International Journal of Computational Intelligence in Engineering (IJCIE)
Vol. 01, No. 03, May 2026, pp. 14-21
ISSN: XXXX-XXXX, DOI: 10.5281/zenodo.20393551

Adaptive Multicore Task Scheduling with Dynamic Voltage and Frequency Scaling for Reduced Energy Consumption

Siddesha K[1], Kavitha Narayan BM[2]

[1]Assistant Professor, Dept. of Electronics and Communication Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, India.
Email: siddesha.ec@drait.edu.in

[2]Assistant Professor, Dept. of Electronics and Telecommunication Engineering, Dr Ambedkar Institute of Technology, Bengaluru, India.
Email: kavithanarayan.et@drait.edu.in

Article Info

Article history:

Received Apr 27, 2026
Revised May 19, 2026
Accepted May 23, 2026

Keywords:

Power Optimization
Energy-Efficient Task Scheduling
Dynamic Voltage and Frequency Scaling (DVFS)
Workload Management
Multicore Processors

ABSTRACT

Dynamic Voltage and Frequency Scaling (DVFS) is a successful method to save energy in multi-core systems. However, if heterogeneous tasks having varying number of computations and time requirements are given to different cores, it is a big challenge to run those tasks on time with minimum energy consumption. If these tasks are not scheduled correctly, they can result in higher energy consumption, loss of resources, and compromised system performance. To overcome the aforementioned challenges, a task scheduling framework for multicore processor systems with DVFS is proposed in this work. The proposed architecture is based on an Adaptive DVFS control along with a central scheduling mechanism. First, tasks are analyzed as they are received to determine the time constraints and resources needed. The DVFS mechanism dynamically modifies the operating voltage and frequency of the processing cores depending on the processor configuration in order to provide energy efficient execution. The proposed scheduling strategy then distributes the tasks to the appropriate processor cores for efficient execution. The scheduling problem is modeled as an optimization problem which minimizes the total amount of energy used while meeting a task's deadlines. Moreover, the proposed approach consumes less queuing delay, which enhances the scheduling efficiency and reduces the execution latency. Experimental and comparative analysis results show that the proposed framework has higher energy efficiency and task scheduling effectiveness than the previous ones.

Corresponding Author:

Siddesha K
Assistant Professor, Dept. of ECE, Dr. Ambedkar Institute of Technology, Bengaluru, India.
Email: siddesha.ec@drait.edu.in

1. INTRODUCTION

High performance computing (HPC) systems featuring multicore processors are now a critical part of today's real-time, data-intensive applications. However, due to growing computational requirements, more power consumption is required, leading to increased cooling expenses, battery life and system reliability [1]. These problems are being solved by runtime power management aspects like Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS) in modern computing platforms. However, DPM slows down or turns off idle components to save power [2] and DVFS dynamically lowers the voltage and frequency of the processor to save power [3]. In general, DPM is used for peripheral devices while DVFS is widely used in multicore processors and CMOS-based systems.

The main thrust of this work is directed towards the development of DVFS enabled scheduling in multicore systems. There are two main types of DVFS approaches: intra-task and inter-task scheduling. Intra-task DVFS involves voltage and frequency tuning during the execution of a single task, while inter-task DVFS is scheduling across multiple tasks [4]. The advent of heterogeneous multicore architecture has made efficient energy management more difficult. Current methods are not suitable for dynamic systems with variable workloads, and are too time consuming to analyze offline [5].

Recently, the combination of task scheduling, mapping, migration, and load balancing with DVFS has garnered great interest in both energy efficiency and Quality of Service (QoS) improvements. But current options have a number of drawbacks. Most of the scheduling methods used today are based on sub-optimal global or partition-based scheduling, and cannot ensure real time execution of tasks. Some of the methods involve reducing energy consumption by turning off multiple cores, but under heavy workloads this is not effective. Furthermore, most of the current models take only periodic tasks into account and are not able to adequately process sporadic tasks with unpredictable arrival times and deadlines [12]. To overcome these drawbacks, researchers have combined DVFS with task allocation and adjustment strategies [4] – the problem of optimization is highly nonlinear and very complex.

In recent times, many models have been proposed in the literature for energy-aware scheduling of multicores. Digalwar et al. [11] presented the Leakage Aware Multi-Core Scheduler (LAMCS) for scheduling mixed tasks with deadline constraints. Other machine learning-based task allocation and scheduling optimization techniques include reinforcement learning [7] and linear regression [8] as well as deep Q-learning [9] [10].

