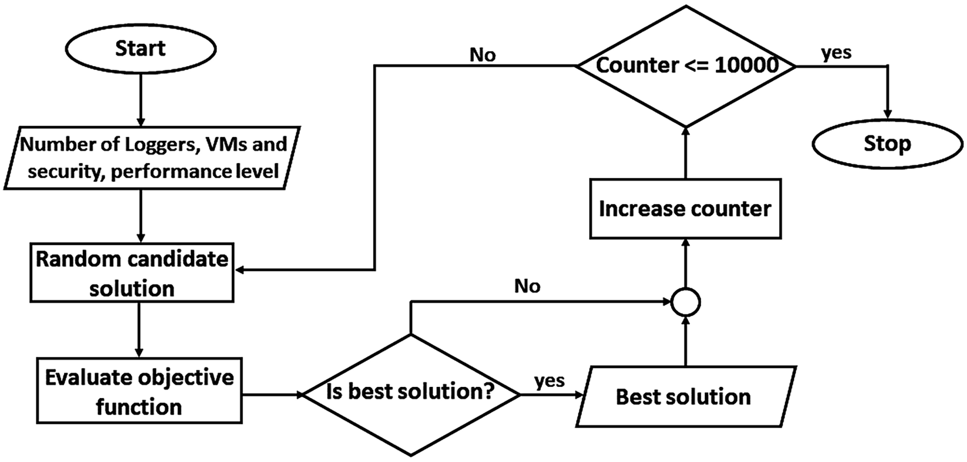
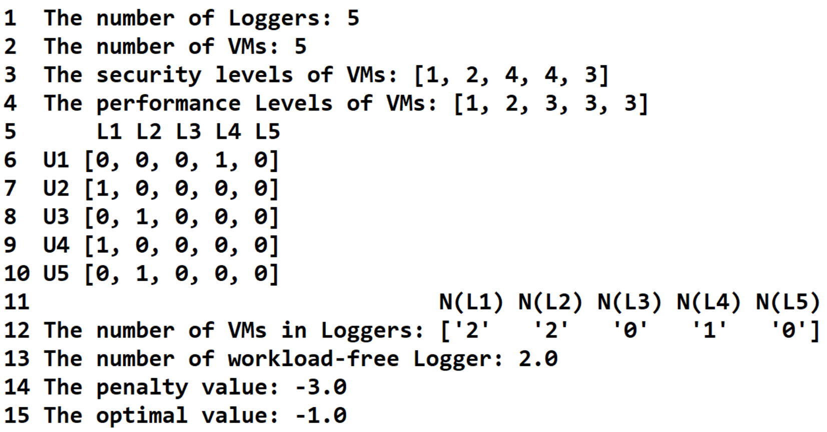

Start
Number of Loggers, VMs and security, performance level
Random candidate solution
Evaluate objective function
Is best solution?
yes
Best solution
No
Increase counter
Counter <= 10000
No
yes
Stop

Figure 4. Flowchart of optimal logging system model for VMs on cloud computing

4. **Numerical Example.** Figure 5 indicates the number of loggers $n = 5$ and the number of VMs $m = 5$. Furthermore, the performance levels of VMs $p_1 = 1$, $p_2 = 2$, $p_3 = 3$, $p_4 = 3$, and $p_5 = 3$ and the security levels of VMs $s_1 = 1$, $s_2 = 2$, $s_3 = 4$, $s_4 = 4$, and $s_5 = 3$. Based on the input data values, the unconstrained optimization problem proposed results in an optimal solution.

$$X^* = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{5\times 5}$$



```
The number of Loggers: 5
The number of VMs: 5
The security levels of VMs: [1, 2, 4, 4, 3]
The performance Levels of VMs: [1, 2, 3, 3, 3]
    L1 L2 L3 L4 L5
U1 [0, 0, 0, 1, 0]
U2 [1, 0, 0, 0, 0]
U3 [0, 1, 0, 0, 0]
U4 [1, 0, 0, 0, 0]
U5 [0, 1, 0, 0, 0]
                             N(L1) N(L2) N(L3) N(L4) N(L5)
The number of VMs in Loggers: ['2'   '2'   '0'   '1'   '0']
The number of workload-free Logger: 2.0
The penalty value: -3.0
The optimal value: -1.0
```

