A Game Theory Approach to Heterogeneous Resource Management: Work-in-Progress

Lara Premi[§], Federico Reghenzani[†*], Giuseppe Massari[†], William Fornaciari[†]
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy
Email: [§]lara.premi@mail.polimi.it, [†]`[name].[surname]`@polimi.it
[*] Corresponding Author

***Abstract*—Heterogeneous computing is a promising solution to scale the performance of computing systems maintaining energy and power efficiency. Managing such resources is, however, complex and it requires smart resource allocation strategies in both embedded and high-performance systems. In this short paper, we propose a game theory approach to allocate heterogeneous resources to applications, with a focus on performance, power, and energy requirements. The game congestion model has been selected and a cost function designed. The proposed allocation strategy is then evaluated by performing a preliminary experimental evaluation.**

***Index Terms*—resource management, heterogeneous computing, game theory**

I. Introduction

In the last years, heterogeneous computing systems emerged as an effective approach to scale performance in an energy-efficient manner. Always more frequently, both embedded and HPC systems come with computing platforms, including CPU cores with heterogeneous performance/power consumption profiles or additional processing units like GPUs and custom accelerators. However, these configuration introduced further complexity in the *task scheduling* and *resource allocation* problems, by increasing the size of the solution space [11] [3]. This complexity requires proper resource management strategies to efficiently allocate the resources [1]. In this paper, we explored the use of Game Theory algorithms for solving the problem of allocating heterogeneous resources to multiple applications, with the objective of managing the performance/power/energy trade-off. We modeled the resource allocation problem to a modified version of a game-theoretical approach, and we created a cost function able to deal with the trade-off mentioned above.

A. Game Theory

The branch of applied mathematics that studies the interaction among players in a well-defined game is called Game Theory. A game is the representation of an everyday life situation with two or more rational players, interacting according to a set of well-specified actions, intending to achieve their own best profit. Games are usually categorized in cooperative ones and non-cooperative: in the former, the players agree on a common strategy to obtain the best global output, while the latter refers to the games creating a contention among the players. The usual goal of non-cooperative games, the subject of this work, is to find the *Nash Equilibrium*, i.e., the situation in which no one of the players has an incentive to play a different strategy if the others are playing the equilibrium strategy. The Nash Equilibrium is not necessarily the global optimum, however, it guarantees that no player can unilaterally improve its position.

B. Related Works

The idea to use game theory approaches to resource management of computing resources gained interest in the scientific community in the last decade. However, most of the works focused on the distributed and cloud computing scenarios [4], [5], [7], [8], [13]–[16]. Both cooperative games [5], [7], [8] and non-cooperative games [4], [13], [15], [16] have been considered. Some of them [5], [15] dealt with energy metrics, trying to find a strategy that minimizes the total energy consumption of the system. Many approaches considered also the timing and performance aspects [4], [7], [13], [15]. To the best of our knowledge, a game theory resource allocation approach for local and heterogeneous machines, that takes into account the multi-objective optimization performance/power/energy, has not been explored yet.

II. Allocation Policies based on Games

We selected two classes of games that are applicable to the problem of resource management in the heterogeneous scenario: the *congestion games* and the *double auctions*. The congestion games [12] are a special class of potential games that are, in turn, non-cooperative games. A set of players compete for the usage of one or more resources. The latter typology, double auctions, is also a class of non-cooperative games, and it is derived from the financial world. In this case, the players are divided into buyers and sellers that agree with the price for a good. Each of the parties proposes an initial offer (bid), which is negotiated according to some specific criteria. The most famous algorithms are VCG and McAfee.

We are currently exploring both game theory models, congestion games and double auctions, to build resource management strategies. In this short paper, we limit our discussion to a policy based on a congestion game, because it is, currently, the most mature approach, leaving the double auction scenario as future works.

