Energy Efficient Task Scheduling in Cloud Using Underutilized Resources

Sumeet Bharti
Department of Computer Science & Engineering
Punjabi University
Patiala,Punjab(India)
bsumeet550@gmail.com

Navneet Kaur Mavi
Department of Computer Science & Engineering
Punjabi University
Patiala,Punjab(India)
mavi_navneet@yahoo.co.in

***ABSTRACT*—Resource scheduling and provisioning in cloud environment are most challenging due to the execution variability and uncertainty of the cloud infrastructure and of the load being set up. In this framework, the task scheduling VM allocation using underused resources has been implemented with the concept of RAM and mips and compared it with the existed Energy-Performance Trade-Off Multi-Resource Cloud Task Scheduling Algorithm. The underutilized elements are found out in this policy. In this way, it uses more number of processing elements as compared to the existing algorithm. The experimental results demonstrate that the resources are being utilized properly in order to reduce the overhead, energy consumption and execution time.**

***Keywords*—Energy consumption, Resource allocation, VM allocation, Task Scheduling**

I. INTRODUCTION

Cloud computing is a setup that is promptly and flexibly equipped on request [1]. Diverse virtual machine configurations which are proficient in implementing workflows are accessible in it. Majorly clouds are classified into public, private, community- based, or hybrid clouds. Public cloud portions amenities through numerous occupants and are debited according to the usage. Private cloud is possessed and handled by a particular association, and it does not debit the amenities it provides. Community-based clouds are planned for a particular community that has shared concerns. Finally, hybrid cloud consists of a mix of public, communal and private cloud, where communal/public cloud assets are combined using private cloud to generate particular surroundings. Here is a extensive kind of cloud service area (e.g., storage policy, virtual machine (VM) types, and cloud data center localities) [2] [3]. That's why, scheduling is a vital feature in a workflow implemented in cloud surroundings and has been the focus of exploration in current ages [4]. Scheduling procedures are planned to outline the jobs of the workflow for VMs founded on scheduling standards and focus to the limits of the operators.

Though, the extensive collection of cloud services and the heterogeneousness of these tasks, lead to an NP-hard optimization problem. Performing optimum scheduling in a realistic period is a test as certain variables need to be considered, for instance, the statement that here are (a) numerous sorts of VMs with poles apart capabilities and rates, (b) tasks with dissimilar loads, and (c) additional characteristics that obstruct the method of optimization.

On account of this, metaheuristic methods have been implemented in workflow setting up to found a near-optimal scheduling scheme in clouds. Nevertheless, exploration on the usage of metaheuristics for workflow scheduling in clouds have been majorly constrained to a distinct procedure. In latest research, it was analyzed that Particle Swarm Optimization (PSO) is the broadly applied method, and above 50% of the scheduling systems were founded on metaheuristics [5]. These verdicts recommend that here is a requirement to comprehend whether scheduling procedures that are enhanced by metaheuristics can attain an improved act than any substitute and whether PSO is, indeed, a worthy optimal.

The remainder of the paper is arranged as follows: Segment 2 audits the related work. Section 3 frames the existed system simulations comprehensively. The proposed VM allocation policy is represented in Section 4. Section 5 illuminates the experimental design and inspects the outcomes of the evaluation. Lastly, in Section 6 there are certain concluding remarks and proposals for future investigation in the area.

II. RELATED WORK

A) Workload scheduling

We predominantly emphasis on failure-aware or/and energy-aware workload scheduling strategies planned for multi-node schemes (e.g. datacenters and clusters) and present the demonstrative mechanisms. Owing to their inherit convolution, structure engineers must study a surplus of features, comprising energy norm (cooling, servers), server temperature threshold, failures (tasks, servers, network, middleware), and Quality of Service (QoS). Load scheduling can be signified as an optimization problem (e.g. maximize task completion rate, or minimize energy usage) underneath particular restrictions (enforced by an SLA).

