Procedia Computer Science

Volume 80, 2016, Pages 1135–1146

ICCS 2016. The International Conference on Computational Science

Applying MGAP Modeling to the Hard Real-Time Task Allocation on Multiple Heterogeneous Processors Problem

Eduardo Valentin, Rosiane de Freitas, and Raimundo Barreto*

Institute of Computing (ICOMP)
Post-graduate Program in Informatics (PPGI)
Federal University of Amazonas (UFAM)
Av. Rodrigo Otávio Jordão Ramos, 6200, Coroado, Zip-code 69077-000, Manaus-Brazil.
{eduardo.valentin,rosiane,rbarreto}@icomp.ufam.edu.br

Abstract

The usage of heterogeneous multicore platforms is appealing for applications, e.g. hard real-time systems, due to the potential reduced energy consumption offered by such platforms. However, the power wall is still a barrier to improving the processor design process due to the power consumption of components. Hard real-time systems are part of life critical environments and reducing the energy consumption on such systems is an onerous and complex process. This paper reassesses the problem of finding assignments of hard real-time tasks among heterogeneous processors taking into account timing constraints and targeting low power consumption. We also propose models based on a well-established literature formulation of the Multilevel Generalized Assignment Problem (MGAP). We tackle the problem from the perspective of different integer programming mathematical formulations and their interplay on the search for optimal solutions. Experimentation shows that using strict schedulability tests as constraints of 0/1 integer linear programming results in faster solvers capable of finding optimum solutions with lower power consumption.

Keywords: Hard Real-time, MGAP, Schedulability, Task Assignment, DVFS

1 Introduction

The power wall is a barrier to improvement in the processor design process due to the power consumption of components. Power consumption has become the primary influence in overall microprocessor design complexity [28] due to ideal geometric scaling and non-ideal electrical scaling. It is no longer viable to simply increase clock speeds of existing designs [13]. Power consumption is a major aspect that limits the performance of computers in different sides of the computing spectrum. The pursuit of energy efficiency is useful for mobile devices to improve operating duration and also helpful for server systems to reduce power bills [9, 10, 11].

*We acknowledge the support granted by CNPq and FAPEAM (project 582/2014 – PRO-TI-PESQUISA).

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.425

Modern computing systems often adopt multiple processing elements to enhance the computing capability and reduce the power consumption, especially for embedded systems [9]. Besides, modern multicore processors for the embedded market are often heterogeneous in nature [4]. Therefore, the heterogeneous multicore platforms have become the *de-facto* solution to cope with the rapid increase of system complexity, reliability, and energy consumption [17]. For instance, on Multi-Processor System-on-a-Chip (MPSoC) platforms, a Field-Programmable Gate Array (FPGA) might appear to provide flexibility to execute tasks/jobs in hardware for acceleration. Discrete Co-sine Transform (DCT), or Fast Fourier Transform (FFT) [9], are examples of offloading processors' workload. Multimedia platforms often contain one general purpose processor and one or more co-processors; e.g. for video codec functionality [12].

Practitioners execute applications with hard deadline restrictions on multiple heterogeneous processors due to the expected energy consumption reduction. Nevertheless, developing software with timing constraints for multiple heterogeneous processors is a complex task. Scheduling becomes especially hard to deal with, particularly under low power constraints. Our approach aims at life-critical hard real-time systems such as Unmanned Aerial Vehicles, Control system in the automotive area, and distributed computing under severe constraints, e.g., tracking and target monitoring, military, and environmental remote monitoring.

The Multilevel Generalized Assignment Problem (MGAP) consists of minimizing the assignment cost of a set of jobs to machines, each having associated therewith a capacity constraint. Each machine can perform a job with different performance states that entail different costs and amount of resources required. The MGAP is originally in the context of large manufacturing systems as a more general variant of the well-known Generalized Assignment Problem (GAP) [15]. In this paper, we correlate MGAP model with the problem of assigning frequencies and distributing hard-real time tasks on heterogeneous processors minimizing energy consumption.

Nowadays processors may be seen as machines with several performance states due to Dynamic Voltage and Frequency Scaling (DVFS) technique. DVFS is a well established power reduction strategy and it has already been a research topic for decades. The premises are the variation of processors' workloads and the quadratic relationship between energy consumption and voltage [6]. The energy consumption depends on dynamic and static energy [28].

Therefore, the problem we are addressing in this paper is: *How to find optimum hard real-time tasks distribution among heterogeneous processors respecting timing constraints and targeting low power consumption?*. The contributions of this work are: (i) the usage of classical MGAP model on this problem that delivers (ii) optimum hard real-time task allocation and optimum frequency to task assignment, (iii) with system energy consumption minimization for (iv) different scheduling policies.

The processor model and task model are defined in Section 2.1 and Section 2.2, respectively. Formulations for different scheduling policies are in Section 3 and their evaluations are in Section 4 and Section 5. Section 6 closes the text with final comments. Effective methods support reducing power bills, improve system reliability, and increase the efficient usage of energy; last but not least, assisting to reduce environmental impacts.

