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

Arr And Fusion Based Virtual Machine Scheduling Techniques For Eucalyptus Cloud

Bharath Kumar Madakatte[1] , **Nagesh H R**[2*]

[1]Research Scholar, Mangalore Institute of Technology and Engineering, Moodabidri, India.

[2]Professor & HoD, ISE, A J Institute of Engineering and Technology, Mangaluru, India.

ABSTRACT

The Cloud computing is popularly utilized innovative technology, and virtualization approach is used to ensure maximum of services to customers. Many of the cloud providers make use of Virtual machines to fulfil customer demands. Successful scheduling of virtual machines is a significant task in cloud environment. In our paper, extensive review of existing works on various methods and scheduling techniques of cloud computing is done and summary is tabulated. The existing Virtual machine scheduling techniques Greedy, Round-Robin and Power Save of eucalyptus cloud are discussed and compared with OpenStack and Open Nebula cloud scheduling techniques. Proper utilization of Power is one among major challenges in cloud data centres, which can be reduced by fusing the workloads and closing down physical machines when it becomes inactive. In our work, two novel techniques for Eucalyptus virtual machine scheduling, Advanced Round-Robin (ARR) and Fusion Methodology are proposed. Both the techniques are designed, implemented, and the outcome are captured for evaluation. The results are graphically analysed and compared with the existing techniques of Eucalyptus cloud framework. The experimental outcomes assured that, proposed techniques works better compared to existing approaches and results in comparatively lesser power utilization on an average.

1. INTRODUCTION

Cloud computing (CC) is distributed high-scale computational paradigm where a group of processing resources, such as programming, networking, and storage services, is accessible to cloud clients through internet with improvement of virtualization tools, cloud suppliers empower their clients for submitting work demands with explicit resource requests and programming stacks and bundle them together into Virtual Machine (VM) [1]. By providing task requests to cloud suppliers, customers never again need to buy and keep up with refined equipment for the resource use in their high load, accordingly decreasing their complete expense of possession. CC has now turned into most emphasized ICT paradigm and is utilized by every web-based customer directly or indirect manner [2].

In the environment of cloud, clients can present their work demands whenever and anyplace. Once getting some work demand, cloud provider must produce an appropriate Physical Machine (PM) and assign expected resources to make sure Quality of Service (QoS), as indicated by client's work requests. Since CC is a market-situated utility, ideal scheduling of VM for CC must permit cloud providers and cloud clients to concentrate on own organizations to improve incentives, accordingly [3].

VM scheduling for CC is a complicated issue, wherein various worries should be considered and appropriately addressed. Fundamentally, those worries could be ordered into two classes, i.e., worries of cloud providers and clients. According to point of view of clients, there exists two central issues in scheduling of VM, i.e., effective rate of executing of VM requests and the consolidated expenses caused. These two worries are significant ones since cloud clients for the most part desire to effectively complete their submitted task demands and simultaneously, at the least conceivable cost.

Then again, according to the viewpoint of the cloud provider, much interesting thing is to utilize the given computational resources for gaining benefits. Suppose workload on a processing resource is much higher, which forms it likely that VMs existing on particular computing resource can't get the necessary resources, which might prompt SLA violation and degrade level of client fulfilment.

The CC framework maps countless resources to VMs utilizing virtualization tools. Scheduling procedure of VM would allocate jobs to VMs, afterward convey them to various PMs to accomplish the sharing of resource, to guarantee QoS and performance of frameworks. Thusly, it’s a vital tool to successfully schedule and send VMs as indicated by client's prerequisite to work on the usage of resources and diminish the expense of energy utilization. The Figure 1 gives general operation of VM scheduling in cloud data centres. The scheduling could be comprehensively categorized into client demands, management of resources and scheduling.

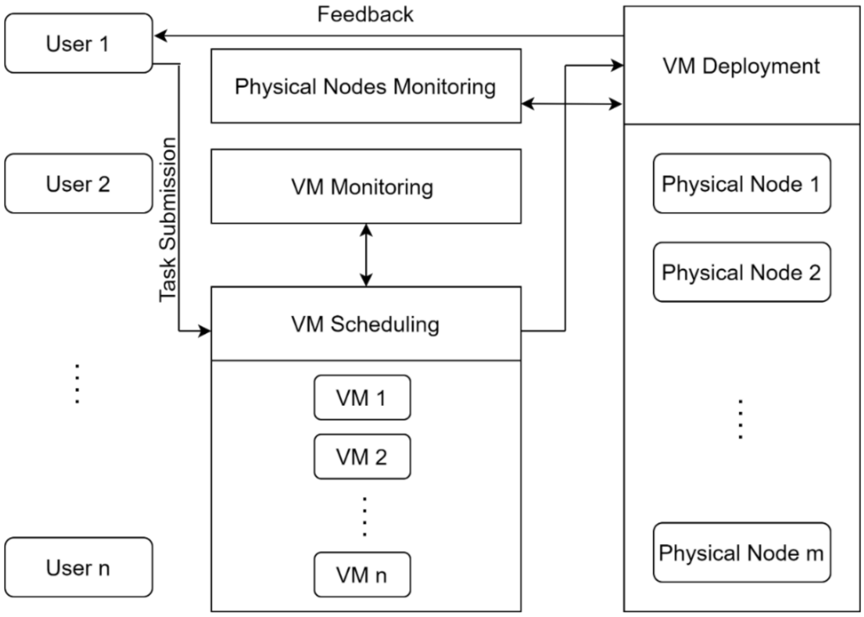

Figure 1 VM scheduling operations

Client requests: clients place requests through web to cloud provider, and create workloads which are handled utilizing resource of the cloud [4].

Management of Resources: scheduling centre screens physical and virtual nodes progressively. At the point when a VM comes up short, jobs in this VM should be moved to other. Also, in the event that a PM fail, VMs deployed on such PM should be made to migrate to others to guarantee legitimate execution of jobs.

Scheduling phase: Scheduling of VM could be isolated into two layers [5]. Initial layer is matching system among jobs and VMs. Sensible scheduling technique is to observe a planning among jobs and VMs that meet specific improvement objectives. Subsequent level is deployment of VM, that is matching operation among VMs and PMs. Deploying VMs to PMs is to guarantee execution of jobs. This can influence nature of services and performance of system.

Agreeing the possibility of MapReduce, a disseminated information handling system, the work will be separated into various jobs after client submits it. Main VM scheduling level is answerable for scheduling these jobs to VMs and subsequent level is liable for deployments of VMs to PMs. Diagrammatical illustration in Figure 2 depicts the scheduling model establishment.

