

# Topic 1

## Support Tools and Environments

Marian Bubak<sup>1</sup> and Thomas Ludwig<sup>2</sup>

<sup>1</sup> Institute of Computer Science and ACC CYFRONET, AGH Kraków, Poland

<sup>2</sup> Ruprecht-Karls-Universität Heidelberg, Germany

At present days parallel applications are becoming large, heterogeneous, with more and more complex structure of their components and complicated topology of communication channels. Very often they are designed for execution on high-performance clusters and recently we observe the explosion of interest in parallel computing on the grid.

Efficient development of this kind of applications requires supporting tools and environments. In the first stage, support for verification of correctness of communication structure is required. This may be followed by an automatic performance analysis. Next, these tools should allow to observe and manipulate the behavior of an application during run time what is necessary for debugging, performance measurement, visualisation and analysis. The most important problems are measurement of utilisation of system resources and inter-process communication aimed at finding potential bottlenecks to improve the overall performance of an application. Important issues are portability and interoperability of tools.

For these reasons elaboration of supporting tools and environments remains a challenging research problem.

The goal of this Topic was to bring together tool designers, developers, and users and help them in sharing ideas, concepts, and products in this field.

This year our Topic attracted a total of 12 submissions. In fact this is a very low number but we do not want to draw the conclusion that there is no more work necessary with support tools and environments.

From the total of 12 papers with accepted 3 as regular papers and 4 as short papers. The acceptance rate is thus 58%. The papers will be presented in two sessions. Session one focuses on performance analysis. For session two there is no specific focus, instead we find various topics here.

The session on performance analysis presents three full papers of well known research groups.

Hong-Linh Truong and Thomas Fahringer present the tool SCALEA which is a versatile performance analysis tool. The paper gives an overview over its architecture and the various features that guide the programmer through the process of performance tuning. Remarkable is SCALEA's ability to support multi-experiment performance analysis.

Also the paper by Philip C. Roth and Barton P. Miller has its focus on program tuning. They present DeepStart, a new concept for automatic performance diagnosis that uses stack sampling to detect functions that are possible

bottlenecks. DeepStart leads to a considerable improvement with respect to how quickly bottlenecks can be detected.

The issue of performance collection is also covered in a paper on the scalability of tracing mechanisms. Felix Freitag, Jordi Caubert, and Jesus Labarta present an approach for OpenMP programs where the trace contains only non-iterative data. It is thus much more compact and reveals performance problems faster.

In our second session we find papers that deal with various aspects of tools and environments.

The paper by A.J.G. Hey, J. Papay, A.J. Keane, and S.J. Cox presents a component based problem solving environment (PSE). Based on modern technologies like CORBA, Java, and XML the project supports rapid prototyping of application specific PSEs. Its applicability is shown in an environment for the simulation of photonic crystals.

The paper by Jozsef Kovacs, Gabor Kusper, Robert Lovas, and Wolfgang Schreiner covers the complex topic of parallel debugging. They present their work on the integration of temporal assertions into a debugger. Concepts from model checking and temporal formulas are incorporated and provide means for the programmer to specify and check the temporal behaviour of the program.

Jorji Nonaka, Gerson H. Pfitscher, Katsumi Onisi, and Hideo Nakano discuss time synchronization in PC clusters. They developed low-cost hardware support for clock synchronisation.

Antonio J. Nebro, Enrique Alba, Francisco Luna, and José M. Troya have studied how to adopt JACO to .NET. JACO is a Java-based runtime system for implementing concurrent objects in distributed systems.

We would like to thank the authors who submitted a contribution, as well as the Euro-Par Organizing Committee, and the scores of referees, whose efforts have made the conference and this specific topic possible.