

Contribution in Private Cloud Computing Development based on Study and KPI Analysis

Chafaa Hamrouni^{*1,2}, Slim Chaoui^{3,4}

¹Department of Computer Sciences, Khurma University College, Taif University, Khurma, 2935, Kingdom of Saudi Arabia

²Research Groups on Intelligent Machines Laboratory, National School of Engineering of Sfax (ENIS), Sfax University, Sfax, 3038, Tunisia

³College of Computer and Information Sciences, Department of Computer Engineering and Networks, Jouf University, Skaka, 72311, Kingdom of Saudi Arabia

⁴Unit- Laboratory of Sciences of Electronics, Technologies of Information and Telecommunications, Sfax University, Sfax, 3038, Tunisia

ARTICLE INFO

Article history:

Received: 26 April, 2020

Accepted: 26 June, 2020

Online: 12 July, 2020

Keywords:

QoS

Virtual Infrastructure Manager

Virtual Machine

Virtual Private Network

Quality of Service

ABSTRACT

The preset paper describes a contributes in the implementation of private Cloud Computing. Exhaustive study made validates Cloud environment creation and successful deployed Solution Configuration choice. Related solution ensures a good investment that reduces energy costs of the data center, minimizes network traffic cost proven as an NP-hard problem. In addition, each virtual machine (VsM) depends on energy within the data center. Used genetic algorithm allows the emergence of new generations, the use of Tabu search algorithm demonstrates a divergence state, however an identification technique, based on a hybrid algorithm, ensures which optimized virtual machine.

1. Introduction

In our research work, we illustrate the simulation tools, main functions in addition the private cloud domain [1]. A developed a solution to solve virtual machine placement problem and memetic algorithm with a costly function. An hybrid algorithm as a combination of genetic algorithm and the Tabou algorithm, and then an experimental development based on metaheuristic algorithms.

2. Design and implementation Step

The CloudSim simulation is divided into two main categories of classes: the classes that model the entities shown in blue as the Data Center, the Broker, the Cloudlet. Each class has its own specific functionality [2]. The first class is the Data Center that simulates the service kernel infrastructure that is offered by resource providers in a cloud computing environment. It encapsulates a set of computing machines that must be

homogeneous or heterogeneous with respect to their resource configurations [3]. While Data Center Broker represents the second class. It models the broker who is responsible for modeling between users and service providers according to customers' QoS conditions and deploys service tasks across the clouds. The third SAN Storage class provides a simple interface for accessing or storing data at any time.

Access to files in a runtime SAN causes additional delays in the required data files across the internal data center network. The fourth class is Virtual Machine [4]. It consists of modeling a virtual machine instance, whose management during its life cycle is a responsibility of the machine (Host). The fifth class Cloudlet models the core application services of Nails deployed in data centers. CloudSim consults the complexity of an application in terms of its IT needs [5]. Each application component has a predefined instruction size and a quantity of data transfer flows. The sixth class is BWProvisioner [6]. This is an abstract class that models the policy of supplying bandwidth to virtual machines that are deployed on a Host component. While the seventh class Memory Provisioner represents the provisioning policy for

*Chafaa Hamrouni, Taif University-Khurma University College-Department of Sciences Computer, Contact:00966546492338,Email:chafa.hamrouni.tn@ieee.org

assigning memory to virtual machines. If the Memory Provisioner component determines that the Host has the amount of free memory then the virtual machine can run and deploy to a Host. For the eighth VM Provisioner class, this abstract layer expresses the provisioning policy that a virtual machine engine uses to allocate VMs to hosts. For the last class VMM Allocation Policy [7] is an abstract implemented by a host component that models policies (shared space and time share) motivated to allocate processing power to virtual machines. Among the principles of Cloud Sim, it minimizes the number of entities in the system by using only the Data Center and the user as an inherited actuator of the Sim Java entities. This design method assists Cloud Sim in modeling a large-scale simulation environment on a machine with balanced processing capability. Virtual machines, provisioning policies, and Hosts are the key components of Cloud Sim that are instantiated as stand-alone, lightweight objects and do not compete with computing power. To take advantage of the benefits of virtualization [8], a private cloud operates on standardized architectures, so you must ensure that hosted applications are developed for these environments. Deploying a private cloud requires several steps:

-Step 1:

It consists in checking security processes proposed by the service provider (private cloud) or the IT department (internal cloud) in the event of a security breach.

