Affinity based virtual machine migration (AVM) approach for effective placement in cloud environment

Gokuldhev M [1], Thanammal Indu V [2]

[1]Assistant Professor, Department of Computer science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R & D institute of Science and Technology, Chennai, India
[2]Assistant Professor, Department of Computer Science and Engineering, Amrita College of Engineering and Technology, Nagercoil, India

---

Abstract - *The VM placement problem has been studied widely based on various metrics like migration cost, communication cost, traffic rate in order to provide optimization in terms of network traffic, energy consumption and resource management etc. The cost between any two communicating VMs is defined as the number of networking devices (e.g., switches) on the end- to-end routing path of the VM pair between the source machine and the destination machine. The proposed AVM approach exploits a meta-heuristic optimization algorithm for VM placement and migration problem. The system proposes an advancement in the existing algorithm by reducing the search space and time required to find an optimal VM placement solution as it imposes non-randomness in the selection of initial population. Furthermore, mutation between the traffic aware and communication cost aware VM placement patterns ensures offspring that reduce the communication cost for VMs to reach the destination. In this system some communication-sensitive applications that involve frequent access to the links are identified to have an affinity or likeliness, and hence follows co-location VM placement pattern ensuring the resource availability constraint. So, the aim of this system is to group the VMs based on their affinity metrics like traffic rate and communication cost. By implementing this work, the communication cost is reduced by 15%. The execution time of the system has also reduced when compared to the existing genetic algorithm.*

Cloud Computing delivers enormous number of services to the users with unlimited scalable and reliable computing resources in a pay-as-you-go basis. Many organizations have setup their public cloud and other enterprises have set up their private clouds managed by frameworks like Open Stack, Open Nebula, VMware vCloud and Cloud-Stack [1]. The strategy for the selection of VM is a key step of multi-VM resource adjustment in a single physical machine (PM). Finding an optimal solution in polynomial time is not feasible since VM migration and server consolidation problems are mostly NP-complete optimization problems.

***Key Words*: Cloud computing, virtual machine migration, AVM approach, Affinity based grouping algorithm, multi-objective optimization, genetic algorithm**

1. INTRODUCTION

Genetic algorithms aim to find an approximation solution through multiple iterations, which are good methods for solving NP-complete problems. Hence, they can be efficiently used for the VM migration and server consolidation problems. The Traditional Genetic Algorithm does not evaluate or filter the initial population [2]. The search space and time to find the optimal placement pattern is higher if the initial population is not filtered. Hence, we use the affinity between the virtual machines to minimize the initial population and to group them together.

Based on this Matrix, if a VM-PM pair has higher affinity then the corresponding VM is placed in that PM, if the Resource requirements of that VM is satisfied by that PM. The allocation pattern is fed as one of the candidates of the initial population, and hence the optimal solution is reached faster based on this affinity-based solution pattern. Then, based on the traffic rate between the VMs after placement, VM migration takes place such that the overhead due to the migration is negligible when compared to the reduction in the traffic rate and communication cost due to migration [4].VM migration only considers dynamic consolidation, for which one or some of the hosts of VMs can be changed in response to the change in workloads. Swarm intelligence algorithms include such algorithms as simulated annealing, tabu search, iterated local search, GA, ACO, PSO, and ABC optimization.

The use of swarm intelligence algorithms to solve many combinatorial optimization problems has been a subject of research for a long time. They are widely used to solve NP-complete problems [5]. Other than the energy consumption and efficiency problems, networking and communication are two great challenges in cloud computing. The cost between any two communicating VMs is defined as the number of switches or hops on the routing path of the VM pair.

Communication-sensitive applications that involves frequent access to some particular links are identified to have an affinity or likeliness, and hence follows colocation VM placement pattern while ensuring their resource requirement constraint [6].

At present, the major objectives of the proposed VM placement algorithm are:

1) To place the Virtual Machines (VM) based on the degree of affinity with the physical machines.

2) To reduce the traffic rate upon virtual machines migration.

3) To reduce the communication cost for the virtual machine's placement.

There are various research challenges involved in this work. Since VM migration and server consolidation problems are mostly NP-complete optimization problems, finding an optimal solution in polynomial time is not feasible. As the number of VMs to be allocated increases, the search space and search time to find the near optimal solution also increases. Since we use genetic algorithm to implement the solution, it is essential to avoid the risk of being trapped in a local minimal solution.

Two or more VMs running the same communication sensitive application might have huge traffic rate prevailing between them. This traffic rate will definitely have an impact on the overall operational cost of the system [7]. The proposed system works in this scenario and helps to reduce the overall traffic rate and communication cost during VM Placement and Migration. There are certain nodes/PMs in a network of physical machines that the other VMs are highly interested in communicating with. These small number of cylindrical shaped high-capacity nodes are functionally different when compared to the rest of the nodes. Here we represent certain special high-capacity nodes that connect the physical machines of one network with the other as sink nodes. So, any VM that is to be placed in one of the ordinary PMs will have certain dependency with these sink nodes [3]. Thus, the major task is to allocate the VM to the PM which ensures minimal communication cost to reach the sink node that the VM is highly dependent. Physical machine is an underlying hardware resource, which can be logically used as similar partitions running different resources. Physical machine provides the resources for virtualization.

Virtualization plays an important role in the success of cloud computing by providing the essential infrastructure for cloud. Virtualization is an abstraction of the logical resources from their underlying physical resources by hiding their heterogeneity, geographical distributions etc. It abstracts the resources of the physical servers and provides multiple complete and isolated execution environments [5]. It is possible to run multiple applications in separate VMs having different configurations due to dynamic sharing and physical resource reconfiguration facilities provided by virtualization. It is possible to run multiple operating systems and multiple applications on the same machine/server at the same time, increasing the utilization and flexibility of the software. Resource utilization of the physical servers can also be improved by virtualization. As a whole, virtualization helps in improving resource utilization, energy management, dynamic resource sharing and provides availability, scalability and reliability of the cloud computing resources.

