Energy-efficient Building Blocks for Rack Scale Computing

Work in Progress

Rami Alkubaty
alkubaty.rami@fh-swf.de

Jan Richling
richling.jan@fh-swf.de

Fachbereich Elektrotechnik und Informationstechnik
Fachhochschule Südwestfalen

Keywords: Energy-efficiency, rack scale systems, heterogeneous multiprocessor systems, machine learning.

In many application areas computational requirements are not constant with respect to utilization of CPUs, memory or IO-bandwidth. The same is true with respect to changes in target functions: For some load scenarios performance is the top goal, for others energy-efficiency or a trade-off of these. In a rack scale system, these problems can be addressed at two different layers: On the rack layer, a rack manager has to decide on the distribution of load to the individual systems of the rack. This, however, can only be done on a more coarse grained level in order to keep the overhead at a reasonable level. Therefore, fine grained decisions has to be done at system level (by a system manager) while implementing a feedback channel to the rack manager.

In order to deal with fluctuating load with heavily changing requirements, we believe that heterogeneity is the way to go both at rack level as well as at system level. This way, the rack manager can select an appropriate system based on load characteristic and target function while the system manager is able to fulfill the requirements by utilizing the local heterogeneous system resources.

The rapid development, diversity and ubiquity of today's microprocessor systems add another challenge. Research in the field of energy-efficiency microprocessor systems runs into the problem that the hardware foundation might become outdated very fast. This is especially problematic if the proposed concept is very closely coupled with some special features of some hardware. Therefore, our approach is to differentiate into a strategy that is mainly hardware-independent and an automatic adaptation to the actual hardware.

In this work, we present a concept of a building block for such systems based on ARM's big.LITTLE approach. While originally targeted mobile systems, we believe that the combination of performance- and energy-optimized cores featuring the same ISA is also a good match for energy-efficient rack scale systems in highly flexible environments. Currently, we are focusing on the system layer only, thus assuming that some rack manager assigns load to our system together with a target specification on a scale between 0 (only performance matters) to 1 (only energy consumption matters).

Dynamic assignment of tasks to a particular type of cores or a combination of them while obeying target functions regarding energy and/or performance is challenging. Schedulers, in general, deal with this issue by using heuristics or permutation of thread schedules [1]. In contrast, our proposed system manager based on Unsupervised Machine Learning (USML) for tasks assignment. USML is characterized by its simplicity (no a priori knowledge needed) and dynamics (ability to track phase changes). This makes USML advantageous over other machine learning approaches like Supervised Machine Learning (SML) (e.g., used in [2]) that requires a priori knowledge and can not easily capture phase changes.

The proposed tasks-assignment is done in two phases: In an off-line phase the system manager will be fed with a number of example applications and corresponding target functions. These applications are executed with different strategies (core assignment, core frequencies). Based on measurements (performance and energy counters), clusters are defined that share a strategy and a target function. In the second phase during run-time, the system manager uses the knowledge gained in the first phase to dynamically assign tasks based on their target function and observations of the run-time behavior (performance counters) to clusters and executes them in an appropriate manner.

With respect to the rack manager, the concept described so far can be extended at a higher layer. Here, the rack manager also applies machine learning techniques by incorporating feedback originating from the different system managers. This way, it can dynamically adapt to different types of systems and different load scenarios thus optimizing the utilization of the rack scale.

1. REFERENCES

[1] Dawei Li, Jie Wu, and Li Dawei Wu Jie. *Energy-aware scheduling on multiprocessor platforms*. Springer, New York, 06 2014.

[2] Josep Ll. Berral, Íñigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà, and Jordi Torres. Towards energy-aware scheduling in data centers using machine learning. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, pages 215–224, New York, NY, USA, 2010. ACM.