Load Balancing In Cloud Computing Using BFO Optimized DWOLB

Rakhi Narang[1], Deepika Arora[2]
[1]M.tech Scholar, RPIIT Karnal
[2]A.P., RPIIT Karnal

***Abstract*—**Cloud Computing is an on-demand type of service which provides services or software to end user whenever he requires and demand it. The term Cloud is used for service provider who have all type of resources for storage, computing and service providing. For realizing full capacity in cloud computing, it needs to support different services such as security, uniform access, task scheduling, resource management, economic computation etc. In this work we worked to minimise the energy consumption in cloud data centres The newly developed whale optimisation algorithm (WOA) is used to assign the VM to physical nodes for maximum utilisation of two resources within available capacity of system. We have compared BFO with Genetic Algorithm (GA) on various parameters and found good results for BFO.

I. INTRODUCTION

The cloud computing relies on the thought of dynamic provisioning, that is applied to services, computing capability, storage, networking, and data technology infrastructure to fulfil user needs. The resources are created offered for the users through the web and offered on a pay-as-use basis from completely different Cloud computing vendors.

Cloud computing infrastructures are designed to support the accessibility and preparation of assorted services orientating applications by the users. Cloud computing services are created offered through the server corporations or knowledge centres. to fulfil the growing demand for computations and enormous volume of information, the cloud computing environments provides high performance servers and high speed mass storage devices [1-2]. These resources are the key supply of the ability consumption in knowledge centres along side air-con and cooling instrumentation. what is more the energy consumption within the cloud is proportional to the resource utilization and knowledge centres are virtually the world's highest shoppers of electricity [5]. because of the high energy consumption by knowledge centres, it needs economical technology to style inexperienced knowledge centre. On the opposite hand, Cloud knowledge centre will cut back the full energy consumed through task consolidation and server consolidation mistreatment the virtualization by workloads will share constant server and unused servers may be converted. the full computing power of the Cloud knowledge centre is that the total of the computing power of the individual physical machine. Clouds uses virtualization technology in knowledge centres to portion resources for the services as per want. Clouds provides 3 levels of access to the customers: SaaS, PaaS , and IaaS. The task originated by the client will take issue greatly from client to the client. Entities within the Cloud are autonomous and self-interested; but, they're willing to share their resources and services to attain their individual and collective goals. In such an open surroundings, the programming call may be a challenge given the suburbanized nature of thesurroundings. every entity has specific needs and objectives that require to attain. Server consolidations are permitting the multiple servers running on one physical server at the same time to reduce the energy consumed in a very knowledge centre. Running the multiple servers on one physical server is completed through virtual machine thought. The task consolidation is additionally called server/workload consolidation downside. Task consolidation downside self-addressed during this thesis is to assign n task to a collection of r resources in cloud computing surroundings. This energy economical resource allocation maintains the employment of all computing resources and distributes virtual machines in a very manner that the energy consumption will minimize. The goal of those algorithms is to keep up accessibility to work out nodes whereas reducing the full energy consumed by the cloud infrastructure. Efficient load balancing using different methods in cloud computing is implemented by [1-10]. But no one implemented WOA in cloud environment to test the results of this newly introduced algorithm.

In this paper we have implemented cloud environment and tested it using newly introduced bacterial foraging optimisation (BFO). We have used two resources first one CPU and second one is disk which is allocated to each Virtual Machine (VM). Performance of load distribution and balancing is compared with genetic algorithm (GA) which is standard optimization algorithm.

II. BACTERIAL FORAGING OPTIMISATION

Bacterial Foraging Optimization Algorithm (BFOA) is proposed by Kevin Passino (2002), is a new comer to the family of nature inspired optimization algorithms. Application of group foraging strategy of a swarm of *E.*coli bacteria in multi-optimal function optimization is the key idea of this new algorithm. Bacteria search for nutrients is a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis. The key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space.

p : Dimension of the search space,
S : Total number of bacteria in the population,
Nc : The number of chemotactic steps,
Ns : The swimming length.
Nre : The number of reproduction steps,
Ned : The number of elimination-dispersal events,
Ped : Elimination-dispersal probability,
C(i): The size of the step taken in the random direction specified by the tumble.