Traditional CC techniques comprises of Greedy, FIFO, fair scheduling techniques [6], so forth. Then again, these scheduling techniques are accomplished depending on static types, and subsequently, there is no dynamic or adaptive alteration framework. Sadly, the resources of CC usually assigned dynamically and unconstrained, in this way, customary scheduling techniques must not guarantee practical necessities of scheduling of CC resource, and large number of resources are genuinely made to waste. There is an immense compact of study with motivation behind shows that scheduling resource isn't just a multi-objective upgrading issue in concentrate, but additionally a NP issue. So, a few Swarms Intelligence (SI) techniques are created and presented in CC technologies, those resource scheduling techniques are Particle Swarm Optimization, Genetic, Clone Selections, Shuffled Frog Leaping algorithms [7], and so forth. In this work, we present two new approaches Advanced Round Robin (ARR) technique and Fusion Methodology for scheduling and migration of VMs in Eucalyptus cloud environment. The ARR improves the conventional RR scheduling method by installing the procedure of resigning and resigning limits. The Fusion Methodology prepares a combination of ARR and First-Fit approaches of scheduling. Further, migration of VM is utilized to combine utilization of servers. The Power and Migration models also presented for estimating utilization of power and migration of VMs. The proposed techniques are analyzed and compare with the three existing scheduling techniques in Eucalyptus cloud such as Greedy, Round-Robin and Power Save.

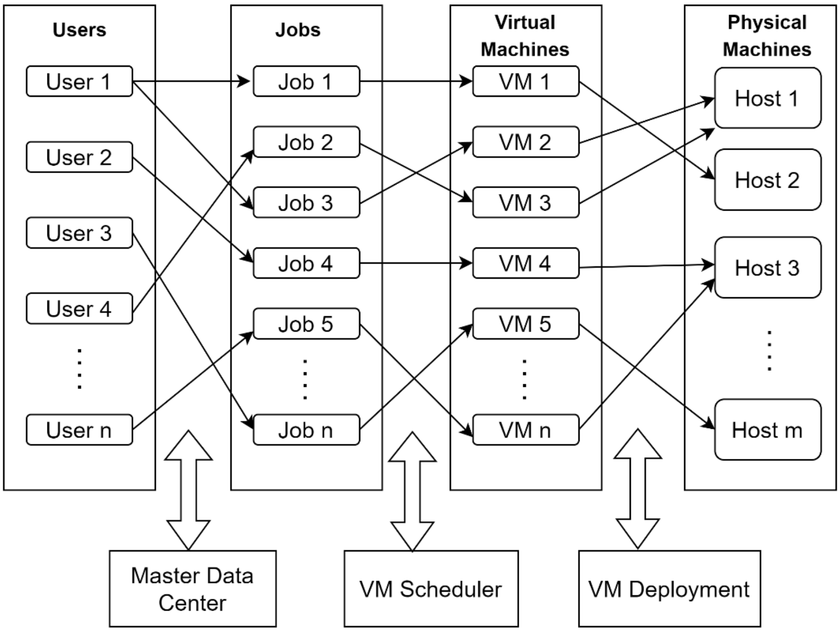

Figure 2 Scheduling model establishment

2. RELATED WORKS

Many efforts have been placed to VM scheduling researches in CC. The work in paper [8] presents an outline of CC technologies and similar investigation of VM scheduling techniques. VM scheduling strategies are analysed depending on different parameters like QoS, scalability, reliability and cloud environment. In study of researcher [9], they have basically broken-down different scheduling techniques by considering numerous metrics for assessment of performances like violation of SLA, power utilization, and so on. The paper [10] gives itemized survey of different investigations on various techniques with the motivation behind is disclosed to overcome normal challenges perceived in various scheduling jobs. In the event that scheduling jobs are accomplished productively, it outcomes to adjust load in cloud in view of resources and jobs. Because the numerous scheduling techniques are utilized by load balancers to figure out which of the backend servers to deploying VM a request. It is additionally obligation of provider to progressively redistribute or relocate VM across PMs for combination of workload and to keep away from over usage or underutilization of resource. In the work [11], authors describe a VM scheduling technique which considers already under execution VM resources utilization per time by validating past VM usage levels for scheduling VMs by optimized performance.

The applications in cloud are usually computation based and can fill exponentially in memory with increment in size in the event that no appropriate viable and effective load balancing strategy is taken on bringing about low-graded solution. To give a superior load balancing methods [12] for CC, with broad information, another novel model is being suggested which prepares classification on quantity of documents existing in cloud utilizing formatting of types of files. The Table 1 discusses the outline of scheduling techniques that are most usually utilized in CC. In CC, jobs are fetched to accessible VMs' which utilizes heuristic-based or metaheuristic-based mechanisms to give ideal ways that are usually not feasible in a given time by the traditional deterministic procedures.

TABLE 1. Survey of cloud computing scheduling techniques.

Mechanisms used	**Measures for QoS**	**Technology for Performance Evaluation**	**Merits**	**Demerits**
Scheduling- Load Balancing [13]	Review	Cloud Sim	Emerging domains detected	Fundamental review
Metaheuristic [14]	Response Time, Cost	Cloud Sim	Reliable, Cost and Response time reduced,	Higher cost of computation, Count of services limited
Scheduling- Energy aware [15]	Execution Time, Energy	Cloud Sim	Better Resource Utilization and energy efficiency, Minimized Execution Time	Process deadlines and count of tasks are limited
Scheduling- Load Balancing [16]	Response Time, Cost	Cloud Sim	Lower cost, Decreased Response Time	Complexity and count of tasks not discussed.
Framework for Resource Provisioning [17]	Response Time, Cost	Cloud Sim	Cost and Response Time reduced, Accuracy.	Energy efficiency and Throughput parameters not discussed
System for Controlling Elasticity [18]	Response Time, Elasticity, Utilization of Resources	Cloud Sim	Lower Response Time, Higher Resource Utilization and Elasticity	Challenges in Scalability
MFO based Scheduling [19]	Execution Time, Make span	iFog Sim	Minimized Execution time and makes pan	Less number of nodes are used and scalability not well established
BWM-VIKOR based Scheduling [20]	Utilization of VM, Throughput Time, Makes pan	Cloud Sim	Higher VM utilization, Minimum makes pan, higher throughput	Less number of VMs and tasks
BWM-TOPSIS based Scheduling [21]	Utilization of Resource, Makes pan, Energy Consumption	Cloud Sim	Higher VM utilization, Decreased Makes pan, Better energy consumption	Reliability challenges, small scale data centers considered
DVFS-PL based scheduling [22]	SLA Violation, Execution Time	Cloud Sim	Minimum SLA violation, Minimized Execution Time.	More number of VMs need to be considered.

