Energy Aware Scheduling for Real-time Multi-Core Systems

Hadeer A. Hassan, Sameh A.Salem, EL-Sayed M. Saad
Department of Electronics, Communications and Computer Engineering
Faculty of Engineering – Helwan University, Cairo, Egypt.
{Hadeer_ahmed@h-eng.helwan.edu.eg - Sameh_Salem@h-eng.helwan.edu.eg - Elsayed012@gmail.com}

Abstract— This paper proposes a new taxonomy for energy aware real time scheduling on multi-core systems. The new taxonomy gives a guideline for choosing the proper energy aware scheduling criterion that meets the excessive requirement of the system. Several factors that influence in evolving the real-time scheduling on multi core architectures are considered. These factors include the execution models (deterministic and probabilistic), the task dependency, the selection of tasks with harmonic nature, the thermal dissipation, the use of reinforcement learning, and the task allocation technique through bin-packing or optimization techniques. Therefore, the future trend towards Internet of Things (IoT) systems with limited resource constraints can be achieved.

Keywords- Real-time scheduling, Multi-core systems, Energy optimization, Dynamic voltage scaling.

I. INTRODUCTION

Recently, it is observed that energy management for real time system plays an important role in the design of embedded systems [1], where saving battery consumption has a crucial role in the future IOT (Internet Of Things). One of the main issues of IOT is the device's power constrain, such as deployed in WSN (Wireless Sensor Network), systems dealing with RFID (Radio Frequency Identification Devices), power grid and others. In these systems the devices have limited power resource so the solution is to minimize the energy consumption to increase the lifetime of the devices.

Two widely used techniques for energy aware real time scheduling are DVFS (Dynamic voltage Frequency Scaling) and DPM (Dynamic Power Management). The power dissipation for processor is divided into static power and dynamic power. DVFS deals with enhancing the performance of system by decreasing the supply voltage leading to decrease the dynamic power and enhances the overall energy dissipation [3]. Taking into account that the decrease in frequency leads to increase in the task's execution time, so the task's time constrains must be taken into consideration [4]. DPM is used to determine a specific point in which the processor is switched into sleep mode to decrease the leakage current [2], leading to decrease in its static power. The choice of the power management points should carefully be chosen to decrease the overall energy consumption. A new hybrid technique between DVFS and DPM approaches benefiting from the advantages of both techniques is appeared.

Real- time systems can use the deterministic or probabilistic approach to represent the task's execution time. Traditionally, almost all papers use the deterministic approach, where a task has a WCET (Worst Case Execution Time) and all task's jobs must finish execution before deadline [47, 6, and 7]. The execution of tasks becomes less predictable, so the deterministic approach leads to assign excess time for task's execution. As a consequence, the processor will be poorly utilized. In contrast, the probabilistic approach utilizes the processors more efficiently by calculating the probability of the task's jobs execution [8, 9, and 10]. Also the timing constrains must be taken in to consideration.

It is a known fact that the scheduling of tasks depends on period relationship between tasks. To be more specific, choosing tasks with harmonic nature (i.e., the periods are integer multiple) is proven to provide the system with better processor's utilization [12, 13, 14, and 15]. The recent challenge is how to determine the tasks with harmonic relationship and their execution time are probabilistic rather than deterministic, which is considered an NP problem. Moreover, tasks can be dependent with precedence constraints or independent. The tasks nature is determined according to the application type. Several literatures are based on independent tasks [8, 17, 18, and 19], while others are based on dependent tasks using DAG (Directed Acyclic Graph) to represents the dependence relationship among different tasks [16].

It is well known that increasing the power leads to rise the operating temperature. Consequently, an increase in the leakage power consumption happens so, the lifetime of processor decreases, and the processor's reliability decreases. Therefore, the thermal dissipation should be considered during design of real time scheduler. As a consequence, thermal energy aware scheduling must be used to enhance the thermal dissipation either by using DVFS or DPM. Many algorithms are used to decrease the energy consumption under a specific temperature threshold. Some use the meta-heuristic search algorithm [25] and other use the linear programming method (ILP) [26] to schedule tasks under thermal constraints. Both require a huge computation. A simpler approach is used, which is the workload balancing approach which assigns tasks among processors with the lower energy under a temperature constrain [17, 24, and 25].

Several introduced researches to enhance the overall energy consumption of the system. Each technique has a different procedure to schedule tasks. Some are based on slack time, deadline, response time, AET execution time and others. Also, there is no algorithm that guarantees to be the optimal solution in all cases. As a consequence, some recent papers teach the scheduler to choose the best approach for current tasks among many existing techniques using the reinforcement learning. As, a result the overall energy consumption of a system is enhanced too much than using individual techniques [18, 27, 28, and 29].

Finally, the problem of partitioning tasks is considered as an NP- problem. So it is transformed into the bin-packing problem or optimization techniques. Many heuristics are based on the variable sized bin packing techniques for multi-core system [14, 17, and 19]. Others are proved to yield better results when using optimization techniques such as, Ant Colony Optimization [35], Genetic [31], Particle Swarm [36], and others. Recently, the researchers use optimization techniques based on hybrid between two existing techniques, such as HGA (Hybrid Genetic Algorithm) [16], and others [32, 33, and 34].

The paper is organized as follows. Section 2 introduces the generic taxonomy for real time energy aware scheduling. Section 3 discusses the various DVFS, DPM, and Hybrid DVFS-DPM algorithms. Section 4 presents one of the recent probabilistic based scheduler. Section 5 presents one of the recent thermal energy based scheduler. Section 6 presents one of the recent reinforcement learning techniques for energy aware scheduling. Section 7 presents one of the recent optimization techniques for energy aware scheduling. Section 8 concludes the entire review.