2 System Models

In this section we present the system models. Section 2.1 describes the processor model we consider. We describe the real-time task model in Section 2.2.

2.1 Processor Model

The processor model resembles a Multi-Processor System-On-Chip (MPSoC) architecture, such as Exynos 5 Octa [26]. The system is composed by a set of m processors, $H_1, H_2, \ldots, H_m$. Each core may operate on l different performance states, $S_1, S_2, \ldots, S_l$. The frequency of performance state k is $F(k)$ and the power consumption is $P(k)$. The idle power of processor i is $P_{idle,i}$.

Our proposal can be used with no extra effort on other architectures. We have exercised on multiple heterogeneous clusters [27]. Due to text size limitation, we focus on a platform with m processors that differ between each other in terms of instruction set, i.e. compiling same task may result in different sizes for different processors, but having same performance states.

2.2 Task Model

In the remaining sections of this text we adopt the following notation. A task model $\mathcal{M}$ is a set composed by n tasks τ_j. A task $\tau_j \in \mathcal{M}$, with $j \leq n$, has the properties: fixed priority p_j; worst-case execution cycle $WCEC_j$; worst-case execution time $C_j(f)$, which is a function of frequency f, thus $C_j(f) = \frac{WCEC_j}{f}$; period of execution T_j; deadline D_j, we consider scheduling policies in which $D_j \leq T_j$; response time R_j; set of high priority tasks $hp(j)$ representing the tasks τ_p with priority higher than the priority of τ_j.

3 Models for different Scheduling Policies

The formulations presented so far in the literature assume the Earliest Deadline First (EDF) scheduling policy [22]. EDF allows up to 100 % utilization per processor and expects periods equal to deadlines. We present as follows two additional formulations, applicable and refined for Rate Monotonic (RM) and Deadline Monotonic (DM) scheduling policies.

3.1 MGAP model

Considering the problem characteristics and the set of frequencies of each processor as machine levels, we propose using the MGAP integer programming mathematical formulation. The classical MGAP formulation is listed in Equation 1.

$$\text{Minimize:} \quad \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} c_{ijk}x_{ijk} \tag{1a}$$

$$\text{s.t.:} \quad \sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} = 1, \qquad j \in \{1,\ldots,n\} \tag{1b}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{l} a_{ijk}x_{ijk} \leq b_i, \; i \in \{1,\ldots,m\} \tag{1c}$$

$$x_{ijk} \in \{0,1\}, 1 \leq i \leq m, 1 \leq j \leq n, \qquad 1 \leq k \leq l \tag{1d}$$

where the tri-dimensional decision variable x_{ijk} represents the assignment to processor i of task τ_j at performance state k. The objective function 1a minimizes the system energy consumed by processors. The tri-dimensional matrix c_{ijk} represents the energy consumed by task τ_j while executing on processor i at performance state k. The constraint 1b models the allocation of task τ_j to a single processor. The limit 1c represents each processor capacity constraint. The

tri-dimensional matrix a_{ijk} represents capacity used by task τ_j while executing on processor i at performance state k. The variable b_i represents the capacity of processor i.

The above problem can be described using the three field notation for theoretic scheduling problems $\alpha|\beta|\gamma$ [5]. The machines environment is unrelated parallel machines ($\alpha = R$) because the matrix a_{ijk} depends on the task and machine relation. The job characteristics (β) are deadline (D_j) and preemption ($pmnt$). Also, the jobs have arbitrary execution time (see C_j in Section 2.2). The optimally criteria (γ) is unspecified because we minimize the overall energy consumption. Thus, the scheduling theory notation is $R|Dj; pmnt|\sum f_i$.

3.2 MGAP Formulation with Utilization Bound for EDF ($D_j = T_j$)

EDF is a dynamic priority based on-line scheduler in which earliest deadlines are first scheduled. Liu and Layland [22] propose a utilization based schedulability test. The condition is: $U_{total} \leq U_{bound}(n)$, where $U_{total} = \sum_{j=1}^{n} C_j(f)/T_j$. The test for n tasks for EDF is $U_{bound}(n) = 1$. Thus, b_i of each processor is 1. The MGAP formulation for EDF is expressed in Equation 2.

$$\text{Min:} \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l}\left[\left\lfloor\frac{LCM}{T_j}\right\rfloor C_l WCEC_{ij} V_{dd,k}^2 + P_{idle,i} LCM\left(1 - \frac{WCEC_{i,j}}{F(k)T_j}\right)\right] x_{i,j,k} \tag{2a}$$

$$\text{s.t.:} \sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} = 1,\ j \in \{1,\ldots,n\} \tag{2b}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{l}\frac{WCEC_{ij}}{F(k)T_j} x_{ijk} \leq 1,\ \in \{1,\ldots,m\} \tag{2c}$$

$$x_{ijk} \in \{0,1\}, 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq l \tag{2d}$$

where $a_{ijk} = \frac{WCEC_{ij}}{F(k)T_j}$ represents the task τ_j utilization, u_{ijk}, while executing in processor i at frequency $F(k)$ of performance state k, LCM is the Least Common Multiple of tasks periods, C_l is the circuit capacitance constant, and $V_{dd,k}$ is the voltage level to achieve frequency $F(k)$. In this case, we make $c_{ijk} = \left[\left\lfloor\frac{LCM}{T_j}\right\rfloor C_l WCEC_{ij} V_{dd,k}^2 + P_{idle,i} LCM\left(1 - \frac{WCEC_{i,j}}{F(k)T_j}\right)\right]$.

