Design-Time Analysis to improve Energy-efficiency of HPC Applications

Extended Abstract

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 is 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. Design-Time Analysis to improve Energy-efficiency of HPC Applications. In *Proceedings of ACM Symposium on High-Performance Parallel and Distributed Computing, Washington D.C. USA, June 2017 (HPDC'17),* 2 pages.
DOI: XX.XXX/XXX

1 INTRODUCTION

Energy management has become one of the dominant factors nowa-days in High Peformance Computing(HPC) systems. Large scale installation of powerful and heterogenous data centers and complex systems have led to enormous electrical power consumption. As a result, from energy-efficient hardware technology, extensive cooling solutions, improvement in applications' algorithm upto tuning applications are required to improve the overall energy-efficiency.

The European Union READEX[1] project develops a tool suite to improve energy efficiency by influencing tuning parameters during application execution at runtime [5]. 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 [3] as the common instrumentation and measurement infrastructure. During design-time, a *tuning model* is precomputed by using the Periscope Tuning Framework (PTF)[1, 4]. A novel tuning plugin of PTF, the READEX Tuning Plugin was developed to determine the best *configurations* or settings for the *runtime situations*(rts's), i.e., dynamic instances of *significant regions* [6]. 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.

This paper briefly presents DTA as the first step of the READEX methodology. This outlines the precomputation of the tuning model by PTF and the READEX tuning plugin in Section 2. Section 3 briefly presents results obtained for the BT benchmark of the NAS Parallel Benchmarks [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 of the tuning parameters and there by generating tuning model used at runtime.

*First two authors are students
[1] www.readex.eu

HPDC'17, Washington D.C. USA
2017. XXX-XXXX-XX-XXX/XX/XX…$15.00
DOI: XX.XXX/XXX

Table 1: Results of DTA for the BT-MZ benchmark using the READEX tuning plugin.

Significant regions	Energy for worst static configuration (1, 1.6)	Energy for best static configuration (4, 1.6)	READEX Energy	# of threads	Core frequency
exch_qbc	3245	6649	2760	1	2.4
x_solve	74219	41341	39962	4	2.0
y_solve	73536	39497	39497	4	1.6
z_solve	76393	40699	40386	4	2.0
SUM	**227393**	**128186**	**122605**		
Energy consumed by Phase	376722	284223			

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.

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 EVALUATION

This section presents the results obtained from `readex-dyn-detect` and DTA. The results are reported for hybrid version of the NAS Parallel Benchmark (NPB) suite.

First, `readex-dyn-detect` is applied to determine the significant regions. For BT-MZ, these regions are: `exch_qbc`, `x_solve`, `y_solve` and `z_solve`.

These regions are exported into the READEX configuration file so that the READEX tuning plugin can read it. The plugin performs 16 different experiments using the exhaustive search strategy and returns the best configuration (setting) for the OpenMP threads and the CPU frequency tuning parameters.

Columns two and three in Table 1 present the total energy consumed by the significant regions and the entire phase for the static worst and best configurations. The static configuration is applied for the entire phase and the worst configuration gives the worst energy consumption for the phase while the best configuration gives the best node energy. As we can see, the best setting is not necessarily the best setting for all the significant regions. This is where the READEX dynamic tuning can be applied. Columns 4 presents the best result for each significant region obtained with a region specific configuration (colmun 5 and 6).

The static energy saving amounts to 43.60% for all the significant regions and 24.60% for the phase region as compared to the worst static energy values in column 2. The energy savings obtained using the READEX tuning for the significant regions is 4.40% when compared to the best static energy for these regions in column 3.

The detailed steps of the design-time anylsis will be presented as live demo with the poster.

4 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 [5], READEX focuses on runtime tuning guided by a tuning model that is pre-computed during application design-time.

REFERENCES

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

[2] Haoqiang Jin and Rob F. Van der Wijngaart. 2004. Performance Characteristics of the Multi-Zone NAS Parallel Benchmarks. In *18th International Parallel and Distributed Processing Symposium (IPDPS).* DOI: https://doi.org/10.1109/IPDPS.2004.1302906 DOI: 10.1109/IPDPS.2004.1302906.

