Extended Abstract: Monitoring-based Thermal Management for Mixed-Criticality Systems

Marcel Mettler*, Martin Rapp†, Heba Khdr†, Daniel Mueller-Gritschneder*, Jörg Henkel†, Ulf Schlichtmann*
* Chair of Electronic Design Automation, Technical University of Munich, Germany
{marcel.mettler, daniel.mueller, ulf.schlichtmann}@tum.de
† Chair for Embedded Systems, Karlsruhe Institute for Technology, Germany
{martin.rapp, heba.khdr, henkel}@kit.edu

***Abstract*—With a rapidly growing number of functions in embedded real-time systems, it becomes inevitable to integrate tasks of different safety integrity levels (SILs) into one mixed-criticality system. Here, it is important to not only isolate shared architectural resources, as tasks executing on different cores may also interfere via the processor's thermal manager. In order to prevent a scenario where best-effort tasks cause deadline violations for critical tasks, we propose a thermal management strategy that guarantees a sufficient thermal isolation between tasks of different SILs, and simultaneously reduces the run-time of best-effort tasks by up to 45% compared to the state of the art without incurring any real-time violations for critical tasks.**

***Index Terms*—Dynamic thermal management, mixed-criticality**

I. Introduction

New applications such as autonomous driving increase the complexity for modern embedded real-time systems. In order to still meet non-functional requirements such as cost, weight and power consumption, there is a trend in industry and academia to integrate tasks of different safety integrity levels (SILs) in one mixed-criticality system (MCS) [1]. As a result, safety standards [2] require that tasks must be isolated to prevent the propagation of faults between tasks of different SILs.

A prominent solution to provide the required isolation are virtualization technologies [3]. However, on many-core processors, an isolation of architectural resources, such as processing elements (PEs), caches, buses, etc. is not sufficient. Different tasks can not only interfere via shared resources but also via the many-core processor's thermal manager, which potentially causes deadline violations of critical tasks.

In order to prevent such a scenario, we present a monitoring-based thermal manager, called MonTM. MonTM aims to maximize the performance of best-effort tasks in MCSs under thermal constraints of the system and real-time constraints of critical tasks. It is based on the following components reflecting our key contributions: (1) A thermal management strategy for MCSs that prevents best-effort tasks from inducing thermal violations into critical tasks. For that purpose, MonTM uses a novel interconnect to communicate the thermal status of critical tasks. Hence, best-effort tasks can be throttled on imminent thermal violations of neighboring critical tasks. (2) A slack monitor that determines the minimal voltage/frequency (V/f) requirement of critical tasks based on their current progress. This enables MonTM to safely reduce the V/f level in scenarios with a large slack, increasing the available thermal headroom.

II. Monitoring-based Thermal Management

A. Problem Formulation

Given a many-core MCS, we differentiate between critical tasks and best-effort tasks. Critical tasks must be mapped to an exclusive resource PE_i to guarantee their schedulability at any time. As critical tasks are typically subject to timing requirements, we model their service level agreements (SLAs) by a tuple (C_i, D_i), where C_i corresponds to the worst-case execution time (WCET) using the maximal frequency $f_{PE_i,max}$ of its exclusive resource PE_i and D_i to the deadline. Best-effort tasks, such as the infotainment system, are typically not subject to timing requirements and, therefore, do not require a specific application model. They can be executed on any available PE that is not reserved for a critical task. Furthermore, we allow both critical and best-effort tasks to be scheduled on demand. As the underlying platform, we consider a network on chip (NoC)-based many-core processor with per-PE dynamic voltage frequency scaling (DVFS) on which all tasks are executed. Our objective is to maximize the performance of the best-effort tasks under the constraint that all critical tasks meet their deadline.

B. Thermal Pre-error Interconnect

The minimal frequency requirement of a critical task is difficult to compute since it depends on its execution behavior [4], i.e. whether it is memory- or compute-bound, and on its SLAs, i.e. its deadline and WCET. Hence, we consider an upper bound of the minimal frequency requirement, which only depends on its WCET C_i and its deadline D_i.

$$ub(f_{i,min}) = \frac{C_i}{D_i} f_{PE_i,max} \tag{1}$$

To be able to guarantee this upper bound for all critical tasks, it is crucial that all critical tasks may run (in absence of the best-effort tasks) in combination without thermal violations. While this can be enforced by design-time analyses using the thermal model of the chip, it furthermore needs to be ensured that no best-effort task interferes with any critical task via the thermal manager. Therefore, we propose a dedicated dynamic thermal manager (DTM) interconnect that communicates thermal pre-errors, i.e. imminent thermal violations, of a critical task to neighboring best-effort tasks such that these can be throttled in the favor of the critical tasks.

978-3-9819263-7-8/DATE23/© 2023 EDAA

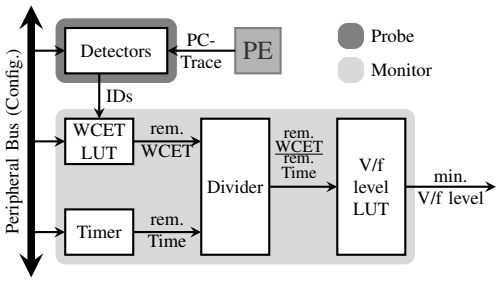

Fig. 1. Hardware architecture of the slack monitor

We define several levels of urgency, ranging from lowest urgency e_0 to the highest urgency e_3, which affect the number of throttled best-effort tasks. Given a thermal pre-error of e_i, with $i \in [0, 2]$, the best-effort tasks within a hop distance of i must be throttled. In a worst-case situation with a thermal pre-error of e_3, all best-effort tasks are halted. Please note that this is an emergency measure that moves the system to the case that only the critical tasks are executed, which are known to run in combination without thermal violations. As usually there is some thermal headroom available for best-effort tasks in MCSs, this mode should rarely be triggered during operation.

