International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12 Number 22 (2017) pp. 12044-12051

Bi-Objective Virtual Machine Placement using Hybrid of Genetic Algorithm and Particle Swarm Optimization in Cloud Data Center

Dilip Kumar[1] and Dr. Tarni Mandal[2]

[1]Assistant Professor, Department of Computer Science and Engineering,
[2]Associate Professor, Department of Mathematics,
[1,2]National Institute of Technology, Jamshedpur, Jharkhand, India.
Orcid Id: [1](0000-0003-2834-6479), [2](0000-0002-2563-4494)

Abstract

Efficient resource management through the virtual machine placement (VMP) is a great concern in data centers. The Bi-objective VPM is a representation of multi-objective combinatorial optimization problem. Energy or cost minimization of cloud data center is highly dependent upon the VMP policy. Allocating the set of virtual machines (VMs) to the set of suitable physical machines (PMs), while considering the cost, CPU utilization, number of active servers and energy consumption of cloud computing, defines the VMP process. In this paper, a cloud model simulated with evolutionary algorithms (genetic algorithm (GA), Particle Swarm Optimization (PSO), and hybrid GA-PSO (HGAPSO)) for the suitable VMP with the objectives of minimizing Energy consumption, and number of active servers, while considering the CPU utilization, RAM, network bandwidth etc. The HGAPSO produced the optimum result and outperformed the other two algorithms.

Keywords: Cloud Computing, Virtualization, Virtual Machine, Multi-Objective, Genetic Algorithm, Particle Swarm Optimization, Meta-heuristics, Energy Aware Computing.

INTRODUCTION

Cloud computing is based on the concept of dynamic provisioning, which is applied to services, computing capability, storage, networking, and information technology infrastructure to meet user requirements. The resources are made available for the users through the Internet and offered on a pay-as-use basis from different Cloud computing vendors [1]. Technology, that makes cloud computing feasible are virtualization, cyber-infrastructure, and service orient infrastructure [2]. It offers scalable and elastic computing and storage services. Cloud users can access computing resources without having to own, manage, and maintain them. There are three common cloud computing models known as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [1], [3]. The core technology in cloud computing is virtualization [4], which separates resources and services from the underlying physical delivery environment. The resources of a single physical machine (PM) are sliced into multiple isolated execution environments for multiple virtual machines (VMs). Virtual Machine Placement (VMP) is an important topic in cloud environment virtualization, in particular in IaaS model. VMP maps a set of virtual machines to a set of physical machines. For the cloud providers, a good VMP solution should maximize resource utilization and minimize power consumption [5]. Through virtualization, a cloud provider can ensure the quality of service (QoS) delivered to the users while achieving a high server utilization and energy efficiency. This research aims to schedule VM placement in a data center by developing a multi-objective mathematical model that minimizes the energy consumptions and number of active physical hosts. The model is simulated and validated with various set of VM request for all three algorithms (GA, PSO and HGAPSO).

RELATED WORK

Several researchers are addressed the importance of placing VMs appropriately [6-12]. Vogels[13] quoted the benefit of packing VMs efficiently in server consolidation. Heuristic techniques and evolutionary algorithms (EA) are the common optimization methods used to solve a multi-task scheduling problem [14-17]. However, some researchers [18-21], have identified several limitations of traditional mathematical programming approaches to solve MOPs. Some of them are the following:

- We need to run many times those algorithms to find several elements of the Pareto optimal set.
- Many of them require domain knowledge about the problem to be solved.
- Some of those algorithms are sensitive to the shape or continuity of the Pareto front.

These complexities call for alternative approaches to deal with certain types of MOPs. Among these alternative approaches, we can find Evolutionary Algorithms (EAs), which are stochastic search and optimization methods that simulate the natural evolution process. At the end of 1960s, Rosenberg [22] proposed the use of genetic algorithms to solve MOPs. However, it was until 1984, when David Schaffer [23] introduced the first actual implementation of what it is now

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12 Number 22 (2017) pp. 12044-12051

called a Multi-Objective Evolutionary Algorithm (MOEA). From that moment on, many researchers [4], [24-28] have proposed a wide variety of MOEAs.

PSO is a population based stochastic technique inspired by social behavior of bird flocking or fish schooling. Extensive reviews on PSO algorithm development can be found in [29-31]. Gao proposed a method called Selectively informed PSO (SIPSO) to allow the particles to learn at difference strategies based on their connections [32].