3.3 MGAP Formulation with Utilization Bound for RM ($D_j = T_j$)

RM is a fixed priority based on-line scheduler in which task priorities decrease with larger periods. RM expects periods equal to deadlines ($T_j = D_j$). Liu and Layland bound for n tasks for RM is $U_{bound}(n) = n(2^{\frac{1}{n}} - 1)$. Therefore, we can propose the MGAP formulation for RM as expressed in Equation 3.

$$\text{Min:} \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l}\left[\left\lfloor\frac{LCM}{T_j}\right\rfloor C_l WCEC_{ij} V_{dd,k}^2 + P_{idle,i} LCM\left(1 - \frac{WCEC_{i,j}}{F(k)T_j}\right)\right] x_{i,j,k} \tag{3a}$$

$$\text{s.t.:} \sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} = 1,\ j \in \{1,\ldots,n\} \tag{3b}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{l}\frac{WCEC_{ij}}{F(k)T_j} x_{ijk} x_{ijk} \leq n_i^*(2^{\frac{1}{n_i^*}} - 1),\ i \in \{1,\ldots,m\} \tag{3c}$$

$$x_{ijk} \in \{0,1\}, 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq l \tag{3d}$$

where n_i^* is the number of tasks assigned to processor i for a given allocation configuration.

3.4 MGAP Formulation with Response Time Bound ($D_j \leq T_j$)

Lehoczky *et al.* [21] present an exact schedulability analysis based on tasks periods and priorities: $R_j \leq D_j, \forall\ 1 \leq j \leq n$, where R_j is computed using the iterative equation $R_j^{n+1} = C_j + \sum_{p \in hp(j)} \left\lceil \frac{R_j^n}{T_p} \right\rceil \times C_p$. Considering the schedulability test proposed by Lehoczky, we can now propose the MGAP formulation using tasks response times as seen in Equation 4. This formulation applies each task deadline as a constraint against their response time in the linear programming. We are not considering precedence or mutual exclusion at this work. However, combining Audsley [2] test with Equation 4 covers for precedence and mutual exclusion.

$$\text{Min:} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} \left[\left\lfloor \frac{LCM}{T_j} \right\rfloor C_l WCEC_{ij} V_{dd,k}^2 + P_{idle,i} LCM \left(1 - \frac{WCEC_{i,j}}{F(k)T_j}\right) \right] x_{i,j,k} \tag{4a}$$

$$\text{s.t.:} \sum_{i=1}^{m} \sum_{k=1}^{l} x_{ijk} = 1,\ j \in \{1, \ldots, n\} \tag{4b}$$

$$R_j^* \leq D_j,\ j \in \{1, \ldots, n\} \tag{4c}$$

$$x_{ijk} \in \{0, 1\}, 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq l \tag{4d}$$

where R_j^* is the response time of tasks τ_j for a given allocation configuration. Equation 4 is applicable for RM scheduling policy ($D_j = T_j$). The above formulation may be also used on DM, a RM extension which allows $D_j \leq T_j$.

4 Computational Experiments

In this section we present the computational experiments. We discuss experiment design and results in Section 4.1 and Section 4.2, respectively.

4.1 Experiment design

We wrote an implementation of the previous models using C++ CPLEX Concert [19] to create a solver for each of them. The solver's input is a configuration file containing the processing model, its power model, and the desired task model. Each solver outputs a boolean, informing if a viable solution is found. Also, when the input has viable solutions, the solver outputs the distribution of hard real-time tasks among the processors that consumes less power among the possible assignments, informing as well in which frequency each tasks may be executed. When asked, the solver may also output the execution time. We set Concert with IloCplex::Threads = 1, IloCplex::WorkMem = 1024, IloCplex::TreLim = 2048, and IloCplex::Param::Parallel = 1.

The objective of this computational experiment is to evaluate the model in terms of performance, objective function, and execution time. The performance of each solver is determined by the amount of valid solutions it is able to find for a given input. The evaluation of solver's objective function aims to understand the ability to reduce energy consumption for a given input. Finally, the execution time is measured to compare the impact on project design phase. We estimate the execution time of each solver using wall clock. The machine used to perform the simulation experiments has an AMD FX ™-9370, 8 cores executed at 1.4 GHz. The machine has 32 GB of DDR3 memory executed at 1.33 GHz and executes Debian GNU/Linux 8.2.

