Chapter 6
Conclusions

Efficiently exploiting thread-level parallelism has been challenging for software developers. As many parallel applications do not scale with the number of cores, not always using the maximum number of available cores running at the highest possible operating frequency will deliver the best performance or energy consumption. However, as discussed in this book, the task of rightly choosing the ideal amount of threads and the CPU operating frequency is not straightforward: many variables are involved (e.g., off-chip bus saturation and overhead of data-synchronization), which will change according to different aspects of the system at hand (e.g., input set, microarchitecture) and even during execution.

In this context, many works have been proposed to optimize the execution of parallel applications through the adaptation of the number of threads and the CPU operating frequency. As discussed in this book, they can be characterized according to the adaptability and transparency. The works that do not present adaptability at runtime can test many configurations without incurring in any overhead during the application execution, but their training time can take several hours. Besides that, when the environment changes, the offline analysis must be re-executed. On the other hand, approaches that provide adaptability at runtime can deal with the environment changes and select the ideal configuration as the application executes. However, this adaptability adds an overhead to the application execution. Therefore, runtime approaches must be efficient (find a solution close enough to the best possible one) and fast (converge to the solution within a few steps).

Most of the proposed approaches found in the literature and discussed in this book were designed to: (1) work only on homogeneous systems (same ISA and microarchitecture). Therefore, considering that heterogeneous multicore systems (HMS) are becoming more and more popular, we believe that one future direction is to design approaches to optimize the execution of applications in those systems.

A. Francisco Lorenzon, A. C. S. Beck Filho, *Parallel Computing Hits the Power Wall*, SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-030-28719-1_6

(2) target performance or energy. Nowadays, different non-functional requirements are becoming more important, such as the processor aging and reliability. (3) consider only one parallel API. However, heterogeneous systems may demand different communication models and, therefore, will probably need a mix of different APIs to communicate with each other.