Real Time Global Scheduling Analysis for Generalized DAG Task Upon Heterogeneous Machine

Adhe Widianjaya, Dadet Pramadihanto, Sritrusta Sukaridhoto
EEPIS Robotics Research Center,
Electronic Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia.
adhe@pasca.student.pens.ac.id

***Abstract*—This paper presents a heterogeneous model of real time task system with novel processing rate work parameter. The model considers the precedence constraint with implicit deadline. Global EDF scheduling algorithm was applied on this model and analyzed in the context of its schedulability and capacity augmentation bound. By combining parallel tasks analysis upon identical multiprocessor and their processing rate upon heterogeneous system, we derived utilization augmentation, which is useful for extending capacity augmentation bound. Our experiments showed that there was a schedulable task system which is characterized by utilization augmentation upon heterogeneous system under Global EDF with capacity augmentation bound of $(4 - m/2)(1 + \sqrt{(U\_Aug)})$. Our model with processing rate is also useful for practical consideration.**

***Index Terms*—Real Time Scheduling; Capacity Augmentation Bound; Heterogeneous Processor; Unrelated Parallel Machine; Parallel Tasks; Directed Acyclic Graph.**

I. INTRODUCTION

Increasing power demand of computer processor has led to the development of multi core technology since early of the 21st century. Consequently, the birth of many core computing systems has triggered various research possibilities. Nowadays, extensive studies have been carried out resulting from the transition of the real-time scheduling upon computer-based system into the multiprocessor scheduling. Such kinds of scheduling open various possibilities, including the existence of high load real time task. Parallel processing of high load task may be realized under the multicore and multiprocessor technology. Considering that there are *m* processing elements, which represent *m* number of cores or processors, the real-time task system utilization is extended to at most *m*. However, to find an optimal scheduler in such condition is proven to be nearly impossible [1], and it is categorized as the NP-Hard. Several researches have utilized heuristic techniques to solve such problems.

There have been some recent works, which offered approximations for multiprocessor scheduling problem. Traditional task model, previously used in real time uniprocessor scheduling, has been extended with the multiprocessor analysis. Accordingly, researchers in this field adopted two major strategies: *global* and *partitioned* scheduling. Global scheduling of hard real time task system tries to schedule the highest priority task available until all processors are busy or there is no more task that is ready. Such method is implemented along with the popular scheduling policy, Earliest Deadline First (EDF) [2]. While partitioned scheduling allocates each task using specific allocation algorithm to decide which processor is assigned to those task, the uniprocessor scheduler, such as EDF, known as the optimal dynamic priority scheduler upon multiprocessor [3], may schedule the whole tasks assigned on specific processor. The task migration to another processor is strictly forbidden in the partitioned scheduling. Then, the *semi partitioned* scheduling was proposed [4] and evaluated [5]. Task migration is forbidden unless scheduling upon assigned processor is no longer feasible.

However, the existing approximations are limited to identical multiprocessor scheduling. Since the development of multicore has reached to heterogeneous core, such methods are irrelevant upon heterogeneous machine. This is based on a task system that is modelled using three parameters sporadic task [6]. A task is characterized by its worst-case execution time C_i, deadline interval D_i, and period interval T_i relative to its arrival time. Scheduling current task upon heterogeneous processors transform the behavior of C_i into C_{ij}, which is the worst case execution time of *i'th* task in *j'th* processor, since the processing capability of each processor for current task is different. A more realistic model of parallel task has been previously discussed [7], added precedence constraint, where each task is represented as *directed acyclic graph* (DAG) that consists of nodes of subtask. They are then called the *work* and the edges, which represent a dependency between *work*. A *work* is ready only if all its predecessors have been executed. Since each *work* may run in different processors, which is unrelated, the worst-case execution time C_i for the whole task execution may vary depending on the condition.

In this paper, we consider an extension to the current implicit deadline DAG sporadic task model, which leads to a novel schedulability analysis of hard real time task system. We considered a rate matrix *R* which characterizes each *work* by *slowdown rate*. The rate of 1 may be understood as the fastest worst case execution time of *k'th* work from *i'th* task. While the rate of 1 tell us that it is not possible to schedule the current *k'th* work from *i'th* task upon *j'th* processor. As previously known that the EDF scheduler is optimal for uniprocessor scheduling[5], we consider a system that consists of global EDF (GEDF) to schedule *n* number of task set and analyze system properties and schedulability. Finally, the contribution of our work are: (1) we provide implicit deadline DAG sporadic task with *R* processing rate model and analysis, and (2) provide utilization augmentation as an extension of previously presented capacity augmentation bound for multiprocessor system. We also open the possibilities that such model may represent multiprocessor scheduling under both partitioned and global strategy.