SYSTEM MODEL

In a cloud environment, we have a set of PMs with applications running on them. The Data Center is fully virtualized and all the applications are running on VMs. The problem of a set of VM to be place on a suitable pool of PMs is related to the multidimensional vector packing problems. Dimensions in the packing problem are various types of resource utilizations.

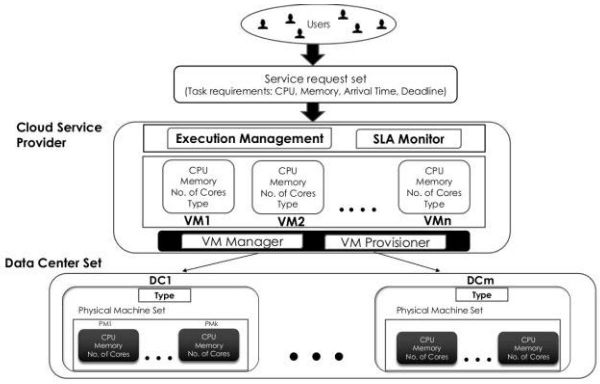

Fig. 1: Cloud Computing Architecture

In our work, we used two dimensions to characterize a VM and a server node CPU and memory. If two VMs are running on the same server, the CPU utilization of the PM is estimated as the sum of the CPU utilizations of the two VMs. To prevent CPU and memory usage of a PM from reaching 100%, we have to impose an upper bound on resource utilization of a single PM with some threshold value. The main idea behind this is that 100% utilization can cause severe performance degradation and VM live migration technology consumes some amount of CPU processing capability on the migrating node. Fig. **1** shows the Cloud Infrastructure of the system. Virtualization allow the creation of multiple virtual machines on any of the available PM. PM consumes energy in an idle state to perform maintenance functions and denoted as P_{min}. PM consumes more energy as per utilization of the CPU by the VMs. PM consumes maximum energy at the pick level and denoted as P_{max}.

Energy Consumption Modeling

The power consumption of servers can be accurately described by a linear relationship between the power consumption and CPU utilization [33, 34]. PMs are turned off when they are idle to save the energy consumptions. Hence, their idle power is not part of the total energy consumption. Finally, we defined the power consumption of the i^{th} server as a function of the CPU utilization as shown in Eq.(1)**.**

$$E_i(\tau) = \begin{cases} (P_{max} - P_{min}) * \dfrac{U_i(\tau)}{100} + P_{min}, & U_i(\tau) > 0 \\ 0, & otherwise \end{cases} \qquad (1)$$

Where the $U_i(\tau)$ represent the CPU utilization of i^{th} host server R_i at time τ. P_{max} and P_{min} are the power consumption at maximum utilization and at idle respectively.

Resource Wastage Modeling

The remaining resources available on each server may vary greatly with different VM placement solutions. To fully utilize multidimensional resources, the following equation is used to calculate the potential cost of wasted resources:

$$W_i = \frac{\left|L_i(p) - L_i(m)\right| + \varepsilon}{U_i(p) + U_i(m)} \qquad (2)$$

Where W_i denotes the resource wastage of the i^{th} server, $U_i(p)$ and $U_i(m)$ represent the normalized CPU and memory resource usage (i.e., the ratio of used resource to total resource). $L_i(p)$ and $L_i(m)$ represent the normalized remaining CPU and memory resource. ε is a very small positive real number (known as wastage error = 0.0001).

Optimization Formulation

Suppose ***m*** number of VMs that are to be placed on ***n*** servers. Let R_{pj} be CPU demand of each VM, T_{pi} be the threshold of CPU utilization associated with each server, R_{mj} be the memory demand of each VM, and T_{mi} be the threshold of memory utilization associated with each server. We use two binary variables x_{ij} and y_i . The binary variable x_{ij} indicates if VM_j is assigned to server i and the binary variable y_i indicates whether server i is in use or not. The placement problem can be formulated [35] as Eq. (**3)-(8)** :

$$Min \sum_{i=0}^{n} E_i = \sum_{n=1}^{\infty}\left[y_i * \left(\left(p_i^{max} - p_i^{min}\right) * \sum_{i=1}^{n}\left(x_{ij} * R_{p_j}\right) + p_i^{min}\right)\right] \qquad (3)$$

$$Min \sum_{i=1}^{n} W_i = \sum_{i=0}^{n}\left[y_i * \frac{\left|\left(T_{p_i} - \sum_{j=1}^{m}\left(x_{ij} * R_{p_j}\right)\right) - \left(T_{ni} - \sum_{j=1}^{m}\left(x_{ij} * R_{ni}\right)\right)\right| + \varepsilon}{\sum_{j=1}^{m}\left(x_{ij} * R_{p_j}\right) + \sum_{j=1}^{m}\left(x_{ij} * R_{ni}\right)}\right] \qquad (4)$$

