An Energy Aware Edge Priority-based Scheduling Algorithm for Multiprocessor Environments

Ashish Kumar Maurya, Anil Kumar Tripathi
Department of Computer Science & Engineering
Indian Institute of Technology (BHU)
Varanasi, India
{akmaurya.rs.cse14, aktripathi.cse}@iitbhu.ac.in

***Abstract*—In Parallel and Distributed computing environments, power optimization is emerging as one of the most important issues in achieving high system performance. Nowadays processing units in computer systems are designed by using software controlled Dynamic Voltage Frequency Scaling (DVFS) to reduce the energy consumption of processing units. In this work, we propose a power aware task scheduling algorithm called Energy Aware Edge Priority Scheduling (EAEPS) for multiprocessor environments which aims to reduce power consumption by exploiting DVFS technique. The proposed algorithm reduces the energy consumption by zeroing the edges of high priorities. The experiments are performed for benchmark task graphs. The performance of the proposed EAEPS algorithm is compared with some existing energy aware scheduling algorithms. The results show that the EAEPS algorithm performs better than the compared algorithms in terms of percentage of maximum energy saving.**

Keywords—Task Scheduling, Clustering, Energy Optimization, Dynamic Voltage Frequency Scaling, Multiprocessors

I. Introduction

Multiprocessors are exploited to compute complex scientific and engineering applications with the aim of providing high performance. However, with the advancement of technology and the society's attention towards environment, these systems have encountered various new challenges like resource outsourcing, and power consumption. Among these issues, power consumption undoubtedly receives the utmost attention. More consumption of power may result in severe ecological, economic, and technical problems. First, high performance computing systems consume lots of power and natural resources, for instance, Google data center consumes as much electricity as a city. Second, these systems indirectly generate lots of CO_2 which aggravates the greenhouse effect and badly affects the environment. Finally, heat dissipation from the systems may cause high temperature which may further increase the failure rate of electronic components. Therefore, minimizing power consumption for multiprocessor systems becomes a vital research topic [1].

Today's processing units are well equipped with the Dynamic Voltage Frequency Scaling (DVFS) technique, which lets the processing units to operate at various frequencies under different supply voltages. The DVFS technique is exploited to minimize the power consumption by scaling down the supply voltages and the frequency levels of the concerned processing units. This reduces the execution speed of the processing units, affecting the schedule length and energy consumption [2]. Our work is concentrated to developing scheduling algorithms which minimizes power consumption of the processing units of multiprocessors by using DVFS technique while executing tasks of a parallel application. A parallel application is a set of tasks having precedence constraints and is described by a Directed Acyclic Graph (DAG), G = (T, E), where T represents a set of nodes in which each node denotes a task and E represents a set of communication edges between tasks.

We have considered a multiprocessor system with software controlled DVFS which has a finite set of frequencies and supply voltages for their processing units. In this work, we make a study of existing scheduling algorithms used for minimization of power consumption and propose a power aware task scheduling algorithm called Energy Aware Edge Priority Scheduling (EAEPS) for multiprocessor environments which aims to reduce power consumption by exploiting DVFS technique. The EAEPS algorithm is an energy aware version of our EPS (Edge Priority Scheduling) algorithm [3]. The proposed algorithm reduces the energy consumption by zeroing the edges of high priorities. The idea presented here leads to the following useful contributions:

- The concept of determining priority of edges and choosing that edge which connects clusters involving high communication costs when perform clustering. This method of prioritization results in obtaining meaningful clustering that can consequently help in minimizing the energy consumption.
- The presented work fruitfully achieves minimization of energy consumption, as shown by the results in comparison to results obtained by such similar approaches. Energy consumption is a significant optimization criterion for the problem of energy aware task scheduling of a parallel application in multiprocessor environments.
- A simulation study is performed on the proposed algorithm and the results are presented for benchmark task graphs which show that the proposed algorithm outperforms the other algorithms.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes the models used in this work and formalizes the problem. Section 4 presents the proposed energy aware scheduling algorithm. Section 4 gives the experimental results and finally section 6 concludes the work.

II. Related Work

In this section, we discuss background and related work of parallel task scheduling algorithms and power aware scheduling algorithms for parallel and distributed systems.