We use random-generated task models. We vary number of tasks in each random-generated models between 5, 10, 15, and 20. We vary the total system utilization between

10%, 20%, 30%, 40%, 50%, and 60%. The total system utilization is estimated using the highest frequency available. The tasks WCEC is uniformly distributed between 50 000 and 100 000 cycles, but a task may have a different amount of cycles for each processor, representing their difference in instruction set. We chose to have tasks periods uniformly distributed between three sets: large period, medium period, and short period. When a task has a large period, its period is chosen from a normally distributed variable between 100ms and 1000ms. Similarly, a medium period is normally distributed between 10ms and 100ms, and a short period is normally distributed between 1ms and 10ms. We choose to make deadlines equal to periods and to disregard precedence and shared resources to perform a fair comparison among policies.

Our experiment considers a target platform of four processors. Processors are based on XScale processors [20]. Each processor may operate at 624 MHz, 520 MHz, 416 MHz, 312 MHz, or 208 MHz. The idle power consumption is 260 mW. The energy consumption is estimated for the duration of the LCM of tasks' periods, as in the objective function of each model.

We chose to split the evaluation into two groups. In group A, we have 200 random-generated task models per combination of number of tasks and total system utilization, regardless of their feasibility. While in group B, we have 30 feasible random-generated task models for each combination of number of tasks and total system utilization. In group B, each considered model has a solution generated by each solver.

4.2 Results

In group A, we evaluate the performance of each model. Figure 1 represents a comparison of the feasibility rate of each solver. As it can be seen, the feasibility rate decreases, for all solvers, with the rise of the number of tasks and when the total system utilization is higher. The solver for Rate Monotonic using utilization bounds (RM_UTIL) has shown a lower feasibility rate when compared to the solver for Rate Monotonic using response time bounds (RM_RESP). The solver for EDF with utilization bounds (EDF_UTIL) has the highest feasibility rate.

Figure 2 illustrates the mean energy consumption of solutions provided by each solver in group B. The energy consumption rises when the number of tasks increases or when the total system utilization increases. RM_RESP shows a curve of energy consumption that grows slower than the energy curve of RM_UTIL. Because RM_RESP explore more feasible assignment configurations, it minimizes further the objective function finding configurations with lower energy consumption. EDF_UTIL presents the same energy curve as RM_RESP.

Figure 3 illustrates the utilization of solutions provided by each solver on group B. We observe solutions produced by RM_RESP have higher system utilization than those solutions provided by RM_UTIL. Because RM_RESP explores more configurations than RM_UTIL, it considers configurations with lower frequencies for each tasks, increasing their execution time, but still reducing the overall system consumption. Such configurations with lower frequencies may be discarded by RM_UTIL because they exceed the utilization bound for Rate Monotonic. EDF_UTIL presents the same curve as RM_RESP.

Figure 4 illustrates the results of wall clock execution time of each solver on group B. In this experiment, we have limited the execution time of each run to 30 minutes for practical reasons. The execution time of all solvers increases when the number of tasks increases or when the total system utilization is high. The execution time of RM_RESP is lower than equal to the execution time of RM_UTIL. Once again, because RM_RESP considers feasible configurations with lower frequencies, its solver is capable to find optimum solutions faster than RM_UTIL. However, the fastest solver is EDF_UTIL.

