International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 11 (2019) pp. 170-177

Received: 27 July, 2018; Accepted: 19 July, 2019; Published: 23 July, 2019

Design of Self-Management Aware Autonomic Resource Scheduling Scheme in Cloud

Bhupesh Kumar Dewangan[1], Amit Agarwal[2], Venkatadri M.[3], and Ashutosh Pasricha[4]

[1] Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies, Energy Acres, Dehradun, Uttarakhand-248007, India
bhupesh.dewangan@gmail.com

[2] Department of Virtualization, School of Computer Science, University of Petroleum and Energy Studies, Energy Acres, Dehradun, Uttarakhand-248007, India
aagarwal@ddn.upes.ac.in

[3] Department of Computer Science, Amity University Madhya Pradesh, Gwalior-474005, India
venkatadri.mr@gmail.com

[4] Schlumberger Asia Services Ltd., Gurgaon, Haryana -110038, India
apasricha@slb.com

Abstract*: Cloud computing is an emerging technology through which resources can be shared over the internet with different users either free or on a rent basis. Resource scheduling in cloud computing is a challenging area for researchers as is maximum utilization can opt through efficient resource scheduling algorithm. In this environment, incalculability, and expansion of resources meeting the problem of allocation. Other than this, virtual machine provisioning, packaging, and availability guarantee decrease the performance. To avoid this, an efficient resource management technique is required, which will manage the resources effectively, guarantee the availability, and execute the workloads automatically. In this paper, we present an autonomic resource management technique named as SMART (Self-management aware autonomic resource management technique for cloud). SMART offers self-optimization for maximum resource utilization, self-healing to avoid failures on execution, self-protection to stop multiple requests from the user, and self-configuration to make sure the availability of resources. We have implemented and evaluated the performance of SMART in cloudsim 3.0 and the experimental results of proposed work perform better in terms of SAL violation rate, cost, energy consumption, resource utilization and execution time.

***Keywords*:** Cloud Computing, Self-management, Performance, Resource Utilization, Cost, Energy, SLA Violation, QoS.

I. Introduction

Cloud computing is one of the best foundations of web computing. At present cloud computing has been using throughout the various nation wherein resources are rendered on a utility premise simply like power [1]. Cloud frameworks are versatile by definition, although that it might be made out of expensive arrangements of segments and thus complex programming structures to be deal with manually in a proficient way [2]. Due to the huge demand for cloud services, the service policy should be flexible so that adjustment in resource provisioning in light of the reliable changes in the customer's workload and deals. In addition it is observed that, to fulfill the demand of the cloud user's a large number of datacenters and servers are installed, which results in more power consumption and energy losses [3]. To overcome these issues, the resources on the cloud should be properly managed so that performance can be optimized and it can be resolved through resource management algorithms. Resource management is one of the leading methods through the efficiency of services can be enhanced. Many resource algorithms are utilizing to optimize the different objective functions as cost-based, profit-based, auction-based, energy-based, bio-inspired, fault-tolerant based and many more. Some of these are non-autonomic and need to be self-managed so that it can be managed without human interventions, which minimizes the time and cost of the system. To adapt self-management characteristics, an autonomic cloud resource management is required.

In continues, Autonomic resource management [4] has capacity to enhance usage of resources and client fulfillment in autonomic frameworks, which are self-optimize, self-configuration, self-protection, and self-healing. Self-healing is a capacity of an astute framework to distinguish, examine and recoup from disturbing shortcomings consequently. Self-configuring is an ability of a perceptive framework to make the adjustments in nature. Self-Optimizing [5] is the ability to proficiently boost asset distribution and usage for fulfilling prerequisites of various clients. Given these qualities, programming frameworks can be computerizing to take remedial activities if undertakings are repressing by blame or disappointment is identifies. The remedial activities can incorporate changes to setups, recuperating a procedure or application that has fizzled, ensure a part over-burden that would cause a bottleneck in the work process, and streamline framework execution. Framework expected to act normally recovering can retouch themselves at runtime in response to changing natural or operational conditions. Our objective is to design a self-management aware autonomic resource scheduling scheme in a cloud to optimize cost, time, and energy and Service Level Agreement (SLA) violation rate.

In continuation of this article, section I contains the introduction of the article area and subarea. Moreover, in section II, the background of the area and state of art literature is presenting and comparing the existing works with different performance metrics with findings, the proposed method to enhance the current framework is presenting in section III. Where in section IV, the simulation results and its analysis is presenting, and in section V, we have concluded our work.

