ITA: THE IMPROVED THROTTLED ALGORITHM OF LOAD BALANCING ON CLOUD COMPUTING

Hieu N. Le[1] and Hung C. Tran[2]

[1]Department of Information Technology,
Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam
[2]Posts and Telecommunication Institute of Technology, Ho Chi Minh City, Vietnam

ABSTRACT

Cloud computing makes the information technology industry boom. It is a great solution for businesses who want to save costs while ensuring the quality of service. One of the key issues that make cloud computing successful is the load balancing technique used in the load balancer to minimize time costs and optimize costs economically. This paper proposes an algorithm to enhance the processing time of tasks so that it can help improve the load balancing capacity on cloud computing. This algorithm, named as Improved Throttled Algorithm (ITA), is an improvement of Throttled Algorithm. The paper uses the Cloud Analyst tool to simulate. The selected algorithms are used to compare: Equally Load, Round Robin, Throttled and TMA. The simulation results show that the proposed algorithm ITA has improved the processing time of tasks, time spent processing requests and reduced the cost of Datacenters compared to the selected popular algorithms as above. The improvement of ITA is because of selecting virtual machines in an index table that is available but in order of priority. It helps response times and processing times remain stable, limits the idling resources, and cloud costs are minimized compared to selected algorithms.

KEYWORDS

Cloud Computing, Load Balancing, Processing Time, Improved Throttle Algorithm.

1. INTRODUCTION

Nowadays, as the demand for using common resources and services on the Internet is rapidly increasing, the issues about ensuring the efficiency of time, service quality and cost-saving are the greatest purposes. To meet that demand, in June 2007, Cloud Computing model was launched, and Amazon promoted research and deployment. Shortly thereafter, with the participation of large companies: Microsoft, Google, IBM, etc. Cloud Computing is promoted to thrive. Cloud computing (Cloud Computing) [1], [2], [3], [4] is the trend of developing computer networks, inheriting previous networks and distributed computing concepts to integrate on-demand machine resources, convenient and faster way. It is allowing users to deliver services, to release resources easily, to reduce communication with suppliers, and users need to pay only for what they used (pay-by-use).

According to [4], QoS in cloud includes the following models: load balancer, system model and the applications of QoS models. From here, we can see that load balancing is one of the important factors in ensuring the quality of service on the cloud and it is one of the research directions for cloud development and improvement. To improve the performance of cloud services, resource management [5] faces basic issues such as resource allocation, resource response, connecting to resources, discovering unused resources, mapping corresponding resources, modelling resources, providing resources, and planning resource utilization. In particular, the plan for resource use is

based on connections over time and the processing time of the service. From there we can study how to improve the processing time, we come up with a solution for the request allocation in load balancing. This is one of the research directions for improving cloud technology and making the cloud more and more perfect and advanced.

Cloud computing [6], [7], [8], [9] has been evolving since the 1980s through several phases including grid and utility computing, Application Service Provider, and software as a service (Software as a Service). Until 2006, the term "cloud computing" really emerged strongly. During this time, Amazon released its Elastic Compute Cloud (EC2) services, which allowed people to "hire computing power and processing power" to run their enterprise applications, started delivering browser-based enterprise applications that year. Three years later, in 2009, Google App Engine was born and became one of the historic milestones of electricity development in cloud computing. However, along with the constant development, this will also lead to some problems on cloud computing, especially the problem of overload on cloud computing. Therefore, many load balancing algorithms are appeared to solve the overload problem.

In this paper, we propose the ITA, Improved Throttle Algorithm, that enhances load balancing with better task processing time. The article includes the following sections: Section I is the introduction; Section II is discussed about the related works; Section III is the proposed algorithm ITA; Section IV are simulation results and evaluation; Section V is the conclusion.

2. RELATED WORK

The researchers have been working on load balancing and they have found out many algorithms to solve these issues on cloud computing. To study cloud computing, Rajkumar Buyya and co-authors [10] had reported their project of developing a Cloud Analyst Tool, a simulation tool for people to create their cloud environment and test it. In this report, they had integrated some famous algorithms inside. This tool is opening a new generation of cloud computing studies, especially the load balancer. Till the year 2018, D. Asir et al. [11] once again evaluated CloudSim and its related tools such as Cloud Analyst. The paper showed us a very positive prospect of CloudSim. They concluded that CloudSim is a great tool for people to develop and study cloud before deploying it, CloudSim also helps us model and simulate the cloud-computing environment efficiently.

After the appearance of CloudSim and its tools, many of research have been conducted and performed well in this simulation environment. In 2012, Klaithem Al Nuaimi et al. [12] had surveyed and gave the readersa big picture of load balancers in cloud. What they had introduced are challenges in load balancing and they had reviewedthe existing load balancing algorithms (static and dynamic load balancers). This study also evaluated some well-known algorithms such as CLBDM (Central Load Balancing Decision Model), MapReduce algorithm, MapReduce algorithm, WLC (weighted least connection), ESWLC (Exponential Smooth Forecast based on Weighted Least Connection), dual direction downloading algorithm from FTP servers (DDFTP), Load Balancing Min-Min (LBMM), Opportunistic Load Balancing algorithm (OLB). In 2015, Abhay Kumar et al. [13] had proposed a new static load balancer in cloud, the authors had combined Monitoring Load Balancing Algorithm with Throttled Load Balancing Algorithm for their new approach. The simulation of this study was compared to Throttled Algorithm, using the Overall Response Time and the datacenter Processing Time. The outcome result was better than the Throttle Algorithm.

With the focus of this study, we would like to introduce about the recent algorithms which are popular and well-known in load balancing on cloud computing.

2.1. Round-Robin Algorithm

N. Swarnkar et al. [14] discussed about the Round-Robin circle approach in load balancing. The Round-Robin load balancing technique is the simplest technique built and widely used for time-sharing systems. Round-Robin performs a fair load distribution in a rotating circular order of VMs. This algorithm has advantages such as simple algorithm, ensuring fairness operation to handle tasks timely, responding quickly if the VMs' processing powers are assumed to be equal. There is no hunger in this algorithm. The disadvantages of Round Robin: each node has a fixed interval time, there is no flexibility and expansion, there are many nodes that can have heavy loads while others can be idle, do not save the previous allocation state of VM.

