International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

Energy-Aware Autonomic Resource Scheduling Framework for Cloud

Bhupesh Kumar Dewangan*, Amit Agarwal, Venkatadri M.
School of Computer Science and Engineering
University of Petroleum and Energy Studies, Dehradun, India
*Corresponding author: bhupesh.dewangan@gmail.com

Ashutosh Pasricha
Schlumberger Pvt. Ltd., New Delhi, India

(Received August 24, 2018; Accepted September 26, 2018)

Abstract
Cloud computing is a platform where services are provided through the internet either free of cost or rent basis. Many cloud service providers (CSP) offer cloud services on the rental basis. Due to increasing demand for cloud services, the existing infrastructure needs to be scale. However, the scaling comes at the cost of heavy energy consumption due to the inclusion of a number of data centers, and servers. The extraneous power consumption affects the operating costs, which in turn, affects its users. In addition, CO_2 emissions affect the environment as well. Moreover, inadequate allocation of resources like servers, data centers, and virtual machines increases operational costs. This may ultimately lead to customer distraction from the cloud service. In all, an optimal usage of the resources is required. This paper proposes to calculate different multi-objective functions to find the optimal solution for resource utilization and their allocation through an improved Antlion (ALO) algorithm. The proposed method simulated in cloudsim environments, and compute energy consumption for different workloads quantity and it increases the performance of different multi-objectives functions to maximize the resource utilization. It compared with existing frameworks and experiment results shows that the proposed framework performs utmost.

Keywords- Cost, Cloud computing, Energy, Optimization, Resource scheduling, SLA, QoS.

1. Introduction

Optimization of resources in the cloud is a set of process, which maximize the utilization of the available resources (Preethi et al., 2014). This technique improvises the efficiency of the system by consideration of multi-objective functions. If available resources are well utilized through the optimization process, then execution time and the cost will be minimized, and though this the power consumption can be optimized. The current optimization algorithm is having static behavior like Genetic Algorithm, Particle swarm optimization algorithm (Kennedy and Eberhart, 2011), Ant colony algorithm and Bacterial foraging optimization. The future technique is intelligence and optimization is need to be dynamic. In advance to this, Self-optimization is the process to improvise the performance of the system by adopting intelligence techniques. It automatically finds the optimal solution for resource provisioning.

Gradient-based (Moller, 1993) and gradient free solvers are two algorithms, which solve the optimization problem. Gradient-based solver use derivatives to find the optimum value of a function. It is widely used, fast and scales well but it requires smooth continues function gradient, cost of computing gradient is expensive and local minima. Where gradient-free algorithm does not require derivatives, this means they can be used for optimization problem (Bose and Pain, 2018) where obtaining of derivatives are difficult. This can include the function discrete, discontinues or noisy. This means gradient free algorithm is very flexible and the types of the problem applied too. It is much slower than a gradient-based algorithm; the common gradient free algorithms are an

exhaustive search, genetic algorithm, particle-swarm, simulated annealing, Nelder-mead (Dennis and Woods, 1987) and other nature-inspired algorithms. In addition, Exhaustive search- try every possible solution and pic best one but this may work for a small problem. In the cloud, many evolutionary search optimization techniques have been introducing, but still, there is some scope to enhance to search the optimal solution. These may achieve by involving adding an important function like cost, energy, performance, SLA violation rate, execution time, carbon-di-oxide emission rate, power consumption and QoS (Dewangan et al., 2018). However, in previous works, the maximum three objective functions have been considering to obtain the optimal solution. If the more than three multi-objective (Tamura and Yamada, 2017) function considers for solving approach, then the more accurate optimal solution can be achieved. Some solving approaches in current techniques have been discussed and obtain their different kind of methods and results of the proposed model, Energy-Aware Autonomic Resource Scheduling Framework for Cloud (ESCORT) with possible improvements.

2. Related Work

A state of art survey has been done to study in the area to find the optimal solution for cloud resource scheduling, which can be categorized as follows:

2.1 Search Based Algorithm

Tri-Objective Resource Optimization Algorithm (Alex and Yamini, 2017), this algorithm is a search-based algorithm, which is maximizing the resource utilization with low power consumption; minimize the cost and carbon-dioxide emissions. The execution way of TROA is to select the new result if it is better than the current result or if both results are, equal then it obtains the new result in the subset.

ABRA (Choi and Lim, 2016), Issues of IoT is to productively distribute assets to cloud client with SLA imperatives and to set up SLA contract amongst client and administration. The SLA is an understanding that determines QoS between a supplier and a client. The creator characterized 2 levels of SLA, first is class-construct where QoS is estimated with respect to the premise of execution measurements and second is work based where QoS is estimated utilizing the measurements of individual occupation. For a proficient asset, assignment and suppliers benefit for each time interim regular victor assurance system utilized with the due date limitation. pMapper to streamline cost execution utilizes Bin-pressing calculation in view of an altered rendition of FFD (First fit decreasing) to put VM on servers. To tackle co-designation issue ABRA (Auction-Based Resource Co-Allocation) display utilized which forces punishment costs on unallocated assets after a bartering so as to enhance the asset use.

