Indonesian Journal of Electrical Engineering and Computer Science
Vol. 21, No. 2, February 2021, pp. 1039~1047
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v21.i2.pp1039-1047

Quality and energy optimized scheduling technique for executing scientific workload in cloud computing environment

Nagendra Prasad S, Subash S kulkarni
PES Institute of technology, Bangalore South Campus, India

Article Info

Article history:

Received Apr 29, 2020
Revised Jun 15, 2020
Accepted Jul 8, 2020

Keywords:

BigData workload
Cloud computing
Multi-core environment
Multi-objective optimization problem
QoS
SLA

ABSTRACT

Modern BigData data-intensive and scientific workload execution is challenging. The major issues are reliable processing, performance efficiency and energy efficacy perquisite of BigData processing framework. This work assume self-aware MC architectures that autonomously adjust or optimize their performance to accommodate users quality of service (QoS) performance requirement, job execution performance, energy efficiency, and resource accessibility. Extensive workload scheduling has been presented to minimize energy consumption in cloud computing (CC) environment. However, the existing workload scheduling model induces higher amount of interaction cost between inter-processors communications. Further, due to poor resource utilization, routing inefficiency these existing model induces higher energy cost and fails to meet workload QoS prerequisite. For overcoming research challenges, this paper presented quality and energy optimized scheduling (QEOS) technique for executing data-intensive workload by employing dynamic voltage and frequency scaling (DVFS) technique. Experiment outcome shows QEOS model attains good trade-off between system performance and energy consumption in multi-core cloud computing (CC) architectures when compared with existing model.

Corresponding Author:

Nagendra Prasad S
PES Institute of technology
Hosur Rd, Konappana Agrahara
Electronic City, Bengaluru, Karnataka 560100, India
Email: nagendraps09@rediffmail.com

1. INTRODUCTION

Cloud computing has been center of attraction for industry as well as academia purpose since it provides flexible computing model and business purpose, it mainly focus on the huge computing resource in distributed resource pool to achieve the large scale as well as efficient RU (resource utilization) from internet using low maintenance cost and minimal platform management. Moreover, workflow is one of the common model for the data intensive applications and LS (large scale) scientific computing that runs on IAAS and these models are formed through the data dependencies and the number of task. Moreover, Workflow can be easily abstracted into the DAG (directed acyclic graph) where the edge denotes the dependencies among the task and node denotes the task itself. In addition, there has been numerous advantage of using the Cloud on workflows; some of the grid workflow such as ASKALON and Pegasus have already started supporting the execution of workflows on the Cloud Platforms. Furthermore, IAAS is cloud service model and it provides the customer with the preconfigured release or the provision abilities virtual machine from the given infrastructure of cloud. Customer can access the Virtual Machine also known as instance in unlimited computer resources and lower TCO (total cost of ownership) to compute the tasks

[1]. Normally the above-described service are considered under the SLA (service level agreement) and defines the QoS (quality of service) as well as the tenure. In addition, WFS (workflow Scheduling) is also considered the NP-complete problem and helps in finding the proper scheme for assigning the services/task in the MP (multi-processor) environment. The multi-core methodology describes a stage change in the quantity of processing cores accessible either in single gadgets or in firmly coupled environment [2]. Using methods and strategies obtained from the Network-on-Chip (NoC) [3], SoCs (system-on-chips) [4], FPGA [5], hardware extension such as Intel SGX [6] and graphical processing units (GPU) environment [7, 8], multi-core frameworks are probably going to considerably affect application advancement, pushing towards data flow computational and memory models. In the meantime, the traditional consideration of "a higher number of computational cores than processors" will be released or transformed, diminishing somewhat the multifaceted (complexities) nature of procedures, for example, fluctuation and power/heat dissemination, load balancing and task mapping etc.