II. Related Work

Many researchers have designed and developed different autonomic resource management approaches as follows:

Singh et al. [6] had proposed SLA-careful autonomic resource organization technique called STAR, which essentially focused on lowering SLA infringement rate for the gainful development of cloud associations where Ziad et al. [7] had proposed ACCRS, a structure for distributed computing frameworks to propel the framework's usage level, lessen the cost, control utilization, and satisfy SLAs. The ACCRS structure utilizes Autonomic Computing fundamental parts, which incorporates state observing, arranging, basic leadership, blame predication, discovery, and main driver investigation for recuperation activities to enhance framework's unwavering quality, availability, and utilization level by scaling resources in light of changes in the cloud system state. Author Kuan et al. [8] had proposed a sensual approach for the automated organization of the aggregate SLA lifecycle, including exchange and provisioning, yet focus on seeing as the driver of contemporary flexibility requirements. The approach segregates the agree to develop requests of entertainers using strong parent– tyke associations with achieve adjustment to inner disappointment stresses into various self-overseeing layers that can be continuously united into a natural, parallelized, effective and capable organization structure and Juntao et al. [9] proposed a novel dynamic undertaking booking calculation in view of enhanced hereditary calculation. In view of the hereditary calculation, the proposed calculation considers the dynamic attributes of the distributed computing condition. The CloudSim reproduction stage is select for recreation; exploratory outcomes demonstrate that the proposed calculation can viably enhance the throughput of distributed computing frameworks, and can essentially decrease the execution time.

Sotiriadis et al. [10] presented a Cloud VM booking calculation that considers effectively running VM asset use after some time by examining past VM usage levels keeping in mind the end goal to plan VMs by enhancing execution. They watch that Cloud administration forms, as VM situation, influence as of now sent frameworks (for instance, this could include throughput drop in a database group), so they expect to limit such execution corruption. In addition, over-burden VMs tend to take assets (e.g. CPU) from neighboring VMs, so their work amplifies VMs genuine CPU usage where Paya et al. [11] presented a vitality mindful operation show utilized for stack adjusting and application scaling on the cloud. The fundamental approach of this work is to characterize the ideal vitality administration and increment the measure of servers working in this administration. Sit out of gear and daintily stacked servers are changed to one of the rest states to spare vitality. Author Mehrotra et al. [12] Proposed a control theoretic, show based, and control mindful approach for parallel circle execution is available, where framework execution and power prerequisites can be, modify progressively, while keeping up the predefined nature of administration (QoS) objectives within the sight of bothers, for example, stack irregularity because of varieties in the accessibility of computational assets. This approach is autonomic, execution coordinated, progressively controlled, and free of the execution of the application and Mashayekhy et al. [13] proposed a technique confirmation polynomial time estimate plot (PTAS) instrument that tackles the VM example provisioning and portion issue. The objective is to discover an assignment of assets to the clients expanding the social welfare, where the social welfare is the aggregate of clients' valuations. An exponential time technique evidence ideal instrument additionally composed that will fill in as a benchmark for the execution of the proposed PTAS system. The PTAS instrument is procedure proof that is, the clients do not have motivators to lie about their asked for packs of VM cases and their valuations. The instrument is configure to adjust to changing conditions and to lead the framework into a harmony in which clients are not motivated to control the framework by untruthfully detailing their asset solicitations and valuations.

Maria Salama et al. [14] planned to give quality-driven outline cases to build up a class of designing engaged by the norms of care. They expounded on the common sense of relating QoS procedures with careful abilities to better respond to QoS run-time requirements and trade-offs. They depicted novel increases, which make the association between QoS methodologies and care express. They measure the feasibility, comprehensive explanation and health of the proposed approach and likewise its potential pertinence to careful outlines where Singh et al. [15] had proposed a QoS metric based resource provisioning system. This methodology considers provisioned resource assignment and arranging of benefits. The essential purpose of this examination work is to separate the workloads, arrange them inlight of fundamental cases and after that course of action the cloud workloads before veritable arranging. The preliminary comes to fruition showed that QoS metric based resource provisioning strategy is capable in diminishing execution time and execution cost of cloud workloads close by various QoS parameters. Authors Viswanathan H. et al. [16] proposed a novel resource provisioning system for sorting out the heterogeneous detecting, figuring, and correspondence capacities of static and cell phones so as to a shape a portable processing lattice. This nearby registering network can be an adventure to empower the novel portable applications that require constant in-the-field information gathering and preparing. They conferred the asset provisioning structure with autonomic abilities, to be specific, self-association, self-streamlining, and self-recuperating, to be vitality and vulnerability mindful in the dynamic portable condition and Singh et al. [17] huge measure of vitality utilization prompts high operational expenses, decreases rate of profitability, and contributes towards carbon impressions to the earth. Consequently, there is need for vitality mindful cloud-based framework that plans processing assets naturally by considering vitality utilization as a vital parameter. In this paper, vitality proficient autonomic cloud framework SOCCER proposed for effective planning of cloud resources in server farms. The proposed work thinks about vitality as a Quality of Service (QoS) parameter and consequently upgrades the effectiveness of cloud resources by lessening vitality utilization. The execution of the proposed framework has been assessing in genuine cloud condition and the test results demonstrate that the proposed framework performs better as far as vitality utilization of cloud assets and uses these assets ideally where Gill et al. [18] to give a productive execution and to execute outstanding tasks, there is a need of nature of administration (QoS) based autonomic asset the executive's approach that oversees assets consequently and gives solid, secure and cost proficient cloud administrations. In this paper, authors present a keen QoS-mindful autonomic resource the executive's approach named as CHOPPER.