Provisioning of Resources [23]	Utilization of Resource, Cost, Response Time	Cloud Sim	Minimized cost and Response Time.	Not considered the requirements of varying diverse users.
MOB and BAT-LBRC Scheduling [24][25]	Accuracy, Efficiency, merging clusters, Decision Making	Cloud Sim	Higher Efficiency and accuracy, Similar clusters Merging, Better decision making	Response time and throughput not addressed, Challenges on scalability.
System for PLB-HDD Optimization [26]	Cost of Execution, Makes pan	CWS	Reduced cost of execution, Better makes pan	Issues of scalability, a smaller number of VMs used.
AEFS-WOA and CSO-IRRO Scheduling [27][28]	Convergence, Execution Time, Throughput Time, Response Time	Cloud Sim	Faster Convergence, Minimized Response, Executing and Throughput Time	Issue of Scalability, a smaller number of datasets utilized, minimized capability in decision making.
TGA-EHO Scheduling [29]	Consistency, Location Search, Accuracy	Cloud Sim	Faster Location Search, Better Accuracy and consistency	Lesser number of nodes are considered.
SA-HHO Scheduling [30]	Scheduling of jobs, makes pan	Cloud Sim	Better scheduling of jobs, minimized makes pan	Lesser jobs and QoS metrics are considers.
EELBP Scheduling [31]	Energy Utilization, Response and Computation Time	Eucalyptus	Better Response Time and Energy Consumption, Minimized Computation Time.	Issues in scalability, ML approach not used.
ICSO Scheduling [32]	Clustering issues, F-Measures, CEC Function	MATLAB	Improved Clustering and CEC functions	No discussions about Energy Consumption, Response Time and Throughput.
Metaheuristic hybrid algorithm [33]	Makes pan, Throughput, execution time	Cloud Sim	Lower makes pan, Increased throughput, better execution time	Energy consumption issues not considered
PSO scheduling [34]	Execution Time, Accuracy	Google Cloud	Higher efficiency	Lower accuracy prediction
SLA-aware load balancing: Scheduling [35]	Migration Time, Energy Consumptions	MATLAB	Lesser migration time, improved	Execution time and throughput are not discussed

			energy consumption	
SLA-agile dependent VM scheduling [36]	SLA violations	Cloud Sim	Reduce SLA Violations	Metrics for QoS are missing
Scheduling: VM migration [37]	Overload, Communication cost	Cloud Sim	Lower overhead and communication cost	Degradation of performance, Computational complexities
P2P scheduling [38]	VM migration	JXTA	Fewer VM migrations	Higher computational and overhead cost,
Vanet Optimization-metaheuristic [39]	Network overhead, Energy	NS2	Decreased overhead, improved consumption of energy	Degradation of performance, increased cost of computations
ACO-BIOSARP scheduling [40]	Overhead Energy efficiency	NS2	Decreased overhead, improved energy efficiency	Efficiency in performance
MPSO-scheduling [41]	Utilization of Resource, Task overhead	Cloud Sim	Higher Resource utilization, reduced task overhead	Issue of scalability not discussed, a smaller number of VMs and tasks considered.
MLP-ABC Scheduling [42]	Accuracy	Cloud Sim NSL-KDD	Improved Kappa Value, MAE and RMSE	Other than accuracy, QoS metrics not validated

The private cloud Eucalyptus is utilized to construct the environment of CC. Eucalyptus upholds different hypervisors and Linux working frameworks [44], comprising five obligatory parts, like Walrus, Cloud Regulator, Cluster Regulator, Node Regulator and Storage Regulator. Cloud regulator settles on significant levels scheduling choices and perform sending of requests to Cluster regulators. It’s liable for maintaining virtualized resources. Cluster regulator plans VM execution on explicit nodes and deals with VM instances. Storage regulator operates with cluster regulator. Node regulator is responsible for beginning and halting VMs. Eucalyptus makes use of Round Robin (RR), Greedy or Power Save scheduling mechanisms for VM scheduling in the cloud. RR scheduling mechanism for VM focus on distribution of loads to all hosts equally, where scheduler assigns one VM to each host in recurrent manner.

The scheduler begins allocating VMs to every host and goes to subsequent VM to convey following host. This mechanism is rehashed for entire hosts until every host have no less than one VM. The primary benefit of RR technique for VM scheduling is that it utilizes entire resources in a fair request. Equivalent count of VMs are assigned to all hosts that ensure reasonableness. Significant disadvantage of utilizing RR mechanism in VM scheduling is that power utilization would be higher as multiple hosts will be made ON for quite a while. While Greedy VM technique picks initial node that can meet underlying prerequisites. Therefore, power utilization is lesser in Greedy yet load balancing isn't accomplished. The Power Save technique for VM scheduling saves more power compared to these two.

Another private cloud Open Nebula upholds numerous hypervisors and operating frameworks, having parts like front-end, hypervisor empowered hosts, information stores, networks for services, and VM networks [45]. Open Nebula make use of Match Making technique for VM scheduling in cloud. The Match making technique allots hosts with high ranked expressions first to VMs, which is significant in application of strategies like striping, packing and load aware policies. Few of the authors also tried apply parallel [46][47] mechanisms to make use of multicore systems in CC, which minimizes switching times.

The cloud OpenStack contains components like Compute, Dashboard, Block storages, Object storages, Identity services, Database and Image services [48]. The components of OpenStack cloud utilize RMQ Protocol for internal communications. VM scheduling is performed via Nova scheduler, which make use of Filter scheduling technique, that utilizes filtering and process of weighing to plan the VMs in Nova computing Node hosts. This filtering of computing node hosts would be performed depending on parameters of filters. Weighing operation will be initiated depending on filtered hosts. The comparisons of VM scheduling techniques in Eucalyptus, OpenStack and Open Nebula clouds are briefly compared in Table-2.

Table 2 Comparing VM scheduling algorithms in clouds

Cloud Environments	VM Scheduling Techniques	Major Properties
OpenStack	Filter Scheduling algorithm	Memory aware based
Open Nebula	Match making algorithm	Cost effectiveness
Eucalyptus	Round Robin algorithm	Time efficiency
	Greedy algorithm	Less power consumption
	Power Save algorithm	Power saver

3. METHODOLOGY AND SYSTEM MODELS

Two models used in our VM system described here includes Power utilization model that estimates usage of power of PM, and Migratory model describing relocation techniques and measurement of costs incurred.

The previous investigations [49] analyses the power utilization of physical machines as amount of inactive power and a Processor load's linear operation. To check power utilization is a linear operation of Processor load, power measuring meters that are built in PM are utilized to quantify usage of power under various Processor loads. The processor load is described as the total loading of every core, for instance, for eight cores containing 100% loading, the PM's processor load would be 800%. The illustration of experimental outcome is shown in Figure 3, in which x-axis indicates count of cores with cent percent loading, where as y-axis indicates power utilization recording by power meters. The PM utilized four processors, each containing four cores. Outcome checks linearity of power utilization as a component of quantity of cores having cent percent loading. The smaller lines demonstrate the power utilization from sending intensely loaded VMs on cores of a similar Processor, and the more extended lines shows the power utilization while conveying VMs to two unique processors however much as could be expected.