Moreover in process of workload scheduling, user have to submit their assigned job to the CS (cloud scheduler), at first CS inquires the CIS (cloud information service) to get the available resource status along with their attributes. Cloud Scheduler assigns the various user jobs to multiple VM (virtual machine). Meanwhile a fair scheduling improvises the cumulative throughput, TAT (turnaround time) and CPU utilization. In addition workload scheduling algorithm, performs on the various parameter in the various ways, it can be allocated statically to the various resources at the given compile time and can be allocated dynamically at the given run time. Moreover recent year have seen various research be performed in the homogeneous CE (computing environments). Furthermore, implied the GA (genetic algorithm) and brings the tradeoff in make-span time reduction and load balancing, however these two method i.e. [9, 10] failed to consider the energy optimization for the workload execution. This results in higher cost execution, moreover to address the reliable processing, efficient performance and energy efficiency for the Big Data frameworks. In [11] SMCF (self-aware multi-core framework) were considered which optimizes the performance and allow the processing in the dynamic environment in accordance with SLA or QoS, resource accessibility, performance requirement and energy constraint. In addition this method traverses right throughout the applications such as task mapping and scheduling to the for power gathering and on a similar fashion DVFS interconnects the manufacture such as DVFS and routing.

Cloud computing is considered as the heterogeneous and distributed computation environment that comprises the various collections of data storage processing or VCM (virtual computing machine) along with the computing environment as well as strategies in the larger scale [12]. These strategies involves the expensive computational cost and directly affects the environment. The above scenario rises mainly due to the HED (higher energy dissipation) in various computational procedure and various storage [13]. Similarly [14] reports that super computer that comprises the 16,000 nodes and it consumes 17,808 KW power. Moreover, Energy Dissipation is the one of the issue, which influences the utilization, and improvisation of computational frameworks. Furthermore in heterogeneous environment priority bound task with the parallel applications are elaborated through DAG (directed acyclic graph), meanwhile DAG node elaborates the edges and jobs which again elaborates the messages the jobs [15-17]. In past several examinations have led to energy dissipations and at the same time it tries to fulfill the requirement or the SLA perquisite [18, 19]. However, the given examination has been confined to the other jobs; nevertheless, heterogeneous framework needs to be improvised regularly

Scheduling task in various environment is considered to be the NP-Hard problems [20], several meta heuristic technique such as ACO (ant colony optimization), tabu search, GA (genetic algorithm), CRO (chemical reaction optimization) have been utilized in the SWS (scientific workflow scheduling) [21-23]. Meanwhile these methods produce the better optimization when compared with the heuristic approach, this occurs mainly due to the bad efficacy and poor FSC (frequent strategy computation) [24]. In [25], the focus was on the integrated pre- fetching model and workload scheduling in the MM (multimedia mobile) Cloud Computing to reduce the cost and enhance the response time performance to process the data (multimedia data). Moreover, the cost optimization and the response time are adapted along with various CR (computation resources) such as QSC (queuing stability constraints, workload conservation) and VM (virtual machine) allocation. Moreover, a heuristic technique is modelled to optimize the cost and response time. In [26], surveyed and observed that there are various challenges in the hybrid cloud environment is to deploy the novel application with minimal cost, various cloud providers and heterogeneous jobs. Moreover, here author introduced job scheduling technique for the heterogeneous workloads in the private cloud is adapted that tries to ensure the absolute resource utilization. Furthermore, task-scheduling technique based on the BP neural network in the integrated cloud and it is modelled to ensure that jobs can be completed in the given deadline. In [27] gives the two distinctive workload design in the HCE (hybrid cloud environment), here at first scheduling model is developed using the objective function Deadline Constrained–OH (optimization for hybrid clouds) to reduce the scheduling workflows cost under the given deadline constraint. Later they

presented multi-objective named MOH (multi-objective optimization in case of hybrid clouds) to optimize the cost and make-span workflows. The main disadvantage of these models that they were not efficient in energy reduction for the scientific workflow in heterogeneous environment. Meanwhile in IAAS cloud, users buys the cloud service that are given the service provider to perform the workflows. Moreover, given workflow is associated with low QoS (quality of service) that imposes penalties on SP (service provider). Furthermore, it is associated to match the deadline and ensures the QoS. Nevertheless SP (service provider) charges primarily based on the QoS and make-span. Hence, in order to ensure the make profitable and improvise QoS, cost reduction and make-span reduction has to be main goal for service provider. However, this workflow scheduling technique considers the fixed execution time in workflow application and their assumption is wrong most of the time in the real time scenario. Further CS (cloud server) supports DVFS (dynamic voltage frequency scaling) method and this was not considered by many WSM (workload scheduling model), hence ensuring the cost optimization and execution time optimization without any effect on resource utilization and system performance remains a major issue. For overcoming research issues, this paper presented a quality and energy optimized scheduling (QEOS) technique for executing data-intensive workflow under heterogeneous CC environment.

