International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 758-768, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.060

Cloud Resource Optimization System Based on Time and Cost

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

Amit Agarwal
School of Computer Science and Engineering,
University of Petroleum and Energy Studies, Dehradun, India.
E-mail: aagarwal@ddn.upes.ac.in

Tanupriya Choudhury
School of Computer Science and Engineering,
University of Petroleum and Energy Studies, Dehradun, India.
E-mail: tanupriya1986@gmail.com

Ashutosh Pasricha
Schlumberger Pvt. Ltd, New Delhi, India.
E-mail: apasricha@slb.com

(Received August 2, 2019; Accepted March 31, 2020)

Abstract
Resource management in cloud could be a time and cost-effective activity if it is managed property. These resources are accessible and computable which is totally dependent upon the management techniques applied in cloud. In a cloud setting, heterogeneous, vulnerability, and scattering of resources creates many issues of distribution among the workloads which need to be compute. Specialists still face inconveniences to pick the prudent, material and expend less time to execution of resource portion to the cloud. This investigation delineates an expansive composed writing examination of asset administration inside the space of cloud typically and cloud asset administration based on SLA with multi-objective functions like cost and time. In this paper, an autonomic cloud resource-management technique is proposed to resolve identified issues by adopting the self-characteristics mechanism and improved Antlion optimization algorithm and tested in cloudsim toolkit and Aws Ec2 environment. The implementation results of proposed work are the evidence that it is better performing as compared with the existing frameworks, however, the performance evaluation method depends upon the different cloud environment and it may vary.

Keywords- Cloud computing, Autonomic computing, Self-optimization, Fuzzy, Resource scheduling.

1. Introduction

Cloud server and virtual machine perform significant jobs to manage resources in cloud. The utilization of resources (Sutar et al., 2020) is additionally high gratitude to a great deal of execution time and resource failure. To deal with the resource in cloud quickly, the energy consumption use should be minimize and avoid the resource failure. The resource computer equipment plans the approaching cloud remaining tasks upheld the outstanding burdens subtle elements. To begin with, get cloud remaining tasks to plan, so acknowledge material, and available assets and cloud outstanding burdens mapped quickly upheld the arranging approaches. Dispatcher is utilize to dispatch the outstanding tasks at hand for execution (Levitin et al., 2020). The work is send exclusively, if the remaining tasks will be dead about the quality of service QoS parameters said in

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 758-768, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.060

service level agreement SLA. Asset screen is utilize to discover the remaining of arranging of resource like regardless of whether the coveted scope of resource is given or not. QoS screen contains the learning with respect to QoS parameters to discover, regardless of whether every one of the outstanding tasks at hand square measure execution among their particular fluctuate or not. Assume point in time could be a QoS parameter, in this manner duty of QoS screen is to learn regardless of whether outstanding tasks at hand square measure dead before wanted point in time or not. There is infringement of SLA (Dewangan et al., 2018) if work executes once wanted point in time. Streamlining of vitality in cloud asset arranging is about of practicable strategy that limit the vitality utilization and expand the virtual machine (VM) use. On the whole, the strength of asset designation is improve through vitality enhancement. This technique limits the execution time, SLA infringement rate, and worth that expansion the framework execution. The present streamlining ways square measure action static conduct like Genetic-Algorithm, Particle-swarm-advancement rule, Ant-settlement principle and microorganism chase improvement. The static conduct of the framework could wind up in moderate in execution and high in worth. The framework needs the keen framework to advance the vitality and diverse target capacities. This clever method will manufacture the dynamic advancement structures, which improve vitality also as worth and time. The dynamic advancement while not human intercession is execute without anyone else's input qualities. In this manner, vitality is self-streamlined (Chen et al., 2020) and piece. It will precisely understand the best goals for asset provisioning. Adaptation to non-critical failure advancement is an ability of a PC structure, electronic structure or framework to pass on ceaseless organization, regardless of at least one in the entirety of its parts growing short. Adjustment to non-basic disappointment in like manner settle potential organization obstruction known with programming or guideline botches. The reason for existing is to balance dark disillusionment that may happen because of a singular motivation behind disappointment. Fault-tolerant (Dewangan et al., 2016) frameworks shall compose for numerous disappointments. Such structures usually distinguish a disappointment of the computer processor unit, I/O system, memory cards, motherboard, and management provide or framework components. The frustration purpose is recognized, and a support fragment or technique during a flash has its spot with no loss of organization. To make sure adjustment to non-basic disappointment, tries desirous to purchase a load of masterminded computer instrumentality and a helper uninterruptible power provide device. The target is to stay the crash of key frameworks and systems, concentrating on problems known with up time and period (Dewangan et al., 2019). Adaptation to internal failure is offer-programming put in instrumentality, or by some mixture of the two. In this research work, the cloud resources are allocating to the workloads submitted by cloud user based on service level agreement (SLA). The multi-objective function like cost and time are main concern of this research. The proposed system is simulating and evaluating based on above multi-objective parameters and the results are obtain in terms of cost and time, which compared with other existing frameworks, observed the utmost performance.