Despite the increased energy efficiency of the multicore computing paradigm, strong scheduling frameworks are still needed for dynamic and sporadic workloads. The existing approaches are not always applicable in high workload scenarios. In this work we propose a dynamic workload scheduling framework for heterogeneous multicore systems that is compatible with DVFS. The model proposed aims to reduce the total energy consumption by choosing the best processing cores based on task demands. Using an energy efficiency and task-handling capability, a linear programming formulation and a greedy core-selection mechanism are proposed.

Multicore systems have a number of task scheduling/power management schemes that support DVFS. Kumbhare et al. [2] discussed the problem of underutilization in HPC systems and suggested a new dynamic power management model to provide a better productivity and job completion rate. Qin et al. [4] talked about the DVFS scheduling techniques, including intra-task and inter-task scheduling techniques, based on path-based analysis and ILP optimization.

There are thermal dissipation problems in multicore architecture, as noted by Kumar et al. [13] and energy-efficient scheduling using DVFS and workload classification. For dynamic workloads, Basireddy et al. [5] proposed adaptive task mapping technique for DVFS without the need of offline analysis. Digalwar et al. [11] used threshold-based core shutdown and scheduling with DVFS to reduce energy consumption (both static and dynamic).

Zand et al. [14] presented a genetic algorithm-based static and dynamic scheduling approaches. Hajiamini et al. [15] presented a per-core DVFS scheduling based on dynamic programming with an optimization criterion of Energy-Delay Product (EDP) and the Viterbi algorithm. Choi et al. [16] used DPM and DVFS in an energy optimization scheme. Liu et al. [17] studied cost-aware task scheduling in multicore embedded systems, and Bao et al. [18] proposed a lightweight runtime DVFS scheme by using the power-profiling approach.

Reddy et al. [19] proposed to deal with runtime resource management issues of the heterogeneous multicore systems via the thread-to-core mapping for these systems. Likewise, Jeevitha et al. [20] suggested a hybrid scheduling model that integrates the concept of shortest job first and round robin scheduling in energy efficient cloud computing environment.

3. PROPOSED MODEL

The proposed DVFS-enabled scheduling framework for heterogeneous multicore processor systems is presented in this section. Current scheduling methods are not efficient to deal with the dynamic and sporadic demands with high loads. In order to solve the above problem, the proposed model takes into account task arrival time, execution time, deadlines and processor energy utilization for energy-aware scheduling.

3.1 Task Model

Let's take an example of a heterogeneous multicore system which has multiple processing cores. A set of incoming tasks is denoted as $T = \{T_1, T_2, \dots, T_m\}$, is a distributed system of tasks across available cores with a best-fit heuristic approach. The tasks are assumed to be sporadic and independent in nature.

Each task set T_k has several periodic tasks; it is represented as:

$$\mathcal{T}_k = \{t_{1,k}(p_{1,k}, w_{1,k})t_{2,k}(p_{2,k}, w_{2,k}), \dots., t_{n,k}(p_{n,k}, w_{n,k})\}$$

where $p_{i,k}$ and $w_{i,k}$ denote the execution period and worst-case execution time of task $t_{i,k}$, respectively. The utilization of a task set is computed as:

$$U_k = \sum_{i=1}^{n} \frac{w_{i,k}}{p_{i,k}}$$

The hyperperiod of the task set is the Least Common Multiple (LCM) of all task periods, use this to synchronize periodic tasks:

$$hyp(\mathcal{T}_k) = LCM\left(p_{1,k}, p_{2,k}, \dots, p_{n,k}\right)$$

3.2 Power Model

The proposed framework is based on a DVFS approximation of power consumption of a heterogeneous multicore processor. Dynamic power consumption of CMOS processor is dependent on operating voltage, clock frequency and given as:

$$P_{dp} = c.f.V^2$$

where c is the capacitance, f is the processor frequency, and V is the operating voltage. The leakage power due to sub-threshold leakage current is given as:

$$P_{lp} = V.I_{subn}$$

To perform the task at voltage V_n and frequency f_n, the total energy consumed is the sum of all the energies:

$$E_n = \rho V_n \tau_n + \lambda f_n \tau_n V_n^2$$

where τ_n is execution time, while ρ and λ represent leakage and dynamic power coefficients, respectively. To calculate total system energy, the sum of all the active processing cores' energy consumption is taken.

