Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Performance Calculation On Virtual Machine Using Advanced Artificial Bee Colony Optimization Algorithm

[1]Abdul Razaak MP , [2]Mohamed Divan Masood

[1,2]Department of Computer Application, B.S. Abdur Rahman Crescent Institute of Science & Technology, Chennai, Tamilnadu, India.

Abstract

Cloud computational intends computational services and memory and get into use of information on the Internet. CC will not store any selective info on PC. The essential objective of cloud is to provide access to the users from data centers. Remote servers are also provide to get the information from CC. The essential project behind cloud computing is that the location of the service and countless details as an sample the hardware and the functions of the design on which the service is running have runt to attain with the user. In this study, a new planning model is proposed as an advanced artificial bee colony optimization method (AABCO). Many virtual technologies make use of cloud data centers, which have the advantage of allowing multiple virtual machines to run on a single server. Deploying a VM is the process of assigning a VM to the appropriate physical machine. The VM integration is used for the improvement of resource utilization for saving even more power using Efficient virtual machine deployment solutions. The old-bee algorithm mimics the bee foraging strategy to track down the finest solution to the optimization problem. The proposed AABCO implements an efficient way to identify the best VM with the lowest power consumption. Simulation results show that AABCO algorithms achieve effective results with a minimum energy consumption of 85%. Furthermore, simulation findings disclose that the recommended AABCO plan remarkably is less power than ABC and SPSO.

Keywords: Cloud Computing's, Virtual Machine Placements, Advanced Artificial Bee Colony Optimizations, Artificial-Bee Colony (ABC), Standard Particle Swarm Optimizations (SPSO).

I. INTRODUCTION

The use of off-site technologies to aid computer cache, organize, process, and swap explains is known as Cloud-Computing. In spite of being cached on the pc or other local warehouse, these off-site vendibles are hosted in the cloud. They can limit from email-server's to tool packages even information retrieval, and even adorning the computation capability of the machine. Computation refers to framework and movement that permits the pc to run, originate, distribute, and work together with data. This admits, preferably storing configuration, systems, or programs on drive or on an on-site server, which can be hosted on potential or online hosts that can link the pc over steady networks. CC's are of three categories: Software-Services, Platform-Services, and Infrastructure-Services.[1] Service function is known as serverless computing, is a well-liked way of CC for ventures (Faa S).

SAAS is web based so can be used directly lieu of plonking software on PC's such as the following:

- ✓ Square, an online payment processor
- ✓ O Google Apps like Calendar and Drive, as well as
- ✓ O Slack, a collaboration and communication tool.

IAAS or Infrastructure As A Service. IAAS favors organization components such as hosts, caches, network's, security, and even cloud. Here is an example:

- ✓ File storage and Sharing system in Dropbox,
- ✓ Microsoft Azure provides backup and disaster recovery services, hosting etc..
- ✓ Rackspace nurtures data, security, and Infrastructure-Services.

PAAS or Platform As A Services. PAAS nurtures a gauge platform such as an OS/2,an environment for running programming languages, do Google-databases, and web servers. Here is an example:

- ✓ Developers may use App Engine and Heroku to create and deploy apps.

• Serverless Computing: Serverless computing titled as Serverless is simply utilizing a server on the cloud. This affords easier maintenance, more malleableness, and is frequently more tag efficient than hosting servers on-site.

A virtual machine allows you to run one OS within another. VM's placement is the practice of mapping a group of VMs onto a set of physical machines (PMs) in a data center with the purpose of maximizing assets utilization and lowering total power squandered by PM's .

It has been propounded and researched the usance of organically inspired algorithms such as AABCO that conform to data centre mapping. The AABCO method, in combination with ABC and SPSO, is intended to tackle the problem of VM's berth in cloud data-collection. We enhanced the AABCO solution to this optimalization hindrance in cloud data centres to solve and reduce the zeal usage.

II. RELATED WORK

Cloud computing environments have enabled to run an assortment maneuvering's that are more economical for both individual consumers and organizations. However, with increasing demand for cloud-based services, energy-consuming in cloud data centers has increased significantly. Therefore, much research try has been composed to play down power consumption. This section outlines the work related to planning.

