RETROSPECTIVE: Scheduling Heterogeneous Multi-Cores through Performance Impact Estimation (PIE)

Kenzo Van Craeynest[⊕], Aamer Jaleel[†], Lieven Eeckhout[‡], Paolo Narvaez[§], Joel Emer[†*]
[⊕]Government of Flanders [†]Nvidia [‡]Ghent University [§]Intel [*]MIT

This paper was published in ISCA 2012, at a time when single-ISA heterogeneous multi-core processors had started to gain popularity. Aamer Jaleel and Joel Emer were members of VSSAD, an advanced R&D group within Intel. Kenzo Van Craeynest was a summer intern in VSSAD visiting from Ghent University, where he was pursuing a PhD under the supervision of Lieven Eeckhout. Paolo Narvaez was an architect in Intel's Data Center Group (DCG). This work was the first to take a principled approach using CPI stacks on one core type to predict application performance on a different core type in a single-ISA heterogeneous multi-core processor.

I. Overview

Early multi-core processors of the PC era were homogeneous and focused on providing better performance by employing increasingly complex micro-architectural components, such as deep out-of-order buffers. This came at the cost of increased power consumption which was ill-suited for the new emerging, battery-sensitive devices that had to operate at a low thermal design point (TDP). To address this market, in 2008 Intel launched the Atom processors, which were initially used to power netbooks, tablets, and low-powered devices. The smaller Atom architecture used mostly the same ISA as the more complex Intel processors, but operated at a lower power range.

The benefits of using designs with different TDP targets is illustrated in Figure 1 using a power vs. performance graph. While 'small' and 'big' processors, or *cores*, can be tuned to operate at different power design points on their respective curves, the ranges are limited. Even at its lowest frequency, a big core cannot reduce its power beyond a minimum level due to other factors (e.g., leakage). Similarly, a small core can obtain higher performance by running at a higher frequency, but only up to a point. While there is some overlap in TDP range, the small core is better suited for low-TDP applications while the big core is better suited for high-TDP applications.

Consequently, processor core types were selected based on the workload characteristics that the system was expected to run. This was adequate for high-performance systems with sufficient power budgets (provisioned with a big core) or for power-constrained systems that did not require high-performance behavior (provisioned with a small core). However, this did not work well for systems with a wide dynamic range of application needs. An example of such a system is a smartphone, which is battery-constrained, and most of the time operates in an efficient low-power mode, but needs to exhibit bursts of high performance during user interaction to provide good user experience. In 2011, ARM announced the first commercial heterogeneous core architecture. Its big.LITTLE architecture combined a small A-7 core with a big A-15 core.

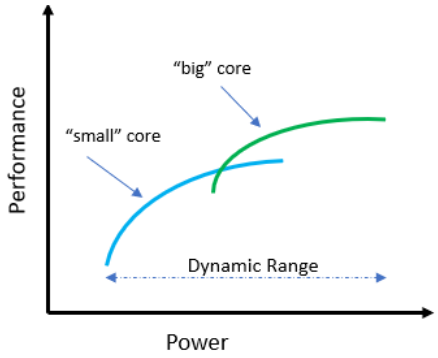

Fig. 1. Dynamic Range of Heterogeneous Multi-Core Processors.

While the initial motivation for combining ISA-compatible heterogeneous cores was to extend the dynamic range of computing systems, its emergence opened up a new field of possibilities on how to schedule and effectively combine applications running on both types of cores. While some of the research efforts at the time focused on how to schedule software threads based on power constraints or user experience requirements, our ISCA 2012 paper focused on maximizing throughput performance given a fixed set of big and small cores and software threads. The goal was to allow the scheduler to use the limited set of big cores to service the workloads that will benefit the most from the bigger cores, while scheduling other workloads to the small cores.

II. Suboptimal Heterogeneous Scheduling

The effectiveness of heterogeneous multi-cores hinges on how well a scheduler can map workloads onto the most appropriate core type. Making wrong scheduling decisions can lead to suboptimal performance and excess energy/power consumption. To address this scheduling problem, proposals at the time used workload memory intensity as an indicator to guide application scheduling [O2, O9, O17, O30].[1] Such

[1] [O*n*] citations refer to citations in the original paper.

proposals scheduled memory-intensive workloads on a small core and compute-intensive workloads on a big core which created suboptimal scheduling.

III. Primary Insight

In general, small (e.g., in-order) cores provide good performance for compute-intensive workloads whose subsequent instructions in the dynamic instruction stream are mostly independent, i.e., high levels of inherent instruction-level parallelism (ILP). On the other hand, big (e.g., out-of-order) cores provide good performance for workloads when the ILP must be extracted dynamically or the workload exhibits a large amount of memory-level parallelism (MLP). Therefore, we showed that scheduling decisions on heterogeneous multi-cores can be significantly improved by taking into account how well a small and big core can exploit the ILP and MLP execution characteristics of a workload.