II. Related Works

Partitioned multiprocessor scheduling under time constrained hard real time system has been previously studied by real time system researcher. However, the current uniprocessor bound in EDF and Rate Monotonic (RM) are used as guide for the upper and lower bound in partitioned scheduling. While partitioned strategy is considered as binary packing problem, various allocation algorithms are analyzed in the context of system utilization bound [2].

Scheduling sporadic task system upon multiprocessor has been presented in [6]. It is known that under GEDF scheduler, *resource augmentation bound* of $2 - 1/m$ has been proven by Bonafici et al. Recently, such model has been extended with the precedence constraint and recently analyzed by Li et al. [10] Li also provided *capacity augmentation bound* to distinguish from previous *resource augmentation bound*. However, both terms are similar to *utilization bound* studied by Liu et al. [3] in uniprocessor scheduling, and Lopez et al. [2] in partitioned multiprocessor scheduling, which equally lead to schedulability analysis. Our work presented in this paper used similar parameter to analyze task system schedulability upon heterogeneous machine.

Raravi et al. [9] provided unrelated processor task assignment algorithm by dividing a set of task into heavy and light task, and treat the machine as two different machine. Anderson et al define the clusters of processor with the same type and schedule the assigned task set using global EDF. Lawler et al. [10] and Wiese et al. [11] formulated linear programming solution for scheduling upon unrelated parallel processor. However, we offer different approach which treat all tasks as *m*-processor assignable with processing rate defined by *R*, unless its rate is equal to 0.

III. Task Model

We considered implicit deadline DAG sporadic task with processing rate extension running on a system with *m* number of heterogeneous processor. An implicit deadline sporadic task $\tau_i \in \mathbb{T}$, where $\mathbb{T}$ is DAG sporadic task set, and *i* is the number of task. Each task τ_i is represented by *directed acyclic graph* (DAG), characterized by its deadline interval D_i and time period T_i. Since we address implicit deadline task system, equation (1) will hold. Task τ_i consists of *r* number of subtasks called *work* w_k^i and characterized by its worst case execution time W_k^i, where $k = 1, \ldots, r$. Each work is represented by node. An edge links node w_p^i to w_q^i, means that w_p^i must be fully executed so that w_p^i becomes ready.

$$\boldsymbol{D_i = T_i} \tag{1}$$

For each task τ_i in $\mathbb{T}$ task set, we provided processing rate matrix R_i with the size of $r \times m$, where *n* is the number of work in task τ_i and *m* is the number of processing elements. This is may be understood that every work of each task in DAG sporadic task system is characterized by rate r_{kj}, which is the elements of processing rate matrix R_i, where $k = 1, \ldots, r$, and $j = 1, \ldots, m$. The value of processing rate r_{kj} is between $0 \leq r_{kj} \leq 1$ and represents the amount of total work done in unit of time. The value of processing rate is inversely proportional to the execution time in *j'th* processor. The lower the processing rate by *j'th* processor, the longer the work is executed in *j'th* processor. Here, we also have definition of identical processing rate (Definition 1 – which is similarly represent identical multiprocessors) and heterogeneous processing rate (Definition 2).

DEFINITION 1. Let task τ_i with *r* is the number of work, and *m* is the number of available processors, R_i is identical, where $r_{kj} = 1$ for all $\forall k$ and $\forall j$, where $k = 1, \ldots, r$, and $j = 1, \ldots, m$.

DEFINITION 2. Let task τ_i with *r* is the number of work, and *m* is the number of available processors, R_i is heterogeneous, where $0 \leq r_{kj} \leq 1$ for $\forall k$ and $\forall j$, where $k = 1, \ldots, r$, and $j = 1, \ldots, m$. This definition does not infer to the processing rate of processors; thus it also does not conclude that the order of processors by its processing rate since the work processing rate r_{kj} is arbitrary for all processors.

