SNOW: a Parallel Programming Environment for Clusters of Workstations

1 Partners

Academic Partners

Prof. Dr. Wolfgang Schröder-Preikschat
Forschungszentrum Informationstechnik GmbH (GMD)
Forschungsinstitut für Rechnerarchitektur und Softwaretechnik (FIRST)
Kekuléstrasse 7 12489 Berlin, Germany
E-mail: `wosch@first.gmd.de`
Phone: +49 30 6392-1841 Fax: +49 30 6392-1805

Prof. Dr. Philippe Olivier Alexandre Navaux
Universidade Federal do Rio Grande do Sul (UFRGS)
Instituto de Informática (II)
Av. Bento Gonçalves 9500 91501-970 Porto Alegre, RS, Brazil
E-mail: `navaux@inf.ufrgs.br`
Phone: +55 51 316-6165 Fax: +55 51 316-1576

Prof. Antônio Augusto Medeiros Fröhlich
Universidade Federal de Santa Catarina (UFSC)
Departamento de Informática e de Estatística (INE)
Campus Universitário Trindade, CP 476 88049-900, Florianópolis, SC, Brazil
E-mail: `guto@inf.ufsc.br`
Phone: +55 48 331-9498 Fax: +55 48 331-9770

Prof. Dr. Sérgio Takeo Kofuji
Universidade de São Paulo (USP)
Laboratório de Sistemas Integráveis (LSI)
Av. Prof. Luciano Gualberto, 158, CP8174 05508-900, São Paulo, SP, Brazil
E-mail: `kofuji@lsi.usp.br`
Phone: +55 11 818-5667 Fax: +55 11 211-4574

Industrial Partners

Dr. Hans-Georg Paap
GENIAS Software GmbH
Computational Fluid Dynamics
Erzgebirgstrasse 2 93073 Neutraubling, Germany
E-mail: `hgp@genias.de`
Phone: +49 9401 9200-30 Fax: +49 9401 9200-92

Eng. Luiz Francisco Gerbase
ALTUS Sistemas de Informática Ltda
Departamento de Automação Industrial
Av. São Paulo, 555 90230-161, Porto Alegre, RS, Brazil
E-mail: `altus@altus.com.br`
Phone: +55 51 337-3633 Fax: +55 51 337-3632

2 Motivation

The parallel computing community has been using clusters of commodity workstations as an alternative to expensive massively parallel processors (MPP) for several years. While MPPs can rely on custom hardware to achieve high performance, their development follows a slow pace, mainly due to the small production scale. Clusters, in the other hand, benefit from the frenetic pace with which the workstation market evolves. As a matter of fact, when an MPP comes to the market, it is very likely that processor, memory and interconnection systems with similar features will already be available at the commodity market. Therefore, it seems evident that both technologies are going to converge into a single one.

However, when we compare the performance of parallel applications running on clusters and on MPPs, the figures show a quite different scenario: clusters are still far behind their expensive relatives. The Numerical Aerospace Simulation Facility from NASA has carried out a careful study on parallel computing performance. This study, better known as the NAS Parallel Benchmark, corroborates the superiority of MPPs.

Taking in consideration these two observations, we concluded that the gap between MPPs and clusters has its origin in the parallel programming software environment normally used in clusters. While MPPs rely on custom software specifically developed to support parallel applications on a given parallel architecture, clusters often apply the “commodity” principle also to the software. Commodity workstation software, however, has not been designed to support parallel computing.

3 Objectives

In this project, we intend to develop a comprehensive programming environment to support parallel computing on clusters of commodity workstations. This software environment shall include a parallel language and a run-time system to support the development of high performance parallel applications. In order to enable existing application to run on the proposed environment, it shall also support traditional standard interfaces from the parallel community, such as POSIX and MPI. Visual tools to configure and manage the environment shall also be considered. Each of these goals will be briefly discussed below.

- **Run-time support system:** cluster resources, e.g. processor time, storage area, and communication channels, shall be made available to parallel applications through a highly scalable run-time support system. This system shall support reconfigurations as to satisfy the requirements of particular parallel applications, yet delivering the expected performance [10] [14]. The aimed run-time system shall also contemplate parallel operations such as group communication and collective operations [1] [2].