II. Generic Taxonomy

In this section a new generic taxonomy of energy aware real time scheduling is presented as shown in Fig. 1.

Fig. 1 shows that the energy aware real-time scheduling is divided into four different points. Firstly, the frequency modification point where the energy aware scheduler enhances the overall energy consumption and utilization by decreasing the speed of processor or switch to the sleep mode and this part is discussed in section 3.

Secondly, the allocation techniques point where the scheduler enhances the allocation technique of tasks to processors. This allocation technique may be Bin- Packing where a number of tasks need to be packed to a number of bins (cores). The traditional Bin- Packing techniques used for single core are: NF (Next Fit), FF (First Fit), WF (Worst Fit), NFD (Next Fit Decreasing), FFD (First Fit Decreasing), and WFD (Worst Fit Decreasing).

For multi-core systems, the Bin-packing is known as VSBPP (Variable Sized Bin-Packing Problem) where there exist a large number of researches in it. For example [37] proposed four new Bin-Packing techniques for multi core system. The first one is for multi core system without DFVS, where all cores operate at the maximum frequency; the proposed SPF-FU (Slowest Processor First-Fully Utilize) is the same idea of traditional FF but processors are arranged according to their speed in an ascending order. The second one is for full-chip DVFS, where all cores in the same chip operate at the same frequency; the proposed FC-DVFS (Full-Chip - DVFS) uses the traditional WFD but gives all processors same utilization with different processor's speed. The third one is per core DVFS system, where each core operates at an individual frequency; the proposed PC-DVFS (Per-Core - DVFS) assigns task to a core and the next task to the next core, so that all cores have the same load with different speeds. The fourth one is for per-island DVFS system, where cores are partitioned into islands and each operates on a frequency; the proposed PI_DVFS (Per-Island- DVFS) assigns task to the worst fit core in island and next task to the worst fit core of the next island and so on.

Optimization techniques can be used for allocation of tasks to cores as genetic, swarm and others, which will be presented in section 7.

Thirdly, the task's nature differs in the scheduling technique. For example, for dependent tasks the precedence constraint is taken into consideration [16] in scheduling and always presented with DAG (Directed Acyclic Graph). On the other side, for independent tasks the precedence constraint isn't taken into consideration. Moreover, the scheduling techniques differ according to the task's execution time. For the past years almost all research papers focus on the deterministic execution time of task WCET (Worst Case Execution Time), but recently research papers take into consideration the probabilistic execution time of task to enhance the overall system utilization [11].

Finally, there are other considerations that are taken when designing energy efficient real-time scheduling. For Example using reinforcement learning in choosing the most suitable scheduling technique for each task set [18]. The reinforcement learning is discussed in section 6.

Some research papers take into consideration the thermal dissipation in scheduling the real-time tasks [17], which is presented in section 5.

