Design of Energy Aware Scheduling Algorithm for Executing Scientific Workflows in Cloud

Balamurugan S, Saraswathi S

***Abstract:** The usage of cloud computing and its resources for the execution of scientific workflow is a rapidly increasing demand. The Scientific applications are generally large in scale; even a single scientific workflow includes more number of complex tasks. Execution of these tasks can be made successful only by deploying it in the cloud virtual machines, because only cloud environment can only provide very large number of computing assets. In cloud, every processing resource is given as Virtual Machine. Any scientific workflow deployed in the cloud needs large number of virtual machines so; huge amount of computational energy is spent by the virtual machines to execute multifaceted scientific workflows. Hence there arises the need to utilize the cloud resources in an energy efficient way. Also, if the virtual machines are planned to schedule in an energy efficient manner there is an increase of makepsan of the workflow which is going to be an important parameter for completing the workflow within the deadline. So, the need for executing scientific workflows in energy efficient way with reduced makespan becomes a major issue among the researchers. It also becomes very challenging task to executing a scientific workflow in within the given deadline of a task in the given workflow. To address these issues, a new Energy Aware workflow scheduling algorithm is proposed and designed with improved makespan for the execution of different scientific applications in cloud environment.*

***Keywords:** Workflow, Scientific Application, Task Scheduling, Virtual Machines; Power Utilization, Energy Efficiency, Task assignment, Task migration, makespan, Genetic Algorithm, Fitness Function.*

I. INTRODUCTION

Cloud is an on-demand service provider of computing system resources, which includes storage, computing servers, databases, software and networking that are distributed and accessed through internet. Virtualization is the most important concept in cloud computing which shares the physical instance of any cloud asset to multiple users. Virtualization of computing servers can run large number of tasks assigned to it from time to time. The tasks can range from simple to very complex tasks like executing scientific workflows in the virtual machine. A scientific workflow is a Directed acyclic graph which is popularly known as DAG, in which the nodes are called tasks and edges are called dependencies among tasks. The main important property of scientific workflow is that it manages the task dependencies and dataflow of any scientific application.

A single workflow consists of large number of tasks and each task performs a specific function of given scientific application with a final goal.

Every Scientific workflow has two types of deadline viz. overall deadline and sub-deadline. Overall deadline is the time taken to complete all the tasks in a given workflow and sub-deadline is the deadline assigned for a each individual task in a workflow.

The enormous usage of computing servers and data centers for the execution of scientific workflows leads to increase in the amount of energy consumption of virtual machines which becomes major issue needs to be address by the cloud researchers. Hence, utilizing the cloud computing assets in more energy efficient way has become more important issue for the researchers to focus in the field of deploying scientific workflows in cloud. Also, when the workflow scheduling algorithm schedules the task to the virtual machine in energy efficient manner, there is a vast increase in the makepsan of the workflow which is an important parameter for completing the workflow within the deadline. So, reducing the makespan is an important thing for the successful execution of workflow. To address these issues, a new Energy Aware workflow scheduling algorithm is proposed and designed with improved makespan for the execution of different scientific applications in cloud environment.

This paper is organized with the following sections. In Section 2, few important energy efficient scheduling algorithms that are commonly used for the execution of scientific workflows in cloud are discussed in brief. The energy model of cloud computing, equations pertaining to energy consumption of a virtual machines are discussed in Section 3. Section 4 proposes a new Energy Aware Workflow Scheduling algorithm and it is discussed in detail. Section 5, discusses about the simulation and results of the work done and Section 6 concludes the paper with the future extension of the work.

II. RELATED WORKS

This section briefly discusses about most important energy efficient scheduling algorithms for executing scientific workflows in cloud and also the algorithms related to makespan improvisation. Xiaolong Xu et.al proposed a resource allocation algorithm called EnReal [1] to reduce the energy consumption of different kind's workflows. In this algorithm, the cloud virtual machines are used dynamically whenever the need arises for execution of scientific workflow.

Revised Manuscript Received on December 30, 2019.