- **Standard interfaces:** many existing parallel applications have been written considering standard interfaces. Independently from the development methodology and the programming environment adopted, most of these applications rely, in their lowest levels, on some subset of POSIX system calls to access local resources, and on some inter-node communication package, typically MPI. Thereafter, these two interfaces shall be supported by our run-time support system in order to bring several existing parallel applications to run in SNOW [12][16]. Nevertheless, only the interfaces shall be ported, not their traditional implementations.

- **Parallel programming language:** writing parallel applications using standard sequential languages enriched with communication libraries is not always adequate. Adopting a parallel language eases the application development at the same time it increases performance due to better resource utilization. Moreover, some optimizations are only possible when the compiler is aware of application parallelism. Thus, SNOW shall include a parallel programming language, offering a suitable mechanism to explore both fine and medium-grain concurrency. As to reduce the impact of a new language, our parallel language shall be a C++ extension, opening the way for distributed and parallel objects [6].

- **Management tools:** one of the most serious problems in a cluster environment is to keep all the paraphernalia working. Noise, heat, and a confusing set of wires are normal when “commodity” is involved. Since avoiding this is not possible, we will work to make this “chaos” manageable with a set of tools, which shall implement a central management console with a friendly graphical interface for SNOW [4].

- **Parallel and embedded applications:** a parallel programming environment is of no value if parallel applications cannot benefit of it. In oder to validate our environment, we intend to port and develop real parallel applications to run on it. Both of our industrial partners shall collaborate intensively with the requirements specification and design of our platform development, in such a way as to grant that SNOW will find its way through real applications.

Aimed application areas are computational fluid dynamics (GENIAS) and parallel embedded systems to control complex industrial processes (ALTUS).

4 Work plan

We believe we can achieve a parallel programming environment that matches the goals described above by merging our current researches. Each of the partners involved in this project is currently working is some particular aspect of the proposed parallel programming environment, however, our current research projects show small intersections. The activities that have to be carried out in order to achieve the proposed parallel programming environment are summarized in table 1, along with a time schedule and a task distribution sketch.

Period	Activity	Partners
4/2001 till **9/2001**	**1 - Requirement analysis and definition of protocols**	
	1.1 - to give the run-time support system an MPI adaption layer	GMD, UFSC, USP
	1.2 - to give the run-time support system a Posix adaption layer	GMD, UFSC, UFRGS
	1.3 - to integrate the run-time support into the environment	GMD, UFSC
	1.4 - to integrate the parallel language into the environment	UFRGS
	1.5 - to integrate the management tools with the environment	USP
	1.6 - to integrate the applications with the environment	GENIAS, ALTUS
10/2001 till **3/2003**	**2 - Parallel programming environment development**	
	2.1 - run-time support system	GMD, UFSC
	2.2 - MPI adaption layer for the run-time support system	UFSC, USP
	2.3 - Posix adaption layer for the run-time support system	UFSC, UFRGS
	2.4 - parallel language	UFRGS
	2.5 - cluster management tools	USP
	2.6 - applications	GENIAS, ALTUS
4/2003 till **9/2003**	**3 - Parallel programming environment integration**	
	3.1 - run-time support system and the MPI adaption layer	GMD, UFSC, USP
	3.2 - run-time support system and the Posix adaption layer	GMD, UFSC, UFRGS
	3.3 - run-time support system into the environment	GMD, UFSC
	3.4 - parallel language into the environment	UFSC, UFRGS
	3.5 - cluster management tools into the environment	UFSC, USP
	3.6 - applications into the environment	UFSC, GENIAS UFRGS, ALTUS
10/2003 till **3/2004**	**4 - Parallel programming environment evaluation**	
	4.1 - software engineering aspects	ALL
	4.2 - performance	ALL
	4.3 - applicability	GENIAS, ALTUS

Table 1: Time and task schedule.

