Scheduling Schemes for Optimizing Energy and Cost in Cloud Computing

Kiran B.R.[1] and Sarojadevi H.[2]

[1,2]*Nitte Meenakshi Institute of Technology, Bangalore*

Abstract -*Subscription based service is fundamental to Cloud Computing where we get guaranteed networked storage space and computer resources. Computing power is provided to consumer in fully or para-virtualized manner i.e., in the form of leases. The major objective of cloud computing is to provide a reliable service maintaining the energy and cost as two important factors pertaining to QOS. In this paper we analyze the methods that optimize energy and cost in the cloud computing environment. The energy efficient scheme (EESS) focuses* on distributing maximum load on minimum number of *virtual machines. Cost based scheduling (GA) with Genetic algorithm focuses on reducing the execution time, which in turn reduces the user cost by allocating suitable resources to the requested tasks. Cloud environment is created using CloudSim simulator. We found that EESS scheme is more energy efficient compared to other schemes and GA has less execution time which reduces cloud usage cost.*

Key Words: Cloud, Cost, Energy, Scheduling, CloudSim.

1. INTRODUCTION

Cloud computing is primarily meant to support virtualization, scalable resource utilization and provide services such as IaaS, SaaS, and PaaS. While Cloud computing provides many advanced features, it still has some shortcomings such as the relatively high operating cost for both public and private Clouds. In the emerging era of Green computing that emphasizes working with limited energy resources and an ever rising demand for more computational power, optimized scheduling techniques for low cost and low energy are going to be increasingly in demand. At present it is estimated that data centers consume 0.5 percent of the world's total electricity usage and it is projected to increase four times by 2020.

However if power consumption continues to increase, power cost can easily overtake hardware cost by a large margin. This situation calls for a major step for carrying out inventions to reduce the energy consumption and cost of computations.

Cloud Computing must ensure that fairness is provided among the requests that are accessing cloud resources since service requests are dynamic in nature. Load balancing is another important factor which significantly affects the overall execution time which in turn affects the cloud usage cost. Hence the scheduler must allocate the right resources to the requests (workload) that significantly reduces execution time of a request and thereby reduces cloud usage cost also.

A scheduler to be developed for cloud deployment is required to schedule the leases in such a way that it conserves more energy and also reduces the cloud usage cost by achieving higher load balancing. As an approach towards this goal, in this paper we have chosen two schemes known as Energy Efficient Scheduling Scheme (EESS) and Genetic Algorithm (GA) which are implemented and analyzed in the cloud environment created using CloudSim simulator. The energy consumption and the execution time is measured, finally the results are analyzed and compared with other scheduling algorithms such as Round-robin (RR), and First-come First-serve (FCFS). Joule meter is used for measuring the energy consumption of the virtual machines. The results show that the EESS scheme conserves more energy than the other scheduling schemes and GA reduces the execution time thereby reducing the cloud usage cost.

2. RELATED WORK

In this section we discuss about the various scheduling schemes that try to reduce the energy consumption of the virtual machines, optimizing the workload distribution, and reducing the overall cost of the cloud usage.

Jiandun Li et al. [1] introduced a hybrid energy efficient scheduling scheme. This approach reduces the incoming request response time by optimally scheduling the workload especially in case of nodes running in low power mode. The scheduling algorithm is based on pre power technique and Least Load First Algorithm. The paper shows that above approach save more energy and optimizes load balancing.

Kejiang Ye et al. [2] presented energy efficient data center architecture for cloud computing, which provides a key technology to consolidate the operation of the server that reduces the overall energy consumption. Server Consolidation achieves energy efficiency by enabling multiple instantiations of operating systems to run simultaneously on a single physical machine. The other technology used is the live migration of VM requests and thereby minimizing the tradeoff between performance and Energy efficiency attaining higher energy saving goal.

AmanKansal et al. [3] proposed methods for power metering and provisioning architecture based on virtual machine in which Joulemeter is used for measuring the power of virtual machines per second in watt in cloud environment.