Short-Term Planning Algorithm (Banu and Saravanan, 2014), the author of STPA prepared model for optimizing the cost for subscription of resource policy. In this determination of how to configure VMs has been providing for time-varying workloads. The optimal number of VMs has been determined and reserved. However, the pricing model is already available; but in this, the price can be optimiz through subscription of VMs.

2.2 Evolutionary Based Algorithm

Genetic Algorithm (Sivanandam and Deepa, 2008)**,** this depends on a developmental model for complex frameworks. This calculation is a worldwide, productive and parallel looking system. It resembles a characteristic advancement technique. Hence, GA (Bhunia et al., 2017) is well reasonable for tackling the asset improvement issue. Numerous point looking is an upgraded

highlight of hereditary calculation. The pursuit space is decreased by utilizing GA (Sahoo, 2017) strategies in asset advancement. In this approach, the hopeful's arrangements are spoken to as far as chromosomes, which are retouched in each stage. GA utilizes the choice approach to discover the survival of wellness work. For every last stage, the most elevated wellness esteem is found and those qualities are acquired to straightaway age.

CCGA (Yusoh and Tang, 2012), Evolutionary Algorithm used to streamline cost in this paper. For the composite asset improvement, the writer expected that asset is now running on their VM's and where, the workload of uses and asset limits are changing after some time, the present position should be adjusted. This paper proposed an answer for a disconnected application position issue, to be specific Application Component Placement (ACP) where the area of the information part is thinking about as one of the central factors and consider as preparing correspondence and capacity necessities.

R-P-E Optimization Technique (Qiu et al., 2017), In this paper it is discussed virtualization which is a center innovation of distributed computing underpins cloud suppliers in creating sound asset planning methodologies to diminish control utilization of a physical server. There exists an imperative tradeoff amongst execution and vitality utilization measurements. The fundamental issue is to adjust the confused unwavering quality execution vitality (R-P-E). For a cure, fundamentally examine R-P-E relationship models and comparing asset advancement advances for cloud administrations utilizing retrying and registration blame recuperation systems. The proposed display envelops Markov models, the Laplace-Stieltjes change, a Bayesian approach and a recursive strategy.

OCRP (Chaisiri et al., 2012), the OCRP algorithm is proposed to limit the aggregate cost for provisioning assets in a specific time. A heuristic technique for benefit reservation and K-closest neighbor's calculation was connected to anticipate the request of assets. Ideal virtual machine position (OVMP) calculation can yield the ideal answer for both asset provisioning and VM situation in two provisioning stages.

2.3 Autonomic Algorithm

Soccer (Singh et al., 2016), it is an automatic optimization technique, which is based on energy optimization for cloud resources in data centers. It considers the energy parameter as a QoS and reduces the energy consumption automatic.

GRASP (de C. Coutinho et al., 2014), in this paper it is discussed control productive plans for reserving, question preparing, and warm administration are required because of the expanding measure of waste warmth that server farms disseminate for application administrations. Moreover, they have estimated, how many shoppers can utilize the transfer speed, CPU, and memory at any given time for the duration of the day. Application schedulers have diverse strategies shift as per the goal work: limit add up to execution time, limit installment cost to execute, limit the interest for control while keeping up an administration level certification for the shoppers, adjust the heap on assets utilized while meeting the due date requirements of the application et cetera. Asset allotment issues in mists are NP-difficult issues, so there are no productive strategies to unravel them. Concerning the cloud customer point of view, registering assets can be obtained by reservation or on-request designs. To explain it, a number programming definition and a heuristic in light of Greedy Randomized Adaptive Search Procedure (GRASP). CHOPPER (Gill et al., 2017) proposed autonomic resource management with self-characteristics to optimize energy and cost.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

2.4 Nature Based Algorithms

Ant Colony Optimization Algorithm (Al Salami, 2009), it is a randomized probabilistic procedure, and used to locate an ideal course which is like the ants to discover the course for their nourishment. For the most part, ants deliver some pheromones when they discover a course. This course is pulled in by additional ants that coordinate to locate the ideal course. In distributed computing condition a cloud specialist organization gives diverse cloud assets on various virtual machines. Here ACO assumes an essential part for streamlining of cloud assets. Virtual machines are considered as hubs, ants are expected as operators and assets are like the nourishment. Insect goes among the hubs and allocates the asked for occupations to the cloud assets.

Particle Swarm Optimization Algorithm (Chen and Yu, 2005), it is a computational strategy which has a place with the super arrangement of swarm insight. In this calculation is self-adaptive iterative strategy is utilized to improve an issue. The hopeful arrangements are iteratively handled with regard to a given measure of value. In PSO, the populace of applicant arrangements is alluded to as 'particles'. The improvement in PSO includes the development of these particles around in the pursuit space. The development is guided by basic numerical formulae over the molecule's speed and position. Every molecule nearby best known position impacts its development. Notwithstanding this, it is moreover guided toward the best known positions in the hunting space, which are refreshed as better positions are found by different particles. The swarm should focalize towards the ideal arrangement by these molecule developments. Like Ant state calculation, Particle swarm calculation is additionally a Metaheuristic calculation. A basic type of calculation works by ginning a populace of the hopeful arrangements (the Swarm). Every competitor arrangement is practically equivalent to a molecule in the framework. The framework should be the arrangement space.