5 Expected Results

By executing the proposed project, we expect to construct a comprehensive parallel programming environment that shall overtake the traditional Unix + MPI + User-level Communication Package environment in the following aspects:

- **Performance:** Snow shall perform better than the traditional environment because, with a tailor-made run-time support systems, applications will face far less overhead. Besides, the rigid structure imposed by ordinary operating systems such as Unix and Windows NT prevents innumerable optimizations.

- **Correctness:** since Snow can be tailored to any given application, and do not need to foresee general conditions like the ordinary environment, we expect the resulting system to be far more compact, yielding less space for bugs.

- **Usability:** Snow management tools shall make the cluster environment easier to use.

- **Programability:** SNOW parallel programming language and run-time support system shall enable programmers to express application parallelism in a more natural and effective way.
- **Scalability:** because SNOW will be developed considering both software and hardware scalability, it shall be able to handle increases in the level of parallelism.

An overview of the expected parallel environment is shown in figure 1. It considers the target cluster to be connected by a service network (FAST-ETHERNET) that will be used to provide access to the server and also for management purposes. Besides the services network, the work-nodes are also connected to a high-speed network that will support communication among the processes of the parallel application. MYRINET will be initially considered for the high-speed network, but further versions shall contemplate SCI and ATM as well. A parallel application running on the cluster will have access to three interfaces: DPC++ (the parallel programming language), MPI and EPOS (the run-time support system). EPOS may be present as a native operating system for the IX86 or may run as a guest operating system on LINUX. Besides processes from the parallel application, work-nodes may also run management agents.

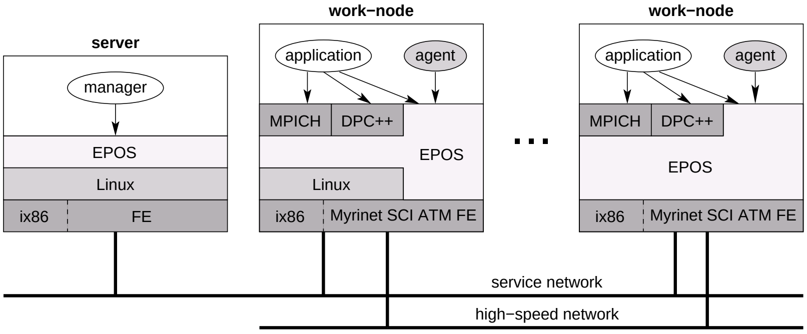

Figure 1: Parallel programming environment overview.

6 Short Bibliography of Project Leaders

Professor Dr. Wolfgang Schröder-Preikschat is full professor at the University of Magdeburg, where he coordinates the Institute for Distributed Systems. He received a Diplom degree in Computer Science from the Technical University of Berlin in 1981, and a Ph.D. in Computer Science from the same university in 1987. His research interests include embedded, distributed/parallel operating systems, object-oriented software construction, communication systems, and computer architecture.

Professor Dr. Philippe Olivier Alexandre Navaux is full professor at the Federal University of Rio Grande do Sul, where he currently occupies the position of Director of the Institute for Informatics. He received a B.Sc. in Electric Engineering from the Federal University of Rio Grande do Sul in 1970, a M.Sc. in Physics from the same university in 1973, and a Ph.D. in Computer Science from the Institut National Polytechnique de Grenoble in 1979. His research interests include computer architecture, operating systems, and parallel processing.

Professor Dr. Sérgio Takeo Kofugi is assistant professor at the University of São Paulo, where he coordinates the Digital Systems Division of the Laboratory for Integrated Systems. He received from the University of São Paulo a B.Sc. in Physics in 1981, a M.Sc. in Electrical Engineering in 1988, and a Ph.D. in Electrical Engineering in 1995. His research interests include high performance computer architectures and networks.

Professor Antônio Augusto Medeiros Fröhlich is assistant professor at the Federal University of Santa Catarina. He received a B.Sc. in Computer Science from the Federal University of Rio Grande do Sul in 1992 and a M.Sc. in Computer Science from the Federal University of Santa Catarina in 1994. He is currently working towards the Ph.D. at the Technical University of Berlin. His research interests include operating systems and software engineering in the realm of parallel and embedded computing.