Deafallah Alsadie presents his work on a Meta heuristic-Framework titled MDVMA for activating VM allotment resorting task scheduling systematized in a Cloud-Computing habitat. MDVMA objectives to construct multi-objective scheduling near as said by a non-dominated sorting genetic subroutine for enhancing the mission forecasting with the goal of decreasing power exertion, make-span, and sum whereas supplying a tradeoff to cloud service providers according to their needs. The proposed MDVMA framework is great to already approach by playing down power consumption by 35.82%, 25.88% and 16.13%. The matter of dynamic mission scheduling was largely focused on paper.

In their paper, Karan D. Patel et al. introduced a load balancing technique that combines 2 subroutines for balancing assignments across cloud-systems. For priority-based activities, we employed a change in queen bee behavior impressive subroutine, and for non-priority-based tasks, we used an upgraded biased round-robin method. Our research is important since it aims to increase performance in the system , resource utilizations, and completion time. The customized bee behaviour virtuosic algorithm was utilized for emphasized based tasks, while the fortified swayed RR Algorithm was employed for non-prioritized based tasks. On a virtual machine, the task's minimal completion time will be considered in this technique.

In their study, SangWook Han et al. suggest employing the Knapsack algorithm to increase energy-consuming by expanding the concretion of virtual machines. Furthermore, by refining the Knapsack algorithm, the suggested method may perform optimized VM transferal in a short-range of time. The propounded method can improve overall data processing capabilities by reducing resource waste and increasing the usage rate of virtual machines. Furthermore, because it estimates the ongoing time, assets standardization may be emended when VM's are released. As the end product of this research, the total server resource consumption rate is enhanced by 5% using the proposed method, which is an optimal VM allocate approach for resource utilization rate.

Cuijie Zhao et al. suggested an enhanced ABCO technique algorithm in their article, which addresses the issues of redeveloped unification and sinking into the local sovereign worth in hyper spectral picture classification. We attempt to employ the enriched chaotic girdling to build a extraneous fractal chaotic sequence, broaden the propagation span of honey radixes, and escape the parish requisite in this study. In the domain delve, the researcher introduced an coherent impulsive step stature to embellish the algorithm's capacity to converge and productiveness.

According to Mingyue Feng and colleagues, the topic of resource distribution in cloud computing is regarded to be a intertwining optimization problem for a large corporation with a high number of clients and owned resources. This hindrance is gotten plying a Particle-Swarm-Optimization subroutine. The technique seeks to notice a proper mission scheduler on resources as said by there are numerous factors as an example whole, entire mission time execution, resource reservation, and mission QOS. We provide an algorithm-based PSO that considers three distinct objectives to identify this problem as a intertwining optimization problem. A trace encoding mechanism is built in the algorithm based on work scheduling characteristics in a CC lodgment, and then traditional particle encoding is applied.

To tackle the job scheduling problem, Zhifeng et al, used a Greedy and Particle-Swarm-Optimization (G&PSO) sited algorithm. It uses a miserly strategy to make out the pioneer tad value of a PSO method received from a VM allotted cloud dock. In this study, the G-PSO method was handed-down to alleviate gross fruition time and poise caseload in each Virtual-Machine, with the goal of solving the mission forecasting difficulties of Virtual-Machines on a cloud tribune.

Table 1: Summary of related works.

Reference	Objective	Algorithm
Deafallah Alsadie	His work presents a metaheuristic framework called MDVMA for dynamic virtual machine allocation with optimized task scheduling in a cloud computing environment.	Model-View-View Model (MVVM)framework algorithm
Karan D. Patel et al	He introduced a load balancing algorithm by combining two algorithmic program for normalizing the caseload more than the cloud mesh. The job is to inhace scheme performance, greater collateral enactment and basal resolution time.	Load Balancing Algorithm
SangWook Han et al	VM relocation way is recommended to inhace the power effectiveness by thriving the stoutness of VMs exploiting the Knapsack subroutine. In addition, it's conceivable over the schemed way in order to get effective VM expatriation in a abbreviated stint by ornate the KP’s algorithm.	Knapsack Algorithm
Cuijie Zhao et al	On his notes the researcher proposed an Improved- Artificial-Bee’s algorithm, which dopes out the haps of early, huddling and sinking into the vicinal grievous worth in the collocation of hyper spectral figment.	Improved-Artificial-Bee-Colony- Optimization.