IV. Performance Impact Estimation

Performance Impact Estimation (PIE) estimates the expected performance of a workload on each target core of a heterogeneous multi-core processor in terms of its Cycles Per Instruction (CPI) stack. CPI stacks are a valuable performance analysis technique that have origins in early work by one of the authors in a 1984 ISCA paper on the VAX-11/780.

PIE collects CPI stack, MLP and ILP profile information during runtime on any one core type, and estimates performance of the workload if it were to run on another core type. In essence, PIE estimates how a core type affects exploitable MLP and ILP, and uses the CPI stacks to estimate the impact on overall performance. In particular, PIE breaks up total CPI into two major components: the base component and the memory component, where the former lumps together all non-memory related components:

$$CPI = CPI_{base} + CPI_{mem}. \tag{1}$$

We found that MLP and ILP ratios provide good indicators on the performance difference between big and small cores. Therefore, we use MLP, ILP and CPI stack information to develop our PIE model (see Figure 2). Specifically, we estimated the performance on a small core while executing on a big core as follows:

$$\begin{aligned} CPI_{small} &= \widetilde{CPI}_{base\_small} + \widetilde{CPI}_{mem\_small} \\ &= \widetilde{CPI}_{base\_small} + CPI_{mem\_big} \times MLP_{ratio}. \end{aligned} \tag{2}$$

We estimate the performance on a big core while executing on a small core in a similar manner:

$$\begin{aligned} CPI_{big} &= \widetilde{CPI}_{base\_big} + \widetilde{CPI}_{mem\_big} \\ &= \widetilde{CPI}_{base\_big} + CPI_{mem\_small}/MLP_{ratio}. \end{aligned} \tag{3}$$

In the above formulas, $\widetilde{CPI}_{base\_big}$ refers to the base CPI component on the big core estimated from the execution on the small core; $\widetilde{CPI}_{base\_small}$ is defined similarly. The memory CPI component on the big (small) core is computed by dividing (multiplying) the memory CPI component measured on the small (big) core with the MLP ratio.

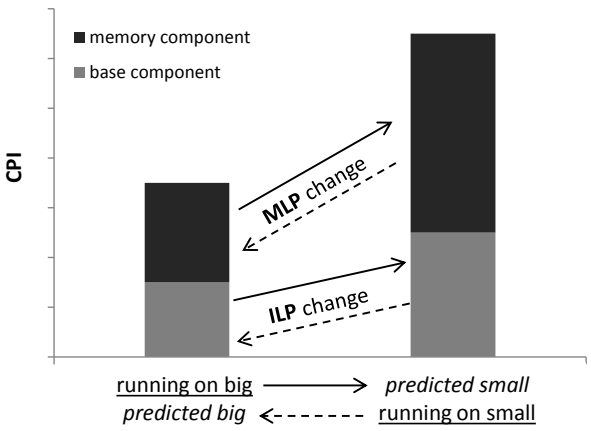

Fig. 2. Illustration of the PIE model.

V. Impact

PIE is an efficient hardware technique to assign workloads to the most suitable core in a heterogeneous multi-processor with the goal of optimizing overall system performance. Follow-on heterogeneous scheduling research developed more advanced techniques to optimize system performance while guaranteeing fairness (i.e., ensuring all threads in a multi-program and multi-threaded workload make equal progress) [3] and while also meeting the power budget [1], [2].

Since this work's publication, we have seen the release of important new commercial products featuring heterogeneous multi-cores. For example, in 2017, ARM announced its DynamIQ architecture, an extension of its original big.LITTLE design. It enables additional scheduling flexibility by allowing different types of cores to be placed in the same cluster and share cache while letting each core be individually frequency-managed. In 2021, Intel released its Alder Lake processor which features a combination of performance cores (P-cores) and efficient cores (E-cores). Like PIE, Alder Lake relies on a hardware-based technology called Intel Thread Director for scheduling. Intel Thread Director captures various performance metrics from the processor to guide the OS to choose the right type of core for each software thread. As more heterogeneous processors arrive in the market with increasing levels of flexibility, heterogeneity and complexity, it will become more critical for efficient scheduling algorithms to achieve the best balance of performance, power and user experience. PIE provides a principled framework for high-performance scheduling on heterogeneous processors.

References

[1] A. Adileh, S. Eyerman, A. Jaleel, and L. Eeckhout, "Maximizing heterogeneous processor performance under power constraints," *ACM Transactions on Architecture and Code Optimization*, vol. 13, no. 3, pp. 1–23, Sep. 2016.

[2] A. Adileh, S. Eyerman, A. Jaleel, and L. Eeckhout, "Mind the power holes: Sifting operating points in power-limited heterogeneous multi-cores," *IEEE Computer Architecture Letters*, vol. 16, no. 1, pp. 56–59, Jan. 2017.

[3] K. Van Craeynest, S. Akram, W. Heirman, A. Jaleel, and L. Eeckhout, "Fairness-aware scheduling on single-ISA heterogeneous multi-cores," in *Proceedings of the IEEE International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Sep. 2013, pp. 177–187.