* Correspondence Author

S. Balamurugan, research scholar in Department of CSE, Pondicherry Engineering College, Puducherry, India.

S. Saraswathi, Professor Department of Information Technology, Pondicherry Engineering College, Pondicherry, India.

The main advantage of this work is that most part of the algorithm works on reducing the energy consumption which incorporate the base level energy consumption of any Physical

Machine, the energy spent by all idle Virtual Machines, the energy spent by all idle physical machines and the data transfer between the processes.

Zhuo Tang et.al. proposed a new workflow scheduling algorithm called DEWTS [2] (DVFS Energy efficient Workflow Task Scheduling). This algorithm executes the workflow task in the cloud with DVFS enabled. The main idea behind this algorithm is to merge the idle virtual machine processors to recover the slack time. After some iteration of merges all the slack times are turned to be a useful one. DEWTS algorithm makes use of HEFT algorithm for task mapping onto the virtual machines. The test simulation result proves that DEWTS has considerably reduced the average power utilization by up to 46.5% for different applications.

Huangke Chen et.al. developed energy efficient online scheduling Algorithm called ENOS [3] for scheduling real time workflows in the cloud. The author has proposed three policies to scale up and scale down the cloud computing resources. These policies can integrated into ENOS to balance weighted square frequencies of hosts.

Khadija Bousselmi et.al. proposed a scheduling scheme called Workflow Partitioning for Energy Minimization (WPEM) [4] and for the execution of scientific workflows the Cat Swarm Optimization (CSO) algorithm is used. The proposed algorithm reduces the overall energy consumption of workflow and also reduces the energy spent in data transfer that occurs in parallel to the execution of tasks. This algorithm makes use of the schedule generated by the CSO algorithm for the created partitions in such a way to set limit for the total energy utilization of the workflow and its execution time. The experimental outcomes demonstrate that their proposed algorithm has decreased the energy utilization with the tested workflows.

In [5] an energy efficient task scheduling algorithm is proposed. This algorithm uses frequency scaling technique to execute the workflow within a given deadline. The frequency of processor is found out by using the ESFS (Stepwise Frequency Scaling) algorithm to execute a task on given virtual machine. The main advantage of this algorithm is that it completes the execution of all the tasks within the given deadline. The algorithm constantly changes the processor frequency for each task in so that the total energy utilization is reduced. The simulation results of this algorithm is compared with HEFT and ESS algorithm and proved that the proposed algorithm comparatively reduces the energy consumption when compared to other two algorithms.

In [6] the author proposed a Pareto-based workflow scheduling algorithm which is multi-targeted. It is the extension of well-known best in class heuristic algorithms that are most capable of computing a set of tradeoffs most constructive solution for energy efficiency and makespan. This approach depends on experimental models which catches the real behavior of energy consumption in heterogeneous parallel framework.

In this section we will discuss about various workflow scheduling algorithm to improve the makespan of the workflow proposed by many researchers.

Ali S. A. Al-Haboobi presented an improved Max-Min algorithm [7]. This algorithm effectively minimizes the makespan of workflow and to increase the resource utilization. The experimental result shows that the proposed algorithm outperforms the existing Max-min algorithm.

Yang Cui, Zhang Xiaoqing proposed a workflow Scheduling optimization algorithm using Genetic Algorithm [8]. The proposed algorithm reduces the execution cost of the workflow under the deadline and budget constraint.

Ahmad M. Manasrah and Hanan Ba Ali proposed a Hybrid GA-PSO [9] algorithm for task allocation to the virtual machines efficiently. The experimental result shows that the proposed algorithm outperforms the existing Genetic Algorithm and PSO algorithm.

Zong-Gan Chen and et al. proposed a Genetic Algorithm approach[10] for a cost-minimization and deadline constrained workflow scheduling model on cloud computing. Also a dynamic objective strategy is proposed to let the Genetic Algorithm to optimize the execution time of the workflow.

Li Liu, Miao Zhang and et al. presented a co evolutionary genetic algorithm [11] to scientific workflow scheduling in cloud. In this approach the crossover and mutation probability to speed up the convergence of the optimal solution.