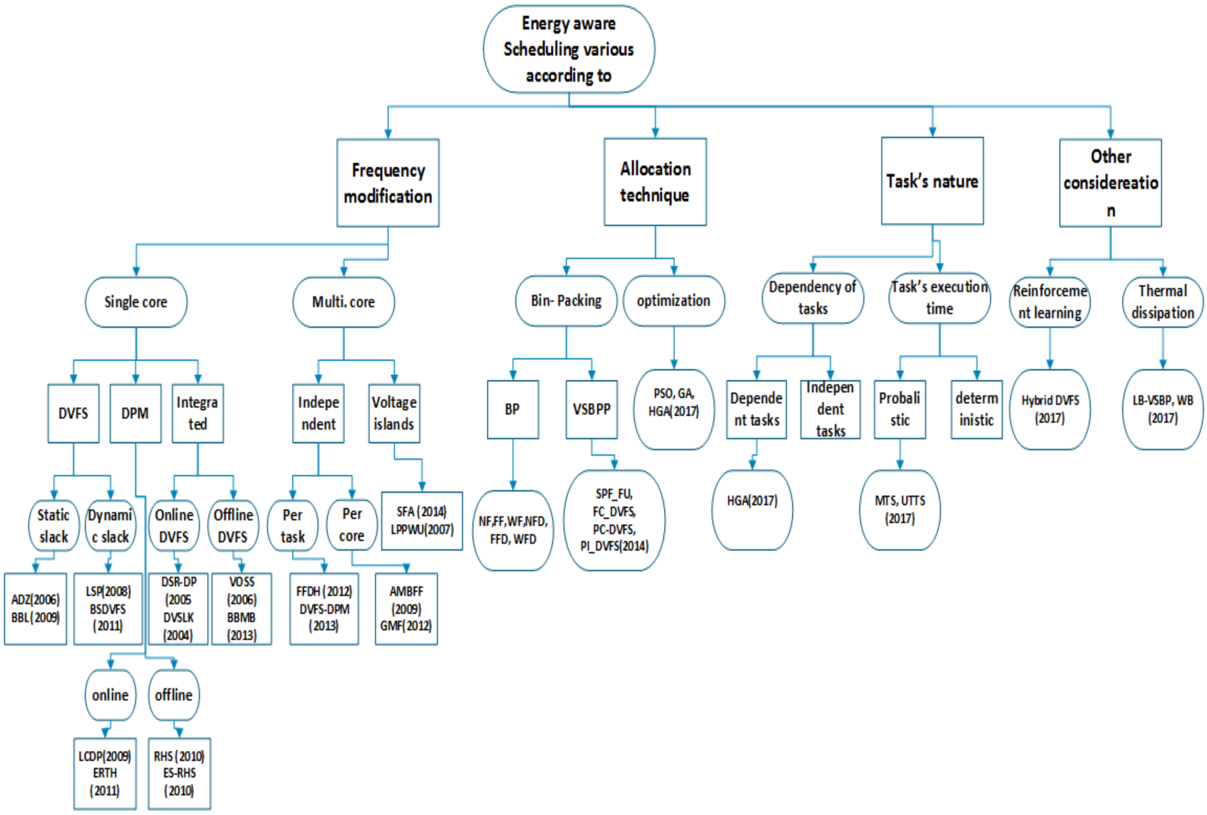

Figure 1: General taxonomy of real time scheduling

III. FREQUECY BASED ALGORITHMS

As discussed in [19] the techniques used for reducing the overall energy consumption of real-time system are divided into DVFS and DPM. The power consumption of a silicon-based CMOS processor is equal to $P_{distotal} = P_{static} + P_{dynamic}$ (1), where P_{static} is the static power or sometimes called the leakage power and equals $P_{static} = V I_{leak}$ (2), where V is the supply voltage and I_{leak} is the leakage current which is independent of the actual frequency and the system activity. Static power is approximately proportional to the leakage current $P_{static}\ \alpha\ I_{leak}$ (3). As a consequence, switching the processor to the sleep mode decrease the I_{leak} and so P_{static} decreases and the overall power dissipation decreases; This is the idea of DPM, where scheduling is done to enter the processor in a sleep mode (i.e., inactive state) as long as possible but must guarantee that tasks' will finish their execution without missing their deadline.

$P_{dynamic}$ is the dynamic power which is consumed during the task's execution time and equals $P_{dynamic} = C V^2 f$ (4), where C is a constant, V is the supply voltage, and f is the operating frequency. As a result, the value of dynamic power is directly proportional to the frequency. So, by decreasing the frequency of the processor leading to a decrease in $P_{dynamic}$ and the overall power dissipation decreases. This is the idea of DVFS, where scheduling is done by decreasing the processor's frequency as long as possible. The decrease in the processor's frequency leads to an increase in the task's execution time should be taken in consideration to avoid task's miss their deadline.

As shown in Fig. 1, for single core it is classified into DVFS scheduling, DPM scheduling, and hybrid between them. For DVFS scheduling it is divided into dynamic slack and static slack. For static slack, it decreases the frequency to utilize the remaining utilization for the worst case execution time of the task set. Task set Γ is the number of tasks need to be executed in the processor where $\Gamma = \{\tau_1, \ldots\ldots, \tau_{n1}\}$ (5) and we have n tasks. Each task i here is considered to have $\tau_i(T_i, WCET_i, D_i)$ (6), period, worst-case execution time, and deadline for task i respectively. From the recent papers published in DVFS scheduling for static slack are ADZ [38] and BBL [39]. In ADZ the algorithm tries to find the optimal speed for each task without missing the task's deadline using EDF (Earliest Deadline Scheduling). Scheduling is achieved with time order of complexity $O(n^3)$. In BBL, the algorithm uses an offline method called BBL to find the optimal speed offline, where it selects the frequency pair that decreases the overall energy consumption and takes into consideration the switching energy consumption.

For dynamic slack, the frequency is decreased to utilize the remaining utilization for the worst-case execution time and the remaining utilization between the actual execution time and WCET for each task set. The recent papers published in DVFS scheduling for dynamic slack are LSP [41] and BSDVFS [40]. In LSP, the utilization value of all real-time tasks is increased to utilize the remaining utilization of WCET and this is achieved before execution. Therefore, the exceeded utilization at run time is handled automatically as the dynamic slack. While BSDVFS takes in to consideration the overhead of switching to a new speed and return to the normal speed to check if the dynamic slack is enough to execute the next job with the new speed or not.

DPM scheduling is used to switch the processor to sleep mode and takes into consideration the break-even-time (i.e. is the biggest time needed to transfer from active to sleep state without using power for speed transition greater than or equal to the saved power). This can be done offline or online. The recent researches for online DPM scheduling are LC-DP [42] and ERTH [43], where the scheduling algorithm runs online. The LC-DP is used to compute the maximum time the task can be delayed without missing their deadline. Then the system switches to sleep mode for this time. In ERTH, the algorithm takes into consideration both the static and dynamic slack to compute the maximum time for system to stay in sleep mode efficiently.

The offline DPM researches are Rate-Harmonized Scheduler (RHS) and Energy-Saving RHS (ES-RHS) [44], where the scheduling algorithm runs offline. The RHS uses the idea of harmonic period (i.e., task's periods being integer multiple of each other). So that the scheduler schedules tasks according to their harmonic period to increase the period in which the processor is idle and puts it in to the sleep mode for a larger time. The ES-RHS is similar to RHS algorithm but it tries to collect idle time to decrease the transition from active state to sleep state.

