Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Efficient Approach for Resource Provisioning to Manage Workload in Cloud Environment

Jyothi S
Dept. of Information Science and Engg.,
Dr. Ambedkar Institute of Technology,
Bangalore, India.

Dr. Shylaja B. S
Dept. of Information Science and Engg.,
Dr. Ambedkar Institute of Technology,
Bangalore, India.

***Abstract* — Cloud Computing is a perspective which applies parallel or distributed computing, or both to manage its resources. The Cloud is built with physical or virtualized resources over centralized or distributed large data centers. The Resources in the Data Center are provisioned which involves scheduling of tasks to enhance the performance of Cloud services during runtime. Existing Resource Provisioning algorithms improves the CPU utilization and turnaround time of executing the cloud resources which in turn impacts latency and energy overhead. The work is carried out for scheduling of tasks for managing workload under Large Scale cloud computing environment. The data intensive applications are employed as workload for carrying out the experiment. An Efficient Approach for Resource Provisioning (EARP) model is developed for task scheduling by employing Dynamic Voltage Frequency Scaling (DVFS). The EARP model aims to minimize the processing time and energy consumption by effectively utilizing system resources in the cloud.**

Keywords — Cloud Computing, Resource Provisioning, Workflow, Workload, Scheduling, Virtual machine.

I. INTRODUCTION

Internet has revolutionized the way people access information and communicate with each other ever since its invention. Since its evolution from information access systems to service-oriented computing platforms; one of such platforms is Cloud Computing. Cloud Computing, as a business administration and computation framework, has pulled in increasingly more consideration from both industries as well as scholarly community. Cloud Computing is modeled by combining huge number of computational resources into a shared resource pool. This involves accomplishing huge scale and proficient resource usage through the Internet with minimum cost and management.

Resource provisioning is a technique which involves with mapping and scheduling of tasks to virtual machines and consequently virtual machines are also mapping and scheduled onto the physical servers [22, 23]. The Cloud Data Center (CDC) maintains physical and virtual resources with diverse capabilities of processor, memory, network devices, disks, cooling devices etc. The operational cost of CDCs becomes challenging and the need for minimizing the energy consumption is of vital importance. The cloud resources are managed by scheduling of tasks to handle the workflow and workload scheduling techniques.

The workflow scheduling process has been widely utilized model for huge scale data-intensive processing along with scientific application services deployed on cloud environment. The workflows are established by various tasks and sub-tasks, data dependencies among each sub-tasks. It tends to be disconnected into a Directed Acyclic Graph (DAG) in which nodes are connected with sub-tasks sets and edge sets signify the dependency among sub-tasks. Many grid workflow execution frameworks such as ASKALON, Pegasus, etc. support the workload execution on Cloud Computing Environments [16].

Cloud infrastructure services are one of the well-known and widely used Cloud service designs, which offer its users with capabilities to provide Computational Nodes (CN) within Cloud environment. The Computational Nodes in Cloud environment are generally called as instance. The cloud user can access number of boundless CN which significantly reduces the overall cost of ownership for processing the workload [1]. In general, these services are provisioned with Service Level Agreement (SLA) constraint that defines and characterizes the Quality of Service. Hence the Cloud service provider can charge its customer as per Quality of Service requested and amount of time spent by the user. The discovering the right kind of methods for allocating task in Large Scale Cloud environment is a challenging problem.

In the process of workload scheduling, the users submit their jobs to the cloud scheduler. The cloud scheduler inquires the cloud information service for getting the status of available resources and their properties and allocates the various tasks on different resources as per the task requirements. Also the Cloud scheduler will assign multiple user tasks to multiple virtual machines. Workload scheduling can be performed based on different parameters in different ways. They can be statically allocated to various resources at compile time or can be dynamically allocated at runtime. A good workflow scheduling algorithm improves the CPU utilization, turnaround time and cumulative throughput of cloud resources. However, very limited work is done employing DVFS and meeting Quality of Service constraint of workload scheduling. This research work aims at presenting an efficient approach of resource provisioning model that considers minimizing processing time and energy consumption by employing DVFS which effectively utilizes the system resources in cloud computational environment.

The work carried out in this paper are as follows:

- An Efficient Approach of Resource Provisioning (EARP) model is developed for execution of scientific workload in Large Scale cloud computing environment.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

- The EARP model schedules the tasks to minimize the execution time and power consumption for execution of scientific workflow.

The paper organization is as follows: The literature survey is discussed in section II. The workload scheduling problem definition is discussed in section III. The proposed Efficient Approach for Resource Provisioning (EARP) of dynamic workload execution in cloud computing environment is presented in section IV. The experimental analyses are presented in the section V. In last section, the research work is concluded with future research direction.

