

Boosting the Performance of Electromagnetic Simulations on a PC-Cluster

Carsten Trinitis, Martin Schulz, and Wolfgang Karl
Technische Universität München (TUM), Institut für Informatik
Lehrstuhl für Rechnertechnik und Rechnerorganisation (LRR)
Arcisstr. 21, D-80333 München
Tel: +49-89-289-{25771,28399,28278}, Fax: +49-89-289-282
E-Mail: {Carsten.Trinitis, Martin.Schulz, Wolfgang.Karl}@in.tum.de

Abstract

One of the crucial aspects in the design process of high voltage apparatus is the precise simulation of the electrostatic and/or electromagnetic field distribution in three dimensional domains. Since these simulations are rather computing- and communication-intensive, the solvers for both electrostatic and electromagnetic computations need to be parallelized. For this kind of code, clusters of commodity PCs, with or without high-speed interconnection technologies, are becoming increasingly important as target platforms, due to their excellent cost-performance ratio.

This paper summarizes the results that were obtained on such a cluster platform installed at ABB Corporate Research using POLOPT, a state-of-the-art parallel simulation environment for both electrostatic and electromagnetic problems. The experiments were conducted on a LINUX cluster using both Ethernet- and SCI-based interconnection technologies to allow a study of the impact of high-speed networking technology. For both electrostatic and electromagnetic field simulations in practical High Voltage Engineering, a high efficiency was obtained in this commodity environment. In addition, the results have shown that some codes, in this case the electromagnetic solver, are highly sensitive to the communication performance and hence can significantly benefit from the presence of fast interconnection mechanisms, as given with SCI.

Keywords: Electric field simulation, Commodity clusters, Parallel Efficiency, High voltage engineering, SCI

1 Motivation

One of the most important tasks in electrical engineering, especially when designing high voltage apparatus such as transformers, switchgear, or insulators, is the accurate simulation of the electrostatic and/or electromagnetic field

distribution in areas of interest. A well established method for carrying out such simulations is the Boundary-Element method (BEM) [2]. The precise knowledge of the electrostatic field distribution helps the electrical engineer to prevent possible flashovers in critical areas of the device being designed. The geometrical shape of the electrodes has a significant influence on the resulting electrostatic field. Eddy currents, a well known problem in the design of transformers, can lead to power losses and heating.

For solving the numerical equation systems resulting from the electrical boundary conditions, iterative solution techniques such as GMRES [15] must be employed. Due to their high computational demands, parallel computing is required in order to reach realistic workloads. As far as the electrostatic problem formulation is concerned, first experiences in this direction have been made using a PVM-based version of the three dimensional electric field simulation program POLOPT [1] on a heterogeneous cluster of UNIX workstations [4]. The PVM based electrostatic solver has then been ported to a cluster of LINUX based PCs [5].

In order to be able to perform simulations for electromagnetic problems, a preconditioner for the eddy-current system of second kind Boundary Integral Equations has been developed by the Seminar for Applied Mathematics at the ETH Zürich, Switzerland [7]. This MPI-based electromagnetic POLOPT solver [7], [17], [18] has also been implemented on a LINUX-based PC cluster. Since the algorithm is rather communication intensive, a high bandwidth and low latency network like SCI [11, 9] is a crucial prerequisite for an efficient solution.

The remainder of this paper is structured as follows. Section 2 introduces the main steps necessary in the simulation process, followed by an in-depth discussion of the actual calculation process in Section 3. The deployed cluster technology along with its software components is discussed in Section 4 and experimental results are presented in Section 5. Finally, the paper is rounded up with some concluding remarks in Section 6.

2 The Simulation Process

In High Voltage Engineering the electrostatic and electromagnetic field distribution is a very important aspect for the design of transformers, switchgear, insulators etc.. In order to determine possible spots for flashovers or losses caused by eddy currents in transformers, comprehensive simulation calculations are necessary. The simulation software POLOPT offers modules for the required calculations based on the Boundary Element Method (BEM).

