FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

On the Value of Diversity: An Insider's View of High-Performance Computing

Friedel Hoßfeld

KFA-ZAM-IB-9611

Mai 1996
(Stand 30.05.96)

Proceedings of the 10th Annual International Symposium on High Performance Computers, IEEE Canada, 1996, CD-ROM.

On the Value of Diversity: An Insider's View of High-Performance Computing

F. Hossfeld

Central Institute for Applied Mathematics
Research Centre Jülich (KFA)
D-52425 Jülich, Germany
e-mail: f.hossfeld @ kfa-juelich.de

Abstract

KFA is one of the largest big-science research centers in Europe. Its scientific and engineering activities are ranging from fundamental research to applied science and technology. KFA's Central Institute for Applied Mathematics (ZAM) is running the large-scale computing facilities, central servers, and network systems at KFA. ZAM is also providing supercomputer capacity to the scientific community all over Germany; this broad service has been a novelty in 1987 when a new user meta-structure named HLRZ ("Höchstleistungsrechenzentrum") has been established in order to further promote computational science and engineering in Germany. In 1996, ZAM's facilities are upgraded, in addition to a 16-node IBM SP2 and a 140-node Intel Paragon, by a heterogeneous new supercomputer complex consisting of CRAY J90 (20 CPUs), T90 (12 CPUs), and T3E (512 processors). Based on the recent recommendations of the German Science Council, also several large university computing centers are preparing the necessary steps to reenforce the apex of the structural pyramid of modern scientific computing. - While the successes of the past two decades in scientific supercomputing have been achieved essentially by the technical breakthrough of the vector-supercomputers, the future of supercomputing is focussed on massively parallel computers. Besides system architecture, node performance, and interconnection topology, important issues are genuine parallel algorithms, programming tools, and programming models. Message Passing on parallel computers with distributed memory is the only efficient programming paradigm available today; from a user's point of view, however, rather the concept of shared virtual memory will be capable to serve as the effective basis in order to bring computing on massively parallel systems from just a computer science toy to the technological breakthrough. - Since large applications, in general, are algorithmically heterogeneous, they will probably benefit best from the diversity of computer structures if these are cooperating and combined via high speed links to heterogeneous supercomputing ensembles.

1 Research at KFA and ZAM's Mission in Supercomputing

The Research Centre Jülich (KFA) is one of the largest big-science centres in Europe carried by the German Federal Government and the local State Government. Today, KFA takes the function and character of a national research laboratory with highly interdisciplinary research and manifold national and international interactions and cooperations with universities, research institutes, and industry. Its research and engineering activities are focussing on five research areas: properties of matter, information technology, energy, environment, and life sciences. Computational Science and Engineering has received high recognition and priority at KFA since many years. In 1987, as kind of a user meta-structure, not as an autonomous institution, the "High Performance Computing Centre" (in German named "Höchstleistungsrechenzentrum": HLRZ) was established jointly carried by the national labs KFA, GMD, and DESY. Hence, KFA is also providing the production supercomputers to support projects in computational science and engineering of the scientific community all over Germany.

The Central Institute for Applied Mathematics (ZAM) at KFA Jülich is responsible for the planning, installation, management, and operation of the central computer systems and of the KFA-wide computer networks. Its mission as a central institute and the needs for scientific services at KFA define ZAM's research and development projects in the fields of mathematics, computing, and computer science. ZAM also runs the supercomputer systems as provided to HLRZ by KFA; presently, about 150 approved projects in computational science are granted via HLRZ on the supercomputers at KFA spreading these invaluable resources over universities and research institutions throughout Germany. Since the 60s, ZAM has run one of the most powerful scientific computing centres in Europe.

Until recently, two vector-supercomputers CRAY Y-MP/M94 and Y-MP8/864 and a massively parallel system Intel Paragon XP/S10, a central unix server IBM SP2 as well as an IBM mainframe ES/9000-620 together with a host of smaller systems for special purposes like visualization and communications have been available to the users. The Cray Y-MP vectorcomputers are just being replaced by a new supercomputer complex consisting of 20-CPU CRAY J90, 12-CPU CRAY T90, and 512-node CRAY T3E. Thus, KFA is returning to the apex of the computing pyramid as it used to be for decades.