2. Related Work

A state of art survey has been conduct on SLA, cost and based resource management and illustrating as follows:

2.1 SLA-Based Resource Management

Resource planning supported SLA has been conduct by following papers, automatic SLA-aware asset arranging algorithmic program to decrease SLA encroachment and violation rate that works in cloud virtual condition to execute benefit while not infringement of SLA. This outline joins three parts: benefit agent, understanding arrange and request preparing to execute benefit while not SLA infringement (Kertesz et al., 2011). QoS based resource-arranging system to estimate the execution underneath disparate asset assignment by resize of VM. Multi-Workflows essentially based arranging strategy to improve the arranging achievement rate together with elective QoS parameters. It settles on cloud providers and learning focuses in an exceedingly multi-cloud climate as an administration supervisor upheld QoS parameters (Xu et al., 2009). Execution of the hatchling is screen by Spe-QuloS in discontinuous way, resources are gave only if there ought to emerge an event of benefit necessities capably, and SpeQuloS upgrades quality of execution and envision the execution time of occupation (Delamare et al., 2012). Random-Key Genetic algorithmic program fundamentally based orchestrating game plan to diminish execution time and running expenses and enhance quantifiability. This algorithmic program settled a substitution numerous composite net administration asset designation and arranging disadvantage in an exceedingly half breed cloud situation wherever there are additionally limited local assets from individual mists and various out their re-sources from open mists (Ai et al., 2011).

2.2 Cost-Based Resource Management

Resource arranging bolstered cost led by following papers. Imperative asset arranging algorithmic run for pack of assignments amid which undertaking is picked bolstered FCFS philosophy. This system limits value, fulfillment time and enhances C.P.U. execution nevertheless, because of drawback of starvation, this instrument is not successful (Oprescu and Kielmann, 2010). Contemplated the enhancement drawback forcing condition like execution of occupation isn't pre-emptible and point constrained amid a multi-supplier crossover cloud surroundings bolstered the needs of data transmission, C.P.U. furthermore, memory, the order of non-supplier relocate capable outstanding burdens is finished (Van et al., 2010). DAG (Directed Acyclic Graph) based for the most part assignment arranging instrument to downsize cost and Makespan exploitation two heuristic ways. First system maps errands to the preeminent effective virtual assets exploitation humanist predominance and second procedure is utilize to diminish the money related costs of non-basic assignment in true applications (Su et al., 2013). Bestowed VM based for the most part asset arranging procedure to measure the general estimation of Gang arranging with starvation dealing with, movements, and execution of superior venture applications. To contact upon starvation in arranging procedure, organized line is utilized to search out the need of every application bolstered their coveted point and so forth (Moschakis and Karatza, 2011).

2.3 Time-Based Resource Management

Resource programming supported time has been conduct by following papers. Asset programming strategy to reduce data exchange and process costs, arrange data measure and vitality utilization. Execution of this algorithmic control is assessed exploitation unique execution parameters like scope of lost due dates, esteem sparing and process strength and break down the effect of estimation mistakes on execution (Van et al., 2013). Programming algorithmic lead to anticipate asset needs of occupation exploitation adjusted machine learning based for the most part indicator and fast explanatory indicator and show that anyway this programming algorithmic run is utilize to foresee asset needs to reduce SLA infringement (Reig et al., 2010). Bestowed Partial vital ways based for the most part ICPCP (Abrishami and Naghibzadeh, 2012) IaaS Cloud Partial essential Paths and IC-PCP with point Distribution (IC-PCPD2) to arrangement and timetable enormous work processes. The calculation time is lesser amid this methodology in any case; this is regularly frail to live measurable execution and UTC precisely. Vitality is amount of vitality devoured by an asset to end the execution of work. Esteem is an amount of significant worth will pay in one hour for the execution of work. In time based for the most part asset programming, client and provider talk over

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 758-768, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.060

QoS parameters inside the style of composed material alluded to as SLA. The comparative analysis is presented in Table 1.

