Optimal VM Placement Using Particle Swarm Optimization

Puvvada Nagesh, Dr.N.Srinivasu, N.Sai Rama Ranganadh, J.Jayanth, A.Naveen

Abstract: In a distributed cloud computing, virtual machine placement became more demanding due to the compensation between multiple objectives. Placing a virtual machine on a similar physical machine will automatically mitigate goals such as memory loss, power consumption and latency of the network. It may have a negative impact on some other goal to maximize an objective. A multi-objective optimization problem is to find a solution where optimality is achieved for all objectives. The Multiobjective Optimization Algorithms have the potential to converge towards pareto-optimal solutionswith the development of the solutions. In this paper, multi-objective Virtual Machine placement problem is inscribed using MOPSO algorithm.The proposed algorithm was implemented as well as comparing the algorithm to the current multi-objective algorithms.Statistically, the results show that the MOPSO does better job than existing algorithms in the Virtual Machime distributed cloud placement.

Keywords : Virtual Machine Placement,Multiobjective Optimization, MOPSO,Memory Wastage,Power Consumption

I. INTRODUCTION

Server virtualization is the process of dividing Physical Machine or Server into several virtual machine machine,which is independent of each other.Each virtual machine can have its own operating system.The purpose of virtualizing a server is to reduce the cost of hosting web service.It increase the availability of the servers,in addition to the efficient use of the resources.It is also helps in disastry recovery as well as in developing and testing the application.server virtualization is used to reduce the hardware requirement which results in cost reduction. Virtual machine placement is process of placing virtual machines (VM) in physical machine(PM) . How we place virtual machines in physical machine plays vital role in calculating power consumption, network delay and CPU resources .virtual machines has to be placed in such a that,it produces less power consumption, network delay and CPU resources. The process of minimizing multiple attributes in virtual machine placement is called multi-objective virtual machine placement. There are several types of algorithms available to optimally place virtual machines in server, such as Heuristic Algorithms, Genetic Algorithms, Bin Packing, Constraint Programming and Linear Programming. Ant Colony Optimization(ACO), Multi Objective Evolutionary Algorithm, Particle Swarm Optimization(PSO) are some of the genetic algorithms. MOPSO is one of the multi objective algorithm used to place virtual machines in server optimally. MOPSO does better job in placing virtual machines in server when compared to other algorithms. The detailed view about the algorithm is given in rest of the paper.

- *Puvvada Nagesh, Assistant Professor, KL University, Vaddeswaram, India. Email: pnagesh@kluniversity.in*
- *Dr.N.Srinivasu, Professor, KL University, Vaddeswaram, India. Email: srinivasu28@kluniversity.in*
- *N.Sai Rama Ranganadh,160030939,Student,KL university,sairamranganadh0986@gmail.com.*
- *A.Naveen, 160030050,Student,KL university.*
- *J.Jayanth, 160030514,Student,KL university.*

II. EXAMPLE OF VIRTUAL MACHINE PLACEMENT IN SERVERS

In IaaS cloud, user sends a request to the cloud provider to create a virtual machine with desirable specifications. A standard request involves parameters including CPU, memory, storage, and operating system. The decision to incorporate a virtual machine into the server is important because the resource wastage has a direct effect.If a virtual machine is installed on physical host overused then the host is unstable. Many VMs can run in a single physical machine in parallel using the technology called virtualization. Hypervisor is the software module that implements on top of the physical hardware system virtualizations. It depicts three virtual machines placed in two physical configuration servers (5 GHz CPU and 5GBRAM) in the cloud data centre.