A. Congestion Games

Modeling the resource management problem with congestion games is quite straightforward. In a congestion game, the players compete for a set of *congestible elements*. Our model is then composed of players, i.e. tasks $\bar{T} = \{T_1, T_2, ..., T_n\}$ and resources. In this paper, we limit this discussion to CPU cores $\bar{C} = \{C_1, C_2, ..., C_m\}$, but they can be any computational resource. The set $\bar{A} = \{A_1, A_2, ..., A_k\}$ is the set of possible actions, i.e. the possible resource assignments (T_i, C_j). The congestion game algorithm solves the optimization problem $\min_{\bar{A}^*} \phi(\bar{A}^*)$ to find one Nash equilibrium (more than one equivalent Nash equilibrium may exist). The set $\bar{A}^* \subset \bar{A}$ is the resource assignment and the symbol $\phi(x)$ represents the potential function $\phi(\bar{A}^*) = \sum_{j=1}^{m} \sum_{k=1}^{cong_j(\bar{A}^*)} c_j(k)$, where $cong_j(\bar{A}^*)$ is the *congestion*, i.e. the number of tasks assigned to the j-th resource. In addition to the traditional congestion game problem, we introduce the percentage of usage of each resource for each task. In fact, being heterogeneous the CPU cores, not the same amount of *share* is required to achieve the performance requirements of the tasks. The percentage of the i-th resource used by the j-th task is represented with the symbol $\eta_{i,j}$. We defined the cost function $c_i(n)$ as follows: $c_i(n) = \sum_{j=1}^{n} \bar{c}_i(j) \cdot \eta_{i,j}$ and the $\bar{c}_i(n)$ vector is:

$$\bar{c}_i(n) = \begin{cases} \alpha_i + \beta_i + \gamma_i^2 & \text{if } n = 1 \\ \alpha_i[n + \alpha_{i,0}(n-1)] + \beta_i[1 + \beta_{i,0}(n-1)] & \text{if } n > 1 \end{cases}$$

where $\alpha_i, \beta_i, \gamma_i$ are the coefficients for, respectively, the performance, the energy and the power. They are computed as: $\alpha_i = \alpha' \cdot \frac{1}{f_i}$, $\beta_i = \beta' \cdot PDP_i$, and $\gamma_i = \gamma' \cdot P_i$, where f_i is the processor frequency, PDP_i is the Power-Delay-Product (a measure of energy-per-operation of the processor), and P_i is the average power consumption. The coefficients α', β', γ' are selected by the user, that tunes the resource management policy to balance the desired multi-objective goals. The coefficients $\alpha_{i,0}$ and $\beta_{i,0}$ are, instead, related to the overhead introduced due to assigning different tasks concurrently on the same resource. The coefficient γ_i is present only for the first assignment, i.e. when the congestion value is 1, because the power is already “consumed” by the first task: assigning one task it means power-up the resource, but assigning more than one task does not change the power level of the system, it would increase the energy consumption.

Solving the problem is performed via the Better-Reply algorithm. Our implementation of the before-mentioned algorithm has an exponential complexity: $O(n^3 m^n log(n))$, where n is the number of tasks and m the number of resources. By testing the algorithm on a standard laptop, it results in being able to carry out the result in the 20s when $n = m = 20$. The execution time can be further reduced by applying an approximation factor ε to the algorithm, which gradually reduces the execution time, but also the solution optimality.

III. Preliminary Experimental Evaluation

A. Experimental Setup

To simulate a heterogeneous platform, we configured a quad-core Intel i7-3770 x86 machine by varying the clock frequency of each core. We measured the power consumption of each core via Intel RAPL, to find the values of P_i and PDP_i,

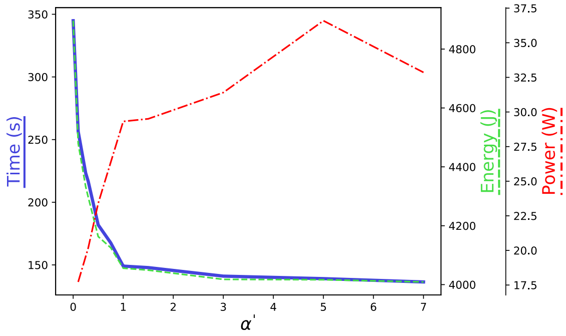

Fig. 1. The overall execution time, energy and maximum power of the task set (15 *blackscholes* and 15 *facesim*), by varying the α' coefficient and maintaining $\beta' = \gamma' = 1$.

