An energy optimization with improved QOS approach for adaptive cloud resources

Danthuluri Sudha[1], Sanjay Chitnis[2]
[1]CMR Institute of Technology, Visvesvaraya Technological University (VTU), India
[2]Dayanandasagar University, India

Article Info

Article history:

Received Apr 13, 2019
Revised Mar 6, 2020
Accepted Mar 18, 2020

Keywords:

ACRR,
Cloud computing,
DVFS,
Energy consumption,
Resource allocation

ABSTRACT

In recent times, the utilization of cloud computing VMs is extremely enhanced in our day-to-day life due to the ample utilization of digital applications, network appliances, portable gadgets, and information devices etc. In this cloud computing VMs numerous different schemes can be implemented like multimedia-signal-processing-methods. Thus, efficient performance of these cloud-computing VMs becomes an obligatory constraint, precisely for these multimedia-signal-processing-methods. However, large amount of energy consumption and reduction in efficiency of these cloud-computing VMs are the key issues faced by different cloud computing organizations. Therefore, here, we have introduced a dynamic voltage and frequency scaling (DVFS) based adaptive cloud resource re-configurability ($ACRR$) technique for cloud computing devices, which efficiently reduces energy consumption, as well as perform operations in very less time. We have demonstrated an efficient resource allocation and utilization technique to optimize by reducing different costs of the model. We have also demonstrated efficient energy optimization techniques by reducing task loads. Our experimental outcomes shows the superiority of our proposed model $ACRR$ in terms of average run time, power consumption and average power required than any other state-of-art techniques.

Corresponding Author:

Danthuluri Sudha,
CMR Institute of Technology,
Visvesvaraya Technological University (VTU),
Bengaluru, Karnataka 560091, India.
Email: danthulurisudha15@gmail.com

1. INTRODUCTION

Due to ever-enhancing demand and popularity of cloud computing applications, various companies has moved their focus to cloud computing to decrease costs and for the better utilization of resources hence it referred as a next generation computing application. Cloud computing application termed as a novel computational model, which provides on-demand resources and required information, network, storage and data to the subscribers. The cloud-computing model combines hardware device locations and various software resources over the cloud network to decrease the management costs. Cloud Computing is a highly emerged technology which offers high amount of storage capacity, instant scalability and work on the principle of pay-per-use which is only for the time period subscribers are utilizing it [1]. Cloud computing applications are distributed into three sections such as Infrastructure-as-a-service (IaaS), Platform as-a-service (PaaS) and Software as-a-service (SaaS). Virtualization is the most essential technique for cloud computing applications, which used to decrease resource utilization. Moreover, virtualization helps to active the numerous virtual machines (VMs) on one machine by allocating every resource, which belongs to the basic hardware [2].

At the same time, enormous growth of cloud computing applications in recent years has led to rise of numerous data centers which can led to massive amount of energy consumption. The vast utilization of electricity led to various environmental issues due to release of huge carbon as well as it enhances the maintenance cost of information processing centers [3]. The key reason behind massive consumption of power is the insufficient utilization of resources [4]. For an instance, in a recent report it is concluded that several researchers have monitored more than 5000 servers for a long 6 months' time-period and observed that most of the servers operates only at 10% to 50% of their whole capacity. It is also observed that generally more number of resources are assigned to VMs than usually required at that time [4]. Moreover, cloud computing applications experiences extremely variable task-loads due to which server cannot continue for a long-time period and there is a need of resource re-management course after a specific time-period to enhance the present resource sharing [3, 5].

Due to modernization in electronic technologies, the performance cost of various devices is highly emerged as well as the energy consumption in these devices is highly increased [6]. Moreover, several issues can be occurred such as ecological, economic, and technical issues due to high amount of energy consumption. Furthermore, extensive power consumption can give rise to cooling costs and if numerous computing facilities remain active for a long period, then chances of heating problem occurrence can increase due to which system reliability and availability can increase. A cloud computing application needs support of compiler time and execution time to run applications precisely. However, due to high-energy consumption, the execution time of task-loads can be increased from the as usual time.

In cloud computing applications, most essential thing is the allocation of resources efficiently to increase the performance of cloud information centers. Therefore, to enhance the performance and counter these issues in recent time different techniques are introduced by various researchers such as Hierarchical reliability-driven scheduling (HRDS) technique [7], Constrained earliest finish time (CEFT) algorithm [8], Contention-aware energy-efficient duplication (FastCEED) technique [9], Dynamic voltage and frequency scaling (DVFS) [10] and voltage and frequency island (VFI) technique [11]. In above all techniques, Dynamic voltage and frequency scaling (DVFS) is one of the most widely emerged technique for efficient scheduling of task-loads. DVFS is a highly established energy consumption optimization scheme for embedded cloud systems and the energy optimization can be achieved by scaling down the voltage of any chip dynamically. DVFS technique helps to achieve high QoS Internet services by reducing energy consumption in cloud devices.

However, these existing techniques works upon the classic prototype which is not genuine prototype and the communication cost of inter processors becomes very high while using these techniques. In [12], an energy and run-time optimization technique is introduced to decrease energy consumption by scheduling of various task-loads on numerous embedded systems. In [13], various energy efficient state-of-art-techniques are discussed and basic requirements of embedded systems are presented to increase the performance of embedded computing. In [14], a task-scheduling algorithm is presented to achieve trade-off between energy and performance. This scheduling algorithm helps to increase the flexibility of heterogeneous computing devices (HCD). This technique attenuates the voltage of various tasks according to computing model task load. Hence, the power consumption is reduced and a novel interaction aware DAG prototype is also presented for heterogeneous computing devices (HCD) to reduce energy consumption [15]. However, these techniques do not have features like $DVFS$ or Dynamic Network Shutdown (DNS) [16].