Multi Object Virtual Machine Placement is

	Physical Machine 1(5Ghz CPU,5GB Ram)			Physical Machine 1(5Ghz CPU , 5GB Ram)	
Objective 1	VM1	VM2		VM3	
VM Utilization	1GHz CPU, 1 GB RAM	3GHz, 3 GB RAM		4 GHz CPU, 3GB RAM	
PM Resource Wastage	PM Wastage =Total-Utilized Resource (1GHz CPU,1GB RAM)			Total-Utilized Resource=PM Wastage (1GHz CPU,3GB RAM)	
	Cloud Resource Wastage =2GHz CPU,3GB RAM				
Objective 1,2	VM1	VM2		VM3	
VM Utilization	1GHz CPU, 1GB RAM	4GHz CPU,3GB RAM		3 GHz CPU,3GB RAM	
RW W.r.t Time (0-1 Hours)	PM Wastage=Total-Utilized Resource (No Wastage in CPU,1GB RAM)			PM Wastage=Total-Utilized Resource (1GHz CPU,2GB RAM)	
	Cloud Resource Wastage-2GHz CPU,3GB RAM				
RW W.r.t Time (1-4 Hours)	PM Wastage=Total-Utilized Resource (No Wastage in CPU,1GB RAM)			The Physical Machine is Powered off to save resource	
	Cloud Resource Wastage=No Wastage in CPU,1GB in RAM				

Illustration of the mapping of VM's with PM w.r.t. Objective

The three requests for a virtual machine have specific request parameters: VM 1 (1 GHz CPU, 1 GB RAM), VM 2 (3 GHz

CPU, 3 GB RAM), and VM 3 (4 GHz CPU, 3 GB RAM). Objective I to minimize the use of resources and thus the loss is calculated based on the physical server occupying the virtual machine. VM1 andVM2 are placed in PM1 while VM3 is placed in PM2 since PM1 cannot accommodate the VM3 request . The total waste of resources is the amount of resource wastage in PM1 and PM2 (2 GHz CPU and 3 GB RAM) combined.

III. PROBLEM FORMULATION

When virtual machines are placed in server, three attributes are considered namely CPU Utilization ,Power Consumption and Network Delay. These attributes are calculated based on the given below formulas.

3.1.1 Resource Wastage

CPU and RAM resources allocation for each virtual machine varies from each other.so the Mathematical Representation is given to calculate Resource Wastage

$$W_i = \frac{\left| L_i^{CPU} - L_i^{RAM} \right| + \varepsilon}{U_i^{CPU} + U_i^{RAM}}$$

where U_i^{CPU} and U_i^{RAM} are the total available CPU and RAM resources, respectively.L_i^{CPU} and L_i^{RAM} represent the amount of resources unused from usable CPU and RAM resource

3.1.2 Consumption of Power

If CPU Utilization is more then Power Consumption is more.When CPU is fully Utilizied it consumes 215 watts,while CPU is Idle it consumes 162 watts.The formula is stated below to find power consumption when CPU is partially Utilized

$$Power_i = \begin{cases} (P_i^{active} - P_i^{IDLE}) \times U_i^{CPU} + P_i^{IDLE}, & U_i > 0 \\ 0 & otherwise \end{cases}$$

3.1.3 Propagation time

Propagation time is known as the time taken for bits to reach from sender to receiver.It mainly depends on transmission medium. We find an effective wired physical medium for accessing data. The frequency for wired medium (optic fiber) data transmission is 2 * 108 m/s.The propagation time can be formulated as

$$PDelay = \frac{distance}{propagation\ time}$$

Distance is determined on the basis of the Euclidean distance between a VM and server.Servers are placed in ten different Geographical Locations. They are uniformly distributed across allocated regions. VM requests arise from random geographic regions. The distance between each server and VM is measured and tabulated on the basis of the server and VM coordinates as

$$DistMatrix = \begin{bmatrix} D_{1,1} & \dots & D_{J,1} \\ D_{2,1} & \dots & D_{J,2} \\ \dots & \dots & \dots \\ D_{i,1} & \dots & D_{i,j} \end{bmatrix}$$

The delay in propagation (PDelay) is estimated and saved as matrix below

$$PDelay = \begin{bmatrix} PD_{1,1} & \dots & PD_{J,1} \\ PD_{2,1} & \dots & PD_{J,2} \\ \dots & \dots & \dots \\ PD_{i,1} & \dots & PD_{i,j} \end{bmatrix}$$