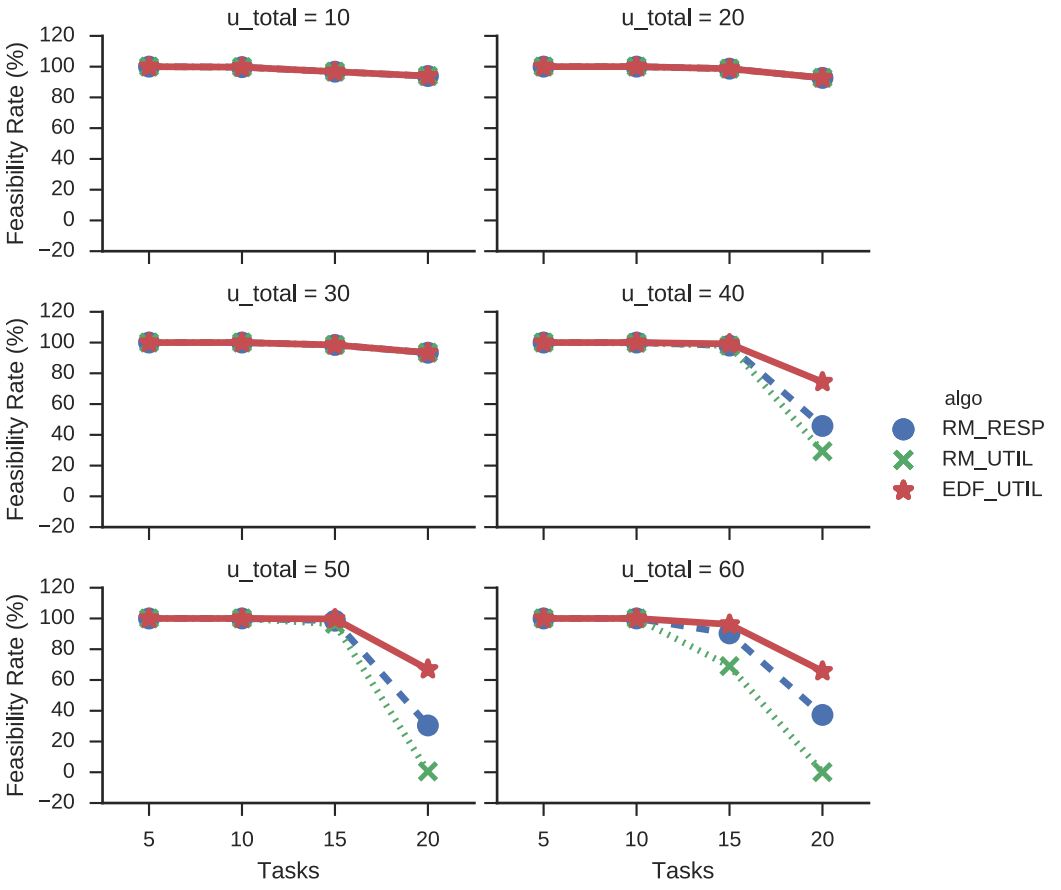

Figure 1: Group A: Feasibility rate of each solver. The RM_RESP finds more feasible solutions than RM_UTIL. EDF_UTIL has the highest feasibility rate.

5 Related Work and Discussion of Results

We have tested the hypothesis of reducing overall system energy consumption by using strict constraints [27]. The interplay between different schedulability analyses and linear integer programming have influence on the performance of solvers.

In the literature, there are strategies to determine hard real-time task distribution in heterogeneous platforms. Their approach typically focus on either heuristics or approximation algorithms [9, 10, 11]. There are also models proposed to cover optimal solutions that minimize the energy consumption of hard real-time systems with multiple heterogeneous processors. The typical formulation is a 0/1 integer linear programming model which considers a continuous processor frequency domain and determines a single operating frequency per processor [1, 4, 8, 9, 10, 11, 12, 17]. GLPK [14] or CPLEX [19] can be used to deriving optimum solutions from their formulations. We propose using the MGAP model instead because it brings a better fit to the problem, considering that practical processors still use a discrete set of frequencies.

The adoption of DVFS is common in optimization procedures, such as task allocation,

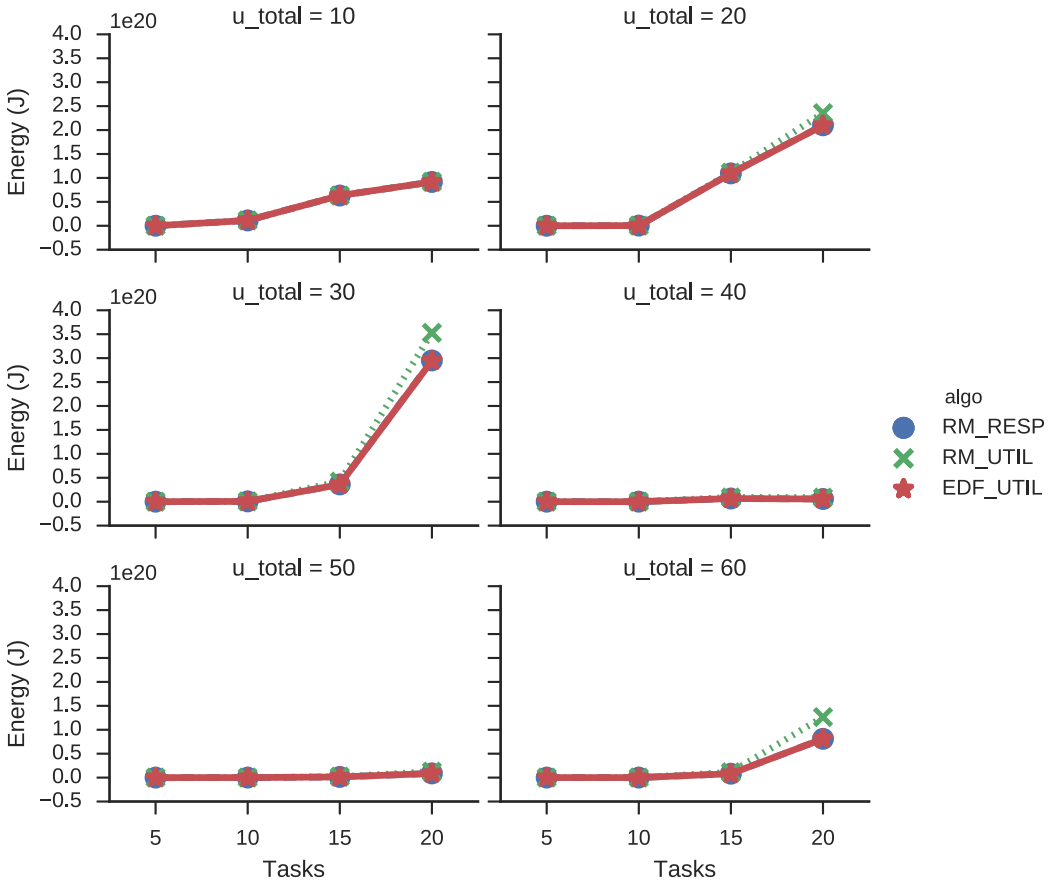

Figure 2: Group B: Energy consumption curves. RM_RESP finds more configurations with lower energy consumption than RM_UTIL. RM_RESP has the same energy curve as EDF_UTIL.

