Volume 6, Issue 4, April 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Survey on Energy Aware Scheduling Algorithms of DAG Applications for Multiprocessor Computing System

Raman Bhinder*
Dept. of CSE, UIET, Panjab University, Chandigarh, India

Nirmal Kaur
Assistant Prof., Dept. of CSE, UIET, Panjab University Chandigarh, India

Abstract— Energy consumption is emerging as an important metric for computing devices these days with which devices are judged along with the performance. Algorithmic approach for energy management in computing devices is however appeared more effective than hardware or system based approaches. The deployment of power management through scheduling algorithms using Dynamic voltage and frequency scaling (DVFS) and Dynamic power management (DPM) techniques is widely held in computing devices. The use of energy saving techniques however affects the schedule of the tasks. So in this article, the need of energy management is discussed and survey on different heuristics of the scheduling algorithms for parallel applications is performed. The algorithms studied here; aim at minimizing the energy consumption using DVFS and DPM techniques on multiprocessor computing systems.

Keywords— Energy management, Scheduling, Multiprocessor systems, DVFS, DPM.

I. INTRODUCTION

The multiprocessor computing system is composed of two or more central processing units (CPUs) that simultaneously execute the tasks of a parallel application. The reason behind the fame of multiprocessor system in recent years is; they provide high performance at the low cost of computation. These days applications require high parallelism and fast speed viz. image modelling, banking systems, space engineering, internet browsing, gaming, video and audio streaming. The multiprocessor system is a true model of parallelism and satisfies the requirement of fast execution for parallel application. But unlike uniprocessor system where performance and throughput are major concerns, the designers of multiprocessor architecture have to deal with additional challenges like load balancing, process synchronization and intercommunication of processors. Besides high and versatile performance, the energy consumption has also become a leading constraint for multiprocessor systems. The early dissipation of battery due to high computation is the growing problem with the portable devices.

Over the past years, the performance of a processor has remained the prime concern, but now energy consumption has become another challenging parameter of scheduling paradigm. The solution to all these challenges can be attained with the help of judicious energy aware scheduling algorithms. The growing energy demand is the one of the critical problems facing by the world. According to survey, Somvat et al. [7] the computing devices consume 3% of the global world electricity consumption. The number of users of computer, laptops and mobile phones is also increasing rapidly day by day. For the motherboard of these portable devices, processors are found to be the most energy consuming component following the memory units and other components [4]. The energy savings in computing devices also contribute in several benefits along with above stated problem like less electricity bills, reduce heat dissipation, reduce wear of component, more reliability, reduce maintenance cost and reduce greenhouse emission. Therefore, if multiprocessor system if consumes less energy, then this leads to direct or indirect monetary and environmental benefits [2]. All these factors emphasize the importance of energy saving in computing devices. With this regard, a survey is contributed towards the various energy-aware scheduling algorithms proposed for parallel applications on multiprocessor system.

The survey is organized as follows: In Section 2 some related surveys present in literature. After that, in Section 3 we provide important models that are adopted throughout the survey. Section 4 introduces heuristics practices for scheduling: list, duplication and clustering. It also overview the existing DVFS and DPM based approaches used by researchers to reduce energy for multiprocessor systems. Section 5 discusses the various existing energy aware scheduling algorithms of directed acyclic graph (DAG) applications on multiprocessor system in detail. Section 6 presents the conclusion of the article with a discussion.

II. RELATED SURVEY

Time to time, researchers has come up with the different surveys for energy reduction techniques in processors. Venkatachalam and Franz [1] surveyed, power reduction techniques for uniprocessor. The authors investigated different techniques at hardware, software level and technologies used at the commercial level for power management. Zhuravlev et al. [2] studied the software level techniques at run time to reduce the CPU energy. The authors discussed the three mechanisms allow to manage energy in CPU: DPM/DVFS, thermal and asymmetric multiprocessor system. Orgerie et al. [3] discussed, power reduction techniques like virtualization, shut down techniques and optimizing the number of resources used to reduce energy consumption for distributed networks. Many papers related to the techniques for energy

management for computing nodes and network resources are surveyed. Above discussed surveys focuses more on the techniques rather the algorithms.

