# 10181 Executive Summary
# Program Development for Extreme-Scale Computing
## — Dagstuhl Seminar —

Jesus Labarta[1], Barton P. Miller[2], Bernd Mohr[3] and Martin Schulz[4]

[1] BSC, ES
jesus.labarta@bsc.es
[2] University of Wisconsin - Madison, USA
bart@cs.wisc.edu
[3] Forschungszentrum Jülich, DE
b.mohr@fz-juelich.de
[4] LLNL - Livermore, USA
schulzm@llnl.gov

**Abstract.** From May $2^{nd}$ to May $7^{th}$, 2010, the Dagstuhl Seminar 10181 "Program Development for Extreme-Scale Computing " was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. This paper provides an executive summary of the seminar.

**Keywords.** parallel programming, performance analysis, debugging, scalability

## 1 Motivation for the Seminar

The number of processor cores available in high-performance computing systems is steadily increasing. A major factor is the current trend to use multi-core and many-core processor chip architectures. In the November 2009 list of the TOP500 Supercomputer Sites, 98.4% of the systems listed have more than 2048 processor cores and the average is about 9300. While these machines promise ever more compute power and memory capacity to tackle today's complex simulation problems, they force application developers to greatly enhance the scalability of their codes to be able to exploit it. This often requires new algorithms, methods or parallelization schemes as many well-known and accepted techniques stop working at such large scales. It starts with simple things like opening a file per process to save checkpoint information, or collecting simulation results of the whole program via a gather operation on a single process, or previously unimportant order O(n2)-type operations that now quickly dominate the execution. Unfortunately many of these performance problems only show up when executing with very high numbers of processes and cannot be easily diagnosed or

Dagstuhl Seminar Proceedings 10181
Program Development for Extreme-Scale Computing
http://drops.dagstuhl.de/opus/volltexte/2010/2674

predicted from measurements at lower scales. Detecting and diagnosing these performance and scalability bottlenecks requires sophisticated performance instrumentation, measurement and analysis tools. Simple tools typically scale very well but the information they provide proves to be less and less useful at these high scales. Clearly, understanding performance and correctness problems of applications requires running, analyzing, and drawing insight into these issues at the largest scale.

Consequently, a strategy for software development tools for extreme-scale systems must address a number of dimensions. First, the strategy must include elements that directly address extremely large task and thread counts. Such a strategy is likely to use mechanisms that reduce the number of tasks or threads that must be monitored. Second, less clear but equally daunting, is the fact that several planned systems will be composed of heterogeneous computing devices. Performance and correctness tools for these systems are very immature. Third, the strategy requires a scalable and modular infrastructure that allows rapid creation of new tools that respond to the unique needs that may arise as extreme-scale systems evolve. Further, a successful tools strategy must enable productive use of systems that are by definition unique. Thus, it must provide the full range of traditional software development tools, from debuggers and other code correctness tools such as memory analyzers, performance analysis tools as well as build environments for complex codes that rely on a diverse and rapidly changing set of support libraries.

Many parallel tools research groups have already started to work on scaling their methods, techniques, and tools to extreme processor counts. In this Dagstuhl seminar, we wanted participants from Universities, goverment laboratories and industry to report on their successes or failures in scaling their tools, review existing working and promising new methods and techniques, and discuss strategies for solving unsolved issues and problems.

This meeting was the forth in a series of seminars related to the topic "Performance Analysis of Parallel and Distributed Programs", with previous meetings being the Dagstuhl Seminar 07341 on "Code Instrumentation and Modeling for Parallel Performance Analysis" in August 2007, Seminar 02341 on "Performance Analysis and Distributed Computing" held in August 2002, and Seminar 05501 on "Automatic Performance Analysis" in December 2005.

## 2 Summary of the Presentations

The seminar brought together a total of 46 researchers and developers working in the area of performance from universities, national research laboratories and, especially important, from three major computer vendors. The goals were to increase the exchange of ideas, knowledge transfer, foster a multidisciplinary approach to attacking this very important research problem with direct impact on the way in which we design and utilize parallel systems to achieve high application performance.

The program consisted of a total of 35 presentations. The talks during the "Demo Sessions" were special reports on performance and debugging tool scalability experiments. A few months before the seminar, tool providers were invited to perform experiments with their tools applied to scientific simulation codes executing on at least 10,000 processor cores and report on their findings and experiences at Dagstuhl. To make it easier to compare the different tools, participants were asked to analyze at least one of the following application codes: PEPC, a plasma physics code from Jülich Supercomputing Centre, Germany, and PFLOTRAN, a multiphase, multicomponent reactive flow and transport code from LLNL in the U.S.

To our knowledge, it is the first time that such a study of scalability of performance and debugging tools has been performed. We were surprised, but very pleased to see that twelve teams were successfully participating in this challenge.