II. LITERATURE SURVEY

In recent time, there has been lot of work been carried out on the workflow scheduling problem in homogeneous computing environments. David eadamiet. al [2], addresses the major issues of how effectively allocate virtual computing resources in dynamic manner based on applications - QoS prerequisites, energy and cost saving by optimizing the amount of computing servers being used. For addressing this problem, they presented cost effective and dynamic virtual computing node allocation design employing existence of Nash Equilibrium. K.Das guptaet. al, [3] employs Genetic algorithm (GA) for bringing tradeoff in balancing load and reduce makespan time. However, both [2] and [3] did not consider minimizing energy consumption for executing workload. Thus, induces higher cost of execution. For addressing energy efficiency, efficient performance, and reliable processing requirement of modern Big Data processing frameworks, J. R. Doppa et. al,[4] considers that self-aware multi-core framework autonomously optimize the performance parameter for permitting computation process in dynamic manner in accordance with user QoS or SLA prerequisite needs, resource accessibility, energy constraint, and performance requirement. This adaptively traverses right from the application such as scheduling and task mapping to the core such as power gating and DVFS.

Further, Large Scale distributed computational environment such as cloud computing environment comprising of various collection of Virtual Computing Machine (VCM) or processing core offers data storage and computing environment and strategies at a large scale [5]. These strategies involve huge computational expenses and impacts environment. This is due to high energy dissipation at different degrees of storage and computational procedures [6]. K. Li [7], analyses that the fastest super computer in china comprising of sixteen thousand nodes consumes about 17,808 kilowatt (KW) of power. Energy dissipation is a noteworthy issue that influences the improvement and utilization of computational frameworks. In Large Scale Cloud environments, a parallel application with priority bound jobs is described by a Directed Acyclic Graph (DAG). Further, in DAG the nodes describes the jobs and the edges describes the correspondence messages among jobs [8], [9], and [10]. Numerous examinations have been led as of late to limit energy dissipation and at same time fulfilling job SLA perquisite or requirement [11], [12]. Nonetheless, these examinations are confined to autonomous jobs. As Large Scale frameworks keep on being improved, DAG-based work flow with priority bounded jobs rise in size.

The issue of scheduling jobs on different or multiprocessing environment is NP-hard [13]. Various meta-heuristic methods, for example, Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm, and annealing are broadly utilized in DAG-based data-intensive and scientific workflow scheduling [14], [15], and [16]. These methodology for the most part produce superior scheduling quality when compared with heuristic method. This is because of poor search efficacy and frequent strategy computation [17]. Khorramnejad K et. al, [18], discussed about enhancing and improving performance in terms of reducing computation cost for processing multimedia information they focused on merging, prefetching and workflow scheduling together. The processing cost, response time minimization and optimization problems are modeled. Also by considering cost, response time, computational resource allocation, and queueing stability limitations a heuristic method for workload scheduling is modeled. Chunlin, L., Jianhanget. al,[19] showed the significant challenges exist for workload scheduling in hybrid cloud platform. These are different Cloud Service Providers (CSP), Large Scale workload, how to deploy and port the services to CC environment with minimal fiscal budget. For assuring superior resource utilization they presented a Large Scale workload scheduling in private Cloud environment. Besides, for assuring the workload execution is completed within deadline constraint a workload scheduling mechanism is presented using Back Propagation Neural Network (BPNN) under hybrid cloud environment. Junlong Zhou et al., [20], suggest that the workflow scheduling design considers the fiscal budget and computation time under hybrid cloud platform. The first scheduling model is designed using Single Objective (SO) function namely Deadline Constrained Cost Optimization (DCOH) model.The DCOH are designed by reducing fiscal budget of workload scheduling with deadline prerequisite. Second, they presented a Multi-Objective (MO) based workload scheduling method namely Multi-Objective Optimization Method (MOH). The MOH method is designed for bringing tradeoffs between fiscal budget and execution time for scheduling workload execution. However, these models are not efficient in minimizing energy for scientific workflow execution under Large Scale computing environment.

In Cloud infrastructure as a service market, the customers procure cloud services offered by CSP for carrying out their workload executions. Every workload generally comprises of certain deadline prerequisite for guarantying QoS. At the same time, poor QoS will impose strict penalty on CSP. The CSP will generally charge its customer according to QoS requested and execution time. Therefore, for earning better profitability with assured QoS, minimizing execution time and assuring QoS of workload execution is major objectives followed by CSP. However, the existing workload scheduling algorithm assumes that the makespan (i.e., computation time) of tasks in the data-intensive workload application are fixed. However, this hypothesis generally doesn't exist in actual environmental conditions [20]. In this view, the Cloud Servers (CS) have already started to support DVFS practice, which is not employed by existing workload scheduling model [20].