Israel Casas, Javid Taheri and et al. proposed an enhanced Genetic Algorithm [12] to improve the makespan of the workflow scheduling in cloud. This algorithm reduces the makespan between 11 to 85 per cent when compared to existing algorithms.

III. ENERGY MODEL OF CLOUD

Cloud offers different types of virtual machines viz. vm_1 , vm_2 , vm_3 … vm_k with different computing performance C_1, C_2,...C_n calculated in MIPS. So, the power consumed P by any virtual machine will differ according to the computing performance. This is given by the equation 1 and it implies that if a virtual machine has more computing performance then it consumes more CPU power.

If C_k>C_{k+1}then P_k>P_{k+1}

(1)

The energy consumption of any virtual machine is of two type's dynamic and static energy consumption. Dynamic state energy consumption is denoted as E_{dyn} and idle state energy consumption is denoted as E_{idle} . So, the power consumption of any virtual machine vm_k is calculated by the equation 2.

$$P_k = V_k^2 * f_k \qquad (2)$$

Here, V_k is the voltage level and f_k is the frequency of execution of any virtual machine vm_k . The energy consumption of the virtual machine is defined as the product of power consumption of virtual machine and time. So, the dynamic energy consumption in a given time (t) is given by the equation 3 and the static energy is given by the equation 4.

$$E_{dyn} = P_k * t_e \qquad (3)$$

Where t_e denotes task execution time.

$$E_{idle} = P_k * t_{idle} \qquad (4)$$

Using the equation 3 and equation 4, total energy consumed by a virtual machine is calculated by the equation 5.

$$E\ (vm_k) = E_{dyn} + E_{idle} \qquad (5)$$

The total energy computed in equation 5 is used to calculate the Power Utility (PU) of a virtual machine, which is discussed in next section.

IV. PROPOSED WORK

This section introduces the proposed algorithm called Energy Aware Workflow Scheduling with task migration. This algorithm comprised of two modules: 1) Energy Aware VM Allocation 2) Task Scheduling using modified Genetic algorithm. In the first module, the optimal virtual machine is found for each task using the Power Utility (PU) model. In the second module, the workflow tasks are scheduled in the optimal virtual machines using the modified Genetic Algorithm.

A. Energy Aware VM Allocation

This algorithm is based on the concept called Power Utility(PU) of a virtual machine. Power Utility of the virtual machine can be defined as the ratio of workload to the total energy consumption of task t_k on virtual machine vm_k and it is calculated by the equation 6. All the tasks in the workflow are assigned to avirtual machine based on the Power Utility model.

$$PU\ (t_k) = \frac{W_i}{E\ (vm_k)} \qquad (6)$$

Where W_i denotes the workload of virtual machine.

Once after calculating the Power Utility, a set consisting of Power Utility is formed for each task t_k and it is denoted by PU (t_k) and it is represented by the equation 7. This set holds all the values of power utilities if a particular task is executed in various virtual machines.

$$PU(t_i) = \{pu(t_i^1),\ pu(t_i^2), pu(t_i^3),\ldots\ldots\ldots, pu(t_i^j)\} \qquad (7)$$

The power utilities in the set are arranged in descending order. The first value in the set denotes the ideal power utility factor to execute task t_k in the first virtual machine and the virtual machine selected is made as an optimal virtual machine and it is denoted by vm_{opt}.

Figure 1 illustrates the diagrammatic representation of Task allocation model. This model consists of two main modules called Power Utilization set module and Task assignment module. The Power Utilization Set Module takes two inputs, viz. Set consisting of list of virtual machines and a task set. The detailed functionality of the two modules is discussed below.

B. Task Allocation Scheme

After completing the formation of Power Utility set, the task allocation is done using the following steps

Step 1: If a task t_i can be finished within the given deadline, then assign task t_i to the optimal virtual machinevm_{opt}. This means task t_i is allocated to vm_{opt}.

Step 2: If the selected virtual machine is already allocated to some other task t_i. Form a new task set called sub-optimal task set and add the unallocated task into it.

