ITMD: Run-time Management of Concurrent Multi-Threaded Applications on Heterogeneous Multi-cores

Basireddy Karunakar Reddy, Amit Kumar Singh, Geoff V. Merrett and Bashir M. Al-Hashimi
Department of Electronics and Computer Science, University of Southampton
{krb1g15, a.k.singh, bmah, gvm}@ecs.soton.ac.uk

Heterogeneous multi-core architectures are computing alternatives for several application domains such as embedded [1] and cloud [2]. These architectures integrate several types of processing cores within a single chip. For example, ARM's big.LITTLE architecture contains two types of cores; big and LITTLE, where big cores are grouped into one cluster and LITTLE cores into another [3]. Heterogeneous multi-cores often deal with multiple applications having different performance requirements concurrently, which generate varying and mixed workloads. Runtime management is required for adapting to such performance requirements and workload variabilities to achieve energy efficiency. In a heterogeneous multi-core system, it is challenging to efficiently exploit i) different types of cores simultaneously, and ii) DVFS potential of cores.

In our University Booth, we present and demonstrate a run-time management system - inter-cluster thread-to-core mapping and DVFS (ITMD), that first selects thread-to-core mapping based on the performance requirements and resource availability. Then, it applies online adaptation by adjusting the voltage-frequency levels to achieve energy optimization, without trading-off application performance. Thread-to-core mapping is facilitated by offline analysis of individual applications for performance and energy consumption when mapped to various possible resource combinations on a given heterogeneous multi-core platform. The *offline analysis* results help to choose the most energy efficient points for a set of applications to be executed at run-time while satisfying their performance requirements. For each application, the chosen point defines the thread-to-core mapping, and the platform is configured following the mapping to start the application execution.

During execution of applications, an *online adaptation* technique first classifies their inherent workload characteristics using the metric Memory Reads Per Instruction (MRPI), and then pro-actively selects an appropriate voltage-frequency pair based on the predicted workload to minimize the switching transitions and energy. Implementation and validation of both the offline and online steps are performed on a real hardware platform, specifically Odroid-XU3 platform. Experiments show up to 33% energy savings compared to existing approaches. Fig. 1 shows adaptation of ITMD to workload variations (primary vertical axis) by controlling frequency (secondary vertical axis) over time, compared to existing approaches: heterogeneous multiprocessing scheduler with ondemand (HMPO) and conservative (HMPC) governors.

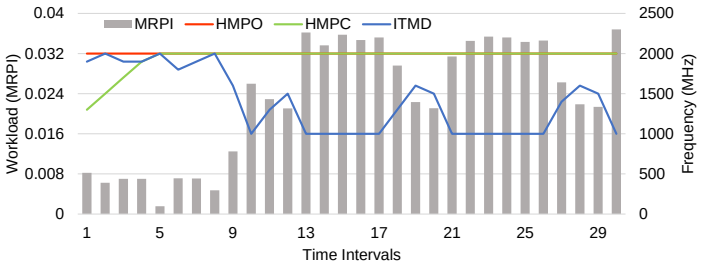

Fig. 1. MRPI and frequency at different time intervals of the application *freqmine* execution for various approaches. Resource allocation: *HMPC* - 4 big, *HMPO* - 4 big and *ITMD* - 4 LITTLE.

A high MRPI leads to scaling down the frequency, thereby *ITMD* approach minimizes the power consumption, whereas *HMPO* and *HMPC* run at max frequency. This is due to the fact that whilst the application is memory intensive, it places a high load on the processor cores as far as the load measured by the kernel is concerned. Therefore, these select the highest frequency even if it does not offer improvement in performance.

Our demonstration would include at least one real mobile development board executing multiple applications concurrently, as well as screen showing live graphs of workload and power consumption. Applications are chosen from PARSEC [4] and SPLASH [5] benchmark suites. Further, user can control the number of active applications to be executed concurrently, performance requirements and run-time management approach.

Acknowledgement

This work is supported in parts by the EPSRC Grant EP/L000563/1 and the PRiME Programme Grant EP/K034448/1 (www.prime-project.org).

References

[1] A. K. Singh *et al.*, "Mapping on multi/many-core systems: survey of current and emerging trends," in *Proceedings of the 50th Annual Design Automation Conference*. ACM, 2013, pp. 1–10.

[2] B. Khemka *et al.*, "Utility maximizing dynamic resource management in an oversubscribed energy-constrained heterogeneous computing system," *Sustainable Computing: Informatics and Systems*, vol. 5, pp. 14–30, 2015.

[3] P. Greenhalgh, "big.LITTLE processing with ARM cortex-a15 & cortex-a7," *ARM White paper*, pp. 1–8, 2011.

[4] C. Bienia and K. Li, "Parsec 2.0: A new benchmark suite for chip-multiprocessors," in *Proceedings of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, vol. 2011, 2009.

[5] S. C. Woo *et al.*, "The SPLASH-2 programs: Characterization and methodological considerations," in *ACM SIGARCH Computer Architecture News*, vol. 23, no. 2. ACM, 1995, pp. 24–36.