For DVFS-enabled CS, to reduce the cost, energy and execution time of workload execution without affecting system

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

performance and better resource utilization is a major issue. For overcoming the research issues, this paper presents an Efficient Approach for Resource Provisioning (EARP) model is applied for scientific workload execution by employing DVFS in Large Scale Cloud Computing environment. EARP model is based on DVFS for addressing the workflow scheduling problems in Large Scale Cloud Computing environment.

III. WORKLOAD SCHEDULING PROBLEM DEFINITION

In this section the workload scheduling problem is described. A general way for describing workload is to utilize DAG. A workflow is a DAG. A workload is represented as DAGW using following equation

$$W = (J, C)$$

where J depicts task sets described as follows

$$J = \{J_1, J_2, \dots, J_n\}$$

and C depicts data or task control dependencies which can described as follows

$$C = \{(J_p, J_q) | J_p, J_q \in J\}$$

The weights given for each task sets depict the reference makespan. The reference makespan depicts the time for processing a task on a virtual computing node with certain configuration, and the edge weights depicts the amount of data in bits to be transferred among different task sets. The reference makespan of J_p is represented as $R(J_p)$ and the amount of data to be transmitted from J_p and $J_q$is represented by $D(J_p, J_q)$. Along with, all predecessors of task J_p is defined using following equation

$$R(J_p) = \{J_p | (J_p, J_q) \in C\}$$

For respective$W$, $J_{\leftarrow}$ depicts incoming task assuring

$$E(J_{\leftarrow}) = \emptyset$$

and $J_{\rightarrow}$ depicts an outgoing task assuring

$$\nexists J_p \in J : J_{\rightarrow} \in E(J_p).$$

Majority of existing workload scheduling method requires a DAG with single $J_{\leftarrow}$and$J_{\rightarrow}$. This can be effectively satisfied by including pseudo$J_{\leftarrow}$. Similarly, by including pseudo $J_{\rightarrow}$with zero weight to the DAG. Thus, this work consider for respective workload used will possess single $J_{\leftarrow}$ and $J_{\rightarrow}$ [16]. Lastly, workload scheduling in next section aims to minimize the processing energy and communication energy employing DVFS.

IV. AN EFFICIENT APPROACH FOR RESOURCE PROVISIONING OF DYNAMIC WORKLOAD EXECUTION IN CLOUD COMPUTING ENVIRONMENT

In this section an Efficient Approach for Resource Provisioning (EARP) of dynamic workload execution in cloud computing environment is proposed. The model employs DVFS technique for scheduling the Large Scale data intensive workload under cloud computing environment. The task scheduling in EARP is handled by employing the distinct frequencies and with respective time slots for each computing nodes for scheduling task. The proposed EARP scheduling model is described in **Algorithm 1**.

Algorithm 1: Efficient approach for Resource Provisioning of dynamic workload execution in Cloud Computing environment.

Step 1. Start

Step 2. Initialize parameter N, I, I_t basic QoS constraints

Step 3. Initialize parameter $R, P_0, \mathbb{f}^{\uparrow}, G_a, \mathbb{C}_{\mathbb{e}}$ processing of N CNs

Step 4. Initialize parameter $h_a, Q_t, P_s, E^{\mathbb{i}}, \delta$ of channels processing of N CNs

Step 5. Collect M_l

Step 6. Verify the attainable constraints in equation (13) and (14)

Step 7. $\mathbb{m}_1 \cong (M_l \leq Q_t . [(I_t - I). R. (2)^{-1}],)$

Step 8. $\mathbb{m}_2 \cong (M_l \leq \sum_{s=1}^{N} IP_R)$

Step 9. if $\approx (\mathbb{m}_1 \& \mathbb{m}_2)$**then**

Step 10. $error$

Step 11. else

Step 12. Optimization complexity:

Step 13. $\min(\gamma_l, \gamma_{T_{\mathbb{C}}}, \gamma_{F_{\mathbb{C}}}, \gamma_{\mathbb{M}_{\mathbb{C}}})$

Step 14. subjected to:

Step 15. conditions in equations (6) and (9)

Step 16. end if

Step 17. return $\gamma_l, \gamma_{T_{\mathbb{C}}}, \gamma_{F_{\mathbb{C}}}, \gamma_{\mathbb{M}_{\mathbb{C}}}$.

Step 18. Stop.