Let $\mathbb{I}$ is a set of identical processing rate matrix, $\mathbb{H}$ is a set of heterogeneous processing rate matrix, *r* is the number of work in task τ_i, and *m* is the number of processor, then equation (2) and (3) will hold for identical multiprocessor and heterogeneous multiprocessor system respectively. We define *m*–Processing Assignable task system (Definition 3) to support further discussions and analysis.

$$\boldsymbol{R_i} \in \mathbb{H} \; \boldsymbol{if} \; \boldsymbol{0} \leq \boldsymbol{r_{kj}} \leq \boldsymbol{1} \text{ for } \forall \boldsymbol{r_{kj}} \tag{2}$$

$$\boldsymbol{R_i} \in \mathbb{I} \; \boldsymbol{if} \; \boldsymbol{r_{kj}} = \boldsymbol{1} \text{ for } \forall \boldsymbol{r_{kj}} \tag{3}$$

DEFINTION 3. Task system is *m*–Processing Assignable only if it holds and is subjected to *R* processing rate parameters.

Another way to understand definition 3 is: let a task system which consists of task τ_i which its *works* is characterized by processing rate r_{kj}, $0 \leq r_{kj} \leq 1$, then task scheduling upon *j'th* processor is subjected to its *work's* processing rate. This understanding also holds although $r_{kj} = 0$, which means that the *k'th* work would not use any resource if it is assigned in *j'th* processor.

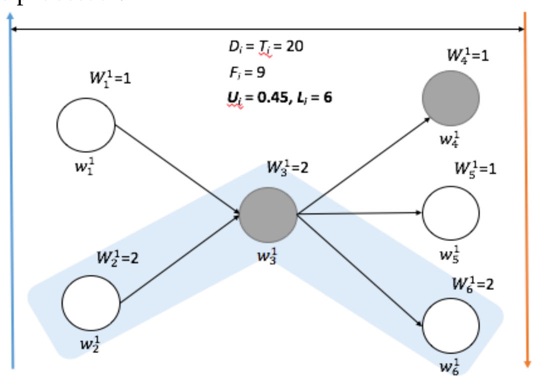

Figure 1: Scheduling DAG Sporadic Task System in Identical Multiprocessor

Let F_i is the function that produces the worst-case execution time of task τ_i in a single processor, which also means as the total W_k^i if $m = 1$. This is similar to the notion *volume* vol_i presented by Li et al.[8] We also followed the definition of critical path L_i. Then, the task utilization U_i and density $dens_i$ may be computed using formula in Equation (4) and (5). By observing Figure 1 (assumed R_i is identical), the critical path may be interpreted as a chain started from node w_2^1 goes to w_3^1 and ends at w_6^1, which is the longest path by considering each W. Since all W_k^i, worst case execution time of works of task τ_i, may be assigned in different processor, the value of F_i and also the form of L_i depends on R_i. Eventually, this condition leads to the definitions of Best Work and Worst Work of implicit deadline DAG sporadic task system.

$$U_i = \frac{F_i}{T_i} \quad (4)$$

$$dens_i = \frac{L_i}{D_i} \quad (5)$$

DEFINITION 4. Task is under the best work condition if the assignment of all k work of task τ_i in the machine system, which consists of m processor, produces the *lowest* worst case execution time among all possible assignments.

DEFINITION 5. Task is under the worst work condition if the assignment of all k work of task τ_i in machine system which consists of m processor and produces the *highest* worst case execution time among all possible assignments.

$$R_1 = \begin{bmatrix} 0.5 & 0.5 & 1 \\ 0.5 & 0.5 & 1 \\ 0.5 & 1 & 0.5 \\ 0.5 & 1 & 0.5 \\ 1 & 0.5 & 0 \\ 1 & 0.5 & 0 \end{bmatrix} \quad (6)$$

$$W_1 = [1 \quad 2 \quad 2 \quad 1 \quad 1 \quad 2] \quad (7)$$

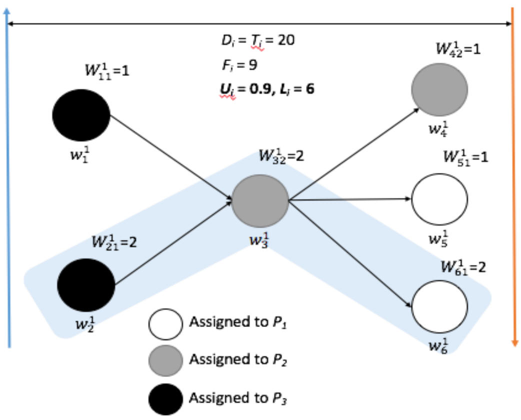

Figure 2: Scheduling DAG Sporadic Task System in Heterogeneous Multiprocessor Under Best Work Condition