VM placement is a strategy to allocate the virtual machine to the corresponding physical machine such that, the overall requirements of the VM are satisfied by the PM. VM placement must also ensure that the placement strategy [4] enforces minimal operational cost, communication cost and traffic rate. VM migration helps the servers to be reconsolidated or reshuffled to reduce the operational costs like energy, communication cost, traffic demand and bandwidth demands [7]. Network communication consumes a large portion of the total operational cost of data centres. Hence, to reduce the traffic by reducing the communication cost, while satisfying the resource requirements of the VM and compensating the data transfer overhead due to migration to its minimum is a challenging task. Dependency between one or more VMs with a special placement pattern running a special application can be termed as affinity [3]. Affinity can be derived from the communication traffic or the data transmission traffic that are generated between the two VMs while running the applications. On simple terms, it is the relationship between the VMs. Fixed VM Placement pattern is used if, we need to place some VMs into a fixed PM. If the customer insists on placing their VM in a particular server, then this pattern is used.

If the same PM could accommodate two or more number of VMs, then it is collocation affinity [3]. If the affinity value between the PM-VM pair is positive (i.e., higher than a particular threshold), then that particular VM can be placed in that PM. So, all those VMs that exceed this positive threshold of affinity with the PM, which is satisfying their resource requirements can be placed in the same PM. If two VMs are placed in different PMs for reasons other than resource constraints, then it is no-colocation placement affinity. Dispersedly/separately placing the VMs in separate PMs [3] can be termed as No-colocation placement affinity.

When we deploy the VMs for communication sensitive applications, the sink nodes connect the VMs to different regions of the distributed cloud region. So, each VM will have a demand to each of the sink nodes of its region that connects itself to other cloud regions [8]. The demand for each of the VM to the sink nodes sums up to 1.0, to make it easy for comparison purpose. Thus, the PM that connects itself with the sink node that is highly demanded by the VM, will be selected for allocation of that VM.

Communication cost can be defined as the total number of networking devices (e.g., switches) between the host machines to the destination machine. Our aim is to reduce the communication cost to its minimum upon migration. Communication cost also has an impact on the overall traffic in the network. Let a, b be two physical machines, then Communication cost (a, b) α Traffic Rate. As the communication cost decreases, the traffic rate also decreases [10].

Cloud deployment models might include public cloud, community cloud, private cloud and hybrid cloud. While the public cloud provides services to everyone via internet, private clouds are managed internally by a single organization [1]. If two or organizations that have similar requirements join hands to deploy a cloud model, then it is a community cloud. Different types of cloud models are interconnected together to form a hybrid cloud model.

Traffic aware VM placement mainly concentrates on the bandwidth demands, network topology, dynamic traffic variation, communication patterns between the VMs during the initial VM placement. Such techniques mostly aim at minimizing the network utilization, while maximizing the resource utilization [8]. This project is based on this network –aware VM placement technique in which, the proposed system correlates the relationship between the average traffic/data rate and the end-to-end communication cost.

Network-aware VM placement always works in-par with most of the energy efficient means of VM placement. This technique aims at reducing the usage of servers by switching the state of idle servers to sleep mode and thereby reducing the power consumption [1]. It also concentrates on reducing the network communication cost after migration with special considerations to inter-VM dependencies.

The major goal of this resource-aware VM placement is to maximize the resource utilization. It mainly concentrates on resources like CPU, storage and RAM capacity [9]. But as for now, maximizing the resource utilization is being considered as a major requirement for any type of VM placement techniques.

In data-aware VM placement, the major emphasis is on optimizing the VM placement by maximum utilization of the inter-VM communication pattern in the network [11]. It also insists on paying extra attention to the dynamic traffic patterns and the changes in the network after VM Placement.

Rest of the report is organized as follows: Chapter 2 contains the literature survey that discusses the previous works and their scope in the domain of discussion. Chapter 3 gives a detailed understanding of the proposed system that is to be implemented with the system overview, overall system architecture and the algorithms. Chapter 4 discusses the implementation steps in detail, tools and techniques involved for implementation and performance and result analysis. Chapter 5 deals with the conclusion and future work associated with the project.

2. RELATED WORKS

2.1 VM Placement Overview

Mohammad Masdari et al. in [1] explain about the VM placement and migration in detail and in step-by step process. The VMs are being migrated to the opted physical machines based on the demand. In an intra-cloud VM placement, first the individual cloud member should be found to host the VM, which PM should host the VM. Some of the objectives factors is to choose a better physical machine to fit in are reducing intra-data center traffic, minimizing inter-data center traffic in federated cloud, preventing congestion in data center's network, distributing traffic evenly in data center's network, mitigating energy consumption, reducing cost for cloud providers and increasing the return on investment (ROI), increasing security, maximizing resource utilization, improving load balancing, high performance, locality: To increase accessibility, high reliability and availability, minimizing the number of active servers, reducing the number of active networking elements (switches), minimizing the SLA violation.

In each placement only some of the objectives are met based on the requirements and follows a placement algorithm to make a low overhead, efficient VM placement in cloud data centers. Most of the placement schemes refers to the existing algorithm for comparison of the proposed placement algorithm. Some of the algorithms are first fit, second fit, random fit, least full first and most full first.