The current system does not give a guarantee of resource availability. The fitness of resources by current techniques are calculating based on processing and memory utilization only. In SMART, we guarantee the resource availability by obtaining fitness of resources based on processing, memory, and bandwidth utilization. To add, following multi-objective parameters can be considered together/ or can consider the combination of some of them like Execution time, Resource utilization, and Energy Efficiency. The comparative study of the above autonomic resource management frameworks with the proposed framework is mentioned in *Table 1*, which shows that most of the frameworks are good in terms of few parameters.

Method	ET	C	SV	RU	EC	R	FD	Aut
[6]	Yes	Yes						
[7]	Yes	Yes	Yes					
[8]	Yes	Yes						
[9]	Yes	Yes	Yes			Yes		
[10]		Yes			Yes		Yes	Yes
[11]	Yes	Yes					Yes	Yes
[12]	Yes	Yes						Yes
[13]	Yes				Yes			
[14]	Yes	Yes						
[15]	Yes	Yes		Yes				
[16]	Yes			Yes			Yes	
[17]	Yes	Yes			Yes			Yes
[18]			Yes				Yes	Yes
SMART	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1. Comparative study of existing resource management frameworks.

Where ET= Execution time, C= Cost, SV= SLA violation rate, FD= Fault detection RU= Resource utilization, EC= Energy consumption, R=Reliability, Aut=Autonomic.

The hybrid system is required which should be better in terms of all parameter which is compared on Table 1.

III. SMART- Proposed Method

In the above Table 1, it analyses that the combination of most self-attributes is not considered together from the previously mentioned literature works. The proposed system is a plan to offer self-administration to customers' needs to upgrade their resource management system RMS with autonomous features. Autonomic computing and self-resource management is an open problem in cloud computing has to be addressed in this research work.

This paper proposes a hybrid resource provisioning approach for cloud services, which is based on a combination self-characteristics which includes self-healing, self-optimization, and self-configuration by considering some standards parameter to enhance the performance, which is including execution time, cost, SLA violation rate, energy consumption rate, and QoS. The architecture of the SMART is presenting in the next section. The workflow of SMART is presenting in Figure 1, according to this, SMART has five module namely Workload Priority, Self-protection, Self-optimization, Self-healing, and Self-configuration.

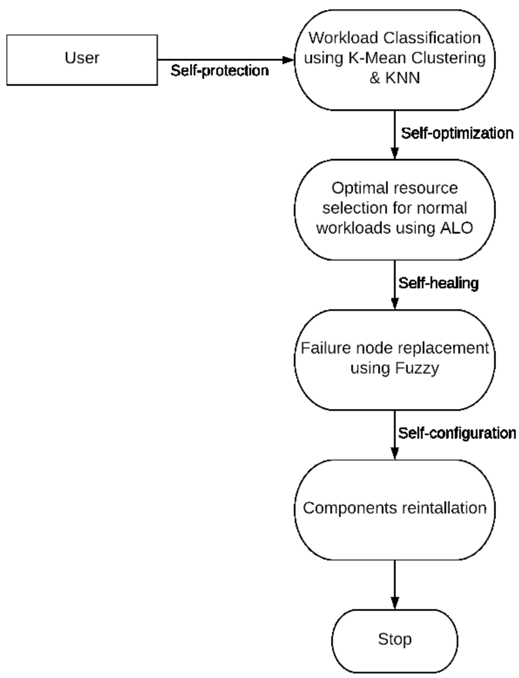

Figure 1. Workflow of proposed method SMART

A. Workload priority

The workload SLA violation rate, execution time and cost is calculated by equation (2), (3) and (4). Each workload is processed by modified K-mean cluster algorithm to classify and cluster according to its type. In this, four centroid values are determined as follows: centroid-01, centroid-02, centroid-03, and centroid-04. Each centroid value is calculated by the weight value of each workload. For centroid-01, the weight value is divided by four, for centroid-02, the weight value is divided by 2, for centroid-03, the weight value is divide by third half, and for centroid-04, the weight value is taken as it is. On the basis of centroid value, all workloads are categorized into four clusters, namely cluster c_1, c_2, c_3, and cluster c_4. Each cluster is having own property as storage, web service, performance, graphics oriented, endeavor software, ecommerce, productivity etc. The classification obtained through algorithm-01.