Figure 5. The best solution for assigning VM/VMs to each logger

Based on the optimal solution of the VM-logger assignment model, the 2nd, 4th VMs is assigned to the 1st logger while the 3rd, 5th VMs is assigned to the 2nd logger and the 1st VM is assigned to the 4th logger. It follows that there are two remaining workload-free of loggers $f(X^*) = 2$. We note that each ith VM has the corresponding values $p_i \leq h(U_i) = N(L_j) \leq s_i$ as follows:

$$\begin{aligned} p_1 = 1 \leq h(U_1) = N(L_4) = 1 \leq s_1 = 1 \\ p_2 = 2 \leq h(U_2) = N(L_1) = 2 \leq s_2 = 2 \\ p_3 = 3 \not\leq h(U_3) = N(L_2) = 2 \leq s_3 = 4 \\ p_4 = 3 \not\leq h(U_4) = N(L_1) = 2 \leq s_4 = 4 \\ p_5 = 3 \not\leq h(U_5) = N(L_2) = 2 \leq s_5 = 3. \end{aligned}$$

This means that the costs of U_1 and U_2 equal to 0 whereas the costs of U_3, U_4 and U_5 equal to -1 summed up to $\sum_{i=1}^{5} cost(U_i) = -3$. This results the optimal value to be $f(X^*) + \sum_{i=1}^{5} cost(U_i) = -1$.

The aim of the mathematical model developed in this research is to minimize the usage of loggers, thereby optimizing the CPU and RAM usage of the service provider's computer. However, cloud IaaS providers must be mindful of the trade-offs between security and performance when granting customers the freedom to choose their logging levels. The allocation of the logging system may not suit all customers, and the penalty value may need to be adjusted to ensure optimal allocation for all. Additionally, if a customer's requirements change, a new allocation may be necessary.

5. Conclusion.

To develop a mathematical model for optimizing the assignment of VM-loggers, we leveraged experimental data from [31] and employed random search techniques to obtain the most effective model for the unused loggers. In this paper, we utilized an adjustment function model that can accommodate complex requirements and support the allocation of the logging system. This model is also capable of handling calculations for identifying suitable methods that align with changes in the computer environment. The mathematical model developed by the researcher facilitated the determination of an optimal allocation method that utilizes the minimum number of loggers, resulting in cost-effective and efficient utilization of the service provider's CPU and RAM resources.

In the future work, it is possible to incorporate additional hardware-related variables into the model to better accommodate real-world computing environments. For this experiment, we are only working with a single physical machine. Furthermore, if there are multiple physical machines and the VMs created do not specify their physical machine, we can extend this method to allocate the VMs to the appropriate physical machine.

REFERENCES

[1] A. Bhawiyuga, D. P. Kartikasari, K. Amron, O. B. Pratama and M. W. Habibi, Architectural design of IoT-cloud computing integration platform, *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol.17, no.3, pp.1399-1408, 2019.

[2] R. Kaur and G. Kaur, Proactive scheduling in cloud computing, *Bulletin of Electrical Engineering and Informatics*, vol.6, no.2, pp.174-180, 2017.

[3] S. Chaisiri, B.-S. Lee and D. Niyato, Optimization of resource provisioning cost in cloud computing, *IEEE Transactions on Services Computing*, vol.5, no.2, pp.164-177, 2011.

[4] L. S. Vailshery, *Public It Cloud Services Global Market Growth 2011-2023*, https://www.statista.com/topics/2739/cloud-infrastructure-as-a-service/#topicOverview, 2022.

[5] W. Wongthai and A. van Moorsel, Quality analysis of logging system components in the cloud, *Information Science and Applications (ICISA 2016)*, pp.651-662, 2016.

[6] The Cloud Security Alliance (CSA), *The Treacherous 12 Top Threats to Cloud Computing + Industry Insights*, Tech. Rep., 2017.

[7] R. Ko, P. Jagadpramana and B. S. Lee, Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments, *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp.765-771, 2011.

[8] Microsoft, *High CPU Usage Troubleshooting Guidance*, https://learn.microsoft.com/en-us/troubleshoot/windows-server/performance/troubleshoot-high-cpu-usage-guidance, 2022.