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12 Number 22 (2017) pp. 12044-12051

Subjected to:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\ j \in J \tag{5}$$

$$\sum_{j=1}^{m} R_{p_j} * x_{ij} \le T_{p_i} * y_i \quad \forall\ i \in I \tag{6}$$

$$\sum_{j=1}^{m} R_{m_j} * x_{ij} \le T_{m_i} * y_i \quad \forall\ i \in I \tag{7}$$

$$y_i, x_{ij} \in \{0, 1\} \tag{8}$$

Constraint Eq. (5) assigns a VM_j to only one of the servers. Constraints Eq. (6) and Eq. (7) model the capacity constraint of the server. Constraint Eq. (8) defines the domain of the variables of the problem. Given a set of ***m*** virtual machines and a set of ***n*** physical machines, there are a total of ***m*n*** possible VM placement solutions.

BI-OBJECTIVE EVALUATION

Decision makers refer to choosing a solution out of all the efficient solutions as a posteriori approach. Pareto is one well known pioneer in multi-objective optimization problems. In this method, Pareto-optimal set is a group of best trade-off schedules, and Pareto-front refers to a set of Pareto solutions [21]. Overall fitness function formulation for two objectives is described by

$$f(x) = \delta_1 f_1(x) + \psi(1-\delta_1) * f_2(x) \tag{9}$$

Where δ_1 is the weight of first objective function and ψ is a ratio to make balance among objectives with different ranges of value [16, 18, 19, 37], which is defined by

$$\psi = \frac{max\ f_1(x)}{max\ f_2(x)} \tag{10}$$

GENETIC ALGORITHM (GA)

GA is a probabilistic optimization algorithm that imitates the progression of natural evolution. The biological evolution process in chromosomes became the idea of GA. It is based on the idea of the fittest survival where new better solutions are obtained by recombination of with each other. Algorithm **1** shows the pseudo-code for GA algorithm for the optimal solution in cloud computing.

Algorithm 1: Pseudocode of a Genetic Algorithm

1 $t \leftarrow 0$
2 Generate an initial population ***P(t)***
3 **while** *the stoping criterion is not fulfilled* **do**
4   Evaluate the objective vector ***f*** for each individual in ***P(t).***
5   Assign a fitness for each individual in ***P(t).***
6   Select from ***P(t)*** a group of parents ***P'(t)*** preferring the fitter ones.
7   Recombine individuals in ***P'(t)*** to create a child population ***P"(t)***
8   Mutate individuals in ***P"(t)***
9   Combine ***P(t)*** and ***P"(t)*** and select the best individuals to get ***P(t+1).***
10     $t \leftarrow t + 1$.
11 **end.**

The fitness assignment scheme requires a ranking of the individuals according to a preference relation and then, assigning a scalar fitness value to each individual using such rank. The selection for reproduction (line 6) is carried out as in the single objective case, for instance, using tournament selection. In contrast, the selection for survival (line 9), intended to maintain the best solutions so far (i.e., elitism). It helps the VM allocator to choose the right machine for the allocation of resources. Usually, the initial population is generated in a random manner. The number of genes in each chromosome equals the total numbers of VMs to be placed,

A. Encoding

A chromosome in this GA consists of $|C_r|$ genes, each represents the VM id (VM_j) to be placed at PM id (PM_k). The value of a gene is a positive integer between 1 and PM_{max}, representing the physical machine where the VM to be placed. Fig. **2** shows an example of individual VM placement and its corresponding chromosome. In this example, task VM_1, VM_2, VM_3, VM_4, VM_5, VM_6, VM_7, VM_8, VM_9 and VM_{10} are placed on PM_1, PM_3, PM_1, PM_1, PM_3, PM_3, PM_2, PM_2, PM_3, and PM_4 respectively. We are also updating the active servers list, if it is used marked as 1 otherwise 0 if not in use.