By combining DVFS-DPM, both features of speed scaling and switching to low power state (i.e., sleep mode) can be obtained. According to the task set, a decision is taken to decrease frequency (i.e., speed) or increases idle time. The integrated DVFS-DPM is divided into online DVFS and offline DVFS. In online DVFS, the calculation of unused computation time is carried out online. There are several researches such as DVSLK [46] and DSR-DP [47]. The Dynamic Slack Reclamation with Dynamic Procrastination (DSR-DP) collects all the unused computation in a free run time; then this unused computation is used to decrease the processor's speed during task's execution time or switch to sleep mode. It chooses to switch to a sleep mode rather than decreasing the speed; if critical speed is reached. DVSLK chooses for each task the suitable speed in order to decrease both the static and dynamic energy. The offline DVFS calculates the unused computation time offline. From recent researches published in offline DVFS are VOSS [45] and BBMB [48].

Online Simulated Scheduling (OSS) simulates the tasks' execution and collects their idle time. So, during run-time tasks are delayed by the collected idle time and processor switch to a sleep mode this time until the first job arrives. VOSS Virtual OSS uses the same idea of OSS with virtual blocking time.

BBMB algorithm performs in two stages. Stage one is offline stage where the least speed the processor can operate on it without missing task's deadline is computed. Also, the maximum time the tasks can be blocked execution without missing their deadline is computed. Stage two is a run time stage where the tasks operate according to the calculated speed in offline stage and enter the sleep mode in the blocking time. Finally, during run time if there is interval the system is idle in it the processor enters the sleep mode.

For multi-processor as shown in Fig. 1, it is divided into independent frequency and voltage islands. In independent frequency per-task DVFS every task operates with the suitable frequency. From the recent researches are FFDH [49] and DVFS-DPM [50]. The FFDH algorithm stores a discrete frequencies and their relative power consumption in a look up table. So that during assignment of tasks to processors looks in the lock up table to find the set of operating frequencies of tasks which minimizes the overall energy consumption. DVFS-DPM algorithm uses the hybrid DVFS and DPM in the scheduling with choosing the suitable execution frequency for each task.

In independent frequency per-core DVFS every core operates in a single frequency. The recent researches are AMBFF and GMF [51]. In Adaptive Minimal Bound First-Fit (AMBFF) uses a lookup table that stores the speed and the corresponding power consumption, where it uses the FF bin packing technique to assign tasks to cores; lock for the suitable speed of core from the lockup table.

Growing Minimum Frequency (GMF) uses DVFS where there exists a set of discrete frequencies and starts with the minimum frequency. Then, the frequency is increased by a step until it reaches the one suitable for the task sets to be assigned to a specific core.

Due to the hardware complexity of assigning independent frequency to every task or core, the voltage islands appeared where a specific set of cores (i.e. islands) is assigned a specific frequency. The recent researches in voltage islands are SFA [53] and LPPWU [54]. Single Frequency Approximation (SFA) is used to choose the minimum fixed frequency that can be assigned to a number of cores and guarantees the schedulability of task sets, which is done after the tasks' allocation to processors.

LPPWU algorithm schedule tasks to processors with precedence constraint that is represented by DAG. The algorithm uses a graph diagram which includes the set of discrete frequencies and the sleep states. Including the energy cost for each transition. According to the graph diagram, tasks are allocated to cores and a set of cores is assigned a fixed frequency and enters a sleep mode to minimize the overall energy consumption.

IV. PROBABILISTIC BASED SCHEDULING ALGORITHM

The scheduling based on the worst-case execution time becomes impractical. As a consequence, many research papers are now considering the task's execution time to be probabilistic rather than deterministic. So in this case the task is considered to have $\tau_i(T_i, C_i, D_i)$ (7), where $C_i = \{c_{\min}, \ldots\ldots, c_{\max}\}, \{P_r(c_{min}), \ldots\ldots, P_r(c_{max})\}$ (8), where $c_k \in [c_{\min}, c_{\max}]$ (9) and $\sum_{k=min}^{k=max} P_r(c_k) = 1$ (10). As shown from (8) every task has a number of execution time that varies from c_{min} to c_{max} and each execution has a probability P_r. Since the execution time is not deterministic so the tasks may miss their deadline. As a result there is a probability that a task may miss its deadline [11] and is defined as $DMP_{i_j} = P_r(R_{i_j} > RD_{i_j})$, DMP_{i_j} is the Deadline Miss Probability for task i and job j (i.e. each task is repeated after a period T which is known as task's job). $DMP_{\Gamma} = \max\{DMP_i\}$ (11), DMP_{Γ} is the Deadline Miss Probability for the entire task set. So during the scheduling, the task set is scheduled when the value of $DMP_{\Gamma} \leq C$ where C is a predefined portion of task set that the deadline miss probability is satisfied.

From the recent researches published in probabilistic based scheduling with harmonic nature is [11], where the algorithm considered the probabilistic execution time of tasks and made four new metrics. The metrics are used to calculate the execution time from its probabilistic values and from it the degree of harmonicity between two probabilistic tasks is determined. Based on these metrics a new partitioning algorithm is proposed. The first metric uses the mean value of the probabilistic task's execution time multiplied by the probability to determine the execution time of a task. The other uses the variance value. One metric uses the cumulative distribution. Finally, the last metric uses the utilization sum. There are two algorithms that are used to choose the best subset and assign it to the processors. The first one is the Mean Based sub Task set Selection (MTS), where it calculates for every task set the mean value of utilization and chooses the higher. The second one is the Utilization Threshold-based sub Task seT Selection (UTTS), where it chooses the task set which has the highest probability of total utilization greater than the threshold value.

