A New Solution based on Multi-objective Algorithm for Multi-application Mappings for Many-Core Systems

M. A. Almeida, I. F. Gallon, and E. C. Pedrino

***Abstract*— Mapping multiple applications onto intra-chip communication structures, such as Network-on-Chip (NoC), commonly used in the context of manycore systems, is a problem classified as NP-hard. This is due to the need for the simultaneous optimization of performance, reliability, and energy efficiency metrics. For instance, it is essential to account for heat dissipation effects throughout the system, while also ensuring fault tolerance and proper load balancing. While there is still a limited number of works in the literature addressing the allocation of multi-applications in NoCs, many studies focus on the mapping of single applications. Therefore, this paper proposes a multi-objective mapping model that targets performance metrics and heat distribution for multi-application scenarios in manycores with NoCs, aiming to contribute to this area. The results obtained are compared with a state-of-the-art algorithm for this scenario, showing promising improvements, with more than 30% reduction in latency and 38% increase in fault tolerance when all applications are considered.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9346

***Index Terms*—many-core, network-on-chip, task mapping, metrics.**

I. Introduction

TECHNOLOGICAL advances in the manufacture of ICs (Integrated Circuits) have enabled the integration of a greater number of components onto a single chip. This progress has led to the construction of devices with an increasing number of processing cores, especially in embedded systems. For some years now, multicore and manycore devices have been utilizing lower operating frequencies [1]–[3] because the performance demands can no longer be met by increasing the frequency of single-core operation alone [4]–[7]. Another challenge in this scenario is the distribution of mono-application or multi-application tasks among these cores, which is not trivial to solve in practice. Depending on the arrangement chosen, latency and/or power distribution throughout the chip may be compromised. In manycore devices, the basic principle used to address this problem involves dividing applications into smaller tasks that can be efficiently allocated to different cores for parallel execution [8], [9]. This approach allows for better synchronization control, memory management, and reliability in executing parallel code [10], [11]. Therefore, mapping application tasks in these systems involves establishing a communication scheme between them, aiming to optimize hardware variables such as energy consumption, computational performance, voltage, frequency, and others, as exemplified in [12]. However, the optimization involving two or more objectives in multi-applications for NoC mapping can be further explored.

Furthermore, as stated in [13]–[15], to optimize communication delays and energy consumption, tasks should be mapped to the same core or close to each other. These optimizations are necessary to meet the constraints of each application. Thus, there is a need to develop more efficient mapping methodologies that can provide optimal mappings satisfying the demands of different applications. This problem is classified as NP-hard [13], so heuristics based on application domain knowledge must be employed to find an optimized solution.

Current strategies for task distribution in systems with many cores depend on the number of applications to be executed. In scenarios with fixed or variable workloads, improvements to task distribution in these architectures are made during the design or execution phase, respectively. These architectures can be uniform (with identical cores) or heterogeneous (with different cores). During runtime distribution, one core manages tasks such as scheduling, resource control, configuration, and migration. This distribution can be centralized, distributed, or a combination of both [8], [16], [17]. Distributions made during the design phase are best suited for static workloads, where a predefined set of applications with known computing and communication behaviors are considered. Dynamic changes to applications while running [18], [19] are not supported on these systems.

In [24] combine the benefits of evolution-based search with a learning-based local search to quickly determine the PE and communication link placement to optimize multiple objectives (e.g., latency, throughput, and energy) in 3D NoC-enabled manycore heterogeneous systems. It is a promising technique that can be benchmarked in future work.

[25] is a proposed technique that uses an ML-based model to extract relevant information from research data and incorporate it into the research process. This results in a more robust model with a higher convergence rate and solution quality.

The associate editor coordinating the review of this manuscript and approving it for publication was Ruth Aguilar (*Corresponding author: Manoel de Almeida*).

E. C. Pedrino is grateful to FAPESP (Grant 2017/26421-3 and 2023/00212-0) and CAPES.

Manoel de Almeida, I. F. Gallon, and E. C. Pedrino are with Federal University of Sao Carlos, Rodovia Washington Luis, São Carlos, Brazil (e-mails: manoel.aranda@ufscar.br, igorfelipegallon@gmail.com, and emerson@ufscar.br).