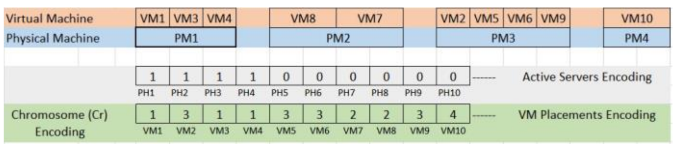

Fig. 2. Individual encoding (chromosome)

B. Chromosome generating

A chromosome (C_r) is a random construct of operations, with subjected to the given in Eq. (5-8) to ensure that the generated encoding is valid individuals. Then, the total fitness values of the efficient frontiers will be calculated based on Eq. (3) and Eq. (4) and overall fitness functions calculated as Eq. (9).

C. New population

New population will be produced based on the below sub-steps: selection, crossover, elitism, and mutation operation.

1) Selection: To constantly enhance the population overall fitness, selection helps to discard the bad and weak designs and only keep the best ones in the population. It increases the likelihood of selection of fitter individuals for the next generation. There are a few different selection methods but their basis is the same. The tournament candidate selection, which is a proportionate random selection method, is used in this study. [38].

2) Crossover: Crossover operator generates two new chromosomes for the next generation out of two selected chromosomes by exchanging some of their genes. This study employs two crossover operators based on partial strings exchange; a one-point crossover and a two-point crossover [39]. The offspring of crossover between the strings may not produce a legal encoding. Therefore, they should be repaired and legalized. For repairing mechanism, validating the overloaded placement from the left, the redundant genes will be deleted and compensate with the valid or empty PM_k ones, in order for each offspring to comprise all the *VMs* will be placed with valid physical machines and satisfy the Eq. (5-8).

3) Mutation: Mutation is another important operator of a GA that initiates extra variability in a population to create and maintain the diversity. Mutation is not applied on chromosomes that are immune. Shift mutation is used in this paper and chromosomes produced out of shift mutation need to be repaired and legalized like crossover. The number of mutations, nM in each generation is calculated using Eq. (11) based on the mutation rate (*rM*), population size *(sP)*, and maximum gene code (g_{max}).

$$nM = sP(g_{\max} * rP) \tag{11}$$

4) Elitism: The first three best chromosomes from each generation are transferred directly to the next generation in the elitism step to avoid annihilation. It is possible to maintain a fixed fitness value in some generations, but elitism makes sure they will never deteriorate.

5) Termination: The loop of chromosome generation is terminated when the number of generation reaches its maximum, then the elite chromosome returns as the best solution.

PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a population based stochastic technique inspired by social behavior of bird flocking [29-32]. The PSO configuration for the mathematical model is described in details in the following steps:

A. PSO Encoding

Initialization involves setting the parameters of the PSO and creating a group of particles to make the initial swarm sa shown in Fig **3.** We devise a two-dimensional encoding scheme, which maps one-to-many relationship between the server and the virtual machine as proposed by [40] for the VM placement problems. In Fig. **3**, the first dimension of a particle is an ***n*** bit binary vector which represent the active PM in a heterogeneous virtualized data center.

Fig. 3. Particle Encoding

Here, '1' indicates that the corresponding PM is active (in use) and '0' indicates that the PM is idle. The 2^{nd} dimension of a particle is a set of subsets that comprises the VMs to be placed in a set of PMs as shown in Fig. **3.**

B. Initializing population (swarm)

A group of particles are needed to create a swarm. Each particle has position (*P*) and velocity (*V*) in the search space at iteration (t), where they are described briefly in the following sub-steps:

a) *Particle position*: First position of i^{th} particle on d^{th} dimension of the particle using Eq. (12).

$$p_{i,d}^{0} = p_{min} + (p_{pmax} - p_{min}) * rand(0,1) \tag{12}$$

Where $p_{min} = 1$, $p_{max} = VM_{max}$.

b) *Particle velocity:* Initial velocities of i^{th} particle on d^{th} dimension are generated by the Eq. (13):

$$v_{i,d}^{0} = v_{min} + (v_{max} - v_{min}) * rand(0,1) \tag{13}$$

Where $v_{min} = 0$, $v_{max} = VM_{max}$.

c) *Multi-objective evaluation*: Once the swarm is generated, each particle is evaluated by both objective function given in eq. (1) and (2). Then, the multi-objective evaluation (total fitness values of the efficient frontiers) will be calculated based on Eq. (9).

d) *Personal best: $PB_i(t)$* represents the best position associated with the best permutation and fitness value of the i^{th} particle obtained in t^{th} iteration and is called the personal best. For each particle, $PB_i(t)$ can be determined and updated at each iteration.

e) *Global best: GB(t)* denotes the globally best solution achieved in the whole swarms.