Task scheduling algorithms are generally belonging to two sub-categories: static scheduling algorithms and dynamic scheduling algorithms. In static task scheduling algorithms, information about tasks' computation and communication costs is known in advance and the task allocation to processing units is computed before applications are executed whereas in dynamic scheduling algorithms, allocation of tasks to processing units is done at runtime. Further, static scheduling algorithms are divided into list scheduling algorithms, duplication-based scheduling algorithms and clustering-based scheduling algorithms [4]. The list scheduling algorithms first prioritize the tasks and then assign them to the appropriate processing units based on their priorities [5], [6]. The duplication-based scheduling algorithms reduce the communication cost among tasks by duplicating tasks on various processing units [4]. The clustering-based scheduling algorithms first group the tasks into clusters and then map tasks of the same cluster onto same processing unit [3]. This work proposes the EAEPS algorithm which is a clustering-based scheduling algorithm with the aim of reducing power consumption.

A lot of work has been done for task scheduling focusing on power optimization. For example, Terzopoulos and Karatza [7] discussed scheduling of independent tasks, and proposed power aware versions of Min-Min and Max-Min algorithms by using DFVS technique.

Zong et al. [8] proposed two energy efficient duplication-based scheduling algorithms such as EAD (Energy-Aware Duplication) and PEBD (Performance-Energy Balanced Duplication) algorithms for parallel tasks on homogeneous clusters that leverage DVFS to conserve energy dissipation in processors.

Lee and Zomaya [9] addressed the problem of scheduling precedence constrained parallel tasks on heterogeneous computing systems and proposed an ECS (Energy-Conscious scheduling) heuristic that uses DVS (Dynamic Voltage Scaling) to minimize energy consumption.

Hu et al. [10] presented an energy aware scheduling algorithm named EASLA for dependent tasks in the context of SLA (Service Level Agreement) on DVFS enabled cluster systems. The algorithm minimizes energy consumption by scaling frequencies down and distributing each slack to a set of tasks.

Aupy et al. [11] proposed several algorithms to solve the problem of scheduling precedence constrained tasks on homogenous computing systems that aim to minimize energy consumption while thinking about a given bound on the makespan and a reliability threshold.

Kimura et al. [12] proposed an algorithm which reduces energy consumption by using DVFS and reclaiming slack times for non critical tasks in parallel programs executed on real power-scalable PC clusters. The algorithm reclaims slack time by controlling the voltages and frequencies.

Kaur et al. [13] proposed a duplication controlled static energy efficient scheduling algorithm called C-SEED for scheduling of dependent tasks on heterogeneous computing systems. The algorithm controls duplications by using a threshold value with DPM (Dynamic Power Management) technique.

Mei et al. [14] proposed an Energy Aware scheduling by Minimizing Duplication (EAMD) algorithm which reduces energy consumption without degrading makespan. The authors claim that the algorithm not only is easier to operate than both DPM and DVFS, but also produces no overhead of time and energy.

Tang and Tan [15] proposed a reliability and energy aware task scheduling algorithm for precedence constrained parallel applications on heterogeneous systems. The algorithm uses a single processor failure rate model based on DVFS and maintains better tradeoff among reliability, performance, and energy consumption with lower complexity.

Sharifi et al. [16] presented a two phase power aware algorithm called PASTA for scheduling precedence constrained tasks on heterogeneous computing resources. The algorithm doesn't require any special hardware support to reduce power consumption and it provides good tradeoff between makespan and power efficiency.

Wang et al. [2] studied the slack time for non-critical tasks to minimize the energy consumption and considered the Green Service Level Agreement in their work. The authors proposed two power aware scheduling heuristics called PALS (Power Aware List Scheduling) and PATC (Power Aware Task Clustering) for parallel applications.

We propose a novel power aware task scheduling algorithm called Energy Aware Edge Priority Scheduling (EAEPS) for multiprocessor environments which aims to reduce power consumption by exploiting DVFS technique.

III. System Model and Problem Formulation

This section introduces the system models and formalizes the scheduling problem.

A. Energy model

The energy consumption of a processor is due to processing, leakage, and short-circuits. According to Mei et al. [14], power is mostly consumed in the execution of the instructions. Therefore, the energy model, adopted here, considers energy consumption only due to execution ignoring other causes. The energy consumption for a processor, $\mathcal{E}$, is the summation of static energy consumption $\mathcal{E}_{static}$, and dynamic energy consumption $\mathcal{E}_{dynamic}$ [2]. The dynamic power consumption causes due to the charging process involved in the CMOS capacitances whereas the static power consumption is caused by the running, bias and leakage currents.

$$\mathcal{E} = \mathcal{E}_{\text{dynamic}} + \mathcal{E}_{\text{static}} \quad (1)$$

According to [17], the dynamic power consumption P_{dynamic} is determined as follows:

$$P_{dynamic} = A \times C_L \times V_{dd}^2 \times f \quad (2)$$

where, A is the percentage of active logic gates; C_L is the total capacitance load; V_{dd} is supply voltage and f is the operating frequency.