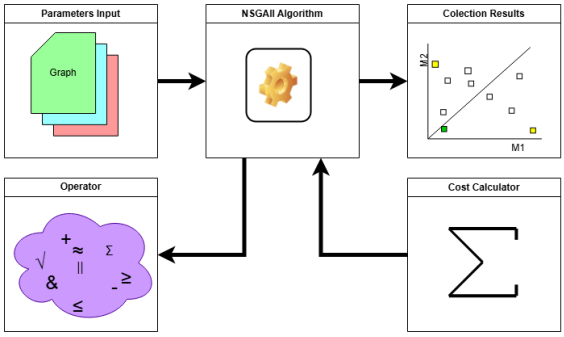


Fig. 1. Block Diagram of the Proposed Model.

In Fen Ge et al. [20], an efficient method called multi-phase is presented, which is used to solve the mapping of many applications in two phases. This work is one of the few in the literature to explore this idea, as most others only deal with mapping mono-applications. The multi-phase algorithm first allocates the best areas for mapping each of the given applications, and then maps these areas to optimized regions of the NoC (Network on Chip). However, the heat distribution throughout the chip, given by this solution, is not optimal, as it does not consider finding idle cores between the mapped tasks. Additionally, the optimization presented occurs in two steps using an evolutionary algorithm, making it less flexible if the user decides to work with other metrics in this scenario. Therefore, the motivation of this article is to propose an alternative and flexible solution, regarding the use of metrics in this context, to the work of Fen Ge et al. [20]. Through a multi-objective optimization algorithm, it is possible to simultaneously improve performance in executing multi-application tasks and achieve better energy distribution throughout the chip. This approach aims to solve the problem encountered by [20].

II. Solution Structure and Problem Presentation

As observed, the problem of mapping multiple applications onto a NoC to balance various objectives including fault tolerance, computational load, latency, communication load, and energy requirements—while optimizing more than one of these requirements simultaneously, remains largely unexplored. Therefore, this paper presents a proposal to map multiple applications with multiple objectives onto a 2D NoC.

In Fig. 1, an overview of the proposal is presented. The system is composed of a Parameter Entry module, which defines execution configurations and organizes the graphs to be used by the PlatEMO [21] optimizer, in which an NSGAII Algorithm [22], [23] module performs the optimization of the simultaneous metrics desired by the user who, in this work, these will be the latency and fault tolerance metrics. The Cost Calculation module calculates the costs of each individual in the population offered by the algorithm, the Operator module performs the task of diversifying the population and, finally, the Results Generation module presents the set of solutions obtained.

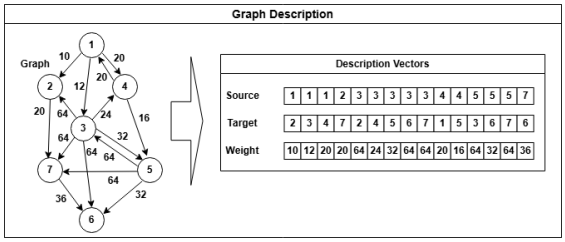


Fig. 2. Conversion of a graph into 3 vectors. Source- source node, Target- destination node, Weigth- Bandwidth occupation of the arc.

A. Parameter Input

1) Parameters: The system needs some parameters for its execution, so it is necessary to provide several application graphs in which you want to find the best arrangement for optimizing the intended metrics, as already mentioned above, and also, the user needs to specify the grid size of the Study NoC.

2) Graphs: Graphs are provided through a set of vectors that describe the origin and destination of each arc of a given graph, their respective communication weight, where the number of positions in each vector is equal to the number of arcs in each graph. Each application is represented by a graph and each graph is represented by a set of three vectors. Fig. 2 illustrates how the graph is represented. The Source vector contains a sequence of tasks and the value of each position represents the origin task of the arc. In the Target vector, each position represents the target task and the Weight vector is the weight of the arc. Thus, in the three vectors, each position represents an arc, for example, in position 1 we have the starting point of an arc in the Source vector, the end point of this same arc in the Target vector, and the communication weight of this arc in Weight.

3) NoC Mapping: The way to represent an application in the NoC structure can be seen in Fig. 3 , and occurs through a pair Task, PE, where Task is the identification of the task, and PE is the processor.

4) Compound Graph: The graphs are joined into a single graph that is used in optimization, as seen in Fig. 4 and algorithm1, the vectors are joined into a sequence, and the tasks receive a new numbering, which allows the algorithm to treat all applications as a single chromosome (individual).