C. New swarm

To produce a new swarm, the position and velocity of the particles should be updated. Updated particles will be evaluated as given below steps and again the best local and global particle will be determined. This procedure will be repeated up to a point where the termination criterion is satisfied.

Updating the velocity of each particle: The velocity of each particle is updated using the Eq. **(**14)**.**

$$v_{i,d}(t+1) = \omega v_{min} + C_1\phi_1(PB_{i,d}(t) - p_{i,d}(t)) + C_2\phi_2(GB_d(t) - p_{i,d}(t)) \quad (14)$$

Where C_1 is self-confidence while C_2 is swarm confidence (values of C_1 and C_2 varies between 0.1 and 0.5). Inertia weight (ω) is a parameter to control the impact of the previous velocity on the current velocity [16]. Let ω be varying with time by the following linear decreasing function Eq. (15).

$$\omega = \omega_{max} - t * \frac{\omega_{max} - \omega_{min}}{t_{max}} \quad (15)$$

Updating the position of each particle: The position of particle is updated using updated velocity as Eq. (16).

$$p_{i,d}(t+1) = p_{i,d}(t) + v_{i,d}(t) \quad (16)$$

D. Termination

The loop of swarm groups is terminated when it reaches the maximum number of iteration, then the particle with global best returns as the best solution.

HYBRID GA-PSO

GA has the capability of simultaneous evaluation of many points in the search area, which increases the probability of finding the global solution of the problem. Generally, PSO functions based on the social interaction knowledge and all the individual particles will be considered in each generation. Unlike PSO, fitter chromosomes will be chosen in GA and the weaker ones will fade away from generation to generation [41], [16]. Hence, by integrating the advantages of the compensatory properties of PSO and GA, their hybrid is used to obtain better result [42-44]. In the proposed GAPSO algorithm for this study, after generating and evaluating the initial swarm and after position and velocity updating, the crossover operation has been used in the GA segment to avoid premature convergence; and a mutation operation was applied to maintain the diversity of the swarms. Elitism step was performed to improve immune particle filter as proposed algorithm [16]. The pseudo-code for the proposed Hybrid of GA and PSO is mentioned in algorithm 2.

Algorithm 2 : Pseudo-code for Hybrid of GA-PSO

for{each particle **i = 1, ..., S** }{
  Initialize the particle's position random vector:
   $\mathbf{x_i = random(b_{lo}, b_{up})}$
  Initialize the particle's best known position to its initial position: $\mathbf{p_i} \leftarrow \mathbf{x_i}$
  if ($f(p_i) < f(g)$) then
    update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p_i}$
  endif
  Initialize the particle's velocity:
  $\mathbf{v_i = random(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)}$
  endfor
While {a termination criterion is not met} **do**
  for {each particle **i = 1, ..., S**} **do**
    for{ each dimension **d = 1, ..., n**}**do**
      Pick random numbers:
      r_p = random(0,1)
      r_g = random(0,1)
      Update the particle's velocity as per Eq. (14).
    endfor
  Update the particle's position as per Eq. (16).
  \\Hybrid of Genetic Algorithm Block
  if { agent $\mathbf{x_{i,d}}$ positions are not satisfying the Eq (5-8) **then**
    Generate random solution for the agent $\mathbf{x_{i,d}}$
  endif
  Use selection Tournament for selecting agents.
  Perform one-point crossover and two-point crossover.
  Repairing off springs.
  Perform Shift Mutation.
  End of Hybrid of GA Block\\
  if ($f(x_i) < f(p_i)$) **then**
    Update the particle's best known position: $\mathbf{p_i} \leftarrow x_i$
    if $f(p_i) < f(g)$ then
      Update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p_i}$
    endif
  endif
 endfor
endwhile

SIMULATION RESULTS

In this paper, the performance of Hybrid GA-PSO is evaluated in the term of active servers, energy consumption, server's utilization and resource wastages. To validate the model, we are investigated on the Cloud Sim simulator, with number of virtual machine requests is 200 for the performance of the three algorithms. The Fig. **4** show the fitness vs Iterations of the three algorithms. The results shown in Fig. **4** proved successfully, in decreasing the fitness value as number of active servers. The optimized model using hybrid GA-PSO obtained the best results in minimum iterations.