Step 3: Take task from the sub-optimal set of tasks one by one, find the next ideal power utility and map the task new optimal virtual machine vm_{opt}.

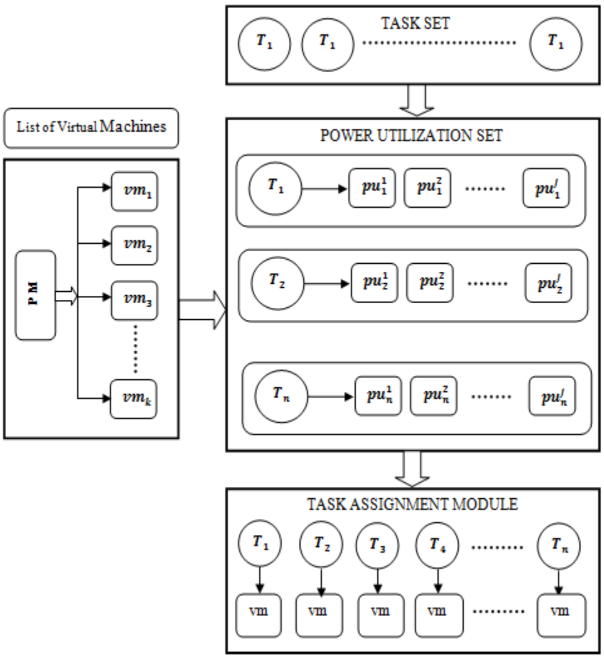

Fig.1. Task Allocation Model

C. Algorithm

The workflows of various scientific applications are maintained as a DAX file. DAX file is the description of workflow in conceptual way, all the information related to the tasks and dependencies among the tasks are represented in XML format. At first the DAX file is parsed using the XML parser to retrieve the task list, overall deadline and sub-deadline of tasks. The algorithm takes as input the DAX file, Virtual machine list VM_LIST and set of all tasks TASK_SET as generated by DAX parser.

Algorithm: Energy Aware Workflow Scheduling Algorithm

Inputs: 1. DAX File
2. VM_LIST

```
String daxFile = "<…/<DAXFile.xml>"
int k = <no. of Virtual Machines>
vmlist0 = createVM(k) // Create the list of virtual machines
VM_LIST = {vmlist0}
TASK_SET = daxParser (daxFile)
int noOfTasks = getSize(TASK_SET) // get the no. of tasks in the task set
for each in the TASK_SET do
    PU ($t_i^k$) = $W_i$ / E ($vm_k$)
    //Arrive power Utility Set for each task
    for each in theTASK_SET do
      $PU(t_i) = \{pu(t_i^1),\ pu(t_i^2), pu(t_i^3), \ldots\ldots\ldots , pu(t_i^j)\}$
    end for
    for each element in $PU(t_k)$
      SORT $(PU(t_i))$
    end for
      for each task in the TASK_SET do
        for each element in $PU(t_i)$
          if (free(pu($t_i^j$))
            $vm_{opt} \leftarrow pu(t_i^j)$
            $vm_{opt} \leftarrow t_i$
           end if
         end for
       end for
    end for
for each in the TASK_SET do
    if (Deadline $(t_{i+n})$ < Deadline $(t_i)$  &&
                                E $(t_{i,i+n}^k) \leq$ E $(t_i^j)$ + E $(t_{i+n}^{j+m})$)
         $t_k \leftarrow$ merge $(t_i , t_{i+n})$
         $vm_k \leftarrow t_k$
         Update $(t_k , vm_k)$
      end if
end for
wfEngine.submitJob(VM_LIST, noOfTasks, TASK_SET,
PU);
```

D. Task Scheduling using modified Genetic Algorithm

The main purpose of selecting Genetic Algorithm for this kind of scheduling problem when compared to other evolutionary algorithms has the following reasons. Genetic Algorithm Have Crossover and mutation but in algorithms like PSO each particle adjusts its flying according to its own experience and others experience. GA will have an individual objective function and it will not compare its value with other individuals. In PSO there is no creation and deletion of individuals. But in GA we can create new individual. In PSO there is a more chance of local minima or local maxima preferred by the particles. GA will guarantee global optimal solution. Moreover GA is the well suited technique for workflow scheduling as it creates new individual with improved fitness values.