Algorithm 1 Create Graph Vectors.

$S \leftarrow (S_1, S_2, \ldots, S_n)$
$T \leftarrow (T_1, T_2, \ldots, T_n)$
$W \leftarrow (W_1, W_2, \ldots, W_n)$
return Create vectors with all applications.

B. NSGAII Agorithm

The NSGAII Algorithm [22], [23] is used as the optimization algorithm for the proposed metrics. In this work, the implementation of this algorithm provided by PlatEMO was used as a base. To use this tool, the proposed chromosome was adapted to it, allowing the description of new customized

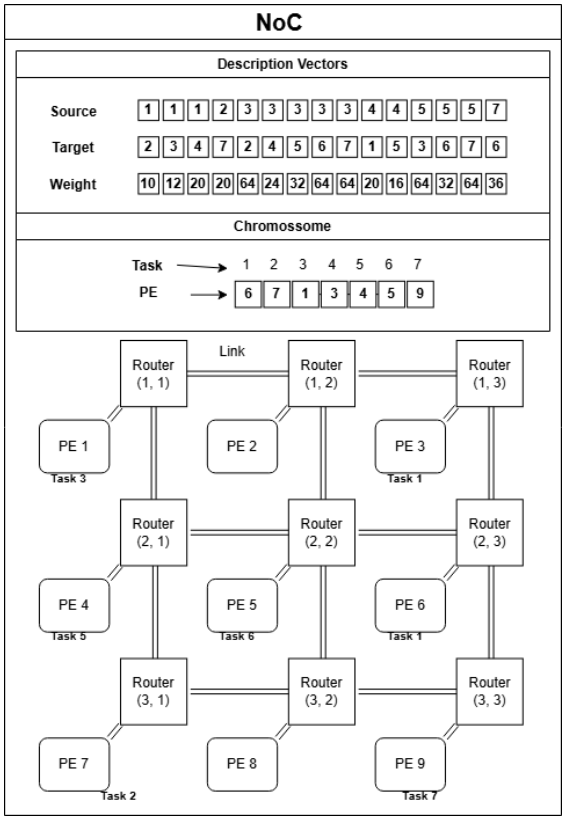


Fig. 3. Conversion of the chromosome into a NoC mapping.

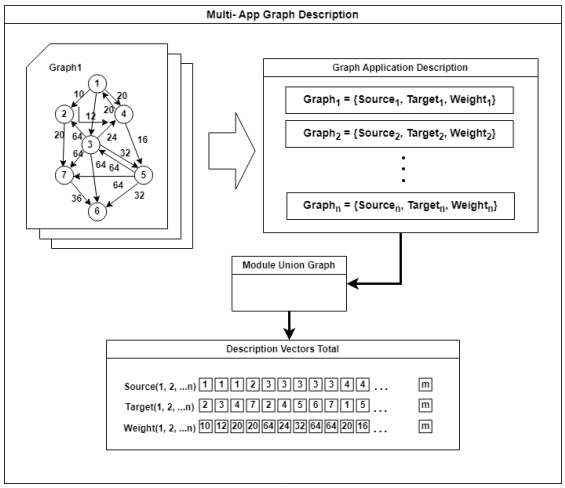


Fig. 4. Conversion of application graphs into a single vector.

solutions regarding the generation of the initial population, genetic operators, and involved cost calculations. More details on this can be found in the PlatEMO Manual. The choice of NSGAII was based on its availability in the tool, its nature as a multi-objective algorithm, ease of configuration and its good performance in initial tests, when compared with NSGAIII, MOPSO and MOEA/D.

C. Cost Calculator

This module receives the algorithm's population and calculates the cost for each expected objective. For the communication cost, the cost between tasks of the same arc is summed, and the communication cost between two tasks is calculated by the cost given by the graph multiplied by the number of routers needed to go from the source task ($source_i$) to the destination task ($target_i$) see(1). The fault tolerance and performance metrics studied are presented below, where r is the number of rows of the NoC and c is the number of columns and $nearestIdle$ is the closest node see(2).

$$Perf = \sum_{i=0}^{c-1} distance[source_i, target_i] - 1 \quad (1)$$

$$Ftol = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} distance[(r, c), nearestIdle] - 1 \quad (2)$$

D. Genetic Operators

The genetic operators used in PlatEMO receive the population, given by the chromosome description mentioned above, and perform the diversity operation, altering the population and searching for new solutions. They first execute a crossover between individuals and then a permutation.