Albers [4] investigated, energy efficient scheduling algorithms for both networks and multiprocessor system. Recently Bambagini et al. [5] surveyed, energy aware scheduling algorithms using DVFS and DPM techniques for real time system on uniprocessor and multiprocessor system. Also, Gerards et al. [6] surveyed, offline scheduling algorithms for deadline constraint applications to reduce energy. These surveys do focus on the algorithms but, less or no focus is given to the energy aware scheduling algorithms for parallel applications.

Here, in this paper a survey is presented that focuses on energy aware scheduling algorithms for DAG applications. Fast Fourier Transform, LU decomposition, Gauss–Jordan elimination and the Gaussian Elimination are some examples of real parallel applications.

III. MODELS

In general, the scheduling can be performed in two ways: offline scheduling and online scheduling. The offline scheduling is performed at compile time, knowing the behavior of upcoming tasks and online scheduling is performed at run time where the behavior of the task may or may not be known. Scheduling is to map the tasks to the processors in order to minimize the completion time of application called makespan. Different papers have different modelling assumptions. This section presents the formal description and the unifying notation for task, processor, and energy models.

A. Task Model

To run a parallel application, it is decomposed into subtasks. The dependencies among subtasks can be represented in the form of a directed acyclic graph (DAG) as shown in fig. 1. To execute the application correctly, the order of tasks in which they execute should be according to the precedence constraints. The DAG is set of four elements $G=(V,E,T,C)$. $V=\{n_i \mid n_i$ is an ordered task, $i=1,2,3...v\}$ represents the vertex set. $E = \{e_{ij} \mid e_{ij}$ is the edge from n_i to $n_j\}$ represents the edge set. $T = \{t_{ij} \mid t_{ij}$ is the computation time of n_i, on processor p_j $i=1,2,3...v$ and $j = 1,2,... p$ $\}$ is the task computation time set and $C= \{c_{ij} \mid c_{ij}$ is the communication time from task n_i to n_j, n_i is predecessor of $n_j\}$ is the communication time set. The $succ(n_i)$ denote the set of successors of task n_i and $pred(n_i)$ the set of predecessors of task n_i. An entry task n_{entry} has no predecessors while exit task n_{exit} has no successors. The *t_level* is top level; it is the length of a longest path from an entry task n_{entry} to task n_i that is the sum of all the tasks computation time and communication time along the path followed. Similarly *b_level* is bottom level, the length of a longest path from a task n_i to exit task n_{exit}.

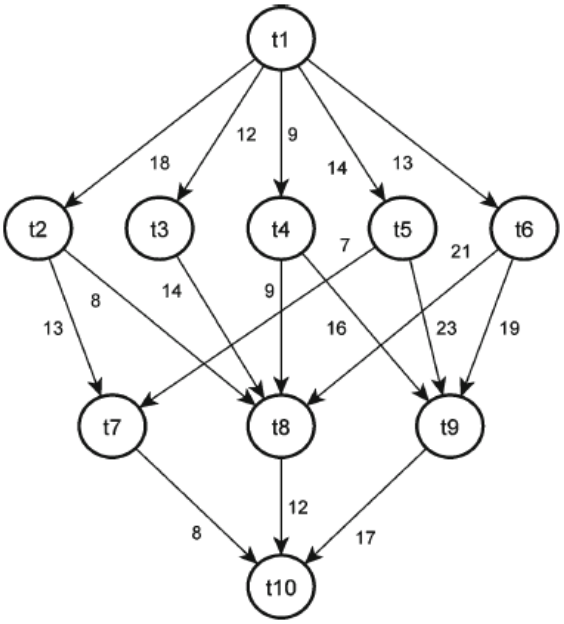

Fig. 1 Example of DAG with 10 tasks

B. Machine Model

The system has independent and fully connected computing resources (CRs). Each of the *CR* has a processor and a local memory. The CRs does not share memory and the communication between the processors is done through message transfer. Let $P=\{p_i \mid p_i$ is the processor, $i = 1,2,... p\}$. The system may be homogeneous or heterogeneous. In homogenous system the tasks have same computation time for all processors; however in heterogeneous system the tasks have different computation time for different processors. The processor can perform communication and computation simultaneously and task preemption is not allowed. The earliest start time $EST(n_i,p_j)$ and earliest finish time $EFT(n_i,p_j)$ [11] of task n_i on processor p_j are two important attributes and expressed as:

$$EST(v_i, p_j) = max \{avail[j], max (AFT(v_t+c_{ti}))\}$$
$$where\ v_t \in pred(v_i)$$
$$EFT(n_i,p_j)= t_{ij} + EST(n_i,p_j) \quad (1)$$

where *avail[j]* is the earliest available time at which processor p_j can be available for taking the next task after completing all its execution. The earliest start time *vt* on processor *pj* is equal to the max of the *avail[j]* and the actual start time actual finish time *AFT(vt)* of task *vt* respectively. For the entry task v_{entry} $EST(v_{entry}, p_j) = 0$ and for the other, the *EST* and *EFT* values are computed according to the equation (1). To compute the *EFT* of a task *vi*., all of the immediate predecessor tasks of *vi* require to be have been scheduled. The mapping relies on arrival time, computation time and completion time of task. Let $F = \{f_i \mid f_i$ is the finish time of processor, $i = 1,2,\ldots p\}$ represent set of finish times of all CRs when a CR finish all its computations and communications, then makespan can be expressed as:

$$makespan = max\{ f : f \in F\} \quad (2)$$

C. Energy Model

The complementary metal–oxide–semiconductor CMOS based processor has two components responsible for its power dissipation: dynamic power and static power as:

$$P_{processor} = P_{dynamic} + P_{leak} \quad (3)$$

Dynamic power $P_{dynamic}$, is the power dissipated due to loading and unloading of the capacitors, gate switching, clock frequency and supplied voltage. It is defined as the function of clock frequency *f* and the supply voltage *V* as follows:

$$P_{dynamic} = \alpha C_L V^2 f \quad (4)$$

where C_L is load capacitance, α is probability of gate switching. Static power P_{leak}, is the power that flows through transistors even though the transistors are not in use or turned off and is expressed as:

$$P_{leak} = V I_{leak} \quad (5)$$

where I_{leak} is the leakage current. Researchers in past years have focused more on to reduce dynamic energy consumption, but due to chip miniaturization, improved idle states, fast memory cache and increase in number of transistors, the static energy dissipation has dominated dynamic power dissipation [8]. In order to get more energy savings it is very important also to focus on the leakage energy component.

IV. BASIC ENERGY EFFICIENT TECHNIQUES AND SCHEDULING HEURISTICS

Several scheduling algorithms are mentioned in literature for multiprocessor systems. In general, these algorithms fall in either of the heuristic based or guided random search based categories. Further, many types of algorithms exist in each of the categories as shown in fig. 2. The dynamic power management (DPM) and dynamic voltage and frequency scaling (DVFS) are two basic techniques used for energy management. This section gives a brief introduction of the different algorithms and the energy management techniques.