Algorithm-01: Algorithm for cluster (Modified K-mean)

```
Start
Initialize Workloads;
   W←W1, W2, W3, ………..Wn;
while i < n do
   sum   sum + weight(W);
   compute centroid vaule;
   c1 ← weight()/4;
   c2 ← weight()/2;
   c3 ← weight()*3/4;
   c4 ← weight()-1;
end
Clustering;
   x ← sum of weights;
while i <= n do
   if(x <= c1) then
         assign workload to cluster1;
      else if(x > c1 && x <= c2) then
         assign workload to cluster2;
      else if(x > c2 && x <= c3) then
         assign workload to cluster3;
      else
      assign workload to cluster4
   endif
End
```

B. Self-protection

User submits the workloads to the framework for VM's allocation. The workloads are processed through the self-protection module and classify through algorithm-01. The redundant workloads are identified and trace its signature and separated from the workload pool through algorithm-02.

Algorithm-02: Algorithm for Self-protection

```
Start
Initialize Workloads;
   W←W1, W2, W3, ………..Wn;
while i < n do
   if(Wfrequancy = 1) then
      assign to workload list;
   else
      identify as malicious workload;
      seperate from workload list;
   end
End
```

C. Self-optimization

In this, the VM's cost, energy, execution time, and SLA violation rate is optimized through the modified ant lion algorithm. This algorithm was applied in network optimization, and performed well. The workloads are sorted according to its priority and VM's are sorted according to its optimal value as discussed in algorithm-03. The fitness of each VM is obtained through Central Processing Unit CPU, Memory, and Bandwidth utilization by following equations:

CPU Utilization

$$U_{VMi}^{cpu} = \sum_{i=1}^{m} \frac{VM_i^{cpu}}{VM_{max}^{cpu}} \quad \text{……….} (1)$$

Where U_{VMi}^{cpu} is CPU utilization by i[th] VM, VM_i^{cpu} is initial CPU value of i[th] VM, VM_{max}^{cpu}= 50 mips (in this research), is maximum CPU that any VM can utilize. The range of CPU utilization will be 0< U_{VMi}^{cpu}<1.

RAM Utilization

$$U_{VMi}^{RAM} = \sum_{i=1}^{m} \frac{VM_i^{RAM}}{VM_{max}^{RAM}} \quad \text{……….} (2)$$

Where U_{VMi}^{RAM} is RAM utilization by i[th] VM, VM_i^{RAM} is initial RAM value of i[th] VM, VM_{max}^{RAM}= 512 mbps (in this research), is maximum RAM that any VM can utilize. The range of RAM utilization will be 0< U_{VMi}^{RAM}<1.

Bandwidth Utilization

$$U_{VMi}^{BW} = \sum_{i=1}^{m} \frac{VM_i^{BW}}{VM_{max}^{BW}} \quad \text{……….} (3)$$

Where U_{VMi}^{BW} is Bandwidth utilization by i[th] VM, VM_i^{BW} is initial Bandwidth value of i[th] VM, VM_{max}^{BW}= 1000 kbps (in this research), is maximum Bandwidth that any VM can utilize. The range of Bandwidth utilization will be 0<U_{VMi}^{BW}<1.

Algorithm-03: Algorithm for resource optimization

```
Start
   Initialize the first VM and Best VM randomly
   Calculate the fitness of VMs
   Find the best VMs and assume it as optimum VM
   while VM¿Best VM do
      for(i=0 to n)
         Calculate the fitness of all VM through Eqs. (1), (2),
         and (3)
         Select a VM using Roulette wheel
         Update the position of VM and Best VM
      Endfor
   End
End
```

D. Self-healing

The best VM's values are identified by its fitness like CPU, memory, and Bandwidth of each VM's and sorted it in order. The VM's which have less than threshold fitness value are identified as fault VM. The fault VM's are separate and replaced by best VM's using fuzzy and sorted in order to allocate to workloads according to the user priority.

E. Self-configuration

Self-configuring in cloud-based autonomic frameworks is the establishment of missed or obsolete segments in light of the alarm created by the framework without human interaction. A few parts might be reinstalled in evolving conditions.

Scheduling of resources done through this.

F. SMART- Architecture

Cloud user submit the workloads to the cloud, which determine the quality of service requirements and processed by the self-protecting module to check the malicious requests (if any) and separates from the workload list. Further, the workloads are processed through modified k-mean algorithm-01 and categorized by its type into four different clusters like administration, communication, computing, and storage. Self-optimization applied to the resource side to find the optimal solution. Where self-healing finds the VM which has very low fitness value and separate from the resource pool by applying best resource utilization value methods. Furthermore, the VM's are sorted in the order of its best utilization value and send to the resource pool. The VM's are ready in the pool to assign the workloads according to its quality of service requirements and user priority. Self-configuration module re-installs the required components if any. Where, resource scheduler, assign the resources to the workloads and get back to user with the quality of service requirements and user's satisfaction.