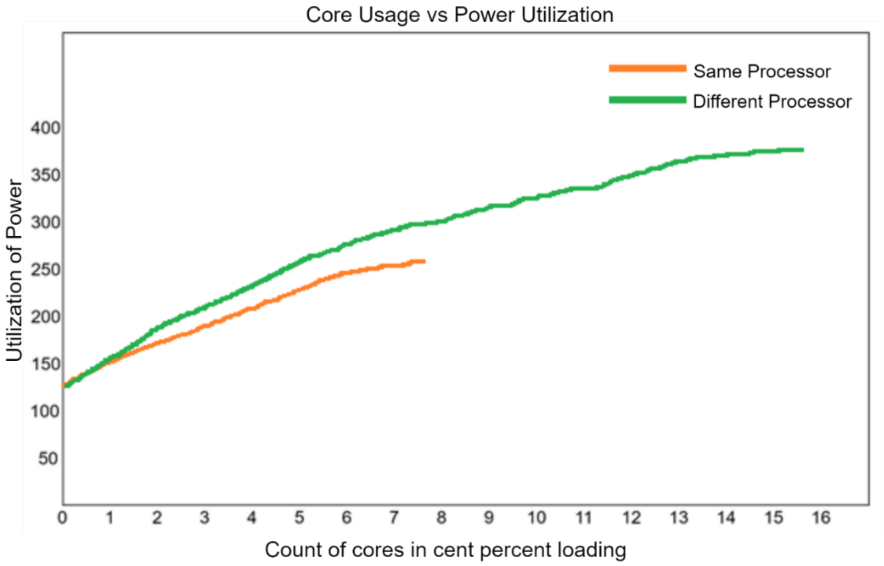

Figure 3 Demonstration of Power Utilization under various Processor loads

The proportion between top power and Inactive power can be analysed, where Top power (T_p)is power utilization when entire sixteen cores of PM contain cent percent loading, that is 1600% in above Figure 3. Inactive power (I_p) indicates the power utilization when none of sixteen cores has any loading. From the analysis in Figure 3, perception is that Inactive power is around half of the Top power. In this way, power utilization (PU) of PM is assessed utilizing the linear power model mathematically as given in equation below.

$$\mathrm{PU} = \left[\frac{C_v}{C_p}(1-\beta)+\beta\right]T_p$$

Where β indicates the level of inactive power versus top power. Considering above Figure 3, β is set to be half, Cv is complete count of cores needed by resident VMs, and Cp is complete count of cores of PM.

In Migratory model [50], live migration is adopted that permits a administrator of server to transfer a running VM to an alternate PM without intruding on execution of VM. Subsequently, the execution time of a VM won't be impacted by relocation. Albeit live relocation doesn't extend the execution time of a VM, it builds energy utilization of sending and receiving PMs where movement happens. At the point when relocation begins, the receiving PM will make a operation to duplicate contents of memory from sending PM, expanding the load, consequently expanding the complete power utilization. Based on experimental outcomes, the energy utilization in the time of relocation is characterized mathematically as in equation below.

$$U_E = U_S + U_D$$

$$U_S = \left[\frac{C_v}{C_p}(1-\beta)+\beta\right]T_p M_t$$

$$U_D = \left\{\left[\frac{C'_v}{C_p}+L\right](1-\beta)+\beta\right\}T_p M_t$$

Where U_E is utilization of energy of migrating VM from sending PM to receiving PM, C_v is quantity of cores needed by moving VM, C_v/C_p and C'_v/Cp are percentage loading of sending and receiving machines accordingly. M_t is migration time, and L is additional load created by process of migration. VMs with various hardware prerequisites, like the quantity of cores and sizes of memory, should be made to deploy to PMs with static hardware limit in a manner which decreases quantity of PMs utilized. This fusing problem is harder and the First-Fit strategy could be changed and applied to this fusing problems. First-Fit technique tries deploying a VM to initial machine in a PM list that can oblige this VM. In the event that no PM is found, another PM will be booted to have VM. Notwithstanding that First-Fit technique is a straightforward and successful heuristic, it may not be a decent answer for our fusing problem.

In this paper, we consider little complicated VM fusing problem. Along with resource prerequisites, each VM additionally contain some Time of Execution and Time of arrival. Each VM only available during its execution time, that begins at its time of arrival, and goes on for its time of execution. After a VM completes it leaves framework and cores it utilized become inactive before they are assigned further to rest of the VMs. Those inactive cores might build the genuine count of PMs needed to run entire VMs. We present two VM scheduling strategies for eucalyptus framework, Advanced Round-Robin (ARR) and a Fusion methodology for VM fusing. The intension of these strategies is to limit the quantity of PMs utilized to run entire VMs. This objective is vital in light of the fact that the quantity of PMs utilized strongly influences absolute power utilization.

Advanced Round-Robin (ARR): This technique is presented as an expansion to Round-Robin strategy. ARR strategy utilizes dual guidelines to assist with fusing VMs. The principal guideline is that, suppose a VM has completed and there are more VMs facilitated on same PM, this PM will not acknowledge any newer VM. Such PMs are alluded to as being in resigning state, implying that when the other VMs complete their executions, this PM can be close down. Second guideline is that suppose a PM is in resigning state for an adequately larger stretch of time, rather than holding up for completion of VMs which are residing, the PM will be compelled to move the other VMs to other PMs, and closure after the relocation completes. This limit of holding up time is signified as resigning limit. PM that is in resigning state yet can't complete all VMs after the resigning limit will be compelled to move its VMs and close down. Our ARR technique involves these two guidelines for fusing VMs deployed by the fundamental RR strategy. The first guideline abstains from adding additional VMs to a resigning PM. The second guideline speeds up fusing operation and empowers ARR to close down PMs, with the goal that it can decrease the count of PMs used to run every single VM, hence accomplish saving of power.

Fusion methodology: For the conservation of more energy the ARR and First-Fit approaches are consolidated in to Fusion Methodology. The count of VMs which are incoming is thought to be a component of time, and follows a normal distribution. Our Fusion Methodology will utilize VM’s incoming rate to direct the VM scheduling. The primary thought of the Fusion Methodology is to figure out which technique to utilize in light of the incoming rate of VMs. The First-Fit strategy is appropriate for higher incoming rate VMs. By realizing there will be countless incoming VMs, it isn't important to be moderate in utilizing resources as the demands will continue to come. The Fusion Methodology utilizes First-Fit during times of heavy traffic to completely use computing power of PMs, and uses ARR for fusing VMs and decrease energy utilization during non-busy times.