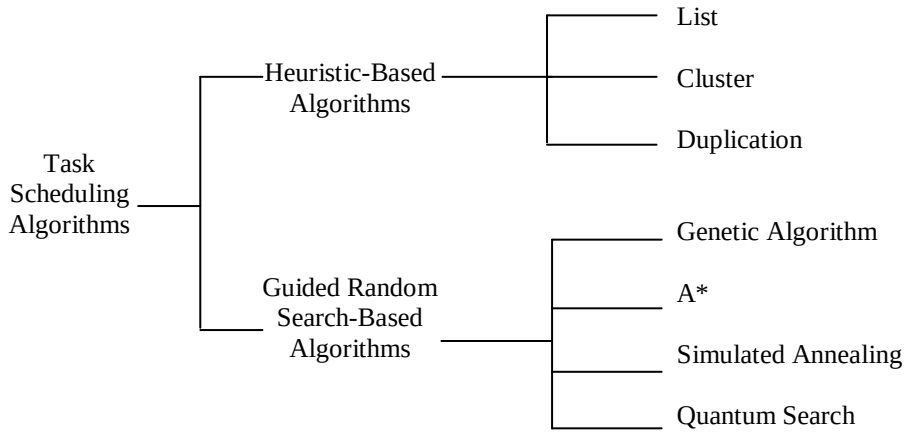

Fig. 2. Types of scheduling algorithms

A. Scheduling Algorithm Approaches

1) Guided random search: This technique is inspired by biological evolution to search the problem space and is getting popularized in parallel computing also. Genetic algorithm, simulated annealing, Quantum search, Tabu search and A* algorithm fall in this category. This type of algorithms develops the solution step by step based on the information available. The output schedule is of better quality than produced by algorithms based on heuristic approach. These algorithm, however cannot work well for large problems as the scheduled time is higher due to poor search efficiency.

2) Heuristic based: The heuristic based algorithms are approximate algorithms not the accurate. This means, they produce a schedule that may not be the optimal, but near to the best one. These algorithms have a function on the basis of which, the schedule is generated. List, cluster and duplication are heuristic based scheduling algorithms. These algorithms produce a quick schedule, if compare to guided random search algorithms. The quality of the schedule depends upon the efficiency of the function used. Different researchers have also used a hybrid of heuristic and guided random search algorithms for scheduling.

List based algorithms involve assigning rank to the tasks and then mapping the tasks to the processors. Cluster based algorithms involve grouping of tasks in clusters such that intercommunication of the processors is less. All tasks of a cluster get executed on the same processor. Duplication based algorithms duplicate the predecessor of the task on the processor where its child is assigned. This can eliminate unnecessary communication delay among processors, thereby

reduces overall communication overheads. There exist many hybrid approaches like list-duplication and cluster-duplication or a blend of all the three types of heuristic. In this paper the focus is given to the available heuristic based energy aware scheduling algorithms.

B. Power Management Techniques in Scheduling

1) DPM: It is the operating system level technique that dynamically reconfigures an electronic system and reduces the number of active components. For example, in laptops a timeout policy is to turns off a component after a fixed inactivity to save the energy. The Advanced Configuration and Power Interface (ACPI) is an industry standard for the interfaces that enables the dynamic power management. When a processor turns off, it takes few time units to turn off. On the other hand to retain the active state from idle it will require some energy units. This energy penalty will have to pay for every idle to active transition of the processor. A processor cannot be turned off for all the idle periods due to transition constraints. Let r joule/sec is the power consumption rate at active state. Let β joule is the energy penalty for idle to active transition. Then an idle period $T \geq \beta / r$ can be beneficial, which means a processor cannot be turned off if idle period is less than T [2]. There may be different power down state like standby, suspend and sleep that offer different energy saving levels. Let working power dissipation (WPD) is the power consumption rate of processor when it is active and idle power dissipation (IPD) when processor is idle, then total energy consumption of processor can be expressed as [14]:

$$E_{processor} = WPD \times t_{active} + IPD \times t_{idle} \quad (6)$$

where t_{active} is the time for which the processor remain active and t_{idle} is the time for which the processor was idle.

2) DVFS/DVS: It requires special hardware build with CMOS technology that enables the processor to run at different speed by scaling the voltage. The energy consumed by the processor is directly proportional to the speed of the processor. Hence, more the processor run at low speed less will be the energy consumed. The challenge lies behind implementing this technique is to calculate the voltage at which a task will execute. The *slack time* of the task is idle period that reside between the finish time of that task and start time of the next task. The task is allowed to finish late and execute at lower voltage if there exist any slack time for the task. The total energy consumption of processor can be expressed as the summation of energy consumption at different voltage levels as follows [13]:

$$E_{processor} = \sum_{m \in M} V_m^2 \times f \times t_m \quad (7)$$

where t_m is the time for which the processor run at V_m voltage level and M is the set of different voltage levels at which processor can be execute.