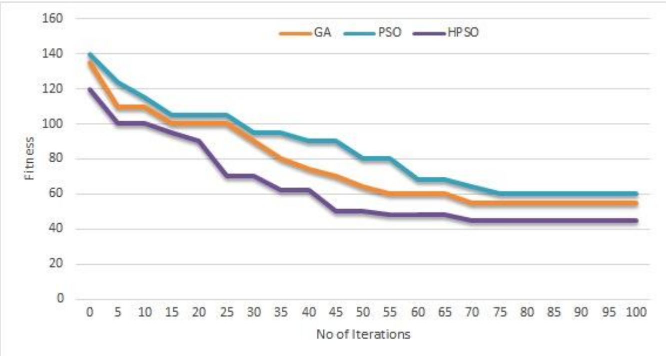

Fig. 4. Performance of evaluation

In Active servers experiment, the number of VM requests are 200, 400, 600, ..., 2000. The result shown in Fig. **5** gives the comparison results of active servers with other approaches. As shown in Fig. **5** with the different size of VM requests, the Hybrid GA-PSO approach always gives the minimum number of active servers.

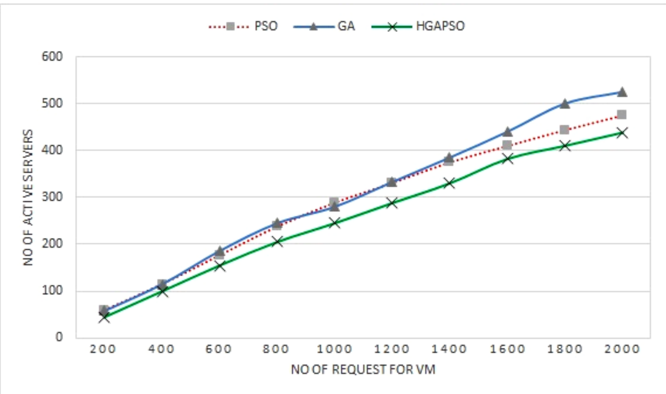

Fig. 5. Active Servers

In Fig. **6** shows the comparison results of Server Utilization of the CPU utilizations of all active servers. As shown in Fig. **6** with the different size of VM requests, the Hybrid GA-PSO approach always gives the better utilization compare to the other approaches.

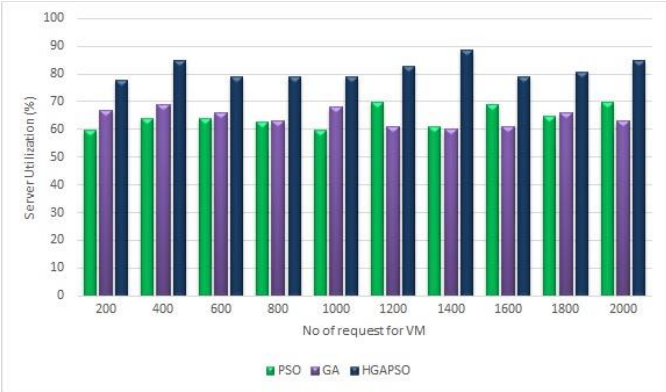

Fig. 6. Server Utilization

In Fig. **7** and Fig. **8** shows the total energy consumptions and resource wastages of all active servers. As shown in fig. **7** and Fig. 8 with the different size of VM requests, the Hybrid GA-PSO approach always gives the lowest energy consumption and resource wastages with compare to the other approaches.