The Cloud provider should make sure the optimal scheduling of workflow tasks in cloud to assure with improved throughput and correct use of resources. Above all the most important metric to be taken care by the cloud provider is to execute the workflow tasks in the cloud resources with minimum makespan. In this paper, we propose a variant of genetic algorithm to optimize the workflow flow schedule to improve the makespan of the given scientific workflow. This section maps the conventional Genetic Algorithm to the workflow scheduling problem and discusses the steps involved in it.

E. Initial Population

The initial population is the set of all virtual machines that is to be used to execute the given workflow. Each virtual machine with its allocated tasks is treated as a single gene. Likewise 'n' number of genes will be present in a given population. Every individual in population is represented as a chromosome to make suitable for genetic operations. The individuals are selected from the initial population and operations like selection and crossover are applied on the selected individuals to form the next generation. The figure below gives the pictorial representation of genes in the population is shown in the figure 2. Each Virtual Machine and the tasks executed it in are encoded in binary bit. Example VM3: T4, T8, T9 (0110 -> 0100, 1000, 1001)

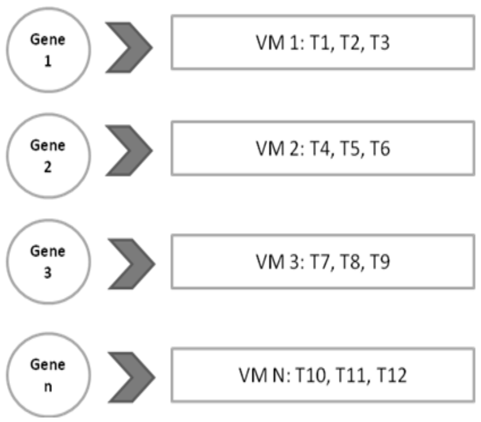

Fig. 2. Representation of Genes

F. Fitness Function

A Fitness function is a type of objective function to check whether an optimal solution of a given problem is attained or not. It is also used to measure of the dominance of an individual with the other individuals in the population. Also, the performance of any individual in the population is compared with other individuals using Fitness function. Therefore, the individuals will keep on survive to next generation or die in the current generation is decided based on the fitness or function value. The main objective of using genetic algorithm for workflow scheduling is to minimize the makespan of workflow. The fitness function f(n) is as given in equation 8.

$$f(n) = \text{makespan} = \max\ (\ CT_{max}\ (VM_j)) \quad (8)$$

Where, $j \in VM$, $j = 1, 2, 3, \ldots..,m$

f(n) is the fitness value of nth iteration of the algorithm. The completion time of any task in a virtual machine is calculated by the equation 9.

$$CT_{max} = C_i/PS_j \quad (9)$$

Where, C_i is the computation complexity of task T_i and PS_j is the processing speed of the virtual machine VM_j. CT_{max} is computed for the all the chromosomes and the maximum value of all the CT_{max} gives the makespan.

G. Selection

Selection is the mechanism where the individual genomes for selected from the population for further genetic operations like crossover or mutation.

The selection process is based on the fitness function evaluated for each individual. The selection operation is very important guiding factor of Genetic Algorithm for its performance.

Retrieval Number: A2013109119/2019©BEIESP
DOI: 10.35940/ijeat.A2013.129219
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In Genetic Algorithm not all the generated chromosomes are passed to next iteration. Instead, the chromosomes evaluated using the tournament selection function; the winning chromosomes are selected as best chromosomes for next iteration. The main reason behind using tournament selection method is to avoid larger population size. Two individuals are chosen at random from the population, these two individuals are selected as a parent.

A random number is chosen if r > k, where k is parameter based on CT$_{max}$ value. The remaining individuals which are not selected are returned to the original population because there might be chance for those getting selected in next iteration. Figure 3 shows the two chromosomes which are selected for the crossover operation.