Thus, to enhance performance and reduce energy consumption of heterogeneous computing devices an effective task scheduling technique is needed and rely upon DVFS technique which offers various energy optimization facilities and helps to achieve better resource utilization and trade-off between performance and energy. Therefore, we have presented a Dynamic voltage and frequency scaling (DVFS) based adaptive cloud resource re-configurability $(ACRR)$ technique for cloud computing devices, which efficiently reduces energy consumption, as well as perform operations in very less time. Our proposed $ACRR$ approach helps to achieve high trade-off between energy and performance in heterogeneous computing devices. This scheduling technique helps to achieve high QoS by operating on lower transmission rates and conceding minimum delay. This technique supports efficiently to achieve objective of enhancing the cost of interaction and computational energy. This technique adaptively schedules tasks, allocates its resources, and provides scalable implantation. Our proposed $ACRR$ technique performs much better compare to other state-of-art-tecniques.

This paper is organized in following sections, which are as follows. In section 2, we present related work in the field of energy consumption in cloud environment. In section 3, we described our proposed $ACRR$ methodology. In section 4, experimental results and performance evaluation shown, and section 5 concludes our paper.

2. RELATED WORK

In recent years, the demand of cloud computing applications has taken immense growth. Therefore, to control high demand from the clients, there is a need of excessive resources of different types. All these resources consumes high amount of electricity and hence power consumption is more. According to a 2013 research in United States of America, the information cloud processing centers consumes $910\ billion\ KWh$ electricity, which is almost same as the summation of 34 thermal energy plants whose electricity generation capacity is $500\ MW$ per year. This consumed energy was sufficient for entire New York City for two years and till 2020, this energy consumption will rise to $1400\ billion\ KWh$ which is enormous electricity consumption and almost same as the summation of 50 thermal energy plants [17]. Hence, the energy consumption in cloud data centers and computing processors has taken drastic growth. Therefore, controlling of power consumption in data centers and embedded processors is a vital and critical requirement, which need to be focused soon. Thus, an extensive research work described in this section on energy balanced scheduling algorithms and their connection with $DVFS$ for various embedded devices.

In [18], level of power consumption in information processing centers of china and performance is measured. They conclude that the power consumption in information processing centers of CHINA is very high and various techniques are introduced to reduce power consumption and enhance performance. In [19], an energy consumption model is presented based on the server maximum power and degree of CPU utilization to predict the total power in the present server. In [20], mobile cloud computing prototype is introduced to reduce the energy consumption at the time of wireless communication based on dynamic energy-aware cloudlets. They provide simulations results based on the practical experiments. However, execution time is very high using this technique, which may degrade its performance. In [21], an efficient resource allocation model is introduced in cloud environment and a review on existing scheduling and energy consumption strategies is presented. To offer better resources in cloud environment and improve relationship with users, scheduling of resources is an extremely essential topic, which can performance of cloud computing VMs.

In [22], a novel energy aware based on VM scheduling technique is introduced. Here, both network components and resources both are considered to provide an efficient scheduling technique. VM placement and VM migration are the two essential scheduling steps to achieve objective. This technique helps to reduce energy consumption as well as traffic over network. In [23], energy aware resource-scheduling technique is presented based on DVFS networked information processing centers for cloud computing VMs. Here, two types of energies are mainly optimized such as computing energy and communication energy to reduce overall energy consumption while following SLA constraints. This technique is difficult to implement in real-time. In [24], an efficient cost minimization and resource utilization approach is introduced for cloud computing devices using stable parallel applications. This approach precisely decreases cost by choosing devices, which follow the methods of least resource utilization. However, the difficulty is to maintain trade-off between performance and energy consumption. In [25], a precise scheduling algorithm is introduced which rely upon DVFS-enabled network processing centers. This algorithm helps to achieve efficient scheduling for cloud computing VMs. However, this technique introduces optimization problem as well.

In above works, different researchers have utilized different energy consumption and scheduling techniques. However, only few scheduling techniques are well-known to be offered in real-time applications due to various problems occurred in techniques [18, 20, 24, 25], like lack of balancing between performance and power consumption, optimization complexity, large run-time. Therefore, to control these issues, we have introduced a novel dynamic voltage and frequency scaling (DVFS) based adaptive cloud resource re-configurability ($ACRR$) technique for cloud computing devices, which efficiently reduces energy consumption, as well as perform operations in very less time. Therefore, this technique is very much efficient to establish a trade-off between performance and energy consumption.

3. PROPOSED ENERGY BALANCED SCHEDULING ARCHITECTURE

This section defines proposed $ACRR$architecture and its various modules. This section also describes about the optimization of computational and re-configuration cost in information processing centers. Figure 1 demonstrates the proposed $ACRR$ architecture. Here, we introduce a novel adaptive cloud resource re-configurability ($ACRR$) technique for cloud computing VMs. The proposed $ACRR$ technique works on the principle of parallel computing which can handle numerous cloud computing VMs and they can be controlled by a central resource handler. Every cloud computing VM finishes the present allocated task as a self-governing processor by self-controlling its memory and resources. Message passing method is used for Intra-cluster interaction. Whenever a new task is assigned, a central resource handler simultaneously starts implementing resource distribution and admission governing.