Bacterial Foraging Optimization Algorithm (Das et al., 2009), it depends on hyper-heuristics technique. In this calculation, the incomplete outcome is signified by a bacterium and the movement of the bacterium as heuristics. The improvement in bacterial scavenging calculation lies in a process like chemo taxis, swarming, proliferation, end, and dispersal. The point of the calculation is the manner by which the creature finds the sustenance is like that the power utilization per time (P/T) is expanded. The state of art comparative analysis of different optimization techniques are presenting in Table 1 and the optimization technique is categorized into four categories namely search-based, Evolutionary-based, Nature-based, and Autonomic-based as mentioned in Figure 1.

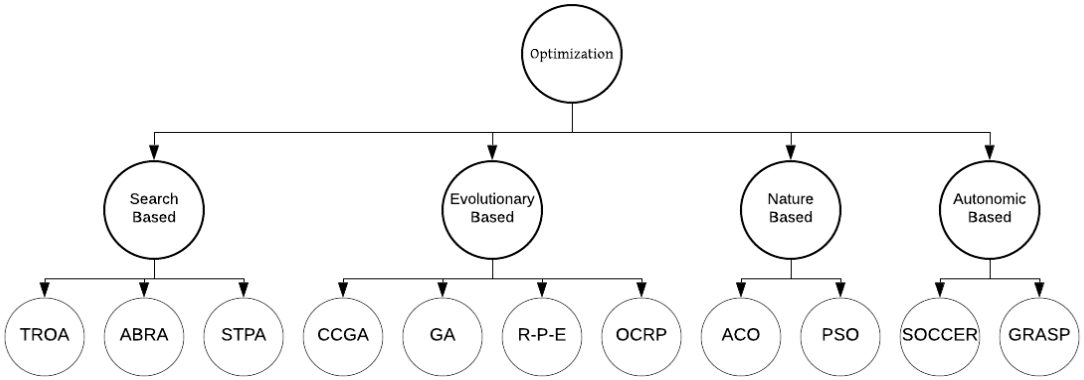

Figure 1. Optimization techniques in cloud

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

The comparison has been carried out based on multi-objective functions.

Table 1. Comparative analysis of different optimization techniques with ESCORT

Performance Metrics	Optimization Techniques								
	TROA	SOCCER	STPA	ABRA	CCGA	OCRP	R-P-E	GRASP	ESCORT
SLA-VR	**No**	**No**	**No**	**Yes**	**No**	**No**	**No**	**No**	**Yes**
QoS	**No**	**No**	**Yes**	**Yes**	**No**	**No**	**No**	**No**	**Yes**
Execution Time	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**Yes**	**Yes**
Cost	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**
Energy	**No**	**Yes**	**Yes**	**No**	**No**	**No**	**No**	**Yes**	**Yes**
Power	**Yes**	**No**	**No**	**No**	**No**	**No**	**Yes**	**No**	**No**
CO2 Emission	**Yes**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**

3. Limitations and Possible Improvements

The resources can be managed based on multi-objective functions and send to the resource pool; then the allocation of the resources carried out according to the QoS requirements. The above study, it is observed that, researcher optimized cost, power, execution time and CO2 emission rate for maximizing the resource utilization in the cloud. In this paper, the Antlion optimization algorithm, which is previously applied in network optimization, applying to optimize the energy consumption to reduce execution cost, time, SLA violation rate, and maximize resource utilization. The novelty of this paper is applying the Antlion optimization technique in resource scheduling if cloud computing to optimize energy consumption and other multi-objective functions. The different nomenclature used in this article has been presenting in Table 2, which is utilizing from equation 1 to 13.

Table 2. Notations

Notation	**Definition**
n	number of workloads
M	number of variables
R_i	failure rate
SV_R	SLA violation rate
w_i	the weight of every SLA
RC_i	request workload completion time
RS_i	workload submission time
SLA_Cost	cost of SLA
M_{Ant}	the position of each ant store in the matrix
M_{OA}	matrix for saving the fitness of each ant
$M_{Antlion}$	is saving the position of each antlion
M_{OAL}	the matrix to store the fitness of each antlion
$R(e)$	random walk
C_s	cumulative sum
e_n	number of iteration
E	step size
$r(e)$	random function
$F1, F2, F3$	objective functions
o_i, p_i	minimum and maximum of the random walk of i-th variable respectively

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

4. Methodology

4.1 Formal Optimization Model

The flow of formal optimization model is representing through Figure 2 and the architecture of the proposed model ant-lion based model is mentioned in Figure 3. In general, to obtain the optimal solution, population and fitness values has to be calculated. On the basis of these values if the method meets the optimization criteria then it will stop to search the solution and resources will be sent to the pool else the process need to be regenerate the population and calculate the fitness values.