[9] D. Saxena, I. Gupta, J. Kumar, A. K. Singh and X. Wen, A secure and multiobjective virtual machine placement framework for cloud data center, *IEEE Systems Journal*, vol.16, no.2, pp.3163-3174, 2021.

[10] H. K. Ala'a Al-Shaikh, A. Sharieh and A. Sleit, Resource utilization in cloud computing as an optimization problem, *Resource*, vol.7, no.6, 2016.

[11] P. S. Rawat, P. Dimri and G. P. Saroha, Virtual machine allocation to the task using an optimization method in cloud computing environment, *International Journal of Information Technology*, vol.12, pp.485-493, 2020.

[12] X. Wei, Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing, *Journal of Ambient Intelligence and Humanized Computing*, pp.1-12, 2020.

[13] S. H. Hosseini, J. Vahidi, S. R. Kamel Tabbakh and A. A. Shojaei, Resource allocation optimization in cloud computing using the whale optimization algorithm, *International Journal of Nonlinear Analysis and Applications*, vol.12, no.Special Issue, pp.343-360, 2021.

[14] W. Wongthai and A. Van Moorsel, Performance measurement of logging systems in Infrastructure as a Service cloud, *ICIC Express Letters*, vol.10, no.2, pp.347-354, 2016.

[15] D.-S. Park, H.-C. Chao, Y.-S. Jeong and J. J. J. H. Park, *Framework of Service Accountability and Policy Representation for Trustworthy Cloud Services*, Springer Singapore, Singapore, 2015.

[16] R. K. L. Ko and M. A. Will, Progger: An efficient, tamper-evident kernel-space logger for cloud data provenance tracking, *2014 IEEE 7th International Conference on Cloud Computing*, 2014.

[17] C. H. Suen, R. K. Ko, Y. S. Tan, P. Jagadpramana and B. S. Lee, S2logger: End-to-end data tracking mechanism for cloud data provenance, *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp.594-602, 2013.

[18] W. Wongthai, *Systematic Support for Accountability in the Cloud*, Ph.D. Thesis, Newcastle University, 2014.

[19] P. Chan-in and W. Wongthai, Performance improvement considerations of cloud logging systems, *ICIC Express Letters*, vol.11, no.1, pp.37-43, 2017.

[20] W. Wongthai and A. van Moorsel, Logging system architectures for Infrastructure as a Service cloud, *Journal of Telecommunication, Electronic and Computer Engineering*, vol.9, pp.35-40, 2017.

[21] L. Projects, *Functions*, http://libvmi.com/api/#Functions, 2019.

[22] Z. Zeng, D. Lu, Y. Hu, G. Augenbroe and J. Chen, A comprehensive optimization framework for the design of high-performance building systems, *Journal of Building Engineering*, vol.65, 105709, 2023.

[23] G. C. Calafiore and L. El Ghaoui, *Optimization Models*, Cambridge University Press, 2014.

[24] S. S. Rao, *Engineering Optimization: Theory and Practice*, John Wiley & Sons, 2019.

[25] Z. B. Zabinsky et al., *Random Search Algorithms*, Department of Industrial and Systems Engineering, University of Washington, USA, 2009.

[26] M. J. Al-Muhammed and R. A. Zitar, Probability-directed random search algorithm for unconstrained optimization problem, *Applied Soft Computing*, vol.71, pp.165-182, 2018.

[27] J. Dogani and F. Khunjush, Cloud service composition using genetic algorithm and particle swarm optimization, *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, pp.98-104, 2021.

[28] S. Mostafavi and V. Hakami, A stochastic approximation approach for foresighted task scheduling in cloud computing, *Wireless Personal Communications*, vol.114, pp.901-925, 2020.

[29] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung and Y. Li, Cloud computing resource scheduling and a survey of its evolutionary approaches, *ACM Computing Surveys (CSUR)*, vol.47, no.4, pp.1-33, 2015.

[30] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian and J. Yang, Sampling network guided cross-entropy method for unsupervised point cloud registration, *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp.6128-6137, 2021.

[31] T. Auxsorn, W. Wongthai, T. Phoka and W. Jaiboon, Performance considerations of a logging system simultaneously with a customer virtual machine in Infrastructure as a Service cloud, *Information Science and Applications (ICISA 2019)*, pp.285-296, 2020.