-Step 2:

It consists in setting up an encrypted protection of the data.

-Step 3: It allows the measurement of the trust granted to the service provider (private cloud).

3. Design and implementation Step

The CloudSim simulation is divided into two main categories of classes: the classes that model the entities shown in blue as the Data Center, the Broker, the Cloudlet. Each class has its own specific functionality [2]. The first class is the Data Center that simulates the service kernel infrastructure that is offered by resource providers in a cloud computing environment. It encapsulates a set of computing machines that must be homogeneous or heterogeneous with respect to their resource configurations [3]. While DataCenterBroker represents the second class. It models the broker who is responsible for modeling between users and service providers according to customers' QoS conditions and deploys service tasks across the clouds. The third SAN Storage class provides a simple interface for accessing or storing data at any time.

Access to files in a runtime SAN causes additional delays in the required data files across the internal data center network. The fourth class is VirtualMachine [4]. It consists of modeling a virtual machine instance, whose management during its life cycle is a responsibility of the machine (Host). The fifth class Cloudlet models the core application services of Nails deployed in data centers. CloudSim consults the complexity of an application in terms of its IT needs [5]. Each application component has a predefined instruction size and a quantity of data transfer flows. The sixth class is BWProvisioner [6]. This is an abstract class that models the policy of supplying bandwidth to virtual machines that are deployed on a Host component. While the seventh class Memory Provisioner represents the provisioning policy for

assigning memory to virtual machines. If the Memory Provisioner component determines that the Host has the amount of free memory then the virtual machine can run and deploy to a Host. For the eighth VM Provisioner class, this abstract layer expresses the provisioning policy that a virtual machine engine uses to allocate VMs to hosts. For the last class VMM Allocation Policy [7] is an abstract implemented by a host component that models policies (shared space and time share) motivated to allocate processing power to virtual machines. Among the principles of Cloud Sim, it minimizes the number of entities in the system by using only the Data Center and the user as an inherited actuator of the Sim Java entities. This design method assists Cloud Sim in modeling a large-scale simulation environment on a machine with balanced processing capability. Virtual machines, provisioning policies, and Hosts are the key components of Cloud Sim that are instantiated as stand-alone, lightweight objects and do not compete with computing power. To take advantage of the benefits of virtualization [8], a private cloud operates on standardized architectures, so you must ensure that hosted applications are developed for these environments. Deploying a private cloud requires several steps:

-Step 1:

It consists in checking security processes proposed by the service provider (private cloud) or the IT department (internal cloud) in the event of a security breach.

-Step 2:

It consists in setting up an encrypted protection of the data.

-Step 3: It allows the measurement of the trust granted to the service provider (private cloud).

4. Implementation Step of the Meta- heuristic Algorithms

We introduce our method by identification of our data sets, then the parameters used for the different algorithms used as well as the results obtained. Finally, we will evaluate the algorithms used. Initial number of virtual machines is the most important parameter for creating an environment that can simulate a real bus. In that case, the genetic algorithm is considered an algorithm that treats a population and contains a set of generations. It begins by setting some parameters namely the size of the initial population P with which the genetic algorithm starts as well as the number of iterations. The number of iterations that correspond to the number of generations created is chosen arbitrarily. But, in our case, we have to apply this algorithm on a single generation that contains a number of individuals chosen systematically. Using the parameter of the genetic algorithm as previously stated for the virtual machine placement problem, the chromosome is arbitrarily entered by the system. Then, each one is converted into a binary number. The calculation of the optimization cost of each individual is necessary to achieve the best optimization. A list is then drawn up to follow the path of each individual on a chromosome. We proceed by a proportional selection that allows us to distinguish individuals with a large fitness. The result obtained by this procedure is subjected to a second selection by tournaments at one principle to make a draw with delivery of two individuals of population, and one makes them. The one with the highest fitness wins with a probability p between 0.5 and 1. This process is repeated n times in order to obtain the n individual population p that will serve as parents. The variance of this method is high and

increasing or decreasing the population value decreases or increases the selection effects [9]. We choose the first individual from the populations on the list to mark the one with the highest value. The result obtained is converted into binary in order to apply the two main methods of the genetic algorithm. Adopted method consists on:

- Choose individuals of the same size.
- compare the first gene of the first individual with the first gene of his binomial. This treatment is applied until the end of the individual list.
- Looking for crossing points when the value is different from one gene to another.
- Reverting the crossing requires a threshold that oscillates between 0.8 and 0.5. The list of the final cross depends on the method followed in paragraph.
- Its filling is carried out with the birth of two individuals C and D. the individual C contains the first part of the individual A and the last part of the individual B. As for the individual D, it contains the first part of the individual B and the second part of the individual A, based on the random crossing. One second method is applied after the crossing process consists of:

-Part of the crossing list obtained and choose two successive individuals.

-Compare the binary numbers of C and the numbers of D. If there is a change between the genes, the number of mutations increases but this number should not exceed 50%. The mutation method is applied once between two unique individuals. This result is stored in a list called mutation list.

During implementation of the memetic algorithm steps, we considered the memetic algorithm is a combination of the genetic algorithm and the taboo search algorithm, the algorithm uses the solution found in the genetic algorithm. We introduce the solutions in a list open only to each descending elements. We release each individual from minimum and place it in a new list. It contains the global and local minimums. We consider the minimum value of this list is exclusively global. Then the same number of individuals used in genetic algorithm for initial population and the choice of the size for the taboo list depends on each configuration given.

In addition, the search space size directly influences this parameter. If in a given configuration, multiple private Data Centers have free space for hosting one or more virtual machines, then the search space increases. In that case, more the movement possibilities numbers increase as the taboo list size, much more the taboo movements limited we have. It contains information about the latest taboo movements and this information includes the moved virtual machine, the old server that hosted it, and the new server. The performance of memetic algorithm is calculated in terms of network-traffic cost, it takes longer to be executed. Indeed, and each iteration, the taboo search algorithm is operated on each chromosome. Both genetic and taboo search algorithms represent the layout of virtual machines in private data centers, and

therefore, its size is relative to the number of VMs. Increasing the number of virtual machines therefore reduces the memetic algorithm operation time.

5. Experimental results

During genetic algorithm implementation, we obtained accepted result. We operate a local research algorithm based on Taboo research algorithm. Time execution increased slightly increases. Optimized virtual machines organization in distributed Data center is determined with accepted QoS. We studied the network traffic possibility, based on memetic algorithm. During execution, we present the Virtual machines number: 18, and to each virtual machine (annoyance) is affected an individual size randomly, that produces genetic algorithm action step. (Table.1 and Table.2).

Table 1: Virtual Machine conversion size list

Virtual Machine Conversion Size (Binary)	1	1	0	0	0	1	1	1	1	1	0	0	1	1
		0	0	0	0	0	1	0	1	1	0	1	1	1
		1	1	0	0		0	1	0	1	1			1
		1	0				0	1	0	1	0			0
		0	1				1	1	1	0	0			0
		1	0				1	1		1	0			1
			0				0	0			0			
							1	1						
Virtual Machine Conversion Size (Decimal)	1	4	2	0	0	3	2	4	5	6	3	1	1	1
		5	0				8	4	7	1	2			1

Table 2: Number of selected Virtual Machine with its size list

Virtual Machine Number	18									
Virtual Machine Size	1	6	7	3	3	2	8	8	7	
Virtual Machine Number										
Virtual Machine Size	2	2	6	6	6	6	8	2	7	