Now, dynamic energy consumption $\mathcal{E}_{\text{dynamic}}$ can be computed as follows:

$$\mathcal{E}_{\text{dynamic}} = \sum_{\Delta t} P_{dynamic} \times \Delta t \quad (3)$$

where, Δt is a time period.

According to [18], $\mathcal{E}_{\text{static}}$ is directly proportional to $\mathcal{E}_{\text{dynamic}}$:

$$\mathcal{E}_{\text{static}} \propto \mathcal{E}_{\text{dynamic}} \quad (4)$$

Hence, the total energy consumption will be directly proportional to the dynamic energy consumption and can be computed as follows:

$$\mathcal{E} = \sum_{\Delta t}(\eta \times V_{dd}^2 \times f \times \Delta t) \quad (5)$$

where, η is a constant determined by processing unit, V_{dd} is the operating supply voltage of processing unit during Δt, *f* is the operating frequency during Δt and Δt is the time period.

B. Processor model

This work considers a multiprocessor system containing homogeneous processors or processing units that are fully connected with the same communication links. Each processing unit can simultaneously perform execution and communication of tasks. The execution of tasks performed by processing units is non-preemptive. Each processing unit is enabled with software controlled DVFS and can operate on a set of operating frequencies *F* and a set of supply voltages *V*.

$$f = \cup_{1 \le i \le K}\{f_i\} \quad (6)$$

$$V = \cup_{1 \le i \le K}\{V_i\} \quad (7)$$

where, f_i is the ith processor operating frequency; V_i is the ith processor supply voltage;

$$f_{min} = f_1 \le f_2 \le \cdots \le f_K = f_{max};$$

$$V_{min} = V_1 \le V_2 \le \cdots \le V_K = V_{max};$$

and K represents the number of operating points for the processing unit.

C. Application model

A parallel application having precedence constrained tasks is modeled as a Directed Acyclic Graph (DAG), G = (T, E), where T represents a set of nodes in which each node denotes a task and E represents a set of communication edges between tasks which show precedence constraints on T:

$$T = \cup_{1 \le i \le N}\{T_i\} \quad (8)$$

where, T_i is ith task and N is the number of tasks in a parallel application.

$e_{i,j}$ denotes an edge between T_i and T_j, i.e. task T_i must be finished before T_j can start, where $1 \le i, j \le N$ and $T_i, T_j \in T$. Here T_j is called a successor of T_i and T_i is called a predecessor of T_j. When two tasks are executed on a same processing unit, their communication cost is ignored. It is assumed in this work that the DAG has only one start task and one end task. Multiple start (multiple-end) tasks are handled by connecting them with a start (end) task having no computation and communication cost. Fig. 1 gives a sample DAG representing the application model and is taken from [19]. The value inside each node presents the computation cost of the task and value on each edge gives the communication cost between tasks.

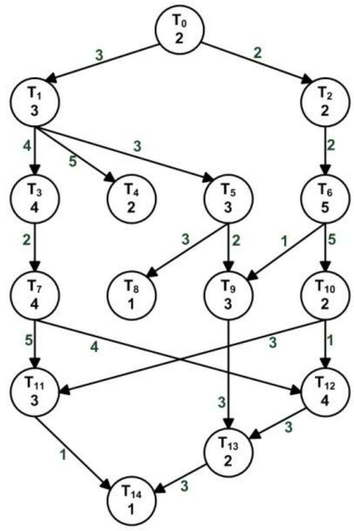

Fig. 1. A sample DAG representing a parallel application with 15 tasks

D. Problem formulation

Given a parallel application G consists of n precedence-constrained tasks and a number of processing units find a schedule which minimizes overall energy consumption multiprocessors.

IV. The Proposed Algorithm

In this section, we present an Energy Aware Edge Priority Scheduling (EAEPS) algorithm for multiprocessor environments. The proposed algorithm is an energy aware version of our EPS (Edge Priority Scheduling) algorithm [3] which is a clustering-based scheduling algorithm and focuses on makespan minimization only. The classical clustering-based scheduling algorithm executes the following steps: (1) clustering of tasks is performed by zeroing the edges; (2) mapping of clusters to appropriate processing units; (3) scheduling of the tasks. The classical clustering-based scheduling algorithm minimizes the makespan whereas our proposed algorithm reduces power consumption by exploiting DVFS technique on non-critical tasks. We are using the concept of voltage scaling for non-critical tasks from [2].