Table 1. Comparative analysis of SLA, Cost and Time-based resource management techniques

System	ET	Cost	EE	RU	OC	SLA	QoS	CE
(Sutar et al., 2020)				√		√		
(Levitin et al., 2020)	√				√			
(Chen et al., 2020)			√					
(Delamare et al., 2012)		√	√				√	
(Ai et al., 2011)		√		√				
(Oprescu and Kielmann, 2010)		√						
(Van et al., 2010)		√			√			√
(Su et al., 2013)	√	√	√			√	√	
(Moschakis and Karatza, 2011)	√		√				√	
(Van et al., 2013)	√	√				√	√	
(Reig et al., 2010)	√	√						

3. Optimization Method

In this research, antlion optimizer (ALO) (Mirjalili, 2015) is used for the fittest VM selection. This algorithm is also used in healthcare, construction and civil engineering to find the optimal solutions. According to this, The ALO imitates collaboration among virtual machines VM's and tasks D in the system. To demonstrate such collaborations, task D are required to submit over the pursuit system, and VM's are permitted to chase them and allot to compute the task D.

3.1 Parameters

$z(t) = Random\ walk\ function$,
$n = maximum\ number\ of\ iteration$,
$cumsum = calculate\ the\ cumulative\ sum$
$d(t) = stochasticfunction$,
$t = step\ of\ random\ walk$,
$A_D = position\ of\ each\ ant$,
$A_{oD} = Stores\ fitness\ value\ of\ each\ ant$,
$A_{VM} = position\ of\ each\ antlion$,
$A_{oVM} = Stores\ fitness\ value\ od\ each\ antlion$

3.2 Mathematical Model

Since ants move stochastically in nature when searching for food, a random walk is chosen for modelling ants' movement as follows:

Since task D submitted stochastically in nature while looking for any computing resources like VM, an irregular walk is picked for displaying task D development as pursues and presented in eq. (1):

$$z(t) = [0, cumsum(2d(t_1 - 1), cumsum(2d(t_2 - 1), cumsum(2d\ (t_3 - 1) \ldots \ldots cumsum(2d(t_n - 1), \quad (1)$$

Computing time span can be presented in eq. (2):

$$d(t) = \begin{cases} 1\ if\ random > 0.5 \\ 0\ if\ random < 0.5 \end{cases} \quad (2)$$

The random task D is allocated to virtual machine VM and fitness of each VM is obtained through CPU, RAM, and Bandwidth Utilization for each VM's. The following optimization equation is used to store the position of task D and presented in eq. (3):

$$A_D = \begin{matrix} A_{11} & A_{12} & ..A_{1n} \\ A_{21} & A_{22} & ..A_{2n} \\ A_{n1} & A_{n2} & ..A_{nm} \end{matrix} \quad (3)$$

The following fitness function is used to store the fitness value of task D presented in eq. (4):

$$A_{oD} = \begin{matrix} fA_{11} & fA_{12} & ..fA_{1n} \\ fA_{21} & fA_{22} & ..fA_{2n} \\ fA_{n1} & fA_{n2} & ..fA_{nm} \end{matrix} \quad (4)$$

The following metric is used to restore the VM fitness values presented in eq. (5):

$$A_{VM} = \begin{matrix} AL_{11} & AL_{12} & ..AL_{1n} \\ AL_{21} & AL_{22} & ..AL_{2n} \\ AL_{n1} & AL_{n2} & ..AL_{nm} \end{matrix} \quad (5)$$

The following fitness function is used to store the fitness value of VM’s and presented in eq. (6):

$$A_{oVM} = \begin{matrix} fA_{11} & fA_{12} & ..fA_{1n} \\ fA_{21} & fA_{22} & ..fA_{2n} \\ fA_{n1} & fA_{n2} & ..fA_{nm} \end{matrix} \quad (6)$$

The roulette wheel is used to select a best VM from its population based on *multi objective fitness* calculation (CPU, RAM, Bandwidth, and Resource Utilization). This highly fit VM to catch the task D with high probability.