The contribution of research work are as follows:

a) This paper presented efficient workload scheduling technique that brings good tradeoff between minimizing energy consumption and processing time for executing data-intensive workflow under heterogeneous CC environment.
b) The QEOS technique attain better performance than existing workflow scheduling model algorithm in terms of power consumption, processing time, and energy efficiency.

The paper organization is as follows: The Section 2 presents quality and energy optimized scheduling technique for data intensive workflow application in heterogeneous CC environment. The result and analysis is discussed in Section 3. In last section, the research work is concluded. Along with, future work is discussed.

2. QUALITY AND ENERGY OPTIMIZED SCHEDULING APPROACH FOR EXECUTING DATA-INTENSIVE WORKLOAD IN CLOUD COMPUTING ENVIRONMENT

This section present quality and energy optimized scheduling (QEOS) technique for executing workload application in CC environment using DVFS technique based on distinct frequencies and their time slots for every VM. The algorithm of proposed QEOS model is described in Algorithm 1.

Algorithm 1: Quality and energy optimized scheduling technique
1. Fix (N, I, I_t)basic SLA constraints
2. Fix $\left(R, P_0, \mathbb{f}^{\uparrow}, G_a, \mathbb{C}_{\circledast}\right)$ Parameter processing of N VMs
3. Fix $(h_a, Q_t, P_s, E^{\mathbb{i}}, \delta,)$Parameter of channels processing of N VMs
4. Collect M_l
5. Verify the attainable constraints in (15) and (16)
6. $\mathbb{m}_1 \cong (M_l \leq Q_t . [(I_t - I). R. (2)^{-1}],)$
7. $\mathbb{m}_2 \cong (M_l \leq \sum_{s=1}^{N} IP_R)$
8. if $\approx (\mathbb{m}_1 \& \mathbb{m}_2)$ *then,*
9. $error(program\ is\ not\ atainable)$
10. else
11. Special optimization complexity :
12. $\min\left(\gamma_l, \gamma_{T_c}, \gamma_{F_c}, \gamma_{\mathbb{M}_c}\right)$
13. subjected to:
14. conditions in (8) and (11)
15. end if
16. return $\gamma_l, \gamma_{T_c}, \gamma_{F_c}, \gamma_{\mathbb{M}_c}$.

First, this work present about the optimization of interaction cost using our QEOS technique. In QOES, virtual machine interacts with the scheduler via a traffic free reliable connection whose transmission rate (TR) Q_s in bits per seconds where$s = 1, \ldots \ldots \ldots, N$, where s describes the virtual machines or server size, and N depicts the information block size. Let the connection is symmetric and bidirectional one. Moreover, let that one-way communication and switching process for s^{th} connection consumes a power $E_s^{\mathbb{M}_{\mathbb{C}}}$ in watts. The consumed power can be

$$E_s^{\mathbb{M}_{\mathbb{C}}} \cong E_I^{\mathbb{M}_{\mathbb{C}}}(s) + E_Q^{\mathbb{M}_{\mathbb{C}}}(s), \quad (1)$$

Where, $E_I^{\mathbb{M}_{\mathbb{C}}}(s)$ represents the consumed power through transmission and frequency switching whereas $E_Q^{\mathbb{M}_{\mathbb{C}}}(s)$ represents power required by the receiver. The absolute power consumption $E_s^{\mathbb{M}_{\mathbb{C}}}$ can be defined as the combination of switching power consumption, big error rate (BER) (noise occurred in the s^{th} connection) and required power for receiver. Further for reducing the interaction cost $E_s^{\mathbb{M}_{\mathbb{C}}}$, the information processing centers should use standard physical servers which can be connected via commodity superfast ethernet switching (ES) devices. Moreover, TCP/IP protocols can be utilized to achieve endwise reliable communication. This protocol helps to attain endwise reliable communication as well as can perfectly operate in presence of congestion/traffic. Therefore, interaction power consumption can be optimized in our framework using following equation

$$E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s) = \delta_s \left(\overline{\mathbb{R}_{t_s}}\, Q_s\right)^2 + E_s^{\mathbb{i}}, s = 1, \dots\dots\dots N, \tag{2}$$

Where,

$$\delta_s \triangleq (j_s)^{-1}\left((\mathbb{S}_{max})^{-1} \cdot \sqrt{2\mathbb{w}.(3)^{-1}}\right)^2 \; s = 1, \dots\dots\dots..N, \tag{3}$$