In classical clustering-based algorithm and PATC algorithm, edges are selected for zeroing on the basis of their communication costs while in this work we define a priority function for the edges as follows:

$$p_{i,j} = \frac{c_{i,j}}{x_i + x_j} \quad (9)$$

where, $c_{i,j}$ is the communication cost of the edge $e_{i,j}$ between task T_i and T_j and x_i and x_j are the execution cost of the task T_i and T_j respectively.

Algorithm 1 The EAEPS algorithm

---

1. **begin**
2. initially each task of a parallel application forms a distinct cluster
3. calculate initial energy consumption
4. calculate priority of edges
5. sort all edges in non-increasing order by their priorities and make a list
6. **repeat**
7.   **for** each edge in the sorted list **do**
8.   zeroing an edge if energy consumption decreases
9.   when two clusters are grouped, the order among tasks is decided by comparing their bottom-levels with each other
10.   **if** bottom-level of one task is equal to the bottom-level of other task **then**
11.   both tasks are ordered according to their topological-order in the cluster
12.   **end if**
13.   update energy consumption
14.   **break**
15.   **end for**
16.   remove an edge from sorted list by which energy consumption reduces
17. **until** energy consumption decreases
18. **end**

As shown in Algorithm 1, the EAEPS algorithm initially assigns each task to distinct cluster and determines energy consumption. After that, the algorithm computes priority of each edge and sort edges in non-increasing order of their priorities, the EAEPS algorithm repeatedly groups tasks by zeroing the edges with high priority if total energy consumption is not increased. When two tasks or clusters are grouped, the order among tasks is decided by their bottom levels. The bottom level of a task T_i is the longest path from T_i to the end task and is computed as follows:

$$bl_i = x_i + \max_{j \in predecessor(i)}\{bl_j + c_{i,j}\} \quad (10)$$

In EAEPS algorithm, the priority function tries to group two tasks from different clusters that have heavy communication cost between them w.r.t their execution costs.

V. Experimental Results

In this section, we provide a simulation study on the proposed EAEPS algorithm. We considered some benchmark task graphs given by Davidovic and Crainic [20]. Table I shows the operating frequencies and supply voltages used in this work that is taken from [2]. The performance is measured in terms of percentage of energy saving that is defined as follows:

$$\mathcal{E} = \frac{\mathcal{E}_{\max} - \mathcal{E}_{\text{algo}}}{\mathcal{E}_{\max}} \quad (11)$$

where $\mathcal{E}_{\max}$ is the energy consumption when all tasks executed at the highest frequency, and $\mathcal{E}_{\text{algo}}$ is the energy consumption when applying a particular algorithm.

TABLE I. FREQUENCY AND SUPPLY VOLTAGE USED IN THIS WORK

Frequency (GHz)	Supply voltage (V)
0.8	0.90
1.0	1.00
1.2	1.05
1.4	1.10
1.6	1.15
1.8	1.20

Table II gives a comparison of our proposed algorithm with other energy aware scheduling algorithms in terms of percentage of maximum energy saving. The EAEPS can achieve up to 31.48 % energy saving for selected benchmark random task graphs used in the simulation. Here, the algorithms EADUS and TEBUS give least energy saving among all compared algorithms as these algorithms are duplication-based and do not use DVFS to minimize energy consumption. The EAEPS gives more energy saving as it minimizes energy consumption during the communication phase. It also reduces energy consumption when a processing unit is idle. The EAEPS algorithm tries to choose edge for zeroing that involves more communication cost and reduces energy consumption.

TABLE II. COMPARISON OF THE PROPOSED ALGORITHM WITH OTHER EXISTING ALGORITHMS

Energy aware scheduling algorithms	Maximum energy saving (%)
EADUS & TEBUS [8]	11.43
Energy reduction algorithm [12]	18.35
ECS [9]	29.47
PATC [2]	30.52
EAEPS	31.48

VI. Conclusions

We have proposed a power aware clustering-based task scheduling algorithm that makes use of priority function for edge zeroing, and we called it Energy Aware Edge Priority Scheduling (EAEPS) algorithm, for the problem of scheduling in multiprocessor environments. The proposed algorithm exploited DVFS technique to reduce the power consumption.