Roughly speaking, the simulation process consists of the input of geometric data (usually done with a CAD modeling program), the creation of an accompanying mesh, the generation of the coefficient matrix, the solution of the linear system, the potential and field calculation in points of interest, and the visualization of the results.

The first and the second part of the simulation process are usually performed with a CAD package like ProEngineer [14], [3], SDRC I-DEAS [16], or similar software.

In order to reduce computation times, it is necessary to parallelize the solver (steps 3 through 5 in the simulation process): In step 3, a coefficient matrix is generated by POLOPT which can be done in parallel since the generation of a matrix line is independent from all other lines. Depending on the problem type (electrostatic or electromagnetic), the resulting equation system is solved or the parallel preconditioner and solver described in [7] are applied. The size of the equation systems are usually in the range of 3–5 orders of magnitude with densely populated coefficient matrices.

Due to these large problem sizes and their corresponding computational requirements, both the electrostatic and the electromagnetic solver have been parallelized on a LINUX based PC cluster installed at ABB Corporate Research, Heidelberg/Ladenburg, Germany.

3 The Field Calculation Process

The numerical analysis of three dimensional electric fields has become a very important part of design practice in the electro-technical industry. One numerical method that is often used for such problems is the Boundary Element Method [2] which is highly efficient and allows to solve complex three dimensional models with satisfactory accuracy. In contrast to a two dimensional analysis, the calculation of realistic three dimensional models is very CPU intensive. From a mathematical point of view, a field calculation problem can be stated as follows: The goal is to calculate the values of the potential $\Phi(x, y, z)$ and the electric field strength with

$$\vec{E}(x, y, z) = -\nabla\Phi \quad (1)$$

In order to achieve this, one has to solve Laplace's differential equation

$$\Delta\Phi = 0 \quad (2)$$

For electrostatic fields this differential equation can be solved by

$$\Phi(\vec{r}_p) = \frac{1}{4\pi\epsilon} \iiint \frac{\rho(\vec{r}_q)}{|\vec{r}_p - \vec{r}_q|} dV \quad (3)$$

with \vec{r}_p being the radius vector in the point of interest and \vec{r}_q being the radius vector in the integration point. In reality electric fields are caused by applying voltage to electrode systems. The charge that causes the electric field is distributed over the electrode surfaces as an area charge with density ρ_A . For practical field problems the integration in (3) cannot be done analytically.

The basic principle of all charge simulation methods is to approximate this real charge density distribution by simulation charges with charge densities that can easily be formulated. This approach makes it possible to calculate the simulation charges' potential functions analytically or numerically. Admissible simulation charges are point-, line-, area- or space charges. The potential that is generated by such a system of simulation charges can be calculated from the sum of each charge type's contribution:

$$\Phi(\vec{r}_p) = \frac{1}{4\pi\epsilon} \left[\sum_{i=1}^{n_p} \frac{Q_i}{|\vec{r}_p - \vec{r}_q|} + \sum_{i=1}^{n_l} \int \frac{\rho_{L_i}(\vec{r}_q)}{|\vec{r}_p - \vec{r}_q|} dL \right. \\ \left. + \sum_{i=1}^{n_a} \iint \frac{\rho_{A_i}(\vec{r}_q)}{|\vec{r}_p - \vec{r}_q|} dA + \sum_{i=1}^{n_v} \iiint \frac{\rho_{V_i}(\vec{r}_q)}{|\vec{r}_p - \vec{r}_q|} dV \right] \quad (4)$$

with n_p being the number of point charges, n_l being the number of line charges, n_a being the number of area charges, and n_v being the number of space charges, respectively. Each simulation source is defined by its geometry and charge density which makes it possible to calculate the integrals in (4). For a total of n_g simulation charges we get