Where, $\mathbb{S}_{max}$ represents maximum size of the sector and $\mathbb{w}$ is the number of sector which is acknowledged. Here, j_s is the noise power ratio of gain to receiver of s^{th} endwise connection and $\overline{\mathbb{R}_{t_s}}$ is the average time of round trip of s^{th} endwise connection which can be less than $1ms$ for standard information processing centers. $E_s^{\mathbb{i}}$ Represents the cost of idle power for s^{th} endwise connection. Here, the respective one way communication delay can be represented using following equation.

$$\mathcal{A}(s) = \sum_{k=1}^{R} P_k t_{sk} \,.\, (Q_s)^{-1} \tag{4}$$

Therefore, the respective one way interaction energy $\gamma^{\mathbb{M}_{\mathbb{C}}}(s)$ can be defined as in joule using following equation.

$$\gamma^{\mathbb{M}_{\mathbb{C}}}(s) \triangleq E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s)\left(\sum_{k=1}^{R} P_k t_{sk} \,.\, (Q_s)^{-1}\right) \tag{5}$$

Here, the energy spend in endwise connection does not have any impact on energy computation cost and both are totally independent from each other.

Further, the proposed QEOS technique helps to tune the task load portions precisely as $\{P_k t_{sk}, s = 1, \dots\dots\dots..N\,, k = 0, \dots\dots\dots R\}$, where R depicts for each virtual computing processor the number of frequencies grouped among highest and lowest frequencies, $P_k t_{sk}$ describes the respective handled information in bits, k depicts the frequency number which will be in range of $0\ to\ R$. r size, and N depicts the information block size. Along with, the endwise connection information transmitting rates $\{Q_s, s = 1, \dots\dots\dots.N\}$ which can be used to reduce the total computation, reconfiguration and interaction energy and can be defined in joule as,

$$\gamma_l \triangleq \sum_{s=1}^{N} \gamma_{T_{\mathbb{C}}}(s) + \sum_{s=1}^{N} \gamma_{F_{\mathbb{C}}}(s) + \sum_{s=1}^{N} \gamma_{\mathbb{M}_{\mathbb{C}}}(s), \tag{6}$$

Where, $\gamma_{F_{\mathbb{C}}}(s)$ represents the reconfiguration cost of $VM(s)$ for the permitted block processing and interaction time underneath SLA constraint I_t. The interaction energy consumption depends on one-way interaction delay $\{\mathcal{A}(s), s = 1, \dots\dots\dots.N\}$ which is occurred by endwise virtual connections. For DVFS technique, the functioning frequency for every VM lies in small range of distinct frequencies. The optimum functioning frequency can be selected by switching the CPU frequencies of VMs 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 sorted out as,

Every VM 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, we have

known distinct frequencies for each VM whereas the corresponding time slots are unknown for each VM. Moreover, every component of time vector defines the time period length during which VM 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,

$$\min_{\{Q_s,t_{sk}\}} \sum_{s=1}^{N}\sum_{k=1}^{R}(G_a \mathbb{C}_{\mathbb{e}} \mathbb{f}_k^3 t_{sk}) + \sum_{s=1}^{N} \gamma_{F_{\mathbb{C}}}(s) + \sum_{s=1}^{N}\sum_{k=1}^{R} 2\left(E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s) P_k t_{sk} \,.\, (Q_s)^{-1}\right), \tag{7}$$

Where, it is subjected to,

$$\sum_{s=1}^{N}\sum_{k=1}^{R} P_k t_{sk} = M_l, \tag{8}$$

$$\sum_{k=1}^{R} t_{sk} \leq I\,,\; S = 1, \ldots\ldots N, \tag{9}$$

$$\sum_{k=1}^{R} 2P_k t_{sk} \,.\, (Q_s)^{-1} + I \leq I_t,\; s = 1, \ldots\ldots\ldots N, \tag{10}$$

$$\sum_{s=1}^{N} Q_s \leq Q_t. \tag{11}$$

Where, the above equation can be described as follows. The (7) defines the combined energy computation and interaction cost in which cost of switching frequencies from the arriving task load is also considered whereas (8) indicates that the summation of products of computing rates of each VMs of their respective time slots must be equal to the arriving task load M_l. Moreover, (9) and (10) presents a factor I which represents the maximum time required for the processing. The total energy computation and interaction time underneath SLA constraint I_t can be distributed in two parts which is shown in, (9) and (10) respectively. Precisely, (9) represents the computational cost and (10) represents the interaction cost. Then, (11) 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 VM bandwidth according to their given task load.