In this phase, the results found, containing the best solutions, are saved in a database on the computer.

III. Experimental Analysis

To verify and validate the new multi-application mapping approach (Multi-Objective), a test similar to the one proposed in [20] is used as a reference (Multi-Phase). Benchmark graphs are used with 12, 16, 20 and 25 tasks, being App 12, App 16, App 20 and App 25 respectively (MPEG4, VOPD, WIFIRX and VCE), benchmark with graphs known in the literature executed simultaneously on a corresponding number of cores in a 9x9 NoC platform, where this process is repeated 100 times.

The proposal is evaluated considering the set of all applications for latency and fault tolerance metrics. In Table I, the methods (Multi-Objective and Multi-Phase) are compared for latency and fault tolerance metrics. Fig. 5 demonstrated, empirically, that the Multi-Objective proposal has a lower total latency than the Multi-Phase proposal. In Fig. 6, the total fault tolerance is much lower than in the Multi-Phase method. The proposed method achieves better results by fully utilizing the available area without being constrained by the geometry of the grid formed by task allocation. It also optimizes two metrics—latency and fault tolerance—while the Multi-Phase method is limited to a rectangular geometry and focuses on a single metric, latency.

In Figs. 8 (best latency - current proposal), 9 (best fault tolerance - current proposal), 10 (balanced latency and fault tolerance - current proposal), it is possible to visualize the results of the application maps generated in each case, for each approach, according to the Pareto curve given in Figs. 7 - A, B and C respectively, generated by the current proposal. The reader should note that the graph is not to scale. In Fig. 11, (compared map - Multi-Phase) the map for the Multi-Phase method is presented for optimizing the latency metric. All the maps are in a 9X9 NoC.

TABLE I
COMPARISON BETWEEN MULTI-OBJECTIVE AND MULTI-PHASE, LATENCY AND FAULT TOLERANCE

Method	Objective	Min	Q1	Q3	Max
Multi-Objective	Latência	8400	**14000**	20290	37800
Multi-Objective	FT	2611	**2724**	2866	3002
Multi-Phase	Latência	11200	25200	39200	75600
Multi-Phase	FT	3604	4260	4260	4260

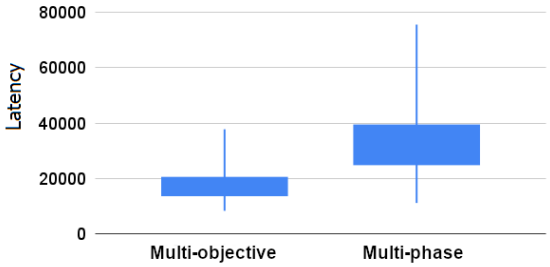


Fig. 5. Comparison chart of Latency obtained in both methods.

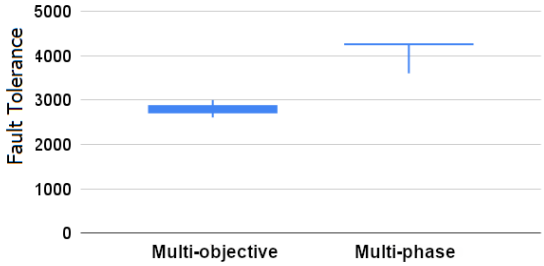


Fig. 6. Comparison chart of Fault Tolerance obtained in both methods.

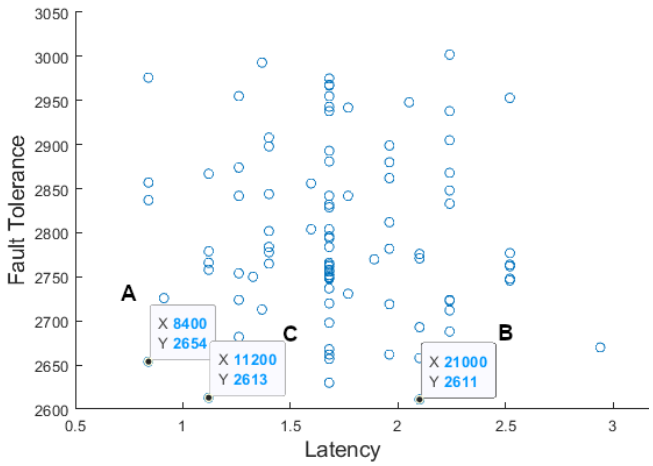


Fig. 7. Multi Objective Results. A-Improved Latency, B-Improved Fault Tolerance and C-Improved General (Latency and Fault Tolerance).