Foraging theory is based on the assumption that animals search for and obtain nutrients in a way that maximizes their energy intake *E* per unit time *T* spent foraging. Hence, they try to maximize a function like E/T (or they maximize their long-term average rate of energy intake). Maximization of such a function provides nutrient sources to survive and additional time for other important activities (e.g., fighting, fleeing, mating, reproducing, sleeping, or shelter building). Shelter building and mate finding activities sometimes bear similarities to foraging. Clearly, foraging is very different for different species. Herbivores generally find food easily but must eat a lot of it. Carnivores generally find it difficult to locate food but do not have to eat as much since their food is of high energy value. The "environment" establishes the pattern of nutrients that are available (e.g., via what other organisms are nutrients available, geological constraints such as rivers and mountains and weather patterns) and it places constraints on obtaining that food (e.g., small portions of food may be separated by large distances). During foraging there can be risks due to predators, the prey may be mobile so it must be chased and the physiological characteristics of the forager constrain its capabilities and ultimate success. Bacterial Foraging optimization theory is explained by following steps.

1 Chemotaxis
2 Swarming
3 Reproduction and
4 Eliminational-Dispersal

These steps are formulated as the mathematical equations to represent the behavior of bacteria. In chemotactic step bacteria updates its position by formula as in 2.1

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i).\Delta(i)}} \qquad ...(2.1)$$

$\theta^i(j,k,l)$ represents *i*th bacterium at *j*th chemotactic, *k*th reproductive and *l*th elimination-dispersal step. *C*(*i*) is the size of the step taken in the random direction specified by the tumble(run length unit). In swarming step, it calls the other bacteria to its located best position, but to avoid the over access of food at best position the powerful bacteria is split with removal of weak part of it from the population which is in elimination and dispersal setp.

2.1 Algorithm

Step 1: Initialize the parameters S, Nc, Ns, Nre, Ned, Ped and the C (i), (i =1, 2,…, S). Choose the initial value for theθ^i,i=1,2….S. These must be done in areas where an optimum value is likely to exist. They are randomly distributed across the domain of the optimization space. After computation of θ is completed, the value of P (position of each member in the population of the S bacteria) is updated automatically and termination test is done for maximum number of specified iterations.

Step 2: Elimination-Dispersal loop: $l= l+1$

Step 3: Reproduction loop: $k= k+1$

Step 4: Chemo taxis loop: $j= j+1$

For i= 1, 2,….,S take a chemo tactic step for bacterium 'i' as follows:

(i) Compute cost $J(i,j,k,l)$.
(ii) Let $J(i,j,k,l)=J(i,j,k,l)+Jcc(\theta^i(j,k,l),P(j,k,l))$
(iii) Let *Jlast*= $J(i,j,k,l)$ to save this value since find better cost via a run
(iv) Tumble: Generate a random vector $\Delta^i \in R^p$ with each element $\Delta^i m$, m= 1, 2,….,p a random number on [-1,1]. Where R is areal number.
(v) Move let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i).\Delta(i)}}$$

(vi) Compute $J(i,j+1,k,l)$
(vii) Swim.
Let m=0 (counter for swim length)
While m<Ns
Let m=m+1
If $J(i,j+1,k,l)<Jlast$(if there is improvement), *let* *Jlast*= $J(i,j+1,k,l)$ and let $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i).\Delta(i)}}$and use this$\theta^i$ $(j+1,k,l)$ to compute the new $J(i,j+1,k,l)$.
Else, let m=Ns. End of while statement

(viii) Go to next bacterium ($i+1$) if *i*z S

Step 5: If *j*<Ncgo to step 3. In this case, continue chemo taxis, since the life of the bacteria is not over.

Step 6: Reproduction

For the given k and *l*, and for each *i*=1,2,….,S, let $J^i_{health} = \sum_{j=1}^{Nc+1} J(i,j,k,l)$be the health of bacterium i. Sort bacteria and chemo tactic parameter C(i) in order of ascending cost *Jhealth.*

The Sr bacterium with the highest *Jhealth*values die and the other Sr bacteria with the best values split

Step 7:

If k<Nre, go to step 2. In this case we have not reached the number

of specified reproduction steps

Step 8: Elimination-Dispersal

For i=1,2,…..,S with probability Ped, eliminate and disperse each bacterium. Eliminate a bacterium and disperse one to a random location on the optimization domain. If l <Ned, then go to step 1, otherwise end.