Bo Li, Jianxin Li et al. [4] came up with algorithm for energy efficiency that aims in distributing maximum workload onto minimum number of virtual machines using Live Application Placement approach. The Live Application Placement approach is abstracted as Bin Packing problem for adjusting the virtual machine resources. Over-provision approach is presented to deal with the varying resource demands of applications in cloud. The experimental results show that the above approach is feasible.

R.Vijindra [5] introduced a ranking algorithm based scheduling framework for reducing energy in cloud computing. It has following three main objectives - 1.reducing completion time of jobs 2.reducing overall energy consumption of datacenter 3.balancing the incoming workloads. The ranking algorithm assigns a rank to each VM's based on the available resources and allows the scheduler to assign the task to suitable VM thereby reducing execution time. Migration and load balancing technique is used to schedule the task to VM's running on lower power mode. The simulation results show that it can conserve more energy as well as it can reduce the total execution time.

Amandeep Verma et al. [6] proposed workflow scheduling with cost time optimization based on deadline and budget. A workflow schedule is developed such that it minimizes the execution cost and works to the time constraints of the user.

Gunho Leey et al. [7] introduced "Heterogeneity- Aware Resource Allocation and Scheduling in the Cloud", in which a new fairness metric is defined to facilitate sharing of the jobs in cloud environment with heterogeneity in the incoming requests.

Ke Liu et al. [8] proposed "A Compromised-Time- Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform" that incorporates characteristics of cloud computing to accommodate instance intensive cost
constrained workflows by compromising execution time and cost.
P.K. Srinivasan [9] presented "Time-Cost Scheduling Algorithm", a novel dispensation time cost scheduling algorithm which considers the characteristics of cloud computing to accommodate order-intensive cost-constrained workflowsby compromising execution time and cost with user input enabled on the fly.
Ruben Van den Bossche et al. [10] presented "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads" that proposes a design of a software architecture model for the HICCAM project in order to highlight and emphasize the purpose of the Optimization Engine component.

3. IMPLEMENTATION

3.1 Basic Architecture of Cloud Computing

The basic architecture of cloud computing for the experimentation used in this proposed work is shown in figure 1. The cloud setup is done using CloudSim simulator with 100 virtual machines (nodes) and a web interface where a user can access the virtual machine from the web console and submit their requests in the form of leases. The virtualization allows multiple operating systems to run on a single machine. The first node acts as hypervisor scheduler that schedules the leases (requests) to virtual machines. Apart from that, the hypervisor also provides facilities such as Clone, Pause, Resume, and Migration.

The scheduling code is placed in host1 that acts as a master node that accepts the lease requests and schedules to other hosts for execution.

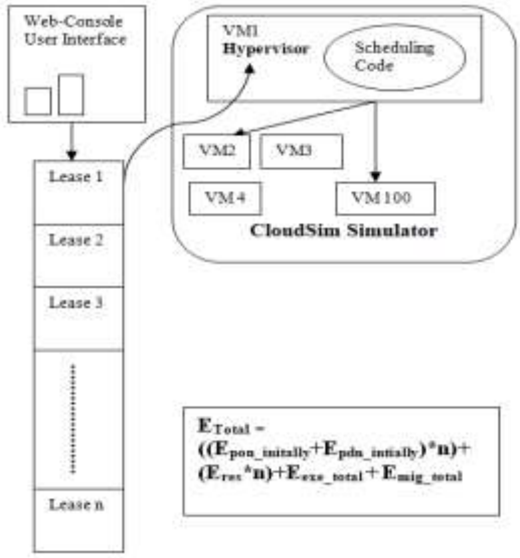

Figure1.Cloud Architecture

3.2 Energy-Efficient Scheduling Scheme

Energy-Efficient Scheduling Scheme works by accepting the requests initially. If no. of requests is zero the process got terminated else it will check for the type of request. If the requested type is PAAS a separate virtual machine is created by using the clone concept. If the requested type is SAAS then the scheduler checks the status of the virtual machines. If the VM's running are sufficient to handle the task then no new VM's are started. If it is not sufficient the scheduler checks for the VM's that are in pause state and brought it to resume and allocates the request. If VM's are switched off it will be powered up and allocates the request. If the machine executing the task goes down, migration is applied and allocated to other machine.