However, supporting the various information processing tasks in a scientific-technical environment requires more than providing access to powerful computer systems, large storage capacity and adequate software functions. To meet these needs, ZAM provides and operates various data communication networks with the following functions, on the one hand interactive access to information services and remote computers using workstations, PCs, or terminals at the work-place; and on the other hand computer-computer communication inside KFA and with external data processing systems. The KFA-wide network KFAnet, a fast Local Area Network based on Ethernet and FDDI (optical fibre backbone), is open to all institutes and organizational units at KFA. KFAnet is being converted into ATM technology.

ZAM is participating in the Regional ATM Testbed Project funded by the Federal Government in five different geographical regions in Germany in order to promote broadband communications in science and research in advance of the German Broadband Science Network (B-WiN). Thus, access to the public networks of the German Telekom and to Internet and the B-WiN is another important service provided by ZAM. Facing the utmost importance of data communication in the scientific-technical environment at KFA, these services must be capable to guarantee a continuously and efficiently running network. There must also be the skill to master the underlying communication techniques, and, if necessary, to develop components to serve requirements which commercial systems do not meet. Such developments and the exploration of new communication methods and techniques are subjects of corresponding research and development projects of ZAM.

A major permanent task is to provide suitable programming languages and tools, to investigate programming methodologies, to maintain basic software, especially in the application fields of graphics, text processing, and databases, and to support users in solving complex programming problems. The programming language Fortran is still predominant in scientific and technical fields. Activities concentrate on adequate compilers and debuggers, on teaching the language and programming techniques, and on assessing future developments of this language in particular with respect to supercomputing. In the area of programming techniques and tools, the methodology of programming aims at the optimization of programs. ZAM provides and, to a great deal, also develops tools for these requirements. Supercomputer simulations need efficient visualization techniques; therefore, ZAM develops software oriented along the user requirements utilizing high-performance visualization systems. Time-dependent processes can be visualized by means of techniques established in a Video Lab. The ZAM Information Centre is the active interface to the users. ZAM also offers classes, training courses, and seminars on many fields of information processing and computing.

ZAM works closely together with many other KFA institutes and HLRZ projects to select or develop appropriate solution methods and mathematical software, especially in the area of mathematical modelling, algorithm and program development and optimization, evaluation of hardware and software as well as support of large-scale applications. Intensive collaborations, partly by contract, exist with manufacturers and software companies. ZAM is also involved in international and national research cooperations with academia and industry funded by the European Union and the German Ministry for Research and Technology.

2 Computational Science & Engineering: The Methodological Tripod

The strategy of supercomputing development at KFA has been strongly guided by the computational needs of science and engineering. During the past decades, modelling and computer simulation, more comprehensively identified as Computational Science and Engineering, has grown and established itself as the third category of scientific methodology. This innovative discipline fundamentally supports and supplements theory and experiment, as the two traditional categories of scientific investigation, in a qualitative and quantitative manner while integrating these into the *methodological tripod* of science and engineering. Its main instruments are supercomputers; its primary technique is simulation [1].

The various strategic position papers [2-4] and government technology programs in the U.S., in Europe, and Japan claim that the timely provision of supercomputers to science and engineering and the ambitious development of innovative supercomputing hardware and software architectures as well as new algorithms and effective programming tools are an urgent research-strategic response to grand challenges arising from huge scientific and technological barriers [5].

Thus, this tripod has proved to provide scientific research and technology with the stable methodological basis and the instrumental laboratory to effectively approach the solutions of the complex problems which are crucial to the future of science, technology, and society. It is essential to recognize that the scientific knowledge and the technical skills, which are available in the field of supercomputers and their applications and which will be further gained from scientific and technical engineering projects within universities and research institutions, will be a crucial factor for the industry in order to meet the requirements of international economic competition especially in the area of high-tech products.