In EARP model, the load is balanced by the execution of tasks equally across computing nodes as follows

$$\{P_k t_{sk}, \quad s = 1, \dots\dots\dots . . N, k = 0, \dots\dots\dots R\} \qquad (1)$$

where $P_k t_{sk}$ depicts resultant processed information in bits, s depicts the number of server or computing nodes, N is the information block size, and$R$ depicts for each computing node (CN)i.e., the processor with the number of frequencies which is divided between highest and lowest frequencies. The endpoint association link information bandwidth is as follows

$$\{Q_s, \quad s = 1, \dots\dots\dots . N\} \qquad (2)$$

whereQ_s depicts the computing node interacts with the scheduler via a traffic free reliable connection bandwidth rate,.

The Eq. (1) and Eq. (2) are utilized for reducing the overall processing and communication energy in joules as described in below equation

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

$$\gamma_l \triangleq \sum_{s=1}^{N} \gamma_{T_{\mathbb{C}}}(s) + \sum_{s=1}^{N} \gamma_{\mathbb{M}_{\mathbb{C}}}(s), \tag{3}$$

where, $\gamma_{T_{\mathbb{C}}}(s)$ and $\gamma_{\mathbb{M}_{\mathbb{C}}}(s)$ represents the total computation time and interaction energy of $CN(s)$ for the permitted block processing and interaction time underneath QoS constraint I_t, respectively. The interaction energy consumption depends on one-way interaction delay $\mathcal{A}(s)$ which can be expressed as follows

$$\{\mathcal{A}(s),\ \ s = 1, \dots \dots \dots . N\} \tag{4}$$

The interaction delay is occurred by endwise virtual connections. For DVFS technique, the functioning frequency is considered for every computing node and lies in small range of distinct frequencies. The optimum functioning frequency can be selected by switching the CPU frequencies of computing nodes over a various range of possible time periods. However, due to the existence of distinct frequencies a non-convex problem can be occurred which can be addressed as discussed below.

First, let every computing node switches from its current distinct frequency to succeeding distinct frequency to finish the task load. Thus, the time is distributed into $R+1$ distinct indefinite time variables. Therefore, the distinct frequencies for each computing node, the corresponding time slots are unknown. Moreover, every component of time vector defines the time period length during which computing nodes s process at the frequency $\mathbb{f}_k$. System keeps the record of working servers so that it can assign next tasks to them which are coming from the gateway. This information is very essential to forward over all the information processing centers and servers so that the average energy consumption can be reduced by minimizing the execution time. Therefore, the above problem can be expressed in following form,

$$\min_{\{Q_s, t_{sk}\}} \sum_{s=1}^{N}\sum_{k=1}^{R} (G_a \mathbb{C}_{\mathbb{e}} \mathbb{f}_k t_{sk}) + \sum_{s=1}^{N}\sum_{k=1}^{R} 2\left(E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s) P_k t_{sk} \,.\, (Q_s)^{-1}\right), \tag{5}$$

where G_a and $\mathbb{C}_{\mathbb{e}}$ depicts the active gate percentage and effective load of capacitance, respectively

The equation (5) defines the combined energy computation and interaction cost in which cost of the switching frequencies from the arriving task load is also considered, which is subjected to satisfying Eq. (6), (7), (8) and (9).

$$\sum_{s=1}^{N}\sum_{k=1}^{R} P_k t_{sk} = M_l, \tag{6}$$

The Eq. (6) indicates that the summation of products of computing rates of each CNs of their respective time slots must be equal to the arriving task load M_l.Moreover, equation (7) and Eq. (8) presents a factor I which represents the maximum time required for the processing. The total energy computation and interaction time underneath QoS constraint I_t can be distributed in two parts which is shown in, Eq. (7) and Eq. (8) respectively.

$$\sum_{k=1}^{R} t_{sk} \ \leq \ I\,, \ \ S = 1, \dots \dots . . N, \tag{7}$$

The Eq. (7) represents the computational cost.

$$\sum_{k=1}^{R} 2P_k t_{sk} \,.\, (Q_s)^{-1} + I \ \leq \ I_t, \ \ s = 1, \dots \dots \dots . N, \tag{8}$$

The Eq. (8) represents the interaction cost.

$$\sum_{s=1}^{N} Q_s \leq \ Q_t. \tag{9}$$

The above Eq. (9) represents that the volume of information transferred through information processing center should not surpass the total capacity of information network center. This equation provides endwise connection for the bandwidth load matching and also fine tunes computing node bandwidth according to their given task load.

