CAnDy-TM: Comparative Analysis of Dynamic Thermal Management in Many-Cores using Model Checking

Syed Ali Asadullah Bukhari*, Faqg Khalid Lodhi*, Osman Hasan*, Muhammad Shaﬁque† and Jörg Henkel‡

*School of Electrical and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan
Email: {ali.asadullah, faqg.khalid, osman.hasan}@seecs.edu.pk
† Institute of Computer Engineering, Vienna University of Technology, Vienna, Austria
Email: muhammad.shaﬁque@tuwien.ac.at
‡Chair for Embedded Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany
Email: henkel@kit.edu

Abstract—Dynamic thermal management (DTM) techniques based on task migration provide a promising solution to mitigate thermal emergencies and thereby ensuring safe operation and reliability of Many-Core systems. These techniques can be classiﬁed as central or distributed on the basis of a central DTM controller for the whole system or individual DTM controllers for each core or set of cores in the system, respectively. However, having a trustworthy comparison between central (c-) and distributed (d-) DTM techniques to ﬁnd out the most suitable one for a given system is quite challenging. This is primarily due to the systemic difference between cDTM and dDTM controllers, and the inherent non-exhaustiveness of simulation and emulation methods conventionally used for DTM analysis. In this paper, we present a novel methodology called CAnDy-TM (stands for Comparative Analysis of Dynamic Thermal Management) that employs Model Checking to perform formal comparative analysis for cDTM and dDTM techniques. We identify a set of generic functional and performance properties to provide a common ground for their comparison. We demonstrate the usability and beneﬁts of our methodology by comparing state-of-the-art cDTM and dDTM techniques, and illustrate which technique is good w.r.t. thermal stability and other task migration parameters. Such an analysis helps in selecting the most appropriate DTM for a given chip.

I. INTRODUCTION

Coping with the high chip temperatures in Many-Core architectures is one of the growing concerns in the nano-scale era [1]. This issue has a direct impact on the reliability of Many-Core systems as excessive heat generation may lead to malfunctioning and physical damage of the chip. Various DTM techniques [2] have been proposed to mitigate this problem. The low-level DTM methods, like Dynamic Frequency and Voltage Scaling (DVFS), overcome the heating problem by slowing down the processor speed and thus compromising on performance [3]. However, managing the thermal issues by dynamic reallocation of the cores’ utilization [4], [5], [6] has been found to be more performance efﬁcient.

DTM techniques can be further categorized as central or distributed depending on the nature of the DTM controller [7]. In central DTM (cDTM), the centralized controller [8], [6] is responsible for managing the power hungry tasks evenly across the chip, and therefore, it needs to be aware of the state of all the cores. On the other hand, in distributed DTM (dDTM) [4], [5], each core or a set of cores is controlled by a dedicated controller (or agent) and the complete system comprises of several agents communicating only with the neighboring agents. The DTM agents regulate the chip temperature by invoking a task migration algorithm based on certain temperature threshold for a system. This task migration decision can either be proactive [9], [6], or reactive [4], [8].

Conventionally, simulation and emulation techniques are employed to analyze these techniques individually [10], [11]. However, these methods can analyze only a limited number of all the possible scenarios owing to a large number of DTM design parameters and the limited computational resources and thus result in a compromised analysis in terms of accuracy and completeness. Such limitations in the analysis may in turn lead to delays in deployment of DTM schemes as happened in the case of the Foxton DTM that was designed for the Montecito chip [12]. Moreover, the variety of DTM design parameters and the limitations of conventional analysis methods complicate the task for comparing different DTM techniques and identifying the most effective one in terms of performance for a given scenario becomes very challenging.