Despite the remarkable investments in research centers and universities in building up supercomputing power and skills and also despite some sporadic efforts in the industry concerning supercomputing in Europe, it took until the 90s that the U.S. and European Union as well as national governments started non-military strategic support programs like HPCC, HPCN, and HPSC [6-8]. Their goals are also to enhance supercomputing as an innovative technology in science and engineering by stimulating the technology transfer from universities and research institutions into industry and by increasing the fraction of the technical community which gets the opportunity to develop the skills which are required to efficiently access the high-performance computing resources.

3 The Infra-Structural Pyramid

For the first time in computing history, due to the ever growing diversity of technologies and products, we will be able to build a balanced pyramid of computing power in scientific and technical computation in which each element of the pyramid consistently supports the others [9]. At the apex of the pyramid resides the highest level of compute power which can be realized by the computer architects and the industry with respect to innovative hardware and efficient software, thus targeting at the teraflops systems requested by the Grand Challenges. This implies that, as a lower level of the pyramid and in order to develop the skills and the applications of future innovative computer architectures, universities and research institutions as well as industrial research divisions should be provided with mid-sized supercomputer systems. This level is required for the demanding science and engineering problems that do not need the very maximum of computing capacity, and for the computer science and computational mathematics community in order to take care of the architectural, operating systems, tools, and algorithmic issues. A third and, according to the structure of the pyramid, much broader level of scientific computing has to be supported by powerful workstations as the effective workbenchs of scientists and engineers, in addition to the tremendous functionality of personal computers.

These facilities have to be networked campus-wide or corporate-wide with easy access to external communication services like Internet, which leads to the very basement of the pyramid - the network. High-speed communication with broadband functionality is promoted in the U.S. on a large scale for scientific as well as commercial applications and also in some european countries strong efforts are made to provide the scientific community with broadband communication services, e.g. in Great Britain with SuperJanet. Other european countries are either still quite far from having access to broadband communications or just start to establish testbeds with innovative technologies like ATM. In Germany, due to the high license costs involved, in the past many universities could not even manage to get interconnected to the German Science Network (WiN) with a maximum transmission rate of 2 mbps available since years. The uncomfortable tariff structure will again be a high barrier to benefit from the new B-WiN in Germany providing now 34 mbps and 155 mbps shortly. The backlash in high-speed communications has been a severe barrier to the nation-wide infrastructure capable to provide supercomputer power and functionality to the scientific community on a large scale with transfer opportunities into the industry [10]. This will hopefully change in the near future due to the deregulation efforts in Europe.

4 The Push from Vector-Supercomputers to Massively Parallel Systems

Partial differential equations have been dominating in the advancement of high-speed computers and in the exploitation of their potential: "numerical windtunnel". The solution methodology for such equations leads via discretization into linear algebra and its numerical concepts and algorithms. The response of computer architecture to these early challenges of PDEs have been the vectorcomputers optimizing vector-pipeline processing and creating the effective instruments of vectorization [11]. Already in 1982, however, Cray Research made the significant step into multiprocessor vector-architectures and, hence, into parallel processing.

On Cray multiprocessor systems, there have been implemented three different concepts which support parallelism on the programming language level: macrotasking, microtasking, and autotasking. The autotasking concept provides a user-friendly interface which can be used efficiently for fine-grain parallelism and removes some of the limitations which have restricted the usage of multitasking in the past [12]. Based on highly optimized library routines, more than 60% or even 80% of the peak performance can be obtained for large scientific applications [13-14].

While the efficiency of multitasking is proven for parallel programs running in dedicated environments since many years, the interest was focussing more and more on information about the effects introduced by parallel programs running in multiprogramming environments. With the development of the benchmark control system PARbench [15], which enables measurements of effects introduced by parallel programs running in a multiprogramming mode, ZAM was able to evaluate and to compare different multitasking implementations, different operating systems, and different computer hardware. Moreover, we have studied scheduling algorithms. It has been shown that efficient scheduling algorithms on the operating system level must cooperate with lower-level work distribution algorithms on the application level which have detailed information about program structure and parallelism requirements [16].