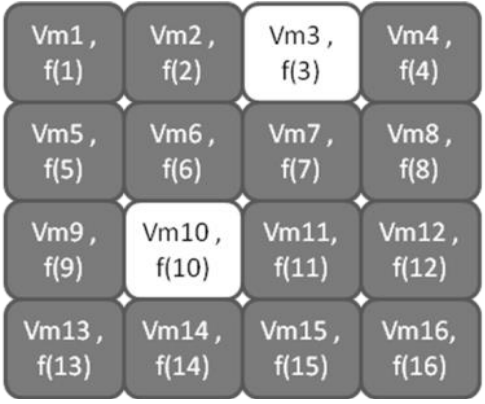

Fig. 3. Selected Chromosomes for Crossover

H. Crossover

In the proposed algorithm, a modified approach of crossover has been used when compared to the original crossover. Figure 4 and Figure 5 shows the parent chromosomes and off-spring chromosomes respectively. The crossover point is chosen randomly and the bytes after the crossover points are swapped. After the crossover process two new chromosomes are generated called offspring's.

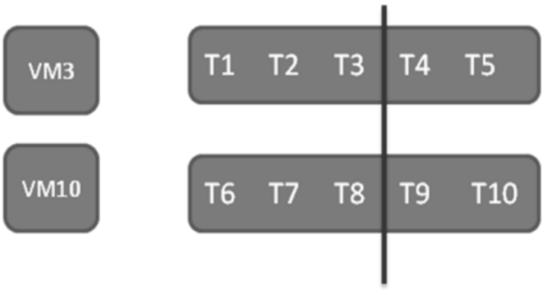

Fig. 4. Parent chromosomes during crossover

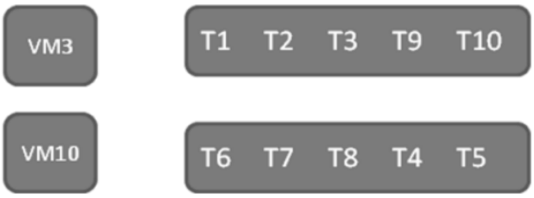

Fig. 5. Produced Offspring's

I. Algorithm

Initialization and Selection

Algorithm: Initialization, Selection

Input: Virtual Machine and task mapping matrix.
Output: Sbest

```
init_population = randomize (vm_task[][]) //initialize
population
set tSize = n // tournament size
set tm[] = 0; // tournament array
for i = 0 to tSize
    id = Math.random() * chromosome.size()
    tm[i] = get_chromosome(id)
end for
for i = 0 to n
chromosome.fitness[i] =  tm[i].getFitness()
end for
return fitness chromosomes
```

The virtual machines with the corresponding task(s) are considered as an initial population. The fitness value for all the chromosomes in the initial population is calculated first. Now, using the fitness value two chromosomes are selected as the parent chromosomes that need to be given as an input for the crossover function.

Crossover

Algorithm: Crossover

Input: two chromosomes
Output: offspring chromosomes

```
// Choose the Crossover point
p = Math.random(chromosome.length)
for i = 0 to p
   // make the copy of parent1
offspring_chromosome1 = chromosome1
   // swap the bytes after crossover point to child1
offspring_chromosome1[i] = chromosome2[i]
end for
for i = 0 to p
   // make the copy of parent2
offspring_chromosome2 = chromosome2
   // swap the bytes after crossover point to child2
offspring_chromosome2[i] = chromosome2[i]
end for
addToPopulation(offspring_chromosome1,
offspring_chromosome2)
```

The crossover function gets the parent chromosomes as input, computes the chromosome length and randomly chooses the crossover point. After choosing the crossover point it will swap the bytes after the crossover point to from child 1 and child 2. After the creation of children chromosome, add both the parent as well as the children chromosomes to the population and make it as new population.

Normally, the parent chromosomes will be discarded from the population after the creation of off-springs. But, in the workflow scheduling, the parent chromosomes are again added to the population along with its off-spring chromosomes. This is because in certain times the children chromosome in the new population will not give better fitness value when compared to its parents.

This scenario is applicable only to workflow scheduling kind of problems.