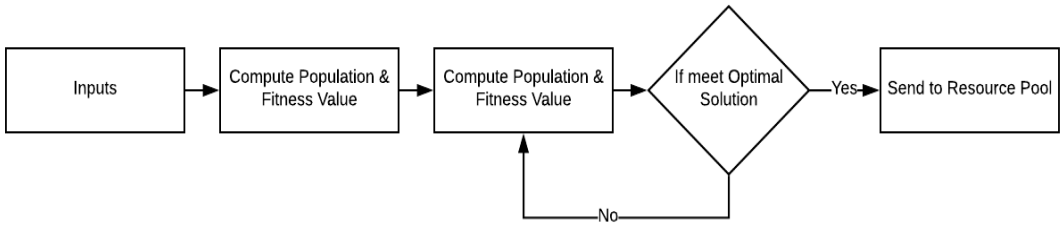

Figure 2. Flow of formal optimization model

4.2 Architecture of Antlion Based Optimization Model

The architecture of proposed Antlion based optimization architecture is presenting in Figure 3, where cloud users are submitting the workload to the task manager for scheduling the resources.

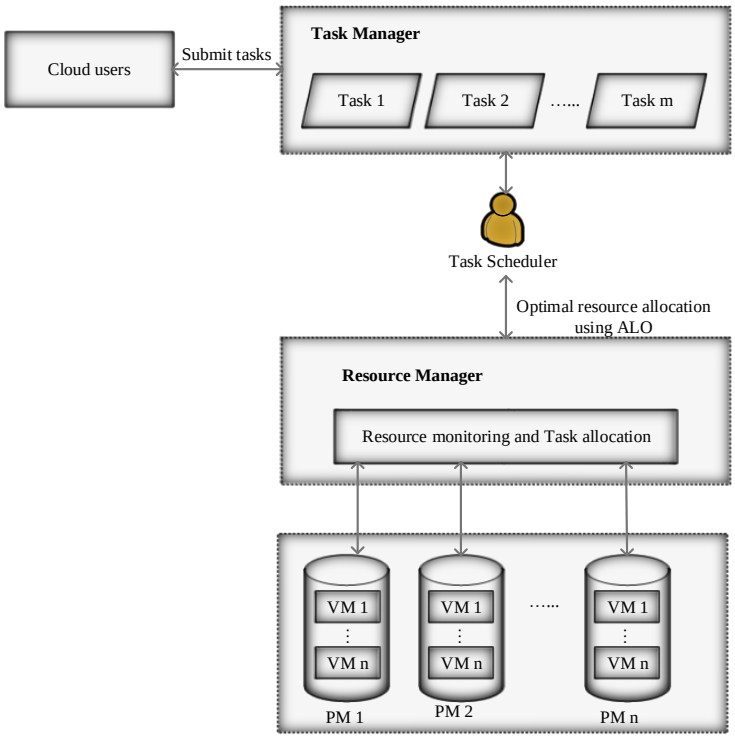

Figure 3. Proposed antlion based optimization model ESCORT

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

4.3 Resource Management

Resources will be managing through the calculation of performance metrics like failure rate, cost, energy, and SLA violation rate.

The failure rate of resources is a failure rate of every workload divisible by a number of workloads.

$$\text{Failure rate} = \frac{failure(R_i)}{n} \tag{1}$$

SLA violation rate is calculating by multiplication of failure rate and weight of every SLA.

$$SVR = failure\,rate * \sum_{i=1}^{n} w_i \tag{2}$$

Execution time is the difference between workload completion time and workload submission time.

$$Execution time = \sum_{i=1}^{n} \left(\frac{RC_i - RS_i}{n}\right) \tag{3}$$

SLA cost is the cost of SLA violation rate and execution time. If both are high, then the cost will increase and vice versa.

$$SLA\_Cost = SV\_R * \sum_{i=1}^{n} Execution time \tag{4}$$

4.4 Resource Allocation

In this paper for the best VM selection used ant lion optimization (ALO). After the centroid positions are finished in the grouping procedure, consider the hubs which are at the closest separation and furthermore the following closest separation from the centroid by utilizing ALO optimization. In this method, the initialization is done by using a random selection of antlions and ants (VM). After that, the multi-objectives (resource utilization, energy, and cost) is calculated between each VMs, from that the fitness value of ant and ant lion is calculated and ant lion fitness value sorted to find the best ant lion is named as elite antlion (best VM).

Ant's VMs are moved at arbitrarily to look for their nourishment. An arbitrary walk is utilized to demonstrate their developments utilizing total entirety work and an irregular capacity connected for an alternate emphasis. The random walk of ant can be calculated by equation (5)

$$R(e) = [0, C_S(2r(e_1) - 1), C_s(2r(e_2) - 1), ... C_s(2r(e_n) - 1)] \tag{5}$$

$r(e) \rightarrow$ is an arbitrary capacity takes the esteem '1' if the irregular number produced is more prominent than 0.5 else it is set to '0'.

To keep ant arbitrary strolls inside the hunting space to refresh their situation concerning chosen subterranean insect lion utilized min-max standardization. Calculate the min-max normalization by using the equation

$$R_i^e = \{[(R_i^e - o_i)(R_i^e - q_i^e)]/(p_i^e - o_i)\} + q_i] \tag{6}$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

To compute fitness value, consider that the ants are caught in any of the haphazardly chose antlion. The roulette wheel is utilized to choose the best subterranean insect lion from its populace in view of wellness count. This very fit insect lion to get the ants with high likelihood.

$$VM_U = VM.CPU_U + VM.BW_U + VM.RAM_U \tag{7}$$

Therefore, the first objective function is,

$$F_1 = \max\{VM_u\} \tag{8}$$

Computation of Energy consumption will be done through

$$S = k * \max + (1-k) * \max * VM_u \tag{9}$$

Therefor the second objective is,

$$F_2 = \min\{s\} \tag{10}$$

And the cost of the resource is,

$$Cost\ C = c * S \tag{11}$$

Where c is the cost of 1 kW power.