3.2 Objective Formulation

If we are placing n VMs in m servers . Let Rpi be the CPU required for VM, Tpj is total CPU utilization available for each server, Rmi be the memory required for VM, and Tmj is total memory utilization available for each server. We employ two xij and yj binary variables. The xij variable indicates whether VM I is assigned to server j, and the yj variable indicates whether or not server j is in use. Our aim is to minimize power consumption and waste of resources at the same time. we can calculate them as follows

3.2.1 Minimizing Resourse Wastage

Minimize

$$\sum_{j=1}^{m} W_j = \sum_{j=1}^{m} \left[y_j \times \frac{\left| \left(T_{p_j} - \sum_{i=1}^{n} (x_{ij} \cdot R_{pj})\right) - \left(T_{mj} - \sum_{i=1}^{n} (x_{ij} \cdot R_{mj})\right) \right| + \varepsilon}{\sum_{i=1}^{n} (x_{ij} \cdot R_{pi}) + \sum_{i=1}^{n} (x_{ij} \cdot R_{mi})} \right]$$

$$X_{ij} = \begin{cases} 1, & if\ VM_j\ allocated\ to\ DC_i \\ o, & otherwise \end{cases}$$

$$Y_i = \begin{cases} 1, & if\ DC_i\ is\ used \\ o, & otherwise \end{cases}$$

where R_{Mj} and R_{pj} are the memory and CPU required for VM and T_{Mi} and T_{Pi} are total availability of memory and CPU.

3.1.2 Minimizing Power Consumption

Minimize

$$\sum_{j=1}^{m} p_j = \sum_{j=1}^{m} \left[y_j \times \left(\left(p_j^{busy} - p_j^{idle}\right) \times \sum_{i=1}^{n} \left(x_{ij} \cdot R_{pi}\right) + p_j^{idle} \right) \right]$$

where P_i^{active} and P_i^{idle} represents Average power value when using a CPU.

3.2.3 Minimizing Propagation Delay

$$\text{Min}\sum_{n=1}^{N} PDelay_{i,j}^{\ n} \quad \forall i \in M, \forall j \in N$$

Where PDelay represents the propagation dalay between I and j servers.

IV. Multi-objective Particle Swarm Optimization (MOPSO) Algorithm

In PSO,Particle means each element in a group is called particle. Collection of something that move somewhere in large numbers is called Swarm. In PSO, Birds look for food in the field at random. Birds don't know where the food is, but in every iteration they know how far the food is. The best strategy here is to track birds closest to food. Each bird contains two attributes pbest & gbest. Pbest=best solution (fitness)it has achieved so far.

Gbest=best value that is tracked by the PSO,best value is global best.

Once two best values have been found, the particle updates its velocity and position.

V[ ] = v[ ] + C1 * rand( ) * (pbest[ ] – present[ ]) + C2 * rand( ) * (gbest[ ] –present [ ])