Load association [6] is a well-organized technique to lessen computing energy by reducing the amount of active nodes [7]. Workload association is frequently demonstrated as a bin packing problematic that has been verified to be NP-complete [8]. To state this problem, investigators susceptible to accomplish the process proficiency at the charge of compact solution accurateness. Coffman et al. [9] offered a modest and instinctive procedure called first-fit declining, whereas Young et al. projected MaxUtil and ECTC to associate loads [6], by the previous trying to make the most of period of tasks running in corresponding, and the later exploiting average CPU operation throughout implementation.

Contradictory from the overhead mechanisms that merely studies workload association to lessen work out energy, exploration accompanied by Yoursri et al. [10] create considerable development in conjoining computing energy decline by CPU temperature switch. Moore et al. [11] planned a system-level explanation to regulate heat generation over temperature aware workload assignment to decrease conserving energy. Tang et al. offered the XInt algorithm [12] to accomplish cooling proficiency via diminishing heat recirculation and peak inlet temperature. A proactive control methodology is planned in [13] that cooperatively enhance the air conditioner compressor liability sequence and fan speed to decrease cooling charge and diminish threat of apparatus destruction because of high temperature. Ayan et al. [14] planned a hybrid technique for synchronized cooling-aware job employment and active cooling organization. The algorithm assigns jobs to decrease the cooling demands of the CRACs, and at that point informs the CRAC control locations founded on high temperature supply simulations.

The exceeding scheduling strategies do not study structure letdowns. Though, letdown happenings in these computing schemes can persuade considerable negative influence on system routine, differing the system from first points [15], particularly for those setups that need assurance enforced constrictions (e.g. a real time system can express severe task targets) [16]. To switch constituent letdowns in computing arrangement, Song et al. demarcated a reliability–capacity metric to concatenate the properties of server reliability and capacity status throughout node collection [17]. They anticipated Pessimistic Best-Fit (PBFIT) and Optimistic Best-Fit (OBFIT) algorithms to define the greatest capable servers on which to incorporate VMs to employ user jobs. Experimentations presented that the advanced degree of productively accomplished jobs was attained by means of PBFIT and OBFIT policies. Though, their scheduling procedures do not study the energy proficiency of a scheme.

The pseudo-code of the ETMCTSA algorithm is as follows [19]:

Altino et al. do additional enhancements [15] through their offered process leveraging proactive fault-tolerance methods to contract with schemes letdowns, and integrating a metric named power-efficiency to regulate the finest contender server. Bahman [18] examine failureaware and energy load planning at the level of federated/hybrid datacenters. They recommend an ascendable hybrid Cloud organization along with source provisioning strategies to assure QoS objectives of the operators, considering the workload model and failure association to forward users' applications to the suitable Cloud sources.

III. Energy-Performance Trade-Off Multi-Resource Cloud Task Scheduling Algorithm

A) Design of trade-offs algorithm

Let $Normal(x, c_i, p_j)$ be the standard reference value of resource c_i and $Normal(x, c_i)$ be the standard value of resource c_i of task x. To create dissimilar resource dimensionless, this reference value is demarcated as:

$$Normal(x, c_i, p_j) = \frac{c_i(x)}{c_i(p_j)} \qquad (1)$$

Let, $Load_{time}(VM_{tar})$ be the time load for VM_{tar}, $Load(VM_{tar}, c_k)$ be the standard load of resource c_k for VM_{tar} and α be the odds parameter $0 < \alpha < 1$.