C. Slack Monitoring of Critical Tasks

In average and best-case scenarios of critical tasks, the minimal frequency requirement, presented in Eq. 1, could be further reduced to increase the thermal headroom that is available for best-effort tasks. Therefore, we propose to determine the minimal frequency requirement of critical tasks based on their progress.

To achieve this, we instrument all critical tasks at specific points of interest and measure the remaining WCET for each of the points. At run-time, the slack monitor, illustrated in Fig. 1, detects the points of interest based on their program counter (PC) address and loads the respective remaining WCET based on the ID of the point. Furthermore, the monitor comprises a countdown timer, which issues the remaining time of the deadline of the task is reached. Together, the remaining WCET and the time remaining until the deadline can be used to compute the frequency requirement using a divider. Finally, a V/f level lookup table (LUT) translates the frequency requirement into the minimal V/f level of the task.

III. Experimental Results

The following evaluations are conducted on the field programmable gate array (FPGA) prototype of an 80-core processor [5]. In order to still accurately model the application specific integrated circuit (ASIC) behavior, we use the ASIC temperature and DVFS emulator presented in [5]. The evaluations are based on synthetic workloads with various run-time characteristics in terms of the number of best-effort tasks N_b and critical tasks N_c, and the variance in the power consumption $var(P)$. The name of the use case is formed from the used configuration according to $< var(P) >\_< N_c >\_< N_b >$.

In Fig. 2, we compare MonTM with the state-of-the-art resource management techniques GDP [6] and PdRM [7]. As all techniques satisfy the thermal requirements of the chip and the deadline requirements of the critical tasks, Fig. 2 presents

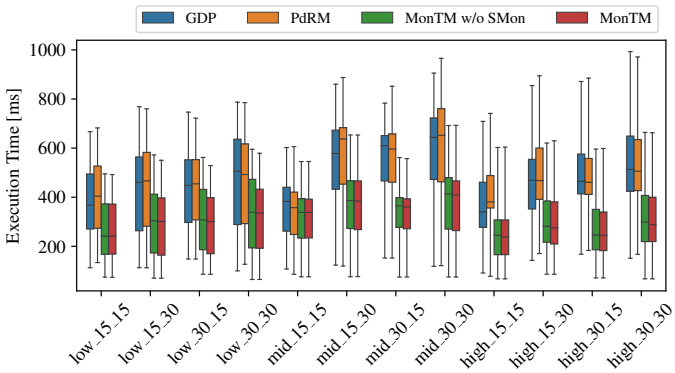

Fig. 2. Average execution time of the best-effort tasks for different use cases

the execution times of the best-effort tasks as a boxplot. It can be seen that MonTM with and without slack monitor outperforms the state-of-the-art methods in all use cases. While GDP and PdRM rely on the peak-power consumption of the tasks, MonTM can fully exploit the available thermal headroom and thereby reduce the average run-time by 7%-44% without slack monitoring. In addition, the slack monitor further reduces the average run-time of the best-effort tasks by another 1%-6%.

MonTM only introduces a neglectable run-time overhead of below 10 μs for the configuration of the hardware and a hardware overhead of 1,997 slice LUTs (1.3%) and 4,274 slice registers (4.4%) per core.

IV. Conclusion

In this paper, we presented MonTM, a monitoring-based thermal management strategy for MCSs. MonTM reduces the average run-time of best-effort tasks by up to 45% compared to the state of the art and simultaneously guarantees that all critical tasks meet their deadline.

Acknowledgment

This work was partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 146371743 – TRR 89 "Invasive Computing".

References

[1] H. Chai, G. Zhang, J. Sun, A. Vajdi, J. Hua, and J. Zhou, "A review of recent techniques in mixed-criticality systems," *Journal of Circuits, Systems and Computers*, vol. 28, no. 07, p. 1930007, 2019.

[2] IEC61508:2010, "Functional safety of electrical/electronic/ programmable electronic safety-related systems," British Standards Institution, London, UK, Standard, 2010.

[3] M. Cinque, D. Cotroneo, L. De Simone, and S. Rosiello, "Virtualizing mixed-criticality systems: A survey on industrial trends and issues," *Future Gener. Comput. Syst.*, vol. 129, no. C, p. 315–330, 04 2022.

[4] K. Choi, R. Soma, and M. Pedram, "Dynamic voltage and frequency scaling based on workload decomposition," in *International Symposium on Low Power Electronics and Design (ISLPED)*, 2004.

[5] M. Mettler, M. Rapp, H. Khdr, D. Mueller-Gritschneder, J. Henkel, and U. Schlichtmann, "An FPGA-based approach to evaluate thermal and resource management strategies of many-core processors," *ACM Trans. Archit. Code Optim.*, vol. 19, no. 3, 05 2022.

[6] H. Wang, D. Tang, M. Zhang, S. X.-D. Tan, C. Zhang, H. Tang, and Y. Yuan, "GDP: A greedy based dynamic power budgeting method for multi/many-core systems in dark silicon," *IEEE Transactions on Computers*, vol. 68, no. 4, pp. 526–541, 2019.

[7] H. Khdr, S. Pagani, E. Sousa, V. Lari, A. Pathania, F. Hannig, M. Shafique, J. Teich, and J. Henkel, "Power density-aware resource management for heterogeneous tiled multicores," *IEEE Transactions on Computers*, vol. 66, no. 3, pp. 488–501, 2017.