2.2. Weighted Round-Robin Algorithm

S. Swaroop Moharana et al. [15] discussed the weighted Robin algorithm for load balancing in the cloud environment. The Weighted Round-Robin algorithm performs circular distribution based on the Round-Robin algorithm and relies on the capacity of each virtual machine through its weight table to distribute the load to the virtual machines respectively. Advantages of this algorithm: improved Round-Robin algorithm, operating in a rotation manner but combined with the weight-table of the processing capacity of virtual machines, more efficient than original Round-Robin in case the virtual machines' processing power is different. Besides, the disadvantages are: no flexibility and expansion, do not save the previous allocation state of virtual machines.

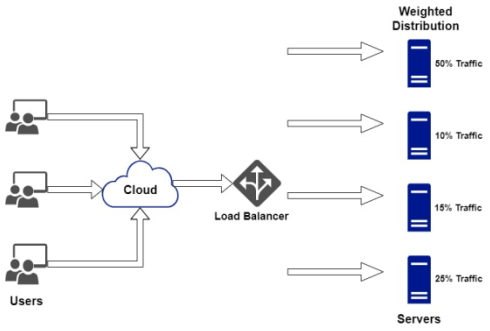

Figure 1. Weighted Round-RobinAlgorithm Model

2.3. Active Monitoring Load Balancer Algorithm

The Active Monitoring Load Balancer Algorithm [16] aims to maintain load balancing on all available virtual machines. The algorithm maintains information about each virtual machine and the current number of requests allocated to the virtual machine respectively. When a request is allocated in a virtual machine, the virtual machine with the least load will be determined by the intermediate datacenter. If there is more than one available VM, the first virtual machine ID is selected. The middle datacenter then sends the request to the virtual machine identified by that ID and updates the request allocation to the virtual machine. When the middle datacenter receives a response from the Cloudlet, it reduces the number of allocations to the virtual machine by one. With this algorithm, the advantages: execution time depends on the state of the system, the load will be transferred from the heavy-load VM to the lighter-loaded VM, improve response time significantly. The disadvantages: only rely on the current load of the VM to allocate new requests, choosing a less-load VM to decide whether to make the next request allocation. This sometimes will not get good results in some cases.

2.4. Throttled Algorithm

Makroo et al. [17] discussed the throttled load balancing method for cloud environment. The Throttled algorithm balances the load by maintaining a configuration table of virtual machines and their status. When a request is required to allocate in virtual machines from the Datacenter, the load balancer chooses the VM which is first found in the list of available VMs. If a virtual machine is found, the request will be allocated to this virtual machine. If no VMis found, it will return the Datacenter with a value = *-1* then Datacenter will put this request/task into the queue and wait for the found VM. This algorithm is a dynamic load balancing algorithm, the advantages: the list of virtual machines is maintained with the status of VMs, good performance, use resources effectively. The disadvantages: need to scan all WMs, it is not effective if the idle VM is at the bottom of the list because the algorithm requires selecting the first idle virtual machine from the list to allocate tasks, do not consider the current load of virtual machines.

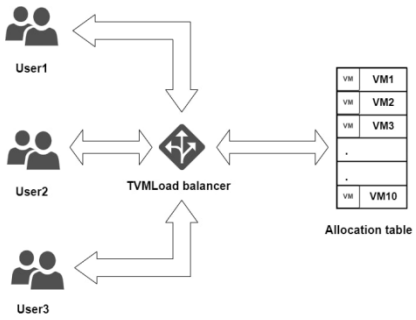

Figure 2. Throttled Algorithm Model

2.5. Other Algorithms

In 2013, Shridhar G. Domanal et al. [18] proposed a modification of the Throttled algorithm. The algorithm is smart that incoming requests for virtual machines are made available through the status list of all virtual machines. This status list is constantly updated to help improve the average system response time.

In 2014 Rakesh Kumar Mishra et al. [19] proposed two effective options for choosing a datacenter to reduce processing time. The first suggestion is to select a preferred datacenter based on Round Robin. Compared to the random selection algorithm, datacenter's processing time is better, but resource is not well-handled. The next proposal is based on the extended Round Robin to choose a datacenter with better processing time than random selection, but this depends on the datacenter configuration and scheduling costs. Instead of selecting a random datacenter, the algorithm can select the datacenter in a Round-Robin way to distribute requests uniformly across all datacenters in cloud. It leads to more resource usage, but datacenters may have different processing speeds and costs.

In 2017 Imtiyaz Ahmad et al. [20] proposed the Advanced Throttled Load Balancing Algorithm. Priority is assigned to each VM. The priority is calculated based on the capacity of the VM and the number and size of tasks assigned to the operation. The improved tuning scheduler selects the VM with the highest priority among the available VM sets. A priority threshold level is also set to avoid overloading. If the priority of the VM is less than the priority level, then the task is not allocated to that VM.

In 2018, Nguyen Xuan Phi et al [21] proposed the TMA, Throttled Modified Algorithm, to reduce response time and processing time on cloud computing. This algorithm has two indexes to

describe the status of available VM, 0 stands for available VM, and 1 stand for unavailable VM. The VM state is maintained with the 2 indexes table instead of scanning all the VMs status. It only needs to allocate the request to the virtual machine in the available table, so this reduces the response time and processing time. The result of the algorithm increases significantly as the number of virtual machines increases.

In 2018, the article of Gupta, S. Et al [22] aims to maximize the throughput and to increase the performance of the cloud services by proposing a new load balancing. The authors selected Throttled, Round-Robin, and their proposed algorithm, Advanced Throttled Algorithm to make a comparison. This ATA is also an improvement of Throttled algorithm by distributing workload evenly between VMs. An index table is maintained by load balancer that contains all the currently allocated request of VM. The new request arrived VM with the least load balance is chosen by the load balancer. Load balancer deallocates the VM after completion of the task. For simulation the authors used the cloud analyst tool, the result showed that the proposed algorithm was more efficient than the existing static and dynamic load balancing algorithms.

In 2019, Tran Cong Hung et al. [23] proposed MMSIA algorithm to improve the Max-Min scheduling algorithm, which improves the execution time of the requests by clustering size of requests and clustering utilization percent of VMs. The algorithm then allocates the largest cluster requests to the VM with the smallest utilization percent, this will be repeated till the request list is empty. In the simulation result, the MMSIA algorithm has improved the execution time.