3.3 Proposed Scheduling Framework

The proposed scheduling architecture comprises of a centralized scheduler and a number of heterogeneous processors labeled as S_j, where $1 \leq j \leq M$. Tasks are queued initially according to First-Come-First-Served (FCFS) strategy. The incoming tasks are queued on initially at according to First-Come-First-Served (FCFS) strategy. The scheduler determines the load of the queues, the load of the processors, and the deadlines of the tasks before assigning them to appropriate cores

Assume that $x_{i,j}$ represents the number of tasks of type i assigned to processor j. The energy of all processors is summed as:

$$E = \sum_{i=1}^{V} \sum_{j=1}^{M} \mu_{i,j} * P_{i,j} * x_{i,j}$$

The goal of the proposed optimization model is to minimize the total energy consumption and meet a set of task deadline constraints. The optimization problem can be stated as follows:

$$\min E$$

subject to:

$$\sum_{j=1}^{M} x_{i,j} = n_i, x_{i,j} \leq B_{i,j} - w_{i,j}$$

where $B_{i,j}$ represents the number of tries, and $w_{i,j}$ denotes the time in the processor queue.

Processing delay and queuing delay are both taken into account in calculating the average response time of the system. If there are no tasks in the queue and the processor load is low, the tasks are scheduled directly to the available processors, reducing both scheduling latency and energy overhead.

Lastly, the proposed framework will use a greedy core-selection strategy. The processing cores are classified based on their energy efficiency and task-handling abilities. Tasks are distributed to the core with the highest availability of energy until the energy is fully consumed. This way, completing tasks becomes more efficient, faster, and consumes less power in heterogeneous multicore systems.