$$\Phi(\vec{r}_p) = \sum_{i=1}^{n_g} p_i \cdot \rho_i \quad (5)$$

with p_i being the potential coefficients (which are independent on the geometry) and ρ_i being the charge density coefficients. Provided that the charge density coefficients/charge values are known, we can calculate the potential in arbitrary points. If we want to calculate the charge values in the contour points with known boundary conditions (potential) we have to set up an equation of the form (5) for each contour point. Thus we obtain a linear equation system of the form

$$\mathbf{P} \cdot \mathbf{q} = \Phi_{\mathbf{KP}} \quad (6)$$

with \mathbf{P} being the potential coefficient matrix, \mathbf{q} being the vector of unknown charges and Φ being the vector of potentials in the contour points. When solving the equation system (6) for the unknown vector \mathbf{q} one can easily obtain the potential in arbitrary points using (5) and the electric field strength by differentiation of (5).

We can summarize that the process of numerical field calculation is composed of the three stages calculation of the coefficient matrix \mathbf{P} , solution of the linear equation system (6), and calculation of field and potential at points of interest.

Such formulations of Laplace's equation are well suited for parallel computations [6]. Due to two important features, a simple but efficient parallelization strategy can be applied: Firstly, all elements of the coefficient matrix \mathbf{P} as well as the field values in arbitrary points can be calculated independently from each other. Secondly, the solution of the linear equation system (6) can be found by means of iterative methods like the Generalized Minimal Residuals Method [15].

3.1 Boundary Element Method

In the Boundary Element Method integral equations are reduced to a system of algebraic equations by discretizing the model's surfaces with well selected, small curvilinear patches (boundary elements) on the interfaces between media with different material characteristics. The characteristics of the media are taken as constants for each relevant region in a model. Over a boundary element, the field is expressed as an analytical interpolation function between the field values at the nodes (element vertices).

The basis for establishing a system of integral equations is the introduction of equivalent electric surface charge densities, which are either distributed over boundary surfaces between dielectrics, taking into account the bound surface charge, over conductor-dielectric boundary surfaces, where the conducting body is at a known potential, or over conductor-dielectric boundary surfaces, where the total quantity of electric charge on the conductor is known and may be zero.

The distribution of these equivalent surface charges over the model's boundary surfaces is fixed by the boundary conditions on the electric field \mathbf{E} and the electric displacement \mathbf{D} between two dielectrics, by prescribed values for the potential function on the conducting surfaces at fixed potentials, and by prescribed values for the net surface charges on the conductors at unknown potentials ("floating potential").

A proper numerical solution of these problems calls for an adequate approximation of both the geometry and the charge distribution. For the geometry approximation POLOPT uses triangular and quadrilateral curvilinear

boundary elements, allowing suitable discretization of curved boundaries with a relatively small number of elements. The surface charges are approximated by a piecewise linear function. The electric field in the region of interest can be considered as the result of the presence of the equivalent surface charges. These charges are the "sources" of the electric field [1].

3.2 Parallelization Concept

The parallelization concept is illustrated in Fig. 1. The algorithm is based on a *master-slave* approach. The input data is replicated on each node so that arbitrary parts of the coefficient matrix can be generated on each node. As men-

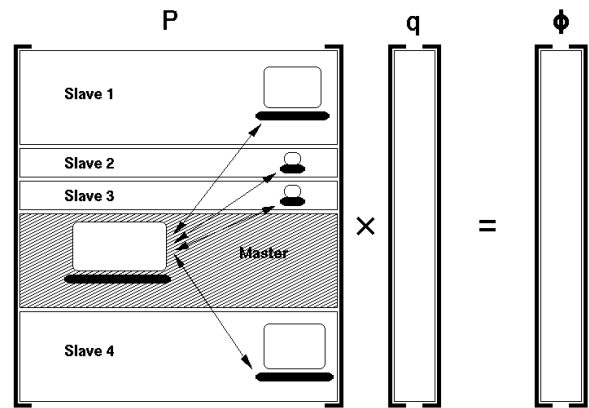


Figure 1. Parallelization concept.

