# Preface

On the 23rd of April, 2001, the 6th Workshop on High-Level Parallel Programming Models and Supportive Environments (LCTES'98) was held in San Francisco. HIPS has been held over the past six years in conjunction with IPDPS, the Internation Parallel and Distributed Processing Symposium.

The HIPS workshop focuses on high-level programming of networks of workstations, computing clusters and of massively-parallel machines. Its goal is to bring together researchers working in the areas of applications, language design, compilers, system architecture and programming tools to discuss new developments in programming such systems.

In recent years, several standards have emerged with an increasing demand of support for parallel and distributed processing. On one end, message-passing frameworks, such as PVM, MPI and VIA, provide support for basic communication. On the other hand, distributed object standards, such as CORBA and DCOM, provide support for handling remote objects in a client-server fashion but also ensure certain guarantees for the quality of services.

The key issues for the success of programming parallel and distributed environments are high-level programming concepts and efficiency. In addition, other quality categories have to be taken into account, such as scalability, security, bandwidth guarantees and fault tolerance, just to name a few.

Today's challenge is to provide high-level programming concepts without sacrificing efficiency. This is only possible by carefully designing for those concepts and by providing supportive programming environments that facilitate program development and tuning.

Past results in parallel computing on one side and distributed systems on the other side present opportunities for an increased transfer of knowledge between the areas. In particular, cluster computing presents a promising framework for parallel computing where advances from distributed systems can be utilized. Achievements in the area of automated performance analysis and performance modeling for parallel systems, on the other hand, may contributed to advances in performance analysis of distributed systems. as well.

Future directions also include alternatives to current standardization practices, for example, by replacing client-server protocols with decentralized ones that may be more suitable for distributed systems. On the other hand, successful programming models, such as the shared-memory paradigm, should be investigated for new trends like cluster computing.

This workshop provides a forum for researchers and commercial developers to meet and discuss the various hardware and software issues involved in the design and use of high-level programming models and supportive environments.

HIPS'01 featured an invited talk and presentations of ten refereed papers. The papers were selected out of 20 submissions. 21 referees prepared multiple reviews for each submission. The recommendations of the referees determined an initial set of papers selected for acceptance. The members of the program committee were then given the option to resolve differences in opinion for the papers they refereed based on the written reviews. After the resolution process, the 10 papers included in the workshop proceedings were selected for presentation and publication. The papers cover the topics of:

- Concepts and languages for high-level parallel programming.
- Concurrent object-oriented programming.
- Distributed objects and components.
- Structured parallel programming (skeletons, patterns, etc.).
- Software engineering principles for parallel system.
- Automatic parallelization and optimization.
- High-level programming environments.
- Automated performance analysis and performance modeling.
- Debugging techniques and development tools.
- Distributed shared memory.
- Implementation techniques for high-level programming models.
- Operating system support for runtime systems and middleware.
- Architectural support for high-level programming models.
- Guarantees for Quality of Service in distributed environments.
- Security of communication for distributed execution.
- Fault tolerance in network computing.

## Steering Committee

## Program Committee

- Henri Bal, Vrije Universiteit, The Netherlands
- Arndt Bode, Technische Universität München, Germany
- Luc Bougé, ENS Lyon, France
- Helmar Burkhart, Universität Basel, Switzerland
- John Carter, University of Utah, USA
- Karsten Decker, ETH Zürich, Switzerland
- Michael Gerndt, Forschungszentrum Jülich, Germany
- Hermann Hellwagner, Universität Klagenfurt, Austria
- Francois Irigoin, Ecole des Mines de Paris, France
- Vijay Karamcheti, New York University, USA
- Peter Keleher, University of Maryland, USA
- Gabriele Keller, University of Technology, Sydney, Australia
- Piyush Mehrotra, ICASE / NASA Lanley Research Center, USA
- Frank Mueller, Lawrence Livermore National Laboratory, USA
- Susanna Pelagatti, Universita di Pisa, Italy
- Thierry Priol, IRISA, France
- Alexander Reinefeld, ZIB, Germany
- Martin Schulz, Technische Universität München, Germany
- Xian-He Sun, Illinois Institute of Technology, USA
- Domenico Talia, ISI-CNR, Italy
- George Thiruvathukal, DePaul University, USA

## External Reviewers

- F. Bodin
- J. Simon
- F. Schintke
- C. Perez
- R. Wismüller
- S. Rathmayer
- Silber
- R. Hood
- W. Smith
- Y. JEGOU
- K. Chanchio
- W. Karl
- R. Perego
- A. Zavanella
- M. Aldinucci

## Acknowledgments

The workshop chair would like to acknowledge the following people:

- the invited speaker for his voluntary contributions to the workshop;
- the steering committee for their support during the planning phase;
- Richard Gerber for making the START software available to assist the submission and review process;
- Alfred Hofmann for his work as a contact person to Springer LNCS;
- and last but not least the eager members of the program committee and the external reviewers.

The Chair,

April 2001

Frank Mueller
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-561
Livermore, CA 94551 (USA)
mueller15@llnl.gov
www.llnl.gov/CASC/people/mueller
+1.925.424.3642 (voice)
+1.925.423.2993 (fax)

[1] This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.