In 2020, the research article of Abiodun K. Moses et al. [24] proposed an approach, which uses both maximum-minimum and round-robin algorithm (MMRR). With this algorithm, there quest having long execution time will be allocated with Max-Min and there quest having smallest execution time will be allocated by Round-Robin. In this paper, the authors also use Cloud analyst tool to implement the proposed technique and make a comparative analysis to the algorithm was conducted to optimize cloud services to clients. The findings of this proposed approach showed that Maximum Minimum Round Robin (MMRR) made significant changes to cloud computing. According to the simulation results, the loading time of datacenter was good in both Throttled and MMRR, but Round Robin was worst. The proposed MMRR performed better from other algorithms which are tested based on the whole response time (227,26ms) and cost-effectiveness (89%).

Also in 2020, Badshaha P Mulla et al. [25] studied about the VM allocation in heterogeneous cloud. The authors have proposed an enhancement in load balancing for efficient VM allocation in a heterogeneous cloud. Their proposal allocates independent user tasks or requests to available virtual machines in datacentre efficiently to manage proper load balancing. They aimed to minimize the response time also the processing time for requests. The proposed work has been simulated on Facebook cloud-based social networking application on the internet. Their results obtained a significant reduction of response time, this leads to the data centre processing time decreasing too, compared to Throttled and Round-Robin algorithms.

In 2021, T. J. B. Durga Devi et al. [26] proposed an application of neuro fuzzy inference system in load balancing. In this article, the authors also mention about the security of VM in cloud environment. NP-hard optimization problem corresponds to load balancing. They follow the Forbes about the introduction of General Data Protection Regulation, security in cloud continue to be an issue, the existing system uses a fuzzy based hybrid LB, but this is not satisfied. The authors focus on opportunities for improving CPU utilization and turnaround time and in terms of security, they proposed their work, MANFIS (Modified Adaptive Neuro Fuzzy Inference System). Parameters of MANFIS are optimized by introducing Fire-fly Algorithm. Security is

imposed on user authentication by using the Enhanced Elliptic Curve Cryptography. This is a password-less mechanism to authenticate users. With the results, the authors tell us the improvement and the satisfactory of security on cloud, they show better performance with respect to resource utilization, cost, and execution time, when compared with existing system, as shown by results of experimentation.

In 2021, an overview by Dalia Abdulkareem Shafiq et al. [27], the review study shows that Load Balancing is an important aspect on Cloud Computing environment, help to enhance the workload distribution and utilize the resources efficiently, especially improve the response time for cloud users. The paper tells us that there are a lot of issues related to LB, they are scheduling of tasks, migration, resource utilization, and so on. The authors survey and analyse the past six years research and studies on Load Balancing. This review also shows us the potential of intelligent approaches such as Artificial Intelligence, Machine Learning for LB on cloud. This study will be helpful for researchers to identify research problems related to load balancing, especially to further reduce the response time and avoid failures in the server. Another advantage of this study for our proposal, it is about the simulation tools and experimental environments. They also show that, CloudSim and Cloud Analyst are the best using for this field of study, it is the benefit of these tools.

The recent studies and research on load balancing are widely developing, but most of them are heuristic based and improvement of existing algorithms due to the cloud characteristics. On cloud, the LB must work real time to serve as fast as possible for user requests, so researchers and cloud providers cannot directly add in or integrate the AI or Machine Learning techniques into the LB. With these directions, we can improve Throttle Algorithm with implementing the lack of it, it is the priority order of VMs. The related works above give us more and more idea to help enhance the performance of LB on cloud, therefore we would like to propose our ITA, an improvement of Throttled Algorithm.

3. PROPOSED ALGORITHM

Currently, due to the dynamic characteristic of Throttled Algorithm, this algorithm in load balancing is mentioned and improved by many articles and research works. With many innovative purposes to reduce the response time and cloud processing time, to reduce resources waste, and to reduce load imbalance, we also would like to propose an improvement of the Throttled algorithm, named Improved Throttled Algorithm (ITA). In this study, the proposed algorithm improves from the work of [21] to help enhance response time, request processing time, and reduce the cost of the Datacenter. With this algorithm, we select a virtual machine for load balancing by using the available index with priority. This approach helps load balancer get faster processing time and minimize idle resources.

3.1. Research Model

Being an improved algorithm from Throttled, ITA chose the first available virtual machine to assign the task with the selected resource. In particular, the selected virtual machine is calculated to be used least (the minimum usage), which usage is calculated by the total processing time (Makespan) of the most recent task.

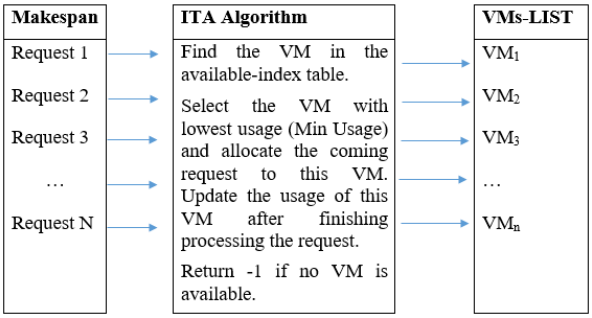

Figure 3. Research Model of ITA

Usage is calculated from the total processing time of the most recent task. datacenter cost is defined by the total of Data Transfer Cost and Virtual Machine Rental Cost. With this research model, we have the assumptions: The load balancer knows in advance the input requirements; The load balancer knows in advance the list of virtual machines, the ability of the virtual machines (these virtual machines have the same configuration of RAM, memory, and CPU; physical hosts have the same configuration of RAM, memory, CPU, bandwidth, and geographic differences of data centres). The target of this research model: Minimize response time and processing time (makespan); Help process requests faster; Limit the load imbalance.

3.2. Steps of the Algorithm

The sequences of ITA algorithm are explained and described in the following steps:

Step 1: Load Balancer of ITA performs load balancing by maintaining and constantly updating the Usage of Virtual Machine list.

- The index table contains the information of virtual machines (VMs) in an available state *'0'* (Available Index)
- Index table contains information of virtual machines (VMs) in a busy state *'1'* (Busy Index).