Most of the proposed approaches in [1] to fulfil the objective use a three-tier in a datacenter cloud whose goal is to interconnect the servers with a high-speed link. It consists of a rack which are bottom level servers connected to either one or two Top-of-Rack (TOR) switches. The TOR switches are further connected to one or two switches in aggregation tier. Finally, each aggregation switch is connected with multiple switches in core tier.

2.2 Network and traffic aware VM Placement

Amir Rahimzad ehilkhechi et al. [2] focuses on specific objective – VM placement with reduced network and traffic in a special cloud from a provider's perspective. The performance of a VM strictly depends on the ability of the infrastructure to meet the traffic of VM. A satisfaction value is introduced for each VM and the appropriate physical machine is chosen to maximize the satisfaction value. The VM placement approach is based on different objectives such as initial placement, throughput maximization, consolidation and service level agreement satisfaction versus provider operating cost minimization.

There are two methods opted for solving this kind of objective based problems. One is to use mathematical models to define the problem and feed it into solvers operating. The second method is to use optimization tools such as Gurobi, GLPK etc., VM placement is also being classified as VM placement in single cloud environment and VM placement in multiple cloud environments. The placement is being done in a given set of physical machines with a set of demands encapsulated in VMs. These fluctuating demands are being placed by a placement controller. The placement controller decides how many instances to be gone to fulfil the objective. To fulfil the special objective the availability of VM and its profile and physical machine and its capacity are taken. The author gives an optimal placement algorithm that is offline provided the communication and flow demand profile of VM are given earlier.

2.3 VM Placement based on network topology and communication cost

The author deals on the network topology and network conditions are considered during the VM placement.

Some special nodes called sink nodes are being introduced which are functionally different from other physical nodes and has the high probability of VM interested to get communicated with these sink nodes. The VM that is dependent on the sink node requires a massive and end-to-end traffic. The efficient communication in terms of communication delay and data flow should be considered. The communication between the physical machines is negligible when compared to the communication between the physical machine and the sink nodes. The entire network is setup in the form of graph given by G(V, E) where E is the link between the nodes, V the set of nodes in graph G. S is the set of sink nodes in the network where S∈V. The VM with a demand for sink should not be placed in a physical machine that satisfies the demand but of high cost. The VM placement should be in such a way that the demand for the sink should be satisfied also with the low cost. Maximizing the satisfaction factor both the customers and the service providers will be in a win-win situation. The win-win situation occurs since the VMs will experience a better-quality service satisfies customer and the links getting less saturated satisfies the providers. Two different approaches are being introduced. Both greedy and heuristic based approaches are being introduced. Each of these approaches has different variants. The satisfactory factor also varies.

The simulation results of these algorithms are being tested using simulations and the result of these algorithms is found to be nearly optimal when compared to the VM placement regardless of their demands. Jianhai Chen et al [3] deals with joint affinity aware grouping and VM placement. Non-affinity aware grouping-based resource allocation towards general VM placement places the VMs into the same physical machine which are to be placed in distinct physical machines and vice versa. This leads to performance degradation of applications in different VMs deployed in multiple machines. A joint affinity aware problem is being defined and a joint affinity aware grouping and bin packing method is proposed to overcome the problem raised in non-Affinity aware VM placement. First the affinity relationships between the VMs are being calculated based on the relationships between the VMs such as collocated or dispersedly located and an affinity problem is being defined.

2.4 VM placement patterns

Then based on the affinity relationship between the VMs an affinity aware resource scheduling framework is being generated. The goal of resource allocation by Jian Hai [3] is to maximize the ratio between performance and cost and hence the data center revenues which involves both performance and energy efficiency. The energy and performance efficiency requires allocation to minimum number of physical machines and allocation to minimize the performance cost respectively. By co-locating the VMs which are required to be dispersedly located may require more resource in a single physical machine. This may lead to performance lag in the VM-physical machine pair. Dispersed location of the virtual machines which are to be co-located may increase the communication between the virtual machines. This may increase the data traffic and also the communication cost in the network. This is the same problem arising in random VM placement. The challenges put forth are to make detailed analysis of the VMs are to be placed in the appropriate physical machines so that the communication cost and the performance of the application running in the VM decreases and increases respectively. Also, a heuristic performance evaluation for the placed VMs is being done to check the quality of VM placement. It is also said that solving the resource scheduling and allocation is similar to the bin packing problem which is a NP-hard problem. So, making decision of the optimal VM placement becomes more challenging when performance maximization and communication cost reduction guarantee is also being considered. Generalization of dependency since current systems does not focus on the generalization of VMs. This method also helps to obtain and identify the relationship between VMs. The bin packing problem also may misplace the colocation placed and dispersedly placed VMs irrespective of their dependencies.

2.5 Virtual Machine Migration in an Over Committed Cloud

The virtual machines are being placed in the physical machines based on random generation or VM placement is done based on certain demands of the customer. This may also include the efficient performance of application present in different virtual machines from the provider perspective. After the placement scheme by Weizhe Zhang et al. [4] of virtual machines in different cloud environments, the communication between different VMs is being continued. Based on the data traffic which influences the communication cost and the performance efficiency of the applications, the VMs are being replaced to another physical machine so that the data traffic gets reduced. This reduction in amount of data transfer will definitely reduce the communication cost between the VMs and also increase the efficiency of the performance of application hosted in different virtual machines. This replacement of virtual machines after analyzing the data transfer between the virtual machines is called the migration of virtual machines.