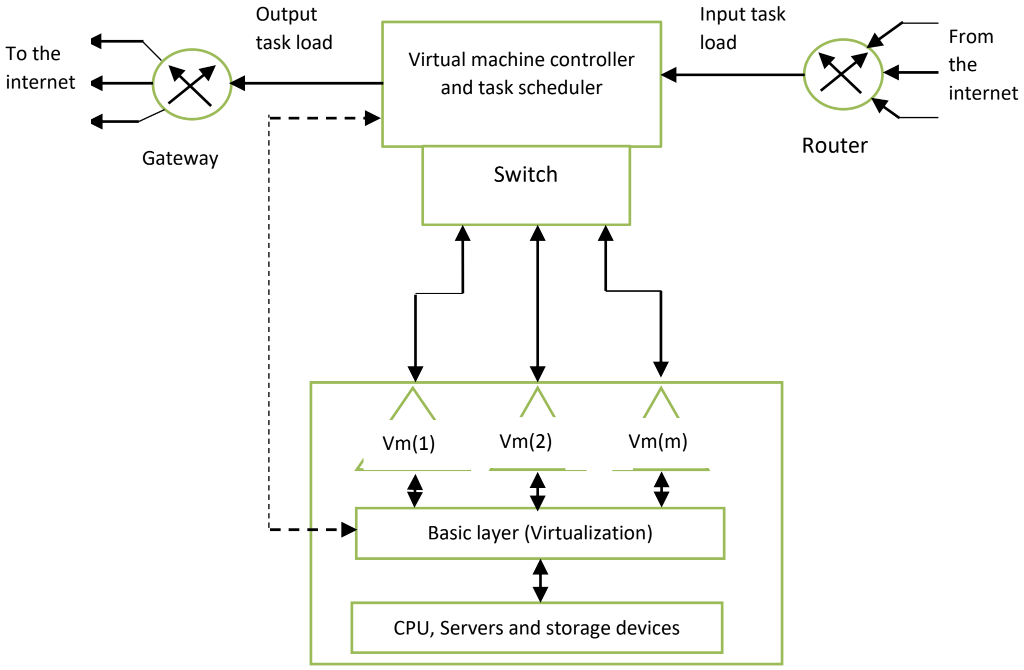
To the internet
Output task load
Gateway
Virtual machine controller and task scheduler
Input task load
From the internet
Router
Switch
Vm(1)
Vm(2)
Vm(m)
Basic layer (Virtualization)
CPU, Servers and storage devices

Figure 1. Architecture diagram of our proposed $ACRR$ model

There are three vital components in our proposed $ACRR$ technique which helps to achieve better resource utilization from an infrastructure perception such as information storage, switched local area network (LAN) and virtual machine handler (VMC) as demonstrated in Figure 1. Whenever, a new task is assigned the arrival time of that task is defined by a_m and size of that task is denoted by S_a in bits. The total processing time of assigned task is less than or equal to the estimated employed time $E_a(sec)$which is very essential for any technique to be adopt in real time scenarios. Our proposed $ACRR$ technique works on some essential parameters which is necessary to work in real time scenarios such as processing task size S_a, the maximum allowed delay E_a in sec and the task granularity which shows the maximum number of tasks $(N_E \gg 1)$ can be grouped into the assigned work.

Assume that the maximum VMs, which can be utilized in the assigned tasks using our proposed $ACRR$methods, can be expressed by $N_J \gg 1$and presented in Figure 1. Our scheduling technique works on the principle that every VM can be demonstrated as a virtual server, which can process i_g bts (Bits per Second). The operating rate $i_g$can be parallelly scaled at the time of execution depending upon the task size S in bits. Assume that all the task follows the interval $[0, i_g^{\uparrow}]$ where $i_g^{\uparrow}$ belongs to the maximum permissible operating rate.

Moreover, the task size S does not affect the estimated time to complete that assigned task by VM, which is fixed prior only to be adopt our model in real time scenarios and denoted by M in seconds. Furthermore, a VM can handle background task-loads of a present assigned task whose size is S and the background task-load size is$S_{\mathbb{b}}$. This background task-load comes under OS (operating system) programs. It is assumed that the background task-load is stored by basic memory of VM. Thus, the background task-load only required computing cost and does not persuade interaction cost. Thus, the utilization parameter β can be expressed as,

$$\beta \triangleq i_g \,.\, \left(i_g^{\uparrow}\right)^{-1} \in [0,1], \tag{1}$$

where, (1) represents that the dynamic elements of the computing energy are the most essential part to decrease the computational cost. Assume that the total energy consumption by VM is to finish a single task of time interval $\omega(\mathbb{d})$is denoted by φ_g in joule at the operating ratei_g. Thus, the dimensionless ratio can be expressed as,

$$\theta(\beta) \triangleq \varphi_g\left(i_g\right).\left(\varphi_g^{\uparrow}\right)^{-1} \equiv \theta\left(i_g\right).\left(i_g^{\uparrow}\right)^{-1}, \tag{2}$$

where, (2) represents the Total Energy Consumption by the concern VM. For an instance, the DVFS based CPU analytical form can be described by the following equation,

$$\theta(\beta) = \beta^2,\ \beta \epsilon [0,1], \tag{3}$$

Here, we can also use μ to compute relative energy cost by concerned VM for the completion of task.

3.1. Modelling for task-load reduction using proposed $ACRR$ technique

In this section, modelling for task-load reduction is discussed. Assume that $N \triangleq \downarrow \{N_{J,}N_E\}$ is the number of tasks which are not overlapped and can be performed in parallel to execute various tasks. Assume that $S_d$is the task size which are assigned to the computing $VM\ (d)$. The process time of different tasks does not rely upon the task lengthS_d. Therefore, the processing rate can be defined as in bits per seconds,

$$i_g(d) = S_d\,.(\omega)^{-1}, \tag{4}$$

This (4) shows that the maximum length permitted for a task is $S_d^{\uparrow} = \omega.i_g^{\uparrow}(d)$. And $\beta \triangleq i_g(d).\left(i_g^{\uparrow}\right)^{-1} \equiv S_d.\left(S_d^{\uparrow}\right)^{-1}$. The total size of a job can be referred as S_a in bits and task size of the task, which is assigned to the $VM(d)$by task scheduler as shown in Figure 1, can be referred as $S_d \gg 0, d = 1, \ldots.,..N$. To reduce task loads, we distribute total job size S_a into N parallel tasks whose size boundary limit can be defined as$\sum_d^N S_d = S_a$.