Further, for reducing the non-convex difficulty, this work divides two energy components into two different events such as computation cost and interaction cost. The both events can be scheduled separately to achieve an efficient scheduling as well as execution. Hence the energy consumption will be minimized. Therefore, the computational optimization problem is expressed using below equation

$$\min_{t_{sk}} G_a \mathbb{C}_{\mathbb{e}} \sum_{s=1}^{N}\sum_{k=0}^{R} \mathbb{f}_s t_{sk}\,, \tag{10}$$

From the above observations it may be considered that Eq. (10) is linear for a control parameter t_{sk} and can be sorted out using the Eq. (6) and Eq. (7). Similarly, the interaction aware non-convex variables are Q_s and $P_k t_{sk}$ optimization problem are expressed using below equation

$$\min_{Q_s} 2\left(E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s) P_{sk} t_{sk} \,.\, (Q_s)^{-1}\right) \tag{11}$$

This interaction aware non-convex variables are Q_s and $P_k t_{sk}$ optimization problem can be addressed using the equations (8) and (9) or the following equation (12) also can be a solution,

$$\sum_{s=1}^{N}\sum_{k=1}^{R} 2\left(E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s) P_k t_{sk} \,.\, (Q_s)^{-1}\right) = \ (I_t - I) \sum_{s=1}^{N}\sum_{k=1}^{R} E_s^{\mathbb{M}_{\mathbb{C}}}.\ (2P_{sk} t_{sk} \,.\ (I_t - I)^{-1}). \tag{12}$$

The optimization problem in Eq. (3) are addressed using following equation

$$M_l \leq \ Q_t \,.\, (I_t - I).\, (2)^{-1}, \tag{13}$$

$$M_l \leq \sum_{s=1}^{N} I P_R \ . \tag{14}$$

where, Eq. (13) and Eq. (14) are essential and suitable for the feasibility of optimization problem occurred in the Eq. (3).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Therefore, all the energies are optimized as well as performance of the model also maintained at very high level. Hence, the tradeoff between performance and energy consumption can be achieved using the proposed EARP which is experimentally proved in below section.

V. RESULTS AND ANALYSIS

The proposed EARP and existing resource provisioning algorithms for workload execution on Cloud environment are simulated using CloudSim. In the simulation scenario the comparison of proposed EARP and existing scheduling algorithms [5], [6], [16], [20] on the basis of energy consumed and execution time. The work is carried out considering various data sets, various data centers and virtual machines which are allocated to different hosts. From analysis it can be seen that finding ways to allocate task loads to every embedded processor and to effectively decrease energy consumption in each processor is an essential significance. In recent times, the demand of embedded processors are extremely enhanced in real world due to ample use of digital instruments, network tools, portable gadgets and information devices etc. Various techniques can be utilized in these embedded processors like multimedia-signal-processing-mechanism. Thus, superior performance of embedded devices becomes mandatory requirement due to the ample demand of these embedded devices in daily life. These embedded processors come with two major drawbacks which can affect their efficiency. Firstly, high amount of power is consumed in these devices. Secondly, lack of balance between performance requirement and power consumption is found. Therefore, in this section, for the evaluation of performance and power consumption, various results are demonstrated using existing and proposed EARP based on DVFS techniques [6]. Here, execution time considering different jobs as 30, 50, 100, and 1000 are evaluated. Different graphs are plotted considering time, number of jobs, power consumption etc. Various parameters are considered to evaluate the execution time and power consumption. The model is tested on Inspiral scientific dataset because it is a data-intensive workflow and it is categorized by having CPU intensive tasks that requires enormous amount of memory. The Inspiral workflow is used for analyzing the information collected from the coalescing of compact binary systems such as black holes and binary neutron stars [21]. A sample structure of Inspiral workflow is shown in Fig. 1.The proposed model is implemented using Java programing language. The model is deployed on 64-bit quad core processor with 8 GB RAM on windows 10 operating system.

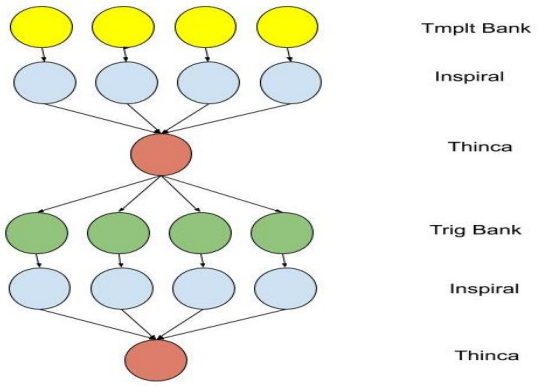

Fig. 1. Inspiral workflow

Performance evaluation: To schedule the task efficiently and to provide proper resource utilization, a novel EARP model based on DVFS technique is introduced. Efficient task scheduling can increase throughput, provide better resource utilization, can avoid overloading of tasks, improve interaction with users etc. Thus, here, the results are compared with other state-of-art techniques in terms of execution time and power consumption.