and frequency to task assignment. The aim is to find optimal energy-aware scheduling on heterogeneous platforms while considering individual task deadlines [9]. Overall system energy reduction is due to workload split and to frequency minimization to meet tasks deadlines. Adoption of the well known utilization based schedulability analyses is common [1, 4, 8, 9, 10, 11, 12, 16, 17, 18, 24, 25, 29, 30]. The simplification on the modeling and solving process as using utilization constraints produces formulations similar to the multiple knapsack problem. We confirm these efforts with EDF_UTIL solver, the fastest in our experiments. However, our work also propose formulations for different scheduling polices for practical processors.

For the MGAP there are extreme fast algorithms. MGAP problem instances are solvable up to hundreds of machines with tens of speed levels, to map hundreds of tasks. For example, Osorio and Laguna [23] proposed a Branch-and-Cut algorithm in 2003 and were able to solve instances up to 60 tasks, 30 machines, and two speed levels. Ceselli and Righini [7], in 2006, proposed a Branch-and-Price strategy, solving up to 400 tasks, 80 machines and five levels. Another Branch-and-Cut algorithm proposed by Avella *et al.* in 2013 [3] can solve 200 tasks, 30 machines and five levels, for specific problem instances. We contribute with the present work

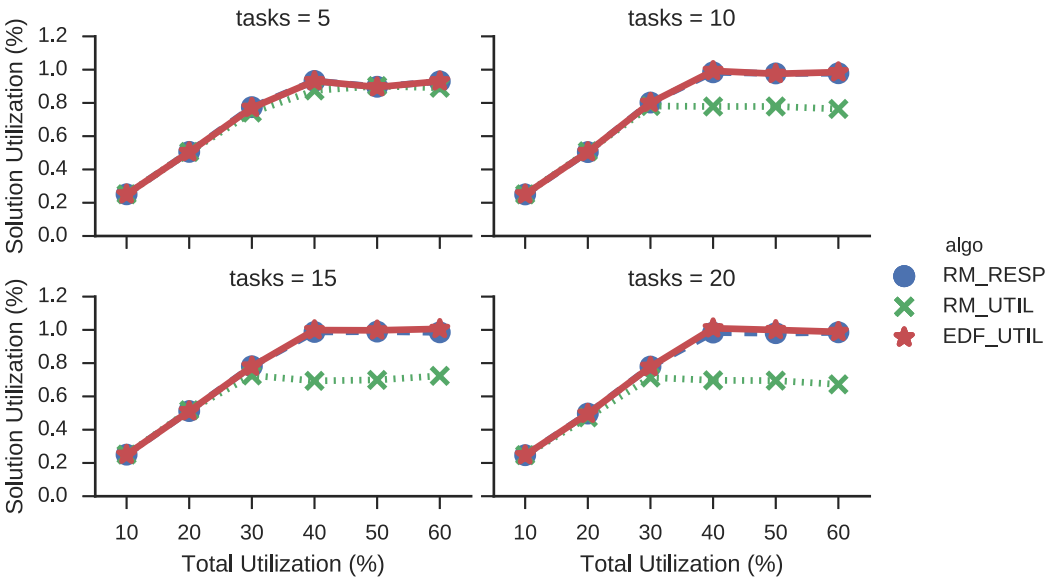

Figure 3: Group B: evaluation of solutions provided by each solver. RM_RESP is able lower the frequencies of tasks more than RM_UTIL. The consequence is RM_RESP produces solutions with higher system utilization. EDF_UTIL presents the same curve as RM_RESP.

with a new application of the MGAP model.

On static priority policies, using response time schedulability test for Rate Monotonic produces a faster solver which finds solutions with lower power consumption when compared to utilization schedulability test for Rate Monotonic. RM_RESP finds feasible configurations composed by lower frequencies. The consequence is a two fold benefit: (i) the RM_RESP solver is able find optimum solutions faster; (ii) the RM_RESP also produces configurations with lower energy consumption. We also observe the solutions found by RM_RESP have a larger total system utilization. Therefore, we suggest the usage of response time schedulability test instead of utilization test on static priority policies, represented by RM in these experiments.

6 Final Remarks

In this paper we assess the problem of how to find optimum hard real-time tasks distribution among heterogeneous processors respecting timing constraints and minimizing low power consumption. Our study focus on optimum solutions after reviewing the existing models.