VM migration reduces the operational cost as the servers are being reshuffled. According to method proposed by the author the demand for the traffic and bandwidth also accounts a bit in the total traffic of the system. VM migration also causes additional data transfer overhead. So VM migration should be in such a way that the data transfer overhead should be reduced. The communication cost and the performance factor should also be reduced compared to the previous placement strategy. A network aware problem is formulated into a non-deterministic polynomial time-complete problem. This study aims to reduce the communication cost. The communication is being reduced by taking into account the inherit dependencies among VMs which forms a multi-tier application and the topology of the physical machines. The migration is done so that a good relationship is maintained between the network communication and VM migration cost. The swarm intelligence algorithm is being used to maintain this trade-off between the network communication and the migration cost of the network. An approximate optimal solution is being found first. Then number of iterations is being performed.

From these iterations a good solution among the optimal solutions is chosen.

Artificial bee colony and Genetic algorithm is adopted and changed by the author to suit the VM migration problem. This change in adopted algorithm reduces the communication cost and network cost. Experimental results verify that network cost is reduced by genetic algorithm when the instances are low. When the number of instances is high, genetic algorithm does not work efficient. In such cases artificial bee colony algorithm reduces the network cost. So, when the number of instances is high artificial bee colony reduces the communication cost and when the instances are less, network cost is being reduced by genetic algorithm. It is also advantageous that the running time of artificial bee colony algorithm is half than that of running time for genetic algorithm. The most commonly used consolidation in VM migration is energy-aware VM migration. The aim of this energy-aware server consolidation is to improve the utilization of each server by consolidating as many numbers of VMs into servers. The remaining unused servers remain in low-power state. This method considers only the server-side constraints including the resources demanded. The network cost due to virtual machine migration is being overlooked. Network communication incurs a great part in the total operational cost. Many of the studies consider only one factor i.e., on migration cost, communication cost or energy consumption. The solution also solves the data center suffering from the long-lived congestion. Core network over-commitment and unbalanced workload causes this long-lived congestion. A comprehensive analysis of the above solution is made which is non-deterministic polynomial complete problem.

2.6 Dynamic Migration of Virtual Machines

The system by Fei Tao [5] also provides high security and isolation when required to the operating systems. Also encapsulates and customizes the applications running in the virtual machines. This also includes the support for legacy applications. Migration is usually done in cloud environment after the VM placement and also analyzing the data flow between the virtual machines placed in different physical machines. Based on the data flow between the VMs, to reduce the communication cost and to increase the performance efficiency of the application running in different virtual machines, migration of virtual machines is being carried out. The author solves the VM migration problem but in the dynamic mode. Based on the total operational cost occurring due to the data traffic in the integrated cloud environment the migration is done. Then this VM placement scheme is analyzed by allowing the flow of data between the VMs in the cloud environment.

To reduce the total operational cost further, the VM is being reshuffled to appropriate physical machines available in the system. This dynamic allocation of virtual machines to the physical machines enables the provider to choose a better placement scheme among the available optimal placement schemes. The inefficiency of the virtual machines is due to the reason that improper allocation of the virtual machines. This leads to imbalanced resource scheduling distribution. Thus, it is said that the proper use of VMs reduces the operating cost and the energy consumption by the system.

The efficiency of the virtual machines is being improved by employing three primary measures. The three primary measures are improving the framework and distribution, scheduling tasks and migrating VMs dynamically. Other than energy consumption and efficiency, networking and communication are the great challenges to be faced during the VM placement and dynamic VM migration. It is known that the data to be transferred to a virtual machine in other physical machine takes much time than the time taken by the VM to communicate with the virtual machine present within the same physical machine. This is because data requires less time to transfer in random memory than via network.

The solution is given in the form of an algorithm. This Non-dominated sorting genetic algorithm is being chosen since it solves multi-objective problems. This algorithm falls into the category of intelligence optimization algorithm. This algorithm can yield better results when some parts of the algorithms are made to be deterministic. The algorithm is proposed in such a way that it is more adaptive to the cloud. This algorithm when implemented produces a set of Pareto optimization solutions. The main contribution is split into three main categories. The multi-objective optimization algorithm contributes the following objectives. The dynamic migration of VMs is made efficient by proposing a triple-objective optimization model

3. AFFINITY AWARE VM MIGRATION (AVM)

3.1 Affinity based grouping

The proposed work deals with placing the virtual machines in the physical machines by reducing the traffic rate and communication cost upon migration. This problem can be considered as an optimization problem, since we have to reduce the traffic rate and communication cost while considering, whether the resource requirements of the VMs can be satisfied by the PMs. This work uses an AVM algorithmic approach to solve this optimization problem. When the applications are deployed on the VMs in a distributed cloud region, each application might have certain demand to those nodes which connects other VMs which also have similar computation environment in the network. So, the demand of each virtual machine to the sink nodes of that region is taken as the input for the initial VM placement algorithm. Based on this demand, the PM satisfying this demand with minimal communication cost is considered as the best option to allocate the VM. There exists an affinity between the VM and the PM that satisfies its demands. Hence this VM-PM pair is grouped together.

3.2 Affinity Calculation

This affinity can be calculated using the following equation

$$\text{Affinity}(vm_i, pm_j) = \sum_{s=1}^{n} \frac{\text{Demand}(vm,s)}{\text{static cost}(pm,s)}$$

where 's' represents the sink nodes.

The initial VM placement algorithm (3.5) begins here and it is used to find the best physical machine to host the virtual machine and the system proceeds to the VM migration algorithm (3.7). The result of this algorithm is the possible VM migration pattern that reduces the communication cost and traffic rate upon migration.

Fitness functions are used for calculating the communication cost and migration cost, which in turn is a minimization function. Mutation is performed between the resultant individuals of these fitness functions to minimize the communication cost and traffic rate. The population size is set to 100 individuals and the mutation rate is set to 0.1. We try to mutate between the VM allocations codes which are referred as bucket codes in the algorithm that are generated using traffic rate affinity and migration cost affinity. The resultant allocation minimizes the traffic rate, thereby reducing the communication cost.