J. Variations in the original genetic algorithm

The proposed Genetic Algorithm for the workflow scheduling has introduced a set of modifications which are more beneficial obtaining optimal solution faster for the problem of workflow scheduling in the cloud.
The tournament selection process is used instead of roulette wheel. The chromosomes which are not chosen in the tournament selection method are added to the new population because they might get selected in the next selection process. The proposed crossover produces four children and the sub populations are added into old populations.

V. SIMULATION, RESULTS AND DISCUSSION

The effectiveness of the proposed work is proved by simulating the Energy Aware Workflow Scheduling Algorithm using the well-known simulator called WorkFlowSim. WorkFlowSim is the extension of existing cloudSim simulator which provides higher layer workflow management. The simulator supports all the packages and corresponding DAX files to simulate different kinds of scientific workflows. A new package for the proposed algorithm has been created and uses the existing scientific workflows like Montage, LIGO, SIPHT, Cybershake and Epigenomics to prove the results. The simulation results are plotted as presented in the Figure 6 and Figure 7.

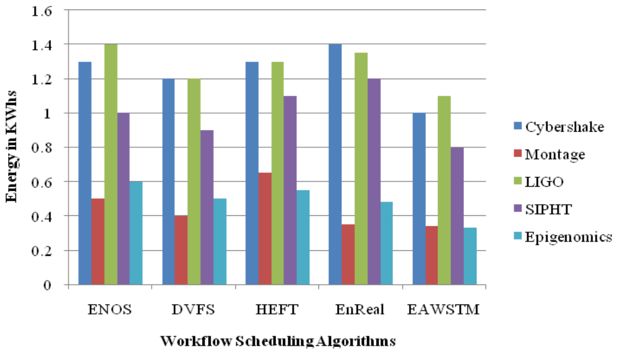

Fig. 6. Comparison of Energy consumption of various workflows

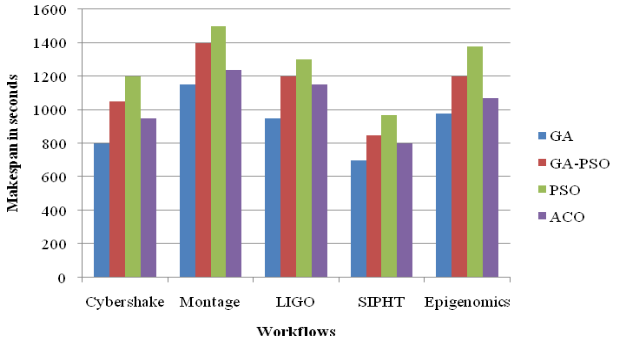

Fig. 7. Comparison of makespan of workflows

The energy consumption of the proposed work for various workflows is compared with the existing well known algorithms and graph is plotted as shown in figure 6. The graph shows that proposed algorithm perform better in terms of energy consumption when compared to other well known algorithms. Similarly the makespan of various workflows is compared with the existing algorithms as shown in figure 7. From the graph it is proved that the proposed algorithm has lesser makespan when compared to other competitive algorithms. Further we are focusing the research work to reduce the energy consumption further by combining the load-balancing factors.

VI. CONCLUSIONS

This paper focuses the issues of higher energy consumption by the cloud virtual machines during the execution of different type's scientific workflow application. The prevalent algorithms related to energy efficient task scheduling in the literature are studied to develop a better scheduling algorithm named Energy Aware Workflow Scheduling using Modified Genetic Algorithm. This algorithm consists of two modules: 1) Energy Aware VM Allocation 2) Task Scheduling using modified Genetic Algorithm. In the first module, the optimal virtual machine is found for each task using the Power Utility (PU) model. The second module is used to optimize the virtual machine allocation to reduce the makespan of the workflow. The proposed algorithm is simulated using WorkFlowSim and results proved that the proposed algorithm perform well in terms of total energy consumption and makespan of workflow.