We proposed the usage of a well-known classical combinatorial optimization formulation, MGAP, to represent the characteristics and constraints of the problem. We first used MGAP to model systems using EDF ($D_j = T_j$), which represents the existing models in the literature. We also extended the proposed MGAP to derive extra models considering utilization bounds for RM ($D_j = T_j$) and response time analysis for RM ($D_j = T_j$) and DM ($D_j \leq T_j$) scheduling policies. The implementation of such models delivers optimum hard real-time task allocation and optimum frequency to task assignment minimizing system energy consumption. Based on our experimental results we recommend for static priority policies the usage of MGAP

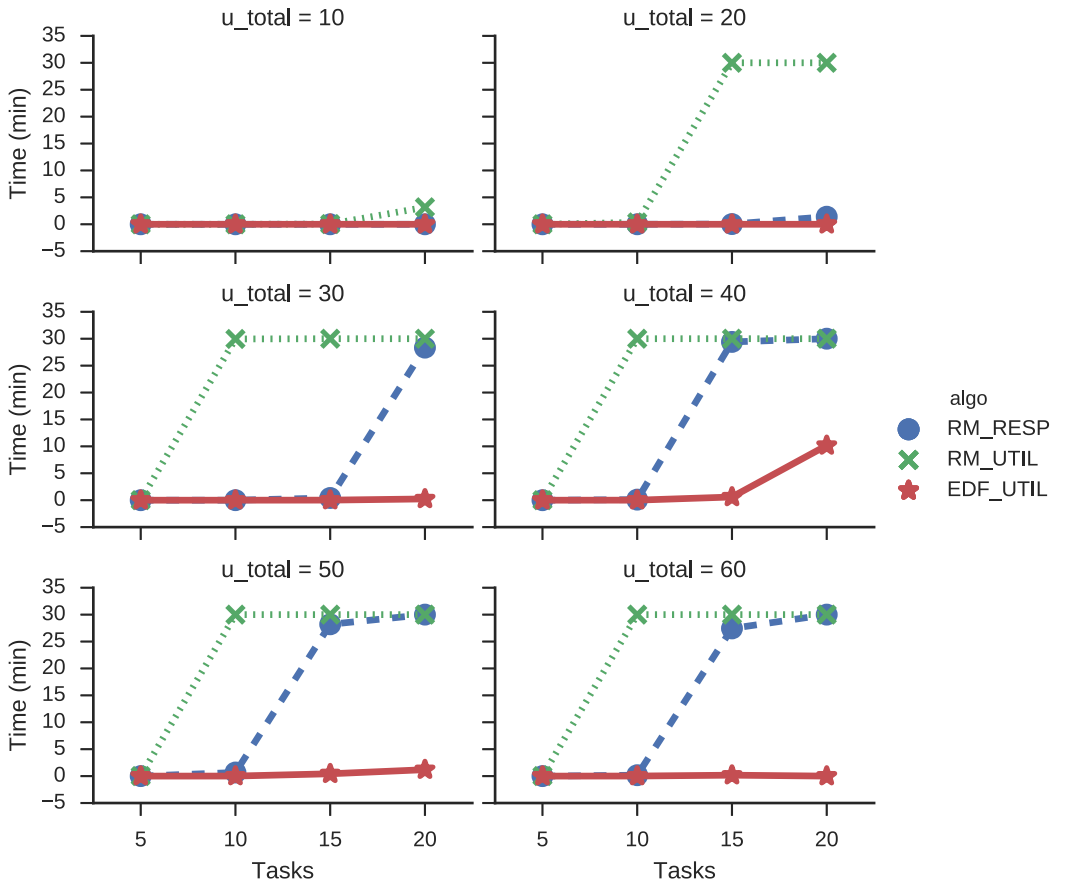

Figure 4: Group B: EDF_UTIL is the fastest solver in this experiment. RM_RESP is faster than RM_UTIL for being able to consider feasible configurations with lower frequencies.

response time based model. The MGAP response time based model, implemented for RM in our experiments, finds configurations with the same energy consumption as the literature EDF representation, but with an execution time penalty.

As future works, we will investigate different algorithms for MGAP applied on the problem of assigning hard real-time tasks among heterogeneous processors with different performance state. We believe that combining the response time analysis with the existing algorithms may produce faster solvers for this problem. We also understand the theoretical limitations of RM and DM policies, however, the usage of response time analysis in 0/1 integer linear programming models presented in this paper can be used as baseline enabling further studies on precedence and mutual exclusion constraints.

References

[1] B. N. Alahmad and S. Gopalakrishnan. Energy efficient task partitioning and real-time scheduling on heterogeneous multiprocessor platforms with QoS requirements. *Sust. Computing: Inform. and*

Systems, 1(4):314–328, December 2011.

[2] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Eng. Journal*, 8(5):284–292, Sep 1993.

[3] P. Avella, M. Boccia, and I. Vasilyev. A branch-and-cut algorithm for the multilevel generalized assignment problem. *Access, IEEE*, 1:475–479, 2013.

[4] M. A. Awan and S. M. Petters. Energy-aware partitioning of tasks onto a heterogeneous multi-core platform. In *Real-Time Technology and App. - Proc.*, pages 205–214, Philadelphia, PA, 2013.