The pseudocode of above method is as follows:

Start
Task initialization
$D \leftarrow D_1, D_2, D_3 \ldots\ldots\ldots D_n$
Compute execution time, and cost of each task D and sort in ascending order
Set priority based on execution time
Resource Management
Apply Antlion optimizer for best resource to be allocated
If(VM_{cpu}>$Threshold_{cpu}$ && VM_{ram}>$Threshold_{ram}$ && VM_{bw}>$Threshold_{bw}$)
VM_{best} ==0
Separate from resource pool
Else
Select VM as best resource
Apply round robin scheduling for VM allocation
End

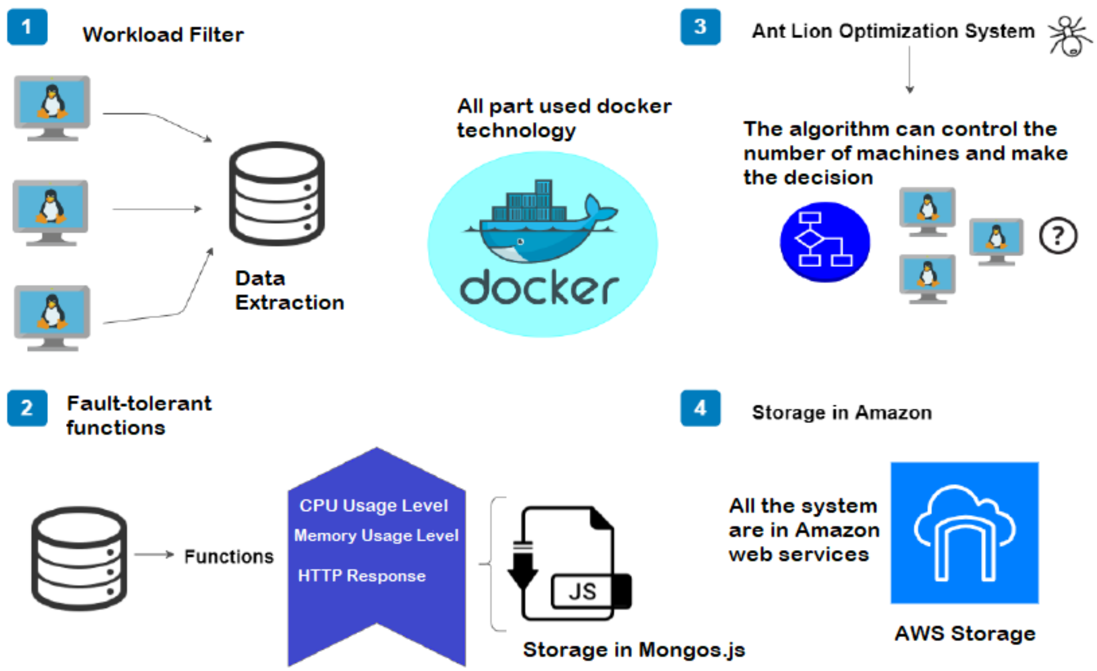

Figure 1. Architecture of Proposes ALO based solution

4. Architecture

The architecture of proposed system shown in Figure 1, is implemented and tested in AWS EC2 (Amazon Web Service Elastic Cloud Computing) dynamic environment. The submitted workloads are extracted and filtered through, then functions (CPU, Memory, and Bandwidth) applied to the EC2 instances which is now optimized through ALO and provide the optimal EC2 instance for scheduling the workloads to optimal EC2 instances. The novel approach is the consideration of Bandwidth function with CPU and Memory function to get the optimal EC2 instances.

5. Results and Analysis

The proposed system is simulated and tested in cloudsim and AWS EC2 environments with 20 VM's and 50-1500 task (population). The comparative analysis is done with SpeQuloS (Delamare et al., 2012) and Oces (Xu et al., 2009). The observations of the test results are as discussed below:

5.1 Execution Time Analysis

The execution time for proposed system is recorded on the basis of successfully scheduling of VM to workloads submitted by cloud user, in this, the systems tested with eight sample data. First the systems generate synthesis test data which is set of 50 to 1500 set of 30 task D. It is defined as the product of total execution time to schedule of tasks to optimal resources (VM). The proposed system scheduled the 50 to 1500 task D successfully and the average time is observed as 960 milliseconds. Whereas the other existing works also be tested in same environment t with same number of tasks and the average time for scheduling of given task is observed as 1061 and 1140 milliseconds, which shows that the proposed system take 9% less time for scheduling with best VM's. The comparative analysis is presented in Figure 2. The overall analysis of proposed work

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 758-768, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.060

based on execution time is evidence of better performance as compared with SpeQulos and Oces. However, the claim may differ the platform other than Asw Ec2 and Cloudsim toolkit.