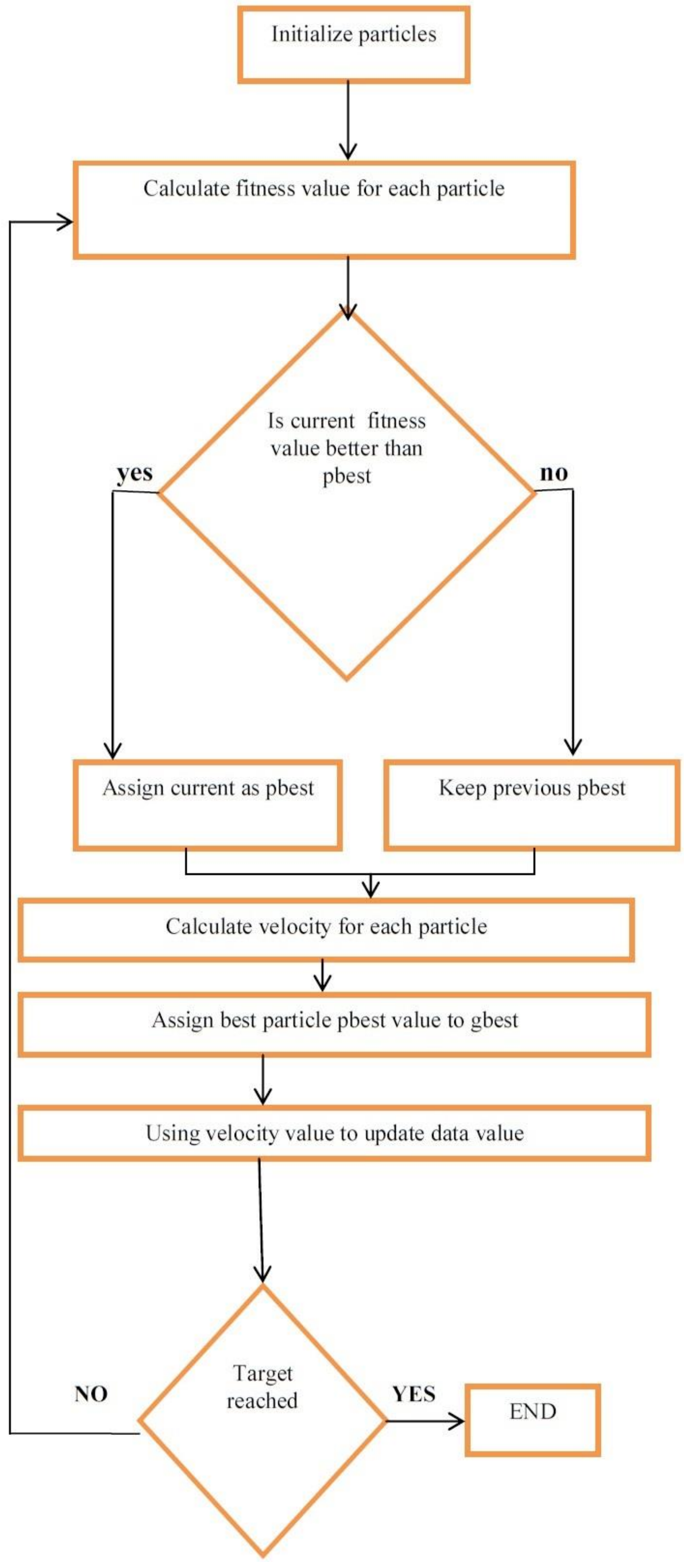
Present[ ] = present[ ] + v[ ]

Where C1,C2 are learning factors

The PSO algorithm for VM placement

```
PSO( VM_REQ_LIST[], tmax , N)
//VMs_REQ_LIST:demands VMs
//PMslist:Active PMS
//tmax:maximum iteration number
//N:number of particles
//p:population
//v:particle velocity
//x:particle position
FOR i=1 to n do
   p[i]<-Generate_initial_population(VMs_Req_List[],PMslist[]);
   xi<-p[i];
   FOR j=1 to PMslist t do
     if p[i][j]!=empty
        v[i][j]<-1;
     else  v[i][j]<- 0;
   END
   XIbest<-Xi;
   if f(XIbest)<f(Xgbest) then//evaluate the fitness value
     Xgbest<-XIbest;
END
FOR t=0 to tmax do
 FOR i=1 to N do
     Update Vi(t+1);
     Update Xi(t+1);
     if(Xi(t+1)) > f(XIbest(t)) then
        XIbesti(t+1) <-Xi(t+1);
     if (Xibesti(t) > f(Xgbest(t))then
        Xgbest(t+1) <-Xibesti(t);
 END
END
Return gbest;
```

FLOW CHART FOR PSO



V. PERFORMANCE EVOLUTION

The performance of the algorithm proposed is compared with other genetic algorithms. all the algorithms are executed in system with same specifications.