In the beginning, all virtual machines (VMs) are in the *"Available Index"* table and the *"Busy Index"* table is empty.

Step 2: The datacenter controller (DCC) receives a new processing task from a request.

Step 3: The datacenter controller (DCC) queries the ITA load balancer for subsequent allocation.

Step 4: The ITA load balancer detects and sends the virtual machine (VM) ID from the *"Available Index"* table with the *smallest Usage*, then return this virtual machine ID to the datacenter controller (DCC).

- The datacenter controller (DCC) will push that request into the virtual machine (VM) identified by that ID to process and notify the ITA load balancer about this allocation. The ITA load balancer will update that the virtual machine ID (VM) into the *"Busy Index"* table and wait for the next request from the datacenter Controller (DCC).
- In case, if the *"Available Index"* table is empty (all VMs are busy). The ITA load balancer will return a value of *-1* to the datacenter controller (DCC).The datacenter controller (DCC) will put that request in a queue for subsequent of allocations.

*Step 5:*After the request has been processed by the VM and the datacenter controller (DCC) receives a response. This will notify the ITA load balancer that the load balancer updates the table of *"Available Index"* and updates the Usage list of this table.

Step 6: If there are many requests, the datacenter controller repeats *Step 3*, and this is repeated until the *"Available Index"*table is empty.

Pseudocode of *getNextAvailableVm* function in step 4ofITA Algorithm

```
Begin
    vmId = -1;min = 0;i=0;
    For eachvminVMList
          usage = vm.getUsage();
          vmId = vm.getId();
          Ifi== 0 then
                min = usage;
          Else
                If min > usage then
                      min = usage;
                End If
          End If
          i++;
    End For
    Allocated(VmId);
    Return vmId;
End
```

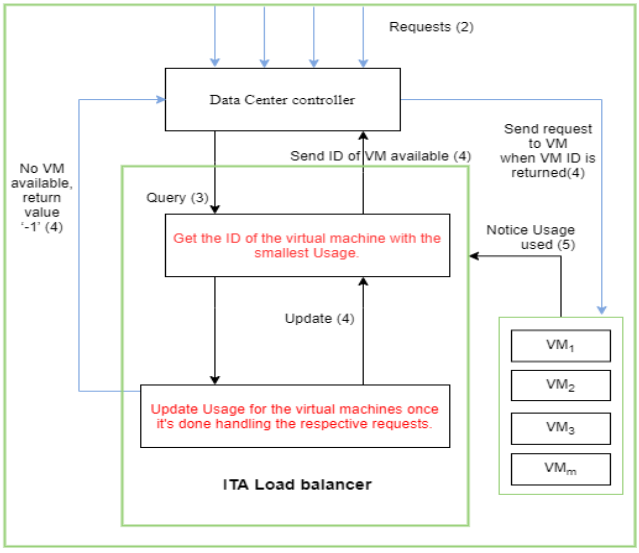

Figure 4. ITA Operation Diagram

4. SIMULATION RESULTS

This study uses the *Cloud Analyst* tool to install an ITA algorithm for simulating and testing, the experimental results show that our proposed method has improved the processing time compared to TMA [21] algorithm, Round-Robin algorithm, Equally Load algorithm, and original Throttled algorithm.

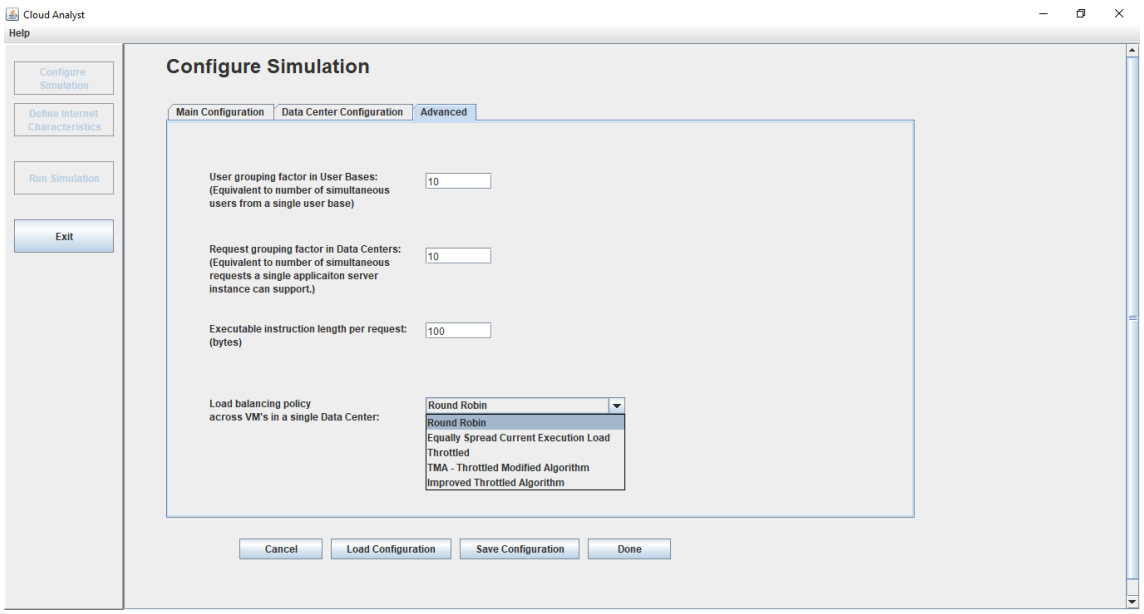

Figure 5. Cloud Analyst Tools with 5 Algorithms

4.1. Simulation Environment

We simulate the cloud environment by using *CloudSim* library set (open-source library suite, provided by http://www.cloudbus.org/) built with Cloud Analyst and program in JAVA language. We have created 4 different cloud simulation environments, 4 scenarios respectively, then we created a random request environment from different geographical *UserBases* but the service is on the same cloud. Then, we implement the ITA algorithm in a simulation environment. Similarly, we run the proposed algorithm and compare with four algorithms: Equally Spread Current Execution Load, Round Robin, Throttled, and TMA.

The proposed ITA algorithm is built by creating a *ThrottledVmITALoadBalancer* class, inheriting from the *VmLoadBalancer* class, updating a number of loading parameters (CPU, RAM …) and related properties. The method *getNumber* is to calculate the usage of the virtual machine and we adjust the built-in functions to match the proposed algorithm.