The total energy consumed is calculated using below equation:

ETotal=((Epon_initally+Epdn_intially)*n)+
(Eres*n)+Eexe_total+ Emig_total

Where ETotal is the total energy consumed by all the VMs for 60s, Epon is energy consumed during VM's powering on**,** Epdn is energy consumed during VM's power off , **Eres** is energy consumed to brought VM's to resume state from pause state, Emig is the energy consumed during migration of VM's.

The algorithm for scheduling the incoming tasks to the virtual machines is shown in figure 2. The algorithm is iterative and allows to run the steps repeatedly for all the virtual machines.

3.3 Genetic Algorithm

(1). Initial Population

The set of solutions that are possible is taken as Initial population. These solutions are considered as chromosomes where solutions are nothing but the resources which are allocating to the arrived task. The chromosomes in the initial population are generated randomly using the symbol. These terms are chosen to solve the particular problem.

(2). Fitness function

It evaluates how the selected function meets the requirements of the problem such as resource allotment. GA evaluates each chromosome by fitness function. Fitness function is used for the measurement of effectiveness of the solution according to the given objective.

F=min {max {ck} +∑f(di)},

Where **ck**=chromosomes **di**=initial population

The Fitness function ensures that right resources are allocated to the arrived task so that execution time reduces which in turn reduces the cloud usage cost.

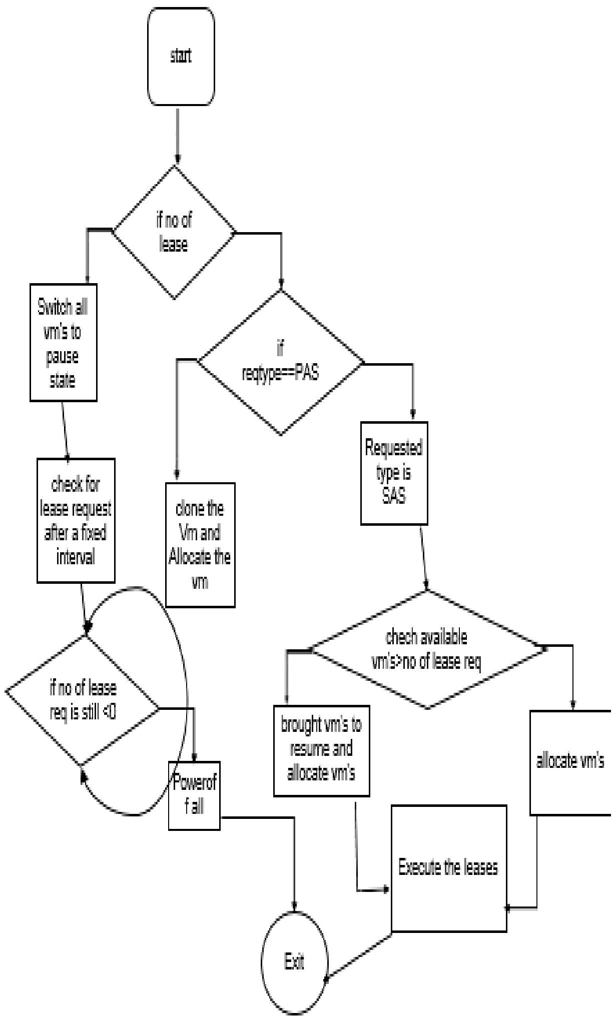

Figure2. Scheduling incoming Tasks to VM

(3). Crossover

It is used for combining two chromosomes to produce next generation chromosomes. It is used for bringing new chromosomes by the mixture of parent chromosomes. Single point crossover is used as only one crossover point is present. In this single crossover point, at the locus, swapping the remaining alleles from parents to others takes place. The operation is performed to select the chromosome.

