Exploiting Application Dynamism for Auto-tuning HPC Applications

Extended Abstract

Madhura Kumaraswamy and Michael Gerndt
Technical University of Munich
Garching, Bavaria
kumarasw@in.tum.de

ABSTRACT

Today, optimizing the energy consumption for Exascale computing is the major challenge in HPC research. The EU Horizon 2020 project READEX aims to improve the energy-efficiency by leveraging the variations or dynamism in the application behavior to auto-tune HPC applications. It selects the best settings of the tuning parameters for different program regions, and also enables the user to specify domain knowledge to further expose the application dynamism, resulting in better dynamic savings.

CCS CONCEPTS

• **Hardware** → **Power estimation and optimization**; • **Computer systems organization** → **Multicore architectures**; • **Software and its engineering** → **Application specific development environments**;

KEYWORDS

Automatic tuning, HPC, DVFS, energy-efficiency

Optimizing energy consumption has become a challenging issue on the road to Exascale computing, especially when HPC systems have a power requirement of multiple MW. The EU Horizon 2020 project READEX (Runtime Exploitation of Application Dynamism for Energy-efficient Exascale computing) overcomes this challenge by providing an auto-tuning framework to dynamically tune HPC applications for energy-efficiency.

READEX quantifies the variations or dynamism in the application characteristics in the form of tuning potential, and uses a two step methodology to perform auto-tuning. The *Design-Time Analysis* (DTA) step is performed by the Periscope Tuning Framework (PTF) [4]. It returns the best tuning parameter settings for the invocations of instrumented program regions, called *runtime situations* (rts's) that show a tuning potential, and stores these in a *tuning model.* The *Runtime Application Tuning* stage reads the tuning model and dynamically switches the system configuration for an rts when it is encountered during production runs.

The tuning potential indicates if tuning the application results in potential savings, and is quantified using two dynamism metrics: execution time and compute intensity. The metrics are collected for the phases, which are individual iterations of the main progress loop, called *phase region*, and the rts's that are called from each phase. Intra-phase dynamism exists if there is a variation in the execution time or the compute intensity across the rts's, while inter-phase dynamism exists if there is a variation in the minimum and maximum execution time for the phase region.

To exploit intra-phase dynamism, DTA executes a Dynamic Voltage and Frequency Scaling (DVFS) tuning plugin called *readex_intraphase.* The plugin creates a search space of the tuning parameters (CPU frequency, uncore frequency, and OpenMP threads) using a user-specified search strategy. It then performs experiments, which request for measurements for the tuning objectives (energy, CPU energy, time, energy delay product or total cost of ownership) for the phase and the rts's. The best setting of the tuning parameters resulting in the lowest objective value for each rts and the phase is determined, and finally, the static and dynamic savings for the phase and the rts's are computed.

The automatic approach can be enhanced by specifying domain knowledge to expose the application dynamism, for example, in the Multi-Grid (MG) [2] benchmark. During the interpolation of the approximate solution from the coarser to the finer grid, the application switches from being compute bound to memory bound at a certain grid level. Special system configurations for compute and memory bound rts's can be determined using a region identifier for the grid level as domain knowledge. Without this, the rts's of the interpolate region may have the same best settings even if they have different behavior.

The poster will present the static and dynamic savings resulting from applying DTA for LULESH [1], miniMD [3], and MG (with domain knowledge), thus highlighting the effectiveness of this methodology.

ACKNOWLEDGMENTS

The research has received funding from the European Union's Horizon 2020 Programme under grant agreement number 671657.

REFERENCES

[1] [n. d.]. *Hydrodynamics Challenge Problem, Lawrence Livermore National Laboratory.* Technical Report LLNL-TR-490254. 1–17 pages.

[2] David H. Bailey. 2011. The NAS Parallel Benchmarks. In *Encyclopedia of Parallel Computing*, David Padua (Ed.). Springer, New York, 1254–1259.

[3] Paul Stewart Crozier, Heidi K Thornquist, Robert W Numrich, Alan B Williams, Harold Carter Edwards, Eric Richard Keiter, Mahesh Rajan, James M Willenbring, Douglas W Doerfler, and Michael Allen Heroux. 2009. *Improving performance via mini-applications.* Technical Report. Sandia National Laboratories.

[4] Michael Gerndt, Eduardo César, and Siegfried Benkner (Eds.). 2015. *Automatic Tuning of HPC Applications - The Periscope Tuning Framework.* Shaker Verlag, Aachen.

womENcourage'18, October 3-5, 2018, Belgrade, Serbia