The exploration of the computing potential of the pipelining principle including programming and compiler techniques, tools, operating system functionality, and shared-memory organization and optimization resulted in an efficient arsenal of knowledge and experience about vectorcomputing. Certainly, vectorcomputers will further develop in functionality and performance towards the 100 gigaflops target by exploiting the architectural and technological potential and expanding the ”weak” parallelism well beyond the available number of processors. Even today, in the end, the sustained performance of these systems, e.g. CRAY T-90, NEC SX-4, or Fujitsu VPP500 and VPP300, turns out to be still ahead of the sustained performance of massively parallel systems for a vast majority of essential applications. Therefore, despite the relative progress of massively parallel computers, the very workhorses of Computational Science and Engineering are still vector-supercomputers [17]. More and more, however, the new powerful superscalar multicomputer systems share the work with them.

Workstations and - despite the crucial communication issue - workstation clusters, on the other hand, provide the excellent capacity to free the higher-class supercomputers from the increasing number of ”small” supercomputer applications by off-loading, thus reserving them for the really large applications of the Grand Challenge category which can justify the high expenditures. However, massively parallel computers are undoubtedly considered as the - only - remedy to the needs of the demanding applications in the Grand Challenge category and maybe yet unrecognized commercial applications.

Unfortunately, in the early 90s the manufacturers of massively parallel systems promised that they would be capable to develop and deliver parallel supercomputers in 1995 which easily be able to reach the magical ”3 T’s” (i.e. Teraflops in execution rate, Terabyte in main memory, and Terabyte/s interconnection bandwidth), thus indicating rather a revolutionary than an evolutionary step of almost three orders of magnitude beyond the then valid state-of-the-art supercomputer performance. In the meanwhile, there has not only happened a shake-out in the respective computer industry. Many investments into this massively parallel computing strategy may be definitely lost; the establishment of new hardware and software platforms will require new investments concerning finances and manpower as well as psychological recovery from the frustrations caused by the unfulfilled promises of those manufacturers.

In 1991, KFA entered the field of massively parallel computing by joining an external partnership with Intel SSD and acquiring, in the end, a 140 node Paragon XP/S10. This system is a distributed-memory scalable multicomputer [18] running OSF/1 as the default operating system. The processing nodes, which are based on the Intel i860 XP RISC processor, are connected by a two-dimensional mesh with relatively high bandwidth (measured: about 90 MByte per second; specification: 200 MByte per second), and the peak performance is about 10.8 GFLOPS. As the utilization of the system at KFA is quite impressive (far more than 70% nowadays), the users have accepted massively parallel computers as a tool for scientific problem solving.

A major objective in the cooperation with Intel has been the development and enhancement of functions to integrate this supercomputer into the production environment of KFA, and to establish parallel supercomputing as a standard offering for scientific computing. The hardware of the Paragon has reached satisfactory maturity; however, due to software instabilities, the usability and availability of the system was poor for quite a time causing major efforts on either side to achieve a reasonable production level targetting at one reboot per week. During last year, the stability situation was improved significantly by the additional installation of a 16-node Paragon system, generously granted by Intel, which is used for testing, debugging, and software development.

A key issue in massively parallel computing is scalability. Parallelizing ”dusty” decks from industry is certainly an important task to do in order to increase the acceptance of parallel computing in commercial environments. However, one cannot expect terrific performance gains in many of these programs just from porting such originally sequential, in many cases also organically grown, codes to parallel systems. There is a big discrepancy between the peak rates of massively parallel systems and the sustained performance [19]. With kernels, the state of the art of massively parallel computers delivers, together with a pretty large variance in the performance data depending on the definite architecture of the system and the algorithm as well, in the average around dissappointing 10 to 20% of the peak rate as sustained performance. Hence, also the price-to-performance ratio of massively parallel computers is loosing part of its attractivity, if compared with vectorcomputers and, in particular, with superscalar multicomputer systems.

5 Parallel Programming and Software Tools

So far, microprocessor chips have been developed with a different market goal in mind. It is extremely difficult to exploit the performance hidden in the hardware design of these processors via high-level programming languages and compiler techniques [20-21]; very often this leads to a loss by a factor of five to ten referred to peak performance of the node [22-23]. It cannot be accepted as a reasonable software-technological approach to switch back to the very old times of assembler programming to reach reasonable performance levels. Convergence of hardware and compiler design together with the development of valuable programming tools must become the future development strategy.