tioned in Section 2, the field simulation process consists of modeling the geometric data, generating an accompanying mesh, computing the (fully populated) coefficient matrix, solving the resulting equation system and calculating field and potential in the points of interest. The part that can be parallelized is the actual numerical calculation, i.e. the latter three steps. Each matrix row can be generated independently from the other rows if the input data has been replicated on each node. The generated parts of the matrix are distributed over the nodes.

The workload is dynamically distributed by the master taking into account each node's speed and current load. This enables a straightforward form of dynamic load balancing leading to optimized performance. The solver being used is the iterative GMRES method [15]. This solver can be parallelized in a straightforward manner because the operation that is performed during each iteration is basically a matrix-vector multiplication. Since the basic parallelization concept is rather algebraic than topological (no domain decomposition), the parallel efficiency depends only on the problem size.

4 Cluster Technology as Suitable Target Architecture

Due to their excellent price/performance ratio driven by the always state-of-the-art single node performance of current PCs, clusters built from commodity components offer a very suitable platform for a large range of high performance codes, including the simulations targeted within this work.

4.1 Clustering with SCI

A large group of applications, however, suffers from the low communication performance of commodity clusters, as they rely on general networking hardware, like Ethernet, and heavy protocol stacks, like TCP/IP or UDP/IP. To compensate for this deficit so-called System Area Networks (SANs) have been developed which both offer, in coordination with appropriate software environments [12], high bandwidth through novel interconnection hardware and low-latency by relying on user-level communication schemes. The latter technique enables application to directly talk to the networking hardware avoiding any operating system, protocol, or driver overhead.

One well-known example for this kind of technology is the Scalable Coherent Interface (SCI) [9, 11]. It is based on a global physical address space allowing each node to directly address any physical memory location within the system using standard user-level read and write operations¹. This leads to one-way latencies of under 2 μ s and a bandwidth of up to 320 MB/s (when connected via PCI busses with 64bit and 66MHz).

4.2 Efficient Messaging with SCI and MPI

Most codes in the high performance computing domain are implemented using an explicit message passing paradigm. In recent years, the Message Passing Interface (MPI) [13] has established itself in this domain as the de-facto standard. In order to be able to harvest the communication advantages of SCI in an MPI environment, several software efforts have been undertaken to implement optimized MPI versions for SCI [19, 10]. Using these environments, one-way latencies of under 5 μ s and a bandwidth of over 95% of the peak bandwidth of the SCI setup can be achieved.

4.3 Support for Legacy Code

While most modern codes, including the electromagnetic section of POLOPT, are based on MPI, legacy codes exist,

¹More on the principle of SCI can be found in [9].

which are parallelized with different parallel libraries. Especially codes based on the messaging layer PVM [8] are still commonplace in many areas. The electrostatic part of POLOPT falls also into this category.

In order to exploit SCI also for these legacy codes, two different approaches are possible: implementing an SCI optimized version of PVM (similar to the MPI libraries discussed above) or by a wrapper which maps the PVM calls to their MPI counterparts. Both approaches have been followed within this project for the deployment of clusters for POLOPT. The SCI-PVM implementation, which is described in [5], however, was burdened by a high complexity of the integration of SCI into the existing PVM implementation. In addition, many features of PVM, including the support for heterogeneous architectures and dynamic task environments, are not important in HPC cluster environments and cause unnecessary runtime overhead even after a port to SCI.

The second solution, a PVM-MPI wrapper, on the other hand, does not infer these deficits as PVM calls are directly map on their MPI counterparts omitting unnecessary features. This also includes the dynamic task model of PVM, which is rarely used to its full extend in typical HPC applications, as they mostly rely either on SPMD-style or master/slave communication with a fixed number of tasks. The communication routines themselves can then directly be mapped to the corresponding MPI message routines without any conversion or translation overhead. This leads to the intended emulation of the PVM environment in transparency for the target code.