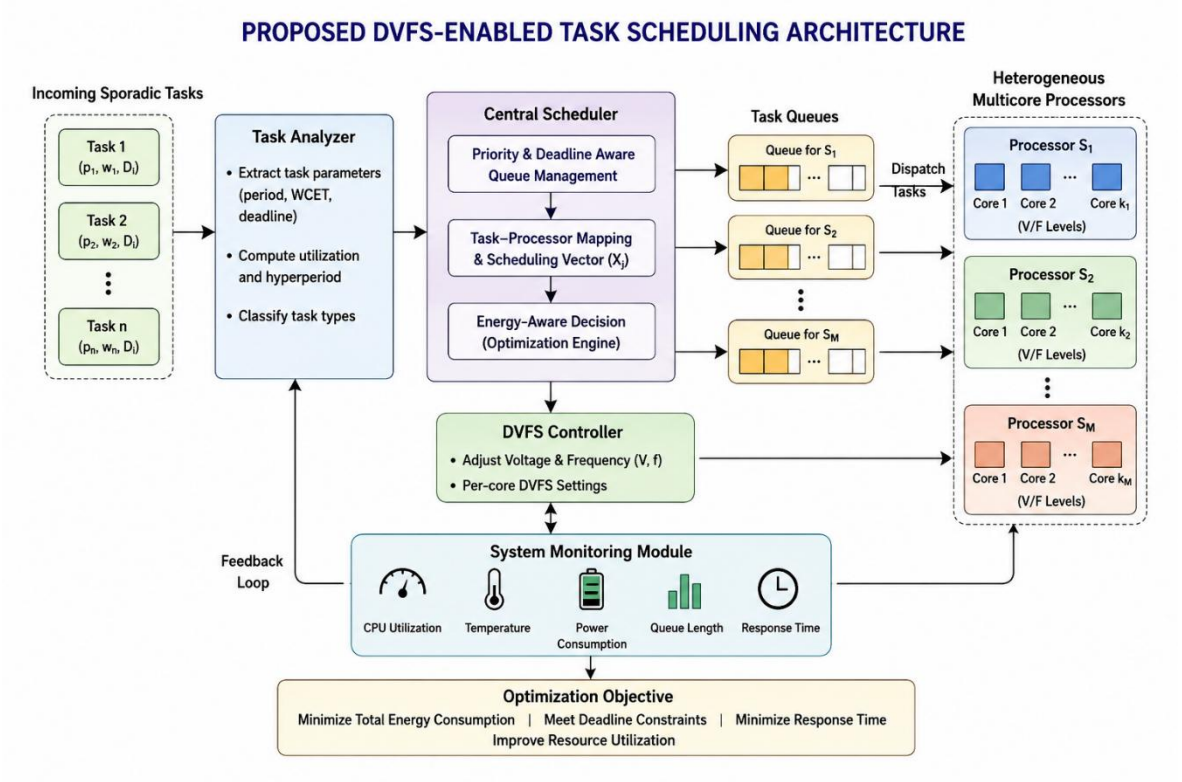

Fig.1.Scheduling architecture

4. RESULTS AND DISCUSSION

This section includes discussions for the experimental results obtained by the suggested approach and compared the performance with the existing schemes. The proposed approach is tested on cloud data center where heterogenous clusters are present. The virtual machines are considered as the multiple cores which are having different configurations. Similarly, the main operating system is considered as physical machine in which all virtual machines are allocated. These physical machine are configured as given below.

Table. 1. Physical machine configuration parameters

Physical Machine	CPU	Min. Power	Max. Power
1	16	210	300
2	52	420	600
3	40	350	500

For the evaluation of the proposed scheduling approach, we have taken the following performance measurement metrices: Waiting time, Energy consumption, Total power on time and Total running time.

- Waiting time: It is a time elapsed during which the job is waiting in the queue due to the other jobs running on all avilable cores.
- Energy consumption: it is a measurement of total energy consumption for specific job to accomblish the task. The overall energy consumption of machine is obtained by summing up the energy consumption by each job.

We have considered a set of tasks which have the total number of tasks as 50, 100, 200, 400 and 800. These tasks are assigned to processors (VMs). First of all, we measure the perforamnce in terms of waiting time for the task before scheduling it to any core. Below given figure 2 shows the performance of proposed appraoch in terms of waiting time.

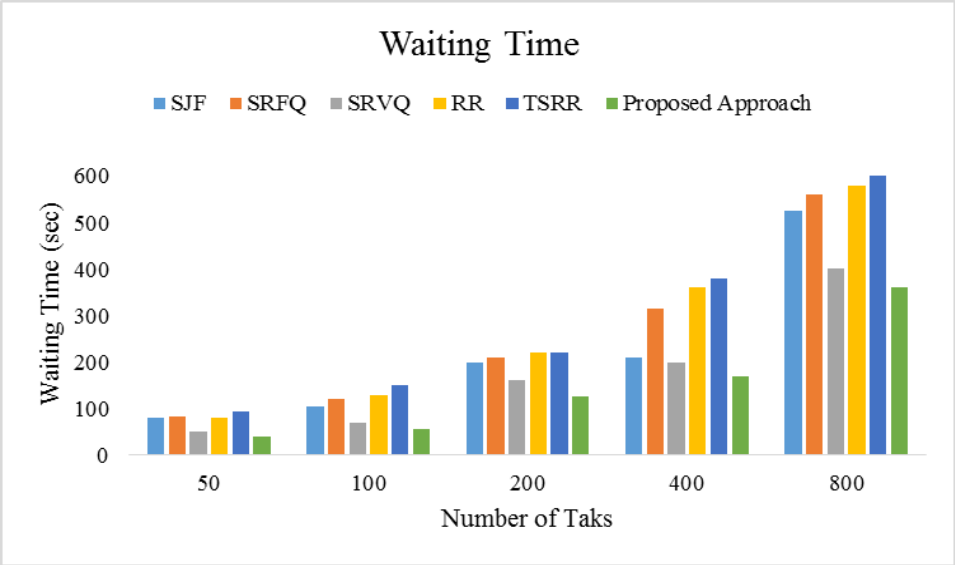

Fig.2. Waiting time comparision

This experiment we got that the current schemes have more scheduling time, so that waiting time increases. The results of proposed approach are compared with the existing technique as described in [20]. These techniques have an average waiting time of 224s, 257.4s, 176s, 274s, 289s and 150s for SJF, SRFQ, SRVQ, RR and Proposed Approach respectively. The traditional scheduling models (SJF, round robin) do not consider the energy consumption status, the resource requirements and deadlines of the cores. Due to these issues, the average waiting time of these techniques increase. Similarly, the improved schemes (SRFQ, SRVQ and TSRR) enhance the performance of these techniques, but in high workload scenarios, these techniques are not able to provide satisfactory results. Whereas the proposed approach is able to handle the dynamic and high workload scenarios. The major impact of the increase in waiting time is on the energy consumption. So, we measure the energy used for each technique and compare the energy used for the technique obtained, with the existing techniques as shown in figure-3 below.