Each VM has a specific demand for each of the sink nodes as specified in the Figure 1. The sum of the demands for each sink node is 1.0.

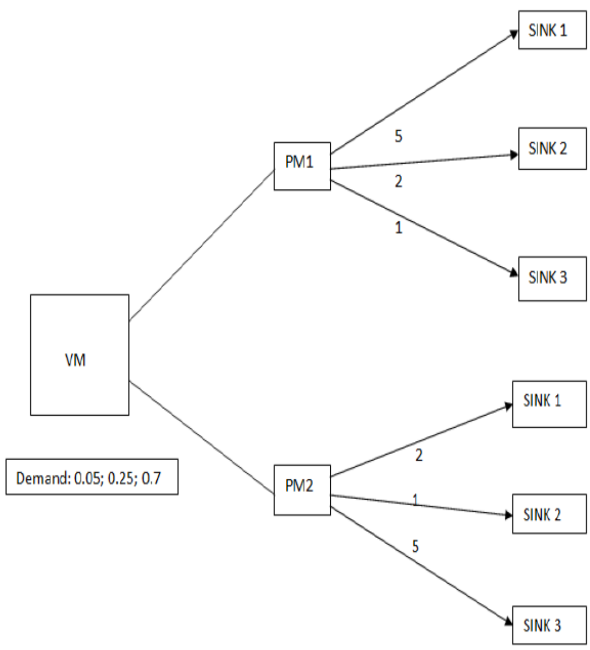


Fig -1: **Communication between sink node and VMs**

The three parameters of the demand vector in the fig 1 represent the demand of the virtual machine to any available sink node in the network to reach the physical machine. So, the demand of 0.7 represents the demand of sink 3 which is the highest to reach the expected physical machine. At the same time assume there are 2 physical machines available with same working environment the VMs will be placed to one that has a smaller number of hops to reach. Hence in the fig 1. PM 1 will be selected to place the VMs as it requires one hop compared to five hops for the PM 2.

The cloud users request to pay for the VMs, to run their applications in them. Every application might require different number of VMs for running. In case of communication sensitive applications, they have huge data/traffic rate between them. So, while deploying the VMs it must try to minimize the traffic rate. Affinity Generator in the architecture (Fig. 2) is responsible for framing all the affinity relations between the VMs while running the applications. The input to the affinity generator consists of the resource data of VMs, PMs and the workload data are received by the affinity generator from the cloud controller.

In order to record the detailed affinity information of the VMs, VM optimizer audits and stores the affinity relations. As a result, analysis of the gathered affinity data is done to decide whether there is affinity between the VM and PM or not.

Resource scheduling that is done based on the affinity is controlled and managed by the affinity scheduler. The scheduling requests from the cloud controller is checked by the affinity scheduler and it decides on the type of scheduling that is to be done (e.g., initial placement, migration, load balancing).

3.3 Proposed AVM Architecture

If a schedule request is received, it is immediately forwarded to the VM Optimizer to make VM placement/migration decisions. Final decisions regarding the VM placement, migration, boot or shutdown decisions are done by the VM optimizer. It correlates two fitness functions to decide upon the affinity grouping.

Based on the network topology, each of the physical machines is assigned a communication cost to each of the sink nodes of the network. In the proposed work a fully connected network topology is considered with 65 physical machines and 27 currently requesting virtual machine.

Based on this demand and communication cost, affinity between the VM and PM can be calculated using (1).

$$\text{Demand of sink} = \frac{\text{No. of sink nodes connecting VMs and PMs}}{\text{No. of VMs requesting PMs via each sink}} \quad (2)$$

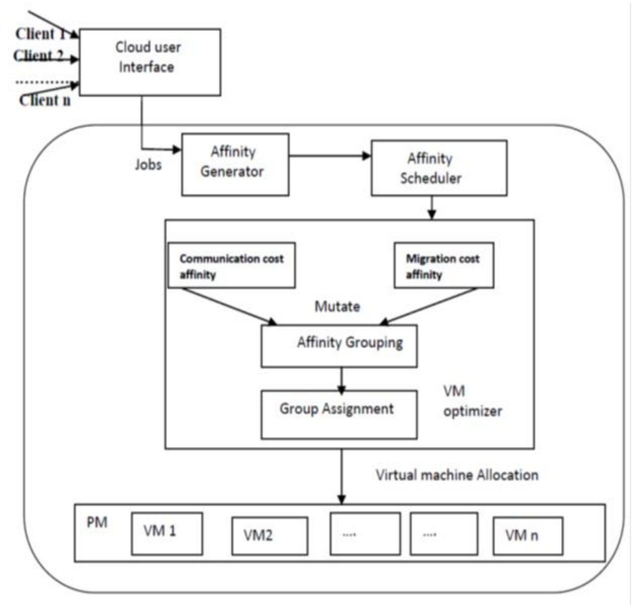


Fig -2: **Proposed AVM architecture**

Table. 1. Demand of the virtual machines for the Sink nodes

SINK / VM	SINK 1	SINK 2	SINK 3
VM 1	0.5	0.3	0.2
VM 2	0.4	0.2	0.4
VM 3	0.2	0.3	0.5
VM 4	0.4	0.4	0.2
VM 5	0.2	0.2	0.6

3.4 Communication Cost Matrix

Each physical machine will have a communication cost, that is calculated by the number of hops or networking devices like switches/routers. The following matrix represents the communication cost between each PM and Sink nodes.

Using this communication cost and demand in (Table 1), the affinity between a VM-PM pair is calculated using (1)

Based on this demand and communication cost, affinity between the VM and PM can be calculated using (1). The notations for the VM Placement Algorithm are described in the following Table 3.