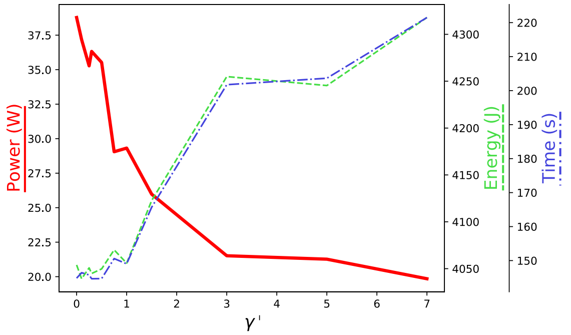

Fig. 2. The overall execution time, energy and maximum power of the task set (15 *blackscholes* and 15 *facesim*), by varying the γ' coefficient and maintaining $\alpha' = \beta' = 1$.

while the frequencies were set as $f_i = [3.4; 2.9; 2.2; 1.6]$ GHz. Alternatively, the P_i values can also be obtained by probabilistic methods [10]. To test the policies with real data, we considered the benchmarks *blackscholes* and *facesim* of the well-known Parsec benchmark suite [2]. The percentage $\eta_{i,j}$ has been computed by setting a Quality-of-Service goal for each benchmark and measuring it on the various cores, similarly to what we performed in [9]. The maximum power and overall energy consumption of each core have been measured via Intel RAPL. We tested the perfect algorithm only, i.e., with $\varepsilon = 0$, which finds the exact Nash Equilibrium.

B. Validation of the coefficients

The first set of tests is focused on understanding the effectiveness of the cost function terms, i.e. α', β', γ', by running 15 *blackscholes* and 15 *facesim* tasks. The effect on the overall execution time, energy, and maximum power of the *performance* coefficient is shown in Figure 1. The larger the α' value, the better the performance (i.e., the smaller the execution time). The energy follows the same trend, while the improvement is at the expense of power consumption. The effect on execution time, energy, and power of the *power* coefficient is shown in Figure 2. The larger the γ' value, the lower the power consumption. Conversely, the energy and time follow the same opposite trend. The plot for β' value is not

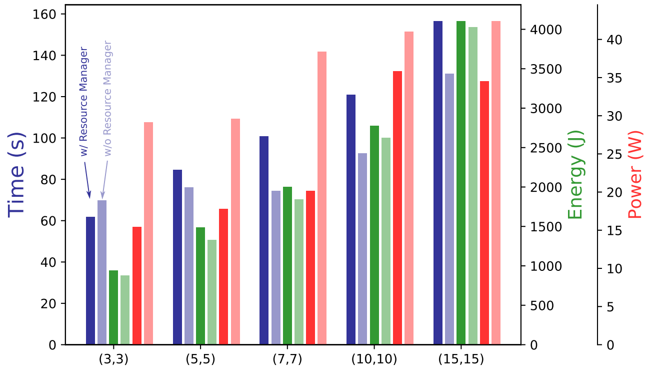

Fig. 3. Comparison of the time, energy and power, with and without the resource manager. The x-axis shows the tasks configuration (nr. of *blackscholes*, nr. of *facesim*).

shown for lack of space and because it is not as effective as expected. This is due to the fact that the cost function does not take into account the benchmark execution time, making the energy estimation via PDP very approximate. Unfortunately, congestion games do not have a way to specify per-task cost functions. This is one of the reasons why we are also investigating the auction games category, which makes it possible to specify per-task parameters. For the sake of completeness, we also tested the EDP (Energy-Delay-Product), without obtaining significant improvements over PDP.

C. Policy effectiveness

Finally, the last experiment compares the usage of the resource manager (with a balanced policy $\alpha' = \beta' = \gamma' = 1$) versus using a random policy. We scaled the number of tasks, from (3,3) to (15,15), selecting in this way a balanced workload among the two benchmarks. The result is depicted in Figure 3. It is immediately clear that without the resource manager, the scheduler does not control the maximum power consumption. Compared to the random policy, the congestion game worsens a little the performance, but it considerably reduces the power consumption, and it maintains the energy mostly unvaried. For example, in scenario $(7, 7)$, the maximum power consumption is doubled if the benchmarks are executed using the standard Linux policy, while the resource manager, governed by the congestion game policy, can keep it under control. By varying the coefficients α', β', γ', the proportion among the three entities change, according to the user willing.