Following this approach, also the PVM-based parts of POLOPT have been converted into an MPI application without major efforts and resulting in less runtime overhead in the communication layer. In addition, this approach leads to a unification of the runtime environments, as both codes now internally run on MPI and its specific environment. This greatly simplifies the use of the code and reduced system administration costs.

4.4 Production Setup at ABB

For the experiments in the remainder of this paper a production cluster installed at ABB Corporate Research in Heidelberg has been used. It consists of eight single-CPU Pentium-III class PCs, each equipped with 1 GB of physical memory. The nodes are connected by both Fast Ethernet and SCI, where the latter one is set up as a 2-dimensional torus in a 4x2 configuration as seen in Figure 2. The MPI implementation used is either MP-MPICH 1.3 for Ethernet communication and ScaMPI [10] for SCI. The latter provides half round-trip latencies of a zero length MPI messages of less than 5 μ s and a sustained bandwidth of over 80MB/s (using the 32bit 33MHz busses).

5 Examples and Results

Two practical models have been calculated for the measurements described in this work. The first model, an electrostatic problem, is depicted in Figure 3. It is an SF_6 gas insulated loadbreak as being used for power supply systems by ABB Norway. The simulation is carried out in order to determine possible hot spots for the electric field which can lead to flashovers in prototypes. The equation system has a dimension of about 15000 unknowns with a fully populated matrix.

The second example, which was introduced in more detail in [17], is a transformer model as manufactured by ABB, as depicted in Figure 4. For this model, the task is to determine the distribution of losses caused by eddy currents in the yoke clamping plates in order to detect possible temperature hot spots. The yoke clamps are modeled as solid parts. The LV winding (inner winding) is modeled as a spiral winding, whereas the HV winding (outer winding) is modeled as a cylinder. The model consisting of the magnetic core, the yoke clamps, the flux plate and the HV windings has been investigated by calculating the eddy currents in the yoke clamp. The problem (1300 unknowns, complex double) has been analyzed with respect to calculation times and parallel efficiency.

The simulation times (run on the platform introduced above) for the electrostatic model can be seen in Table 1. They show an excellent scalability of the code with a parallel efficiency of 89% on 8 nodes for the overall computation time. In the three computational phases even a slight superlinear speedup was achieved due to the increased aggregated memory bandwidth. However, the use of SCI in this case only provided a marginal benefit, as the code belongs to the class of codes that is not very communication intensive and hence can not fully exploit the increased performance in the underlying network.

The computations for the electromagnetic problems, of which the results are shown in Table 2, had to be calculated on a minimum number of 2 nodes. This is due to the fact that the coefficient matrix had to fit into main memory. Therefore, the parallel efficiency has been measured relative to the computation times for two nodes.

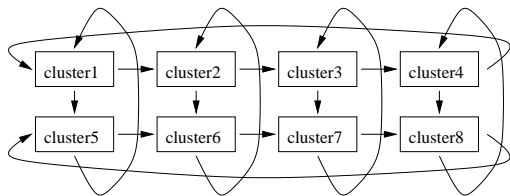


Figure 2. Torus configuration of the ABB SCI cluster.