The performance of the EAEPS algorithm is examined by performing a simulation study for some selected benchmark random task graphs. The experimental results show that the EAEPS algorithm achieves more energy saving that other compared energy aware scheduling algorithms. Future work includes the study and deployment of the proposed algorithm in some real-world applications like Gaussian elimination, and Fast Fourier Transform.

REFERENCES

[1] J. Jiang, Y. Lin, G. Xie, L. Fu, and J. Yang, "Time and energy optimization algorithms for the static scheduling of multiple workflows in heterogeneous computing system," Journal of Grid Computing, vol. 15, no. 4, pp. 435-456, 2017.

[2] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C-Z Xu, et al., "Energy-aware parallel task scheduling in a cluster," Future Generation Computer Systems, vol. 29, no. 7, pp. 1661-1670, 2013.

[3] A. K. Maurya, and A. K. Tripathi, "An edge priority-based clustering algorithm for multiprocessor environments," Concurrency and Computation: Practice and Experience.

[4] O. Sinnen, Task Scheduling for Parallel Systems, John Wiley & Sons; 2007.

[5] A. K. Maurya, and A. K. Tripathi, "Performance comparison of HEFT, Lookahead, CEFT and PEFT scheduling algorithms for heterogeneous computing systems," In Proceedings of the Seventh International Conference on Computer and Communication Technology 2017 (ICCCT'17), ACM, pp.128–132, 2017.

[6] A. K. Maurya, and A. K. Tripathi, "On benchmarking task scheduling algorithms for heterogeneous computing systems," The Journal of Supercomputing, 2018.

[7] G. Terzopoulos, and H. D. Karatza, "Power-aware Bag-of-Tasks scheduling on heterogeneous platforms," Cluster Computing, vol. 19, no. 2, pp. 615-631, 2016.

[8] Z. Zong, A. Manzanares, X. Ruan, and X. Qin, "EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters," IEEE Transactions on Computers, vol. 60, no. 3, pp.360-374, 2011.

[9] Y. C. Lee, and A. Y. Zomaya, "Minimizing energy consumption for precedence-constrained applications using dynamic voltage scaling," 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009 (CCGRID'09), IEEE, 2009.

[10] Y. Hu, C. Liu, K. Li, X. Chen, and K. Li, "Slack allocation algorithm for energy minimization in cluster systems," Future Generation Computer Systems, vol. 74, pp.119-131, 2017.

[11] G. Aupy, A. Benoit, and Y. Robert, "Energy-aware scheduling under reliability and makespan constraints," 19th International Conference on High Performance Computing (HiPC 2012), IEEE, 2012.

[12] H. Kimura, M. Sato, M., Y. Hotta, T. Boku, and D. Takahashi, "Emprical study on reducing energy of parallel programs using slack reclamation by dvfs in a power-scalable high performance cluster," In IEEE International Conference on Cluster Computing, IEEE, 2006.

[13] N. Kaur, S. Bansal, and R. K. Bansal, "Duplication-controlled static energy-efficient scheduling on multiprocessor computing system," Concurrency and Computation: Practice and Experience, vol. 29, no. 12, 2017.

[14] J. Mei, K. Li, and K. Li, "Energy-aware task scheduling in heterogeneous computing environments," Cluster Computing, vol. 17, no. 2, pp. 537-550, 2014.

[15] X. Tang, and W. Tan, "Energy-Efficient Reliability-Aware Scheduling Algorithm on Heterogeneous Systems," Scientific Programming 2016, p.14, 2016.

[16] M. Sharifi, Sa. Shahrivari, and H. Salimi, "PASTA: a power-aware solution to scheduling of precedence-constrained tasks on heterogeneous computing resources," Computing, vol. 95, no. 1, pp. 67-88, 2013.

[17] R. Ge, X. Feng, K. W. Cameron, "Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters," in: SC, 2005, p. 34.

[18] M. Y. Lim, V. W. Freeh and D. K. Lowenthal. "Adaptive, transparent frequency and voltage scaling of communication phases in MPI programs," In proceedings of the ACM/IEEE SC 2006 conference, IEEE, pp. 14-14. 2006.

[19] D. Kadamuddi, J.J. P. Tsai, "Clustering algorithm for parallelizing software systems in multiprocessors environment,", IEEE Transactions on Software Engineering, vol. 26, no. 4, pp. 340-361, 2000.

[20] T. Davidovic, T. G. Crainic, "Benchmark-problem instances for static scheduling of task graphs with communication delays on homogeneous multiprocessor systems," Computers & Operations Research, vol. 33, no. 8, pp. 2155–2177, 2006.