4.2. Evaluation Criteria

Experiment cloud simulation with the above parameters and run the existing Cloud Analyst load balancing algorithm: Round Robin, Throttled and Equally Spread Current Execution Load, additional TMA algorithm [21] and run the proposed algorithm, same input, compare the outputs, especially the response time parameter (Response Time), cost of datacenter (Costing).

With this simulation results, we use 4 key output variables to evaluate the performance of the 5 algorithms. They are *Overall Response Time (ORT)*, *datacenter Processing Time (DPT)*, *datacenter Request Servicing Time (DPT)*, and *datacenter Cost (DCC)*. According to cloudSim and CloudAnalyst Tool, the overall response time is the Series of the response time after serving requests from the cloud users. Response time is the sum of processing time and data transferring time via the networks. The processing time is the time that a machine / VM needs to take to process the task along with the coming request. This processing time is labeled with the datacenter, locally. The request service time is also calculated as the serving time of the datacenter, excluding the transferring time to the user. The datacenter cost is calculated from the

sum of total works in the VMs, the works may be various such as ALU action, CU action, storage action etc. Each action, we need to spend some power and time on it to work on, this number is the cost of an action, and in VMs we can call it as MIPS. If the job needs more MIPS, we need more costing for it. The total cost is the sum of all actions costing. With each variable, we get the max value, min value and average value after processing the user 's requests.

ORT = DPT + [Transfering Time]
DCC = [Action cost] * [Total actions – Total MIPS]

For Response Time, we use response time of virtual machine to evaluate its performance. The less predictive response time of the cloud is, the better efficiency of the algorithm is, the lower the cost is and the better that technique is. Requests are represented by *Cloudlets* in CloudSim and the size of the Cloudlet is randomly generated by the JAVA random function. The number of Cloudlets is from 100 to 1500, respectively.

Table 1. Parameters of Request/Cloudlet

Length	File Size	Output Size	Number of CPU (PEs)
100 ~ 1700	5000 ~ 45000	450 ~ 750	1

4.3. Simulation Results

We run testing with 4 cases of different input (different userbase, different datacenter, and VM quantity), 4 scenarios respectively. *Scenario 1*, simulation environment includes 01 Datacenter with 20 virtual machines, 2 Userbases. *Scenario 2*, simulation environment includes 01 Datacenter with 5 virtual machines, 3 Userbases. In s*cenario 3*, simulation environment includes 01 Datacenter with 5 virtual machines, 4 Userbases. Finally, s*cenario 4*, simulation environment includes 01 Datacenter with 50 virtual machines, 01 Datacenter with 5 virtual machines, 5 Userbases.

The simulated environments' detail parameters are displayed in *Figure 6, 7*. The experiment results of 4 scenarios are in *Table 2*.

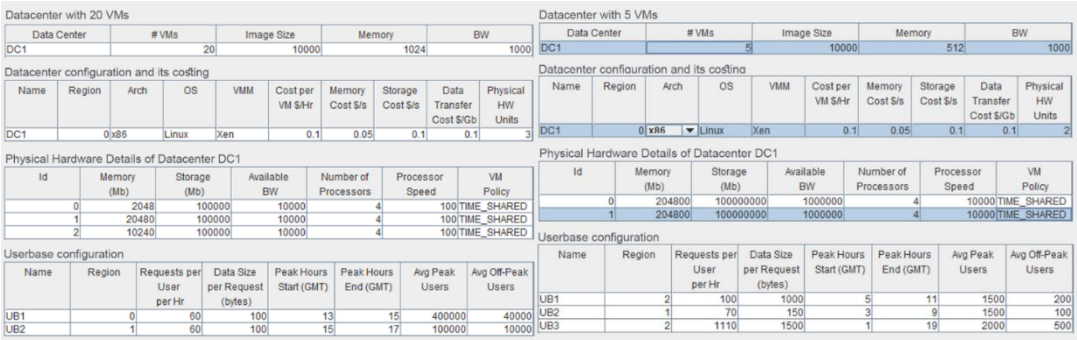

Figure 6. ***Scenario 1*** 's configuration (1 datacenter, 20 VMs, and 2 Userbases) and ***Scenario 2*** 's configuration (1 datacenter, 5 VMs, and 3 Userbases)

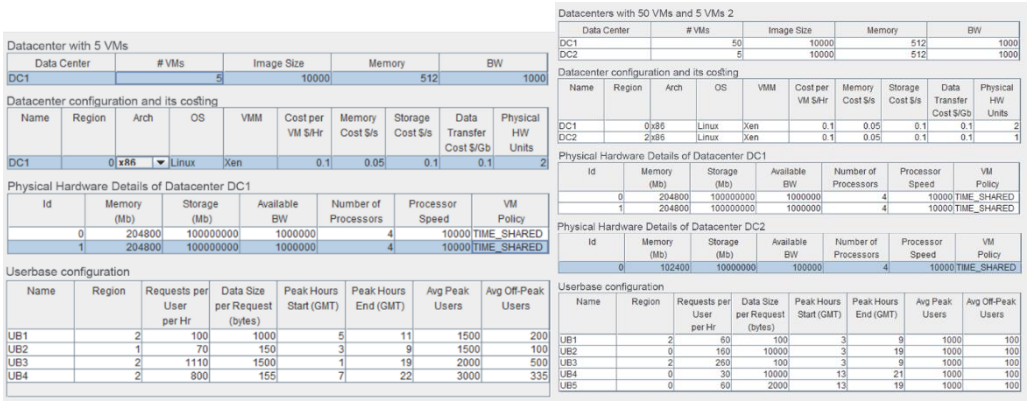

Figure 7. ***Scenario 3*** 's configuration (1 datacenter, 5 VMs and 4 Userbases) and ***Scenario 4*** 's configuration (2 datacenters, 50 + 5 VMs and 5 Userbases)