To reduce the non-convex difficulty, we divide all three energy components into three different events such as computation cost, frequency reconfiguration cost and interaction cost. All three 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 can be defined as follows,

$$\min_{t_{sk}} G_a \mathbb{C}_{\mathbb{e}} \sum_{s=1}^{N}\sum_{k=0}^{R} \mathbb{f}_s^3\, t_{sk}\,, \tag{12}$$

From the above observations we can consider that (12) is linear for a control parameter t_{sk} and can be sorted out using the (8) and (9). Similarly, the interaction aware non-convex variables are Q_s and $P_k t_{sk}$ optimization problem can be defined as follows,

$$\min_{Q_s} 2\left(E_s^{\mathbb{M}_{\mathbb{C}}}(Q_s) P_{sk} t_{sk} \,.\, (Q_s)^{-1}\right) \tag{13}$$

This interaction aware non-convex variables are Q_s and $P_k t_{sk}$ optimization problem can be sorted out using the (10) and (11) or the following (14) also can be a solution,

$$\sum_{s=1}^{N}\sum_{k=1}^{R} 2\left(E_s^{\mathbb{M}_c}(Q_s)P_k t_{sk} \,.\, (Q_s)^{-1}\right) = \; (I_t - I) \sum_{s=1}^{N}\sum_{k=1}^{R} E_s^{\mathbb{M}_c}.\,(2P_{sk}t_{sk} \,.\, (I_t - I)^{-1}). \tag{14}$$

The optimization problem in (6) can be sorted out using following equation

$$M_l \leq \; Q_t \,.\, (I_t - I).\,(2)^{-1}, \tag{15}$$

$$M_l \leq \sum_{s=1}^{N} IP_R \,. \tag{16}$$

Where, (15) and (16) are essential and suitable for the feasibility of optimization problem occurred in the (6). Now, for reconfiguration cost $\gamma_{F_c}(s)$ can be divided into two parts as external and internal reconfiguration cost. First, the cost of distinct frequency changes from $\mathbb{f}_k$ to $\mathbb{f}_{k+h}$. Where, in $\mathbb{f}_{k+h}$ factor h is the movement to outreach the following active distinct frequency for all $VM(s)$. Second, to switch from one active distinct frequency of its respective time period to the following active distinct frequency of its respective time period for all $VM(s)$. Hence, total internal switching cost and external switching cost to get updated reconfiguration cost all VMs can be defined as follows,

$$h_a \sum_{s=1}^{N}\sum_{h=0}^{H} (\Delta \mathbb{f}_{sh}^2), h \,\epsilon\, \{0,1, \ldots\ldots\ldots..H\}, \tag{17}$$

where $H \leq R$ is the total number of active distinct frequencies for every $VM(s)$.

$$\sum_{s=1}^{N} \gamma_{F_c} \;=\; h_a \sum_{s=1}^{N}\sum_{h=0}^{H} (\Delta \mathbb{f}_{sh})^2 \;+\; h_a \sum_{s=1}^{N} \mathbb{E}_c \,. \tag{18}$$

where $\mathbb{E}_c$ represents the first active distinct frequency in the following arriving task load. 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 our proposed QEOS approach which is experimentally proved below.

3. RESULT AND ANNALYSIS

This section presents experiment analysis of proposed QEOS workflow scheduling model over existing workflow scheduling model [12, 13, 23, 27]. The proposed and existing model are implemented using Java programing language. The cloudsim simulator is used for evaluating proposed QEOS and existing workflow scheduling model. For carrying out experiment Inspiral Workflow is used [23, 27]. The workflow scheduling of both proposed and existing model is executed on 64-bit quad core I-7 processor on window OS platform with 16 GB RAM. The performance of both proposed QEOS and existing workflow scheduling model is evaluated in terms of processing time and power consumption (i.e., energy efficiency).