For example, from equation (6), consider R_1 is the processing rate matrix of task τ_1, which $k = 6$, W^1 is a vector that represents normalized worst case execution time of works of task τ_1 (from equation (7)). Then $W_{kj}^1$is the worst case execution time of *k'th* work of task τ_1 assigned in *j'th* processor. $W^1$is defined in a particular way to simplify the illustration. Such system is then scheduled in $m = 3$ in heterogeneous processors. Figure 2 illustrates the heterogeneous scheduling since there exists a possibility of $r_{kj} = 0$ for system under the worst work condition. There exists a value L_i of infinite, which gives unrealistic and probably pessimism for finding schedulable approximations. However, the allocation algorithm may schedule such task in non zero processing rate condition so that the Reasonable Worst Work condition may apply.

DEFINITION 6. The task is under reasonable worst work condition if the assignment of all k work of task τ_i in machine system, which consists of m processor, limited to non zero processing rate $r_{kj} > 0$, and produces the *highest* worst case execution time among all possible assignments.

Figure 3 provides proper assignment of 5'th work and 6'th work for the reasonable worst work condition. The model also provides understanding that there are limited resource assignments for specific works of task τ_1. Practically, the processing rate gives an overview for adjusting CPU affinities so that the worst work condition would not apply.

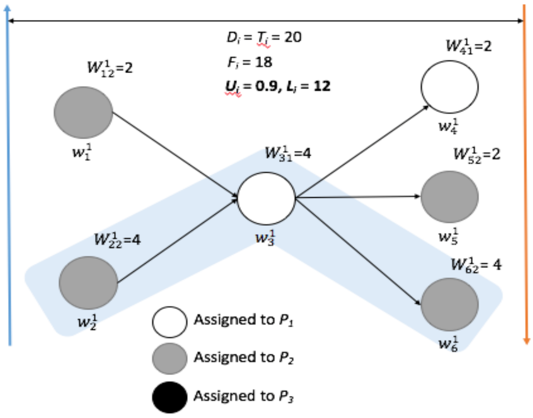

Figure 3: Scheduling DAG Sporadic Task System in Heterogeneous Multiprocessor Under Reasonable Worst Work Condition

IV. SYSTEM SCHEDULABILITY UNDER G-EDF

For identical multiprocessor $\mathbb{H} = \{\emptyset\}$, and $|\mathbb{H}| = 0$, the processing rate for all task $\tau_1 \in \mathbb{T}$ is one. Therefore, the maximum utilization provided by m identical multiprocessor is also m. On the other hand, for heterogeneous multiprocessor there exists $r_{kj} \neq r_{(k+1)j}$, so that$|\mathbb{H}| > 0$, which add more utilization demand by task system for m maximum utilization machine. In this condition, however, even an ideal scheduler may not be able to schedule all tasks before their deadline. Resource augmentation strategy is needed and its number depends on the utilization of augmentation of heterogeneous task system. The capacity of augmentation bound of $4 - \frac{m}{2}$ for Global EDF in identical multiprocessor may be extended to provide better schedulability analysis.

DEFINITION 7. In heterogeneous system with m number of processor, utilization augmentation U_{Aug} is the absolute value of difference between the utilization in identical multiprocessor m and utilization in heterogeneous multiprocessor m'.

Since utilization demand is proportional to capacity augmentation bound, we may formulate the new capacity augmentation bound under heterogeneous system with m number of processor and utilization augmentation U_{Aug} is $\left(4 - \frac{m}{2}\right)\left(1 + \sqrt{U_{Aug}}\right)$. Henceforth, the heterogeneous system is characterized with the utilization augmentation.

V. Evaluation and Discussions

We conducted several simulations to observe the behavior of the parallel task upon heterogeneous system modelled by R processing rate. Synchronous tasks were generated until the total utilization reach 99% of m. R, which is $n \times m$ matrix was generated and filled with one. Consider p parameter, where p < m, then p, the number of heterogeneous processors was then decided randomly and p-column, which represents the processing rate of task τ_1 in p*'th* processor was filled with 0.5. This condition produced a heterogeneous processor with the processing rates of 0.5 and 1. The numbers of heterogeneous processor are 25% and 50% of the simulated processors. Hundreds of generated tasks were then scheduled using global EDF with increasing speedup, and failure ratio was observed. Since the task system is *m-Processor Assignable*, task-processor assignment consideration was temporarily abandoned, though the best and worst case result were also presented. Simulations were performed using the number of processors $m = 4$, and $m = 8$.