3.2. Optimization of reconfiguration cost using proposed $ACRR$ technique

This section provides detailed modelling for the optimization of reconfiguration cost. The VM module controller is used to perform two key operations such as balancing the task loads and controlling of virtual machines. For the controlling of virtualization layer as demonstrated in Figure 1, the virtual machine controller $(VMC)$is required which helps to achieve final mapping of VM resources on numerous computing VMs. The VM's characteristics parameters can be described by (9) as,

$$\{\omega, i_g^{\uparrow}(d), \theta_d(\beta_d), \varphi_g^{\uparrow}(d), \mu(d), S_{\mathbb{b}}(d),\ d = 1, \ldots\ldots..N\ \} \tag{5}$$

where, all these parameters can be stated using Virtualization Layer and then they are transmitted to virtual machine controller (VMC) as demonstrated in Figure 1. The operating rate i_g can be scaled up or scaled down using an efficient frequency- scaling scheme, which is controlled by VMC. The power consumption while switching from operating frequency i_1 to frequency i_2 can be $\varphi(i_1{:}\,i_2)$ in joule. This power consumption mainly rely upon the technique used and on the CPU's present in the workstation. This function $\varphi(i_1{:}\,i_2)$ consists of some properties such as the function $\varphi(i_1{:}\,i_2)$ rely upon the entire frequency gap $|i_1 - i_2|$, it becomes zero at $i_1 = i_2$and remain non-decreasing in the entire frequency gap$|i_1 - i_2|$, it is combined convexly at i_1 andi_2. Our model $ACRR$ have some characteristics which can be shown using (10),

$$\varphi(i_1{:}\,i_2) = \mathbb{e}_f(i_1 - i_2)^2,\ joule \tag{6}$$

where, $\mathbb{e}_f$ represents reconfiguration cost for the unit switching of frequency and the values of $\mathbb{e}_f$ is bounded only to some hundreds of $\mu joules/(MHz)^2$. In our model $ACRR$, for every job the size S_a remains same over the respective operating time E_a and any kind of fluctuations not occurred in the task-loads during task-execution. Various tasks can be parallel executed at run-time, due to the induced time overhead using frequency-scaling technique is very less in few μsec for DVFS-enabled architectures. The above-mentioned prediction that the utilization parameter β can be continuous valued and it requires continuous computational rates, which is denoted byi_g. The VMs can offer an instance of CPU's which offer a finite set as,

$$\mathbb{H} \triangleq \{\hat{\imath}^{(0)} \equiv 0, \hat{\imath}^{(1)}, \ldots, \hat{\imath}^{(\mathbb{L}-1)} \equiv i_g^{\uparrow}\}, \tag{7}$$

where, these finite set consists of discrete computational rate $\mathbb{L}$. The optimality loss from both continuous and discrete DVFS enabled techniques, can be eliminated by (8) as,

$$\mathbb{F} \triangleq \{\hat{\beta}^{(0)} \equiv 0, \hat{\beta}^{(1)}, \ldots\ldots, \hat{\beta}^{(\mathbb{L}-1)} \equiv 1\}, \tag{8}$$

where, the discrete value set of β which represent the frequency set $\mathbb{H}$ as shown in (7). A *virtual power consumption curve* can be denoted as $\tilde{\theta}(\beta)$ and formed using *piecewise linear interpolation* and the permitted operating points are,

$$\left\{\left(\hat{\beta}^{(\mathbb{o})}, \theta\left(\hat{\beta}^{(\mathbb{o})}\right)\right), \mathbb{o} = 0, \dots \dots .., (\mathbb{L}) - 1\right\} \tag{9}$$

where, the corresponding vertex points can be presented as,

$$\left(\hat{\beta}^{(\mathbb{o})}, \theta\left(\hat{\beta}^{(\mathbb{o})}\right)\right) \; and \; \left(\hat{\beta}^{(\mathbb{o}+1)}, \theta\left(\hat{\beta}^{(\mathbb{o}+1)}\right)\right) \tag{10}$$

This above mentioned *virtual power consumption curve* maintains the continuity and can be used for the provisioning of VM resources. The use of *piecewise linear interpolation* suggests that with the help of *virtual power consumption curve*, the average energy cost of DVFS enabled techniques remains under the estimated interval of time duration ω. Here, every VM configuration rely upon CPU type, size of memory and cost per time. The VMs cost rely upon the type of configuration. The internal cost of VMs assumed to be zero in all information cloud centers.

3.3. Modelling of efficient resource allocation

In this section, modelling for effective resource allocations is presented. Here, VMC offers two types of services such as balancing of the load and sharing of computational resources. Precisely, these service used to fine-tune the rate of computation $(i_d \;, d = 1, \dots \dots \dots ., N)$and size of task $(S_d \;, d = 1, \dots \dots \dots ., N)$ for the DVFS enabled cloud computing VMs as demonstrated in Figure 1. The main objective is to reduce the total computational energy in joule, which is defined in (11),

$$\varphi_{\mathbb{T}} \triangleq \sum_{d=1}^{N} \varphi_g(d), \tag{11}$$

where, the total computational energy consumption $\varphi_{\mathbb{T}}$ rely on the run-time/job E_a in seconds, operating time/task (ω) needed by VMs demonstrated in Figure 1. Precisely, in Figure 1 all the N links are operated by the switching unit adaptively. The entire computational overhead for the d^{th} link can be expressed as,

$$2M(d) + \omega, \tag{12}$$

where, the condition on total run-time per job to satisfy the solution of optimization problem can be expressed as,

$$\max_{1 \le d \le N} \{2M(d)\} + \omega \le E_a, \tag{13}$$

Assume that the total computational energy optimization issues can be expressed as following,

$$\min_{\{\mathbb{C}_d, i_d, S_d\}} \sum_{d=1}^{N} \theta_d \left(i_d . \left(i_d^{\uparrow}\right)^{-1}\right) \mu(d)\, \varphi_d^{\uparrow} + \mathbb{e}_f (i_d - i_d^{n})^2, \tag{14}$$

where it states that,

$$\left(S_d + S_{\mathbb{b}}(d)\right) \le i_d \omega, \quad d = 1, \dots \dots \dots N, \tag{15}$$

$$\sum_{d=1}^{N} S_d = S_a, \tag{16}$$

$$0 \le i_d \le i_d^{\uparrow}, \; d = 1, \dots \dots \dots N, \tag{17}$$

$$S_d \ge 0, \; i = 1, \dots \dots ., N, \tag{18}$$

where, in (14) starting first term represents computational energy and the following term represents re-configuration energy, which can be expressed by $\varphi_g(d)$ jointly by the computing VMs. Moreover, in (14), i_d^n represents the present rate of computation and i_d is the required rate of computation. Here, i_d^n remains constant while processing of task and shows present state of $VM(d)$ whereas i_d can be variable and changes task to task. Here, (15) shows the condition for which assigned task must be executes in ω seconds whereas (16) shows the condition in which the assigned job must be divided into N parallel tasks.