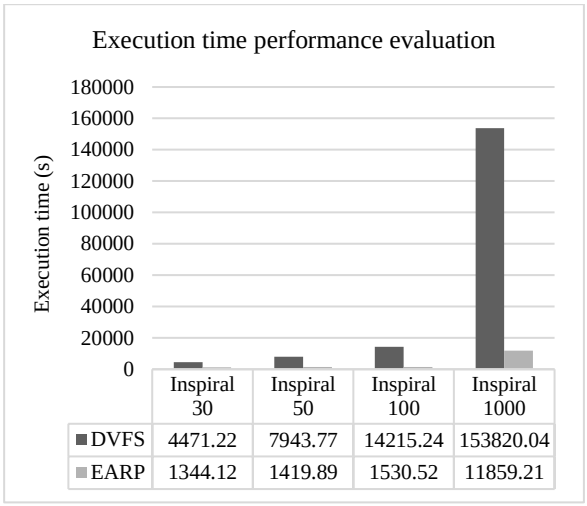

Fig. 2. Execution Time Comparison of EARP over existing DVFS method using scientific workload Inspiral.

Fig. 2 shows performance outcome attained by proposed EARP over existing DVFS model [6] in terms of total execution time considering varied task/job size and virtual computing node for executing Inspiral workflow. The job size of Inspiral is 30, 50, 100, and 1000 are considered. From the results attained, it can be seen that the total execution time of existing resource allocation model for executing scientific workflow Inspiral 30 is 4471.22 sec, Inspiral 50 is 7943.77 sec, Inspiral 100 is 14215.24.41 sec and Inspiral 1000 is 153820.04 sec. Similarly, the total execution time of EARP model for executing scientific workflow Inspiral 30 is 1344.12 sec, Inspiral 50 is 1419.89 sec, Inspiral 100 is 1500.52 sec and Inspiral 1000 is 11859.21 sec. From the overall result attained, the proposed EARP model reduces the average total execution time by 83.4% when compared with the existing resource provisioning model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

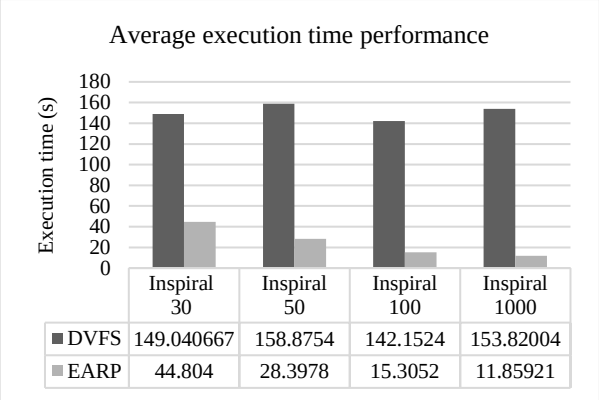

Fig. 3. Average Execution Time of EARP over existing DVFS method using scientific workload Inspiral.

Fig. 3 shows performance outcome attained by the proposed EARP model over existing DVFS in terms of total execution time considering varied task/job size and virtual computing node for executing Inspiral workflow. The job size of Inspiral is 30, 50, 100, and 1000 is considered. From result attained, it can be seen the average execution time of the existing resource allocation model for executing the scientific workflow Inspiral 30 is 149.04 sec, Inspiral 50 is 158.8754 sec, Inspiral 100 is 142.1524 sec and Inspiral 1000 is 153.82004 sec. Similarly, average execution time of EARP for executing scientific workflow Inspiral 30 is 44.804 sec, Inspiral 50 is 28.3978 sec, Inspiral 100 is 15.3052 sec and Inspiral 1000 is 11.85921 sec. From the overall result attained, the proposed EARP reduces the average execution time by 30.29% when compared with standard DVFS model.

Fig. 4 shows performance comparison of average execution time attained by EARP over existing resource allocation model [16], [20]. The average execution time of existing resource allocation model [16], [20] for executing scientific workflow Inspiral 30 is 206.78 sec, Inspiral 50 is 226.19 sec, Inspiral 100 is 206.12 sec and Inspiral 1000 is 227.25 sec. Similarly, for proposed EARP the average execution for executing scientific workflow Inspiral 30 is 44.804 sec, Inspiral 50 is 28.3978 sec, Inspiral 100 is 15.3052 sec and Inspiral 1000 is 11.85921 sec. From overall result attained it can be seen proposed EARP reduce average execution time by 88.41% when compared with existing resource allocation model.