Therefor the third objective function is,

$$F_3 = \min\{C\} \tag{12}$$

Antlions are building their trap with their wellness esteem. On the off chance that the subterranean insect is inside the trap, at that point it shoots the sand outside the focal point of the pit. It influences the subterranean insect to slide down when it attempts to escape from the antlion.

Solution updating, ant achieves the base of the pit, and after that the antlion pulls it in the sand and eats its body. Next, the antlion adjusts its position and fabricate another trap to get another prey.

Antlion gets an insect, when the wellness of antis more noteworthy than the wellness of antlion. It is given by,

$$AL_j^t = A_i^t \text{ If } \quad fitness(A_i^t) > fitness(AL_j^t) \tag{13}$$

The algorithm of the above method is presenting in the next section.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

4.5 Enhanced Antlion Optimization Algorithm for Cloud

The enhanced Antlion Optimization based scheduling algorithm involves four modules namely, workload priority, QoS computation, best VM's, and assigning the workloads to the VM's.

1. Start
Workload Priority
2. W: a set of 'n' workloads
3. Resource initialization: Datacenter, host, VM
4. ALO initialization: Number of Ant lions, Number of ants, MaxIter=100;
5. Calculate the workload priority based on the priority approach
6. End
QoS Computation
7. For each workload, Compute QoS metrics using the equation
8. resource utilization = CPU Utilization + Bandwidth Utilization + RAM Utilization
9. set maximum amount of power host max = 1 and k = 0.5
*10. energy consumption s=k*max+(1-k) *max*utilization*
*11. resource cost = cost of 1 KW power * Energy consumption*
12. End
Best VM's
13. add Energy consumption, resource cost and utilization for each workload
14. compare best VM's value of each workload
15. Sort in ascending order
16. if (Best VM's(w1) > Best VM's(w2))
17. apply swap
18. else next comparison till the list sorted
Assigning the workloads to VM's
19. If MaxIter reached
20. For W= n to 1 // n is the number of VMs
21. Choose a nth task from the sorted list
22. Compute fitness function
23. Assign nth task to selected VM
24. Else, Update the ant lions position randomly. And Perform the ant lion algorithm
25. Return as the solution, the solution attributable to the most expert ant
26. End
27. The primary subterranean Ant in the lion subterranean Ant is mapped to every one of the mappings performed in stage 27
28. Mapping alternate ants in the insect lion calculation, with arbitrary task of every single existing assignment to various kinds of apparatus accessible.
29. End

5. Implementation Results and Analysis

The proposed model ESCORT has been implemented and simulated in clouds. In this simulation, 100 workloads are submitted to the system, and 20 VM's are available in the resource pool to submitted workloads. The workloads are clustered through k-mean clustering algorism and it is clustered to four centroid values C1, C2, C3, and C4. The workloads are sorted and arranged according to user priority, i.e. execution time.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

5.1 Computing Energy Consumption

The energy consumption rate for each workload has been computed and taken an average of each iteration. The outcome of this experiment is presenting in Table 3.

The common workloads are computed in existing frameworks, SOCCER and CHOPPER and observed that when the workloads are increasing the energy consumption rate are also increasing as mentioned in Figure 4.

Table 3. Energy consumption rate (KWh)

Number of workloads	200	400	600	800	1000
ESCORT	9.7 kw	13.5 kw	20 kw	27kw	37 kw
CHOPPER	12 kw	15 kw	26.5 kw	32 kw	41.3 kw
SOCCER	10.5 kw	14.7 kw	23 kw	29.4 kw	38.5 kw