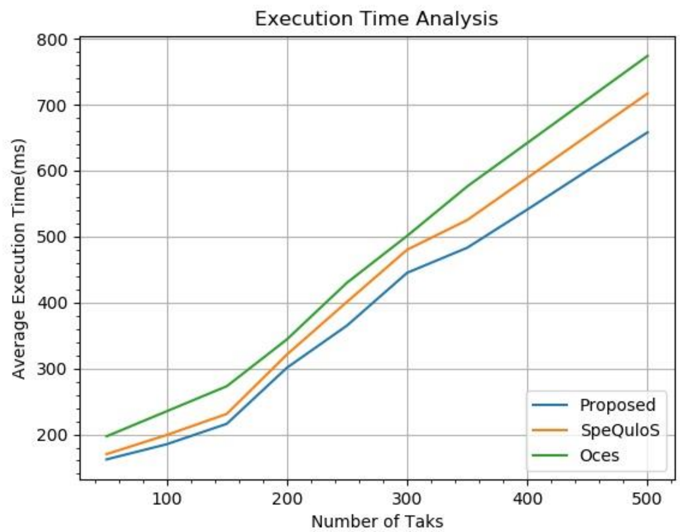

Figure 2. Execution Time Analysis of proposed work (ms)

5.2 Cost Analysis

The cost for proposed system is recorded on the basis of successfully scheduling of VM to workloads submitted by cloud user. First the systems generate synthesis test data which is set of 50 to 1500 set of 30 task D. It is defined as the product of total execution time to schedule of tasks to optimal resources (VM). The proposed system scheduled the 50 to 1500 task D successfully and the average cost is observed as $ 2.8 K. Whereas the other existing works also be tested in same environment t with same number of tasks and the average cost for scheduling of given task is observed as $3.1 K and $ 3.4 K, which shows that the proposed system take 5% less cost for scheduling with best VM's. The comparative analysis is presented in Figure 3. The overall analysis of proposed work based on cost is evidence of better performance as compared with SpeQulos and Oces. However, the claim may differ the platform other than Asw Ec2 and Cloudsim toolkit.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 758-768, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.060

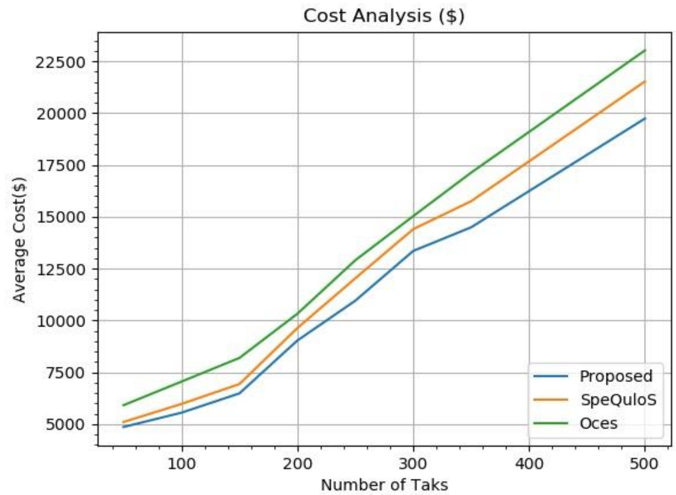

Figure 3. Cost Analysis of proposed work (ms)

5.3 SLA Violation Rate Analysis

The proposed system scheduled the 50 to 1500 task D successfully and the average SLA violation rate is observed as 0.42 milliseconds. Whereas the other existing works also be tested in same environment with same number of tasks and the average SLA violation rate for scheduling of given task is observed as 0.82 and 0.85 milliseconds, which shows that the proposed system take 48% less for scheduling with best VM's. The comparative analysis is presented in Figure 4. The overall analysis of proposed work based on SLA is evidence of better performance as compared with SpeQulos and Oces. However, the claim may differ the platform other than Asw Ec2 and Cloudsim toolkit.

