Optimization of throughput, fairness and energy efficiency on asymmetric multicore systems via OS scheduling

Thesis Overview

Optimization of throughput, fairness and energy efficiency on asymmetric multicore systems via OS scheduling

Adrian Pousa apousa@lidi.info.unlp.edu.ar
Universidad Nacional de La Plata, Argentina

Most of chip multiprocessors (CMPs) are symmetric, i.e. they are composed of identical cores. These CMPs may consist of complex cores (e.g., Intel Haswell or IBM Power8) or simple and lower-power cores (e.g. ARM Cortex A9 or Intel Xeon Phi). Cores in the former approach have advanced microarchitectural features, such as out-of-order super-scalar pipelines, and they are suitable for running sequential applications which use them efficiently. Cores in the latter approach have a simple microarchitecture and are good for running applications with high thread-level parallelism (TLP).

CMPs are integrated in several platforms that run very diverse workloads, which present different demands and require the operative system to optimize goals such as global throughput, fairness or energy efficiency. However, CMPs present a limitation: a CMP model can turn out to be ideal regarding watt efficiency for a set of applications, but not for all of them.

Asymmetric Multicore Processors (AMPs) were suggested as an alternative to CMPs in order to overcome this limitation [1, 2]. An AMP is composed of two kinds of core: big or fast cores and small or slow cores. All cores use the same instruction set architecture (ISA). Using the same ISA enables running the same binary on all cores without the need to recompile the code for each core type.

Having cores of different types on an AMP enables us to employ specialization, (that is, to guarantee that an application is run on the type of core that delivers the best performance/energy trade-off). In general, abundant simple cores are good for running highly scalable parallel applications. On the other hand, complex and powerful cores are good for running single-threaded sequential applications. Another advantage of having both ?fast? and ?slow? cores in the system is that the fast ones can be used to accelerate sequential phases of parallel applications.

Despite these benefits, AMPs give rise to significant challenges [3, 4]. One of them is to distribute the fast and slow cores cycles among the different applications in an efficient way. This responsibility lies on the

operative system scheduler; therefore, it is not necessary to modify the application code.

At the beginning of this thesis, the goal of most of the previously proposed scheduling algorithms for AMPs was to optimize performance [2, 5]. To that end, the scheduler has to execute on the fast cores those applications which utilize them most efficiently. However, the optimization of other objectives, such as fairness or power efficiency, had not been investigated by researchers extensively. In this thesis we demonstrate that the schedulers that try to optimize only performance in fact degrade these other aspects.

The main goal of this thesis is the design of OS-level scheduling algorithms for AMPs, in order to optimize global throughput, fairness and power efficiency. To achieve this goal it was necessary to overcome three major challenges:

We developed a scheduling framework that facilitates the implementation and analysis of scheduling strategies on a realistic scenario: a real operative system on a real asymmetric multicore hardware. Due to the complexity of this task, many researchers assessed their scheduling strategies on simulators. In this thesis we implement such strategies on a real operative system; this allowed us to identify certain limitations of some proposals that do not appear on emulated environments.

We equipped the OS scheduler with a mechanism to estimate, at runtime, the relative benefit (speedup) that an application experiences running on a fast core relative to a slow core. By assigning applications to cores based on this speedup diversity is key to improve the global throughput, fairness or the energy efficiency on AMPs. For single-threaded applications, the speedup matches the speedup factor (SF) of its single runnable thread, defined as $\frac{\text{IPS}_{\text{fast}}}{\text{IPS}_{\text{slow}}}$ where IPS $_{\text{fast}}$ and IPS $_{\text{slow}}$ are the thread?s instructions per second ratios achieved on fast and slow cores, respectively. For parallel applications, the speedup is obtained from the SF of its threads, the number of threads and the number of fast cores. Measuring the SF directly is subjected to inaccuracies and generates overhead [6]. For this reason, we estimate the SF. For that, we constructed SF estimation models based on performance hardware counters.

We selected metrics to quantify the global throughput, fairness and energy efficiency on an AMP. Previously proposed metrics were defined for symmetric systems and they were not suitable for AMPs. Therefore, we constructed new metrics and adapted previously proposed metrics defined for CMPs.

The main contributions of this thesis are:

- We constructed an analytic model to find the theoretical schedulers that optimize global throughput, fairness and energy efficiency, respectively. From this model, we analyzed the interplay among these three objectives and demonstrated that it is not possible to optimize them simultaneously.

- We proposed the following scheduling algorithms for AMPs: Prop-SP, ACFS, EEFDriven y ACFS-E. Prop-SP is the first fairness-aware scheduling algorithm for AMPs that takes into account the diversity in relative speedups among applications. Although Prop-SP outperforms other scheduling strategies, it does not optimize fairness. To overcome this limitation, we proposed ACFS that maximizes fairness on AMPs and allows gradually improving throughput while keeping fairness under control. EEF-Driven maximizes the energy efficiency on AMPs based on the Energy-Efficiency Factor (EEF). The EEF of a thread is a new metric proposed in this thesis, which is defined as $\dfrac{SF}{EPI_{fast}}$, where EPIfast represents the energy per instruction rate when the application runs on a fast core. ACFS-E is a variant of ACFS and is the first asymmetry-aware scheduling scheme that can be configured to optimize fairness, energy efficiency or throughput, by employing a single yet flexible algorithm. ACFS-E is equipped with configurable parameters enabling the system administrator to gradually trade fairness for energy efficiency or throughput.
- We proposed a methodology to construct accurate estimation models of SF and EEF, based on performance hardware counters.
- Unlike other proposals, our strategies offer support to accelerate different types of parallel applications.

References

[2] Michela Becchi y Patrick Crowley. ?Dynamic Thread Assignment on Heterogeneous Multiprocessor Architectures?. Proc. of CF 06 (2006), pags. 29-40.

[3] N. Chitlur y col. ?QuickIA: Exploring heterogeneous architectures on real prototypes?. HPCA 12 (2012), pags. 1-8.

[4] Matt Gillespie. ?Preparing for The Second Stage of Multi-Core HW: Asymmetric (Heterogeneous) Cores?. Intel White Paper (2008).

[5] David Koufaty, Dheeraj Reddy y Scott Hahn. ?Bias Scheduling in Heterogeneous Multi-core Architectures?. Proc. of Eurosys 10 (2010).

[6] Daniel Shelepov y col. ?HASS a Scheduler for Heterogeneous Multicore Systems?. ACM SIGOPS OSR 43.2 (2009).

Additional information