In the past couple of decades, the use of formal methods has been proposed to overcome and complement the limitations of simulation and emulation methods for DTM techniques [13], [14], [8]. More speciﬁcally, model checking, which is a state-space-based analysis technique, has been employed to analyse some of the state-of-the-art DTM techniques, like Thermal-aware Agent-based Power Economy (TAPE) [8], [15], [16]. In [8], the analysis of TAPE has been done for both functional and timing properties for a 3 × 3 core grid. However, during the analysis, the continuous parameters, like temperature, were abstracted using a small set of integer values and the number of cores could not be increased because of the infamous state-space explosion problem [17] of model checking. A solution to this problem has been proposed in [18] by using continuous-valued parametric modeling support in the nuXmv model [19] checker for analyzing a DTM-based task migration algorithm [4] for a 9 × 9 core grid. Probabilistic model checking has also been used for formal analysis of TAPE in [15] and [16], for 3 × 3 and 9 × 9 grid sizes, respectively. The results showed that the TAPE achieved thermal stability but did not perform the temperature balancing across the chip. In a recent work [20], a Formal Analysis Methodology for Task Migrations, i.e., FAMe-TM, has
been proposed for dDTMs using nuXmv. FAME-TM identifies a set of properties based on task migration principles and provides a generic methodology for formal modeling and analysis of task migration based dDTMs. In the above-mentioned works, individual analyses for given cDTM or dDTM techniques are provided, but a common analysis framework applicable to both is missing. Moreover, a comparison between cDTM and dDTM, based on communication overhead, which can be a key factor in performance evaluations of DTM techniques, is also missing. In the current paper, we aim at overcoming these deficiencies and bridging these gaps to provide a robust comparative framework. **Our Novel Contributions:** The novel features of CANdy-TM include:

1) **Identification of a set of generic functional and performance properties** that can be formally verified for comparing cDTM and dDTM techniques.

2) A **step-wise procedure for developing formal models of both cDTM and dDTM techniques** and the ability to conduct an exhaustive comparative analysis, due to the completeness of model checking [17].

3) **Open-source code for CANdy-TM, and the nuXmv models for state-of-the-art DTM schemes are available at** [21]. This will help the DTM community to reproduce comparative results and design verification-friendly DTM techniques.

II. **Proposed Methodology - CANdy-TM**

The first key step in the comparative analysis of DTM schemes using the proposed methodology, i.e., CANdy-TM, as depicted in Fig. 1, is the modeling of a Many-Core grid system. Each core in the system is modeled as an individual module including the performance parameters that each core updates using counters for these parameters. These individual modules are then instantiated to compose an overall grid. The behavior of a cDTM or dDTM scheme is modeled by creating modules for the central and the distributed agents, respectively. In case of a cDTM scheme, one of the cores is declared a central agent (equivalent to a centralized controller/manager) that is responsible for the DTM decisions for the complete system. However, in case of a dDTM scheme, the behavior of a distributed agent is replicated on each core of the system. Next, we identify a set of functional properties, to be verified for the state-space models, and formally express them using Linear Temporal Logic (LTL) operators available in nuXmv. These LTL properties check the functional correctness of the given DTM for a Many-Core system by verifying its stability w.r.t. threshold avoidance and thermal balancing. For a Many-Core system, stability w.r.t. threshold avoidance is defined as the state when temperature of each core in the system is below a certain threshold. Similarly, stability w.r.t. thermal balancing is the state, where the temperature difference of each core in the system with the average temperature is less than an allowed threshold variation. We have also defined a thermally safe state where both the stability conditions hold for a given Many-Core system. These LTL properties are based on DTM principles, that are common to both cDTM and dDTM in order to provide a common ground for comparative analysis. The LTL properties and the state-space models are then passed to nuXmv for verification. Besides the functional properties, the performance parameters (included in each core module) are also calculated during the verification. The most important performance parameter for a DTM is the number of state transitions it takes to reach a thermally stable state. Other performance properties include parameters like, task load completed, the number of task migrations, the number of tasks stalled, the number of hot spots created and the communication overhead. All these performance parameters are calculated till the system achieves the stable condition.

![Fig. 1: An Overview of our CANdy-TM Methodology](image)

III. **Analysis of DTM Schemes**

In this section, we present the formal analysis of two state-of-the-art DTM schemes to show the effectiveness and rigor of our proposed methodology, i.e., CANdy-TM. To show the generality of our approach, we have selected the DTM schemes from both central and distributed domains:

1) **Multi-Objective DTM - mDTM** [6]. It is a centralized proactive DTM technique that avoids thermal threshold as well as balances the cores temperatures across the chip.

2) **Hot Spots Reduction DTM - hDTM** [4]. This technique performs distributed task migration with the primary goal of reducing the number of hot spots in the chip.

More details about these schemes can be found in their respective papers [6], [4].

