Preface

Today advances in computer hardware are no more driven by increased operation frequencies. They are driven by parallelism and specialised accelerating features—whether by SIMD registers and compute units inside traditional CPU cores or by additional accelerator cards. In the field of high-performance computing (HPC), up to multiple thousands of these compute units can be combined to one big system via sophisticated high-speed networks nowadays to solve large computational problems within a reasonable time frame.

These advances steadily allow to solve more complex and even new problems. But it comes with the cost of complexity of the hardware, which increases the necessary programming effort to utilise the entire potential of these systems. Application developers have to write codes which can cope with parallelism keeping millions of processing elements running today. Therefore, extracting parallelism and efficient load balancing are becoming more and more an issue. At the same time the memory system holding computational data is also undergoing massive changes: Up to four levels of coherent caches can be seen in today's CPUs, while accelerators often use directly programmable caches. The arising problems are absolutely nontrivial to solve as program data structures may not fit for all CPUs and accelerators. And in a heterogeneous system, this may become even more complicated.

The European Technology Platform for High Performance Computing, an industry-led forum, emphasises in its 2013 Strategic Research Agenda the importance of the programming environment—including tools for performance analysis, debugging and automatic performance tuning—for the development of efficient, massively parallel and energy-efficient applications. This shows how vital tools in the field of HPC are not only for researchers in the academic field but also for more and more industrial users.

While traditional parallel programming models for distributed and shared memory systems as well as accelerators exist and are already standardised—just mentioning the Message Passing Interface (MPI) and Open Multiprocessing (OpenMP) API—they may not be easy to use for or even capable to exploit the potential of today's and upcoming HPC systems. So besides the traditional standards, new parallel programming models are developed—just mentioning dependency-driven task-based programming models like OmpSs or new functional programming approaches. These models may not only help programmers to ease the work of writing parallel programs but also allow the development of tools which can support them with the task of program parallelisation.

Since 2007 the International Parallel Tools Workshop provides once a year the opportunity for the leading HPC tool developers worldwide to exchange their experiences in optimisation techniques and development approaches. It covers the state-of-the-art development of parallel programming tools, ranging from debugging over performance analysis to fully and semi-automatic tuning tools as well as best practices in integrated developing environments for parallel platforms. The workshop is jointly organised by the High Performance Computing Center Stuttgart (HLRS)[1] and the Center for Information Services and High Performance Computing of the University of Dresden (ZIH-TUD).[2]

This book comprises a continuation of a successful series of publications that started with the first tools workshop in 2007. It contains contributed papers presented at the 8th International Parallel Tools Workshop,[3] held 1–2 October 2014 in Stuttgart, Germany. More than ten different tools covered different aspects of the software optimisation process and global community developments.

So, the Scalasca tool developers reported on their effort to bring their tool to the community instrumentation and measurement infrastructure Score-P—which was introduced to reduce tool development and maintenance effort for different HPC platforms itself. Allinea MAP was extended to support Hybrid MPI + OpenMP codes without adding large overheads and also added new power-usage metrics in the sense of green IT. Two tools introduced their approaches to help developers with parallelising serial programs: While DiscoPoP uses a bottom-up approach starting with compute units at the instruction level, Tareador uses an opposed approach refining tasks from code blocks until it explores enough parallelism for the target system.

Different tool developers reported on their progress in analysing large applications by aggregating and filtering important performance parameters and events. New multidimensional data aggregation methods were shown and evaluated for the trace analyser Ocelotl. A new approach using tracking analysis techniques to study performance characteristics of applications was presented, and trace and sampling data were combined to improve the detail level of performance data for large parallel programs while not increasing runtime and performance data sets too much at the same time. The effects of heterogeneous compute environments with accelerators were analysed by critical-blame analysis on the basis of the Score-P workflow, and program and runtime parameter tuning techniques were presented using the Periscope Tuning Framework. And finally a new idea of multidomain

[1] http://www.hlrs.de

[2] http://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/

[3] http://toolsworkshop.hlrs.de/2014/

performance analysis was introduced where performance data may be mapped not only to hardware topology or features but also to actual simulation data.

We appreciate the interesting contributions and fruitful conversations during the workshop from the speakers and all participants. Special thanks go to Steffen Brinkmann and Mathias Nachtmann for their support organising the workshop.

Stuttgart, Germany
January 2015

Christoph Niethammer
José Gracia
Andreas Knüpfer
Michael M. Resch
Wolfgang E. Nagel

http://www.springer.com/978-3-319-16011-5

Tools for High Performance Computing 2014
Proceedings of the 8th International Workshop on Parallel Tools for High Performance Computing, October 2014, HLRS, Stuttgart, Germany
Niethammer, C.; Gracia, J.; Knüpfer, A.; Resch, M.M.; Nagel, W.E. (Eds.)
2015, X, 229 p. 127 illus., 109 illus. in color., Hardcover
ISBN: 978-3-319-16011-5