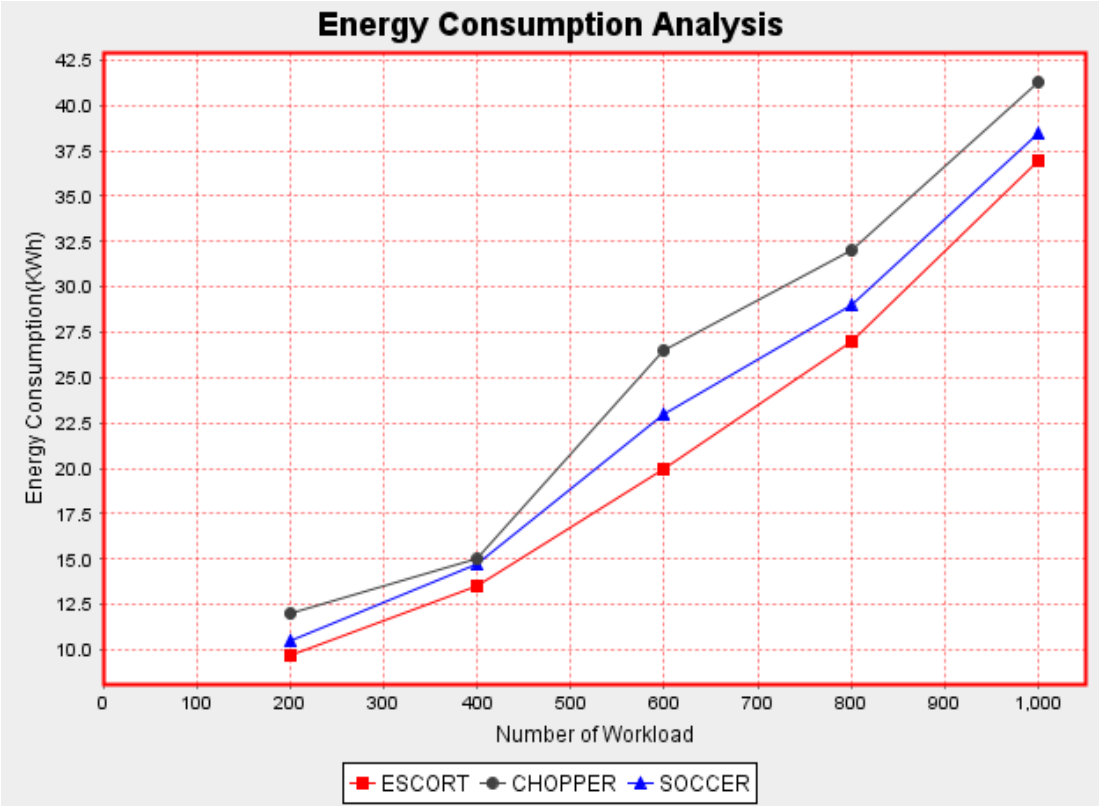

Figure 4. Energy consumption analysis

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

The above results are evidence that the proposed method is better in optimization when it compared with energy consumption rate with existing models. Cloud user submitted 200 workloads to allocate 20 VM's as mentioned in Figure 4 and recorded the consumption rate for 400, 600, 800, and 1000 workloads. ESCORT is cost-effective, produce less SLA violation rate and consumes less energy as compared to existing methods.

5.2 Computing SLA Violation Rate

SLA violation rate and Execution time for each workload have been computed and taken an average of each iteration. The common workload dataset is computed in ESCORT, SOCCER and CHOPPER, recorded as mentioned in Table 4, when the execution time of workloads is increasing the SLA violation rate are also increasing.

Table 4. SLA violation rate

Number of workloads	10	20	30	40	50	60
ESCORT	200	250	300	450	650	695
CHOPPER	300	350	500	680	800	900
SOCCER	250	300	400	620	750	870

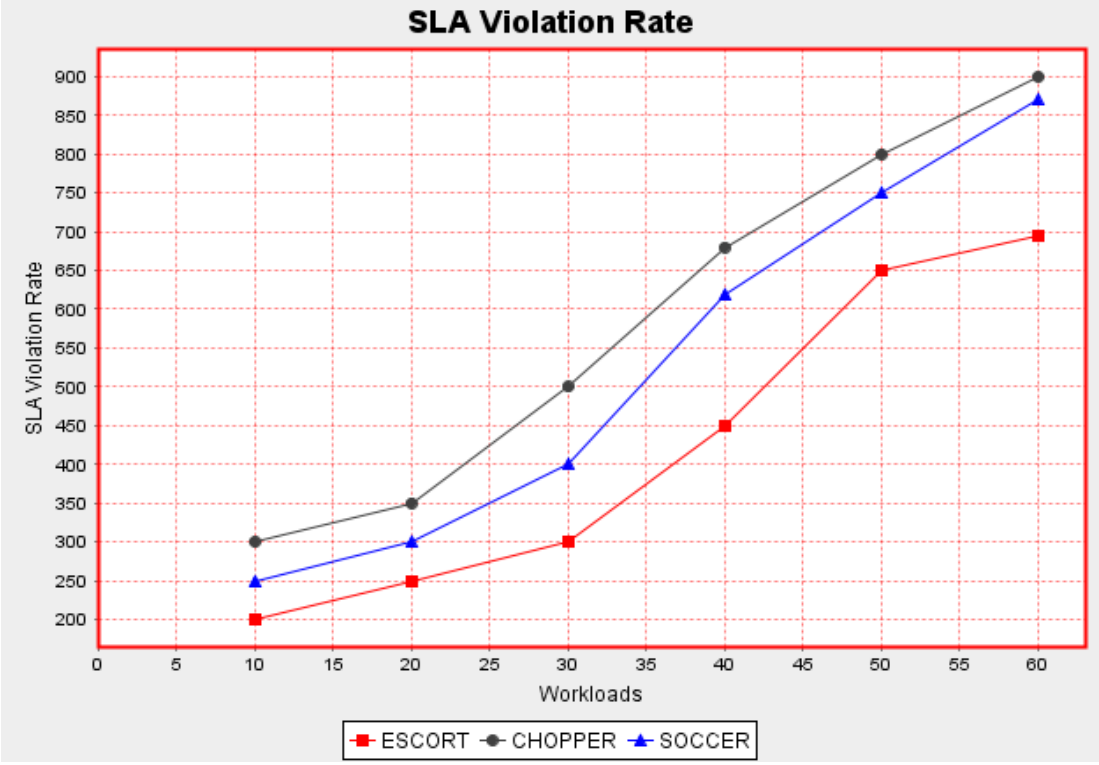

Figure 5. SLA violation rate

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 1, 41–55, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.1-004

The average values of SLA violation rate have been recorded for the proposed method and the other two existing methods. When the number of workloads is increasing the SLA violation rate are also increasing. The graphical representation of different methods is mentioned in Figure 5.