Resigning Limit (RL): The main issue in implementation is to decide a legitimate resigning limit. The RL can be predicted by considering the time required for migration and VM's leftover execution time. The instinct is that, suppose a VM nearly completes its execution, it isn't important to migrate. Again, if leftover execution season of a VM is still long, fusing it with other VMs is probably going to saving of energy. Two conditions of general energy utilization, one with choice to migrate a VM and other without migrating a VM, are presented below, which depicts that the energy utilization is no different for either migrating or without migrating. Legitimate RL can be calculated from these conditions. Energy utilization equation for migration of VM is presented in previous section, and the energy utilization equation for not migrating the VM is presented below.

$$\mathrm{U_E = U_S + U_D}$$

$$\mathrm{U_S} = \left[\frac{\mathrm{C_v}}{\mathrm{C_p}}(1-\beta)+\beta\right]\mathrm{T_p R_t}$$

$$\mathrm{U_D} = \left[\frac{\mathrm{C'_v}}{\mathrm{C_p}}(1-\beta)+\beta\right] + \mathrm{T_p R_t}$$

Where U_E and U_D are utilization of energy of sending and receiving PMs when not migrating the VM, respectively. At the time the PM resign, in place of taking decision to migrate every VM by its leftover time for execution, R_t, that is unknown, the limit α is utilized as resigning limit, given below.

$$\alpha = (1-\mathrm{L})\mathrm{M_t} + \frac{\mathrm{L}}{\beta}$$

Assuming a VM with leftover execution time R_t less equivalent compared to α, means that, not migrating is higher saving of energy, the VM will ultimately complete before PM begins to migrate VMs. Then again, on the off chance that R_t is larger compared to α, means that, migrating is a superior decision, VM will be migrated after the resigning limit exceed. The leftover time for execution R_t isn't required while running ARR technique. The limit α is determined and fixed. A resigning VM will be closed down after a holding up time of α. Each incomplete VM would be migrated regardless of how long leftover time for execution is.

4. RESULTS AND DISCUSSIONS

The experimental environment is setup using homogenous cluster of eight nodes, in which every PM has quad-core processor, by adopting a power loading model having inactive power as fifty percent of top power. The value of L (cost of migration) is taken as $0.025 \times 1/8 \times 0.2$, so that every migration activity maximizes Twenty percent loading to one among 8 cores in PM, where M_t is considered as 2-time units. Every Testing data record consists a VMs set with every VM being part of small-scale or large-scale type, time of arrival and time for executions. The normal distribution is used to produce time of arrival for VMs and time for execution of VM is random which varies from 2 to 10 hours.

The small-scale experimentation is carried out to evaluate power and migration models by installing Hypervisor Xen on every PM for process of virtualization. The data estimated by proposed power model and data of actual usage of power measured using power meters are collected. The implementation of VM provision and fusing framework is done. The system produces VMs as per their time of arrival, and a scheduling technique of VM predicts which PM must host the arriving VM. Once the VM is generated on PM, it begins to execute and maintains core with cent percent loading. The utilization of power is noted down for each minute and the mean utilization of power is computed at the end. The comparisons of proposed ARR technique with First-Fit power save approach is tabulated in Table 3, where we can find minor difference among power estimated and measured, because the VM will be loaded 100% whereas the actual loads are complicated. There can also be the reason for this discrepancy is that, proposed power model slightly over estimates the overhead of power. Hence, estimation done by proposed model can be results as enough accurate to validate various scheduling techniques.

Table 3 Results of Power Utilization using ARR and First-Fit Power save

	Power Estimated (W)	**Actual Power from Power Meters**
ARR	726.63	690.30
First-Fit Power save	694.20	659.28
Achieved improvements	32.43	31.02

For experimenting with large-scale VMs, the simulation is done by connecting 500 octa-core servers to ethernet switches. Every run of a simulation considers 3000 arriving VMs, by recording count of PMs utilised, migrations count and usage of energy at regular time units during the simulation.

Initially, various resigning limits such as 10, 20 ad 30 are applied in simulation to determine affects in performance by the limits, along with the processor load to predict the resigning of PM. The graphical analysis in Figure 4 demonstrates the mean power utilization under various resigning limit using ARR compared with RR and First-Fit techniques. Taking power utilization of First-Fit power saving technique as baseline, the normalization of mean power utilization is achieved. The first three columns in graph of Figure 4 are the outcome of ARR by making use of various resigning limits RL10, RL20 and RL30, fourth column marked 'PL' utilizes Processor Load as resigning criteria, and fifth and sixth columns are outcome of RR and First-Fit power saving techniques. The analysis proves that when resigning limit reduces, the saving of power increases.

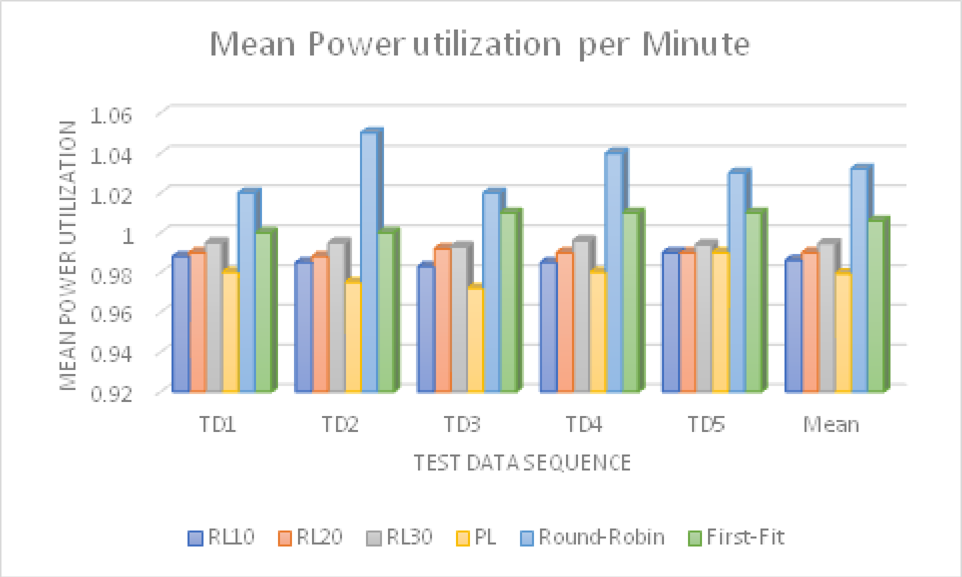

Figure 4 Mean Utilization of Power under various Resigning limit utilizing ARR

Considering First-Fit as basis, ARR and Fusion Methodology are compared and analysed with other techniques such as Best-Fit, RR and First-Fit, which are executed in power saving ways by closing down inactive PMs to save energy. The metrics such as mean utilization of power, mean count of

PMs used and migrations count are taken in to consideration. The Fusion Methodology need to be defined in busy hours. Because of the time of arrival of VMs are normally distributed, busy hours described as the time intervals focused in mean time of arrival and addition or subtraction of standard deviation. The graphical analysis in Figure 5 demonstrates normalized power utilization of five techniques considering five test data sequences. The ARR and Fusion Methodology make use of PM loads as resigning limit, as it conserves maximum energy. The analysis proves that Fusion Methodology performs better compared to other techniques.