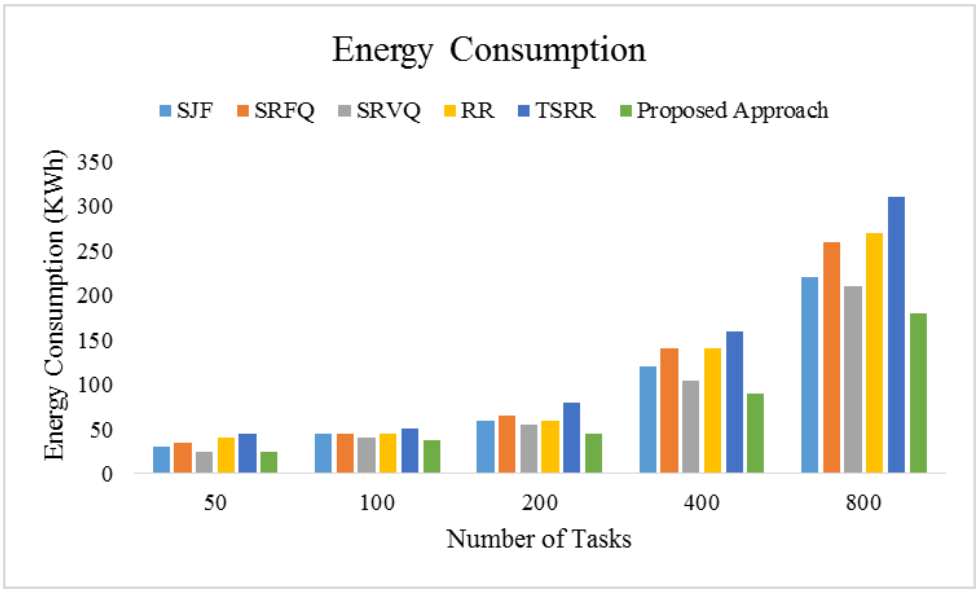

Fig.3. Energy consumption performance

This experiment resulted in an average energy consumption of 95kwh, 109kwh, 87kwh, 111kwh, 129kwh and 75.6kwh for SJF, SRFQ, SRVQ, RR, TSRR and Proposed Approach, respectively. The increasing number of tasks leads to increase in energy consumption. Furthermore, the longer the wait time, the longer the switch on time of the core which consumes more energy. This experiment demonstrates the efficiency with which this approach works in scheduling the jobs.

Further, we extend the experimental analysis and measured the energy consumption performance for varied number of tasks and processor. The obtained normalized energy consumption performance is compared with the existing techniques as mentioned in [21]. In this experiment, number of tasks as 10, 20, 40, 80, 100, and 200 for varied number of processors as 3, 6, 9 and 12. The obtained performance for these experiments is depicted in below given figure 4 and 5 for varied tasks and processors

.

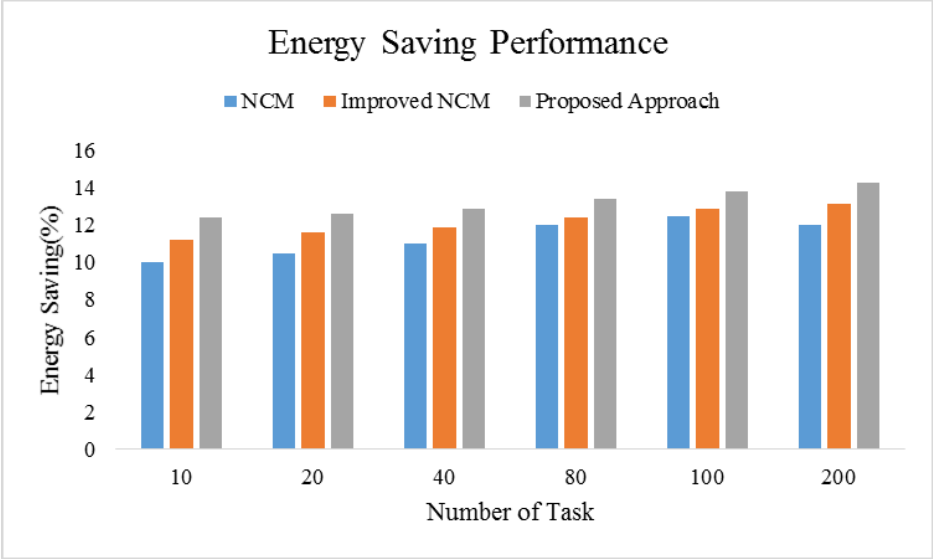

Fig.4. Energy saving performance for varied number of task (12 processors)