3) DVFS with sleep states: Researchers have come up with the approach to use both the DPM and DVFS together. This hybrid approach helps to reduce dynamic as well as the static energy consumption of the processor. Awan et al. [26] used this hybrid approach for heterogeneous real time system over the sporadic tasks. The authors first analyzed the behaviour of each task at different frequencies on each processor and chose the best set of frequency. The sleep state could be achieved by migrating tasks from core to another core or increasing the frequency of a core. The improvement achieved after selecting frequency is matched against the cost of transferring those tasks to another core. The one that reduced energy consumption is selected.

V. ENERGY AWARE SCHEDULING ALGORITHMS

There has been tremendous work done by researcher to solve the scheduling problem for multiprocessor system on list, cluster and duplication scheduling heuristics. This section discusses existing algorithms to reduce the energy consumption additionally schedule the parallel tasks to processors.

A. Energy Aware Scheduling in List-Based Algorithms

Gruian and Kuchcinski [9] suggested low-energy scheduling (LeNES) algorithm with a priority function that is based on energy gain/loss. If a task is associated with positive priority the task could be scheduled at the very moment. If it is associated with negative priority the task should be scheduled later. The priority is recalculated and the priority function is tuned at every step of scheduling. The authors used Enhanced task graph (ETG) instead of simple DAG model in which a task is divided into a pair of start and end node. The tasks are then assigned to the homogenous processors and task graph is tried to execute as fast as possible. The slack time is used for scaling the task in order to reduce the energy.

De Langen and Juurlink [10] proposed leakage-aware multiprocessor scheduling (LAMPS) algorithm that focused on static energy as well as the dynamic energy consumption. To reduce static energy consumption authors came with a strategy to calculate sufficient number of processors using binary search. The number of processors should be able to execute the task graph on time, that is the makespan should be less than or equal to earliest deadline first (EDF) scheduler. Further, after assigning the task to homogenous system the scaling of tasks is performed to reduce dynamic energy.

Baskiyar and Palli [11] discussed low power heterogeneous makespan (LPHM) algorithm that used the heterogeneous earliest finish time (HEFT) to calculate priority of each task and assigned to the processors. After assignment the tasks having the slack time are scaled. But before scaling the tasks are merged wherever possible so that these tasks can operate at same down voltage. This reduced the number of scaling operations, the runtime overhead and adverse effect on performance caused due to scaling the voltage and frequency. It therefore also helped to improve system reliability.

Shekar and Izadi [12] recommended energy-dynamic level scheduling (EDLS) algorithm. Dynamic level scheduling is a popular list scheduling algorithm that traverses the graph according to the static level of tasks. The priority or Dynamic levels (DL) of all those tasks are calculated over all processors having highest static level. The tasks are scheduled

according to the processor with highest DL. The process is then repeated for the unscheduled tasks. In EDLS authors used the energy based priority function and calculated energy dynamic level (EDL) of the task according to their static level. The priority function is a function of execution time and energy consumption of task on each processor. The EDL gave the best possible match of a task-processor pair according to its execution and energy consumption.

Baskiyar and Abdel-Kader [13] proposed energy aware DAG scheduling (EADAGS) algorithm for heterogeneous system that used decisive path scheduling (DPS) in which the task are traversed in is *t_level* order and the task are mapped to processor which give minimum EFT. The slack time can be used to scale the task and run the task at lower voltage only if does not increase the makespan. If it increases the makespan, then only the idle period is scaled and task is executed at original voltage.

Sharifi et al. [14] advised a power-aware scheduling of parallel tasks (PASTA) algorithm for the on heterogeneous systems. The approach used in the paper can be considered as a true parallelized approach. At first, the best subset of processors equal to the number of tasks that can run parallel from the available processors is selected. For this, the score for each processor is computed and then processors are sorted in increasing order of their score. Then a best subset of processor is evaluated such that schedule length for the selected subset is equal to schedule length of HEFT. The tasks are traversed in order of *b_level*. The priority function is used to calculate the score of each task which is the multiplication of EFT and energy consumption for all the selected processors. The task is assigned to the processor that produced a minimum score. The authors kept their focus on reducing the energy by judicious scheduling however, the makespan is increased compared to HEFT.