5.3 Computing Execution Cost

SLA violation rate and Execution cost for each workload have been computed and taken an average of each iteration. The same workloads are also computed in existing frameworks, SOCCER and CHOPPER. When the execution cost of workloads is increasing the SLA violation rate are also increasing as results recorded in Table 5.

Table 5. Execution cost ($)

Number of workloads	0-10	10-20	20-30	30-40	40-50	50-60
ESCORT	18	20	25	38	40	55
CHOPPER	30	35	48	58	63	75
SOCCER	25	30	40	50	60	70

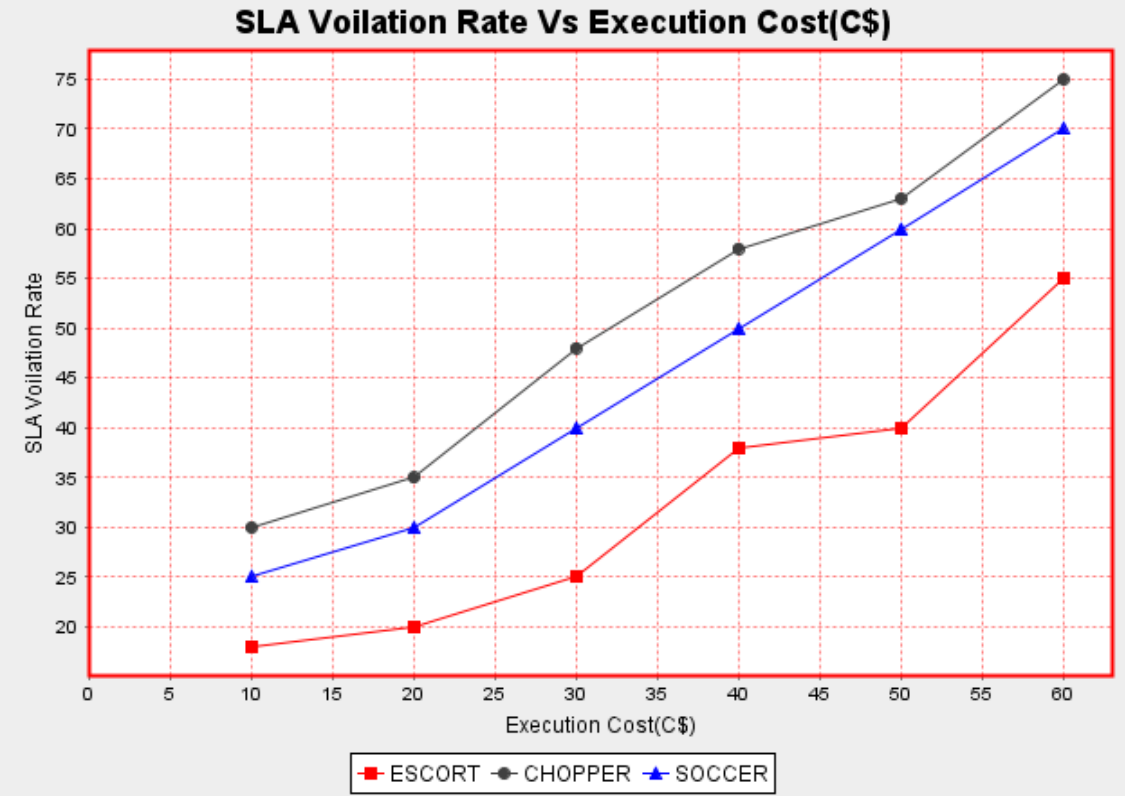

Figure 6. Execution cost ($)

The average values of execution cost have been recorded for the proposed method and the other two existing methods. The graphical representation of different methods is mentioned in Figure 6.

5.4 Comparative Analysis

The average of different multi-objectives performance metrics has recorded in Table 6. Overall evaluation of ESCORT founds that, it producing efficient results as compared to existing frameworks.

Table 6. Analysis of performance metrics (average)

Frameworks	SLA Violation Rate (Workload 10 to 60)	Execution Cost (Workload 10 to 60)	Energy Consumption (Workloads 200 to 1000)
ESCORT	424.16	$ 32.66	21.44 KW
CHOPPER	588.33	$ 51.5	25.36 KW
SOCCER	531.66	$ 45.83	23.14 KW

5.5 Scheduling

The workload submitted by the user to ESCORT is clustered and sorted as per the user priority, then best VM value computed through CPU, RAM, Bandwidth, SLA violation rate, Execution cost, and Energy consumption rate, and sorted in the resource pool. By using round-robin algorithm all workload has been assigned to best VM's according to the user priority.