V. THERMAL ENERGY BASED SCHEDULING ALGORITHM

As discussed before that $P_{distotal} = P_{static} + P_{dynamic}$, but here the thermal dissipation is considered to avoid the hot spots and prolong the lifetime of cores. To clarify that the total power dissipation will be overwritten in the form $P_{distotal} = \alpha(v_i) + \beta T_i(t) + \gamma(v_i) v_i^3$ (12), where α, γ, and β are constants, v_i is the supply voltage, and $T_i(t)$ is the temperature. Also $P_{static} = \alpha(v_i) + \beta T_i(t)$ (13) and $P_{dynamic} = \gamma(v_i) v_i^3$ (14).

The temperature value is calculated according to the heat transfer and the electrical phenomena [17] as shown in the formula $\frac{dT(t)}{dt} = AT(t) + C^{-1}(\psi(v) + \eta)$ (15), where A is a coefficient matrix, $T(t)$ temperature vector, C is the diagonal of thermal capacitance. $\psi(v) = \alpha(v_i) + \gamma(v_i) v_i^3$ (16), $\eta = \frac{T_{amp}}{R_{ii}}$, where R_{ii} is the thermal resistance of core i to itself and T_{amp} is the ambient temperature.

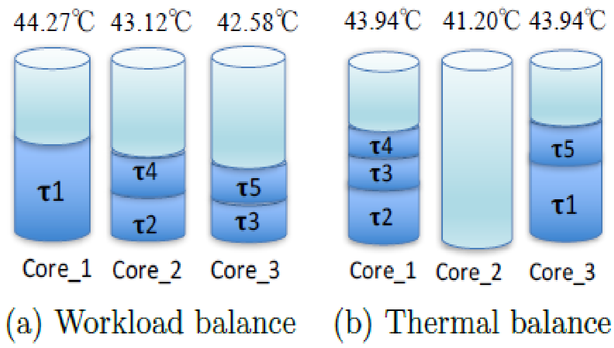

Fig 2: Example of workload and thermal balance [17]

From recent researches in thermal energy based scheduling algorithms is the [17], which takes temperature value into consideration and assigns tasks to the processors without exceeding the temperature threshold. As shown in Fig. 2 an example for assigning five tasks on three cores. The first uses the traditional scheduling algorithm without taking temperature into consideration known as the Workload balance. The second one is the Thermal balance, which schedules tasks to processors until reach the temperature threshold. According to Fig. 2

using the workload balance three cores are used, while in thermal balance only two cores are used and the third enters the sleep mode. So the overall energy consumption is reduced.

VI. REINFORCEMENT LEARNING ENERGY BASED SCHEDULING ALGORITHM

It is impossible to have one single scheduling algorithm that is suitable for all different forms of workloads. As a consequence, the technique of reinforcement learning is used to switch between different known scheduling algorithms to the best one fits the current workload on fly.

The learning is done only based on the past experiences. From the recent researches published in this point is [18], where a number of DVFS scheduling techniques exists and a DVFS controller is the one responsible for choosing the most suitable DVFS technique for the current task set. As shown in Fig. 3, the DFVS controller first invokes the penalty calculator to calculate the energy consumption in the last period of that task set to determine the penalty value given from the previous selected DVFS technique. Secondly, updates the State-tech map based on the penalty value calculated. Finally, the state observer runs to determine the present system state. From the present state it chooses the best DFVS technique from the State-tech map which has the least power consumption.

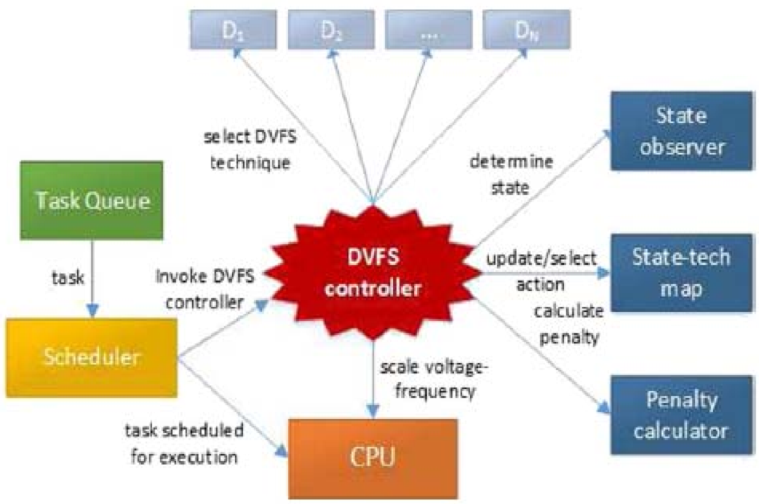

Fig 3: The system diagram of reinforcement learning DVFS [18]

VII. ENERGY AWARE SCHEDULING ALGORITHM BASED ON OPTIMIZATION TECHNIQUE

The optimization techniques like Genetic, Particle Swarm, Stochastic Evolution, Ant, and others algorithms are used to determine the optimal solution of a problem. Recent researches in energy aware scheduling uses the optimization techniques to solve the Bin-packing and scheduling problem.

As shown in [16] a new hybrid Genetic algorithm, which improves the overall energy consumption.

The precedence constraints for dependent tasks is always represented by the $DAG\ (T, E)$ as shown in Fig. 4, where T is set of tasks and E is the dependence.