3.4. Solution for optimization problem using $ACRR$

This section provide various solution for handling optimization problem using our efficient scheduling architecture $ACRR$. Firstly, let the time delay occurred while switching of frequencies remains constant which is induced by DVFS-enabled-techniques. However, the time delay occurred while switching of frequencies between various VMs can be optimized by a non-negative function$\rho_d(i_d,\ i_d^n)(\mathbb{d})$ and the solution for optimization problem, which is stated in (15) and (18), can be derived. This non-negative function$\rho_d(i_d,\ i_d^n)(\mathbb{d})$ helps to retain two characteristics such as,

a. The non-negative function$\rho_d(.,,.)$ remains non-decreasing throughout the interval$|i_d - \ i_d^n|$.
b. The product of i_d and non-negative function$\rho_d(.,,.)$ remain convex ini_d.

To control optimization problem, the first term in (14) which represent computational energy $\sum_{d=1}^{N} \theta_d \left(i_d.\left(i_d^{\uparrow}\right)^{-1}\right).\mu(d).\varphi_d^{\uparrow}$ can be interchanged by the energy function $\mathbb{G}(i_1, \dots \dots \dots, i_N)$. Similarly, the re-configuration energy $\mathbb{e}_f(i_d - \ i_d^n)^2$ can be interchanged by the energy function $J(i_1, \dots \dots \dots . i_N)$ which are conjointly convex in the interval$\{i_d,\ \ d = 1, \dots \dots \dots . . N\}^2$. Similarly, in (15) the parameter ω can be interchanged by $\omega(d)$ without changing any condition or linear equation. The optimization problem presented from (14) to (18) is not a convex type. In fact, this problem is a loosely coupled optimization type, where the parameter $S_d,\ d = 1, \dots \dots \dots N$ represents the computational problems. The solution for optimization of computational and re-configurational problem is presented as,

$$\min_{\{S_d, i_d\}} \sum_{d=1}^{N} \left(\theta_d \left(i_d.\left(i_d^{\uparrow}\right)^{-1}\right)\mu(d)\varphi_d^{\uparrow} + \ \mathbb{e}_f(i_d - i_d^n)^2\right) \tag{19}$$

Assume that, $\{S_d^*,\ i_d^*\ ,\ d = 1, \dots \dots \dots N\}$ represents the solution set for computational and re-configurational optimization problem, which is shown in (15) to (18). At last, we present our proposed $PEBS$ model in an efficient algorithm form, which is as follows:

```
Algorithm : $ACRR$ algorithm
Fix $(N_J, E, E_a)$                                    basic SLA constraints
Fix $(i_d, \mu(d), i_1{}^{\uparrow}, S_a)$                           Parameter processing of VMs $N_J$
Fix $(\varphi_g, S_a, \rho_d, \theta(\beta), \beta, )$Parameter for channel processing of $N_J$ VMs
Collect $S_d$
Verify the attainable constraints of (14)
$\mathbb{i}_1 \cong (S_d + S_{\mathbb{b}}(d)) \leq i_d\omega, \quad d = 1, \dots \dots \dots N,$
$\mathbb{i}_2 \ \cong \sum_{d=1}^{N} S_d = \ S_a,$
if $\approx (\mathbb{i}_1 \& \mathbb{i}_2)\ then,$
        $error(program\ is\ not\ atainable)$
 else
  Special optimization complexity :
  $\min\left(\varphi_{\mathbb{T}}, \mathbb{e}_f, \varphi_g(d)\right)$
 subjected to:
   conditions in (15) and (16)
end if
 return $\varphi_{\mathbb{T}}, \mathbb{e}_f, \varphi_g(d)$
```

4. PERFORMANCE EVALUATION

Now days, the request of cloud computing devices has highly emerged in real-time due to the extensive utilization of informative devices, digital instruments, network appliances and portable gadgets etc. Multimedia-signal-processing method is well-known technique, which can be utilized in these cloud-computing devices. Therefore, the performance of these computing devices must be superior due to the extensive demand of these computing devices in day-to-day life. However, high-energy consumption in these computing devices can disturb their performance. Thus, this section discusses about the balancing between performance and power consumption. To achieve these objectives, we have introduced a Dynamic voltage and Frequency Scaling (DVFS) based Adaptive Cloud Resource Re-Configurability ($ACRR$) technique for heterogeneous computing devices, which efficiently reduces energy consumption, as well as provide superior performance. The run-time can be evaluated considering various jobs as 30, 50, 100, and 1000. Graphical representation of our outcomes is also presented considering execution time, number of tasks and energy consumption. The run-time and total power consumed can be evaluated using different parameters in Table 1, which is demonstrated in the following section. Our proposed $ACRR$ model is tested on $CyberShake$ scientific dataset. We have considered different sizes of scientific workflow experiments as 30, 50, 100 and 1000. Our proposed $ACRR$model employed on 64-bit windows 10 OS with 16 GB RAM which consists on INTEL (R) core (TM) i5-4460 processor. It consists of 3.20 GHz CPU. This project is simulated using $EclipseWS$ Neon.3 editor and code is written in JAVA.