Mingyue Feng et al	In his work, he explained how resource allocation in cloud computing is regarded to be a combinatorial optimization issue for a large corporation with a large number of clients and owned resources. This problem is solved using Particle-Swarm - Optimisation technique.	Particle-Swarm-Optimisation (PSO).
Zhifeng Zhong et al	To tackle the job scheduling problem, he used a Greedy's and Particle-Swam-Optimisation (G&PSO) method. It solves the initial shred value of a particle swarm optimization method obtained from a virtual machine-based cloud platform using a greedy approach.	Greedy and Particle Swam Optimisation (G&PSO)

Table 1 summarizes the review work done on the performance of CC but the percentage obtained by this method was high, so this paper overcomes the existing problems.

III. PROPOSED SYSTEM

Individuals and companies are now operating a range of apps that are more cost-effective because of cloud computing. However, as the demand for cloud-based services has risen, a cloud data center's energy usage has soared. As a result, various research efforts have been undertaken to reduce energy usage as well as other parameters like as makes pan, cost, and so on. Many scholars first focused on heuristic techniques for dealing with work scheduling problems in CC's. The primary letch of heuristic algorithms is finding an consummate or near-consummate solvent.

In the original ABC algorithm, an employee bee must update the hive table every time, and the onlooker bee must choose the food source with the most nectar, but in the proffered AABCO, a stalwart retainer bee is inducted to the datacenter to contemporize the modish VM's status, rate of production, retention handiness, cache, and shielding level in the hub. As a result, the cloud environment's Queen-bee's doesn't always desired to look for the apex fettle. This retrieves time because when giving a work to the VM, a judgement is actualized allotted, in case it is crammed or underloaded. Thither be done throughout the project. Resource consumption improves as a result. Migration will be minimized as a result of this. The cloud formation metallurgy is epitomized in Figure 1.

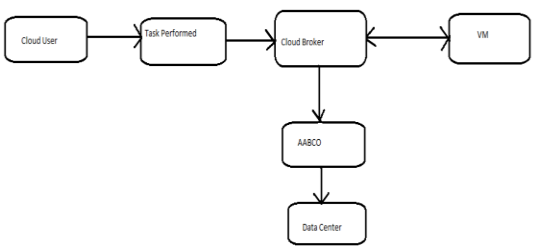

Fig 1: Architecture Diagram

A) Standard-Particle-Swarm-Optimization (SPSO)

For the peculiar conclave on valid stricture, the SPSO algorithm, or Standard Particle Swarm Optimisation, was developed. Based on the most recent and difficult metric assignments available, as well as the most recent standard PSO, which seeks to provide a minimum reference level for analysing the consequences of future PSO algorithm developments. The finest performance of SPSO was obtained with unimodal and separable functions, whereas values shut to the valid possible were obtained with five multimodal tasks. The outcomes reveal that SPSO can quickly converge towards the global optimal region, even for transformed and rotated functions.

However, SPSO obtains the maximum feasible value for most of the compound and multimodal functions within the given limited number of operate evaluations. A particle swarm optimisation approach is similar to solve a multi- detached optimisation problem as single-detached problem . They used suitable weights for the objective functions to calibrate the fitness function value as a single detached function value. The global convergence misfortune was solved utilizing a quantum behavioral particle-swarm-optimisation method that employs utilized wave function to determine the state of the particle in spite of the particle's situation and velocities. In reality, natheless, identifying an foremost concession factor value for each objective function is difficult.