Figure 1 shows processing time performance outcome attained by proposed QEOS over existing DVFS based workflow scheduling model in terms of total processing 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. From result attained it can be seen total processing time of existing workload scheduling method for executing scientific workflow Inspiral 30 is 4471.22 sec, Inspiral 50 is 23896.98 sec, Inspiral 100 is 51551.41 sec and Inspiral 1000 is 153820.04 sec. Similarly, the total processing time of QEOS model for executing scientific workflow Inspiral 30 is 1344.12 sec, Inspiral 50 is 1419.89 sec, Inspiral 100 is 2563.76 sec and Inspiral 1000 is 11859.21 sec. From overall result attained it can be seen proposed QEOS reduce average total processing time by 87.82% when compared with existing workload scheduling model.

Figure 2 shows performance outcome attained by proposed QEOS 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. From result attained it can be seen average execution time of existing resource allocation model for executing scientific workflow Inspiral 30 is 149.04 sec, Inspiral 50 is 477.94 sec, Inspiral 100 is 515.51 sec and Inspiral 1000 is 153.82004 sec. Similarly, average execution time of QEOS for executing scientific workflow Inspiral 30 is 44.804 sec, Inspiral 50 is 28.3978 sec, Inspiral 100 is 25.6376 sec and Inspiral 1000 is 11.85921 sec. From overall result attained it can be seen proposed QEOS reduce average execution time by 91.46% when compared with standard DVFS model.

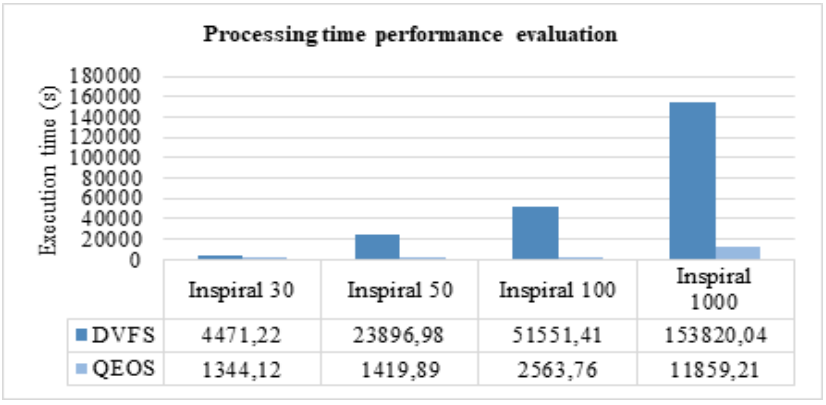

Figure 1. Processing time QEOS over existing DVFS method using scientific workload inspiral

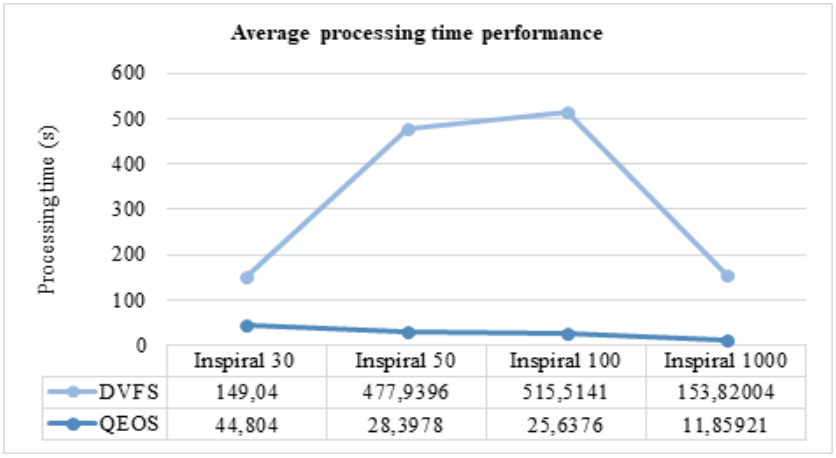

Figure 2. Average processing time of QEOS over existing DVFS based workflow scheduling model for executing Inspiral workflow

Figure 3 show performance comparison of average processing time attained by QEOS over existing workflow scheduling method [23, 27]. The average processing time of existing workflow scheduling method [23, 27] 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. From overall result attained it can be seen proposed QEOS reduce average execution time by 87.068% when compared with existing resource allocation model.