**Session "Power and Tuning"**

- *Scalable Methods for Performance and Power Data Collection and Analysis*
  Karen L. Karavanic (Portland State University, US)
- *Online Adaptive Code Generation and Tuning*
  Jeff Hollingsworth (University of Maryland - College Park, US)

**Session "Scalable Tracing"**

- *ScalaTrace and Beyond: Ultra-scalable Tracing, Analysis and Modeling of HPC Codes*
  Frank Mueller (North Carolina State University, US)
- *Scalable Event Tracing on High-End Parallel Systems*
  Kathryn Mohror (LLNL - Livermore, US)
- *Benefits of Sampling in Tracefiles*
  Harald Servat (Barcelona Supercomputing Center, ES)

**Session "Languages and Tool Infrastructure"**

- *Development Environment for X10*
  Evelyn Duesterwald (IBM TJ Watson Research Center - Hawthorne, US)
- *Cray Debugging Support for Large Scale Systems*
  Luiz DeRose (Cray Inc. - Saint Paul, US)
- *Toward Performance Prediction of Tree-Based Overlay Networks on the Cray XT*
  Philip Roth (Oak Ridge National Lab., US)

**Session "The Need for Components"**

- *Scalable Tool and Middleware Development using Group File Operations*
  Michael J. Brim (University of Wisconsin - Madison, US)
- *Toward (More) Scalable, Autonomous Tool Infrastructure*
  Dorian C. Arnold (University of New Mexico - Albuquerque, US)

**Demo Session "Integrated Toolsets"**

- *PFLOTRAN Performance Analysis Using the Cray Toolset*
  Heidi Poxon (Cray Inc. - Saint Paul, US)
- *Analyzing PEPC with Open—SpeedShop with 12000 Cores*
  Jim Galarowicz (The Krell Institute - Ames, US)
- *PFLOTRAN Meets TAU*
  Allen D. Malony (University of Oregon, US)
- *Experiences with Scalasca at Scale*
  Brian J. N. Wylie (Jülich Supercomputing Centre, DE)
- *Identifying Scalability Bottlenecks In Large-scale Parallel Programs Using HPCToolkit*
  Nathan Tallent (Rice University, US)

**Demo Session "Tracing"**

- *Scalable Performance Analysis with the Vampir Toolset*
  Matthias S. Müller (TU Dresden, DE)
- *Using CEPBA-Tools to Analyze PEPC and PFLOTRAN at Large Scale*
  Judit Gimenez (Barcelona Supercomputing Center, ES)

**Demo Session "Analysis"**

- *Scalable in-situ Analysis Techniques*
  Todd Gamblin (LLNL - Livermore, US)
- *IBM High Productivity Computing Systems Toolkit*
  I-hsin Chung (IBM TJ Watson Research Center - Yorktown Heights, US)
- *Online Performance Analysis*
  Hans Michael Gerndt (TU München, DE)

**Demo Session "Debugging and Support Tools"**

- *Challenges and Successes in MRNet*
  Matthew LeGendre (University of Wisconsin - Madison, US)
- *Petascale Debugging - and Beyond?*
  David Lecomber (Allinea Software Ltd - Warwick, GB)

**Session "Measurement Techniques"**

- *Hybrid Parallel Performance Measurement and Analysis*
  Allen D. Malony (University of Oregon, US)
- *Effective Performance Measurement at Petascale using IPM2*
  Karl Fürlinger (University of California - Berkeley, US)
- *Combining PMPI Event Profiling and Clock Sampling in Scalasca*
  Zoltan Szebenyi (Forschungszentrum Jülich, DE)
- *Performance Diagnosis through Classification of Computation Bursts to Known Computational Kernel Behavior*
  Kevin A Huck (Barcelona Supercomputing Center, ES)

**Session "In-situ Analysis"**

- *Applications of Spectral Analysis in Data Acquisition, Multiplexing Hardware Counters and Architecture Simulation*
  Marc Casas (Barcelona Supercomputing Center, ES)
- *Maximizing Information/Data Ratio at Run-time*
  German Llort (Barcelona Supercomputing Center, ES)
- *Towards an Automatically Distributed Evaluation of Event Data*
  Roland Wismüller (Universität Siegen, DE)
- *Scalable Root-Cause Analysis*
  David Böhme (Forschungszentrum Jülich, DE)
- *Demo: Scalable Load-Balance Analysis with Libra*
  Todd Gamblin (LLNL - Livermore, US)

**Session "Towards Exascale"**

- *Tool Strategies for Sequoia and Beyond*
  Bronis R. de Supinski (LLNL - Livermore, US)
- *Short Presentation of Exatec*
  Bettina Krammer (University of Versailles, FR)
- *Reports from Various Exascale Computing Initiatives*
  Bernd Mohr (Jülich Supercomputing Centre, DE)

## 3 Conclusions

Despite the larger than normal number participants, the seminar was still very successful due to the very helpful and professional staff of Dagstuhl on one side and the dedicated professionalism and discipline of the participants on the other side. Every day, lively discussions and spontaneous computer demonstrations continued well beyond midnight. We want to note (and we are sure to speak for all participants) that the group meeting and residential aspects of Dagstuhl and the five-day format provide a continuity of thought and discussion unavailable in other conference, workshop, or meeting settings. At Dagstuhl, we have time for considered (and reconsidered!) dialogs whose impact last well beyond the meeting week.

A half-day excursion including a boat tour of the "Saarschleife", a Saar wine-tasting and a dinner at the Mosel river completed the program.