On parallel computer architectures, software support is still one of the major obstacles to open the usage of such systems to a broader range of applications. Experience has shown that user-friendly tools supporting, for instance, the performance analysis and debugging process are extremely helpful and can drastically shorten the time-to-solution for a given problem. At ZAM, we have developed the X Window based PARvis environment which - on a post-mortem basis - translates a given trace file generated on massively parallel systems like Intel Paragon or CRAY T3D into a variety of graphical system views which provide a reasonable basis for system understanding and program optimization [24]. The statistics features in combination with the time-line displays are the strength of the system. Based on the extremely flexible zooming and scrolling function in the time-line displays, analysis operations are supported which can significantly improve the understanding of observed performance problems. (PARvis is now distributed under the trademark VAMPIR by the german software house Pallas.)

Another worthwhile effort is the parallelization of existing sequential applications for distributed memory parallel systems, as parallelization for this type of machines is by its nature not a local operation. The tool suite TOP^2 developed at ZAM assists users of such parallel systems in porting existing sequential Fortran applications by supporting the separation of compute-intensive kernels out of the sequential code and providing a development environment for the parallelization of these code segments [25]. Thus, the parallelization of large applications can be broken up into several smaller tasks as local optimization steps.

The remaining sequential and the parallel code are run simultaneously as a distributed application on both systems and automatically exchange context data between both components. A main feature in this process is the provision of cross-domain message passing for the automatic distribution of program data from the sequential machine (Sun, CRAY) to the distributed memory system (Intel Paragon, CRAY T3D); another important faeture is on-line debugging of the parallelized code. The data distribution features of TOP^2 are a subset of those defined in HPF Fortran and, thus, especially support algorithms on regular data structures exploiting data parallelism in the context of SPMD programming. Despite the already available spectrum of valuable instruments, more powerful and effective software tools are needed and therefore have to be developed within the next few years; only they will be able to increase programmers productivity to the reasonable amount required in order to establish parallel computing as a well-accepted paradigm. More concretely, effective debugging tools, performance visualization environments, and multi-user operating systems have to be created to extend the efficient usage of massively parallel systems.

6 Programming Models: Message Passing versus Shared Virtual Memory

Another important issue is programming models. While Message Passing is widely and effectively used on distributed memory systems as the only efficiently implemented programming paradigm at present, one can hardly imagine that this programming model will carry all future efforts to introduce massively parallel computing as the unique future technology. Despite the failure of the first commercially available massively parallel computer system which supported the programming paradigm of the Shared Virtual Memory (SVM), the efforts on this programming model should continue, because from a user's point of view, but also, maybe, from the language point of view, this SVM paradigm [26-27] seems to carry enough potential to overcome fundamental deficiencies which can be experienced from the Message-Passing paradigm.

At ZAM, we have developed SVM-Fortran as a language extension of Fortran 77 for shared-memory parallel programming on distributed memory systems. It provides special language features for optimization of data locality and load balancing. SVM-Fortran is designed for shared virtual memory systems as well as for highly parallel computers with a hardware-based global address space. Currently, the language is translated by a source-to-source compiler into Fortran 77 with runtime library calls. Shared data structures are mapped to System V shared segments. This shared-memory interface is supported on the Intel Paragon, as the target architecture, by the Advanced Shared Virtual Memory system (ASVM) developed within a cooperative project at the Intel European Supercomputer Development Center (ESDC) in Munich [28]. Parallelization of real-world scientific applications requires the integration of the language, the SVM implementation, as well as of programming tools into a homogeneous programming environment [29]. Currently, the programming environment consists of the SVM-Fortran compiler, of OPAL, a source code based performance analysis tool, and the visualization tool PARvis (respectively, VAMPIR).