The objective function for SPSO:

1. Create particles that are evenly distributed throughout X.

2. Taking into account the objective function,
evaluate the positions of each particle,
$z=f(x, y) = \sin y^2 + \sin x^2 + \sin x \sin y$

3. If the ongoing locus of the streak is grander than the previous position then update the particle.

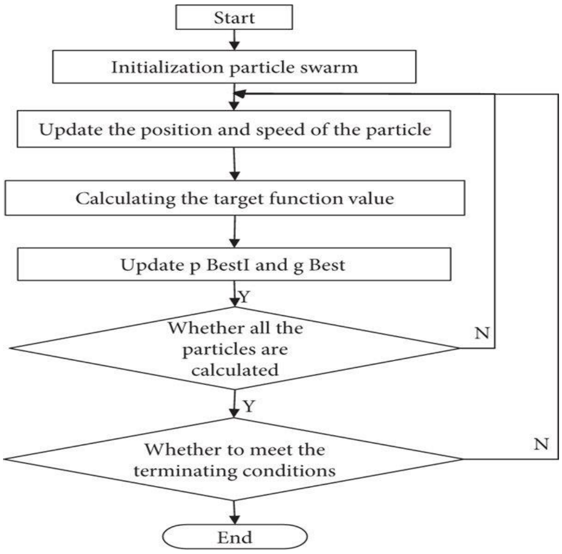

Fig 2: Standard Particle Swarm Optimization Algorithm Flow chart

Initialize population
for $t = 1$: maximum generation
    for $i = 1$: population size
        if $f(x_{i,d}(t)) < f(p_i(t))$ **then** $p_i(t) = x_{i,d}(t)$
            $f(p_g(t)) = \min_i (f(p_i(t)))$
        end
        for $d = 1$: dimension
            $v_{i,d}(t+1) = wv_{i,d}(t) + c_1 r_1 (p_i - x_{i,d}(t)) + c_2 r_2 (p_g - x_{i,d}(t))$
            $x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$
            if $v_{i,d}(t+1) > v_{\max}$ **then** $v_{i,d}(t+1) = v_{\max}$
            else if $v_{i,d}(t+1) < v_{\min}$ **then** $v_{i,d}(t+1) = v_{\min}$
            end
            if $x_{i,d}(t+1) > x_{\max}$ **then** $x_{i,d}(t+1) = x_{\max}$
            else if $x_{i,d}(t+1) < x_{\min}$ **then** $x_{i,d}(t+1) = x_{\min}$
            end
        end
    end
end

Fig 3: Standard Particle Swarm Optimization Algorithm

4. Find the best particles

5. Updates the velocity of the particles.

$$V_i^{t+1} = V_i^t . W + c_1 U_1^t \left(P_{b_1}^t - P_i^t\right) + c_2 U_2^t (g_b^t - P_i^t)$$

6. Pull the particles to the new orientation.

$$P_i^{t+1} = v_i^{t+1} + P_i^t$$

7. Continue step 2 until the stop criteria are met.

B) Artificial-Bee-Colony (ABC)

Artificial-Bee-Colony simply before said as ABC , provides the benefits of clarity, elasticity, and ruggedness. However, the basic artificial bee colony algorithm ultimately reaches the local extreme point, and several researchers have worked to enhance the ABC algorithm to easily search the performance from a variety of aspects.

ABC is an iterative procedure, similar to swarm-based algorithms. The evolution of an ABC population is derived from two key processes: the variation process, which allows for exploration of diverse areas of the request space, and the alternative process, which facilitates the exploitation of preceding experiences. Still later on the fact is that the population has not converged to a local optimum, ABC has been illustrated,
sometimes cease progressing toward the global optimum. The ABC method is divided into four phases: initialization, hired bees, observer bees, and scout bees.

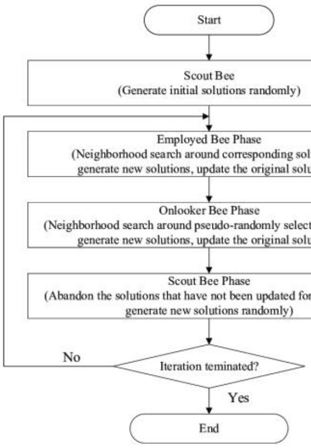

Fig 4: Flowchart of ABC Algorithm

The ABC process is divided into four phases: initialization, hired bees, observer bees, and scout bees, each of which is described below:

Delivered a dull dimensioned operate F & the demos of n connected bee's (i.e., n = neb + nob), Initially, and in the act of spying an EB i (i ∈ [1...neb]) is given an irregular place in the search space pi = (xi 1,...,xi dim). The whole lot EB dares to enhance it's exhibit situation at the beginning of an iteration by generating a new brooms situation pi utilizing the after adjacent quest reign.

$$p* i = ((xi\ 1, \dots ,xi\ j) + r)and(-1,1)(xk\ j - xi\ j), \dots ,xi\ dim) \quad (1)$$

Where j ≤ dim and k = I intends a casually pulled EB, & rand(-1,1) is an authentic favored arbitrary numbered obtained from an homogeneous partition midst -1and1. It's worth noting that Equation 1 only impacts one dimension. Every EB chooses in-case to negate pi in aid of pi utilizing the greedy choice technique described below.

$$pi = pi \text{ if } F(pi) > F(p_* i) \text{ else } I * p \quad (2)$$

Here F(p) raises the suited position of p. Following every EB updates its position, each OB utilizes classic choice to choose part of the present EB positions in specified a way that the prospect of Pi chosing the EBi situation is pi.

```
1: Each bee is placed at a random position in
the search space
2: While, stop criteria are not done; do
3: For all busy bees; do
4: If # is occurred at same position = 1
then
5: Select a random position
6:Else
7: find the better position
8: If a better position is found
Then;
9: Move from the current → found position
10:   end if
11:  end if
12:   end for
13:  for all onlooker bee do;
14:   Select employed bee & move to that
position
15: Else improve the position
16: End for
17: end while
```

Fig 5: ABC Algorithm

$$Pi = F(pi) / \sum neb\ k{=}1\ F(pk) \quad (3)$$

After OB owns selected the locus of an EB, it uses Equation 1 trying to discover a good and better location. If the OB settles on a better location, the related EB adjusts its position as mentioned before. This program keeps tracking the steps taken by an EB that stays in the same location. When an EB's take numerous steps at the rear site enters a limit
$l \geq 1$, it cops out it and searches for a new location.

C) Advanced Artificial Bee Colony Optimization (AABCO)

In the unique ABC subroutine, the proletarian bee need to modify the honeycomb table each time, and the onlooker bee's need to pick up the meal source with the the vast majority of nectar, but in the schemed AABCO, a committed proletarian bee is labelled to the data-centric to modify the common VM's rung, expertise draft, retention affability, larder, and shielding level in the honeycomb table. As a result, the cloud environment's honey bee does not forever covet to endless for the finest lustiness.

This emancipates the moment since a judgments are made depending on whether the VM is overloaded or underloaded when a job is assigned to it. This will be done for the duration of the job. As a result, resource utilization improves. As a result, task migration will be reduced. Figure 1 shows the cloud scheduling architecture. The AABCO algorithm's nature, as stated above, minimizes both completion time and cost. It also assures the data center's load balancing. Migration of tasks is also decreased.

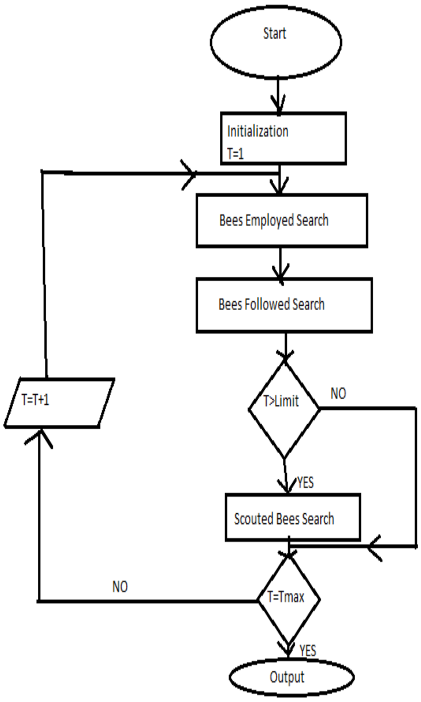

Fig 6: Flowchart of AABCO Algorithm

A task in a venture is portrayed by the organized acyclic-graph DAG (T, D). Where position T = {t1, t2 ,...tn}. D indicates interdependence between tasks and represents a task set. The priority restriction d (i, j) states that job ti must be completed before task tj could begin. Tj's direct ancestor is ti, who is symbolized by pre (ti), and his direct descendants are suc (ti). A task can receive one or more inputs, and when they're all finished, the job's implementation begins.