B. Energy Aware Scheduling in Duplication-Based Algorithms

Zong et al. [15] suggested energy-aware duplication scheduling EADUS and time-energy duplication scheduling algorithm TEBUS algorithms for homogenous system. The algorithms are based on the fact that duplication of task can lead to consumption of more energy so; the duplication should be performed only for those tasks that do not increase the energy more than the given threshold. To implement this idea authors in EADUS, grouped highly communicable tasks and assign the group to same processor and performed conditional duplication to reduce interprocessor communication. The other algorithm TEBUS deal with trade-offs for energy saving and performance. The condition for duplication is to maintain the cost ration at a low level that is energy savings to schedule length.

Zong et al. [16] suggested energy-aware duplication (EAD) and performance-energy balanced duplication (PEBD) algorithms. In these algorithms DVFS technique is embedded with EADUS and TEBUS to minimize energy consumption of the processor for homogenous system and derived algorithms EAD and PEBD. The authors performed experiments on DVS algorithm and non DVS algorithm and concluded that for the applications that are highly computation intensive DVFS technique performed better than the non DVFS technique.

Li et al. [17] discussed processor reduction optimization (PRO) technique which used only the required number of processors to execute the task rather than using all of the available processors. Moreover, it tried to minimize makespan and maintained the load balancing of the homogeneous system. In this technique an idle slot for the processor is founded that is heavily utilized, this is filled with the tasks of a processor which is less utilized. So, low utilized processors could be switched off permanently. This technique is experimented for the existing duplication scheduling algorithms and showed effective results in energy savings.

Mei et al. [18] investigated energy-aware scheduling minimizing duplication (EAMD) algorithm that also stated the fact that some duplication policies duplicate a task in exceed to minimize the makespan that results high energy consumption and wastage of the resources. The authors dealt this problem by working on a scenario that if any task has two copies; original and duplicated then original copy can be deleted only if the duplicated copy is able to retain the precedence constraint and provide data to all of its children. This technique is for non-DVS heterogeneous system.

Liu et al. [19] proposed adaptive energy-efficient scheduling (AES) algorithm to minimize the duplication and save energy for homogenous system. For this, authors calculated an optimal threshold; depending on the makespan and the power of the processor. The task can be duplicated only up to this threshold. Moreover, the task is duplicated only if it satisfied the duplicating criteria that, it would not increase the energy consumption and retained the performance. The tasks are grouped according to their favourite predecessor and group of task is mapped to any of the available processors. The tasks are allowed to execute at the highest voltage, but DVFS is used to scale the idle times only.

C. Energy Aware Scheduling in Cluster-Based Algorithms

Ma et al. [20] mentioned energy-efficient scheduling on a task dependent graph using DVS-unable (ESTD) algorithm for homogenous system and used the hybrid approach of clustering and duplication. The algorithm is useful for the data-intensive application preoccupied by DAG. The author used CASS [21] to obtain the grouping of task and task ordering then it applies the DPM method to further decrease static energy consumption, then task duplication is used in a controlled way to minimize energy consumption. ESTD reduced data transmission time, makespan, leakage energy and total system energy.

Fangfa et al. [22] studied objective-flexible clustering algorithm (OFCA) algorithm in which the authors took the advantage of lineal clustering. In this the tasks that follow a key path are grouped to same cluster and help to implement the full parallelism. This approach use combination of the cluster based duplication method to deal with the energy consumption and time taken in homogenous system. To reduce the inter cluster communication the duplication strategy is used. Duplication is performed according to the influence of time and energy of task over a processor on which it will be duplicated. Also the clusters combination can be applied if it does not increase the makespan.

Liu et al. [23] suggested energy efficient clustering based scheduling algorithm (ECSTD) for parallel tasks on homogeneous DVS-enabled system. The algorithm first calculates the *e _value*. It is the lower bound on start time of each task. The tasks are traversed in topological order and clusters are produced depending on these *e_values.* The extra clusters are deleted depending upon the calculated energy consumption in inter-cluster communication. For idle periods the DVS technology is used. The solution is effective to maintain the balance between the performance and the energy consumption.