Both Figure 4 and 5 show the same result for the identical multiprocessor [8]. The failure rate became zero for processor speedup ≈ 2. For $m=4$, utilization augmentation of 25% and 50% heterogeneous processors were 1 and 4 respectively, while utilization augmentation of 25% and 50% heterogeneous processors in $m = 8$ were 2 and 8. Since we extended the capacity augmentation bound of $\left(4 - \frac{m}{2}\right)\left(1 + \sqrt{U_{Aug}}\right)$ for U_{Aug} is the utilization augmentation, capacity augmentation bound for 25% and 50% heterogeneous processors increased between 4 and 8. This result show that the speedup above 4 and $U_A \approx 8$ was schedulable under global EDF scheduling algorithm.

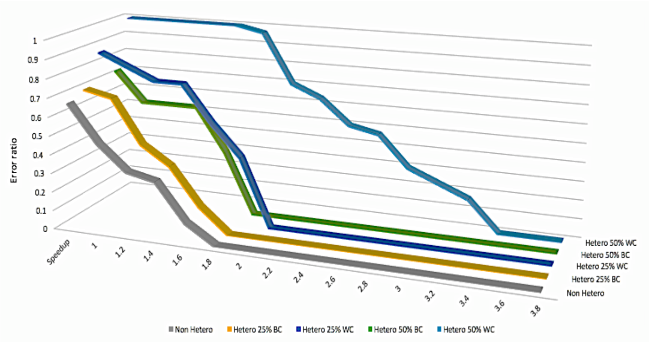

Figure 5: System failure ratio under identical and heterogeneous multiprocessor system for m = 4

Capacity augmentation bound of $4 - \frac{m}{2}$ as previously stated by Li et al. [8] may be understood that if the processor speedup is 2, the system is schedulable under global EDF scheduling algorithm. Since we provided a multiplication with $\left(1 + \sqrt{U_{Aug}}\right)$, then in our experiment, we had speedup of 4 for maintaining system schedulability under utilization augmentation between 1 and 8. This result was also reflected on the best work and worst work condition as previously described in section 3. Under the best condition, the total system utilization may reach near to m on the identical multiprocessor. Therefore, all the Best Condition (labelled as BC) results show similar behavior with the non heterogeneous system.

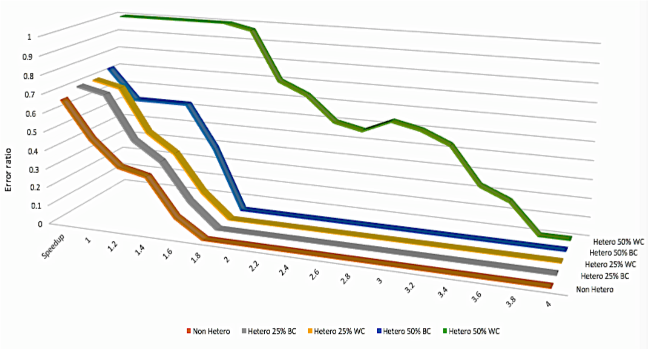

Figure 6: System failure ratio under identical and heterogeneous multiprocessor system for $m = 8$

Figure 7 presents the system failure ratio comparison for different number of processors. We inserted the Best Case (BC) results only since we wanted to analyze the system under different number of processors. The results become worst for higher number processors. These are caused by the higher number generated tasks to fulfill higher utilization. As the task set size becomes larger, the time and deadline collision among tasks are increased, leading to the difficulty for the Global EDF to schedule all tasks before their deadline. However, the capacity augmentation bound of $\left(4 - \frac{m}{2}\right)\left(1 + \sqrt{U_{Aug}}\right)$ still applies.