[5] Peter Brucker. *Scheduling Algorithms*. Springer Publishing Company, Incorporated, 5th edition, 2010.

[6] T. D. Burd and R. W. Brodersen. Processor design for portable systems. *Journal of VLSI signal processing systems for signal, image and video technology*, 13(2-3):203–221, 1996.

[7] A. Ceselli and G. Righini. A branch-and-price algorithm for the multilevel generalized assignment problem. *Operations Research*, pages 1172–1184, 2006.

[8] H. Chen, A. M. K. Cheng, and Y.-W. Kuo. Assigning real-time tasks to heterogeneous processors by applying ant colony optimization. *Journal of Parallel and Distributed Computing*, 71(1):132–142, January 2011.

[9] J.-J. Chen, A. Schranzhofer, and L. Thiele. Energy minimization for periodic real-time tasks on heterogeneous processing units. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–12, May 2009.

[10] J.-J. Chen and L. Thiele. Task Partitioning and Platform Synthesis for Energy Efficiency. *2009 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 393–402, August 2009.

[11] J.-J. Chen and L. Thiele. Platform synthesis and partitioning of real-time tasks for energy efficiency. *Journal of Systems Architecture*, 57(6):573–583, 2011.

[12] Y.-S.a Chen, L.-P.b Chang, and T.-W.c Kuo. Multiprocessor frequency locking for real-time task synchronization. In *Proceedings of the ACM Symposium on Applied Computing*, pages 289–293, Fortaleza, Ceara, 2008.

[13] R. Cochran. *Techniques for Adaptive Power and Thermal Sensing and Management of Multi-core Processors by*. Thesis, Brown University, 2013.

[14] Free Software Foundation. Gnu linear programming kit. http://www.gnu.org/software/glpk/glpk.html, December 2012.

[15] F. Glover, T. J. Hultz, and D. Klingnian. Improved computer-based planning techniques, part ii. Interfaces 9/4, 1979.

[16] J. Goossens, D. Milojevic, and V. Nélis. Power-aware real-time scheduling upon dual cpu type multiprocessor platforms. In *Proc. of the 12th Int. Conference on Principles of Distributed Systems*, OPODIS '08, pages 388–407, Berlin, Heidelberg, 2008. Springer-Verlag.

[17] D. He and W. Mueller. Enhanced schedulability analysis of hard real-time systems on power manageable multi-core platforms. In *Proceedings of the 14th IEEE International Conference on HPCC - 9th IEEE ICESS*, pages 1748–1753, Liverpool, 2012.

[18] D. He and W. Mueller. A heuristic energy-aware approach for hard real-time systems on multi-core platforms. *Microprocessors and Microsystems*, 37(8, Part A):858 – 870, 2013. Special Issue {DSD} 2012 on Reliability and Dependability in {MPSoC} Technologies.

[19] IBM. Ilog cplex. http://www.ilog.com/products/cplex/.

[20] Intel Corp. Intel xscale. http://developer.intel.com/design/xscale/, March 2003.

[21] J. P. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: exact characterization and average case behavior. *Proceedings of the IEEE Real Time Systems Symposium*, pages 166–171, 1989.

[22] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM*, 20(1):46–61, January 1973.

[23] M. A. Osorio and M. Laguna. Logic cuts for multilevel generalized assignment problems. *European Journal of Operational Research*, 151(1):238 – 246, 2003.

[24] V. K. Prasanna and Y. Yu. Power-aware resource allocation for independent tasks in heterogeneous real-time systems. In *Ninth International Conference on Parallel and Distributed Systems, 2002. Proceedings.*, pages 341–348, 2002.

[25] K. Prescilla and A. Immanuel Selvakumar. Modified binary particle swarm optimization algorithm application to real-time task assignment in heterogeneous multiprocessor. *Microprocessors and Microsystems*, 37(6-7):583 – 589, 2013.

[26] Samsung Electronics Co.Ltd. Samsung exynos. http://www.samsung.com/global/business/semiconductor/minisite/Exynos/products5octa_5410.html, March 2014.

[27] E. Valentin, M. Salvatierra, R. de Freitas, and R. Barreto. Response time schedulability analysis for hard real-time systems accounting dvfs latency on heterogeneous cluster-based platform. In *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2015 25th International Workshop on*, pages 1–8, Sept 2015.

[28] V. Venkatachalam and M. Franz. Power reduction techniques for microprocessor systems. *ACM Comput. Surv.*, 37(3):195–237, 2005.

[29] C.-Y. Yang, J.-J. Chen, T.-W. Kuo, and L. Thiele. An approximation scheme for energy-efficient scheduling of real-time tasks in heterogeneous multiprocessor systems. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, pages 694–699, April 2009.

[30] Y. Yu and V. K. Prasanna. Resource allocation for independent real-time tasks in heterogeneous systems for energy minimization. *Journal of Information Science and Eng.*, 19(3):433–449, 2003.