III. PROPOSED WORK

In this work we have allocated multiple virtual machines to different number of hosts. Efficient allocation of resources to given number of VM (virtual machine) is a quite complex task in cloud computing. This is a NP hard problem which can't be solved mathematically. As discussed in literature survey many researchers have worked for this kind of problem but they used artificial intelligence for it. In our work we solve this problem with BFO optimisation algorithm and compared the results with Particle swarm optimisation which is used in reference paper. The optimal resource allocation is NP hard problem so the algorithm should run to minimize the Euclidean distance as given in equation 1.3.

$$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_i^j - ubest_i)^2} \qquad 1.3$$

Where is the dimension which denotes kinds of resources, such as CPU, disk, memory, and bandwidth and denotes the number of hosts in cloud data centre.u_i^j is the utilization for host j and the resource i,$ubest_i$ is the best utilization for u_i^j. The total Euclidean distance denotes the optimal balance between multi resources utilization and energy consumption. Minimizing the total Euclidean distance will get optimal energy efficiency in the whole system. In this situation, the multi resources energy efficiency model is described as follows:

$$objective : min\ \delta \qquad 1.2$$

$$constraints: x_h^j = 0$$

$$\sum_h x_h = 1 \qquad 1.3$$

Where x_h^jdenotes virtual machine VM allocated to node h;$x_h^j = 0$ denotes VM is not allocated to resources and expression 3 states that each VM can be allocated to one node only. In order to satisfy the limitations, each resource must satisfy the following inequality constraints as follows:

$$\sum_j r_j^{CPU} * x_h^j \le c_h^{CPU}, \quad \sum_j r_j^{RAM} * x_h^j \le c_h^{RAM}, \quad \sum_j r_j^{BW} * x_h^j \le c_h^{BW},$$

$$\sum_j r_j^{DISK} * x_h^j \le c_h^{DISK}, \qquad 1.4$$

Here in this expression $r_j^{CPU}, r_j^{RAM}, r_j^{BW}, r_j^{DISK}$ denotes the demand of resources and c_h denotes the capacity of these resources. The above expression must be satisfied while assigning optimal nodes to VMs. The capacity is the maximum resource available to allot to VMs. In our work we have assumed only two kind of resources which are CPU and disk. The maximum and minimum allotted capacities of these are given in table 1.3.

Table 1.3: maximum and minimum limit of resources allocated to each VM

		Low	High
1	CPU (MIPS)	60	150
2	Disk (GB)	100	200

Each VM must be allocated the available resources within this range. So this problem has many constraints to fulfil and object to minimise the Euclidean distance, it becomes the NP hard problem and BFO is used in our proposed work to solve this equation.

Resource Allocation using BFO

In cloud based environment, resource allocation is very important thing. Genetic Algorithm is standard algorithm which is used in cloud based resource allocation. In this case we are using BFO algorithm instead of GA for resource allocation. Here we have taken some Virtual Machines and some physical nodes. These physical nodes are to be assigned virtual machine on the basis of two parameters namely CPU utilization and Disk utilization. Initially random values of resource allocation to each VM is assigned which is to be tuned iteration by iteration using BFO algorithm. A cost function is created using two parameters CPU utilization and Disk utilization. BFO algorithm will minimize this cost function for every updated value of position of bacteria. After set number of iteration we get optimized value of position of bacteria which is actually resource allocation value.

IV. RESULTS

The proposed work is implemented in MATLAB R2017. We have compared our work with BFO optimization results using same availability of resources and number of virtual machines. The allocation of VM to hosts should be such that total distance settles to a minimum value. Comparative results are also tested for different number of VMs. Table 4.1. Shows the input parameters used for energy minimization in cloud computing data centre, these values are picked from [6].

Table4.1: Input Parameters considered for cloud computing data center

Number of VMs	10,20,30,40,50,60
Number of physical nodes	100
Number of resources	2 (hard disk, CPU)
Bets resource utilisation ratio(hard disk, CPU)	[0.5,0.7]
Capacity of physical nodes	[2260 MIPS,21000 TB]

For BFO algorithm we have to initialise some parameters of Bacteria. These parameters are tabulated in table 4.2.