[3] 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. 2012. Score-P: A Joint Performance Measurement Runtime Infrastructure for Periscope, Scalasca, TAU, and Vampir. In *Tools for High Performance Computing 2011*, H. Brunst, M. Müller, W. E. Nagel, and M. M. Resch (Eds.). Springer, Berlin, 79–91.

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

[5] Y. Oleynik, M. Gerndt, J. Schuchart, P. G. Kjeldsberg, and W. E. Nagel. 2015. Run-Time Exploitation of Application Dynamism for Energy-Efficient Exascale Computing (READEX). In *Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on*, C. Plessl, D. El Baz, G. Cong, J. M. P. Cardoso, L. Veiga, and T. Rauber (Eds.). IEEE, Piscataway, 347–350. DOI: https://doi.org/10.1109/CSE.2015.55

[6] 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. 2017. The READEX formalism for automatic tuning for energy efficiency. *Computing* (2017), 1–9. DOI: https://doi.org/10.1007/s00607-016-0532-7 DOI: 10.1007/s00607-016-0532-7.

Design-Time Analysis to improve Energy-efficiency of HPC Applications

READEX
Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

Anamika Chowdhury; Madhura Kumaraswamy; Michael Gerndt
Technical University of Munich (TUM)

European Commission
Horizon 2020
European Union funding for Research & Innovation

Abstract

- Tune HPC applications dynamically for improved energy-efficiency and performance.
- Switching between configurations by exploiting dynamic characteristics of HPC applications.
- Develop tool aided auto-tuning methodology.
  - Design-time analysis
  - Runtime Application Tuning
- Detect during design-time, exploit during runtime.