4.1. Comparative study

In this modern era, computing devices has ruled market in different fields like medical, healthcare solutions, trading, software companies etc. Thus, future expertise is clearly in favor of these cloud-computing devices due to their extensive requirements. However, the efficiency of these computing devices may be reduced due to high-energy consumption and lack of efficient resource utilization techniques. Consequently, these issues can be sorted out using efficient task scheduling techniques. Therefore, to allocate resource properly and schedule all the tasks efficiently to overcome power consumption problem, we have presented a novel Dynamic voltage and Frequency Scaling (DVFS) based Adaptive Cloud Resource Re-Configurability ($ACRR$) technique. A precise task scheduling technique can enhance throughput of the system, increase connections with subscribers, offer better resource utilization and can afford to handle multiple tasks at a time etc. The results are demonstrated in contrast to other state-of-art techniques in terms of energy consumption, run-time, power sum and average power as shown in Table 1 using scientific model CyberShake for various jobs as 30, 50,100 and 1000. Energy consumption using our proposed $ACRR$ technique for CyberShake 30 is 1303.74 Watts, CyberShake 50 is 1330.92 Watts,CyberShake 100 is 1436.83 Watts and CyberShake 1000 is 3228.60 Watts demonstrated in Table 1, which is very less in contrast to other state-of-art techniques. Table 2 demonstrates Average Execution time evaluation using our $ACRR$ technique and comparison with other state-of-art-techniques presented for scientific model CyberShake for various jobs as 30, 50,100 and 1000. The Average Execution Time using our $ACRR$ technique for scientific model CyberShake 100 is 31.162 sec and CyberShake 1000 is 4.974 sec.

Table 1. Various parameters comparison for proposed ACRR technique vs DVFS using scientific model $CyberShake$

Parameters	DVFS				$ACRR$			
	$CyberShake$ 30 VM=20	$CyberShake$ 50 VM=30	$CyberShake$ 100 VM=60	$CyberShake$ 1000 VM=20	$CyberShake$ 30 VM=20	$CyberShake$ 50 VM=30	$CyberShake$ 100 VM=60	$CyberShake$ 1000 VM=20
Total Execution Time (s)	6359.41	14448.90	30124.41	74543.57	2938.22	2953.86	3116.22	4794.39
Power Sum (W)	12175922.64	29068552.89	61177338.40	149968122.08	4646390.51	4696963.35	4955133.77	7623610.31
Average Power (W)	19.146320	20.118183	20.308229	20.118184	15.813623	15.901103	15.901103	15.901105
Power Consumption (Wh)	3495.42	8518.39	18966.33	236303.28	1303.74	1330.92	1436.83	3228.60

Table 2. Average execution time comparison of proposed $ACRR$ technique with other State-of-art-techniques using scientific model $CyberShake$

DAGs	Number of nodes	Average Execution time (s)		
		EMO-based algorithm [26]	DVFS	$ACRR$
$CyberShake$ 100	100	31.53	301.244	31.162
$CyberShake$ 1000	1000	22.71	74.54	4.974

4.2. Graphical representation

This section demonstrates the graphical representation of our evaluated outcomes. Here, Figure 2 shows run time comparison of our proposed $ACRR$ technique with DVFS technique using scientific workload $CyberShake$for different jobs as 30, 50, 100 and 1000. Here, Figure 3 shows Power Sum Comparison of our proposed $ACRR$ technique with DVFS technique using scientific workload $CyberShake$ for different jobs as 30, 50, 100 and 1000. Here, Figure 4 shows Average Power Required Comparison of our proposed $ACRR$ technique with DVFS technique using scientific workload $CyberShake$ for different jobs as 30, 50, 100 and 1000. Similarly, Figure 5 shows Power Consumption Comparison of our proposed $ACRR$ technique with DVFS technique using scientific workload $CyberShake$ for different jobs as 30, 50, 100 and 1000. Furthermore, Figure 6 shows Average Run Time Comparison of our proposed $ACRR$ technique with DVFS technique using scientific workload $CyberShake$ for different jobs as 30, 50, 100 and 1000. This result demonstrates the superiority of our proposed $ACRR$ technique in terms of power consumption, execution time, average power and time.