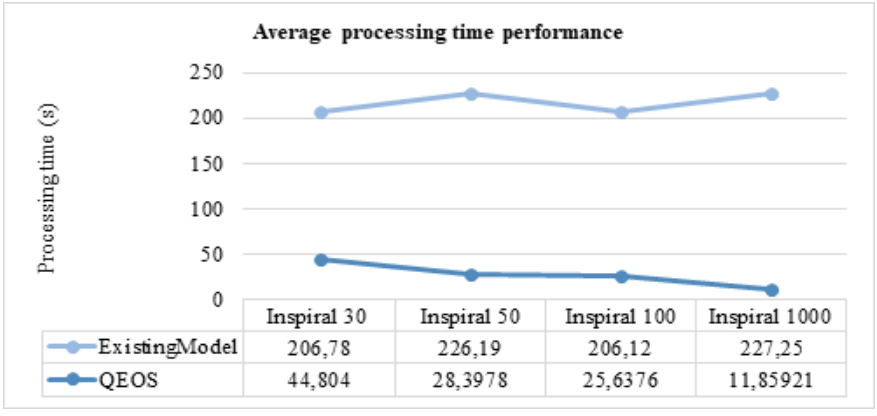

Figure 3. Processing time of QEOS over existing workload scheduling model for inspiral workflow execution

Figure 4 shows performance outcome attained by proposed QEOS over existing DVFS 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. From result attained it can be seen Average power consumption of existing resource allocation model for executing scientific workflow Inspiral 30 is 19.146 watts, Inspiral 50 is 20.12 watts, Inspiral 100 is 20.39 watts, and 19.146 watts. Similarly, the QEOS for executing scientific workflow Inspiral 30 is 15.815 watts, Inspiral 50 is 15.901 watts, Inspiral 100 is 15.9011, watts, and Inspiral 1000 is 15.81 watts. From overall result attained it can be seen proposed QEOS reduce average power consumption by 19.45% when compared with standard DVFS model.

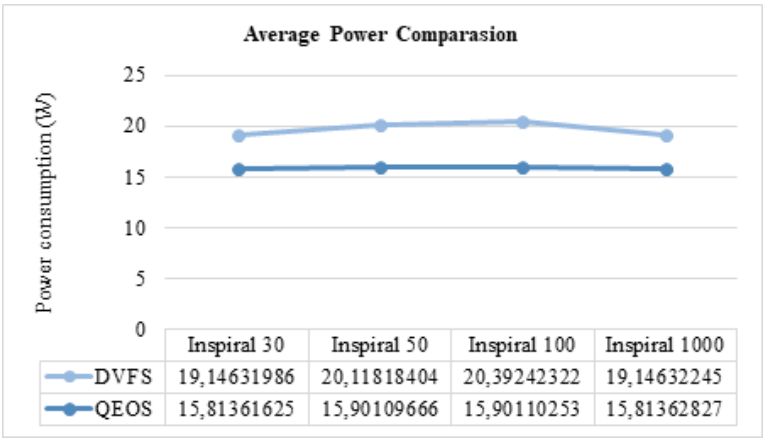

Figure 4. Average power comparison of QEOS over existing DVFS method using scientific workload Inspiral

4. CONCLUSION

Finding ways to allocate task loads to every embedded processor and effectively decrease 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 a novel Quality and energy optimized approach based on Dynamic Voltage and Frequency Scaling technique is provided. Here modelling to solve optimization problem is presented which occurs in most of the existing state-of-art techniques. Methods to reduce task load and increase efficient resource utilization is also presented. The results are demonstrated in terms of processing time taken and reduction in power consumption required for processors. An average processing time performance improvement of 91.46% and 87.068% is attained by QEOS over DVFS and existing workflow scheduling model, respectively. Similarly, average power consumption reduction of 19.45% is attained by QEOS over DVFS based workflow scheduling model. From overall result attained it can be seen proposed attain good tradeoffs between minimizing energy consumption and meeting SLA of workload execution. Future work would further consider performance evaluation considering varied workflow. Along with, would further improve the resource utilization of proposed scheduling model by reducing I/O overhead by better utilizing cloud resources.

REFERENCES

[1] 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*, pp. 1563-1572, 2012.

[2] Ji Wu, Dezun Dong, Xiangke Liao, Li Wang, "Energy-efficient NoC with multi-granularity power optimization," *The Journal of Supercomputing*, vol. 73, no. 4, pp. 1654-1671, 2017.