This process contains task heterogeneity, which might be memory-intensive, CPU-intensive, or data-intensive. You may need to distribute VMs depending on the nature of the task in order to conduct the process normally and efficiently. The transfer time is estimated as follows for workloads when the output of task ti-1 is transferred to task
ti:

$$TT\ (t_i) = TD_i/B \text{ ----- } (1)$$

Where TDi represents the entire quantity of input data for ti and B represents the data center's bandwidth. The selected VMys then executes the input data for task ti. In VMys, the task ti execution time is determined as follows:

$$EXT\ (VM^{y}_{s},\ t_i) = WK_i/Cap\ (VM^{y}_{s})\ (2)$$

Millions of floating point operations per second are performed here. Integrity is applied depending on the security requirements of the user and prevents data from being changed without their knowledge. Confidential services are also applied to prevent unauthorized access to the task's data. Therefore, the processing time PT (ti, VMys) is calculated as follows:

$PT\ (VM^{y}_{s,}\ t_{i}) = EXT\ (VM^{y}_{s,}\ t_{i}) + TT\ (t_{i})$

If tasks ti-1 and ti are deployed to run on the same VMys, the transfer time is zero.

start

1. Determine the number of VMs in your data center
2. Initialize the VM with certain heterogeneous parameters
3. do
4. employee Bee associated with each data center updates the hive table
Uses the characteristic parameters of the VM in the current data center
5. Until the data center job is completed
6. For each job given to the data center
do
7. If the task in the job is Work low
8. Set the number of Work low tasks and their dependencies
9. Save the task with the dependencies in the Hive queue
10. For Each task based on the dependency from j = 1 to T
11. For j = 1 to N for each VM
12. Evaluate the execution time of each task on the VM
13. Each task has its own user-specific requirements such as cost, security level, error,
Tolerance, completion time, etc.
14. Onlooker bee compares the current task request with the updated status of the current VM
At the hive table
15 Then assign the task to the best VM in the structure table
16. end for
17. end for
18. From j = 1 to T for each task
19. Update security level, cost, time, etc.
20. end for
21. end if
22. If the tasks in the job are parallel jobs, for each task tj
23. Repeat steps 11-20 without dependency parameters
24. end if end for
25. end for

END

Fig 7: Ant Colony Optimization Plus Algorithm

PERFORMANCE VALIDATION

This allotment shows the execution of the projected AABCO Algorithm. The cloud atmosphere is simulated utilizing the cloud framework. This can be a self-reliant mission or a dependent workflow schedule. Our studies focus on performance with negligible fabricating margins, costs, migrations, and load balancing.

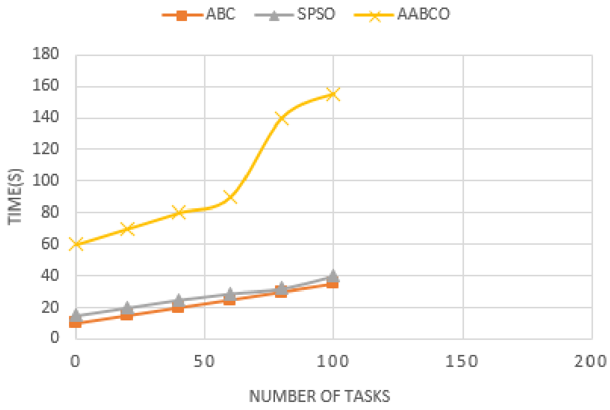

Fig:8 Comparison of execution time

Figure 8 shows the difference in output times is utilizing AABCO and other algorithms. This demonstrates that the proposed AABCO simplifies the execution time compared to other metaheuristic algorithms. Assigning missions via hive table the data is done by the onlooker bees, reducing quest time and receiving the mission to get the the vast majority of optimal VM. This causes a disagreement in accomplishment duration. The comparison of execution time utilizing AABCO shows 85% fulfillment adaptation.