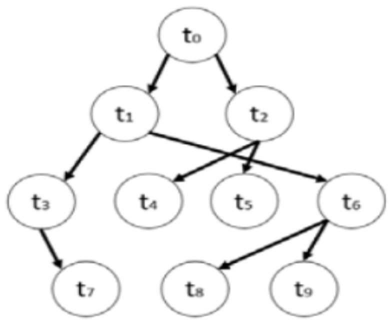

Fig 4: Example of DAG [16]

The precedence constraint is presented by (17, 18, 19,20) , $Pre(t_i) = \{ \text{t j} \mid \text{t j} \in \text{T}, e_{ji} \in \text{E}\}$ (17) is the set of all tasks j before task i in precedence. $succ(t_i) = \{ \text{t j} \mid \text{t j} \in \text{T}, e_{ij} \in \text{E}\}$ (18) is the set of all tasks j that are immediately after task i . e_{ij} means task i should complete execution before task j starts execution.

$aPre(t_i) = \{ t_j, t_k , t_1 , \ldots\ldots, t_p \}$ (19) is the set of all predecessor of task i ; if $\{e_{jk} , e_{kl}, \ldots e_{(p-1)p} , e_{pi} \epsilon$ E and $Pre(t_j) = \emptyset$ } (20).

As shown in the Fig. 4, $succ(t_4) = \emptyset$ and $succ(t_6) = \{ t_8 , t_9\}$. Also $Pre(t_2) = \{ t_0 \}$, $(t_5) = \{ t_2 \}$, $aPre(t_5) = \{ t_0, t_2\}$ and $aPre(t_7) = \{ t_0, t_1, t_3\}$.

LST and EST is defined to guarantee that tasks don't miss their deadline with precedence constraint, where $LST_i = D_i - C_i$ (20) and EST_i={$max_{j\epsilon \, pre(t_i)}$ {FT_j}if pre(t_i) ≠ ∅ else 0} (21) , $FT_j = ST_i + C_i$ (22) where $FT_j$is the finish time of task j . ST_i is the actual start execution time. Task i will not miss deadline if $EST_i \leq ST_i \leq LST_i$.

HGA algorithm tries to reach the optimal solution using hybrid between Genetic and Stochastic evolution algorithms. Each chromosome as shown in Fig. 5 is used to represent a solution. The chromosome is a two dimension matrix in which the row numbers represent the task number, the processor number assigned to it, and the speed of processor respectively. The columns represent the tasks in the task set. The order of tasks in a row represents the precedence constraint.

The algorithm uses the roulette wheel to choose the new offspring by swapping two randomly selected segments of two chromosomes. Then a mutation is performed to change a bit in the solution. In this algorithm rather than using a single mutation every iteration it uses the stochastic evolution search to find the appropriate value of permutation to be performed.

Task	1	2	3	4	5
Processor #	2	3	2	1	1
Voltage level	1	3	2	1	3

Fig 5: Example of chromosome [16]

VIII. CONCLUSION

The increase of the system demands for homogeneous and heterogeneous multi-core systems along with the future vision towards IoT system with limited resource constraints leads to the need of improving the energy aware scheduling for real time systems. Consequently, this paper addresses different perspectives of energy aware scheduling for real time systems. In this context, several factors such as execution models, the task dependency, the selection of tasks with harmonic nature, the thermal dissipation, the use of reinforcement learning, and the task allocation technique through bin-packing or optimization techniques are considered. Finally, this paper provided a guideline for choosing the proper energy aware scheduling criterion that meets the excessive requirement of the system.

REFERENCES

[1] G.M. Tchamgoue, K.H. Kim, and Y.K. Jun, “Power-aware scheduling of compositional real-time frameworks,” in Journal of Systems and Software, vol. 102, pp. 58–71, April 2015.

[2] E. Seo, J. Jeong, S. Park, and J. Lee, “Energy efficient scheduling of realtime tasks on multicore processors,” in IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 11, pp. 1540–1552, November 2008.

[3] W. Knight, “Two heads are better than one [dual-core processors],” in IEEE Review, vol. 51, no. 9, pp. 32–35, Septemper 2005.

[4] J. W. S. Liu, “Real-Time Systems,” Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[5] A. Atlas, and Abestavros, “Statistical rate monotonic scheduling,” in the 19th IEEE Proceedings of Real-Time Systems Symposium, pp. 123–132, 1989, DOI: http: //dx.doi.org/10.1109/REAL.1998.739737.

[6] J.P. Lehoczky, “Fixed priority scheduling of periodic task sets with arbitrary deadlines,” in 11th Proceedings of Real-Time Systems Symposium, pp. 201–209, 1990 DOI:http://dx.doi.org/10.1109/REAL.1990.128748.

[7] J. Lehoczky, LuiSha, and Y. Ding, “The rate monotonic scheduling algorithm: exact characterization and average case behavior,” in Proceedings of Real Time Systems Symposium, pp. 166–171, 1989, DOI: http:// dx.doi.org/ 10.1109/REAL.1989.63567.

[8] Dorin. Maxim, and L. Cucu-Grosjean, “Response Time Analysis for Fixed-Priority Tasks with Multiple Probabilistic Parameters,” in 34th IEEE proceeding of Real-Time Systems Symposium (RTSS), pp. 224–235, 2013, DOI:http://dx.doi.org/10.1109/RTSS.2013.30.

[9] Yue Lu, T. Nolte, I Bate, and L. Cucu-Grosjean, “A Statistical Response-Time Analysis of Real- Time Embedded Systems,” in 33rd IEEE proceeding of Real-Time Systems Symposium (RTSS), pp. 351–362, 2012, DOI:http: // dx.doi.org/10.1109/RTSS.2012.85.

[10] Li. Kenli, and Tang. Xiaoyong, “Energy-Efficient Stochastic Task Scheduling on Heterogeneous Computing Systems,” in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 11, pp. 2867 – 2876, November 2014.