A random CPU (R_{CPU}) and RAM (R_{RAM}) for 100 and 200 VMs is generated by using correlation. A linear correlation coefficient for resource instance generation is rendered for uniform server distribution for both RAM and CPU usage. P-value is used for tuning purposes to control the probability of correlation coefficients. For each instance , VM request, two reference values and five correlation coefficients were used in the design. Two reference values include R_{CPU} = R_{RAM} = 25% and 45%, respectively. Maximum resource a VM can access from a server is 90%. Based on server access limitations, the generated instances are requested to remain within the resource limit. For all reference values the probability value is set at 0, 0.25, 0.50, 0.75 and 1.For reference value R_{CPU} = R_{RAM} = 25%, when setting the P value as above, the average linear correlation coefficients were obtained as - 0.760, - 0.326, - 0.045, 0.24, 0.742 for each set of instances. For R_{CPU} = R_{RAM} = 45%, it is - 0.746, - 0.372, - 0.049, 0.326, 0.762 for each set of instances.

VI. RESULT AND DISCUSSION

Experimental Results of R_{CPU} = R_{RAM} = 25% for 100 VM Instances

CORR	ALG	Performance Measures w.r.t Objective Functions		
		Wastage	Power	Delay
Strong Negative	PSO	4.6401	7.24E+03	37.279
	ACO	6.2579	7.24E+03	41.171
	MOPSO	4.6259	7.07E+03	45.519
	MOACO	7.8312	7.57E+03	42.178
Negative	PSO	4.4659	7.14E+03	35.978
	ACO	6.1814	7.63E+03	38.158
	MOPSO	4.6455	7.30E+03	43.756
	MOACO	6.4307	7.63E+03	41.499
Zero	PSO	3.791	7.33E+03	37.147
	ACO	4.9941	7.66E+03	41.52
	MOPSO	3.5477	7.33E+03	44.429
	MOACO	6.2627	7.66E+03	42.129
Positive	PSO	2.7832	6.88E+03	36.771
	ACO	3.6863	7.21E+03	40.436
	MOPSO	3.1695	6.88E+03	43.486
	MOACO	3.8029	7.05E+03	42.468
Strong Positive	PSO	2.2179	7.05E+03	36.922
	ACO	2.6337	7.37E+03	40.356
	MOPSO	2.2043	7.21E+03	43.287
	MOACO	2.3958	7.37E+03	40.715

VII. CONCLUSION

Virtual Machine Placement is one of Cloud Computing's main isuues. The way we place virtual machines in servers plays a major role in minimizing the CPU utilization, power consumption and network delay. The proposed results of the algorithm were tested and compared to existing algorithms. The results show that the algorithm proposed is better than the existing algorithms when placing virtual machines in servers.

VIII. REFERENCES

[1] Ankit Anand, J Lakshmi,SK Nandy" Virtual Machine Placement optimization supporting performance SLAs" IEEE International Conference on IEEE,2013.

[2] Chao Liu , Chenyang Shen ,Sitian Li , Sinong Wang" A New Evolutionary Multi-Objective Algorithm to Virtual Machine Placement in Virtualized Data Center" IEEE International Conference on IEEE,2014.

[3] Divya Bharathi P, Prakash P, Vamsee Krishna Kiran M" Virtual Machine Placement Strategies in Cloud Computing" IEEE International Conference on IEEE,2017.

[4] Foudil Abdessamia,Yu Tai, WeiZhe Zhang , Muhammad Shafiq" An Improved Particle Swarm Optimization For Energy-Efficiency Virtual Machine Placement" IEEE International Conference on IEEE,2017.

[5] Chuangen Gao, Hua Wang, Linbo Zhai, Yanqing Gao, Shanwen Yi" An energy-aware ant colony algorithm for networkaware virtual machine placement in cloud computing" IEEE International Conference on IEEE,2016.

[6] Xiao-Fang Liu, Zhi-Hui Zhan, Jeremiah D. Deng" An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing" " IEEE International Conference on IEEE,2018.

[7] Shahram Jamali, Sepideh Malektaji" Improving Grouping Genetic Algorithm for Virtual Machine Placement in Cloud Data Centers" IEEE International Conference on IEEE,2014.