IV. CONCLUSION

In this proposal, a multi-objective optimization solution for generating multi-application maps on manycore architectures was presented and compared with the multi-application mapping proposal multiphase presented by Fen Ge et al. [20]. The Multi-Objective proposal demonstrates efficiency

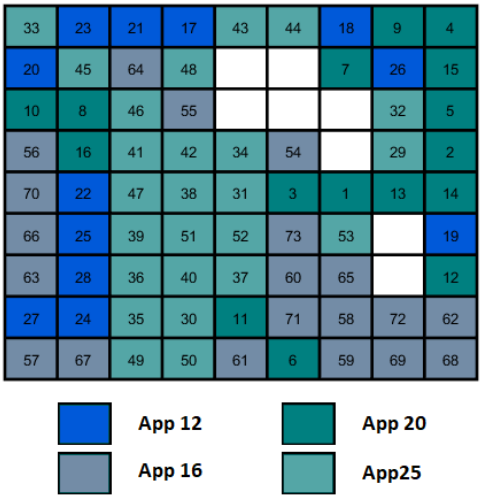


Fig. 8. A- Mapping Representation in NoC for Multi-Objective Improved Latency.

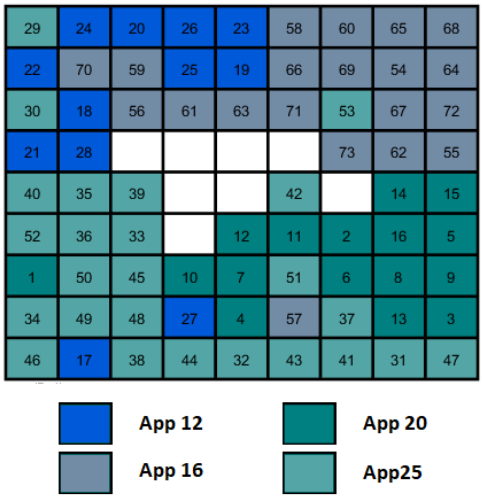


Fig. 9. B- Mapping Representation in NoC for Multi Objective Improved Fault Tolerance.

in achieving balanced results between the chosen metrics, performing a more effective mapping in the use of the NoC, better exploiting the available area, with improvements of over 30% in latency and 38% in fault tolerance. Additionally, it executes more quickly by optimizing all applications at once, thus better exploring the distribution of tasks in the NoC, without restricting areas or blocking processors. Furthermore, individual applications may have better latency values with the [20] proposal, but this does not guarantee better latency in the Noc when all are used. In this work, only one optimization algorithm, NSGAII, is used to evaluate the performance of a multi-objective algorithm compared to a single-objective algorithm in mapping tasks in a NoC. Therefore, an evaluation of other algorithms is necessary. In future works, Multi-objective optimization algorithms will be compared for mapping tasks in NoC across multiple applications.

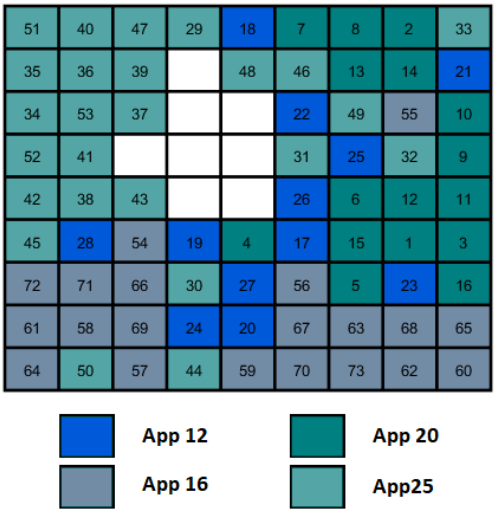


Fig. 10. C- Mapping Representation in NoC for Multi Objective Improved General (Latency and Fault Tolerance).