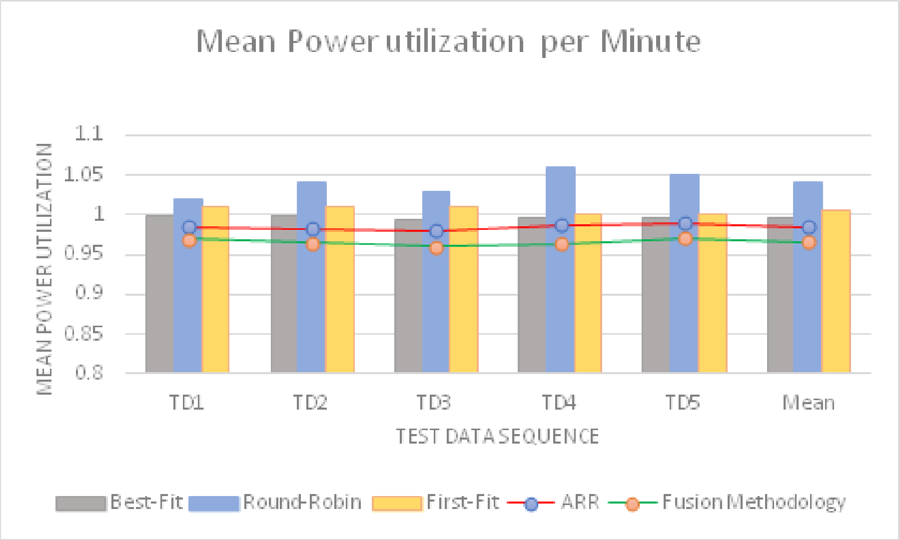

Figure 5 Comparing mean power utilization of proposed technique with other techniques

Performance Analysis with Eucalyptus VM Scheduling Techniques: Three major VM scheduling techniques in Eucalyptus such as RR, Greedy and Power Save, in which first two techniques do not close down the machines, and Power Save behaves similar to First-Fit power saving approach. The power utilization of these three existing eucalyptus scheduling techniques is measured and compared with proposed ARR and Fusion Methodology approaches. The count of powered-on PMs in RR and Greedy is equal to count of PMs needed to operate all arriving VMs. The graphical analysis done in Figure 6 and Figure 7 demonstrates mean count of powered-on PMs and mean power utilization for those five techniques, taking Eucalyptus RR as baseline. The analysis done shows that the count of powered-on PMs and power utilization decreases drastically when proposed techniques are used compared to existing eucalyptus techniques.

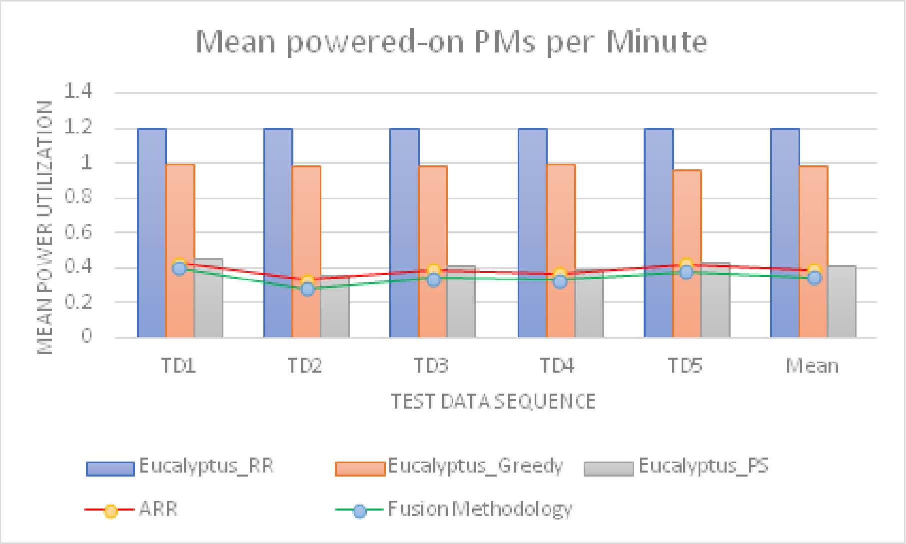

Figure 6 Analysis of mean powered-on PMs using RR base

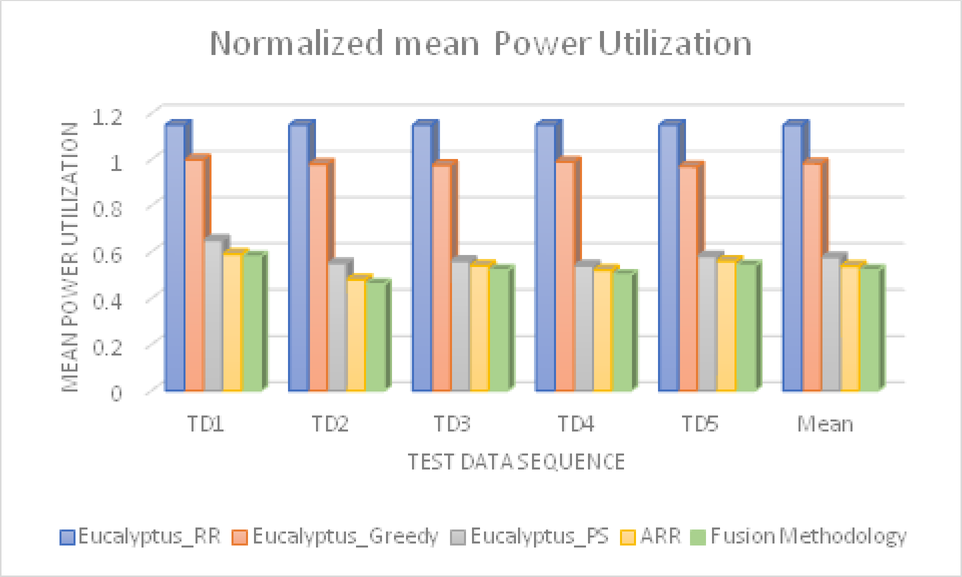

Figure 7 Analysis of normalized mean power utilization using RR base

5. CONCLUSION

The work showed the adequacy of proposed techniques in both smaller-scale and larger-scale simulated experiments. Two strategies Advanced Round-Robin which improves the traditional RR technique and Fusion Methodology which combines ARR with First-Fit strategy, for VM scheduling in Eucalyptus cloud for intention of saving power. The experimental outcomes validated the framework proposed and proved that ARR utilizes lesser energy compared to existing approach. The simulated results demonstrated that utilizing PM load as resigning limit saves maximum energy compared to involving time as resigning limit. The mean power utilization of proposed VM scheduling technique for Eucalyptus cloud is analysed and compared with various existing techniques, and proved that proposed techniques minimal mean power utilization. The Fusion Methodology ensures higher performance compared to all other techniques, that minimizes quantity of powered-on PMs and mean utilization of power.