First performance results demonstrate that the shared-memory programming model can be efficiently used on future global address space multiprocessors via SVM-Fortran, since data locality can be enforced through well-designed work distribution annotations. However, much research and development work has to be completed to achieve production-oriented SVM implementations with tolerable overhead. Unfortunately, powerful tools, in order to monitor the progress in the parallelization process, and strong support on the hardware level cannot be seen to be commercially available in the near future.

7 Conclusions

Researchers in the field of innovative computing believe that there will be no single all-encompassing architecture which will be capable to satisfy the heterogeneous spectrum of requirements with equally optimal performance. On the other hand, it is well known that a user tends to invest tremendous efforts in order to extract even that small level of sustained performance out of an innovative computer system for his specific application, although his heterogeneous requirements cannot be efficiently satisfied by the single target system he is focussing on just because it is available to him: per aspera ad astra. Nevertheless, it is for sure now: parallel computing works! [30]

The experiences with the different supercomputer architectures with their strengths and weaknesses, the technological obstacles for major performance steps in vector-computing, the large variance in the performance data for algorithms on different parallel machines quite naturally lead to the concept of heterogeneous computer systems. Heterogeneous computing is an attractive concept because it takes into account that the individual parallel machines, and vectorcomputers as well, spend much of their time on tasks for which they are unsuited; these effects lead to the experienced breakdowns in sustained performance and also to scalability problems. On heterogeneous systems [31], the computational work can be split across different computers in order to achieve in total the fastest possible execution, where the individual portions of the work are sent to those computer systems which have been proved to be best, either scalar, vector, vector-parallel or massively parallel, for the specific characteristics of the work. The goal of heterogeneous computing is definitely that the efficiency of computation and, thereby, the efficiency and cost-effectiveness of both computers and programmers may benefit from the ever increasing value of diversity of innovative systems and solutions. Therefore, with the new supercomputer complex ZAM will try hard to respond to this new challenge.

References

1. Special Double Issue: Grand Challenges to Computational Science, Future Generation Computer Systems **5** (1989), No. 2&3.
2. Committee on Physical, Mathematical, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, Grand Challenges 1993: High Performance Computing and Communications, The FY 1993 U.S. Research and Development Program, Office of Science and Technology Policy, Washington, 1992.
3. Board on Mathematical Sciences of the National Research Council (USA), The David II Report: Renewing U.S. Mathematics - A Plan for the 1990s, in: Notices of the American Mathematical Society, May/June 1990, 542-546; September 1990, 813-837; October 1990, 984-1004.
4. Commission of the European Communities, Report of the EEC Working Group on High-Performance Computing (Chairman: C. Rubbia), February 1991.
5. U. Trottenberg et al., Situation und Erfordernisse des wissenschaftlichen Höchstleistungsrechnens in Deutschland - Memorandum zur Initiative High Performance Scientific Computing (HPSC), Februar 1992, published in: Informatik-Spektrum **15** (1992), H. 4, 218
6. The Congress of the United States, Congressional Budget Office, Promoting High-Performance Computing and Communications, Washington, June 1993.
7. High-Performance Computing Applications Requirements Group, High-Performance Computing and Networking, Report, European Union, April 1994. - High-Performance Networking Requirements Group, Report, European Union, April 1994.
8. Bundesministerium für Forschung und Technologie, Initiative zur Förderung des parallelen Höchstleistungsrechnens in Wissenschaft und Wirtschaft, BMFT, Bonn, Juni 1993.
9. NSF Blue Ribbon Panel on High Performance Computing, From Desktop To Teraflop: Exploiting the U.S. Lead in High Performance Computing, Report, NSF-CISE, 19 October 1993.
10. R. Bayer, Plädoyer für eine Nationale Informations-Infrastruktur, Informatik-Spektrum **17** (1994), H. 5, 302.
11. K. Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, 1993
12. F. Hossfeld, R. Knecht, and W. E. Nagel, Multitasking: Experience with Applications on a CRAY X-MP, Parallel Computing 12 (1989), No. 3, 259.
13. S. Knecht, E. Laermann, and W.E. Nagel, Parallelizing QCD with Dynamical Fermions on a CRAY Multiprocessor System, Parallel Computing 15 (1990), No. 1-3, 3.
14. J. Frank and S. Knecht, Lattice Quantum Hadrodynamics on a CRAY Y-MP, The Journal of Supercomputing 6 (1992), 195.
15. W. E. Nagel and M.A. Linn, Benchmarking Parallel Programs in a Multiprogramming Environment: The PAR-Bench System, In: Advances in Parallel Computing, Vol. 8: Computer Benchmarks (Dongarra, J., Gentzsch, W., eds.), Elsevier Science Publishers B. V. (1993), 303.
16. W. E. Nagel, Ein verteiltes Scheduler-System für Mehrprozessorrechner mit gemeinsamem Speicher: Untersuchungen zur Ablaufplanung von parallelen Programmen, Berichte des Forschungszentrums Jülich Jül-2850 (1993).
17. J. J. Dongarra, H.-W. Meuer and E. Strohmaier (eds.), TOP500 Report 1993; TOP500 Report 1994, University of Mannheim, Germany, May/November 1994.