6. Conclusion

In this paper, the ESCORT framework presented to optimize energy consumption, execution cost and SLA violation rate. This framework is implemented and simulated in the cloudsim environment. The result may vary when it will be implemented in the real cloud. In this, it is presented a detailed flow of ESCORT that, how to apply the proposed framework to minimize the energy consumption, SLA violation rate, execution time, and cost to maximize the performance for scheduling in the cloud. The experimental results show that ESCORT average energy consumption is 21.44 KW, while other techniques, which is compared, consume energy 23.14 KW and 25.36 KW. The analysis of proposed work with other techniques ensures that ESCORT performs utmost. ESCORT simulation results minimize the SLA violation rate, execution cost and maximize the utilization of the resource. The limitation of this work is, no filtering techniques are applied for malicious workload, in future, it will be implemented to improvise the proposed work.

Conflict of Interest

The authors declare that there is no conflict of interest to declare for this publication.

Acknowledgement

The authors will like to thank anonymous reviewers for their constructive comments to improve the quality of this paper.

References

Al Salami, N. M. (2009). Ant colony optimization algorithm. *UbiCC Journal*, *4*(3), 823-826.

Alex, G. M., & Yamini R. (2017). Comparision of resource optimization algorithms in cloud computing. *International Journal of Pure and Applied Mathematics*, *16*(21), 847-855.

Banu, M. U., & Saravanan, K. (2014). Optimizing the cost for resource subscription policy in IaaS cloud. *International Journal of Engineering Trends and Technology*, *6*(5), 296-301.

Bhunia, A. K., Duary, A., & Sahoo, L. (2017). A Genetic Algorithm based hybrid approach for reliability-redundancy optimization problem of a series system with multiple-choice. *International Journal of Mathematical, Engineering and Management Sciences*, *2*(3), 185-212.

Bose, G. K., & Pain, P. (2018). Metaheuristic Approach of Multi-Objective Optimization during EDM Process. *International Journal of Mathematical, Engineering and Management Sciences, 3*(3), 301-314.

Chaisiri, S., Lee, B., & Niyato, D. (2012). Optimization of resource provisioning cost in cloud computing. *IEEE Transactions on Services Computing*, *5*(2), 164-177.

Chen, G. C., & Yu, J. S. (2005). Particle swarm optimization algorithm. *Information and Control, 34*(3), 318.

Choi, Y., & Lim, Y. (2016). Optimization approach for resource allocation on cloud computing for IoT. *International Journal of Distributed Sensor Networks*, *12*(3), 3479247.

Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In *Foundations of Computational Intelligence Volume 3* (pp. 23-55). Springer, Berlin, Heidelberg.

De C. Coutinho, R., Drummond, L. M., & Frota, Y. (2014). Optimization of a cloud resource management problem from a consumer perspective. *Euro-Par 2013: Parallel Processing Workshops*, 218-227.

Dennis, J. E., & Woods, D. J. (1987). Optimization on microcomputers: the Nelder-Mead simplex algorithm. In *New Computing Environments: Microcomputers in Large-Scale Computing (pp. 6-122), SIAM* Philadelphia.

Dewangan, B. K., Agarwal, A., Venkatadri, & Pasricha, A. (2018). Resource scheduling in cloud: a comparative study. *International Journal of Computer Sciences and Engineering. 6*(8). 167-173.

Gill, S. S., Chana, I., Singh, M., & Buyya, R. (2017). CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing, 21(2*), 1203-1241.

Kennedy, J. & Eberhart, R. (2011). *Particle swarm optimization in encyclopedia of machine learning* (pp. 760-766). Springer, Boston, MA.

Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, *6*(4), 525-533.

Preethi, B., Kamalanathan, C., Ramesh, S. M., Shanmathi, S., & Bama, P. S. (2014). Optimization of resources in cloud computing using effective load balancing algorithms. *International Advanced Research Journal in Science, Engineering and Technology*, *1*(1). 20-22.

Qiu, X., Dai, Y., Xiang, Y., & Xing, L. (2017). Correlation modeling and resource optimization for cloud service with fault recovery. *IEEE Transactions on Cloud Computing*, 5(1), 1-13.

Sahoo, L. (2017). Genetic algorithm based approach for reliability redundancy allocation problems in fuzzy environment. *International Journal of Mathematical, Engineering and Management Sciences, 2(4),* 259-272.

Singh, S., Chana, I., Singh, M., & Buyya, R. (2016). SOCCER: self-optimization of energy-efficient cloud resources. *Cluster Computing*, *19*(4), 1787-1800.

Sivanandam, S. N., & Deepa, S. N. (2008). Genetic algorithm optimization problems. In *Introduction to Genetic Algorithms* (pp. 165-209). Springer, Berlin, Heidelberg.

Tamura, Y., & Yamada, S. (2017). Dependability analysis tool based on multi-dimensional stochastic noisy model for cloud computing with big data. *International Journal of Mathematical, Engineering and Management Sciences, 2(4),* 273-287.

Yusoh, Z. I. M., & Tang, M. (2012, June). Composite SaaS placement and resource optimization in cloud computing using evolutionary algorithms. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* (pp. 590-597). IEEE.