[11] Tianyi Wang, SoamarHomsi, LinweiNiu, ShaoleiRen, OuBai, Gang Quan, and MeikangQiu, “Harmonicity Aware Task Partitioning for Fixed Priority Scheduling of Probabilistic Real-Time Tasks on Multi-Core Platforms,” in ACM Transactions on Embedded Computing Systems, 28 July 2017.

[12] Tianyi Wang, Qiushi Han, Shi Sha, Wujie Wen, Gang Quan, and MeikangQiu, “On harmonic fixed-priority scheduling of periodic real-time tasks with constrained deadlines,” in 53rd ACM Proceedings of the Annual Design Automation Conference, pp. 5-9, 2016.

[13] Mitra Nasri, Gerhard Fohler, and Mehdi Kargahi, “A Framework to Construct Customized Harmonic Periods for Real- Time Systems,” in 26th IEEE Euromicro Conference on Real-Time Systems (ECRTS), pp. 211–220, 2014.

[14] Ming Fan and Gang Quan, “Harmonic semi-partitioned scheduling for fixed-priority real-time tasks on multi-core platform,” in Design Automation Test in Europe Conference Exhibition (DATE), pp. 503–508, 2012. DOI: http://dx.doi.org/10.1109/DATE.2012.6176521.

[15] Ming Fan and Gang Quan, “Harmonic-Aware Multi-Core Scheduling for Fixed-Priority Real-Time Systems,” in 25 IEEE Transactions on Parallel and Distributed Systems, vol. 6, pp. 1476–1488, June 2014, DOI:http://dx.doi.org/10.1109/TPDS.2013.71.

