PVM in Germany and at Technical University Munich

Arndt Bode, Thomas Ludwig
Institut für Informatik
Lehrstuhl für Rechnertechnik und Rechnerorganisation
Technische Universität München
80290 München
Tel.: (089) 2105-8240, Fax: (089) 2105-8232
e-mail: {bode|ludwig}@informatik.tu-muenchen.de

1. PVM in Germany

The following text will give some insight into activities of the PVM users in Germany. The overview presented is far from being complete and all information was collected by directly contacting persons known to use PVM. However, more than a dozen of sites at universities and research institutes sent us material about their programs, hardware configurations etc. They also gave details on their experiences using PVM and on their expectations concerning future versions of PVM. We hope that this data has to some extend the quality of being representative for many PVM users in Germany.

For our survey we got information material from people at:

> RWTH Aachen, Universität Augsburg, Freie Universität Berlin, Fachgruppe Bionik und Evolutionstechnik Berlin, Technische Universität Braunschweig, Universität Erlangen-Nürnberg, Technische Universität Hamburg-Harburg, BASF at Ludwigshafen, DLR Oberpfaffenhofen (German Aerospace Research Institute), Universität Paderborn, Universität Rostock, Rechenzentrum der Universität Stuttgart, Universität Tübingen, Rechenzentrum der Universität Tübingen, DESY in Zeuthen.

PVM users from these groups can be divided into several classes:

- pure users, i.e. application programmers that just use PVM as is;
- researchers who use PVM to make research on programming environments in general or who use it as a basis for new environments;
- researchers who provide support for the first two classes by designing and implementing tools or add-ons for PVM.

Let us first have a closer look onto the applications being implemented with PVM. Not astonishing, many of them belong to the class of compute intensive programs. We can find

- calculation of electromagnetic fields (Hamburg-Harburg);
- particle image velocimetry (Hamburg-Harburg);

- calculation of artificial neural networks (Berlin);
- radiosity and geometrical problems (Paderborn);
- computational fluid dynamics (Stuttgart, Aachen, Rostock);
- distributed simulators for Petri-nets (Erlangen);
- calculation with particle physics (Zeuthen).

A nice application which is not compute intensive is done at BASF in Ludwigshafen. They use the distributed environment of PVM to support the system aministration of their workstation cluster. This provides more efficiency and robustness than using e.g. rsh-commands for this purpose. However, there is no detailed information on this, especially not on the problem of how to guarantee security.

A comparison of PVM with other programming environments is done at many sites but often only for porting programs to or from other machines or for improving performance of applications. However, some researchers also do a more profound investigation of PVM and compare it to P4 or MPI or compare different hardware architectures (e.g. networking facilities) with the same software. At Technische Universität Braunschweig they make a sound investigation of communication mechanisms and of the functionality of various programming models, comparing PVM and Express with native UDP and TCP programming. Two sites use PVM to develop new environments:

- support of distributed functional calculations by providing an asynchronous RPC-mechanism with call syntax known from thread packages (Tübingen);
- support of object oriented distributed programming with inheritance (Berlin).

Both projects use PVM as an implementation vehicle and put additional layers onto the user interface.

There is only one site (except TUM), which is known to develop tools for PVM. At the university of Augsburg a tool is designed to support a dynamic distribution of program modules onto workstations. Various strategies are examined to find a user-friendly load distribution, i.e. one that has only small interference with interactive users. The tool is integrated into Hence and is available as public domain software.

Most of the sites use heterogeneous workstation environments, i.e. they put together all their resources for production runs of their code. We can find Sun Sparc, SunMP, SGI Challenge, Convex, HP, IBM SP-1 and SP-2. Amazing and maybe typical for Germany is the fact that many institutions use Parsytec systems with Transputer and PowerPC. A big installation is situated in Paderborn with an SC of 320 T800 and a GC of 1024 T805 and 200 PowerPC. Also Hamburg-Harburg uses an SC of 128 T800 and a GC with 128 PowerPC. Developers at these sites use PVM under Parix and usually complain about its low performance. Some of the researchers report also a special need for I/O-facilities. With the particle image velocimetry pictures are produced that need 1 GByte each. Up to now there is no good solution available how to meet this requirement.

Considering PVM itself most users seem to be satisfied with its functionality. They appreciate the possibility to keep already available Fortran-codes and to parallelize them by only adding some message passing commands. However, several users complain about the Fortran77 interface of PVM because they can not use the modern features of Fortran90 (e.g. pointers). Troubles are mainly caused by newer constructs (e.g. shared virtual memory) or newer hardware architectures (e.g. SunMP, SGIMP). PVM provides an easy to understand interface which can quickly be learned even by non computer scientists. Most users complain about lack of tool support. They think that Hence is in its current version not a useful tool for program design. They appreciate e.g. XPVM but consider it to be only a first step because of its low speed. Especially sophisticated tools for debugging and performance analysis are standing on top of the wish list.