(4). Mutation

Mutation performs the permutation of existing chromosomes. It provides new gene values added to the gene pool. Mutation provides small alterations at each individual. It is used for finding new points in search space hence the population variation is maintained.

(5). Termination

The selection process is able to copy the chromosome that is having the maximum fit in the given population. This helps to obtain the highest fit value. The termination depends on the size of the chromosomes that has been sorted.

Cloud usage cost is calculated as shown below:

Let et_i denote the time to process workload of task v_i at a node n_j

Therefore, the computation cost of a task v_i at Cloud node n_j can be estimated as follows.

$$Pc^j_i = et^j_i * rc_j$$

The selection process is able to copy the chromosome that is having the maximum fit in the given population. This helps to obtain the highest fit value. The termination depends on the size of the chromosomes that has been sorted.

Cloud usage cost is calculated as shown below:

Let et_j denote the time to process workload of task v_i at a node n_j:

$$et_{j_i} = w_i + p_j$$

Therefore the computation cost of a task v_i at Cloud node n_j can be estimated as follows

$$Pc^j_i = et^j_i * rc_j$$

Where et^j_i is the total execution time of task **i** in node c_j and **rcj** is the node resource usage cost

Hence the total cloud usage cost is computed as shown below:

$$Cost(i,j) = Pc^j_i + tc^j_i$$

Where, Pc^j_i is computation cost tc^j is communication cost.

4. Experimental Setup and Results

The experiment is conducted on a CloudSim simulator that consists of 100 virtual machines (nodes) which consists of same resources. Hosts are connected by means of virtual LAN connection. Host1 is selected as a scheduler node that schedules the incoming requests. The total execution time and total energy consumed is calculated by using the API provided by the Joulemeter. The EESS and the GA can be extended to any number of virtual machines. The execution time and the energy are calculated by using the API provided by the Cloudsim simulator.

The Round Robin (RR) and FCFS algorithms are chosen for comparison since both the algorithms are used to Schedule the task in the cloud environment in which no criteria's need to be met. RR ensures zero starvation whereas FCFS results in a longer starvation in case of longer running task. Thus by considering execution time as a criteria RR yields better results than FCFS, whereas FCFS has low energy consumption.

Workload

The workload chosen to deploy and run in cloud environment is the Linear Search Program and Linear Program that is used to create the parabola and other geometric figures by passing the values to the program. The cloud users pass the values to the above programs from the web interface and the results computed are given back to the users through the web interface only.

4.1 Comparison and Analysis

4.1.1. Based on the total energy consumed(E)

The experimental results show that the Round- Robin does not save any energy since it allocates each task to single virtual machine thereby making multiple virtual machines to run.

EESS scheme greatly reduces the overall energy consumption by using the pause, clone, resume, migration concepts rather than Power The Off and Power On of VM's every time.FCFS also conserves energy but it becomes worst if the loud environment is elaborated. The overall energy consumption for the 100 VM's is calculated by running the tasks for 60 seconds in the cloud environment and the energy consumed is measured using the Joulemeter tool. The graphs in figure3 show how much energy is consumed by each of the algorithms for running the same task in the cloud.

4.1.2. Based on the total execution time(T)

In our cloud architecture the MIPS rating available is 500. Hence based on that MIPS the cost has been calculated by using the above formulae, which is as shown in the table 2.

Table1: Services available for tasks

Service ID	MIPS rating	Processing Time(Sec)	Cost ($)
1	500	20	6
2	1000	10	12
3	2000	5	18

Table 2: Cost Calculation

Workflows	Execution cost	Communication cost	Total cloud usage cost
Workflow 1	18	5	23
Workflow 2	12	3	15

The experimental result shows that the FCFS algorithm reduces the execution time only in case of small number of requests. If the numbers of requests are more Round-Robin is suitable but fails for large scale computing.