[3] A. Ivanov and G. De Micheli., "Guest editors' introduction: The network-on-chip paradigm in practice and research," *Design Test of Computers IEEE*, vol. 22, no. 5, pp. 399–403, Sep 2005, doi: 10.1109/MDT.2005.111.

[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, 2017.

[5] Mckeen, F., Alexandrovich, I., Berenzon, A., Rozaz, C. V., Shafi, H., Shanbogue, V., and Savagaonkar, U. R. "Innovative instructions and software model for isolated execution," *In Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy (2013)*, HASP '13, 2013.

[6] Caufield, A. M., Chung, E. S., Putnam, A., Angepat, H., Fowers, J., Haselman, M., Heil, S., Humphrey, M., Kaur, P., Kim, J.-Y., *et al.* "A cloud-scale acceleration architecture," In *Microarchitecture (MICRO), 49th Annual IEEE/ACM International Symposium*, pp. 1-13, 2016.
[7] Softlayer GPU Accelerated Computing, [Online]. Available: http://www.softlayer.com/GPU.
[8] Amazon EC2 Pricing, [Online]. Available: https://aws.amazon.com/ec2/pricing/, 2016.
[9] Davideadami, Stafano Giordano, Michele Pagano, Simone Roma,"A Virtual Machine Migration in a cloud data center scenario: An Experimental Analysis," *2013 IEEE International Conference on Communications (ICC)*, pp. 2578-2582, 2013.
[10] K.Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm based load balancing strategy for cloud computing, " *Procedia Technology*, vol. 10, pp. 340-347, 2013.
[11] 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.
[12] G. Xie, G. Zeng, R. Li and K. Li, "Energy-aware processor merging algorithms for deadline constrained parallel applications in heterogeneous cloud computing," in *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 62-75, 2017.
[13] 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, 2018.
[14] 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.
[15] Ahmed Subhi Abdalkafor1, Khattab M. Ali Alheeti2, "A hybrid approach for scheduling applications in cloud computing environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1387-1397, 2020, doi: 10.11591/ijece.v10i2. pp1387-1397.
[16] Manujakshi B.C.1, K B Ramesh2, "Framework for cost-effective analytical modelling for sensory data over cloud environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 3822-3832, Oct. 2019, doi: 10.11591/ijece.v9i5.
[17] G. Xie, L. Liu, L. Yang, and R. Li, "Scheduling trade-off of dynamic multiple parallel workflows on heterogeneous distributed computing systems," *Concurrency Comput.-Parctice Exp.*, vol. 29, no. 8, pp. 1-18, 2017.
[18] 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, 2016.
[19] K. Li, "Scheduling precedence constrained tasks with reduced processor energy on multiprocessor computers," *IEEE Trans. Comput.*, vol. 61, no. 12, pp. 1668-1681, 2012.
[20] J. D. Ullman, "NP-complete scheduling problems," *Journal of Computer and System sciences*, vol. 10, no. 3, pp. 384-393, 1975.
[21] 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, pp. 1-29, 2015.
[22] K. Sumalatha1, M. S. Anbarasi2, "A review on various optimization techniques of resource provisioning in cloud computing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 629-634, 2019, doi: 10.11591/ijece.v9i1.
[23] Mehran Tarahomi1, Mohammad Izadi, "A hybrid algorithm to reduce energy consumption management in cloud data centers," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 554-561, 2019, doi: 10.11591/ijece.v9i1.
[24] S. Chen, Z. Li, B. Yang, and G. Rudolph, "Quantum-inspired hyper-heuristics for energy-aware scheduling on heterogeneous computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 6, pp. 1796-1810, 2016.
[25] Khorramnejad, K., Ferdouse, L., Guan, L. Anpalagan, A., "Performance of integrated workload scheduling and pre-fetching in multimedia mobile cloud computing," *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1-14, 2018. doi: 10.1186/s13677-018-0115-6, 2018.
[26] Chunlin, L., Jianhang, T. and Youlong, L., "Hybrid Cloud Adaptive Scheduling Strategy for Heterogeneous Workloads", *J Grid Computing*, vol. 17, no. 3, pp. 419-446, 2019, doi: 10.1007/s10723-019-09481-3.
[27] Junlong Zhou, *et al.*, "Cost and makespan-aware workflow scheduling in hybrid clouds," *Journal of Systems Architecture*, vol. 100, 2019. doi: 10.1016/j.sysarc.2019.08.004, 2019.