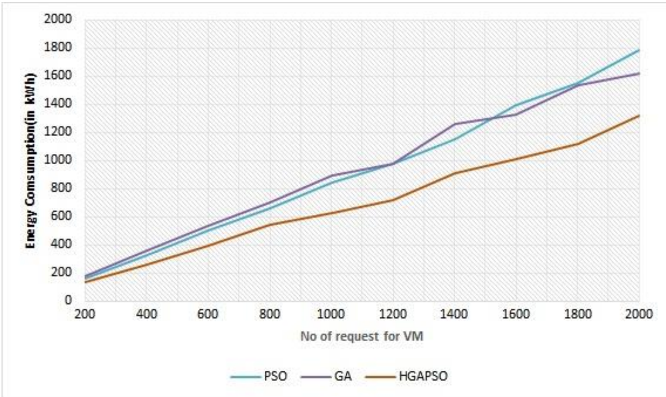

Fig. 7. Energy Consumptions

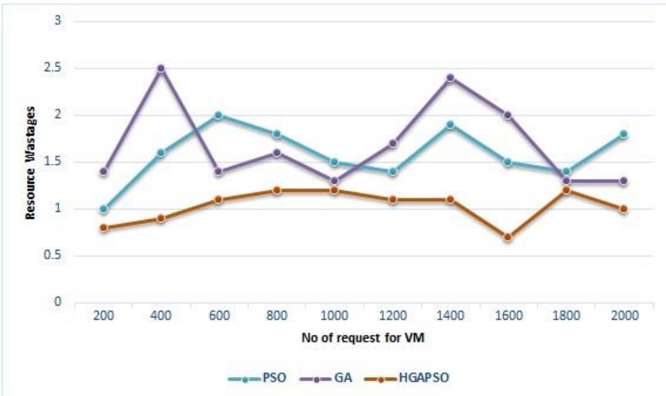

Fig. 8. Resource Wastages

CONCLUSION

In this paper, we presented the optimized VMP for a cloud data center to reduce the total energy consumptions with better VM placement algorithm using HGAPSO. Our approach accounts for the multi-objective resource management. The simulation based result validate the effectiveness of our approach compared to the other approaches. Future work focused on providing the VM placement with the migrations.

REFERENCES

[1] R. Buyya, J. Broberg, and A. M. Goscinski, Cloud computing: Principles and paradigms. Wiley. com, 2010, vol. 87.

[2] M. A. Vouk, Virtualization of Information Technology Resources. In Electronic Commerce: A Managerial Perspective 2008. 5th Edition, Prentice-Hall Business Publishing, 2008.

[3] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," NIST special publication, vol. 800, no. 145, p. 7, 2011.

[4] K. Li, H. Zheng, J. Wu, and X. Du, "Virtual machine placement in cloud systems through migration process," International Journal of Parallel, Emergent and Distributed Systems, vol. 30, no. 5, pp. 393–410, 2015.

[5] H. Zhao, Q. Zheng, W. Zhang, Y. Chen, and Y. Huang, "Virtual machine placement based on the vm performance models in cloud," in Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance. IEEE, 2015, pp. 1–8.

[6] M. Cardosa, M. R. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on. IEEE, 2009, pp. 327–334.

[7] L. Grit, D. Irwin, A. Yumerefendi, and J. Chase, "Virtual machine hosting for networked clusters: Building the foundations for autonomic orchestration," in Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing. IEEE Computer Society, 2006, p. 7.

[8] K. Li and H. Shen, "Proxy placement problem for coordinated en-route transcoding proxy caching," Computer Systems Science and Engineering, vol. 19, no. 6, pp. 327–335, 2004.

[9] K. Li and H. Shen, "Optimal placement of web proxies for tree networks," in e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on. IEEE, 2004, pp. 479–486.

[10] L. Keqiu and S. Hong, "Optimal proxy placement for coordinated enroute transcoding proxy caching," IEICE transactions on information and systems, vol. 87, no. 12, pp. 2689–2696, 2004.

[11] K. Li, H. Shen, F. Y. Chin, and S. Q. Zheng, "Optimal methods for coordinated enroute web caching for tree networks," ACM Transactions on Internet Technology (TOIT), vol. 5, no. 3, pp. 480–507, 2005.

[12] K. Li, H. Shen, F. Y. Chin, and W. Zhang, "Multimedia object placement for transparent data replication," Parallel and Distributed Systems, IEEE Transactions on, vol. 18, no. 2, pp. 212–224, 2007.

[13] W. Vogels, "Beyond server consolidation," Queue, vol. 6, no. 1, pp. 20–26, 2008.

[14] A. S. Vasava, "Scheduling of automated guided vehicle in different flexible manufacturing system environment," International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 1, no. 8, pp. 262–7, 2014.

[15] J. A. Ventura, S. Pazhani, and A. Mendoza, "Finding optimal dwell points for automated guided vehicles in general guide-path layouts," International Journal of Production Economics, vol. 170, pp. 850–861, 2015.

[16] M. Mousavi, H. J. Yap, S. N. Musa, F. Tahriri, and S. Z. M. Dawal, "Multi-objective agv scheduling in an fms using a hybrid of genetic algorithm and particle swarm optimization," PloS one, vol. 12, no. 3, p.e0169817, 2017.