Furthermore, with the purpose of saving time deprived of worsening resources, all tasks could be finished at the same time is the preeminent way, on the other hand in the real arranging procedure it is not guaranteed that all the tasks would finish at the same time.

```
Input: CloudletList, VmList, HostList, S(VmList, HostList), c1*, c2*, ..., cm*, α, REFERENCE_TIME , REFERENCE_CPU , Output : Assignment of CloudletList
Init Load(vm_i, c_j) = 0, i = 1, ..., n, j = 1, ..., m;
For each Cloudlet Calculate Priority
Sort CloudletList By Priority
For each cloudlet x in CloudletList
    Random t
    IF t<α
        Calculate Normal(x, c_1), ..., Normal(x, c_j) , find the minimum Normal(x, c_k);
        Find the VM_tar with the minimum Load(VM_tar, c_k);
    ELSE
        Find the VM_tar with the minimum Load_time(VM_tar)
    Assign cloudlet x to VM_tar,
        Load(VM_tar, c_j) = Load(VM_tar, c_j) + Load(x, c_j, VM_tar), j = 1,2, ..., m;
        Load_time(VM_tar) = Load_time(VM_tar) + Load_time(x, VM_tar)
Return Assignment of CloudletList.
```

IV. PROPOSED ENERGY-EFFICIENT VM ALLOCATION POLICY

This strategy demonstrates the allocation of vm to particular host and run the cloudlets on it after the allocation.

Cloudsim simulator provides the VmAllocation Policy Simple () allocation policy. In this policy, host is set up for the Pes elements that are used in lesser amount. In proposed work, these are adapted by configuring RAMRatio and mipsRatio. The steps used are as follows:

1. First find the host that is underused.

2. If (Host is underused)

then Allocate VM

Else

Pause to get underused host

3. The ways to find the Underusedhost are:

A) Free pes criteria.

B) Free RAM criteria.

Suppose the host H_1 that has a lesser amount of pes in use. Thus in this scenario, the usage of RAM is considered by the underutilized Host is H_1 and H_2. Consequently, the host H_2 is underused in this case. Thus the computation of the proportion of RAM of the host and proportion of mips of the host would take place. So, calculate the value for comparisons by using below formula.

$mipsRatio$ = (freemips(H1) / (freemips(H1) + freemips(H2))) * .5

$RAMRatio$ = (freeRAM(H2) / (freeRAM(H2) + freeRAM(H1))) * .5

If (mipsRatio > RAMRatio || H1 == H2)

Then

$VMAllocation$ on host H_1

Else

$VMAllocation$ on host H_2

4. Method to get free RAM and free pes.

Cloudsim simulator defines $getfreeRAM()$ and $getfreePes()$ functions.

5. The VM would be demolished afterward the provision and implementation of cloudlets.

$host.vmDestroy$(vm);

The pictorial rtepresentation of the algorithm is as follows:

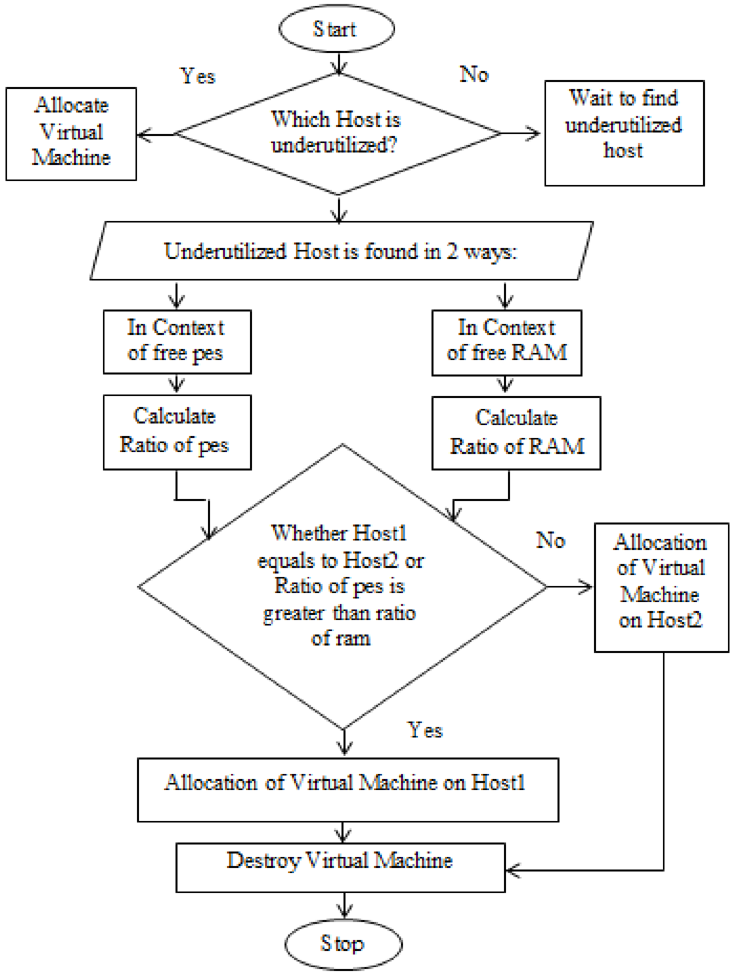

Fig. 1: Flow Chart of Proposed VM Allocation policy

V. EXPERIMENTAL RESULTS

The experimentations are performed using CloudSim 3.03simulator. Xen is considered as the virtual machine monitor (hypervisor).The proposed VM allocation policy algorithm is employed in Java and verified on a Dell workstation with Intel i7 3.07 GHz CPU and 24 GB memory. The models are implemented to a set of varied cloud assets, i.e., VMs along with hosts, and various input provision requirements. The source prerequisite and the dimension of the resource requirements are likewise produced arbitrarily. The properties for the VMs are created arbitrarily and the sources of all the VMs installed on a host are a lesser amount of than the resource measurements of that host. To calculate the effectiveness of our proposed algorithm, we have compared the proposed algorithm with existed system. To estimate the presentation, we have compared the combined values for the energy consumption and Overhead of all VMs, Execution time, and Finish time.

