

RePAiR: A Strategy for Reducing Peak Temperature while Maximising Accuracy of Approximate Real-Time Computing: Work-in-Progress

Shounak Chakraborty*, Sangeet Saha†, Magnus Sjalander*, Klaus McDonald-Maier†

*Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
{shounak.chakraborty, magnus.sjalander}@ntnu.no

†Embedded and Intelligent Systems Laboratory, University of Essex, Colchester, UK
{sangeet.saha, kdm}@essex.ac.uk

Abstract—Improving accuracy in approximate real-time computing without violating thermal-energy constraints of the underlying hardware is a challenging problem. The execution of approximate real-time tasks can individually be bifurcated into two components: (i) *execution of the mandatory part* of the task to obtain a result of acceptable quality, followed by (ii) *partial/complete execution of the optional part*, which refines the initially obtained result, to increase the accuracy without violating the temporal-deadline. This paper introduces *RePAiR*, a novel *task-allocation strategy* for approximate real-time applications, combined with *fine-grained DVFS* and *on-line task migration* of the cores and *power-gating* of the last level cache, to reduce chip-temperature while respecting both deadline and thermal constraints. Furthermore, gained thermal benefits can be traded against system-level accuracy by extending the execution-time of the optional part.

Index Terms—Approximate Computing, Thermal/Energy Efficiency, Real-Time Scheduling, CMPs (Chip Multi-Processors)

I. INTRODUCTION AND BACKGROUND

Contemporary real-time systems are not only constrained by their strict temporal *deadlines*, but are also forced to maintain a strict power/energy-limit of the underlying on-chip circuitry to prevent a catastrophic elevation in temperature. In such scenario, Approximate Computing (AC) approaches [1] can reduce the possibility of tasks missing their deadlines due to thermal constraints. In the AC approach, a task is decomposed into (i) a mandatory part, which can generate an acceptable output with minimal acceptable accuracy, followed by (ii) an optional part [2], which can fine-tune the accuracy of the outputs of the prior part. However, the optional part can be executed partially or fully for refining the obtained result based on the current energy-thermal-timing restrictions of the system. A recent theoretical analysis of AC-aware off-line real-time task allocation techniques having energy and deadline constraints is presented by L. Mo et al. [3], however, comprehensive case studies that include architectural parameters (e.g., cache misses, processor stalls, IPC, etc.) on-the-fly, are yet to be conducted.

This paper introduces *RePAiR*, a task-allocation strategy that intends to maintain the peak temperature of a chip within the prescribed thermal envelop while maximising accuracy

of the results for an AC real-time system. Primarily, a novel **thermal-aware task-allocation** will be developed through on-line task profiling with detailed periodic updates. At runtime, to reduce the peak temperature, we will further employ **fine-grained DVFS (FG-DVFS)** to scale down voltage-frequency (V/F) of the cores during long memory stalls and a restricted **task migration** will take place at the end of an execution phase, if two adjacent tasks have generated hotspots on their respective cores. We will also **selectively gate last-level-cache (LLC) portions**, to reduce power consumption along with generating thermal buffers on-chip. In the next section, our primary assumptions along with the adopted system model will be discussed.

II. SYSTEM MODEL & ASSUMPTIONS

We assume a homogeneous multi-core platform consisting of m cores, denoted as $C = \{C_1, C_2, \dots, C_m\}$, each core has the same DVFS capability. We consider a set of n independent AC real-time tasks $T = \{T_1, T_2, \dots, T_n\}$, and $n > m$. Execution time, l_i , for each task T_i ($1 \leq i \leq n$) is logically decomposed into M_i , running time for the mandatory part, and O_i , maximum execution time for the optional part. μ_i ($\in [0, 1]$) denotes the *proportion of the executed optional part* for T_i . At some fixed frequency *freq*, execution time l_i can be defined as:

$$l_i = M_i + \mu_i \times O_i \quad (1)$$

and we define result's accuracy of a task T_i as:

$$Acc_i = \mu_i \times O_i \quad (2)$$

Thus, the overall system level accuracy (Acc_{system}) can be defined as the sum of the executed CPU cycles of O_i for all the tasks in the system [2], which can be mathematically represented as:

$$Acc_{system} = \sum_{i=1}^N Acc_i \quad (3)$$

Primarily, by targeting an embedded system, we assume that the tasks are executed in a *frame-based* manner [4] and therefore all tasks have to finish their executions within a common

deadline/period, D . Hence, we can now define the temporal resource demand of a task T_i by the tuple $\langle S_i, l_i, D \rangle$, where S_i denotes the start-time of T_i . Our fixed set of persistent tasks (T), is assumed to be known offline and to arrive periodically for execution [5].

In order to enhance the result's accuracy of the individual tasks, more portions from the optional part need to be executed at higher processing speed which can be achieved by significantly boosting up the clock frequency of the core. However, stimulating clock frequency will potentially increase the power consumption (P) which might significantly increase the core's temperature. Practically, the temperature of j -th core (θ_j) is a function of its power consumption and can be represented as:

$$\forall j : 1 \leq j \leq m \mid \theta_j = \text{func}(P), \quad (4)$$

and, the system peak temperature (θ_p) can be written as:

$$\forall j : 1 \leq j \leq m \mid \theta_p = \max(\theta_j). \quad (5)$$

III. PROBLEM FORMULATION

Our proposed technique *RePAiR* intends to maximise the result's accuracy, while respecting system constraints. Towards that, *RePAiR* will employ a set of thermal management techniques to maintain the peak temperature of the chip below a predefined threshold. We have formulated *RePAiR* as follows-

Objective: Maximise Acc_{system} .

Subject to:

- 1) All tasks must meet the common deadline, D , i.e., $\max((S_i + l_i) \mid \forall T_i \in T) \leq D$.
- 2) System peak temperature (θ_p) is below the given thermal envelope (θ_{th}) i.e. $\theta_p \leq \theta_{th}$.
- 3) At a certain time instant, only one task can start executing exclusively on a certain core without any preemption.
- 4) At any time instant, number of tasks executing on the system should not be greater than the total number of cores.
- 5) $\mu_i \in [0, 1], \forall T_i \in T$.

After formulating the problem, we will now detail about the proposed methodology in the next section.

IV. REPAiR

RePAiR intends to maximise the system level accuracy (Acc_{system}) by executing more from the optional parts (O_i) of individual task while maintaining θ_p below the predefined thermal envelope (θ_{th}). Towards that, *RePAiR* will orchestrate a temperature-cognizant task-allocation (Section IV-A) along with a dynamic thermal management (DTM) (Section IV-B) by considering several run-time critical parameters both at task-level as well as architectural-level.

A. Temperature-cognizant Task Allocation

First, each task is individually analysed off-line to collect their respective number of ALU and memory instructions. Primarily our allocation will avoid to assign two compute-bound tasks (higher ratio of ALU/memory instructions) on