References

[1] M. Barreto, P. Navaux, and M. Rivière. DECK: a New Model for a Distributed Kernel Integrating Communication and Multithreading for Support of Distributed Object-Oriented Applications with Fault Tolerance. In *Proceedings of IV CACIC*, Neuquén, Argentina, 1998.

[2] M. Barreto, R. Ávila, R. Cassali, A. Carissimi, and P. Navaux. Implementation of the DECK Environment with BIP. In *Proceedings of the First Myrinet User Group Conference*, pages 82–88, Lyon, France, Sept. 2000. Rocquencourt, INRIA.

[3] M. Barreto, R. Ávila, and P. Navaux. The MultiCluster Model to the Integrated Use of Multiple Workstation Clusters. In *Proceedings of PC-NOW'2000*, Cancun, Mexico, May 1999.

[4] V. B. Bernal, S. T. Kofuji, G. M. Sipahi, and A. Anderson. PAD Cluster: an open, modular and low cost high performance computing system. In *Proceedings of the 11th Symposium on Computer Architecture and High Performance Computing*, Natal, Brazil, Sept. 1999.

[5] L. Büttner, J. Nolte, and W. Schröder-Preikschat. ARTS of PEACE: A High-Performance Middleware Layer for Parallel Distributed Computing. *Journal of Parallel and Distributed Computing*, 59(2), 1999.

[6] G. Cavalheiro and P. Navaux. DPC++: uma Linguagem para Processamento Distribuído. In *Proceedings of the 5th Symposium on Computer Architecture and High Performance Computing*, Florianópolis, Brazil, 1993.

[7] A. A. Fröhlich, R. B. Avila, L. Piccoli, and H. Savietto. A Concurrent Programming Environment for the i486. In *Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis*, Orlando, USA, July 1996.

[8] A. A. Fröhlich and W. Schröder-Preikschat. SMP PCs: A Case Study on Cluster Computing. In *Proceedings of the 24th Euromicro Conference - Workshop on Network Computing*, pages 953–960, Västeras, Sweden, Aug. 1998.

[9] A. A. Fröhlich and W. Schröder-Preikschat. EPOS: an Object-Oriented Operating System. In *Proceedings of the 2nd ECOOP Workshop on Object-Orientation and Operating Systems*, pages 38–43, Lisbon, Portugal, June 1999.

[10] A. A. Fröhlich and W. Schröder-Preikschat. High Performance Application-oriented Operating Systems – the EPOS Aproach. In *Proceedings of the 11th Symposium on Computer Architecture and High Performance Computing*, pages 3–9, Natal, Brazil, Sept. 1999.

[11] A. A. Fröhlich and W. Schröder-Preikschat. Tailor-made Operating Systems for Embedded Parallel Applications. In *Proceedings of the 4th IPPS/SPDP Workshop on Embedded HPC Systems and Apllications*, pages 1361–1373, San Juan, Puerto Rico, Apr. 1999.

[12] E. Horta and S. T. Kofuji. Using a reconfigurable switch to improve MPI performance. In *Proceedings of JICS'98*, volume 3, North Caroline, U.S.A., Oct. 1998.

[13] J. Nolte and W. Schröder-Preikschat. Dual Objects – An Object Model for Distributed System Programming. In *Proceedings of the Eigth ACM SIGOPS European Workshop, Support for Composing Distributed Applications*, 1998.

[14] F. Schön, W. Schröder-Preikschat, O. Spinczyk, and U. Spinczyk. Design Rationale of the PURE Object-Oriented Embedded Operating System. In *Proceedings of the International IFIP WG 10.3/WG 10.5 Workshop on Distributed and Parallel Embedded Systems*, Paderborn, Germany, Oct. 1998.

[15] W. Schröder-Preikschat. *The Logical Design of Parallel Operating Systems*. Prentice-Hall, Englewood Cliffs, U.S.A., 1994.

[16] M. Torres and S. T. Kofuji. The barrier Synchronization Impact on the MPI-Programs performance using a cluster of workstations. In *Proceedings of the International Symposium on Parallel Architecture, Algorithms and Networks*, Taipei, Taiwan, 1997.