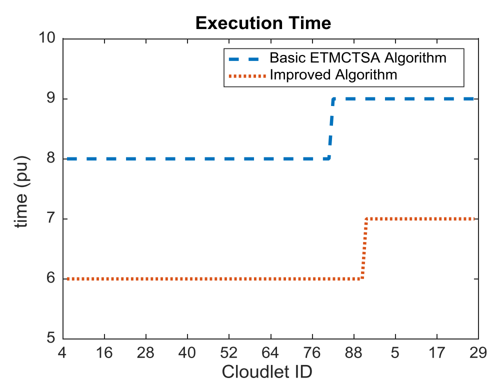

Fig. 2: Comparison of Execution time for Basic ETMCTSA and Improved Algorithm

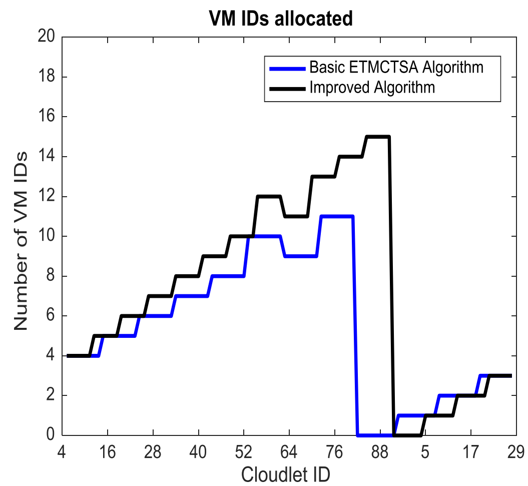

Fig. 3: VM ID allocated to cloudlets for Basic ETMCTSA and Improved Algorithm

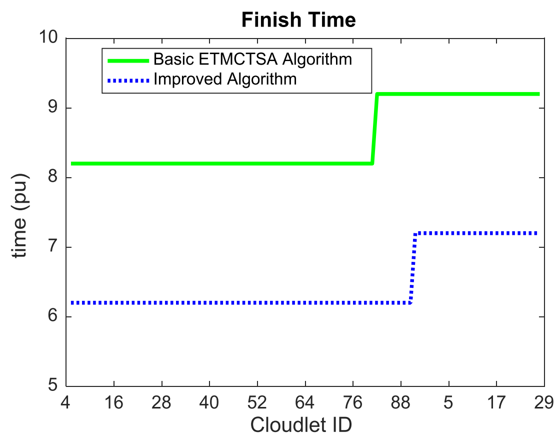

Fig. 4: Comparison of Finish time for Basic ETMCTSA and Improved Algorithm

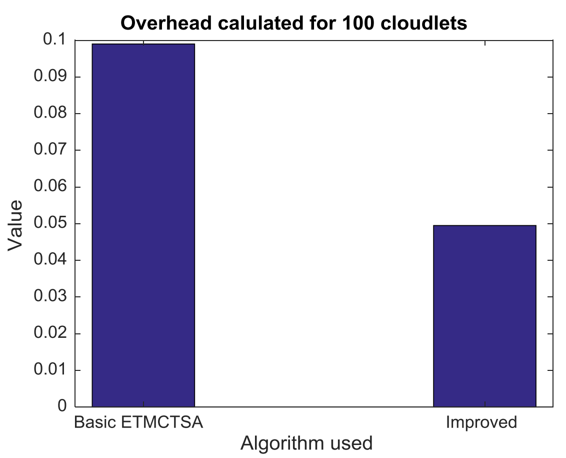

Fig. 5: Comparison of Total Overhead for Basic ETMCTSA and Improved Algorithm

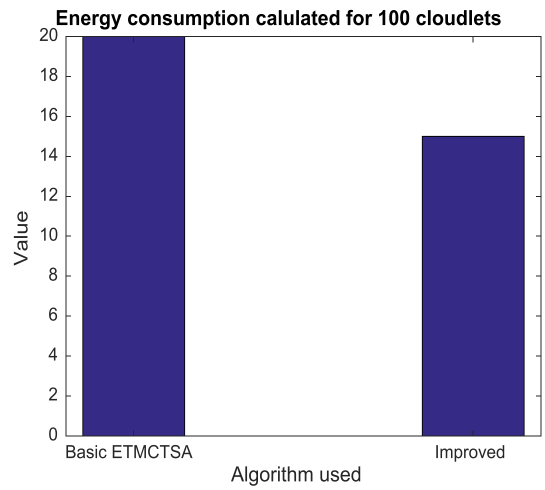

Fig. 6: Comparison of Energy Consumption for Basic ETMCTSA and Improved Algorithm

The outcomes as accessible in Figs. 2–6 prove that proposed algorithm executes better than existing compared algorithm. The proposed algorithm provides improved average energy consumption and less overhead than existed algorithm which ultimately reduces the execution time of cloudlets.

VI. CONCLUSION

Energy consumption has been a major concern in the modern computation system like datacenter and cloud environment that consumes great volumes of energy. The operational cost is directly increases with the increase in energy consumption and affects the surroundings as it increases greenhouse gas emissions. In the current study, the ETMCTSA has been implemented for workflow scheduling in a heterogeneous cloud environment. To minimize the energy consumption and overhead, we have altered the VM allocation policy by using underutilized resources in the context of the ram and mips. The experimental results clearly show that, in the proposed method, the execution time, finish time and the no. of VM utilized are improved because all the underutilized elements are properly handled. Moreover, the aggregate value of energy consumption and total of overhead is less in the proposed work as compared to the existing algorithm. In the future, we would compute further qualitative metrics and study dissimilar operator requests for categorization the virtual machines. For the stage of resource allocation, afore handing over the tasks, we would allocate tasks to a VM as much as probable so as to decrease unallocated space. Future work may take account of the concern of an improved amount of different VMs for cost evaluation.