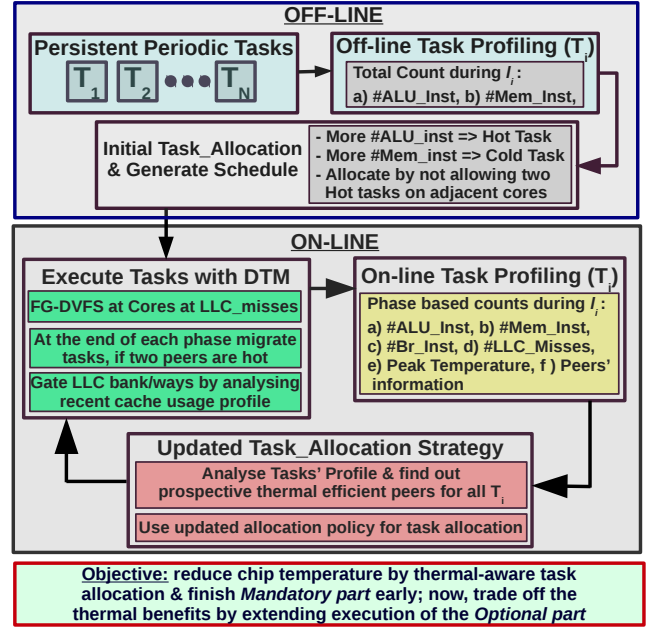


Fig. 1. *RePAiR*: Working mechanism.

adjacent cores by assuming them as comparatively hotter than the memory-bound tasks. But, this heuristic does not provide an optimal solution and no comprehensive analysis about their thermal-energy-performance issues is undertaken.

Furthermore, for any application, the total execution-time can be divided into multiple execution-phases. The variation in the characteristics within one application across such multiple execution-phases produces diverse thermal characteristics of the system over time. Hence, *RePAiR* will profile a detailed phase-based information for the individual task (T_i) that includes phase-wise $\#ALU_Inst$, $\#Mem_Inst$, $\#Br_Inst$ (Branch Instructions), $\#LLC_Misses$, and $Peak\ Temperature$, at the end of each phase of the periodic-execution. By analysing all of these on-line information, our task-allocation scheme will be updated and a new task-mapping will be created at the beginning of the next period. Note that, our technique will also profile the dynamic branch behaviour with due consideration to the inputs. Figure 1 depicts how *RePAiR* combines task-allocation along with its associated DTM.

B. Dynamic Thermal Management (DTM)

We will leverage the fast switching time of on-chip voltage regulators [6] to adapt FG-DVFS at the cores during stalls caused by LLC misses. This FG-DVFS will be applied at the task level, which implies that scaling of voltage and frequency for any core during an LLC-miss will depend on the respective task's behaviour. Our work will further tackle the situations incurred by MLP (memory level parallelism) in out-of-order (OoO) multi-cores, where individual LLC-miss never guarantees a stall at the requester core. Hence, to intelligently speculate the memory-stalls early at the OoO cores, a predictor will be developed.

On the other hand, large on-chip LLCs (built in 32nm or smaller technologies) of modern multi-cores often have

significant leakage power consumption. The existing diversity in accesses at different cache locations keeps large portions of the LLC underutilised, which if turned off can assist in reducing the chip temperature [7]. In addition with our FG-DVFS, we will implement power-gated LLC portions (ways or banks) as a supplementary thermal management technique, by considering past and present cache usages for our task-set. However, a restricted dynamic task-migration will also be employed at the end of any execution-phase, for stimulating thermal efficiency further if two adjacent cores are hot enough. Our system will always attempt to reduce chip temperature, while finishing execution of mandatory parts as fast as possible. This optimisation will enable us to trade off the achieved thermal benefits, while executing optional parts (with the highest possible value of μ_i), resulting in enhanced result accuracy.

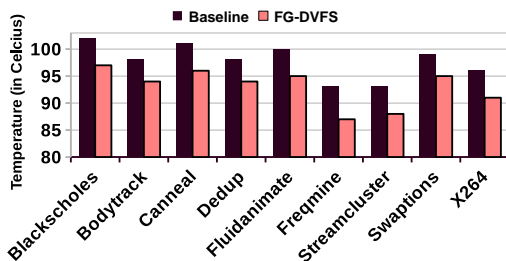


Fig. 2. Effects on peak temperature due to FG-DVFS.