A. Results and Discussions

The verification of the selected DTM schemes has been done for different grid sizes from $3 \times 3$ up to $12 \times 12$, using nuXmv 1.0.1, running on an Intel Core i7-6700T Quad-Core server operating at 3.06 GHz with 32 GB of RAM. For modeling the thermal behavior of a core, we have used the following Equation 1 from a real-world Intel Xeon processor [22]:

$$T_i = \Psi_{CA} \times W + T_{LA}$$  

where $T_i$ is the core’s temperature, $\Psi_{CA}$ is the thermal resistance, $W$ is the power consumption and $T_{LA}$ is the local ambient temperature of the core. The values of variables $\Psi_{CA}$ and $T_{LA}$ given in Equation 1 are taken as 0.655°C/W and 25°C, respectively [22]. For illustrative purposes and for accelerated evaluation, the threshold temperature to invoke task migration...
is taken as 41.5°C. The values of other parameters used in mDTM and hDTM are taken from their respective implementation details given in [6], [4]. Each task is assigned power units non-deterministically from 1 to 10. For calculating the communication overhead, we have modeled a shared memory system for the Many-Core systems. Therefore, in case of a migration, only a minimal data structure (e.g., pointers to the program and data memory) is transferred from one core to another. We have assumed a 64-bit pointer for our model. In addition to this communication during task migration, each core in a dDTM and cDTM is communicating with its neighbors and the central core, respectively. This communication includes 64 bit values for temperature, core ID and destination core ID for migration. One bit is used to indicate a threshold violation and another bit is used to indicate whether a suitable destination core has been found or not.

1) Functional Verification: The functional verification of mDTM and hDTM is done using the nuXmv’s bounded model checking (BMC) support for real numbers with a k-bound value of 15. The verification results show that mDTM ensures thermal safety of the Many-Core system by attaining stability w.r.t. both threshold avoidance and thermal balancing, owing to separate controllers for each purpose. However, hDTM achieves stability only in terms of threshold avoidance and is not stable for thermal balancing and safety. This temperature variance across the chip may cause reliability issues and even result in physically damaging the chip. Therefore, mDTM is a better choice than hDTM as the foremost objective of a DTM scheme is to achieve thermal stability in a Many-Core system.

2) Performance Evaluation Results: In addition to thermal stability, we suggest to compare the selected DTM techniques based on the performance properties given by CANDy-TM. These properties are evaluated till the DTM achieves thermal stability and provide further insights into working of the given DTM schemes. Fig. 2 shows the number of transitions required by each algorithm for different grid sizes to reach thermally stable state with threshold avoidance.

![Fig. 2: Number of state transitions to stability](image)

Table I shows the results of other performance properties for different grid sizes till the stability w.r.t. threshold avoidance. We provide the details of some key observations as follows:

**Observation 1:** The verification of the performance properties shows that the central DTM, i.e., mDTM, outperforms the distributed hDTM technique. For a 12 × 12 grid size, the number of transitions required by mDTM to achieve threshold stability is 58.4% less than hDTM. It is important to note that this comparison is made w.r.t. the threshold avoidance as hDTM does not meet the thermal safety criterion.

**Observation 2:** For 12 × 12 grid size, mDTM performs 40.9% lesser task migrations as compared to hDTM. The fewer number of task migrations means that mDTM achieves stability with lesser overhead in comparison with hDTM. Also, for the same grid size, the performance of mDTM in terms of completion of task load is 62% more than hDTM.

**Observation 3:** Table II shows the number of state transitions required by mDTM to achieve stability according to different criteria given by CANDy-TM. For each grid size, the number of transitions required to reach a thermally safe state is always 2.88 and 5.49 times the number of transitions required for achieving thermal balance and threshold stability, respectively.

**Observation 4:** Fig. 3 shows the average number of communication bits transferred by each core till the stability is achieved. This graph shows that at 3 × 3 grid size, mDTM requires 4.62 times less number of bits than hDTM. However, at 12 × 12 grid size, mDTM requires 3.85 times more number of bits than hDTM. This trend shows that for smaller grid sizes (up to 6 × 6), the central DTM, i.e., mDTM, performs better than the distributed DTM, i.e., hDTM, in terms of communication overhead.

**Observation 5:** Table III shows that the memory requirement for verification of stability property for mDTM and hDTM. Due to the complexity of mDTM, its verification requires more memory than that for hDTM. For example, for grid sizes 6 × 6 and up, the verification of mDTM uses full memory resources. However, the verification of hDTM utilizes full memory resources only for the grid size of 12 × 12. Moreover, the verification of mDTM always takes more time as compared hDTM due to its complexity.

**IV. Conclusion**

In this paper, we have presented a generic analysis methodology, i.e., CANDy-TM, for analyzing and comparing central
Embedded Systems” (SPP 1500 - spp1500.ietc.kit.edu) and in parts by the DAAD “Deutsch-Pakistansiche Forschungskooperationen” project.

REFERENCES