Table 4.2: Input parameters of BFO algorithm

Number of Bacteria	Equal to physical nodes
Number of iterations	100
Dimension of search space	Equal to Number of VMs
Number of chemotactic steps	25
Number of swim steps	4
Number of reproductive steps	4
Number of elimination and dispersal steps	2
The probability that each bacteria will be eliminated/dispersed	0.5
Upper and lower limits	[1,0]

We have tested results for 10-60 virtual machines over 100 physical nodes with same capacity of resources available. Using any kind of optimisation is bounded by restriction of randomness. Every optimisation algorithm is initialised randomly, so is ours and due to this random initialisation, results will be different in each trial. So we pasted best results here in 5-6 trials. Since cost function equation is to be minimised so the objective function value must be decreased with number of iterations. If it is not so then fine tuning of algorithm is required.

Since testing has been done for different number of VMs, so It has been noted from figure 4.1 that Euclidean distance of BFO algorithm is less than GA. For these final Euclidean distance figure 4.1 shows bar plot comparison.

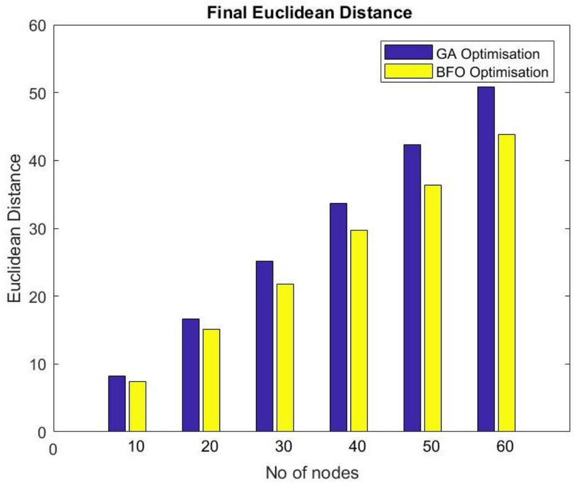

Figure 4.1: Euclidean distance comparative bar plot

From the analysis of figure 4.1 it is clear that Euclidean distance in BFO is less in comparison to GA. It means resource allocated using BFO, provides maximum resource utilization, as Euclidean distance gives.

Table 4.3 : Comparison of final disk allocation by both algorithms

Number of virtual machines	**BFO Algorithm (in GB)**	**GA Algorithm (in GB)**
10	7286.20285836438	1690.15609733109
20	12613.0597525932	3110.10573783506
30	22898.1955576938	3551.50794318088
40	25555.0363068507	3721.10228870964
50	37961.2125846768	4094.84792684344
60	43761.1925868016	3965.59767447076

From the analysis of table 4.3 and 4.4 it is clear that for all VMs BFO algorithm is performing well by increasing the disk and CPU utilisation.

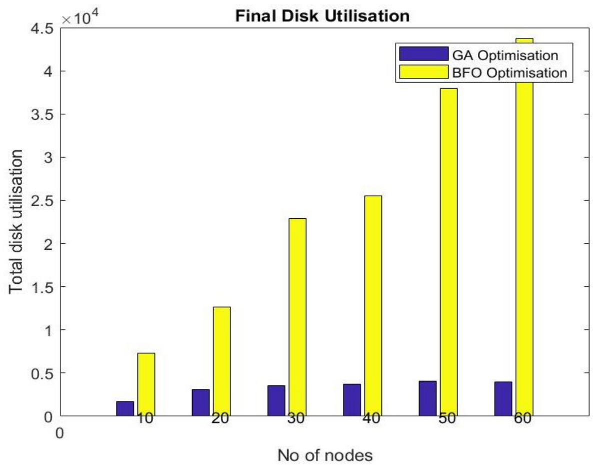

Figure 4.2: final disk utilisation comparative bar plot

From the analysis of figure 4.2 it is clear that disk utilization by BFO is very high in comparison to GA. It means resource allocated using BFO , provides maximum resource utilization in terms of disk utilization.