V. INITIAL OBSERVATIONS

Primarily, we have implemented our idea of applying FG-DVFS at cores during LLC-misses. We simulated 9 multi-threaded PARSEC applications [8] using the gem5 simulator [9], and analysed power consumption (McPAT [10]) and temperature (HotSpot [11]) for a 16-core based homogeneous tiled multi-core (equipped with 16 Alpha 21364 cores [12]). All cores in this baseline architecture execute instructions in-order. Each of the cores are equipped with each 64KB, 4-way set-associative private L1 data and instruction caches. A physically-distributed yet logically-shared (among the cores) L2 is used as on-chip LLC. This shared-LLC is 16-way set-associative cache with a total size of 8MB and is physically-distributed into 16 uniform banks. We maintain a cache-block size of 64 bytes across the cache-levels.

Our simulation result (see Figure 2) shows noticeable reduction in peak temperature for all of the 9 PARSEC applications [8]. This empirical result further motivates us to perform detailed implementation of the proposed dynamic thermal management and task-allocation strategy. The combination of all of these proposed strategies will result in further reduction in peak as well as average chip temperature while maintaining the real-time constraint.

VI. CONCLUSION & FUTURE WORK

In this paper, we introduce *RePAiR*, a novel *task-allocation strategy* for approximate real-time applications, combined with *FG-DVFS* and *on-line task migration* of the cores and selective

power-gating of the LLC-portions, to reduce chip-temperature while respecting both deadline and thermal constraints. Furthermore, the gained thermal benefits will be traded against system-level accuracy by extending the execution-time of the optional part.

Towards completion, *RePAiR* will be shaped through detailed implementation of (1) temperature aware AC task-allocation (both with and without precedence constraints) and scheduling, where the *MEGA* tool [2] will be used to incorporate approximation on real-life applications, and (2) combined thermal management through FG-DVFS of the cores, power-gated LLC-portions, and a restricted task migration. Moreover, we intend to apply DVFS at the cores during generated slacks to reduce chip-temperature, but with some relaxation, so that result's accuracy can be enhanced by maximising execution-time of the optional part, the prime objective of *RePAiR*.

ACKNOWLEDGEMENT

This work is supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) through grants EP/R02572X/1 and EP/P017487/1, and ERCIM Post-doctoral fellowship awarded to Shounak Chakraborty.

REFERENCES

- [1] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, 2016.
- [2] K. Cao, G. Xu, J. Zhou, T. Wei, M. Chen, and S. Hu, "QoS-adaptive approximate real-time computation for mobility-aware IoT lifetime optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, 2019.
- [3] L. Mo, A. Kritikakou, and O. Sentieys, "Approximation-aware task deployment on asymmetric multicore processors," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019.
- [4] A. Roy, H. Aydin, and D. Zhu, "Energy-aware standby-sparing on heterogeneous multicore systems," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017.
- [5] S. Saha, A. Sarkar, A. Chakraborti, and R. Ghosh, "Co-scheduling persistent periodic and dynamic aperiodic real-time tasks on reconfigurable platforms," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 1, 2018.
- [6] E. A. Burton, G. Schrom, F. Paillet, J. Douglas, W. J. Lambert, K. Radhakrishnan, and M. J. Hill, "FIVR — fully integrated voltage regulators on 4th generation Intel-Core SoCs," in *2014 IEEE Applied Power Electronics Conference and Exposition - APEC 2014*, 2014.
- [7] S. Chakraborty and H. K. Kapoor, "Exploring the role of large centralised caches in thermal efficient chip design," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 24, no. 5, 2019.
- [8] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *2008 International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2008.
- [9] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, 2011.
- [10] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2009.
- [11] R. Zhang, M. R. Stan, and K. Skadron, "HotSpot 6.0: Validation, acceleration and extension." in *University of Virginia, Tech. Report CS-2015-04*, 2015.
- [12] P. Bannon, B. Lilly, D. Asher, M. Steinman, D. Webb, R. Tan, and T. Litt, "Alpha 21364: A single-chip shared memory multiprocessor, government microcircuits applications conference 2001," *Digest of Papers, Defense Technical Information Center, Belvoir, Va*, 2001.