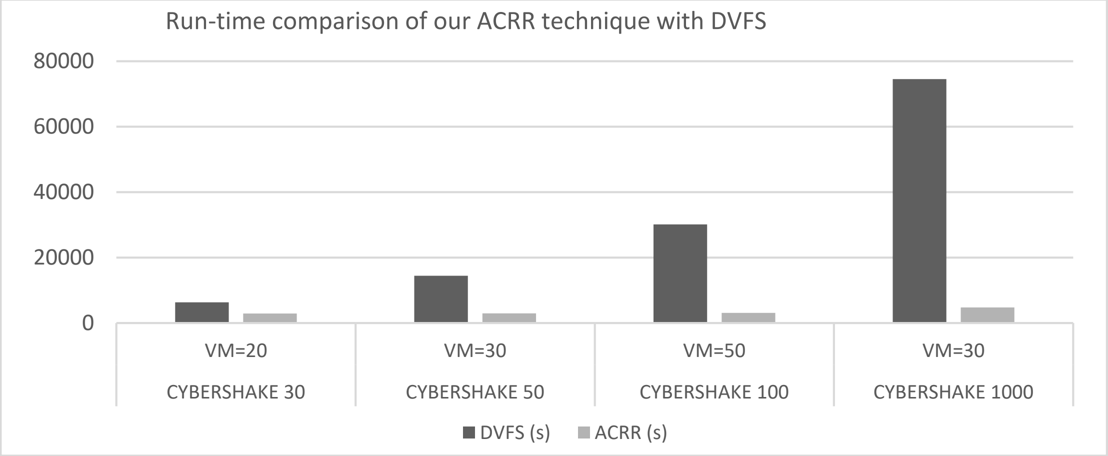

Figure 2. Run-time comparison using our ACRR technique with DVFS

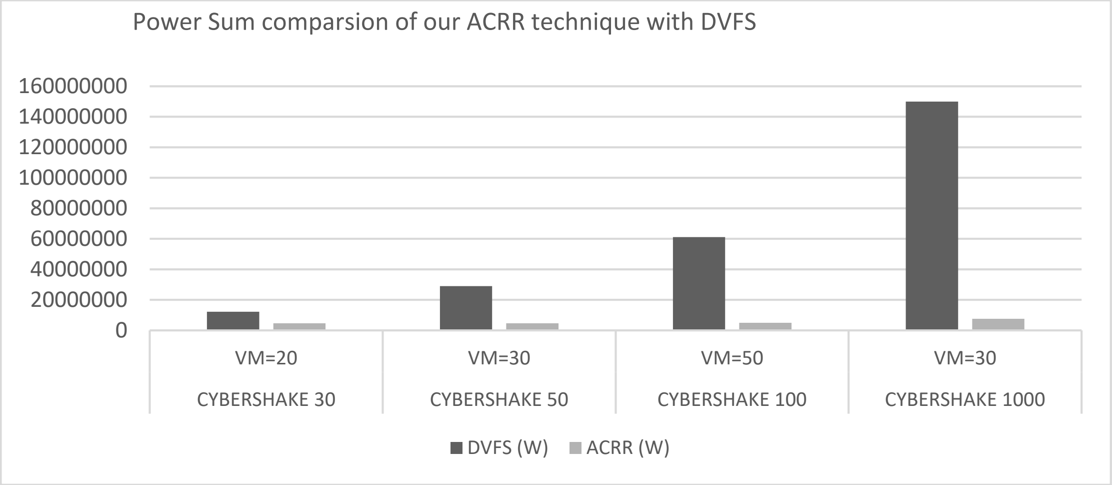

Figure 3. Power sum comparison using our ACRR technique with DVFS

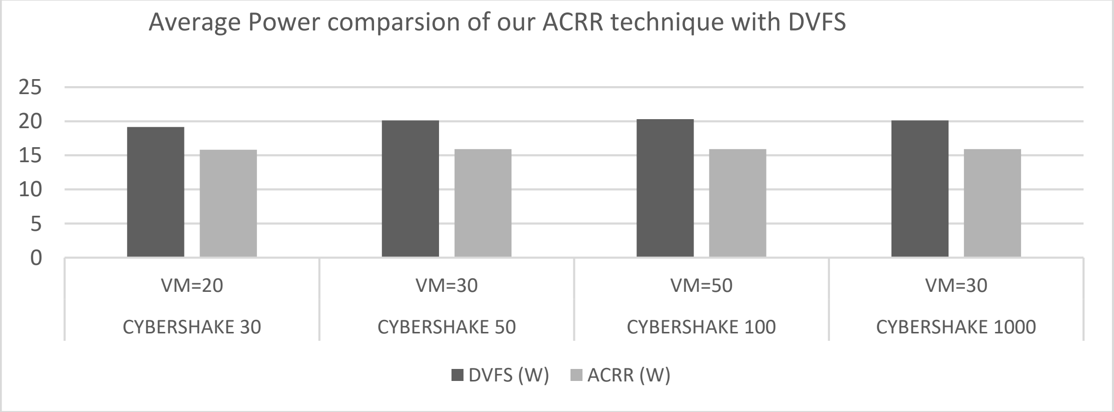

Figure 4. Average power comparison using our ACRR technique with DVFS

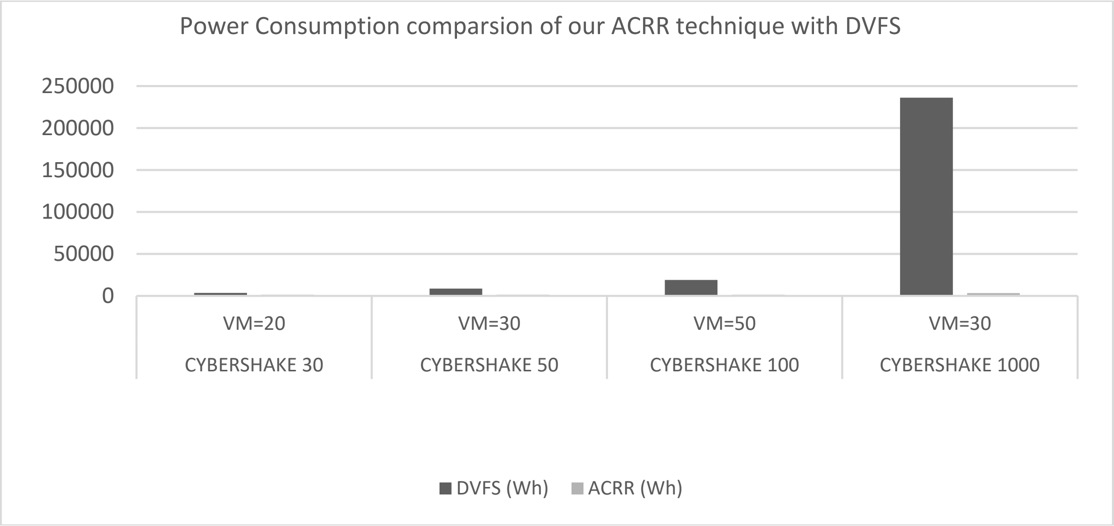

Figure 5. Power consumption comparison using our ACRR technique with DVFS

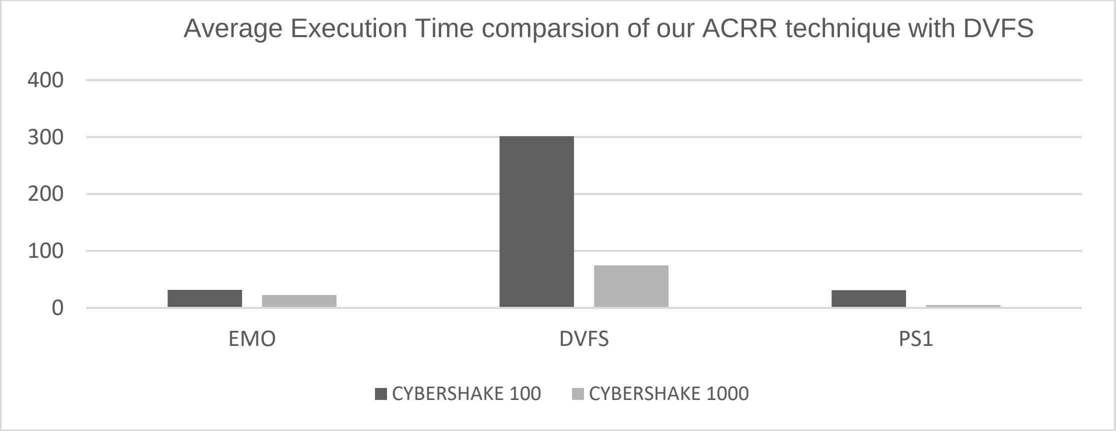

Figure 6. Average execution time comparison using our ACRR technique with DVFS

5. CONCLUSION

The significance of controlling high-energy consumption and task load allocation for every cloud computing VMs is very essential. Therefore, to attain balancing between power consumption and performance for computing processors in cloud environment, we have introduced a novel dynamic voltage and frequency scaling (DVFS) based adaptive cloud resource re-configurability ($ACRR$) technique for cloud computing VMs. Efficient modelling for all the three types of cost such as computation cost and re-configuration cost is presented. The performance of the model is enhanced by reducing all three costs. Furthermore, modelling for efficient resource allocation and utilization is presented and task loads are reduced which is a great challenge for other state-of-art techniques. Numerous resources can be efficiently allocated adaptively at a time using this technique. The experimental results are shown in terms of run time taken, reduction in energy consumption and average power required for cloud computing VMs. The Average Run Time using our proposed $ACRR$ model for scientific model $CyberShake$ 100 is 31.162 sec and $CyberShake$ 1000 is 4.974 sec. Similarly, Power Consumption for $CyberShake$ 30 is 1303.74 Watts, $CyberShake$ 50 is 1330.92 Watts, $CyberShake$ 100 is 1436.83 Watts and $CyberShake$ 1000 is 3228.60 Watts which is very low compare to other state-of-art-techniques. Our experimental results verifies the superiority of our model $ACRR$ in terms of performance and power consumption in contrast to other state-of-art-techniques.Therefore, a trade-off between performance and energy consumption is maintained using our proposed $ACRR$ algorithm. In future, efficient modelling to optimize communication cost will be presented.

REFERENCES

[1] T. Zaidi and R. Rampratap, “Virtual machine allocation policy in cloud computing environment using CloudSim,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 344-354, 2018.

[2] Kumaraswamy S. and M. K. Nair, “Bin packing algorithms for virtual machine placement incloud computing: a review,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 512-524, 2019.

[3] A. Orgerie, *et al*., “A Survey on Techniques for Improving the Energy Efficiency of Large-Scale Distributed Systems,” *ACM Comput. Surveys*, vol. 46, no. 4, 2014.

[4] A. Beloglazov, *et al*., “A taxonomy and survey of energy-efficient data centers and cloud computing systems,” *Advances in Computers*, vol. 82, pp/ 47-111, 2011.

[5] A. Hameed, *et al*., “A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems,” Springer-Verlag Wien, 2014.

[6] Wikipedia, “Moore’s law,” 2012. [Online]. Available: http://en.wikipedia.org/wiki/Moore’s_law.

[7] X. Tang, *et al*., “A hierarchical reliability-driven scheduling algorithm in grid systems,” *Journal of Parallel and Distributed Computing*, vol. 72, no. 4, pp. 525-535, 2012.