Wang et al. [24] proposed power aware task clustering (PATC) and solved the trade-off between energy consumption and makespan using green service level agreement (GSLA) for homogenous system. The authors considered every task as a cluster at first. The communication edges are sorted in descending order of their execution time and marked unexamined. Now the high communication cost edge is zeroed and the two clusters are merged if energy consumption does not increase. The process is repeated for all the unexamined edges. After the clusters are obtained, the tasks are sorted in each cluster by *b_level.* Then clusters are scheduled on processors. To minimize energy, all the communication phases and idle phases are executed at lowest voltage. If the performance has to be maintained it scaled only the non-critical jobs, else if performance can be sacrificed then critical jobs are also scaled and the makespan is increased.

VI. CONCLUSIONS

In this survey, many scheduling algorithms are discussed that reduce energy consumption of multiprocessors by speed scaling, turning the processor off or obtaining a schedule that consume less energy. For instance DVFS reduces the dynamic power consumption instead of reducing the total energy consumption. Also, due to the advancement of technology the DVFS is becoming unable to provide enough energy savings. DVFS result adversely on performance of the processor, whenever the frequency is reduced. Most of the algorithms studied in this survey use DVFS technique and don't consider these lags. Secondly, only a few algorithms are available those focus on DPM technique and reduce static energy consumption that is dominating dynamic energy consumption. For DPM technique, it is necessary to consider the energy penalty to be paid, when a processor is transitioned from a sleep state to the active state, which is not taken account in the algorithms reviewed. Thirdly, some algorithms tried to schedule tasks depending upon on an energy function in order to reduce the system energy, just by considering power efficiency and selecting an efficient processor and while developing the schedule and remained independent of above stated technologies.

From the three heuristic list, duplication and clustering, it is found that hybrid approach turned to be more effective, if compared to the isolated approach. Most of the related work available, have considered a homogeneous system. S. Bansal et al. [25] stated that make-span and energy consumption tend to reduce when the few processors from homogeneous system are replaced by the heterogeneous processor with higher clock speeds. So, heterogeneous systems can be helpful to reduce the system energy. So, it is worth to explore more methods to reduce static power consumption of processor along with the dynamic power consumption and taking the power of the combined approach of the various existing heuristic to solve the problem of energy consumption in heterogeneous systems.

REFERENCES

[1] Venkatachalam, V., & Franz, M. "Power reduction techniques for microprocessor systems," ACM Computing Surveys (CSUR), 37(3), 195-237, 2005.

[2] Albers, S. "Energy-efficient algorithms," Communications of the ACM, 53(5), 86-96, 2010.

[3] Zhuravlev, S., Saez, J. C., Blagodurov, S., Fedorova, A., & Prieto, M. "Survey of energy-cognizant scheduling techniques. Parallel and Distributed Systems," IEEE Transactions on, 24(7), 1447-1464, 2013.

[4] Orgerie, A. C., Assuncao, M. D. D., & Lefevre, L. "A survey on techniques for improving the energy efficiency of large-scale distributed systems," ACM Computing Surveys (CSUR), 46(4), 47, 2014.

[5] Bambagini, M., Marinoni, M., Aydin, H., & Buttazzo, G, "Energy-Aware Scheduling for Real-Time Systems: A Survey," ACM Transactions on Embedded Computing Systems (TECS), 15(1), 7, 2016.

[6] Gerards, M. E., Hurink, J. L., & Hölzenspies, P. K. "A survey of offline algorithms for energy minimization under deadline constraints," Journal of Scheduling, 1-17, 2016.

[7] Somavat, P., Jadhav, S., & Namboodiri, V. "Accounting for the energy consumption of personal computing including portable devices," In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (pp. 141-149). ACM, 2010.

[8] Sueur, E., & Heiser, G. "Dynamic voltage and frequency scaling: The laws of diminishing returns," In Proceedings of the international conference on Power aware computing and systems (pp. 1-8). USENIX Association, 2010.

[9] Gruian, F., & Kuchcinski, K. "LEneS: task scheduling for low-energy systems using variable supply voltage processors," In Proceedings of the Asia and South Pacific Design Automation Conference (pp. 449-455). ACM, 2001.