During application step operating, the system created from the virtual machines size list of the individuals according to its determined size (randomly), in a binary elements; then process to convert them subsequently into decimal. we apply the proportional selection method to fix the first integer of the list, and compare it with the other individuals, in order to select those which have the highest value in decimal, as a practice example, the maximums in decimal are: 45, 20, 2, 205, 189, 46, 3, 28, 44, 57, 61, 32, 114. Subsequently, individuals of the same size are selected to apply the main methods of the genetic algorithm. The crossing method is considered the basis of algorithm operation. We test the percentage of growth that does not exceed 80% compared to the percentage found between two individuals. If founded percentage is less than 80%, the crossover position between the two individuals will be randomly selected. Those beyond to the percentage is excluded. Result after this process will be getting two new one. In this case, the percentage is equal to 50% so we can apply this method to create a new element. We evaluate the used algorithms. So, for the memetic algorithm, it presents the best results for both small and large instances. The execution time of the memetic algorithm is high, explained by the intervention of the Taboo search algorithm on each individual in each generation. A taboo search algorithm greatly contributes to improvement the solution except that execution several times in the same iteration, slightly increases the processing time of the memetic algorithm. The compromise

between the execution time of the algorithm as well as the improvement of the solution obtained is.

6. Conclusion

Working on Data Center virtual machines mission are successfully accomplished, so infrastructure access facilitates virtual machines distribution and design operation. Identification and optimization of virtual machine number is based on genetic and taboo algorithms used for simulate by cloudsim. Mimetic Algorithm adjustment operations, that result minimization of virtual machines number and energy consumption, are validated. During the present research work, we successfully contribute in the implementation of private Cloud Computing. The study is approved per experimental tests in terms of Cloud environment creation as well as deployed Solution Configuration choice. Solution. We use genetic algorithm that allows the emergence of new generations, and Tabu search algorithm demonstrates an convergence state. Proposed heuristic approach contributes in virtual machines location and minimizes energy consumption.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The Authors would like to acknowledge the financial support of this work by Taif University- Khurma Univerity Collegue- Kidom of Saudi Arabia.

References

- [1] A. Khilla and I. Wolff, Field "Theory Treatment of H-Plane Waveguide Junction with Triangular Ferrite Post", *IEEE Transactions on Microwave Theory and Techniques*, **41**, 274-281, 1993. Doi:10.1109/TMTT.1978.1129365
- [2] N.S. Murthy and P. Srikanth Gautham, "Adaptive Equalizers in Smart Antenna Systems", *IETE Journal of Education*, **50**, 111-117, 2009. Doi: doi.org/10.1080/09747338.2009.10876058
- [3] A. H. Levy and A. S. Willsky, "An efficient maximum entropy technique for 2-D isotropic random fields", *IEEE Journals & Magazines*, **2**, 741_744, 1988. Doi: 10.1109/29.1589
- [4] P. Seong Chuah, C. Zhenzhong and Y. C. Peng Tan, "Energy Minimization for Wireless Video Transmissions With Deadline and Reliability Constraints", *IEEE Transactions on Circuits and Systems for Video Technology*, **23**, 467-481, 2013. Doi:10.1109/TCSVT.2012.2210655
- [5] M.Vesperini, F. Breon and D.Tanre, "Atmospheric water vapor content from space borne POLDER measurements", *IEEE Transactions on Geoscience and Remote Sensing*, **37**, 1613_1619, Doi: 10.1109/36.763275, Feb. 1999.
- [6] A.Verma, N.Srivastava and B. K. Kanaujia, "Analysis of Gunn Loaded Rectangular Microstrip Antenna", *IETE Journal of Education*, **53**, 80-87, 2012. Doi:10.1080/09747338.2012.10876096
- [7] K.L.Rao Member, "A Simplified Approach to the Antenna Problem", *IETE Journal of Education*, **23**, 7-8, 1982. Doi:1080/ 09747338.11450499
- [8] P.K. Singhal, P.j Singh Tomar and N.Verma, "Topologies and Applications of Meander Line Antenna", *IETE Journal of Education*, **48**, 121-131, 2007. Doi:10.1080/09747338.2007.11657877
- [9] Q. Yu, L.S. Yan, S. Lee, Y. Xie and A. E. Willner, "Loop synchro. polarization scrambling technique for the simulating polarization effects using recirculating fiber loops", in the *Journal of Lightwave Technology*, **21**, 1593-1600, 2003. Doi: 10.1109/JLT.2003.810103