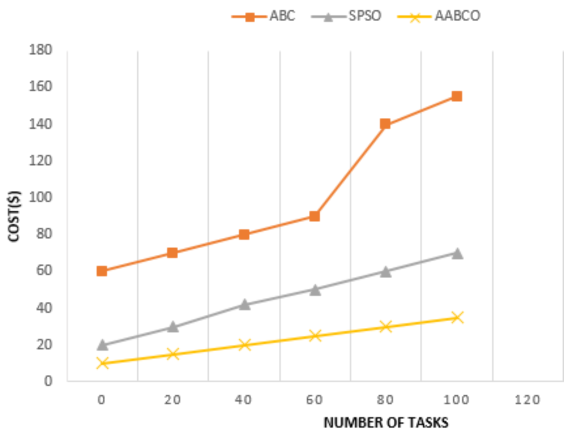

Fig. 9 Cost Comparison

Figure 9 shows a cost comparison when using AABCO with other algorithms. The formed AABCO algorithm gives 55.94% cost increase to that of the ABC algorithm due to user provisions are taken into account when assigning a mission to a VM and the the vast majority of proper VM is assigned to the piece of work.

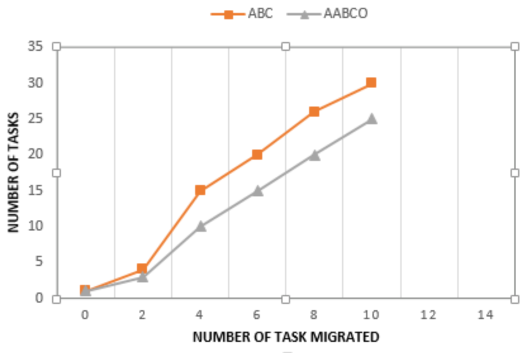

Fig. 10 Number of task migration

Figure 10 shows the number of mission migrations.

As said earlier, AABCO checks the load on each VM prior to assigning a mission to a specific VM. This type of tasks ensures ensures a balanced state across all VMs in the details center, reduces task migration and enhaces performance by 61.52% compared to the original ABC algorithm.

VI) Conclusion

Advanced artificial bee colony optimization for efficiency and performance in cloud systems is proposed in this article. The utilization of hive tables in every data center helps with manufacturing margins, costs, security problems, workload migration, and virtual machine load balancing.. The suggested approach outperforms existing artificial bee colony algorithms for cloud users in terms of productivity, effectiveness, and energy usage while retaining work quality In the future, further novel ways to improve safety efficiency and hybridize various optimization techniques may be used.

REFERENCES

1) Martin Middendorf, Andrej Aderhold, Alexander Scheidler, and Konrad Diwold, J.R. González et al. (Eds.): "Artificial Bee Colony Optimization: A New Selection Scheme and Its Performance "NICSO 2010, SCI 284, pp. 283–294, 2010.Springer-Verlag Berlin Heidelberg 2010
2) HONGDONG ZHAO, CUIJIE ZHAO ,GUOZHEN WANG , AND HONG CHEN, "Improvement SVM Classification Performance of Hyperspectral Image Using Chaotic Sequences in Artificial Bee Colony" ,IEEE ACCESS,VOLUME 8, 2020,accepted April 10, 2020, date of publication April 14, 2020, date of current version2020, May 1
3) Imran Ali Khan, Adnan Sohail, Adeel Ahmed, Abdullah Baz, Muhammad Junaid, Hosam Alhakami,"A Hybrid Model for Load Balancing in Cloud using File Type Formatting", VOLUME XX,2017,DOI10.1109/ACCESS.2020.3003825, IEEE Access
4) DEAFALLAH ALSADIE,VOLUME 4, 2016,"A metaheuristic framework for dynamic virtual machine allocation with optimized task scheduling in cloud data centers ",DOI 10.1109/ACCESS.2021.3077901, IEEE Access