G. Resource management through SMART

The resources are allocating through a SMART algorithm-04 using autonomic characteristic.

Algorithm-04: Autonomic resource scheduling algorithm

Start
Initialize workloads W←W_1, W_2, W_3, ………..W_n;
Apply **algorithm-01** for clustering
Call self-protection to separate malicious workloads from the submission list
Compute workload priority and arrange it in order
Call **self-optimization** to find optimal VM's
if (VM > BestVM) **then**
change the position
Sort the best VM's in order
Call **self-healing** for evaluation of faulty VM's
if (Best VM < threshold) **then**
Reject the VM and send it to failure list
Separate all rejected VM's from resource pool
Sort all the VM's in order
Call the round-robin resource scheduling algorithm to assign best VM's to priority based workloads
End

IV. Results and Discussion

The SMART framework is simulated in cloudsim toolkit with 20 virtual machines VM's. SMART schedules cloud workloads to the Best VM's and the following metrics are calculated to evaluate the efficiency of proposed SMART system. SMART compared with SOCCER[16], and CHOPPER[17] as both the existing frameworks are autonomic resource management frameworks which considered the self-characteristics techniques and evaluated by many parameters.

A. SLA violation rate analysis

When the workloads are submitted to SMART for VM management, some workloads are failed due to priority, weight etc. which is evaluated through equation (4).

$$F_{rate} = \frac{R_i}{n} \quad \text{......... (4)}$$

Where F_{rate} is Failure rate, Ri= Failure rate of workload and n= Total number of workloads.

The analysis for SLA violation rate is presented in Figure 2. In this, SVR is calculated for SMART and other two existing autonomic frameworks SOCCER and CHOPPER, it is observed that SMART's SLA violation rate is less than the other two frameworks. It is calculated through equation (5), and tested for different workloads. We compared SLA violation rate of SMART with existing autonomic frame works SOCCER and CHOPPER and the values obtained are presented in *Table 2.*

$$SLA_{rate} = F_{rate} * W_i \quad \text{......... (5)}$$

Where i belongs to 1 to n, and W= Workloads

Workload	SMART	CHOPPER	SOCCER
10	200	300	250
20	250	350	300
30	300	500	400
40	450	680	420
50	650	800	750
60	695	900	870

Table 2. SLA violation rate analysis.

As per the data received from the simulator, the SLA violation rate is increasing by increasing the number of workloads to SMART and other frameworks, and as per the Figure 2, as compare SMART has less violation rate.

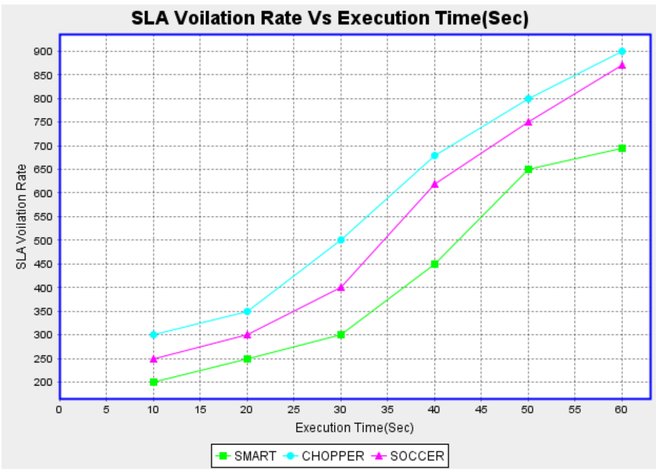

Figure 2. SLA violation rate analysis.

B. Execution time analysis

The execution time for each workload is computed by equation (6).

$$ET = \frac{WC_i - WS_i}{n} \quad \text{......... (6)}$$

Where WC_i workload completion on ith time, WS_i is workload submission on ith time and 1 < i < n.

It is the difference between resource creation and resource submission time. Data received from simulator for SMART and other two frameworks are presented in Table 3.

Workload	SMART	CHOPPER	SOCCER
200	0.70	0.90	0.85
400	0.60	0.85	0.78
600	0.40	0.80	0.60
800	0.35	0.70	0.58
1000	0.24	0.70	0.56

Table 3. Execution time analysis (sec).

The above values are plotted in Figure 3, and compared with the SLA violation rate, according to this, when the SLA violation rate is increasing, the execution time of the frameworks are decreasing as expected, but when it is compared with other autonomic frameworks than it is observed that SMART take less time to execute.