References

[1] R. Buyya, C. S. Yeo ,S. Venugopal, J. Broberg, I. Brandic, “Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the fifth utility,” Future Gen Comput Syst, Vol. 25, Issue 6, pp. 599–616, 2009

[2] E. Deelman, D. Gannon, M. Shields, I. Taylor, “Workflows and e-science: an overview of workflow system features and capabilities,” Future Gen. Comput. Syst., Vol. 25, Issue 5, pp. 528–540, 2009

[3] M.A. Rodriguez, R. Buyya, “A taxonomy and survey on scheduling algorithms for scientific workflows in IAAS cloud computing environments,” Concurr. Comput. Pract. Exp., 2016

[4] F. Wu, Q. Wu, Y. Tan, “Workflow scheduling in cloud: a survey,” J Supercomput, Vol. 71, Issue 9, pp. 3373–418, 2015

[5] M. Masdari, S. ValiKardan, Z. Shahi, S.I. Azar, “Towards workflow scheduling in cloud computing: a comprehensive analysis,” J Netw Comput Appl Vol., 66, 2016

[6] Y.C. Lee, A. Y. Zomaya, “Energy efficient utilization of resources in cloud computing system,” J. Supercomput. 60 (2) (2012) 268–280

[7] Y. K. Jiang, W. Z. Hui, J. X. Hong, H.E. Qin-Ming, “Power management of virtualized cloud computing platform,” Chinese J. Comput. Vol. 35, Issue 6, pp. 1262–1285, 2012

[8] L. Xin, Q. Zhuzhong, L. Sanglu, W. Jie, “Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center,” Math. Comput. Modelling, Vol. 58, Issue 5, pp. 1222–1235, 2013.

[9] E.G. Coffman, J. Csirik and G.J. Woeginger, "Approximate solutions to bin packing problems," Handbook of applied optimization, pp. 607- 615, 2002

[10] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej, A. Streit, "Load and Thermal-Aware VM Scheduling on the Cloud," International Conference on Algorithms and Architectures for Parallel Processing ICA3PP 2013: Algorithms and Architectures for Parallel Processing pp. 101-114, 2013

[11] J. Moore, J. Chase, P. Ranganathan, R. Sharma, “Making scheduling ‘‘cool’’: Temperature aware workload placement in data centers,” Conference on Usenix Technical Conference, pp. 61–75, 2005

[12] Q. Tang, S. Gupta, G. Varsamopoulos, “Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach,” IEEE Trans. Parallel Distrib. Syst., Vol.19, Issue 11, pp. 1458–1472, 2008

[13] E. K. Lee, I. Kulkarni, D. Pompili, M. Parashar, "Proactive thermal management in green datacenters," The Journal of Supercomputing, Vol. 60, Issue 2, pp. 165–195, 2012

[14] A. Banerjee, T. Mukherjee, G. Varsamopoulos, S. K. S. Gupta, “Cooling-aware and thermal-aware workload placement for green HPC data centers,” Proceedings of Green Computing Conference, pp. 245–256, 2010

[15] A.M. Sampaio, J.G. Barbosa, “Dynamic power-and failure-aware cloud resources allocation for sets of independent tasks,” IEEE International Conference on Cloud Engineering (IC2E), pp. 1–10, 2013

[16] M. Islam, P. Balaji, P. Sadayappan, D. K. Panda, “QoPS: A QoS based scheme for parallel job scheduling,” Job Scheduling Strategies for Parallel Processing, pp. 252–268, 2010.

[17] S. Fu, “Failure-aware resource management for high-availability computing clusters with distributed virtual machines,” J. Parallel Distrib. Comput., Vol. 70, Issue 4, pp. 384–393, 2010

[18] B. Javadi, J. Abawajy, R. Buyya, “Failure-aware resource provisioning for hybrid cloud infrastructure,” J. Parallel Distrib. Comput., Vol. 72, Issue 10, pp. 1318–1331, 2012

[19] L. Mao, Y. Li, G. Peng, X. Xu, W. Lin, “A multi-resource task scheduling algorithm for energy-performance trade-offs in green clouds,” Sustainable Computing: Informatics and Systems, Vol. 19, pp. 233-241, 2018