[17] H.-F. Wang and C.-H. Chan, "Multi-objective optimisation of automated guided dispatching and vehicle routing system," International Journal of Modelling in Operations Management, vol. 4, no. 1-2, pp. 35–52, 2014.

[18] A. Ghane-Kanafi and E. Khorram, "A new scalarization method for finding the efficient frontier in non-convex multi-objective problems," Applied Mathematical Modelling, vol. 39, no. 23, pp. 7483–7498, 2015.

[19] I. Giagkiozis and P. J. Fleming, "Methods for multi-objective optimization: An analysis," Information Sciences, vol. 293, pp. 338–350, 2015.

[20] K. Miettinen, Nonlinear multiobjective optimization. Springer Science & Business Media, 2012, vol. 12.

[21] M. Kaisa, "Nonlinear multiobjective optimization," Boston, Massachusetts: Kluwer Academic Publishers, 1999.

[22] S. Greco, J. Figueira, and M. Ehrgott, "Multiple criteria decision analysis," Springer's International series, 2005.

[23] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," Knowledge and Information systems, vol. 1, no. 3, pp. 269–308, 1999.

[24] K. Deb, Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, 2001, vol. 16.

[25] D. B. Fogel, Artificial intelligence through simulated evolution. Wiley-IEEE Press, 2009.

[26] D. Fogel and Z. Michalewicz, "How to solve it: Modern heuristics," Ed Springer, 2000.

[27] R. S. Rosenberg, "Stimulation of genetic populations with biochemical properties: I. the model," Mathematical Biosciences, vol. 7, no. 3, pp. 223–257, 1970.

[28] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in Proceedings of the 1st international Conference on Genetic Algorithms. L. Erlbaum Associates Inc., 1985, pp. 93–100.

[29] Y. Shi et al., "Particle swarm optimization: developments, applications and resources," in evolutionary computation, 2001. Proceedings of the 2001 Congress on, vol. 1. IEEE, 2001, pp. 81–86.

[30] M. L. Song, "A study of single-objective particle swarm optimization and multi-objective particle swarm optimization," in Applied Mechanics and Materials, vol. 543. Trans Tech Publ, 2014, pp. 1635–1638.

[31] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," IEEE Transactions on Evolutionary Computation, vol. 16, no. 2, pp. 210–224, 2012.

[32] Y. Gao, W. Du, and G. Yan, "Selectively-informed particle swarm optimization," Scientific reports, vol. 5, 2015.

[33] L. Minas and B. Ellison, "The problem of power consumption in servers," Intel Corporation. Dr. Dobb's, 2009.

[34] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," The Journal of Supercomputing, vol. 60, no. 2, pp. 268–280, 2012.

[35] A. J. Younge, G. Von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," in Green Computing Conference, 2010 International. IEEE, 2010, pp. 357–364.

[36] V. Pareto, Corso di economia politica. P. Boringhieri, 1961.

[37] G. Eichfelder, Adaptive scalarization methods in multiobjective optimization. Springer, 2008, vol. 436.

[38] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," Addion wesley, vol. 1989, p. 102, 1989.

[39] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," Machine learning, vol. 3, no. 2, pp. 95–99, 1988.

[40] S. Wang, Z. Liu, Z. Zheng, Q. Sun, and F. Yang, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers," in Parallel and Distributed Systems (ICPADS), 2013 International Conference on. IEEE, 2013, pp. 102–109.

[41] C.-D. Liou, Y.-C. Hsieh, and Y.-Y. Chen, "A new encoding scheme based hybrid algorithm for minimizing two-machine flow-shop group scheduling problem," International Journal of Systems Science, vol. 44, no. 1, pp. 77–93, 2013.

[42] M. Mehta, "Hybrid genetic algorithm with pso effect for combinatorial optimization problems," International Journal of Advanced Computer Research, vol. 2, no. 4, pp. 300–305, 2012.

[43] C.-H. Wu, N. Dong, W.-H. Ip, C.-Y. Chan, K.-L. Yung, and Z.-Q. Chen, "Chaotic hybrid algorithm and its application in circle detection," Applications of evolutionary computation, pp. 302–311, 2010.

[44] L. Wang and G. Si, "Optimal location management in mobile computing with hybrid genetic algorithm and particle swarm optimization (gapso)," in Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on. IEEE, 2010, pp. 1160–1163.