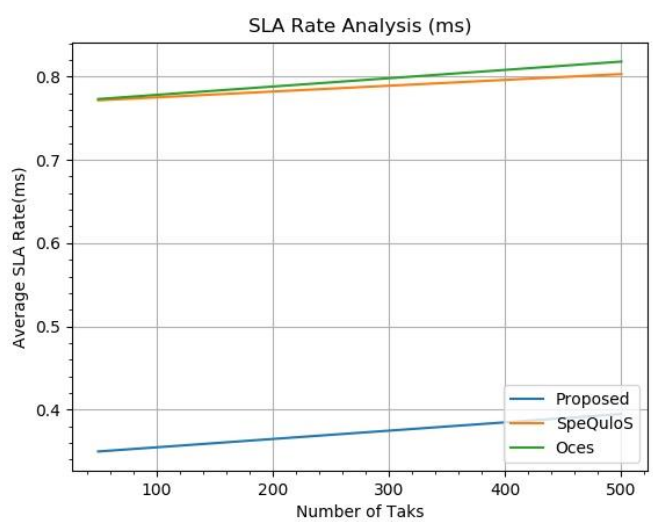

Figure 4. SLA Violation rate Analysis of proposed work (ms)

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 758-768, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.060

5.4 Observations

Real-time cloud platform generally schedules the task based on their arrival time, moreover in this research, the fundamental approach is overruled and optimization-based policy has been introduced to select better and available resource to execute the desired task. The proposed system is tested in simulation as well as real-time environment and the data obtained after execution of proposed framework is presented in Table 2.

Table 2. SLA, Cost and Execution time analysis of proposed system

Workloads	Execution Time Analysis (ms)			Cost Analysis ($)			SLA Rate Analysis (ms)		
	Proposed	SpeQulos	Oces	Proposed	SpeQulos	Oces	Proposed	SpeQulos	Oces
50	162	170	197	4860	5100	5910	0.3497	0.7715	0.773
100	185	199	235	5550	5970	7050	0.3547	0.7750	0.778
150	216	231	273	6480	6930	8190	0.3597	0.7785	0.783
200	301	321	344	9030	9630	10320	0.3647	0.7820	0.788
250	365	401	430	10950	12030	12900	0.3697	0.7855	0.793
300	445	480	501	13350	14400	15030	0.3747	0.7890	0.798
350	483	525	576	14490	15748	17127	0.3797	0.7925	0.803
400	541	589	642	16238	17673	19092	0.3847	0.7960	0.808
450	600	653	708	17987	19598	21057	0.3897	0.7995	0.813
500	658	717	774	19736	21524	23023	0.3947	0.8030	0.818
550	716	782	840	21484	23449	24988	0.3997	0.8065	0.823
600	774	846	906	23233	25374	26953	0.4047	0.8100	0.828
650	833	910	973	24981	27299	28918	0.4097	0.8135	0.833
700	891	974	1039	26730	29224	30883	0.4147	0.8170	0.838
750	949	1038	1105	28478	31149	32848	0.4197	0.8205	0.843
800	1008	1102	1171	30227	33074	34813	0.4247	0.8240	0.848
850	1066	1167	1237	31975	35000	36779	0.4297	0.8275	0.853
900	1124	1231	1303	33724	36925	38744	0.4347	0.8310	0.858
950	1182	1295	1370	35472	38850	40709	0.4397	0.8345	0.863
1000	1241	1359	1436	37221	40775	42674	0.4447	0.8380	0.868
1050	1299	1423	1502	38970	42700	44639	0.4497	0.8415	0.873
1100	1357	1487	1568	40718	44625	46604	0.4547	0.8450	0.878
1150	1416	1552	1634	42467	46550	48569	0.4597	0.8485	0.883
1200	1474	1616	1700	44215	48476	50535	0.4647	0.8520	0.888
1250	1532	1680	1767	45964	50401	52500	0.4697	0.8555	0.893
1300	1590	1744	1833	47712	52326	54465	0.4747	0.8590	0.898
1350	1649	1808	1899	49461	54251	56430	0.4797	0.8625	0.903
1400	1707	1872	1965	51209	56176	58395	0.4847	0.8660	0.908
1450	1765	1937	2031	52958	58101	60360	0.4897	0.8695	0.913
1500	1824	2001	2098	54707	60027	62326	0.4947	0.8730	0.918

6. Conclusion

Resource management makes the cloud cost effective, where optimizing the resource is minimized the cost and time to be taken for scheduling and task to the workload. This research is focused on optimizing time, cost, and SLA rate so that the efficiency of the system increases. With this objective, the system is proposed, which is simulated and implemented in *cloudsim* as well as tested in *Amazon web services AWS*. The proposed work is managed the scheduling activity with low cost and time. The experimental results show that the proposed model is more efficient as compared with the existing systems. The less SLA rate is evidence of the proposed work that is more trustable on account of cloud end user. The analysis proofs that proposed system is more efficient based on cost, time and SLA rate.