The total energy saving percentage for various number of tasks are compared in the above given figure. As the number of tasks are increasing, the energy consumption rate also increases. This case illustrates a great improvement in performance for greater number of tasks. The average energy saving percentage for the task set given is calculated as 11.33, 12.18, and 13.22 for the NCM, Improved NCM and Proposed Approach, respectively. Likewise, the experiment is extended for different number of processors for 200 tasks. Below given figure 5 depicts the performance of energy saving this experiment.

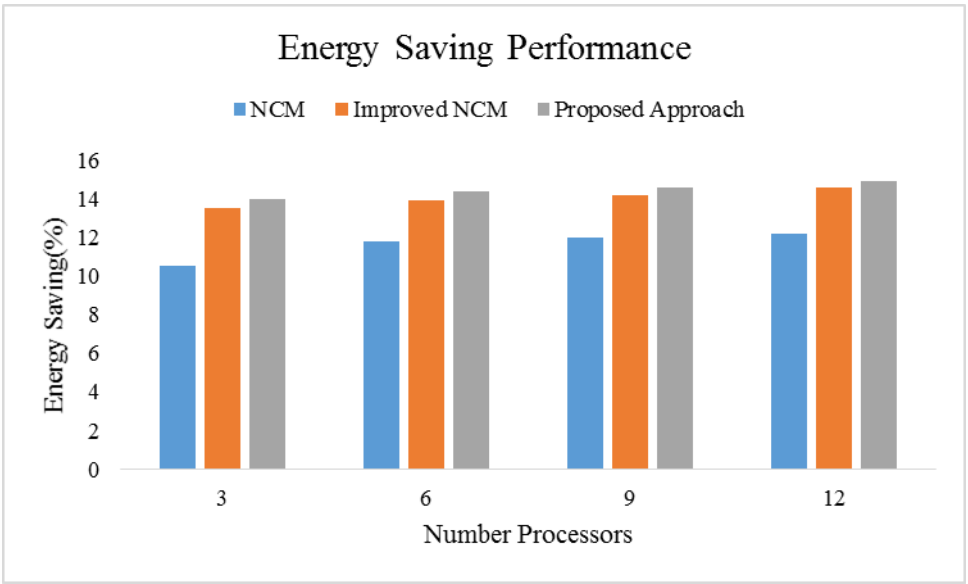

Fig.5. energy saving performance for varied processors (200 tasks)

In this experiment we changed the number of processor and calculated the energy saving performance of 200 tasks. The average results of the experiment are 11.62 for NCM, 14.05 for Improved NCM and 14.46 for Proposed Approach. More cores lead to better dynamic energy saving ratio.

5. CONCLUSION

The present need for high computing system has become very high and consuming energy is a matter of concern which adversely affects the performance of the computing system. To alleviate this problem of too much energy use, several schemes have been developed. In this work, we are interested in reducing the energy usage of multiple cores with heterogeneous DVFS. We present an optimization function to select the best core based on their energy efficiency and applied a greedy selection mechanism which considers the deadline constraints, highest computation capacity and energy efficiency to select the best solution for task processing. An extensive simulation is carried out which shows a significant improvement in the performance in DVFS enabled high performance computing data centers.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY

Data can be provided on genuine request.

REFERENCES

[1] Hajiaminia, S., & Shirazib, B. A. (2020). A study of DVFS methodologies for multicore systems with islanding feature. Advances in Computers, 119, 35.

[2] Kumbhare, N., Akoglu, A., Marathe, A., Hariri, S., & Abdulla, G. (2020). Dynamic power management for value-oriented schedulers in power-constrained HPC system. Parallel Computing, 102686.