IV. Conclusions

In this work, we modeled the resource allocation problem for heterogeneous platforms as *congestion game*. A cost function has been proposed, and an experimental evaluation showed that the game could be effective, especially when the system is oversubscribed, i.e., the number of tasks is larger than the number of resources. Even if this work is preliminary, it shows the potential of the proposed policy, and future research works can explore different directions, including the in-depth analysis of congestion games and the study of other game theory models, e.g., the double auctions.

Acknowledgments

This work was funded by the European H2020 project RECIPE [6] (grant nr. 801137).

References

[1] G. Agosta, W. Fornaciari, G. Massari, A. Pupykina, F. Reghenzani, and M. Zanella. Managing heterogeneous resources in hpc systems. In *Proceedings of the 9th Workshop and 7th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms*, PARMA-DITAM '18, page 7–12, New York, NY, USA, 2018. Association for Computing Machinery.

[2] C. Bienia, S. Kumar, J. Pal Singh, and K. Li. The parsec benchmark suite: Characterization and architectural implications. In *17th international conference on Parallel architectures and compilation techniques*, pages 72–81, 2008.

[3] R. Canal, C. Hernandez, R. Tornero, A. Cilardo, G. Massari, F. Reghenzani, W. Fornaciari, M. Zapater, D. Atienza, A. Oleksiak, W. Piatek, and J. Abella. Predictive reliability and fault management in exascale systems: State of the art and perspectives. *ACM Comput. Surv.*, 2020. In Publishing.

[4] H-L Chen, J.R. Marden, and A. Wierman. On the impact of heterogeneity and back-end scheduling in load balancing designs. In *INFOCOM*, pages 2267–2275. IEEE, 2009.

[5] M. Dong, T. Lan, and L. Zhong. Rethink energy accounting with cooperative game theory. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 531–542. ACM, 2014.

[6] W. Fornaciari, G. Agosta, D. Atienza, C. Brandolese, L. Cammoun, L. Cremona, A. Cilardo, A. Farres, J. Flich, C. Hernandez, M. Kulchewski, S. Libutti, J.M. Martínez, G. Massari, A. Oleksiak, A. Pupykina, F. Reghenzani, R. Tornero, M. Zanella, M. Zapater, and D. Zoni. Reliable power and time-constraints-aware predictive management of heterogeneous exascale systems. In *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*, SAMOS '18, page 187–194, New York, NY, USA, 2018. Association for Computing Machinery.

[7] M. M. Hassan, B. Song, and E.N. Huh. Game-based distributed resource allocation in horizontal dynamic cloud federation platform. In *Int. Conference on Algorithms and Architectures for Parallel Processing*, pages 194–205. Springer, 2011.

[8] V. Mazalov, N. Nikitina, and E. Ivashko. Hierarchical two-level game model for tasks scheduling in a desktop grid. In *6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, pages 541–545. IEEE, 2014.

[9] F. Reghenzani, G. Massari, and W. Fornaciari. Mixed time-criticality process interferences characterization on a multicore linux system. In *2017 Euromicro Conference on Digital System Design (DSD)*, pages 427–434, 2017.

[10] F. Reghenzani, G. Massari, and W. Fornaciari. A probabilistic approach to energy-constrained mixed-criticality systems. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2019.

[11] F. Reghenzani, G. Massari, and W. Fornaciari. The Real-Time Linux Kernel: A Survey on PREEMPT_RT. *ACM Comput. Surv.*, 52(1), February 2019.

[12] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, dec 1973.

[13] F. Teng and F. Magoules. Resource pricing and equilibrium allocation policy in cloud computing. In *2010 10th IEEE International Conference on Computer and Information Technology*, pages 195–202. IEEE, 2010.

[14] X. Xu and H. Yu. A game theory approach to fair and efficient resource allocation in cloud computing. *Mathematical Problems in Engineering*, 2014.

[15] B. Yang, Z. Li, S. Chen, T. Wang, and K. Li. Stackelberg game approach for energy-aware resource allocation in data centers. *IEEE Transactions on Parallel and Distributed systems*, 27(12):3646–3658, 2016.

[16] D. Ye and J. Chen. Non-cooperative games on multidimensional resource allocation. *Future Generation Computer Systems*, 29(6):1345–1352, 2013.