All users are aware of MPI, some of them already tried it or use it regularly. The reports show that users would like to profit from MPI's advanced group management, the collective functions, the topology features, and the possibility to encapsulate libraries. Thus, they intend to migrate from PVM to MPI as soon as there are enough stable and efficient implementations. There was a considerable request for tools automatically porting PVM to MPI or a PVM compatibility interface on top of MPI.

Detailed informations and email addresses of contact persons can be obtained from Thomas Ludwig.

2. PVM at Technische Universität Munich

In 1990, supported by the German Science Foundation, the project "'Methods and Tools for the Use of Parallel Systems"' was started. Main focus of the project was the development of methods and tools to enhance productivity when developing parallel programs for systems with distributed memories. After the consideration of multiprocessor systems in the first years, since about 1992 homogenous and heterogenous networks of workstations are being considered. Today, about 100 academic staff are working in 12 groups at the departments of Computer Science and Electrical Engineering of Technische Universität München as well as the industrial partners corporate Research of Siemens AG, Munich and European Supercomputer Development Center ESDC of Intel GmbH, Munich [3].

Research is subdivided into two project areas:

- Methods and tools (parallel algorithms and complexity, specification and verification tools for parallel and distributed programs based on algebraic and Petri-net-methods, parallel and distributed operating systems, development and runtime tools, methods for parallel theorem proving, efficient parallel algorithms)
- applications (CAD for VLSI, parallel databases, numerical simulation).

The main machines used are: a network of more than 600 workstations of all brands, a number of distributed memory multiprocessors (iPSC/860, iPSC/2, PARSYTEC SC, POWER EXPLORER) and virtual shared memory multiprocessors (KSR-1, KSR-2). Within the workstations, a homogenous network of 120 HP workstations, subdivided into 12 subnetworks is available. In the near future, an ATM-crossbar-switch with heterogenous workstations and 155 Mbit/s will be available. The main programming models used are:

- PVM
- MPI
- P4
- NX
- PARIX
- MMK/X

NX, PARIX and MMK are message-oriented programming libraries developed for specific multiprocessor systems. PVM, MPI, P4, NXLib and MMK/X are message-oriented programming libraries for networks of workstations. For portability reasons NXLib and MMK/X were implemented to support the specific multiprocessor programming models onto networks of workstations ([1]). On the other hand, PVM is also available for programming the multiprocessor architectures. In the following chapters, the main activity areas related to PVM are shortly scatched.

The integration of transparent dynamic ressource scheduling mechanisms into PVM and other message oriented communication libraries is a first area of interest. The introduction of scheduling systems such as CONDOR and CODINE for batch-oriented and parallel programming covering distributed scalable IO, dynamic load balancing and fault tolerance is currently under work (compare companion paper by G. Stellner).

A parallel theorem prover SPTHEO has been implemented under PVM 3.3 using 120 HP workstations. After initial problems using beyond 70 nodes, SPTHEO is currently one of the fastest theorem provers of the world. It has been tested with a collection of ”‘Benchmark-problems”’, most of which - for runtime reasons - may not be solved on sequential processors. A thorough evaluation will be published soon (Suttner, 1994 [2]).

The group working on CAD for VLSI has especially considered layout synthesis, testing, simulation and partitioning. For the different purposes, up to 100 workstations have been used with P4 and PVM. In most cases, linear speedup could be obtained [5].

The group working on CAD for VLSI has especially considered layout synthesis, testing, simulation and partitioning. For the different purposes, up to 100 workstations have been used with P4 and PVM. In most cases, linear speedup could be obtained [5].

In the parallel database-group, the architecture of MIDAS has been designed (compare companion paper by M. Pawlowski [4]). The main problem of the concepts of a parallel data base relate to implementation issues:

- Necessity of shared memory communication for internal purposes such as cache-management in addition to the PVM skeleton
- separation of users and database-management processing.

A number of programming tools (performance analysis, visualization, debugging, load balancing) are being implemented to support the program development process in PVM and MPI environments.

In the numerical applications group, a study and implementation on the parallel solution of 3D partial differential equations on a network of workstations using PVM has been performed. The results obtained show, that performance beyond CRAY YMP may be obtained [6].

3. References

[1] G. Stellner, A. Bode, S. Lamberts, T. Ludwig: Developing Applications for Multicomputer Systems on Workstation Chesters, Gentzsch, Harms (eds.). High Performance Computing and Networking, II, 286 - 292, Springer LNCS Vol. 797, 1994

[2] C. B. Suttner, I. Schumann: Parallel Automated Theorem Proving, In: Parallel Processing for Artificial Intelligence, Elsevier, 1994

[3] A. Bode, M. Dal-Cin: Parallel Computer Architectures. Springer LNCS Vol. 732, 1993

[4] M. Pawlowski, R. Bayer: Parallel Sorting of Large Volumes on Distributed Memory Multiprocessors, Springer LNCS Vol. 702, 246 - 264, 1993

[5] H. Spruth, F. Johannes: Parallel Routing of VLSI circuits based on net independing, Proc. IPPS, 1994

[6] M. Griebel, W. Huber, T. Störtkuhl, C. Zenger: On the parallel solution of 3D PDEs on a network of workstations and on vector computers, Springer LNCS Vol. 732, 1993