[10] De Langen, P., & Juurlink, B. "Leakage-aware multiprocessor scheduling for low power," In Parallel and Distributed Processing Symposium, IPDPS 20th International (pp. 8-pp). IEEE, 2006.

[11] Baskiyar, S., & Palli, K. K "Low power scheduling of dags to minimize finish times," In High Performance Computing-HiPC 2006 (pp. 353-362). Springer Berlin Heidelberg, 2006.

[12] Shekar, V., & Izadi, B. "Energy aware scheduling for DAG structured applications on heterogeneous and DVS enabled processors," In Green Computing Conference, (pp. 495-502). IEEE, 2010.

[13] Baskiyar, S., & Abdel-Kader, R. "Energy aware DAG scheduling on heterogeneous systems," Cluster Computing, 13(4), 373-383, 2010.

[14] Sharifi, M., Shahrivari, S., & Salimi, H. "PASTA: a power-aware solution to scheduling of precedence-constrained tasks on heterogeneous computing resources," Computing, 95(1), 67-88, 2013.

[15] Zong, Z., Manzanares, A., Stinar, B., & Qin, X. "Energy-aware duplication strategies for scheduling precedence-constrained parallel tasks on clusters," In Cluster Computing, IEEE International Conference on (pp. 1-8). IEEE, 2006.

[16] Zong, Z., Manzanares, A., Ruan, X., & Qin, X. "EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters," Computers, IEEE Transactions on, 60(3), 360-374, 2011.

[17] Li, X., Zhao, Y., Li, Y., Ju, L., & Jia, Z. "An Improved Energy-Efficient Scheduling for Precedence Constrained Tasks in Multiprocessor Clusters," In Algorithms and Architectures for Parallel Processing (pp. 323-337). Springer International Publishing, 2014.

[18] Mei, J., Li, K., & Li, K. "Energy-aware task scheduling in heterogeneous computing environments," Cluster Computing, 17(2), 537-550, 2014.

[19] Liu, W., Du, W., Chen, J., Wang, W., & Zeng, G. "Adaptive energy-efficient scheduling algorithm for parallel tasks on homogeneous clusters," Journal of Network and Computer Applications, 41, 101-113, 2014.

[20] Ma, Y., Gong, B., & Zou, L. " Energy-efficient scheduling algorithm of task dependent graph on DVS-Unable cluster system," In Grid Computing, 10th IEEE/ACM International Conference on (pp. 217-224). IEEE, 2009.

[21] Liou, J. C., & Palis, M. A. "An efficient task clustering heuristic for scheduling dags on multiprocessors. In Workshop on Resource Management," Symposium on Parallel and Distributed Processing (pp. 152-156), 1996.

[22] Fangfa, F., Yuxin, B., Xinaan, H., Jinxiang, W., Minyan, Y., & Jia, Z. "An objective-flexible clustering algorithm for task mapping and scheduling on cluster-based noc," In Laser Physics and Laser Technologies (RCSLPLT) and Academic Symposium on Optoelectronics Technology (ASOT), (pp. 369-373). IEEE, 2010.

[23] Liu, W., Duan, Y., & Du, W. "An energy efficient clustering-based scheduling algorithm for parallel tasks on homogeneous DVS-enabled clusters," In Computer Supported Cooperative Work in Design (CSCWD), IEEE 16th International Conference on (pp. 575-582). IEEE, 2012.

[24] Wang, L., Khan, S. U., Chen, D., Kołodziej, J., Ranjan, R., Xu, C. Z., & Zomaya, A. "Energy-aware parallel task scheduling in a cluster," Future Generation Computer Systems, 29(7), 1661-1670, 2013.

[25] S., Bansal, K., & Bansal, R. K. "Energy consumption on heterogeneous computing platforms," In Parallel, Distributed and Grid Computing (PDGC), (pp. 355-359). IEEE, 2014.

[26] Awan, M. A., Yomsi, P. M., Nelissen, G., & Petters, S. M. "Energy-aware task mapping onto heterogeneous platforms using DVFS and sleep states," Real-Time Systems, Springer 1-36, 2015.

