READEX Tool Suite for Energy-efficiency Tuning of HPC Applications*

Anamika Chowdhury, Madhura Kumaraswamy, Michael Gerndt
Technical University of Munich
Boltzmannstr. 3
Garching 85748, Germany
chowdhua@in.tum.de

ABSTRACT

The European Union Horizon 2020 READEX project is developing a tool suite for dynamic energy tuning of HPC applications. The tool suite performs an analysis during design-time before production run to construct a tuning model encapsulated with the best-found configurations that are then fed to the runtime tuning library. The library switches the configurations at runtime to adapt the application for energy-efficiency.

CCS CONCEPTS

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

KEYWORDS

Automatic tuning; High Performance Computing; tools for parallel computing; energy-efficiency

ACM Reference format:
Anamika Chowdhury, Madhura Kumaraswamy, Michael Gerndt. 2017. READEX Tool Suite for Energy-efficiency Tuning of HPC Applications[1]. In *Proceedings of SEM4HPC '17, Washington, DC, USA, June 26, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3085158.3091994

1 INTRODUCTION

In READEX[2] project, we develop READEX tool suite to improve energy efficiency by influencing tuning parameters during application execution at runtime [4]. The READEX methodology is a two-stage approach and consists of Design Time Analysis (DTA) and Runtime Application Tuning (RAT). Both uses the same monitoring system, Score-P [2] as the common instrumentation and measurement infrastructure. During design-time, a *tuning model* is precomputed by using the Periscope Tuning Framework (PTF)[1, 3]. A novel tuning plugin of PTF, the READEX Tuning Plugin was developed to determine the best *configurations* or settings fo the *runtime situations*(rts's), i.e., dynamic instances of *significant regions* [5]. PTF evaluates configurations during experiments with an instrumented version of the application. This tuning model is forwarded to the READEX Runtime Library (RRL) to tune the application by dynamically switching to the best configurations for upcoming rts's at runtime.

It is also possible to specify domain knowledge to distinguish more application characteristics in order to detect more accurate configuration. The means are called *identifiers* to provide expert domain knowledge to the tool suite. Currently, READEX supports region identifiers to distinguish rtsfis, phase identifiers to distinguish phase characteristics and input identifiers to distinguish executions with different application inputs. These identifiers will improve the tuning model by distinguishing rts's and assigning them to different scenarios to potentially select a better configuration. The domain knowledge also includes Application-level Tuning Parameters (ATP) that switch the application control flow and expose tuning potential in the target application.

This paper briefly presents DTA as the first step of the READEX tool suite. This outlines the precomputation of the tuning model by PTF and the READEX tuning plugin in Section 2.

2 DESIGN-TIME ANALYSIS

Pre-analysis steps are performed prior to DTA in which the application is instrumented with Score-P and analyzed from performance measurement for tuning potential. There is signficant instrumentation overhead caused by fine granular regions. This is reduced by creating a filter file consisting of these regions. Score-P reads this filter file and suppresses all measurements for those fine granular regions. Coarse granular program regions that cover most of the execution time and posess tuning potential are selected for dynamic tuning. For this purpose, we developed another Score-P profile based tool, `readex-dyn-detect`. The tool detects coarse granular regions and computes tuning potential for those regions by investigating dynamically changing main characteristics.

DTA is performed by PTF, a distributed framework consisting of a frontend and a hierarchy of analysis agents. The agents connect to the application through the Online-Access Interface of Score-P to send/receive measurement request/results and tuning requests from Score-P. The PTF frontend executes the READEX tuning plugin to peform DTA for certain aspects (such as tuning parameter, search strategy and significant regions) and given tuning objectives (Energy, CPU Energy, Execution Time, Energy Delay Product or Energy Delay Product Squared). It applies an online search, executing experiments with different configurations of the tuning parameters for a single program run. It evaluates tuning objectives of the rts's of the significant regions giving the best configuration

[1] Pre-print of the accepted paper. The final publication is available in the ACM digital library through http://dx.doi.org/10.1145/3085158.3091994
[2] www.readex.eu

SEM4HPC '17, Washington, DC, USA
 ISBN 978-1-4503-5000-6/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3085158.3091994

of the tuning parameters and there by generating tuning model used at runtime.

At first, the READEX plugin reads a READEX configuration file provided by the user and initializes the ranges of the tuning parameters. The READEX tuning plugin supports currently three tuning parameters: core frequency, uncore frequency and the number of OpenMP threads. Next, it defines a search space with a search algorithm (exhaustive, individual, random and genetic search) given by the user or loads default search algorithm (exhaustive search). Next, the plugin sets the objective to application tuning. It then reads the significant regions from the configuration file and creates experiments consisting of the values of the tuning parameters, the objective, and the rts's for which the objective will be evaluated. The experiments measure the objective value for all the valid rts's of the significant regions. To compute the consumed energy, the plugin aggregates the values returned by the designated processes of all the nodes for an MPI application.

After executing all the experiments, the plugin returns the best configuration for the phase and individual best for the rts's. In the end, the plugin creates dynamically three additional experiments by setting the tuning parameters for the phase and the individual rts's to their best configuration. The first experiment determines the dynamic energy saving that is achieved as compared to the static configuration setting. The other two experiments verify the reproducibility of the measured objective values.

3 CONCLUSION

This paper presented the Design-Time Analysis step of the READEX methodology for tuning the energy efficiency of HPC applications. In contrast to previous approaches [4], READEX focuses on runtime tuning guided by a tuning model that is pre-computed during application design-time.

ACKNOWLEDGMENTS

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

REFERENCES

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

[2] A. Knüpfer, C. Rössel, D. an Mey, S. Biersdorff, K. Diethelm, D. Eschweiler, M. Geimer, M. Gerndt, D. Lorenz, A. D. Malony, W. E. Nagel, Y. Oleynik, P. Philippen, P. Saviankou, D. Schmidl, S. S. Shende, R. Tschüter, M. Wagner, B. Wesarg, and F. Wolf. Score-p: A joint performance measurement run-time infrastructure for Periscope, Scalasca, TAU, and Vampir. In H. Brunst, M. Müller, W. E. Nagel, and M. M. Resch, editors, *Tools for High Performance Computing 2011*, pages 79–91. Springer, Berlin, 2012.

[3] Robert Mijaković, Michael Firbach, and Michael Gerndt. An architecture for flexible auto-tuning: The periscope tuning framework 2.0. In *Green High Performance Computing (ICGHPC), 2016 2nd International Conference on*, pages 1–9. IEEE, 2016.

[4] 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 C. Plessl, D. El Baz, G. Cong, J. M. P. Cardoso, L. Veiga, and T. Rauber, editors, *Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on*, pages 347–350, Piscataway, Oct 2015. IEEE.

[5] Joseph Schuchart, Michael Gerndt, Per Gunnar Kjeldsberg, Michael Lysaght, David Horák, Lubomír Říha, Andreas Gocht, Mohammed Sourouri, Madhura Kumaraswamy, Anamika Chowdhury, Magnus Jahre, Kai Diethelm, Othman Bouizi, Umbreen Sabir Mian, Jakub Kružík, Radim Sojka, Martin Beseda, Venkatesh Kannan, Zakaria Bendifallah, Daniel Hackenberg, and Wolfgang E Nagel. The READEX formalism for automatic tuning for energy efficiency. *Computing*, pages 1–9, 2017. DOI: 10.1007/s00607-016-0532-7.