[8] Songtai Dai,Ao Zhou,shangguang wang"The performance evolution of virtual machine placement based webcloudsim" IEEE International Conference on IEEE,2018.

[9] Abdelkhalik Mosa and Rizos Sakellariou" Dynamic Virtual Machine Placement Considering CPU and Memory Resource Requirements " IEEE International Conference on IEEE,2019.

[10] Priyanka C.P , Sankari Subbiah" Comparative Analysis on Virtual Machine Assignment Algorithms" IEEE International Conference on IEEE,2017.

[11] Xiao-Ke Li, Chun-Hua Gu, Ze-Ping Yang, Yao-Hui Chang" Virtual Machine Placement Strategy based on discrete Firefly algorithm in cloud environments" IEEE International Conference on IEEE,2015.

[12] Yifan Ding, Guangzhong Liao1, Siyuan Liu1" Virtual Machine Placement Based On Degradation Factor Ant Colony Algorithm" IEEE International Conference on IEEE,2018

[13] Milad Seddigh, Hassan Taheri,Saeed Sharifian" Dynamic prediction scheduling for virtual machine placement via ant colony optimi"ation" IEEE International Conference on IEEE,2015.

[14] Dan Liu, Xin Sui" An Energy-efficient Virtual Machine Placement Algorithm in Cloud Data Center" IEEE International Conference on IEEE,2016.

[15] Lanka, S., Madhavim, R., Abusahmin, B. S., Puvvada, N., & Lakshminarayana, V. (2017). Predictive data mining techniques for management of high dimensional big-data. Journal of Industrial Pollution Control, 33, 1430-1436. Retrieved from www.scopus.com

[16] Madhavi, R., Karri, R. R., Sankar, D. S., Nagesh, P., & Lakshminarayana, V. (2017). Nature inspired techniques to solve complex engineering problems. Journal of Industrial Pollution Control, 33(1), 1304-1311. Retrieved from www.scopus.com

[17] Puvvada, N., & Prasad Babu, M. S. (2018). Semantic web based banana expert system. International Journal of Mechanical and Production Engineering Research and Development, 8(3), 364-371. Retrieved from www.scopus.com

[18] Radha Madhavi, M., Nalleboyina, V., & Nagesh, P. (2019). Influence of magnetic field, heat radiation and external surface temperature on nanofluids with different base fluids in mixed convective flows over a vertical circular cylinder. International Journal of Innovative Technology and Exploring Engineering, 8(5), 497-504. Retrieved from www.scopus.com

[19] Rajesh, B., Nagesh, P., Gowtham, K., Vivek, G., & Srinivasu, N. (2019). A new scheme to safeguard data for cloud integrated internet things. International Journal of Innovative Technology and Exploring Engineering, 8(4), 174-178. Retrieved from www.scopus.com

[20] Venkata Ramana, N., Kolli, C. S., Ravi Kumar, T., & Nagesh, P. (2019). Hybrid K-mir algorithm to predict type of lung cancer among stoicism. International Journal of Innovative Technology and Exploring Engineering, 8(4), 283-287. Retrieved from www.scopus.com

[21] Venkata Ramana, N., Nagesh, P., Lanka, S., & Karri, R. R. (2019). Big data analytics and iot gadgets for tech savvy cities doi:10.1007/978-3-030-03302-6_12 Retrieved from www.scopus.com

[22] Venkata Ramana, N., Seravana Kumar, P. V. M., & Nagesh, P. (2018). Analytic architecture to overcome real time traffic control as an intelligent transportation system using big data. International Journal of Engineering and Technology(UAE), 7(2.18 Special Issue 18), 7-11. Retrieved from www.scopus.com

[23] Venkataramana, N., Nagesh, P., Seravana Kumar, P. V. M., & Vignesh, U. (2018). IoT based scientific design to conquer constant movement control as a canny transportation framework utilizing huge information available in cloud networks. Journal of Advanced Research in Dynamical and Control Systems, 10(7 Special Issue), 1395-1402.