Table 2. Experimental results of 4 Scenarios

Scenario 1	Overall response time			Data Center processing time			DC Request Servicing Times			Data Center Cost		
	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	VM Cost ($)	Data Transfer cost	Total ($)
Equally Load	21,198.80	3,637.19	39,052.12	21,051.19	3,466.10	39,001.62	21,051.19	3,466.10	39,001.62	0.50	28.61	29.11
Round Robin	21,199.02	3,637.19	39,052.12	21,051.41	3,466.10	39,001.62	21,051.41	3,466.10	39,001.62	0.50	28.61	29.11
Throttled	9,774.77	289.48	27,750.66	9,668.98	250.02	27,696.11	9,668.98	250.02	27,696.11	0.50	28.83	29.33
TMA	9,774.77	289.48	27,750.66	9,668.98	250.02	27,696.11	9,668.98	250.02	27,696.11	0.50	28.83	29.33
ITA	9,562.11	159.12	38,642.87	9,463.10	1,729.10	26,334.34	9,463.10	1,729.10	26,334.34	-	22.68	22.68
Scenario 2	**Overall response time**			**Data Center processing time**			**DC Request Servicing Times**			**Data Center Cost**		
	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	VM Cost ($)	Data Transfer cost	Total ($)
Equally Load	299.96	157.13	397.45	0.56	0.02	1.10	0.56	0.02	1.10	0.50	89.66	90.16
Round Robin	299.96	157.13	397.45	0.56	0.02	1.10	0.56	0.02	1.10	0.50	89.66	90.16
Throttled	300.31	157.13	397.90	0.57	0.02	1.10	0.57	0.02	1.10	0.50	118.80	119.30
TMA	299.96	157.13	397.45	0.56	0.02	1.10	0.56	0.02	1.10	0.50	89.66	90.16
ITA	299.96	157.13	397.45	0.56	0.02	1.10	0.56	0.02	1.10	-	89.66	89.66
Scenario 3	**Overall response time**			**Data Center processing time**			**DC Request Servicing Times**			**Data Center Cost**		
	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	VM Cost ($)	Data Transfer cost	Total ($)
Equally Load	299.97	164.13	395.74	0.50	0.01	1.10	0.50	0.01	1.10	0.50	93.99	94.49
Round Robin	299.97	164.13	395.74	0.50	0.01	1.10	0.50	0.01	1.10	0.50	93.99	94.49
Throttled	299.97	164.13	395.74	0.50	0.01	1.10	0.50	0.01	1.10	0.50	93.99	94.49
TMA	299.97	164.13	395.74	0.50	0.01	1.10	0.50	0.01	1.10	0.50	93.99	94.49
ITA	299.97	157.12	395.74	0.50	0.01	1.10	0.50	0.01	1.10	-	93.99	93.99
Scenario 4	**Overall response time**			**Data Center processing time**			**DC Request Servicing Times**			**Data Center Cost**		
	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	Avg (ms)	Min (ms)	Max (ms)	VM Cost ($)	Data Transfer cost	Total ($)
Equally Load	50.60	38.94	65.13	0.75	0.02	1.62	1.57	1.00	2.51	5.50	22.03	27.53
Round Robin	50.60	38.94	65.13	0.75	0.02	1.62	1.57	0.10	2.50	5.50	22.03	27.53
Throttled	50.60	38.94	65.13	0.75	0.02	1.62	1.57	0.10	2.50	5.50	22.03	27.53
TMA	50.60	38.94	65.13	0.75	0.02	1.62	1.57	0.10	2.50	5.50	22.03	27.53
ITA	50.60	38.94	65.13	0.75	0.02	1.62	1.57	0.10	2.50	0.05	22.03	22.08

In Table 2, we can see that the results of the ITA algorithm are always at the minimum level among the 5 algorithms, with all 4 scenarios with different parameters. We can see the stable of ITA despite the variant inputs, because the 04 selected scenarios are randomly and very different from each other.

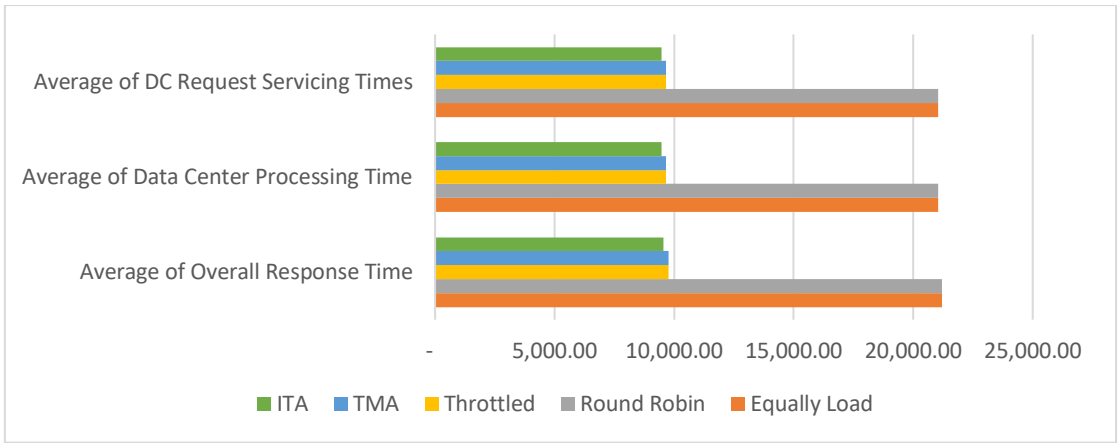

Figure 8. Result simulation of Scenario 1

Figure 8 represents for the 1st scenario, it can show that ITA algorithm has much better results than Throttled algorithm, TMA algorithm, Round-Robin algorithm, Equally Load algorithm with the same input data. With the3 params: Average of Overall Response Time, Average of Data Centre Processing Time, Average of DC Request Servicing Times, ITA algorithm is best performance algorithm. We can easily realize that Throttled algorithm, TMAalgorithm and ITA algorithm are the dynamic load balancers, so they must perform better then the group of static load balancer like Round-Robin and Equally Load.