5) Mingyue Feng, Xiao Wang, Yongjin Zhang, Jianshi Li, "MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION FOR RESOURCE ALLOCATION IN CLOUD COMPUTING",978-1-4673-1857-0/12/$31.00 ©2012 IEEE
6) Warangkhana Kimpan, Boonhatai Kruekaew;"Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning" ,IEEE Access (Volume: 10),DOI: 10.1109/ACCESS.2022.3149955,09-2-2022
7) Zhenyu Tang, Jiazhou Li1, Luocheng Shen1,a,Yan Wu1, Yi Wang, "Optimization of Artificial Bee Colony Algorithm Based Load Balancing in Smart Grid Cloud",978-1-7281-3520-5/19/$31.00 ©2019 IEEE
8) SUN Yixin, PENG Yunjian*, DENG Feiqi,"Improved SPSO-based Parameter Identification of Solar PV cells I-V Model ",2017 International Conference on Computer Systems, Electronics and Control (ICCSEC),978-1-5386-3573-5/17/$31.00 ©2017 IEEE, Dec 25-27, 2017
9) Andrej Aderhold, Konrad Diwold, Alexander Scheidler, and Martin Middendorf, "Artificial Bee Colony Optimization: A New Selection Scheme and Its Performance", NICSO 2010, SCI 284, pp. 283–294, 2010.Springer-Verlag Berlin Heidelberg 2010
10) AJITH SINGH. N AND M. HEMALATHA,"CLUSTER BASED BEE ALGORITHM FOR VIRTUAL MACHINE PLACEMENT IN CLOUD DATA CENTRE",.ISSN: 1992-8645, E-ISSN: 1817-3195,2005 - 2013 JATIT & LLS. Journal of Theoretical and Applied Information Technology,30th November 2013. Vol. 57 No.3
11) Sarvesh Rai, Monika Rathore, Navdeep Saluja,"Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud",(IJCSIT) International Journal of Computer Science and Information Technologies, 2015, 4128-4132 , Vol. 6 (4)
12) Patel, Tosal M.Bhalodia, Karan D. P" An Efficient Dynamic Load Balancing Algorithm for Virtual Machine in Cloud Computing", Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019), IEEE Xplore Part Number: CFP19K34-ART; ISBN: 978-1-5386-8113-8.
13) Paridhi Naithani, "Genetic Algorithm Based Scheduling To Reduce Energy Consumption In Cloud",5th IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC-2018), 2018, 20-22 Dec,
14) P. Uma Maheswari , M. Roshni Thanka · E. Bijolin Edwin,"An improved efficient: Artificial Bee Colony algorithm for security and QoS aware scheduling in cloud computing environment" ,Received: 15 June 2017 / Revised: 21 September 2017 / Accepted: 25 September 2017,Springer LLC
15) PeiYun Zhang, Member, IEEE, and MengChu Zhou," Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy", IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING,1545-5955, 2017
16) Ram Mohana Reddy Guddeti, Rajkumar Buyya, Shridhar Domanal,"A Hybrid Bio-Inspired Algorithm for Scheduling and Resource Management in Cloud Environment", IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. X, NO. X, JULY 2016,1939-1374.
17) Zhifeng Zhong, Kun Chen, Xiaojun Zhai, and Shuange Zhou, "Virtual Machine-Based Task Scheduling Algorithm in a Cloud Computing Environment", TSINGHUA SCIENCE

AND TECHNOLOGY ISSNll1007-0214ll07/09llpp660-667 Volume 21, Number 6, 2016 December

18) Haibing Guan a ,Yongqiang Gao a, Zhengwei Qi a, Yang Houb, Liang Liu, Elsevier Inc. 2013 "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing"
19) D. Rukmani Devi,P. Sathyaraj, "Designing the routing protocol with secured IoT devices and QoS over Manet using trust-based performance evaluation method", Received: 31 March 2020 / Accepted: 18 July 2020, part of Springer Nature 2020,Springer-Verlag GmbH Germany
20) X. F. Liu,Z. H. Zhan, K. J. Du, and W. N. Chen, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach", in Proc. ACM Genetic Evol. Computer, pp.41-48, 2014.
21) Wang Jiangyong; Peng Jian; Li Hongyou; Wang Junfeng; Liu Tang, "Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres", Volume: 10, Issue: 1,IEEE,DOI: 10.1109/CC.2013.6723884, 2, Dec. 2013