[16] Amjad Mahmood, A. Salman Khan, FawziAlbalooshi, and Noor Awwad, "Energy-Aware Real-Time Task Scheduling in Multiprocessor Systems Using a Hybrid Genetic Algorithm," in Journal of Electronics, 2017, DOI:10.3390 / electronics6020040.
[17] Shi Sha, Wujie Wen, ShaoleiRen, and Gang Quan, "A Thermal-Balanced Variable-Sized-Bin-Packing Approach for Energy Efficient Multi-Core Real-Time Scheduling," in ACM, pp. 10-12, May 2017, DOI: http://dx.doi.org/10.1145/ 3060403.3060444.
[18] Fakhruddin Muhammad Mahbubul Islam and Lin. Man, "Hybrid DVFS Scheduling for Real-Time Systems Based on Reinforcement Learning," in IEEE Journal of Systems, vol. 11, no. 2, June 2017.
[19] Mario Bambagini, Mauro Marinoni, HakanAydin, and Giorgio Buttazzo, "Energy-Aware Scheduling for Real-Time Systems: A Survey," in ACM Transactions on Embedded Computing Systems, vol. 15, No. 1, 7:1-7:34, Januaury 2016.
[20] J. J. Chen, and L. Thiele, "Task partitioning and platform synthesis for energy efficiency," in 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. RTCSA '09, pp. 393–402, 2009.
[21] D. Li, and J. Wu. Minimizing, "Energy consumption for frame-based tasks on heterogeneous multiprocessor platforms," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, pp. 810–823, March 2015.
[22] S. Saha, Y. Lu, and J. S. Deogun, "Thermal-constrained energy-aware partitioning for heterogeneous multi-core multiprocessor real-time systems," in 18th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) , pp. 41–50, 2012.
[23] V. Hanumaiah, and S. Vrudhula, "Energy-efficient operation of multicore processors by dvfs, task migration, and active cooling," in IEEE Transactions on Computers, vol. 63, pp. 349–360, 2014.
[24] W. Sun, and T. Sugawara, "Heuristics and Evaluations of energy-aware task mapping on heterogeneous multiprocessors," in IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), pp. 599–607, 2011.
[25] H. Izakian, A. Abraham, and V. Snasel, "Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments," in International Joint Conference on Computational Sciences and Optimization, vol. 1, pp. 8–12, April 2009.
[26] M. Bhatti, C. Belleudy, and M. Auguin, "Hybrid power management in real time embedded systems: An interplay of DVFS and DPM techniques," in Journal of Real-Time System, vol. 47, no. 2, pp. 143–162, March 2011.
[27] H. Shen, Y. Tan, J. Lu, Q. Wu, and Q. Qiu, "Achieving autonomous power management using reinforcement learning," in Journal ACM Transaction on Design Automation of Electronic Systems (TODAES), vol. 18, no. 2, pp. 24:1–24:32, April 2013.
[28] R. Ye, and Q. Xu, "Learning-based power management for multi-core processors via idle period manipulation," in 17th proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC) , pp. 115–120, 2012.
[29] R. Glaubius, T. Tidwell, C. D. Gill, and W. D. Smart, "Real-time scheduling via reinforcement learning," Cornell University, 2012, [Online] Available: http://arxiv.org/ abs/1203.3481.
[30] W. Zhang, E. Bai, A. Cheng, "Solving Energy-Aware Real-Time Tasks Scheduling Problem with Shuffled Frog Leaping Algorithm on Heterogeneous Platforms," in Journal of Sensors, vol. 15, pp. 13778–13804, 2015.
[31] J. Oh, C. Wu, "Genetic-algorithm-based real-time task scheduling with multiple goals," in Journal of System Software, vol. 71, pp. 245–258, 2014.
[32] G. Kanagaraj, S.G. Ponnambalam, N. Jawahar, "A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems," in Journal of Computer Industry Engineering," vol. 66, pp. 1115–1124, 2013.
[33] M. Hadded, F. Jarray, G. Tlig, H. Hasni, "Hybridisation of genetic algorithms and tabu search approach for reconstructing convex binary images from discrete orthogonal projections," in Journal of Metaheuristics, vol. 3, pp. 291–319, 2014.
[34] A. Zeb, M. Khan, N. Khan, A. Tariq, L. Ali, F. Azam, and S.H. Jaffery, "Hybridization of simulated annealing with genetic algorithm for cell formation problem," in Journal of Advanced Manufacture Technology, vol. 86, pp. 2243–2254, 2016.
[35] K. Hyung, K. Sungho, "Communication-aware task scheduling and voltage selection for total energy minimization in a multiprocessor system using Ant Colony Optimization," in Journal of Information Science, vol. 181, pp. 3995–4008, 15 September 2011.
[36] W. Zhang, H. Xie, B. Cao, A. Cheng, "Energy-Aware Real-Time Task Scheduling for Heterogeneous Multiprocessors with Particle Swarm Optimization Algorithm," in Journal of Mathematical Problems in Engineering, Article ID 287475, 2014.
[37] Abdullah Elewi, Mohamed Shalan, MedhatAwadalla, and M. Saad. Elsayed, "Energy-efficient task allocation techniques for asymmetric multiprocessor embedded systems," in Journal of ACM Transactions on Embedded Computing Systems (TECS) Special Section ESFH'12, vol. 13, no. 2, Article no. 71, January 2014.
[38] Hakan Aydin, Vinay Devadas, and Dakai Zhu, "System-level energy management for periodic real-time tasks," in 27th IEEE International Proceedings of the Real-Time Systems Symposium (RTSS'06), 2006
[39] Enrico Bini, Giorgio C. Buttazzo, and Giuseppe Lipari, "Minimizing CPU energy in real-time systems with discrete speed management," in ACM Transactions on Embedded Computing Systems, vol. 4, pp. 31:1–31:23, July 2009.
[40] Mario Bambagini, Francesco Prosperi, Mauro Marinoni, and Giorgio C. Buttazzo, "Energy management for tiny real-time kernels," in Proceedings of the IEEE International Conference on Energy Aware Computing (ICEAC'11), 2011.
[41] Martin Lawitzky, David C. Snowdon, and Stefan M. Petters, "Integrating real-time and power management in a real system," Workshop on Operating System Platforms for Embedded Real-Time Applications (OSPERT), 2008.
[42] Yann-Hang Lee, P. Reddy Krishna, and C. Mani Krishna, "Scheduling techniques for reducing leakage power in hard real-time systems," in the 15th Proceedings of Euromicro Conference on Real-Time Systems (ECRTS'03), 2003.
[43] Muhammad Ali Awan and Stefan M. Petters, "Enhanced race-to-halt: A leakage-aware energy management approach for dynamic priority systems," in Euromicro Conference on Real-Time Systems (ECRTS'11), 2011.
[44] Anthony Rowe, Karthik Lakshmanan, Haifeng Zhu, and Ragunathan Rajkumar, "Rate-harmonized scheduling and its applicability to energy management," in IEEE Transactions on Industrial Informatics, vol. 3, pp. 265–275, 2010.
[45] Ravindra Jejurikar and Rajesh Gupta, "Dynamic slack reclamation with procrastination scheduling in real-time embedded systems," in Proceedings of the Conference on Design Automation Conference (DAC'05), 2005.
[46] Linwei Niu and Gang Quan, "Reducing both dynamic and leakage energy consumption for hard real-time systems," in Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES'04), 2004.
[47] Ravindra Jejurikar and Rajesh Gupta, "Procrastination scheduling in fixed priority real-time systems," in Conference on Languages, Compilers and Tools for Embedded Systems (LCTES'04), 2004.
[48] Mario Bambagini, Marko Bertogna, and Giorgio Buttazzo, "On the effectiveness of energy-aware realtime scheduling algorithms on single-core platforms," in the 19th Proceedings of Conference on Emerging Technologies and Factory Automation (ETFA'14),2014.
[49] Huiting Xu, Fanxin Kong, and Qingxu Deng, "Energy minimizing for parallel real-time tasks based on level-packing," in the 18th IEEE Proceedings of International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'12), 2012.
[50] Gang Chen, Kai Huang, and Alois Knoll, "Energy optimization for real-time multiprocessor system-onchip with optimal DVFS and DPM combination," in ACM Transactions on Embedded Computing Systems, vol. 13, Issue 3, June 2013.

[51] Gang Zeng, Tetsuo Yokoyama, Tomiyama, Hiroyuk., and Hiroaki Takada, "Practical energy-aware scheduling for real-time multiprocessor systems," in the 15th IEEE Proceedings of International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'09), 2009.
[52] Gabriel A. Moreno and Dionisio De Niz, "An optimal real-time voltage and frequency scaling for uniform multiprocessors," in the 18th IEEE Proceedings of International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'12), 2012.
[53] Santiago Pagani and Jian-Jia Chen, "Energy efficiency analysis for the single frequency approximation (SFA) scheme," in ACM Transactions on Embedded Computing Systems, vo. 13, Issue 5, pp. 158:1– 158:25, Septemper 2014.
[54] Krishnan Srinivasan and Karam S. Chatha, "Integer linear programming and heuristic techniques for system-level low power scheduling on multiprocessor architectures under throughput constraints," in VLSI Journal Integration, vol. 40, no. 3, April 2007.