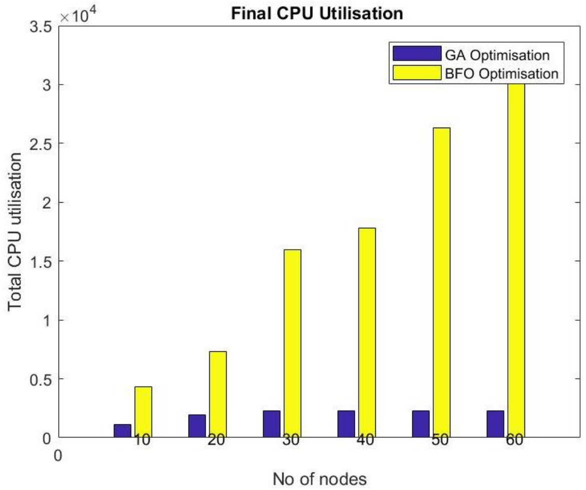

Figure 4.3: final CPU utilisation comparative bar plot

Table 4.4: Comparison of final CPU utilisation by both algorithms

Number of virtual machines	BFO Algorithm	GA Algorithm
10	4354.40085802196	1134.51935227877
20	7348.62061459957	1925.79962153231
30	15979.9409955825	2259.61268350403
40	17839.2538962931	2258.16979053925
50	26348.2431032123	2258.14814333619
60	31291.6865739134	2259.61807087033

From the analysis of figure 4.3 it is clear that CPU utilization by BFO is very high in comparison to GA. It means resource allocated using BFO algorithm, provides maximum resource utilization in terms of CPU utilization.

V. CONCLUSION

Our work is based on utilising the maximum resources for a particular number of VM within the available capacity of each resource. For this purpose Euclidean distance between hosts and VMs is considered as deciding factor since minimum is the distance, less is the energy consumption. So, we used BFO optimisation algorithm for this purpose since this is not the linear problem which can be solved mathematically, this is a problem bounded with many constraints and parameters. The outcome of algorithm is checked for various number of VM like 10,20,30,40,50 and 60. Their performance with BFO algorithm is compared with GA and it has been noted that whatever is the algorithm, resource utilisation is increasing with number of VM.

REFERENCES

[1]. SaurabhBilgaiyan, SantwanaSagnika, Madhabananda Das, "An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms", International Journal of Computer Applications (0975 – 8887) Volume 89 – No.2, March 2014.

[2]. Korir Sammy, RenShengbing, Cheruiyot Wilson, "Energy Efficient Security Preserving VM Live Migration In Data Centers For Cloud Computing", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012. ISSN (Online): 1694-0814.

[3]. Manasa H.B, AnirbanBasu , "Energy Aware Resource Allocation in Cloud Datacenter", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-5, June 2013 277 .

[4]. WannengShu, Wei Wang and Yunji Wang, "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing" ,EURASIP Journal on Wireless Communications and Networking 2014, 2014.

[5]. N.R.RamMohan, E.Baburaj, "Resource Allocation Using Interference Aware Technique in Cloud Computing Environment", International Journal of Digital Content Technology and its Applications(JDCTA) Volume8, Number1, February 2014.

[6]. An-ping Xiong and Chun-xiangXu, "Energy Efficient Multiresource Allocation of Virtual Machine Based on PSO in Cloud Data Center" , Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 816518.

[7]. Arindam Banerjee, Prateek Agrawal and N. Ch. S. N. Iyengar, "Energy Efficiency Model for Cloud Computing" , International Journal of Energy, Information and Communications Vol.4, Issue 6 (2013), pp.29-42. ISSN: 2093-9655 IJEIC Copyright ⓒ 2013 SERSC.

[8]. Akshat Dhingra and Sanchita Paul, "Green Cloud: Heuristic based BFO Technique to Optimize Resource Allocation", Indian Journal of Science and Technology, Vol 7(5), 685–691, May 2014.

[9]. Giuseppe Portaluri, Stefano Giordano, DzmitryKliazovich and Bernab´eDorronsoro, "A Power Efficient Genetic Algorithm for Resource Allocation in Cloud Computing Data Centers", 2014 IEEE 3rd International Conference on cloud Networking (Cloud Net).

[10]. Vanitha, Marikkannu," Effective resource utilization in cloud environment through dynamic well-organized load balancing algorithm for virtual machines", Computers and Electrical EngineeringElsevier December2016.

[11]. Sayedali Mirjalili, Andrew Lewis,"The Whale Optimization Algorithm",Advance in Engineering Software Vol-95, May 2016, Pages 51-67.