The READEX Project for Dynamic Energy Efficiency Tuning Pre-Print Version of accepted paper.

Michael Gerndt
Technical University of Munich
Boltzmannstrasse 3
85748 Garching
gerndt@in.tum.de

ABSTRACT

High Performance Computing (HPC) systems consume a lot of energy. The overall energy consumption is one of the biggest challenges on the way towards exascale computers. Therefore, energy reduction techniques have to be applied on all levels from the basic chip technology up to the data center infrastructure. The READEX project explores the potential of dynamically switching application and system parameters, such as the clock frequency of the cores, to reduce the overall energy consumption of applications. An analysis is performed during application design time to precompute a tuning model that is then input to the runtime tuning library. This library switches the application and system configuration at runtime to adapt to varying application characteristics.

CCS Concepts

•Computer systems organization → Parallel architectures; •Hardware → Power and energy; •Software and its engineering → Development frameworks and environments;

Keywords

High Performance Computing; Autotuning; Energy Efficiency

1. INTRODUCTION

The goal of the Horizon 2020 project READEX (Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing) [1] is to explore dynamic techniques for increasing the energy efficiency of HPC applications. The overall energy consumption of future exascale systems is one of the biggest challenges on the way towards their realization.

Exascale systems will consist of millions of cores arranged in a hierarchical architecture. These systems will consist of shared memory accelerated compute nodes arranged into partitions that are interconnected by a hierarchical network architecture. The goal is to limit the power consumption of such systems to 20–30 MW. Therefore, energy reduction techniques have to be applied on all levels from the basic chip technology up to the data center infrastructure.

The READEX project focuses on application and system parameters that can be tuned dynamically to reduce the application's energy consumption [5]. It explores hardware parameters such as core and uncore frequency as well as, for example, hardware prefetcher settings in Intel Haswell processors [3]. It also looks out for energy saving opportunities by setting parameters of the system software, e.g., via control parameters of the MPI library and the number of threads used in OpenMP. Finally, application parameters are investigated selecting different code-paths, e.g., selecting alternative algorithms or dynamic offloading of computation onto accelerators.

READEX is based on the automatic tuning approach developed in the European AutoTune project and on the scenario-based tuning methodology from embedded systems [1]. The AutoTune project[2] developed the Periscope Tuning Framework (PTF) [2] that enables static autotuning of applications. It is a distributed framework consisting of a hierarchy of agents that monitor and control the potentially huge number of application processes. PTF provides analysis services that automatically detect performance bottlenecks in HPC applications, such as load imbalances or parallelization overhead. PTF also provides tuning plugins that focus on automatic tuning of individual tuning aspects of HPC applications. Several tuning plugins are available for selecting the best combination of compiler flags, the best combination of MPI library parameters as well as to tune the number of processes and threads used for program execution. PTF tuning plugins select the best static configuration which is then applied for the entire program execution.

2. THE READEX APPROACH

READEX will go beyond this static optimized configuration. Applications exhibit dynamic behavior, such as having different code regions with different characteristics or dynamically changing characteristics depending on the state of the computation. The READEX tool suite will dynamically reconfigure the application and its execution environment to adapt to such changes in the application behavior.

The approach focuses on the typical structure of HPC ap-

[1] www.readex.eu

[2] www.autotune-project.eu

SEM4HPC'16, May 31 2016, Kyoto, Japan

ISBN 978-1-4503-4351-0/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2916026.2916033

plication applications, that consist of an iterative approach. Simulation codes have a main progress loop, that steps, for example, through the simulated time in discrete time steps. The dynamism can either appear inside of an iteration, also called *program phase*, or over multiple phases. The first type of dynamism is called *intra-phase dynamism*, the second *inter-phase dynamism*.

READEX applies the scenario-based tuning approach from embedded systems. It first computes a *tuning model* at design time and applies this tuning model at runtime for dynamic switching of configurations. While overheads due to searching for best configurations are not significant at design time, at runtime, such overheads might easily cancel any gains in energy consumption. Design time analysis is carried out with PTF's tuning plugins. PTF will be extended to not only search for best configurations for different program regions, but also for different contexts of program regions, e.g., the callpath of a subroutine. Individual executions of program regions are called *runtime situations*. These runtime situations are classified into *scenarios*, such that all runtime situations in a scenario have the same best configuration.

The set of scenarios and their best configurations build the tuning model that is then forwarded from the design time analysis to the runtime tuning library, which is based on the Score-P monitoring system [4]. It reads the tuning model at application start and will, whenever a region is executed, look up the best configuration and may switch to that configuration, if the switching overhead will not cancel the expected gain. The runtime library might also dynamically calibrate the tuning model. This might be necessary, if runtime situations occur that were not seen at design time. For such runtime situations, a best configuration might be guessed from the information in the tuning model and calibrated during multiple occurrences.

3. EXPECTED RESULTS

The outcome of the automatic READEX methodology is expected to be at least 50% of the manually achievable gains. In contrast to the complicated and time consuming analysis required to find and switch the configurations manually, the READEX tool suite will increase energy efficiency automatically.

The READEX project is also trying to increase the tuning result by allowing the user to provide domain knowledge about the application via program annotations. This *domain knowledge specification* can provide additional application-level tuning parameters, specify high-level program regions, and provide identifiers for program characteristics. These identifiers will help at runtime to distinguish runtime situation with different characteristics and thus to provide a more detailed tuning model.

4. ACKNOWLEDGMENTS

The work presented in this paper is funded by the European Commission in the Horizon 2020 framework under Grant no. 671657 (READEX).

5. REFERENCES

[1] I. Filippopoulos, F. Catthoor, and P. G. Kjeldsberg. Exploration of energy efficient memory organisations for dynamic multimedia applications using system scenarios. *Design Automation for Embedded Systems*, 17(3-4):669–692, 2013.

[2] M. Gerndt, E. César, and S. Benkner, editors. *Automatic Tuning of HPC Applications - The Periscope Tuning Framework*. Shaker Verlag, 2015.

[3] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer. An energy efficiency feature survey of the Intel Haswell processor. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, pages 896–904, May 2015.

[4] A. Knüpfer, C. Rössel, D. an Mey, S. Biersdorff, K. Diethelm, D. Eschweiler, M. Geimer, M. Gerndt, D. Lorenz, A. Malony, et al. Score-p: A joint performance measurement run-time infrastructure for periscope, scalasca, tau, and vampir. In *Tools for High Performance Computing 2011*, pages 79–91. Springer, 2012.

[5] Y. Oleynik, M. Gerndt, J. Schuchart, P. G. Kjeldsberg, and W. E. Nagel. Run-time exploitation of application dynamism for energy-efficient exascale computing (readex). In *Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on*, pages 347–350, Oct 2015.