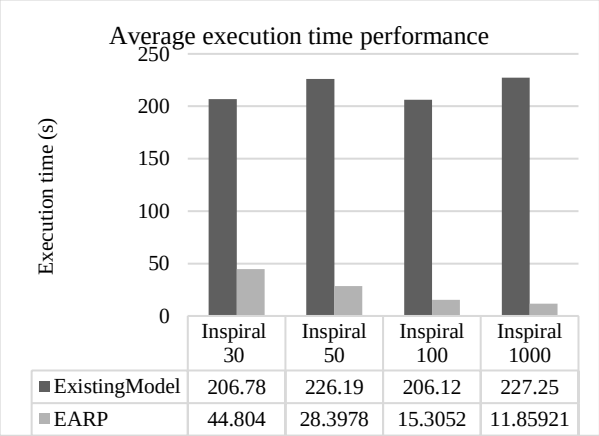

Fig. 4. Average Execution Time of EARP over existing resource allocation method using scientific workload Inspiral.

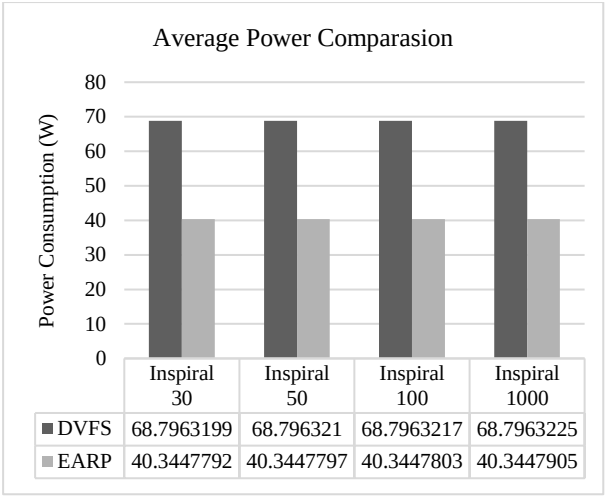

Fig. 5. Average Power Comparison of EARP over existing DVFS method using scientific workload Inspiral.

Fig. 5 shows performance outcome attained by proposed EARP over existing DVFS [6] in terms of average power consumption considering varied task/job size and virtual computing node for executing Inspiral workflow. The job size of Inspiral is 30, 50, 100, and 1000 is considered. From results attained, it can be seen that the average power consumption of existing resource allocation model for executing scientific workflow Inspiral 30 is68.7963199 watts, Inspiral 50 is 68.7963217 watts, Inspiral 100 is 68.7963217 watts, and 68.7963255 watts. Similarly, the EARP for executing scientific workflow Inspiral 30 is 40.3447792 watts, Inspiral 50 is 40.3447797 watts, Inspiral 100 is 40.3447803, watts, and Inspiral 1000 is 40.3447905 watts. From the overall results attained, it can be seen that the proposed EARP reduce the average power consumption by 41.355% when compared with standard DVFS model.

Results and discussion: The work presents a workload scheduling model namely EARP. For evaluating performance of EARP Inspiral scientific workflow. The reason for selecting Inspiral is because it CPU intensive with memory constraint. First experiment is carried out over existing scheduling model that aim to bring good tradeoff between energy minimization meeting workload QoS constraint. From experiment it is seen the EARP model achieves much better performance than existing DVFS model [6]. An execution time performance enhancement of 83.4% is achieved by EARP over existing DVFS based workload scheduling model [6]. Further, experiment are conducted to evaluate performance of average execution time for executing each task of Inspiral workflow using EARP over existing workflow scheduling model [6], [16], and [20]. From result attained it is seen EARP improves processing time by 30.29% over workload scheduling model [6] and 88.41% over workload scheduling model [16] and [20]. The EARP reduces the energy (i.e., power) consumption for executing scientific workflow by 41.355% over existing DVFS based workload scheduling model [6]. From overall result attained it can be seen proposed EARP model bring good tradeoffs between minimizing execution time and energy for executing CPU and memory intensive task.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

VI. CONCLUSION

The paper presents the survey of various state-of-art QoS and energy efficient real time dynamic Large Scale workload scheduling algorithm. From the analysis it can be concluded that finding ways to allocate task loads to every embedded processor and to effectively decrease the energy consumption in each processor is of essential significance. Therefore, a solution to sort out the difficulties to achieve trade-off between performance and energy consumption for virtual machines in a cloud environment using Efficient Approach for Resource Provisioning (EARP) based on DVFS technique is provided. The results are demonstrated in terms of execution time and reduction in power consumption required for processors. From the result it can be seen that the EARP model achieves much better performance than existing workload scheduling model.

REFERENCES

[1] M. Armbrust et al.,B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of cloud computing services: A total cost of ownership approach," in 45th Hawaii Int. Conf. Syst. Sci. IEEE, 2012, pp. 1563–1572.