Table 2. Communication Cost Matrix between PMs and Sink Nodes

Sink / PM	Sink1	Sink2	Sink3
PM 1	3	7	2
PM 2	4	5	5
PM 3	7	4	7
PM 4	5	8	4
PM 5	3	9	3

3.5 Algorithm for Initial VM placement

Inputs: VM set, PM set, Resource demand and availability vectors, Static cost between the PM- sink pair, demand of VM-sink pair.

Output: Solution matrix indicating the assignment of VMs to PMs.

Step-1 Declare initial affinity.

Step-2 for each vm_i in VM do

Step-3 for each pm_i in PM do

Step-4 for each s_i in S do

Step-5 Calculate using the formulae:

$$\text{Affinity}(vm_i, pm_j) = \sum_{s=1}^{n} \frac{\text{Demand}(vm, s)}{\text{static cost}(pm, s)}$$

Step-6 end for

Step-7 assign Aff[vm][pm]=affinity and reset affinity to zero.

Step-8 end for

Step-9 For each row in Aff[M][N] find the maximum affinity to a pm_i for Each vm_j.

Step-10 If RA[pm]>RD[vm] then assign X[vm][pm]=1

Table 3. Notations for Initial Vm Placement

ACRONYM	DESCRIPTION
M	Number of VMs
N	Number of PMs
VM Set	VM [1…….M]
PM Set	PM[1…….N]
m_i	i^{th} VM
pm_j	j^{th} PM
S	Number of Sink Nodes
Sink Node Set	S[1…….S]
s_i	i^{th} sink node
X[M][N]	Assignment matrix/Solution matrix
D[M][S]	Demand Vector for each VM-Sink pair
RA[N]	Resource Availability matrix for PMs
RD[M]	Resource Demand matrix for VMs
SC[N][S]	Static cost between PM-sink pair
DC[N][S]	Dynamic cost due to congestion in the network
Affinity[M][N]	Affinity between VM-PM pair
$fitness_{commn}$	Fitness calculation based on communication cost
$fitness_{mig}$	Fitness calculation based on migration cost.

Step-11 else Aff[vm][pm]=0, goto Step 9

Step 12 Display the solution matrix X[vm][pm]

3.6 Affinity Types

(A) Communication cost Affinity

The communication cost in this context can be described as the number of hops (Networking devices like routers or switches) between the source machine and the destination machine. The two virtual machines have communication cost affinity, if they run the same communication sensitive application, but with relatively higher communication cost. The following equation is the fitness function for calculating the communication cost, and the system minimizes it. The traffic rate is measured by some active or passive techniques of network traffic measurement depend on the environment. In this work it is measured using cloudsim traffic generator.

$$\text{Communication cost} = \sum_{i,j=1}^{m,n} \frac{\text{Static cost}(vm,pm) * \text{traffic rate}(pm,sink)}{(\max(\text{static cost}) * \max(\text{traffic rate}))} \quad (3)$$

(B) Migration Cost Affinity

Traffic rate is a dynamic indicator. It changes from time to time based on the type of applications that are running on the VM. Higher the traffic rate, higher is the operational cost. Hence, in order to minimize the traffic rate in par with reducing the communication cost, the traffic rate is considered as an important constraint, while calculating the migration cost. (3) is to calculate the fitness function (minimization) of migration cost affinity

$$\text{Migr. cost} = \sum_{i,j=1}^{m,n} \frac{\text{Size}(\text{vm to be migrated}) * \text{traffic rate}(\text{initial and current solutions})}{(\text{maxSize}(vm) * \max(\text{traffic rate}))} \quad (4)$$

VMs will have to be migrated only if, the traffic rate and communication cost will get reduced after migration. So, the migration cost is calculated by considering the size of the VM and the traffic rate between them in the network.

The VMs that show maximum affinity to a particular PM are grouped together. This group is then considered as a single unit, and is allocated to that PM, provided that the resource requirements are satisfied. Once the resource requirements of the particular unit of VMs are satisfied by the PM, then the VM Optimizer decides to allocate that particular set of VMs to that PM

3.7 Algorithm for VM migration

Population size=100

Mutation rate=0.1

Input: bucket code generated from algorithm 1

Output: near optimal bucket code for assignment

Step-1 Generate a randomized initial population of bucket codes.

Step-2 Add the bucket code generated from algorithm 1 to the randomized populations

Step-3 Calculate $fitness_{commn}$ (vm, pm) =

$$\sum_{i,j=1}^{m,n} \frac{\text{Static cost}(vm, pm) * \text{traffic rate}(pm, sink)}{(\max(\text{static cost}) * \max(\text{traffic rate}))}$$

Step-4 Calculate $fitness_{mig}$(vm, pm) =

$$\sum_{i,j=1}^{m,n} \frac{\text{Size}(\text{vm to be migrated}) * \text{traffic rate}(\text{initial and current solutions})}{(\text{maxSize}(vm) * \max(\text{traffic rate}))}$$

Step-5 Choose two individuals each with maximum $fitness_{commn}$, $fitness_{mig}$ and perform two-point crossover.

Step 6 Calculate the fitness of the resultant code, and check if it is better than the best fitness so far.

Step 7 If yes, then add it to the population else discard and move to step 5

Step 8 Stop if matrix is reached or the best fitness doesn't change over much iteration.

4. IMPLEMENTATION AND RESULT ANALYSIS

4.1 Tools and Techniques

The project implementation is done in NetBeans IDE, in which cloudsim 3.0 is installed to analyse the behaviour of the algorithm in real-time cloud environment. Cloudsim doesn't provide the GUI to visualize the result of the analysis in graphical format. So, Cloud Reports is used for GUI support.Net Beans is an Integrated Development Environment for developing Java applications and also Server and web related applications. Application which has database manipulations operations also developed with help this IDE. The Net Beans IDE is primarily used for development of Java, but also supports other languages, such as PHP, C / C++ and HTML 5.