REFERENCES

[1] Heyang Xu, Yang Liu, Wei Wei, Wenqiang Zhang, “Incentive-aware virtual machine scheduling in cloud Computing”, The Journal of Supercomputing, 2018, 74:3016-3038, doi:10.1007/s11227-018-2349-y

[2] Madni, Hamid & Shafie, Abd Latiff & Yahaya, Coulibaly & Abdulhamid, Shafi’i, “Resource Scheduling for Infrastructure as a Service (IaaS) in Cloud Computing: Challenges and Opportunities”, Journal of Network and Computer Applications, 2016, 68, doi: 10.1016/j.jnca.2016.04.016.

[3] R. Zhang, K. Wu, M. Li and J. Wang, "Online Resource Scheduling Under Concave Pricing for Cloud Computing," in IEEE Transactions on Parallel and Distributed Systems, 2016, vol. 27, no. 4, pp. 1131-1145, doi: 10.1109/TPDS.2015.2432799.

[4] Jennings B, Stadler R, “Resource Management in Clouds: Survey and Research Challenges”, Journal of Network and Systems Management, 2015, 23, 567–619, doi:10.1007/s10922-014-9307-7.

[5] L. Liu and Zhe Qiu, "A survey on virtual machine scheduling in cloud computing," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), 2016, pp. 2717-2721, doi: 10.1109/CompComm.2016.7925192.

[6] Eawna, Marwah & Hamdy Elsayed, Salma & El-Horbarty, El-Sayed, “Hybrid Algorithm for Resource Provisioning of Multi-tier Cloud Computing”, Procedia Computer Science, 2015, 65. 682-690. 10.1016/j.procs.2015.09.012.

[7] Eswaraprasad, Ramkumar & Raja, Linesh, “A review of virtual machine (VM) resource scheduling algorithms in cloud computing environment”, Journal of Statistics and Management Systems, 2017, 20, 703-711, doi: 10.1080/09720510.2017.1395190.

[8] M. Kumar, Suman, S. Singh, “Survey on Virtual Machine Scheduling Algorithms in Cloud Computing”, International Journal of Computer Sciences and Engineering, 2018, Vol.6(3), E-ISSN: 2347-2693.

[9] Samarth Bhandari, Ihit Srivastava, S. Nagadevi, “Comparative Study of Virtual Machine Scheduling Algorithms”, International Journal of Pure and Applied Mathematics, 2017, Volume 115 No. 6, 697-703.

[10] S. Rekha, C. Kalaiselvi, “Review Of Scheduling Methodologies Of Virtual Machines (Vms) In Heterogeneous Cloud Computing”, International Journal Of Scientific & Technology Research, 2019, Vol 8, Issue 9, ISSN 2277-8616.

[11] Stelios Sotiriadis, Nik Bessis, Rajkumar Buyya, “Self-managed virtual machine scheduling in Cloud systems”, Information Sciences, 2018, 381–400, doi: 10.1016/j.ins.2017.07.006.

[12] Muhammad Junaid, Adnan Sohail, Adeel Ahmed, Abdullah Baz, Imran Ali Khan, And Hosam Alhakami, “A Hybrid Model for Load Balancing in Cloud Using File Type Formatting”, IEEE Access, 2020, Vol 8, doi: 10.1109/ACCESS.2020.3003825.

[13] Pradhan A., Bisoy S.K., Mallick P.K, “Load Balancing in Cloud Computing: Survey”, In: Sharma R., Mishra M., Nayak J., Naik B., Pelusi D. (eds) Innovation in Electrical Power

Engineering, Communication, and Computing Technology, 2020, Lecture Notes in Electrical Engineering, vol 630. Springer, Singapore. doi: 10.1007/978-981-15-2305-2_8.

[14] Ghobaei-Arani, Mostafa & Rahmanian, Ali & Aslanpour, Mohammad Sadegh & Dashti, Seyed, "CSA-WSC: cuckoo search algorithm for web service composition in cloud environments", Soft Computing, 2018, 22, doi: 10.1007/s00500-017-2783-4.

[15] Safari, Monire & Khorsand, Reihaneh, "Energy-aware scheduling algorithm for time-constrained workflow tasks in DVFS-enabled cloud environment", Simulation Modelling Practice and Theory, 2018, 87, doi: 10.1016/j.simpat.2018.07.006.

[16] A. K. Duggal and M. Dave, "A comparative study of load balancing algorithms in a cloud environment',' Advances in Computing and Intelligent Systems,2020, Singapore: Springer, pp. 115-126.

[17] Ghobaei-Arani, Mostafa & Khorsand, Reihaneh & Ramezanpour, Mohammadreza, "An autonomous resource provisioning framework for massively multiplayer online games in cloud environment", Journal of Network and Computer Applications, 2019, doi: 10.1016/j.jnca.2019.06.002.

[18] M. Ghobaei-Arani, A. Souri, T. Baker and A. Hussien, "ControCity: An Autonomous Approach for Controlling Elasticity Using Buffer Management in Cloud Computing Environment," IEEE Access, 2019, vol. 7, pp. 106912-106924, doi: 10.1109/ACCESS.2019.2932462.

[19] Ghobaei-Arani, Mostafa & Souri, Alireza & Safara, Fatemeh & Norouzi, Monire, "An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing", Transactions on Emerging Telecommunications Technologies, 2020, 31, doi:10.1002/ett.3770.

[20] Rafieyan, Elham & Khorsand, Reihaneh & Ramezanpour, Mohammadreza, "An Adaptive Scheduling Approach based on Integrated Best-worst and VIKOR for Cloud computing", Computers & Industrial Engineering, 2020, 140, 106272, doi: 10.1016/j.cie.2020.106272.

[21] Khorsand, Reihaneh & Ramezanpour, Mohammadreza, "An energy-efficient task-scheduling algorithm based on a multi-criteria decision-making method in cloud computing", International Journal of Communication Systems 2020, 33, doi:10.1002/dac.4379.

[22] Safari, Monireh & Khorsand, Reihaneh, "PL-DVFS: combining Power-aware List-based scheduling algorithm with DVFS technique for real-time tasks in Cloud Computing", The Journal of Supercomputing, 2018,74, 1-23, doi:10.1007/s11227-018-2498-z.

[23] Khorsand, Reihaneh & Ghobaei-Arani, Mostafa & Ramezanpour, Mohammadreza, "A self-learning fuzzy approach for proactive resource provisioning in cloud environment", Software: Practice and Experience, 2019, 49, doi: 10.1002/spe.2737.