Conflict of Interest

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

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors sincerely appreciate the editor and reviewers for their time and valuable comments.

References

Abrishami, S., & Naghibzadeh, M. (2012). Deadline-constrained workflow scheduling in software as a service cloud. *Scientia Iranica*, *19*(3), 680-689.

Ai, L., Tang, M., & Fidge, C. (2011) Resource allocation and scheduling of multiple composite web services in cloud computing using cooperative coevolution genetic algorithm. In: Lu BL., Zhang L., Kwok J. (eds) *Neural Information Processing*. ICONIP 2011. Lecture Notes in Computer Science, vol. 7063. Springer, Berlin, Heidelberg.

Chen, X., Wang, H., Ma, Y., Zheng, X., & Guo, L. (2020). Self-adaptive resource allocation for cloud-based software services based on iterative QoS prediction model. *Future Generation Computer Systems*, *105*, 287-296.

Delamare, S., Fedak, G., Kondo, D., & Lodygensky, O. (2012, June). SpeQuloS: a QoS service for BoT applications using best effort distributed computing infrastructures. In Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing (pp. 173-186). https://doi.org/10.1145/2287076.2287106.

Dewangan, B.K., Agarwal, A., & Pasricha, A. (2016, October). Credential and security issues of cloud service models. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* (pp. 888-892). IEEE. Dehradun, India.

Dewangan, B.K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2018, December). Autonomic cloud resource management. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (Pp. 138-143). IEEE. Solan, Himachal Pradesh, India

Dewangan, B.K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Self-characteristics-based Energy-Efficient Resource Scheduling for Cloud. *Procedia Computer Science*, *152*, 204-211.

Kertesz, A., Kecskemeti, G., & Brandic, I. (2011, February). Autonomic SLA-aware service virtualization for distributed systems. In *2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing* (Pp. 503-510). IEEE. Ayia Napa, Cyprus.

Levitin, G., Xing, L., & Xiang, Y. (2020). Optimization of time constrained N-version programming service components with competing task execution and version corruption processes. *Reliability Engineering & System Safety*, *193*, 106666.

Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, *83*, 80-98.

Moschakis, I.A., & Karatza, H.D. (2011, June). Performance and cost evaluation of gang scheduling in a cloud computing system with job migrations and starvation handling. In *2011 IEEE Symposium on Computers and Communications (ISCC)* (Pp. 418-423). IEEE. Kerkyra, Greece.

Oprescu, A.M., & Kielmann, T. (2010, November). Bag-of-tasks scheduling under budget constraints. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science* (Pp. 351-359). IEEE. Indianapolis, IN, USA.

Reig, G., Alonso, J., & Guitart, J. (2010, July). Prediction of job resource requirements for deadline schedulers to manage high-level SLAs on the cloud. In *2010 Ninth IEEE International Symposium on Network Computing and Applications* (Pp. 162-167). IEEE. Cambridge, MA, USA.

Su, S., Li, J., Huang, Q., Huang, X., Shuang, K., & Wang, J. (2013). Cost-efficient task scheduling for executing large programs in the cloud. *Parallel Computing*, *39*(4-5), 177-188.

Sutar, S.G., Mali, P.J., & More, A.Y. (2020). Resource utilization enhancement through live virtual machine migration in cloud using ant colony optimization algorithm. *International Journal of Speech Technology*, *23*(1), 79-85.

Van Den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2010, July). Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In *2010 IEEE 3rd International Conference on Cloud Computing* (Pp. 228-235). IEEE. Miami, FL, USA.

Van Den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2013). Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Generation Computer Systems*, *29*(4), 973-985.

Xu, M., Cui, L., Wang, H., & Bi, Y. (2009, August). A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing. In *2009 IEEE International Symposium on Parallel and Distributed Processing with Applications* (pp. 629-634). IEEE. Chengdu, China.