The Net Beans Platform provides a framework for the development of interactive Java Swing real time applications. Net Beans plugins and NetBeans Platform based applications; no additional SDK is required. Applications can install plugins and private libraries depending upon the necessarily of the applications.

(A) CloudSim

Cloudsim toolkit is developed at Melbourne University by the GRIDS laboratory. It is a multi-layered simulation structure which includes modeling and simulation of data-center environments. It also provides good management of VM interfaces, memory, storage etc. It also induces scheduling and allocation policies for cloud platform. It has also been identified as the best tool for cloud computing since it allows the modeling of IaaS, PaaS and SaaS services.

(B) Cloud Reports

Cloud Reports is a GUI tool for the simulation of cloud computing in a distributed environment. Cloud Reports is the graphical format of Cloudsim which is user friendly, reliable and customizable. It also allows create new customers, datacenters, hosts with different bandwidth capacity, Resources like storage, memory MIPS etc.

It extends all the functionalities of CloudSim like VM allocation policies, power consumption and broker policies and other resource allocation and resource utilization models. Cloud Reports provide capability for user by setting the number of virtual machines each customer owns, a broker responsible for allocating these virtual machines and resource consumption algorithms.

4.2 Input and Output vectors

Each VM will have a demand for each of the sink nodes. The sum of the demands of a VM to all the sink nodes must be equal to 1.0. The matrix in the Table 3. represents the demand matrix between five VMs and 3 Sink nodes that are present in the scenario represented earlier. The resource requirement of the virtual machines (Table 4) must be satisfied by the physical machine in which, the VM is to be placed.

The two arrays represent the units of resource available in each PM and the units of resource required by each VM. Using the above matrix inputs, the affinity between each PM and VM can be calculated using the affinity formula, and the resultant matrix (Table 5) displays the affinity between the VMs and PMs within the scale of 0.00 to 1.00.for which 0 being the lowest and 1 being the highest.

Table 4. Resource Availability and Demand Vectors

Id / PM/VM	1	2	3	4	5
PM	20	5	5	10	25
VM	2	4	5	7	9

Table 5. Affinity Matrix between VMs and PMs

PM / VM	PM 1	PM 2	PM 3	PM 4	PM5
VM 1	0.27	0.22	0.19	0.24	0.00
VM 2	0.24	0.20	0.18	0.22	0.00
VM 3	0.21	0.21	0.17	0.27	0.00
VM 4	0.26	0.24	0.20	0.00	0.00
VM 5	0.20	0.19	0.16	0.24	0.27

The proposed system thus calculates the affinity between the VMs and PMs, using which the VMs having greater affinity to a particular PM will be grouped together to be placed in the particular PM.

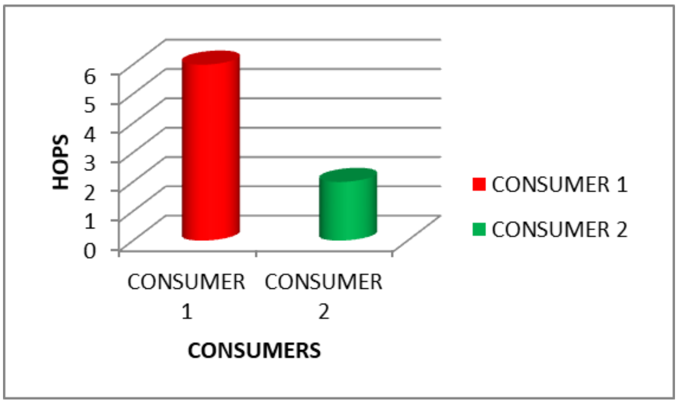
4.3 Performance Metrics

The performance of the optimisation problem can be calculated by the following two metrics:

- Traffic Rate
- Communication Cost

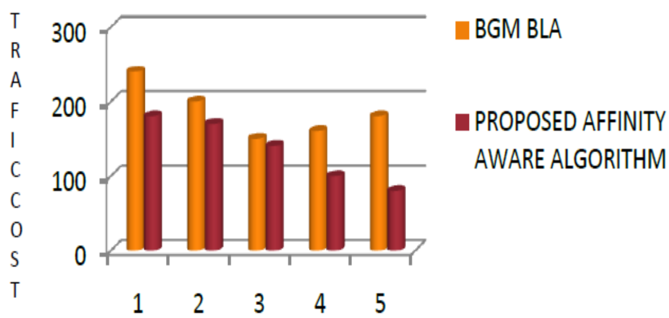
(A) Reduction in Communication Cost

The following bar chart represents the reduction in the communication cost after migration. By including the communication cost affinity, it reduces the communication cost by 15% while comparing with the existing system as specified in the bar chart in Figure 3. Here, Consumer1 uses BGM-BLA and Consumer 2 uses AVM Placement Solution.