[2] Davideadami, Stafano Giordano, Michele Pagano, Simone Roma,"A Virtual Machine Migration in a cloud data center scenario: An Experimental Analysis," IEEE ICC 2013.

[3] K.Dasgupta, BrototiMandal, ParamarthaDutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm based load balancing strategy for cloud computing ,"in Elsevier 2013.

[4] J. R. Doppa, R. G. Kim, M. Isakov, M. A. Kinsy, H. Kwon and T. Krishna, "Adaptive manycore architectures for big data computing: Special session paper," 2017 Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Seoul, pp. 1-8, 21017.

[5] G. Xie, G. Zeng, R. Li and K. Li, "Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Large Scale Cloud Computing," in IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 62-75, 1 April-June 2017.

[6] Z. Li, J. Ge, H. Hu, W. Song, H. Hu and B. Luo, "Cost and Energy Aware Scheduling Algorithm for Scientific Workflows with Deadline Constraint in Clouds," in IEEE Transactions on Services Computing, vol. 11, no. 4, pp. 713-726, 1 July-Aug. 2018.

[7] K. Li, "Power and performance management for parallel computations in clouds and data centers," J. Comput. Syst. Sci., vol. 82, no. 2, pp. 174–190, Mar. 2016.

[8] H. Arabnejad and J. G. Barbosa, "List scheduling algorithm for Large Scale systems by an optimistic cost table," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 3, pp. 682–694, Mar. 2014.

[9] Z. Tang, L. Qi, Z. Cheng, K. Li, S. U. Khan, and K. Li, "An energyefficient task scheduling algorithm in DVFS-enabled cloud environment," J Grid Comput., vol. 14, no. 1, pp. 55–74, Mar. 2016.

[10] G. Xie, L. Liu, L. Yang, and R. Li, "Scheduling trade-off of dynamic multiple parallel workflows on Large Scale distributed computing systems," Concurrency Comput.-Parctice Exp., vol. 29, no. 8, pp. 1–18, Jan. 2017.

[11] G. Zeng, Y. Matsubara, H. Tomiyama, and H. Takada, "Energyaware task migration for multiprocessor real-time systems," Future Gen. Comput. Syst., vol. 56, pp. 220–228, Mar. 2016.

[12] K. Li, "Scheduling precedence constrained tasks with reduced processor energy on multiprocessor computers," IEEE Trans. Comput., vol. 61, no. 12, pp. 1668–1681, Dec. 2012.

[13] J. D. Ullman, "NP-complete scheduling problems," J. Comput. Syst. Sci., vol. 10, no. 3, pp. 384–393, Jun. 1975.

[14] D. Tama¸s -Selicean and P. Pop, "Design optimization of mixed criticality real-time embedded systems," ACM Trans. Embedded Comput. Syst., vol. 14, no. 3, p. 50, May 2015.

[15] Y. Xu, K. Li, L. He, L. Zhang, and K. Li, "A hybrid chemical reaction optimization scheme for task scheduling on Large Scale computing systems," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 12, pp. 3208–3222, Dec. 2015.

[16] Z. Zhu, G. Zhang, M. Li and X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud," in IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, pp. 1344-1357, 1 May 2016.

[17] S. Chen, Z. Li, B. Yang, and G. Rudolph, "Quantum-inspired hyper-heuristics for energy-aware scheduling on Large Scale computing systems," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 6, pp. 1796–1810, Jun. 2016.

[18] Khorramnejad, K., Ferdouse, L., Guan, L. et al. "Performance of integrated workload scheduling and pre-fetching in multimedia mobile cloud computing", J Cloud Comp, 7: 13. https://doi.org/10.1186/s13677-018-0115-6, 2018.

[19] Chunlin, L., Jianhang, T. & Youlong, L., "Hybrid Cloud Adaptive Scheduling Strategy for Large Scale Workloads", J Grid Computing (2019) 17: 419. https://doi.org/10.1007/s10723-019-09481-3.

[20] Junlong Zhou et al., Cost and makespan-aware workflow scheduling in hybrid clouds. https://doi.org/10.1016/j.sysarc.2019.08.004, 2019.

[21] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M. Su and K. Vahi, "Characterization of scientific workflows," 2008 Third Workshop on Workflows in Support of Large-Scale Science, Austin, TX, pp. 1-10, 2008.

[22] Bhaskar R, Dr. Shylaja B. S, Knowledge Based Reduction Technique For Virtual Machine Provisioning In Cloud Computing, International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 7, July 2016.

[23] Sharvari J N, Jyothi S, Neetha Natesh, "A Study Based on the Survey of Optimized Dynamic Resource Allocation Techniques in Cloud Computing", International Journal of Emerging Technology & Research,Volume 1, Issue 4, May-June, 2014.