THE READEX Tool-Suite

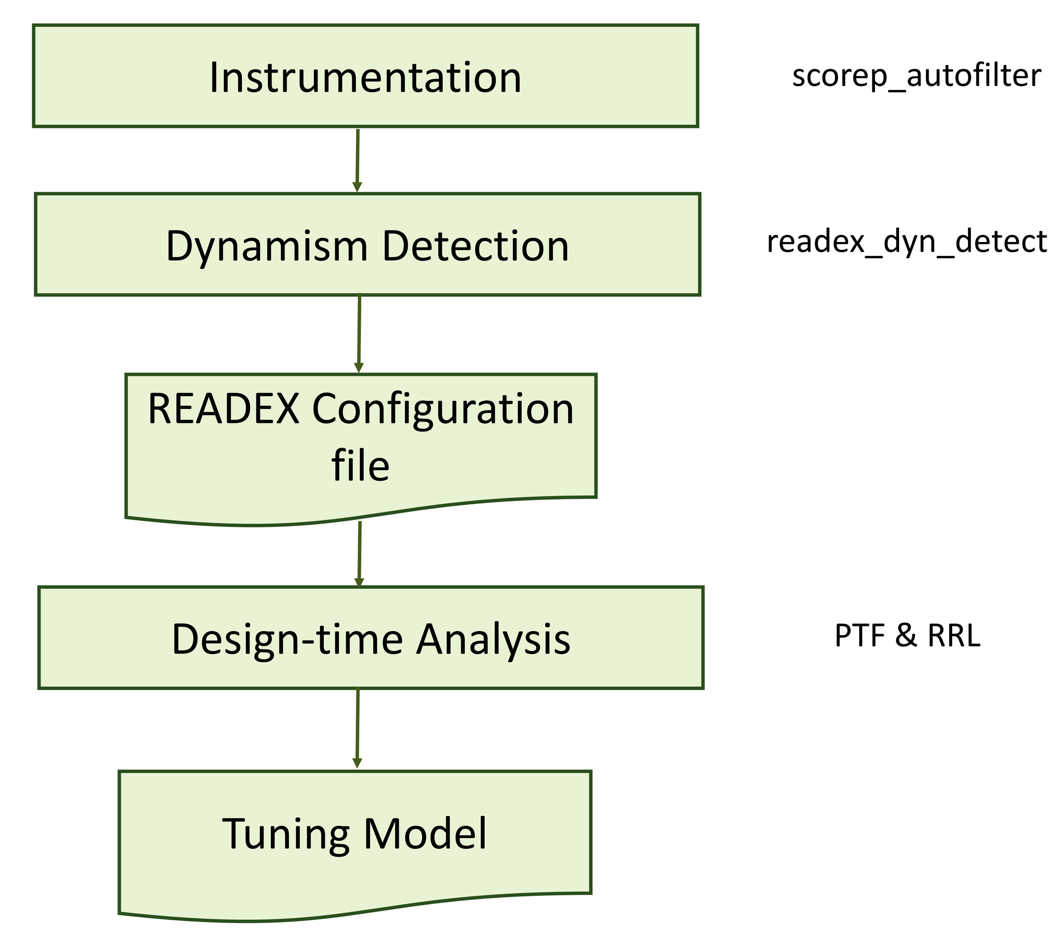
Design-Time Analysis

- Detect program regions having variations in characteristics.
- Determine different Runtime Situations (RTS) of the detected regions.
- Determine best configurations for RTSs.
- Classify RTS's based on similar configurations into scenarios.
- Encapsulate the scenario information into a tuning model.

Runtime Application Tuning

- Propagate the generated tuning model for the production run.
- Performed by the READEX Runtime Library
  - Lightweight.
  - Switch to the best configuration for a detected RTS retrieved from the tuning model.
- Calibration mechanism
  - Calibrates regions which were not seen during design-time.

Design-Time Analysis Workflow



Design-Time Analysis (DTA)

- Performed by the **Periscope Tuning Framework (PTF)**
  - Tunes performance and energy.
  - Evaluates alternatives online.
  - Supports different tuning strategies.
- **The READEX Tuning Plugin**
  - Multiple objectives.
  - Configurable search space via READEX configuration file.
  - Multiple search strategies for searching.
  - Tuning Parameters: core frequency, uncore frequency, # of threads
  - Experiments for selected configurations
  - Energy and time measured for all RTS's.
  - Identification of static best for phase and specific best configurations for RTS's.

Initial DTA Results

- Dynamism Detection:
  - performed by **readex-dyn-detect**.
  - detected significant regions: exch_qbc, x_solve, y_solve, and z_solve.
- Performed by the READEX Tuning Plugin
  - 16 experiments
  - The exhaustive search strategy
  - Returns best configurations for the no. of threads and core frequency

Table 1. Results of DTA for the BT-MZ benchmark using the READEX Tuning Plugin

Significant regions	Energy for worst configuration (1, 1.6)	Energy for best configuration (4, 1.6)	The READEX Energy	# of thread	Core frequency (GHz)
exch_qbc	3245	6649	2760	1	2.4
x_solve	74219	41341	39962	4	2.0
y_solve	73536	39497	39497	4	1.6
z_solve	76393	40699	40386	4	2.0
SUM	227393	128186	122605		
Energy for phase	376722	284223			

Discussion

- The static configuration is applied for the entire phase.
- The worst configuration
  - worst energy consumption for the entire phase.
- The best configuration
  - best energy consumption for the entire phase.
- The best setting for the phase
  - not necessarily for all the significant regions.
- Static energy savings w. r. t. worst energy consumption
  - 43.60% for all the significant regions.
  - 24.60% for the phase
- Dynamic energy saving due to switching configuration dynamically w. r. t. best energy consumption.
  - 4.40% for all the significant regions.

Conclusions

- Presents the Design-Time Analysis step of the READEX methodology for tuning to improve energy efficiency.
- READEX focuses on runtime tuning guided by a tuning model pre-computed during design-time.

Future Goals

- Inter-phase dynamism.
- Handling multiple input files.
- Domain knowledge specification.
  - Allows the user to provide domain knowledge as *identifiers.*
  - Application Tuning Parameters.
  - Input identifiers.

Contact Information

Robert Schoene
Technische Universitaet Dresden (TUD)
Email: Robert.schoene@tu-dresden.de
Phone: +49 (351) 463-42483

Collaborations

TUM, TECHNISCHE UNIVERSITÄT DRESDEN, IT4Innovations national supercomputing center, NTNU, intel, gns, ICHEC

Acknowledgements

- Funded by the European Union's Horizon 2020
- Grant agreement no. 671657

More Info

- www.readex.eu
- www.researchgate.net/project/READEX