(B) Reduction in the Traffic Rate

Since affinity aware grouping is used to group the VMs based on their demand and Communication cost, the traffic rate is observed to be reduced eventually when compared to the existing BGM-BLA algorithm. Even though the number of migrations increases, the traffic rate for the proposed system is seen to get reduce gradually.



material

No. Of migrations

Fig. 4. Traffic Rate Comparison

(C) Efficiency achieved in proposed AVM:

The corresponding graph measures the efficiency of the proposed method compared to the existing method. From the graph (Fig.5) it is inferred that, over the iterations with different datasets the proposed method out performs existing approaches in its efficiency which is measured by the time taken to complete the task even after the VM migration

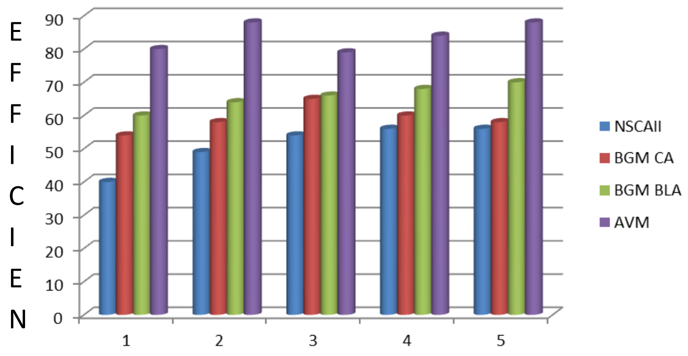


Fig. 5. Efficiency of the proposed AVM method comparison

5. CONCLUSION AND FUTURE WORK

The initial VM placement algorithm considers the affinity factor as an important constraint in grouping and placing the VMs. Some communication sensitive applications, when deployed in VMs with heavy traffic rate between them, incur high operational cost. This work defined an affinity matrix between the PMs and VMs based on the demand of the virtual machine to the corresponding high-capacity sink node of the region and the traffic rate between the VMs. The affinity is used between the virtual machines to minimize the initial population and to group them together. The affinity aware VM placement algorithm reduces the communication cost and traffic rate between the VMs that run communication sensitive applications upon migration. Since the proposed work reduce the search space and search time by introducing affinity aware grouping, the execution time required for running the algorithm also gets reduced. The system considers this as an optimization problem and proposes this network traffic-aware VM Placement Solution. The proposed system thus reduces the traffic rate and hence reducing the communication cost by 15%. There are multi-objective VM placement Solutions in this sector, that proposes solutions based on network traffic, topology, resource utilisation and energy efficiency. So, we can transform this optimization solution into a multi-objective one with more than two parameters to optimize. The Affinity aware VM Placement technique can further be improvised using machine learning algorithms which we planned to implement in future.

REFERENCES

[1] Mohammad Masdari, Sayyid Shahab Nabavi, Vafa Ahmadi, “An overview of virtual machine placement schemes in cloud Computing”, Journal of Network and Computer Applications 66, 2016

[2] Amir Rahimzadeh Ilkhechi , Ibrahim Korpeoglu, Ozgur Ulusoy, “Network-aware virtual machine placement in cloud data centers with multiple traffic-intensive components”, Computer Networks 91, 2016

[3] Weiwei Fang, Xiangmin Liang, Shengxin Li, Luca Chiaraviglio, Naixue Xiong, “VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers”, Computer Networks 57, 2013

[4] Jianhai Chen, Qinming He, Deshi Ye, Wenzhi Che, Yang Xiang, Kevin Chiew, Liangwei Zhu, “Joint affinity aware grouping and virtual machine placement”, Microprocessors and Microsystems, 2016

[5] Fei Tao, Chen Li, T. Warren Liao, and Yuanjun Laili, “BGM-BLA: A New Algorithm for Dynamic Migration of Virtual Machines in Cloud Computing”, IEEE transactions on services computing, Vol. 9, No. 6, November/December 2016

[6] [6] M. Aibinu, H. Bello Salau, Najeeb Arthur Rahman, M.N. Nwohu, C.M. Akachukwu, “A novel Clustering based Genetic Algorithm for route optimization”, Engineering Science and Technology, an International Journal 19, 2016

[7] Alvaro Garcia-Piquer, Jaume Bacardit, Albert Fornells, Elisabet Golobardes, “Scaling-up multi objective evolutionary clustering algorithms using stratification”, Pattern Recognition Letters, 2016

[8] Arash Chaghari, Mohammad-Reza Feizi-Derakhshi, Mohammad-Ali Balafar, “Fuzzy clustering based on Forest optimization Algorithm”, Journal of King Saud University – Computer and Information Sciences, 2016

[9] A.Azadeh, S. Elahi, M. Hosseinabadi Farahani, B. Nasirian, “A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transhipment”, Computers & Industrial Engineering, 2017

[10] Chao-Lung Yang, R. J. Kuo, Chia-Hsuan Chien, Nguyen Thi Phuong Quyen, “Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering”, Applied Soft Computing, 20

[11] Jiangtao Zhanga, Xuan Wanga, Hejiao Huanga, Shi Chena, “Clustering based virtual machines placement in distributed cloud computing”, Future Generation Computer Systems 66, 2017

[12] Juan Zamora, Marcelo Mendoza, Héctor Allende, “Hashing-based clustering in high dimensional data”, Expert Systems With Applications, 2016

[13] Satish Chander, P.Vijaya, Praveen Dhyani, "Multi kernel and dynamic fractional lion optimization algorithm for data clustering", Alexandria Engineering Journal, 2017

[14] Xiumin Wanga, Xiaoming Chenc, Chau Yuend, Weiwei Wue, Meng Zhangf, Cheng Zhang, "Delay-cost tradeoff for virtual machine migration in cloud data centers", Journal of Network and Computer Applications 78, 2017

[15] Yongqiang Gao, Haibing Guan, Zhengwei Qi, Yang Houb, Liang Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing", Journal of Computer and System Sciences 79, 2013

BIOGRAPHIES

Dr. Gokuldhev M is currently working as Assistant Professor, Department of Computer science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R & D institute of Science and Technology, Chennai

Thanammal Indu V is currently working as Assistant Professor, Department of Computer science and Engineering, Amrita college of Engineering and Technology, Nagercoil, India.