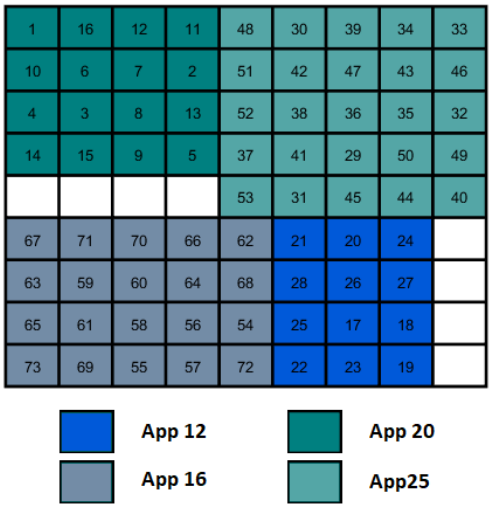


Fig. 11. Representation of Mapping in NoC for the Multi-Phase Solution.

REFERENCES

[1] W. Wolf, "Multiprocessor Systems-on-Chips," IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06), Karlsruhe, Germany, 2006, pp. 4-4, doi: 10.1109/ISVLSI.2006.65.

[2] S. Borkar, "Thousand core chips: a technology perspective, 2007, Association for Computing Machinery, New York, NY, USA, doi: 10.1145/1278480.1278667.

[3] Z. Sustran and J. Protic, "Migration in Hardware Transactional Memory on Asymmetric Multiprocessor," in IEEE Access, vol. 9, pp. 69346-69364, 2021, doi: 10.1109/ACCESS.2021.3077539.

[4] A. Oussous, F-Z. Benjelloun, A. A. Lahcen, S. Belfkih, (2017). Big Data Technologies: A Survey. Journal of King Saud University - Computer and Information Sciences. doi: 10.1016/j.jksuci.2017.06.001.

[5] M. Gheisari, G. Wang and M. Z. A. Bhuiyan, "A Survey on Deep Learning in Big Data," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 2017, pp. 173-180, doi: 10.1109/CSE-EUC.2017.215.

[6] M. Mohammadi, A. Al-Fuqaha, S. Sorour and M. Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923-2960, Fourthquarter 2018, doi: 10.1109/COMST.2018.2844341.

[7] J. Amin, M. Sharif, M. Yasmin, S. L. Fernandes, (2018). Big data analysis for brain tumor detection: Deep convolutional neural networks. Future Generation Computer Systems, 87, 290-297, doi: 10.1016/j.future.2018.04.065.

[8] M. A. Faruque, R. Krist and J. Henkel, "ADAM: Run-time agent-based distributed application mapping for on-chip communication," 2008 45th ACM/IEEE Design Automation Conference, Anaheim, CA, USA, 2008, pp. 760-765, doi: 10.1145/1391469.1391664.

[9] S. Kobbe, L. Bauer, D. Lohmann, W. Schroder-Preikschat and J. Henkel, "DistRM: Distributed resource management for on-chip many-core systems," in 2011 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Taipei, 2011 pp. 119-128. doi: 10.1145/2039370.2039392.

[10] L. Benini and G. De Micheli, "Networks on chips: A new soc paradigm," computer, vol. 35, no. 1, pp. 70–78, 2002, doi: 10.1109/2.976921.

[11] J. Ceng et al., "Maps: an integrated framework for mpsoc application parallelization," in Proceedings of the 45th annual Design Automation Conference, pp. 754–759, ACM, 2008, doi: 10.1145/1391469.1391663.

[12] T. Xu and V. Leppanen. (2016). "An efficient dynamic energy-aware application mapping algorithm for multicore processors." 119-124, doi: 10.1109/ICDIPC.2016.7470803.

[13] A. Singh, M. Shafique, A. Kumar and Henkel, Jörg. (2013). Mapping on multi/many-core systems: Survey of current and emerging trends. Proceedings of the 50th Annual Design Automation Conference, doi: 10.1145/2463209.2488734.

[14] P. Tendulkar, Mapping and Scheduling on Multi-core Processors using SMT Solvers. Theses, Universite de Grenoble I - Joseph Fourier, Oct. 2014.

[15] S. Mittal, "A survey of techniques for architecting and managing asymmetric multicore processors," ACM Computing Surveys, vol. 48, feb 2016, doi: 10.1145/2856125.

[16] S. Hong, S. H. K. Narayanan, M. Kandemir, and Ö. Özturk, "Process variation aware thread mapping for chip multiprocessors," in Proceedings of the Conference on Design, Automation and Test in Europe, pp. 821–826, European Design and Automation Association, 2009, doi: 10.1109/DATE.2009.5090776.