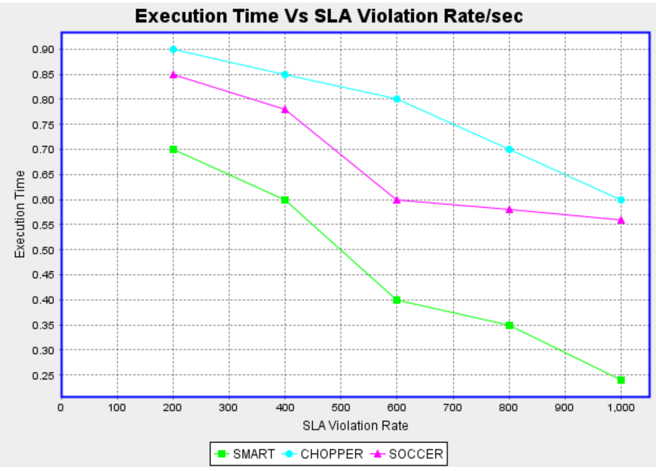

Figure 3. Execution time analysis based on SLA rate.

C. Energy analysis

The energy consumption by SMART is evaluated through equation (7).

$$EC = k * max + (1 - k) * max * U_{vm} \quad \text{........(7)}$$

Where k is 0.50 and U_{vm} is resource utilization obtained by equation (8).

$$U_{vm} = U_{vm}^{cpu} + U_{vm}^{RAM} + U_{vm}^{BW} \quad \text{........(8)}$$

Where central processing unit CPU utilization, bandwidth BW utilization and memory RAM utilization computing from the simulator. The energy consumption rates are recorded for different workloads and presented in Table 4.

Workload	SMART	CHOPPER	SOCCER
200	9.7	12	10.5
400	13.5	15	14.7
600	20	26.5	23
800	27	32	29.4
1000	37	41.3	38.5

Table 4. Energy consumption rate analysis (kw/h)

The above data are plotted in the graph and presented in Figure 4.

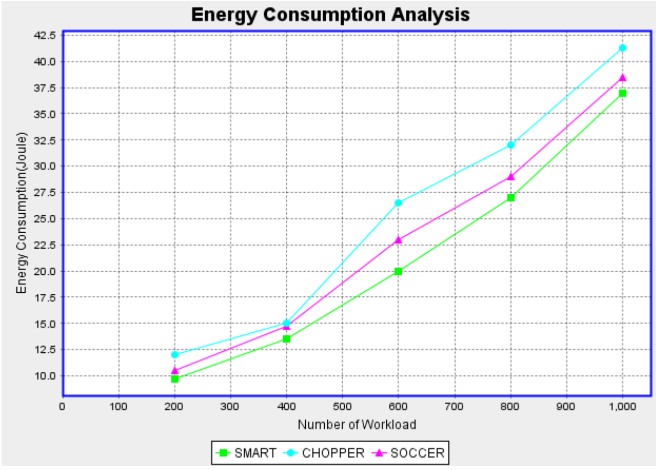

Figure 4. Energy consumption analysis.

When the number of workloads is increasing, the consumption of energy is also increasing, but when it is compared to another autonomic framework than it is found that SMART consume less energy. Where the energy efficiency of SMART and other two frameworks is presented in Figure 5.

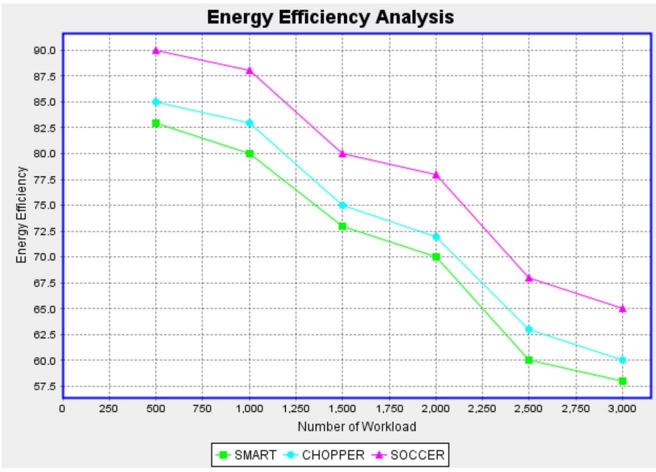

Figure 5. Energy efficiency analysis.

The energy efficiency of all the above frameworks is decreasing when workloads are increasing as expected. In Figure 5, it observes that the energy efficiency of SMART is more than existing autonomic frameworks.

D. Resource cost analysis

SMART framework consumes less resource cost as it takes minimum execution time, less energy consumption rate, greater energy efficiency and produce less SLA violation rate. The resource cost is evaluating through equation (9).

$$CostC = c * \text{EC} \quad \text{........(9)}$$

The resource cost of SMART and other two frameworks computed through equation (9) are presented in Table 5.

Workload	SMART	CHOPPER	SOCCER
10	18	30	25
20	20	35	30
30	25	48	40
40	38	58	49
50	40	63	61
60	55	75	69

Table 5. Resource cost analysis ($).

The graphical presentation of Table 5 is plotted in Figure 6.