[3] Attia, K. M., El-Hosseini, M. A., & Ali, H. A. (2017). Dynamic power management techniques in multi-core architectures: A survey study. Ain Shams Engineering Journal, 8(3), 445-456.

[4] Qin, Y., Zeng, G., Kurachi, R., Li, Y., Matsubara, Y., & Takada, H. (2019). Energy-efficient intra-task DVFSS scheduling using linear programming formulation. IEEE Access, 7, 30536-30547.

[5] Basireddy, K. R., Singh, A. K., Al-Hashimi, B. M., & Merrett, G. V. (2019). AdaMD: Adaptive mapping and DVFS for energy-efficient heterogeneous multi-cores. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[6] Mo, L., Kritikakou, A., & Sentieys, O. (2018). Energy-quality-time optimized task mapping on DVFS-enabled multicores. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(11), 2428-2439.

[7] ul Islam, F. M. M., Lin, M., Yang, L. T., & Choo, K. K. R. (2018). Task aware hybrid DVFS for multi-core real-time systems using machine learning. Information Sciences, 433, 315-332.

[8] Gupta, M., Bhargava, L., & Indu, S. (2020). Dynamic workload-aware DVFS for multicore systems using machine learning. Computing, 1-23.

[9] Zhang, Q., Lin, M., Yang, L. T., Chen, Z., Khan, S. U., & Li, P. (2018). A double deep Q-learning model for energy-efficient edge scheduling. IEEE Transactions on Services Computing, 12(5), 739-749.

[10] Pagani, S., PD, S. M., Jantsch, A., & Henkel, J. (2018). Machine learning for power, energy, and thermal management on multi-core processors: A survey. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[11] Digalwar, M., Raveendran, B. K., & Mohan, S. (2017). LAMCS: A leakage aware DVFS based mixed task set scheduler for multi-core processors. Sustainable Computing: Informatics and Systems, 15, 63-81.

[12] Zhang, D., Guo, D., Chen, F., Wu, F., Wu, T., Cao, T., & Jin, S. (2012). TL-plane-based multi-core energy-efficient real-time scheduling algorithm for sporadic tasks. ACM Transactions on Architecture and Code Optimization (TACO), 8(4), 1-20.

[13] Kumar, N., & Vidyarthi, D. P. (2020). A novel energy-efficient scheduling model for multi-core systems. Cluster Computing, 1-24.

[14] Zand, H. V., Raji, M., Pedram, H., & SharifAbadi, H. H. (2020). A genetic algorithm-based tasks scheduling in multicore processors considering energy consumption. International Journal of Embedded Systems, 13(3), 264-273.

[15] Hajiamini, S., Shirazi, B., Crandall, A., & Ghasemzadeh, H. (2019). A dynamic programming framework for DVFS-based energy-efficiency in multicore systems. IEEE Transactions on Sustainable Computing, 5(1), 1-12.

[16] Choi, J., Jung, B., Choi, Y., & Son, S. (2017). An adaptive and integrated low-power framework for multicore mobile computing. Mobile Information Systems, 2017.

[17] Liu, J., Li, K., Zhu, D., Han, J., & Li, K. (2016). Minimizing cost of scheduling tasks on heterogeneous multicore embedded systems. ACM Transactions on Embedded Computing Systems (TECS), 16(2), 1-25.

[18] Bao, W., Hong, C., Chunduri, S., Krishnamoorthy, S., Pouchet, L. N., Rastello, F., & Sadayappan, P. (2016). Static and dynamic frequency scaling on multicore CPUs. ACM Transactions on Architecture and Code Optimization (TACO), 13(4), 1-26.

[19] Reddy, B. K., Singh, A. K., Biswas, D., Merrett, G. V., & Al-Hashimi, B. M. (2017). Inter-cluster thread-to-core mapping and DVFS on heterogeneous multi-cores. IEEE Transactions on Multi-Scale Computing Systems, 4(3), 369-382.

[20] Jeevitha, J. K., & Athisha, G. (2020). A novel scheduling approach to improve the energy efficiency in cloud computing data centers. Journal of Ambient Intelligence and Humanized Computing, 1-11.

[21] Maurya, A. K., Modi, K., Kumar, V., Naik, N. S., & Tripathi, A. K. (2020). Energy-aware scheduling using slack reclamation for cluster systems. Cluster Computing, 23(2), 911-923.