The Genetic Algorithm overcomes the tradeoff by using the fitness function that allocates the right resources to the task thereby reducing the overall execution time as shown in fig 3. The total execution time is as shown in fig. 4.

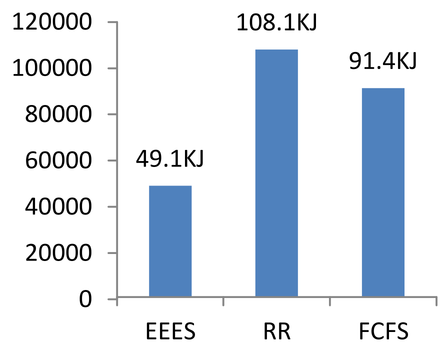

Figure 3. Energy Consumption

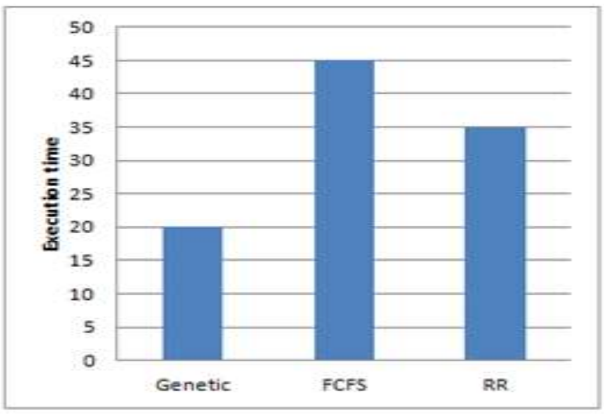

Figure 4. Execution time

5. Conclusion

As the workload dynamically varies in cloud computing, the scheduling of workload is more challenging that results in minimization of energy as well as the execution time that in turn reduces cloud usage cost. Hence in this paper we have chosen two advanced scheduling algorithms to analyze the results in the same cloud environment. The EESS is related to energy optimization and the GA is related to time optimization. The comparison shows that both these algorithms are efficient in their own way. Combining both can offer better result that can overcome the tradeoff between energy and the execution time.

REFERENCES

[1]. Jiandun Li, Junjie Peng, and Wu Zhang, "An Energy efficient Scheduling Approach Based on Private Clouds", Journal of Information & Computational Science, Vol. 8(4), 716-724, 2011.

[2]. Kejiang Ye, Dawei Huang, Xiaohong Jiang, Huajun Chen, Shuang Wu 2010),"Virtual Machine Based Energy Efficient Data Center Architecture for Cloud Computing: A Performance Perspective", IEEE 2010.

[3]. Amankansal, Feng Zhao, Jie Liu, Nupur Kothari, Arka A. Bhattacharya, "Virtual machine power metering and provisioning", ACM 2010.

[4]. Bo Li, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong, "Ena Cloud: An Energy-saving Application Live Placement Approach for Cloud Computing Environments", IEEE International Conference on Cloud Computing 2009.

[5].R.Vijindra, "Energy Efficient Scheduling Framework for Cloud Computing using Ranking Algorithm", Vol.3(10), IJSER, 2012

[6]. Amandeep Verma, Sakshi Kaushal, "Deadline and Budget Distribution based Cost-Time Optimization Workflow Scheduling Algorithm for Cloud", iRAFIT 2012.

[7]. Gunho Lee, Byung Gon Chunz, Randy H.Katz, "Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud", HotCloud, 2011

[8]. Ke Liu, Hai Jin, Jinjun Chen, Xiao Liu, Dong Yuan and Yun Yang, "A Compromised Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform", The International Journal of High Performance Computing Applications, 2010

[9]. P. K. Srinivasan, "Time-Cost Scheduling Algorithm", ICCCE 2012.

[10]. Ruben Van den Bossche, Kurt Vanmechelen and Jan Broeckhove, "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads", IEEE 2010.

[11]. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software - Practice and experience, 2011.