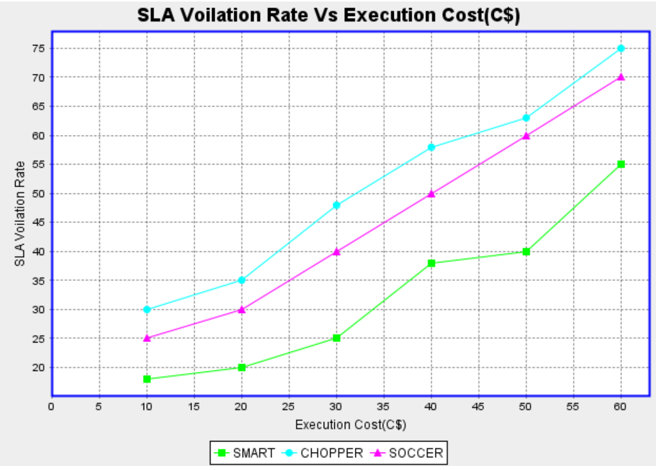

Figure 6. Resource cost analysis

The resource execution cost is increasing with increase in SLA violation rate. The proposed framework SMART takes less cost to execute the resource allocation to priority based workloads, and observe that it performs utmost.

The overall performance evaluation based on the SLA violation rate, execution time, energy consumption and efficiency, and resource cost analysis carried out and it is observed that SMART gives an utmost performance.

The above-mentioned issues can be also some metaheuristics algorithms like cost can be optimized through Parallel Hurrican Optimization Algorithm PHOA [19], the performance can be optimized through krill herd artificial bee colony KHABC [20], also a hybrid metaheuristic cuckoo search and krill herd algorithm CSKH to optimize objective functions [21], where ACRM provide optimized method for scheduling [22].

V. Conclusion

Resource management in the cloud is one of the challenging area, which affects the performance of the system. The article proposed as an autonomic self-managed resource management system for cloud called SMART. In this, some self-management characteristics like self-optimization through Antlion algorithm applied, where self-healing characteristics used to identify the faulty VM, and resources are scheduled through self-configuration. In this, the VM's allocated to the workloads submitted by cloud user with consideration of best VM's value and workload priority. SMART simulates in cloudsim toolkit and tested for 20 VM's and 3000 workloads submitted by a cloud user. The experimental results compared with existing autonomic resource management frameworks, and is notified that SMART reduces the cost by 35 percent take 39 percent less time to compute to allocate the resources to workloads with reduces SLA violation rate by 25 percent and maximizes the resource utilization by 25 percent. In addition, SMART reduces energy consumption rate by 5 percent and increase energy efficiency in the same rate as well. SMART gives the better efficiency as compared with the existing autonomic frameworks. The SLA violation rate, execution time and energy consumption analyzed are less as compared with the results. It also gives the guarantee of VM availability. In continues, the threshold value in optimization algorithm is fixed, that may affect the results if best VM value is nearby the threshold value. Hence, SMART is an efficient autonomic resource management technique in cloud computing. The scope of improvements in SMART is to maximize the energy efficiency by calculation of optimal threshold value for energy consumption rate by applying machine learning and linear regression method.

References

[1]. Dewangan B.K., Agarwal, A., Venkatadri, M. & Pasricha, A. (2019). Sla-Based Autonomic Cloud Resource Management Framework by Antlion Optimization Algorithm. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(4), 119-123.

[2]. Dewangan B.K., Agarwal, A., Venkatadri, M. & Pasricha, A. (2019). A Self-Optimization Based Virtual Machine Scheduling to Workloads in Cloud Computing Environment. *International Journal of Engineeringand Advanced Technology (IJEAT), 8*(4), 91-96.

[3]. Dewangan B. K., Agarwal, A., Venkatadri, M. & Pasricha, A. Energy-Aware Autonomic Resource Scheduling Framework for Cloud, International Journal of Mathematical, Engineering, and Management Sciences, Vol 4(1), pp. 41-55, 2019.

[4]. Dewangan B. K., Agarwal, A., Venkatadri, M., & 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.

[5]. Dewangan B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2019, January). Self-characteristics based energy-efficient resource scheduling for cloud. In *2019* International Conference on Pervasive Computing Advances and Applications. Procedia Computer Science 152 (2019) 204–211.

[6]. Singh, S., Chana, I., and Buyya, R. STAR: SLA-aware autonomic management of cloud resources, IEEE Transactions on Cloud Computing, pp. 1-14, 2017.

[7]. Al-Sharif, Z. A., Jararweh, Y., Al-Dahoud, A., and Alawneh, L. M. ACCRS: autonomic based cloud computing resource scaling, Cluster Computing, Vol 4(1), Vol 3, pp. 1-10, 2017.