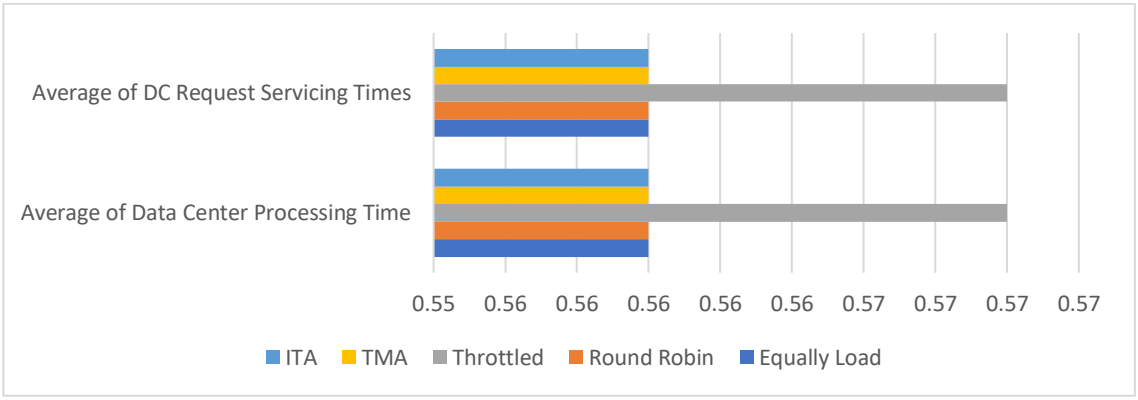

Figure 9. Result simulation of Scenario 2

Figure 9 represents for the 2nd scenario. We can see clearly about ITA, it is in the best ones among the 5 algorithms. We can notice that Average of Data Centre Processing Time and Average of DC Request Servicing Times have the smallest number (0.56) among them (0.56 – 0.75). No other algorithms (among 5 selected) can be better than ITA. In this case, the original Throttled Algorithm may work worst due to the number of VMs, we set only 5 VM for this case and we see the disadvantages of Throttled.

In scenario 3 and 4, we do not see the big different of all 5 algorithms because the configurations are so complicated. We have 3 and 5 userbases with different geographical characteristics. Then all the 5 algorithms perform well, all 3 parameters (Average of Overall Response Time, Average of Data Centre Processing Time, Average of DC Request Servicing Times) are equally. But the cost of datacenter shows that ITA is the best one. We can see more detail in figure 10.

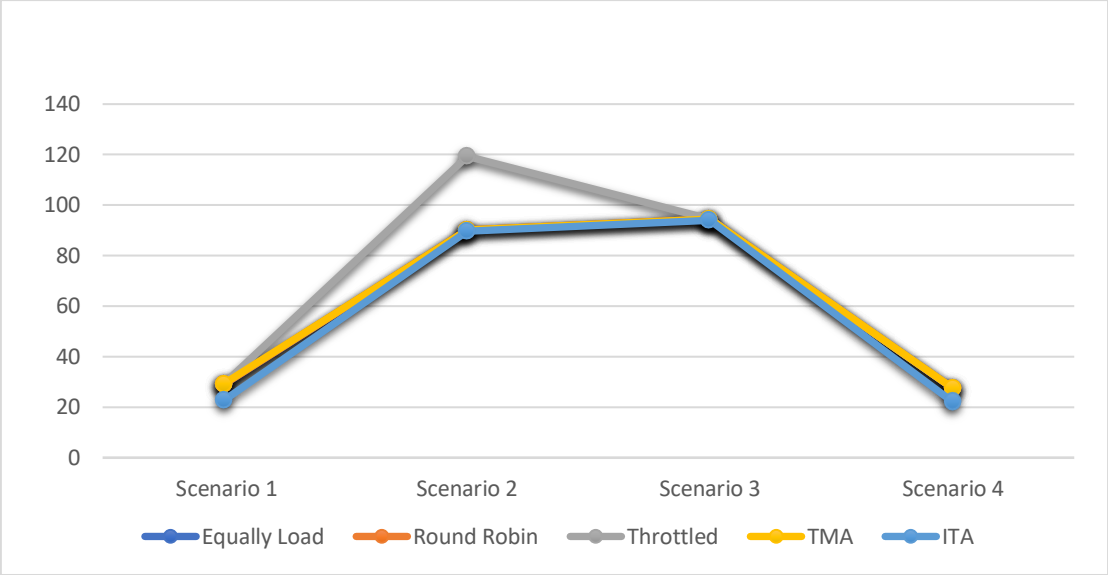

Figure 10. Total datacenter Cost of 4 Scenarios

In total, *figure 10* shows that ITA is the best saving cost for the cloud. ITA always keeps the lowest amount of costing in both 4 scenarios, and it can show the good improvement of ITA.

This part presents a simulation of the proposed ITA algorithm, the parameters as well as the given 4 scenarios based on the request process of users in the cloud environment. From there, record the response time, processing time, and cost of the cloud. Simulating ITA with the parameters of different datacenters, different numbers of virtual machines, bearing loads from 100 to 1500 requests, ITA reduces the cost of the cloud, the allocation of requests to the virtual machines handles evenly. Running the ITA algorithm simulation with a larger number of VMs on different datacenter also good results compared to other algorithms, in terms of improved costs than previous algorithms. But ITA will show the best performance if it serves the same userbase of requests.

5. CONCLUSIONS

This article presents an overview of the cloud, popular and recent load balancing techniques using in cloud computing environments. It also studies the approach of improving cloud computing performance by improving the load balancer. Through simulation using the GUI tool of Cloud Analyst, we implement and simulate load balancing techniques including Round Robin, Throttled, TMA (Throttle Modified Algorithm), Equally Load and ITA (Improved Throttled algorithm – the proposal). The results of the proposed algorithm (ITA) meet the goals such as improved time response, limited resource starvation, virtual machines with strong processing power to process more requests and saving cost. This study also helps balance the load more effectively, compared with popular algorithms like Round Robin, Throttled and Equally Load. The datacenter cost of this proposal is always lower than the others, which shows that the ITA algorithm can be used in practice and potential for further development. In the real world, we can apply our ITA to a real datacenter for testing and further research. Because it is based on Throttle Algorithm, which is present everywhere in our current networks. ITA is the potential to apply and implement with further testing and researching.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGMENTS

We, the authors would like to thank the Posts and Telecommunication Institute of Technology, Campus in Ho Chi Minh City, Vietnam. We also appreciate a lot the support from Ho Chi Minh City Open University, Vietnam.