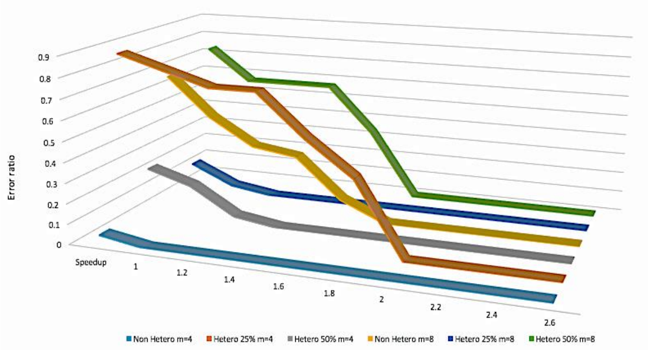

Figure 7: System failure ratio comparison under identical and heterogeneous multiprocessor for $m = 4$ and $m = 8$

Industrial vendors have already implemented non-symmetric multiprocessor system. ARM big.LITTLE13 provides two type processors with different processing speed and power consumption. Freescale iMX SoloX14 has ARM Cortex A9 processor that is side by side with ARM Cortex M4 microcontroller. The remaining challenges reside in the role of real time operating system. While there exists possibilities for scheduling strategies: partitioned, semi-partitioned, clustered, and global scheduling. Due to the fact that the unrelated processors exist, the task and machine model must be advanced in order to provide better real time analysis. At this point, R processing rate is useful to define both practical consideration (processor affinity, and clustering strategies) and schedulability analysis.

VI. Conclusions

In this paper, parallel task upon heterogeneous machine were analyzed under implicit deadline DAG sporadic task model with a novel extension of R processing rate. By combining parallel tasks analysis upon identical multiprocessor and their *processing rate* upon heterogeneous system, we derived utilization augmentation, which is useful for extending capacity augmentation bound. Our experiments showed that there is schedulable task system, which is characterized by its utilization augmentation upon heterogeneous system under Global EDF with capacity augmentation bound of $\left(4 - \frac{m}{2}\right)\left(1 + \sqrt{U_{Aug}}\right)$. Our model with R processing rate is also useful for practical consideration. Mathematical proof and further analysis are considered as further works.

REFERENCES

[1] Fisher, N., Goossens, J., and Baruah, S. Optimal online multiprocessor scheduling of sporadic real-time tasks is impossible. Real-Time Systems, 45, 1 (2010), pp. 26–71.

[2] Lopez J. M., Diaz J. L., Garcia D. F. Utilization bounds for EDF scheduling on real-time multiprocessor systems. Real-Time Systems, 28, 1 (2004), pp. 39-68.

[3] Liu, C. L., and Layland, J.W. Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the ACM, 20, 1 (1973), pp. 46-61.

[4] Kato, S. Yamasaki, N., Ishikawa, Y. Semi-Partitioned Scheduling of Sporadic Task Systems on Multiprocessors. Proceedings of 21st Euromicro Conference on Real-Time Systems (2009), pp. 249-258.

[5] Bastoni, A., Bjorn, B., Anderson, J. Is Semi-Partitioned Scheduling Practical?. Proceedings of 23rd Euromicro Conference on Real-Time Systems (2011), pp. 125-135.

[6] Baruah, S., Bertogna, M., Butazzo, G. Multiprocessors Scheduling for Real Time Systems. Springer International Publishing. (2015).

[7] Bonifaci V., et. al. Feasibility Analysis in the Sporadic DAG Task Model. Proceedings of 25rd Euromicro Conference on Real-Time Systems (2013), pp. 225-233.

[8] Li J., Agrawal K., Lu C., and Gill C., Analysis of Global EDF for Parallel Task. Proceedings of 23th Euromicro Conference on Real-Time Systems (2013), pp. 3-13.

[9] Raravi G., Andersson B., Bletsas K., and Nelis V., Task Assignment Algorithm for Two-Type Heterogeneous Multiprocessors. Real Time Systems, vol. 50, Issue 1 (2014), pp. 87-141.

[10] Lawler E.L., Labeteoulle J., On Preemptive Scheduling of Unrelated Parallel Processors by Linear Programming. Journal of the ACM, 25, (1978), page no. 612-619.

[11] Sviridenko M., and Wiese A., Approximating the Configuration-LP for Minimizing Weighted Sum of Completion Times on Unrelated Machines. Proceedings of the 16th Conference on Integer Programming and Combinatorial Optimization (IPCO 2013), pp. 387-398.

[12] Cordeiro, D., et. al. Random graph generation for scheduling simulations, Proceedings of he 3rd International ICST Conference on Simulation Tools and Techniques (2010).

[13] "ARM big.LITTLE". https://en.wikipedia.org/wiki/ARM_big.LITTLE. Accessed on July 1st, 2016.

[14] "i.MX". https://en.wikipedia.org/wiki/I.MX. Accessed on July 1st, 2016