[17] T. Theocharides, M. K. Michael, M. Polycarpou, and A. Dingankar, "Towards embedded runtime system level optimization for mpsocs: on-chip task allocation," in Proceedings of the 19th ACM Great Lakes symposium on VLSI, pp. 121–124, ACM, 2009, doi: 10.1145/1531542.1531573.

[18] H. Orsila, T. Kangas, E. Salminen, T. D. Hämäläinen, and M. Hännikäinen, "Automated memory aware application distribution for multi-processor system-on-chips," Journal of Systems Architecture, vol. 53, no. 11, pp. 795–815, 2007, doi: 10.1016/j.sysarc.2007.01.013.

[19] L. Thiele, I. Bacivarov, W. Haid, and K. Huang, "Mapping applications to tiled multiprocessor embedded systems," in Seventh International Conference on Application of Concurrency to System Design (ACSD 2007), pp. 29–40, IEEE, 2007, doi: 10.1109/ACSD.2007.53.

[20] F. Ge, C. Cui, F. Zhou, and N. Wu. 2021. "A Multi-Phase Based Multi-Application Mapping Approach for Many-Core Networks-on-Chip" Micro-machines 12, no. 6: 613., doi: 10.3390/mi12060613.

[21] Y. Tian, R. Cheng, X. Zhang and Y. Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]," in IEEE Computational Intelligence Magazine, vol. 12, no. 4, pp. 73-87, Nov. 2017, doi: 10.1109/MCI.2017.2742868.

[22] F. Zhang, "Constructing a multi-objective optimization model for engineering projects based on nsga-ii algorithm under the background of green construction," Decision Making: Applications in Management and Engineering, vol. 7, pp. 37–53, 11 2023, doi: 10.31181/dmame712024895.

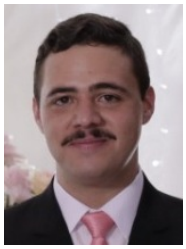
[23] S. Barakat, A. I. Osman, E. Tag-Eldin, A. A. Telba, H. M. Abdel Mageed, and M. Samy, "Achieving green mobility: Multi-objective optimization for sustainable electric vehicle charging," Energy Strategy Reviews, vol. 53, p. 101351, 2024, doi: 10.1016/j.esr.2024.101351.

[24] Qi, S., Li, Y., Pasricha, S., & Kim, R.G. (2023). MOELA: A Multi-Objective Evolutionary/Learning Design Space Exploration Framework for 3D Heterogeneous Manycore Platforms. 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), 1-6, doi: https://doi.org/10.48550/arXiv.2303.06169.

[25] Jitesh Choudhary, Chitrapu Sai Sudarsan, Soumya J., A performance-centric ML-based multi-application mapping technique for regular Network-on-Chip, Memories - Materials, Devices, Circuits and Systems, Volume 4, 2023, 100059,

ISSN 2773-0646, https://doi.org/10.1016/j.memori.2023.100059. (https://www.sciencedirect.com/science/article/pii/S2773064623000361)

Manoel Aranda de Almeida is a PhD student at the Department of Computer Science, Federal University of São Carlos, Brazil. He holds a degree in Information Systems from Faculdade João XXIII and a master's degree in Computer Science from UFSCar. His research focuses on image and video processing, machine learning, remote sensing, manycore architectures, and hardware development using FPGAs.



Igor Felipe Gallon is a PhD student at the Department of Computer Science, Federal University of Sao Carlos, Brazil, with B.S.E. in Computer Engineering from the Federal University of São Carlos in 2018. He is currently pursuing a Master's degree in Computer Science at the same institution. His research interests encompass Reconfigurable Architectures, Many-Core Architectures, and Big Data.

Emerson Carlos Pedrino is an Associate Professor in the Department of Computing at the Federal University of São Carlos, Brazil, with degrees in Electrical Engineering and Computational Physics from USP, where he graduated first in his class. He holds a Master's and Ph.D. in Electrical Engineering from USP, as well as a Postdoctoral fellowship at the University of York, UK, funded by FAPESP. His work focuses on real-time image and video processing using FPGAs, machine learning, remote sensing, and manycore architectures. He also serves as an editorial board member for scientific journals and as a project reviewer for FAPESP.