REFERENCES

1. Xiaolong Xu, Wanchun Dou, Xuyun Zhang, and Jinjun Chen.: EnReal: An Energy-Aware Resource Allocation Method for Scientific Workflow Executions in Cloud Environment. In: IEEE Transactions on Cloud Computing, Vol.4, No.2, pp.166-179, April-June (2016).
2. Zhuo Tang, Ling Qi, Zhenzhen Cheng, KenliLi , Samee U. Khan, Keqin Li.: An Energy-Efficient Task Scheduling Algorithm in DVFS-enabled Cloud Environment. In: Journal of Grid Computing, Springer Science, DOI 10.1007/s10723-015-9334-y, March (2015).
3. Huangke Chen, Xiaomin Zhu, DishanQiu, HuiGuo, Laurence T. Yang, peizhonglu.: EONS: Minimizing Energy Consumption for Executing Real-Time Workflows in Virtualized Cloud Data Centers. In: IEEE International Conference on Parallel Processing Workshops, DOI 10.1109/ICPPW.2016.60, (2016).
4. Khadija Bousselmi, Zaki Brahmi, Mohamed Mohsen Gammoudi.: Energy efficient partitioning and scheduling approach for Scientific Workflows in the Cloud. In: IEEE International Conference on Services Computing, DOI 10.1109/SCC.2016.26, (2016).
5. Pietri, Ilia, and RizosSakellariou.: Energy-aware workflowscheduling using frequency scaling. In: Parallel Processing Workshops (ICCPW), 43rd International Conference, pp. 104-113. IEEE Press, (2014).
6. Juan J. Durillo, Vlad Nae, RaduProdan.: Multi-objective energy-efficient workflow scheduling using list-based heuristics. In: Future Generation computer systems, Vol.36, pp.221-236, (2014).
7. Ali S. A. Al-Haboobi , "Improving Max-Min scheduling Algorithm for Reducing the Makespan of Workflow Execution in the Cloud ," in International Journal of Computer Applications (0975 – 8887) Volume 177 – No.3, November 2017.
8. Yang Cui and Zhang Xiaoqing, "Workflow Task Scheduling Optimization Based on Genetic Algorithm in Cloud," 3rd IEEE International Conference on Cloud computing and Big Data Analysis, 978-1-5386-4301-3/18/$31.00, 2018.
9. Ahmad M. Manashrah and Hanan Ba Ali, "Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing," in Journal of Wireless Communication and Mobile Computing, Wiley , https://doi.org/10.1155/2018/1934784 2018.
10. Zong-Gan Chen, Ke-ling Du, Zhi-Hui Zhan and lun Zhang, "Deadline Constrained Cloud Computing Resources Scheduling for Cost Optimization Based on Dynamic Objective Genetic Algorithm" IEEE Transaction, 978-1-4799-7492-4115/$31.00, 2015.
11. Li Liu, Miao Zhang, RajkumarBuyya and Qi Fan, "Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing" Concurrency Computat.: Pract. Exper. 2016, pp. 1-12.

12. Israel Casas, Javid Taheri, Rajiv Ranjan, Lizhe Wang, Albert Y.Zomaya, “GA-ETI: An Enhanced Genetic Algorithm for the Scheduling of Scientific Workflows in Cloud Environments”,Journal of Computational Science http://dx.doi.org/10.1016/j.jocs.2016.08.007

AUTHORS PROFILE

S. Balamurugan, is a research scholar in Department of CSE, Pondicherry Engineering College, Puducherry, India. He is currently working as Assistant Professor in the Department of Information Technology, Perunthalaivar Kamarajar Institute of Engineering and Technology (PKIET), Karaikal, Union Territory of Puducherry, India. His special research interests include scheduling algorithms for scientific workflows in cloud environment.

S. Saraswathi, is currently a Professor at the Department of Information Technology, Pondicherry Engineering College, Pondicherry. She completed her B.Tech in Computer Science and Engineering from Pondicherry Engineering College and her M.Tech. in Computer Science and Engineering from Pondicherry University. She completed her Ph.D. in Speech Processing from Anna University, Chennai, in the year 2008. She has over 100 papers in national and international journals and conferences. Her special research interests include design of intelligent systems, language technology and speech processing.