[8]. Lu, K., Yahyapour, R., Wieder, P., Yaqub, E., Abdullah, M., Schloer, B., and Kotsokalis, C. Fault-tolerant Service Level Agreement lifecycle management in clouds using actor system, Future Generation Computer Systems, Vol 54, pp. 247-259, 2016.

[9]. Ma, J., Li, W., Fu, T., Yan, L., and Hu, G. A Novel Dynamic Task Scheduling Algorithm Based on Improved Genetic Algorithm in Cloud Computing. In Wireless Communications, Networking and Applications, pp. 829-835, 2016.

[10]. Stelios Sotiriadis. Self-managed virtual machine scheduling in Cloud System. In Information Sciences, pp. 1-43, 2017.

[11]. Paya, A., and Marinescu, D. C. Energy-aware load balancing and application scaling for the cloud ecosystem, IEEE Transactions on Cloud Computing, Vol 5(1), pp. 15-27, 2017.

[12]. Mehrotra, R., Banicescu, I., Srivastava, S., and Abdelwahed, S. A power-aware autonomic approach for performance management of scientific applications in a data center environment. In Handbook on Data Centers, pp. 163-189, 2015.

[13]. Mashayekhy, L., Nejad, M., and Grosu, D. A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources, IEEE Transactions on Parallel and Distributed Systems, Vol 26(9), pp. 2386-2399, 2015.

[14]. Salama, M., and Bahsoon, R. Quality-driven architectural patterns for self-aware cloud-based software. In Cloud Computing (CLOUD), pp. 844-851, 2015.

[15]. Singh, S., and Chana, I. Q-aware: Quality of service based cloud resource provisioning, Computers and Electrical Engineering, Vol 47, pp. 138-160, 2015.

[16]. Viswanathan, H., Lee, E. K., Rodero, I., and Pompili, D. Q-aware: Quality of service based cloud resource provisioning, Uncertainty-aware autonomic resource provisioning for mobile cloud computing, Vol 26(8), pp. 2363-2372, 2015.

[17]. Singh, S., Chana, I., Singh, M., and Buyya, R. SOCCER: self-optimization of energy-efficient cloud resources, Cluster Computing, Vol 19(4), pp. 1787-1800, 2016.

[18]. Gill, S. S., Chana, I., Singh, M., and Buyya, R. CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing, Cluster Computing, pp. 1-39, 2017.

[19]. Rizk-Allah, R. M., El-Sehiemy, R. A., and Wang, G. G. A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution, Applied Soft Computing, Vol 63, pp. 206-222, 2018.

[20]. Wang, H., and Yi, J. H. An improved optimization method based on krill herd and artificial bee colony with information exchange, Memetic Computing, Vol 10(2), pp. 177-198, 2018.

[21]. Wang, G. G., Gandomi, A. H., Yang, X. S., and Alavi, A. H. A new hybrid method based on krill herd and cuckoo search for global optimization tasks, International Journal of Bio-Inspired Computation, Vol 8(5), pp. 286-299, 2016.

[22]. Dewangan B. K., Agarwal, A., Venkatadri, M. & Pasricha, A. (2019, January). Autonomic cloud resource management. In *2018* Fifth IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC)**.** (2018) 138–143.

Author Biographies

Bhupesh Kumar Dewangan pursued bachelor of engineering and master of engineering in computer science and engineering. He is pursuing PhD in computer science and engineering and at present working as Assistant Professor, in Department of Informatics, in University of Petroleum and Energy Studies. He has more than 20 research publications in various international journals and conferences. He has research interests in Autonomic Cloud Computing, Resource Scheduling, Software Engineering and Testing.

Amit Agarwal is currently associated with University of Petroleum and Energy Studies, Dehradun as Professor and Head of Department of Virtualization. He has completed Ph.D in Computer Science and Engineering from Indian Institute of Technology, Roorkee. He is having more than 18+ Years of academic, administrative and research experience of reputed Organizations. His area of research includes Cloud Computing, Network Security, Internet of Things and Network on Chip. He has published more than 70 research publications and technical articles in reputed journals and conferences.

Venkatadri M. presently working as Professor and Head Department of Computer Science and Engineering, in Amity University, Madhya Pradesh, India. He has more than 20 research publications in various international journals and conferences. Editor for Smart and Innovative Trends in Next Generation Computing, Vol-827 and vol. -828, Communications in Computer and Information Science Series, Springer. (Scopus Indexed). Guest Editor for "Smart and Innovative Trends for NexGen Computing and Communications", in SCIE and Scopus indexed journals. His research interests in Cloud Computing and Cyber Security, Data Analytics and Artificial Intelligence.

Ashutosh Pasricha pursued bachelor of engineering and master of engineering from REC Kurukshetra and PhD from IIT Delhi, Presently working as Account Head in Schlumberger: Oilfield Services. He has more than 20 research publications in various international journals and conferences. His research interests in Cloud Computing, Big data, cloud security and Resource Scheduling.