[24] Strumberger, Ivana & Tuba, Eva & Bacanin, Nebojsa & Beko, Marko & Tuba, Milan., "Modified and Hybridized Monarch Butterfly Algorithms for Multi-Objective Optimization", 2020, doi: 10.1007/978-3-030-14347-3_44.

[25] Adhikari, Mainak & Nandy, Sudarshan & Amgoth, Tarachand, “ Meta heuristic-based task deployment mechanism for load balancing in IaaS cloud”, Journal of Network and Computer Applications, 2018, 128, doi:10.1016/j.jnca.2018.12.010.

[26] Kaur, Amanpreet & Kaur, Bikrampal, “Load Balancing Optimization based on Hybrid Heuristic-Metaheuristic Techniques in Cloud Environment”, Journal of King Saud University - Computer and Information Sciences, 2019, doi: 10.1016/j.jksuci.2019.02.010.

[27] Strumberger, Ivana & Bacanin, Nebojsa & Tuba, Milan & Tuba, Eva, “Resource Scheduling in Cloud Computing Based on a Hybridized Whale Optimization Algorithm”, Applied Sciences, 2019, 9, 4893, doi:10.3390/app9224893.

[28] Torabi, Shadi & Safi, Faramarz, A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing The Journal of Supercomputing 2018, 74, doi: 10.1007/s11227-018-2291-z.

[29] Bhandary, Abhir & G, Ananth & Basthikodi, Mustafa & M, Chaitra, “Early Diagnosis of Lung Cancer Using Computer Aided Detection via Lung Segmentation Approach”. International Journal of Engineering Trends and Technology, 2021, 69. 85-93, doi:10.14445/22315381/IJETT-V69I5P213.

[30] Attiya, Ibrahim et al. “Job Scheduling in Cloud Computing Using a Modified Harris Hawks Optimization and Simulated Annealing Algorithm”, Computational Intelligence and Neuroscience, 2020, doi: 10.1155/2020/3504642.

[31] Patel, Durga & Gupta, Rajeev & Pateriya, R, “Energy-Aware Prediction-Based Load Balancing Approach with VM Migration for the Cloud Environment”, 2019, doi:10.1007/978-981-13-6351-1_6.

[32] Kumar, Yugal & Singh, Pradeep, “Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering”, Applied Intelligence, 2018, 48, doi:10.1007/s10489-017-1096-8.

[33] Anwar, Nazia & Deng, Huifang, “A Hybrid Metaheuristic for Multi-Objective Scientific Workflow Scheduling in a Cloud Environment”, Applied Sciences, 2018, 8, 538, doi:10.3390/app8040538.

[34] Zhong, Wei & Zhuang, Yi & Sun, Jian & Gu, Jingjing, “A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine”, Applied Intelligence,2018, 48, doi:10.1007/s10489-018-1194-2.

[35] M. Ashouraei, S. N. Khezr, R. Benlamri and N. J. Navimipour, "A New SLA-Aware Load Balancing Method in the Cloud Using an Improved Parallel Task Scheduling Algorithm", 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), 2018, pp. 71-76, doi: 10.1109/FiCloud.2018.00018.

[36] N. Sharma and S. Maurya, "SLA-Based Agile VM Management in Cloud & Data center," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMIT Con), 2019, pp. 252-257, doi: 10.1109/COMITCon.2019.8862260.

[37] Xiao Song, Yaofei Ma and Da Teng, “A Load Balancing Scheme Using Federate Migration Based on Virtual Machines for Cloud Simulations”, Mathematical Problems in Engineering, 2015, doi: 10.1155/2015/506432.

[38] Rouzaud-Cornabas, Jonathan, “A Distributed and Collaborative Dynamic Load Balancer for Virtual Machine”, 2010, 6586. 641-648, doi:10.1007/978-3-642-21878-1_79.

[39] Toutouh, Jamal & Alba, Enrique, “Metaheuristics for energy-efficient data routing in vehicular networks”, International Journal of Metaheuristics, 2015, 4, No. 1. 27-56. Doi:10.1504/IJMHEUR.2015.071750.

[40] Saleem, Kashif & Fisal, Norsheila, “Enhanced Ant Colony algorithm for self-optimized data assured routing in wireless sensor networks”, IEEE International Conference on Networks, ICON, 2012, 422-427. Doi:10.1109/ICON.2012.6506595.

[41] Mohanty, Subhadarshini & Patra, Prashanta & Ray, Mitrabinda & Mohapatra, Subasish, “A Novel Meta-Heuristic Approach for Load Balancing in Cloud Computing”, International Journal of Knowledge-Based Organizations,2018, 8, 29-49, doi: 10.4018/IJKBO.2018010103.

[42] Hajimirzaei, Bahram & Navimipour, Nima, “Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm”, ICT Express, 2018, 5, doi: 10.1016/j.icte.2018.01.014.

[43] Manoj Mathew, “Virtualization and Scheduling In Cloud Computing Environment – A Study”, IOSR Journal of Computer Engineering (IOSR-JCE), 2018, Vol 20, Issue 4, Ver. III, PP 23-32, Corpus ID: 215821421.

[44] Nandimandalam Mohan Krishna Varma, Eunmi Choi, “Study and Comparison of Virtual Machine Scheduling Algorithms in Open-Source Clouds”, Springer Science+ Business Media Singapore, 2016, J.J. (Jong Hyuk) Park et al. (eds.), Advanced Multimedia and Ubiquitous Engineering, Lecture Notes in Electrical Engineering, 393, doi:10.1007/978-981-10-1536-6_46.

[45] Varma NMK, Choi E, “A scalable grid infrastructure in the eucalyptus cloud environment”, The second international conference on computers, networks, systems, and industrial applications, 2012, pp 286–289.

[46] Basthikodi, M., Faizabadi, A. R., & Ahmed, W., HPC Based Algorithmic Species Extraction Tool for Automatic Parallelization of Program Code. International Journal of Recent Technology and Engineering, 8(2S3)(2019) 1004–1009. https://doi.org/10.35940/ijrte.b1188.0782s319

[47] M. Basthikodi and W. Ahmed, "Classifying a program code for parallel computing against HPCC," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2016, pp. 512-516, doi: 10.1109/PDGC.2016.7913248.

[48] Varma NMK, Min D, Choi E, “Diagnosing CPU utilization in the Xen Virtual Machine Environment”, 6th international conference on computer sciences and convergence information technology, 2011, pp 58–63.

[49] Gong Chen, Wenbo He, Jie Liu et al, “Energy-aware server provisioning and load dispatching for connection-intensive internet services,” Proceedings of the 5th USENIX Symposium on

Networked Systems Design and Implementation, NSDI'08, pages 337–350, Berkeley, CA, USA, 2008. USENIX Association.

[50] Lin, Ching-Chi & Liu, Pangfeng & Wu, Jan-Jan, "Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems", Proceedings - 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011, 81-88, doi: 10.1109/UCC.2011.21.