[8] M. A. Khan, “Scheduling for heterogeneous systems using constrained critical paths,” *Parallel Comput*, vol. 38, no. 4, pp. 175-193, 2012.

[9] J. Singh, *et al*., “Contention aware energy efficient scheduling on heterogeneous multiprocessors,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1251-1264, May 2015.

[10] L. Zhang, *et al*., “Contention-Aware Reliability Efficient Scheduling on Heterogeneous Computing Systems,” in *IEEE Transactions on Sustainable Computing*, vol. 3, no. 3, pp. 182-194, 2018.

[11] J. Kim and M. Park, “Dynamic frequency scaling regarding memory for energy efficiency of embedded systems,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, pp. 1798-1804, 2018.

[12] K. Li, “Energy-efficient task scheduling on multiple heterogeneous computers: Algorithms, analysis, and performance evaluation,” *IEEE Transactions on Sustainable Computing*, vol. 1, no. 1, pp. 7-19, 2016.

[13] A. Munir, *et al*., “High-performance energy-efficient multicore embedded computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 684-700, 2012.

[14] X. Zhu, *et al*., “3e: Energy-efficient elastic scheduling for independent tasks in heterogeneous computing systems,” *Journal of Systems and Software*, vol. 86, no. 2, pp. 302-314, 2013.

[15] D. Kliazovich, *et al*., “Ca-dag: Modeling communication-aware applications for scheduling in cloud computing,” *Journal of Grid Computing*, vol. 14, no. 1, pp. 23-39, 2015.

[16] R. N. Calheiros, *et al*., “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.

[17] Peter K.D., “Mind the Gap: Energy Availability and the Disconnect with Data,” *Data-Center-Issue-Paper-final826*, [Online]. Available: https://verneglobal.com/uploads/VG-Mind-the-Gap-Energy-Availability-and-the-Disconnect-with-Data-2016.pdf

[18] L. J., *et al*., “Research on national energy consumption and efficiency of data center,” *Energy of China*, vol. 11, pp. 42-45, 2010.

[19] A. Belogazov, *et al*., “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012.

[20] K. Gai, *et al*., “Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing,” *Journal of Network & Computer Applications*, vol. 59, pp. 46-54, 2016.

[21] S. Selvarani, *et al*., “Enhanced two stage heuristics algorithm for VM scheduling,” *2016 Inter. Conf. on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram, pp. 726-729, 2016.

[22] E. I. Nehru, *et al*., “Two stage optimal VM Scheduling scheme for IaaS Cloud,” in *Proc. International Conference on Communication and Security, Pondicheerry Engineering College*, 2016.

[23] N. Kaur, *et al*., “Towards energy efficient scheduling with DVFS for precedence constrained tasks on heterogeneous cluster system,” *2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, Chandigarh, pp. 1-6, 2015.

[24] G. Xie, *et al*., “Resource consumption cost minimization of reliable parallel applications on heterogeneous embedded systems,” in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1629-1640, Aug 2017.

[25] K. A. Sultanpure, *et al*., “An Efficient Cloud Scheduling Algorithm For The Conservation Of Energy Through Broadcasting,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 179-188, 2018.

[26] Z. Zhu, *et al*., “Evolutionary multi-objective workflow scheduling in cloud,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1344-1357, May 2016.