18. R. Berrendorf, H. C. Burg, U. Detert, R. Esser, M. Gerndt, and R. Knecht, Intel Paragon XP/S - Architecture, Software Environment, and Performance, Interner Bericht KFA-ZAM-IB- 9409, (1994).
19. D. H. Bailey et al., NAS Parallel Benchmark Results 3-94; NAS Parallel Benchmarks Update 9-94, RNR Technical Report RNR-94-006, NASA Ames Research Center, Moffett Field, California, March and September 1994.
20. D. F. Bacon et al., Compiler Transformations for High-Performance Computing, ACM Computing Surveys **26**(1994), No. 4, 345-420.
21. H. Zima and B. Chapman, Supercompilers for Parallel and Vector Computers, ACM Press, 1990.
22. A. Mlynski-Wiese, Leistungsuntersuchung des iPSC/860 RISC-Knoten-Prozessors: Architekturanalyse und Programmoptimierung, Zentralinstitut für Angewandte Mathematik, Berichte des Forschungszentrums Jülich Jül-2766, Mai 1993.
23. A. Krumme, Performance-Analyse des DEC Alpha-Chip: Moderne Compiler-Techniken zur Programmoptimierung, Zentralinstitut für Angewandte Mathematik, Berichte des Forschungszentrums Jülich Jül-2912, Mai 1994.
24. W. E. Nagel and A. Arnold, Performance Visualization of Parallel Programs: The PARvis Environment, Proceedings 1994 Intel Supercomputer Users Group (ISUG) Conference, 24-31; Technical Report CCSF-47, Concurrent Supercomputing Consortium, Pasadena, California, May 1994.
25. U. Detert and M. Gerndt, TOP2 - Tool Suite for the Development and Testing of Parallel Applications, Proceedings CONPAR'94 (1994), .
26. K. Li, Shared Virtual Memory on Loosely Coupled Multiprocessors, PhD Thesis, Yale University, September 1986.
27. R. Berrendorf, M. Gerndt, Z. Lahjomri and Th. Priol, A Comparison of Shared Virtual Memory and Message Passing Programming Techniques Based on a Finite Element Application, in: B. Buchberger and J. Volkert (eds.), Parallel Processing: CONPAR 94 - VAPPVI, Lecture Notes in Computer Science Vol. 854, Springer Verlag, 461-472.
28. Mairandres, M., Virtuell gemeinsamer Speicher mit integrierter Laufzeitbeobachtung, PhD Thesis (Dissertation), Technical University of Aachen (RWTH Aachen), February 1996.
29. M. Gerndt and R. Berrendorf, Parallelizing Applications with SVM-Fortran, in: Bob Hertzberger, G. Serazzi (eds.), Proceedings HPCN'95, Lecture Notes in Computer Science, Springer Verlag, 1995, Vol. 919, pp. 793-798.
30. Fox, G. C., Williams, R. D., and Messina, P. C., Parallel Computing Works!, Morgan Kaufman Publ., San Francisco, 1994
31. A. A. Khokhar et al., Heterogeneous Computing: Challenges and Opportunities, IEEE Computer **26** (1993), No. 6, 18.