REFERENCES

[1] S. Kemp, “Digital 2019: global internet use accelerates,” 30 January 2019. [Online]. Available: https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates.
[2] Soni Gulshan and Mala Kalra, “A novel approach for load balancing in cloud datacenter,” Advance Computing Conference (IACC), 2014 IEEE International, 2014.
[3] Y. Wen and C. Chang, "Load balancing job assignment for cluster-based cloud computing," 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 199-204, 2014.
[4] K. Verma, "Cloud Computing and its Types," Cloudkul, [Online]. Available: https://cloudkul.com/blog/what-is-cloud-computing/.
[5] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F. and Buyya, R., ""CloudSim: atoolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Journal of Software: Practice and Experience, vol. 41, pp. 23-50, 2010.
[6] D. C.Marinescu, Cloud Computing (Second edition), Elsevier, 2018.
[7] Bui Thanh Khiet, Nguyen Thi Nguyet Que, Ho Dac Hung, Pham Tran Vu, Tran Cong Hung, "A Fair VM Allocation for Cloud Computing based on Game Theory," Proceedings of the 10th National Conference on Fundamental and Applied Information Technology Research (FAIR'10), 2017.
[8] Bhaskar, R., Deepu, S.R., Shylaja, B.S., "Dynamic allocation method for efficient load balancing in virtual machines for cloud computing," Advanced Computing An International Journal, vol. 3, 2012.
[9] Rajwinder Kaur, Pawan Luthra, "Load Balancing in Cloud Computing," Recent Trends in Information, Telecommunication and Computing, Association of Computer Electronics and Electrical Engineers, pp. 374-381, 2014.
[10] Bhathiya Wickremasinghe and Assoc Prof and Rajkumar Buyya Contents, "Cloud Analyst: A CloudSim- based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments," 2010.
[11] D. Asir Antony Gnana Singh, R. Priyadharshini and E. Jebamalar Leavline, "Analysis of Cloud Environment Using CloudSim," in Springer.
[12] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," Second Symposium on Network Cloud Computing and Applications, 2012.
[13] Abhay Kumar Agarwal and Atul Raj, "A New Static Load Balancing Algorithm in Cloud Computing," International Journal of Computer Applications, vol. 132, p. 0975 – 8887, 2015.
[14] N.Swarnkar, A. K. Singh and Shankar, "A Survey of Load Balancing Technique in Cloud Computing," International Journal of Engineering Research & Technology(IJERT), vol. 2, pp. 800-804, 2013.
[15] S. S. Moharana, R. D. Ramesh and D. Powar, "Analysis of Load Balancers in Cloud Computing," International Journal of Computer Science and Engineering (IJCSE), vol. 2, pp. 101-108, 2013.
[16] Vikas Kumar, Shiva Prakash, "Modified Active Monitoring Load Balancing Algorithm in Cloud Computing Environment," International Journal for Scientific Research and Development (IJSRD), vol. 2, pp. 132-135, 2014.
[17] A. Makroo and D. Dahiya, "An efficient VM load balancer for Cloudm," in The 2014 International Conference on Applied Mathematics, Computational Science & Engineering (AMCSE 2014), Varna, 2014.
[18] Rakesh Kumar Mishra, Sreenu Naik Bhukya, "Service Broker Algorithm for CloudAnalyst," International Journal of Computer Science and Information Technology (IJCSIT), pp. 3957-3962, 2017.

[19] Er. Imtiyaz Ahmad , Er. Shakeel Ahmad, Er. Sourav Mirdha, "An Enhanced Throttled Load Balancing Approach for Cloud Environment," International Research Journal of Engineering and Technology (IRJET), 2017.
[20] Nguyen Xuan Phi, Tran Cong Hung, "LOAD BALANCING ARGORITHM TO IMPROVE REPSPONSE TIME ON CLOUD COMPUTING," International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2017.
[21] Nguyen Xuan Phi, Cao Trung Tin, Luu Nguyen Ky Thu, Tran Cong Hung, "Proposed Load Balancing Algorithm to Reduce Response time and Processing time on Cloud Computing," International Journal of Computer Networks & Communications (IJCNC), 2018.
[22] Gupta, S., Dixit, N., & Yadav, P., "An Advanced Throttled (ATH) Algorithm and Its Performance Analysis with Different Variants of Cloud Computing Load Balancing Algorithm," Communication, Networks and Computing, pp. 385-399, 2018.
[23] Tran Cong Hung, Phan Thanh Hy, Le Ngoc Hieu, Nguyen Xuan Phi, "MMSIA: Improved Max-Min Scheduling Algorithm for Load Balancing on Cloud Computing," Proceedings of The 3rd International Conference on Machine Learning and Soft Computing (CMLSC 2019), pp. 60-64, 2019.
[24] Moses, A. K., Joseph, A., Oluwaseun, O. R., Misra, S., & Emmanuel, A., "APPLICABILITY OF MMRR LOAD BALANCING ALGORITHM IN CLOUD COMPUTING," International Journal of Computer Mathematics: Computer Systems Theory, 2020.
[25] B. P. Mulla, C. Rama Krishna, and R. Kumar Tickoo, “Load balancing algorithm for efficient VM allocation in heterogeneous cloud,” International Journal of Computer Networks & Communications (IJCNC), vol. 12, no. 1, pp. 83–96, 2020.
[26] A. S. a. P. A. T. J. B. Durga Devi, “Modified adaptive neuro fuzzy inference system based load balancing for virtual machine with security in cloud computing environment,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 3, pp. 3869-3876, 2021.
[27] N. Z. J. a. A. A. D. A. Shafiq, “Load balancing techniques in cloud computing environment: A review,” Journal of King Saud University – Computer and Information Sciences, 2021.

AUTHORS

Tran Cong Hung was born in Vietnam in 1961. He received a B.E in electronic and Telecommunication engineering with first-class honors from HOCHIMINH University of technology in Vietnam, 1987. He received a B.E in informatics and computer engineering from HOCHIMINH University of technology in Vietnam, 1995. He received the Master of Engineering degree in telecommunications engineering course from the postgraduate department of Hanoi University of Technology in Vietnam, 1998. He received Ph.D. at Hanoi.

Hieu Le Ngoc has been working in the IT industry as an IT System Architect since 2010. As of now, he is working as an IT lecturer for HCMC Open University (Vietnam). His major study is about cloud computing and cloud efficiency for better service; minor studies are education, education in IT line, Languages Teaching (Chinese & English), Business and Economics.
More info *https://www.researchgate.net/profile/Hieu-Le-24*
.