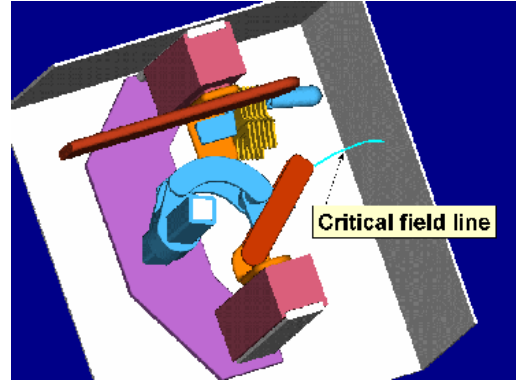


Figure 3. Loadbreak Model

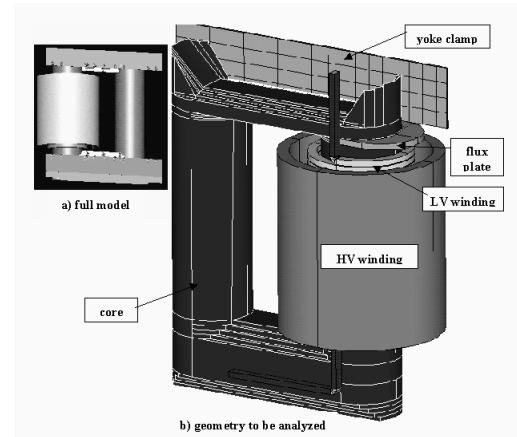


Figure 4. Transformer Model

| Network | Seq. | FE. | FE. | SCI | SCI |
|-----------|------|------|------|------|------|
| Nodes | 1 | 4 | 8 | 4 | 8 |
| Total | 7659 | 1991 | 1075 | 1996 | 1064 |
| Init | 140 | 143 | 140 | 140 | 140 |
| Generate | 3084 | 785 | 395 | 776 | 390 |
| Solve | 1139 | 279 | 148 | 273 | 144 |
| Calculate | 3269 | 784 | 392 | 777 | 390 |

Table 1. Computation times in seconds for full loadbreak problem.

| Network | FE. | FE. | FE. | SCI | SCI | SCI |
|-----------|-----|-----|-----|-----|-----|-----|
| Nodes | 2 | 4 | 8 | 2 | 4 | 8 |
| Init | 428 | 423 | 425 | 433 | 420 | 421 |
| Generate | 349 | 175 | 87 | 348 | 173 | 87 |
| Solve | 387 | 304 | 241 | 365 | 185 | 96 |
| Calculate | 523 | 290 | 127 | 523 | 290 | 127 |

Table 2. Computation times in seconds for transformer problem.

The part of the simulation with the most communication between the nodes is the solver phase. As can be seen from the computation times for solving the equation system, for Fast Ethernet a parallel efficiency of 64% (4 nodes) to 40% (8 nodes) compared to 98.7% (4 nodes) to 95% (8 nodes) on SCI can be obtained for the transformer model. These results show that the use of SCI in this case yields a significant improvement in parallel efficiency for the transformer model as the electromagnetic simulation is communication intensive and hence relies on a fast interconnection medium in order to achieve optimal scalability.

6 Conclusions

High performance cluster computing is a suitable way to provide the required computational power needed for realistic simulations in the area of electrical and electromagnetic simulation. This has been demonstrated using ABB's POLOPT simulation environment, which has been successfully deployed on a Linux cluster. In addition, SCI has been integrated as an interconnection technology for both PVM and MPI, in order to overcome the bottleneck of conventional interconnects like Ethernet.

PVM for SCI has been implemented by developing a wrapper that directly maps arbitrary PVM calls into the matching MPI code. For electrostatic problems, this allowed a clean integration of the PVM code base into the existing MPI production environment. Performance measurements with the electrostatic module have shown that the parallel efficiency is very good for both Fast Ethernet and SCI based networks. For electromagnetic problems, however, the use of the SCI network resulted in a significant performance gain for the solver which is due to the fact that the algorithm for solving this equation system is rather communication intensive.

References

- [1] Z. Andjelic. *POLOPT 4.5 User's Guide*. Asea Brown Boveri Corporate Research, Heidelberg, 1996.
- [2] R. Bausinger and G. Kuhn. *Die Boundary-Element Methode*. Expert Verlag, Ehingen, 1987.
- [3] A. Blaszczyk. Modelling with Pro/Engineer for POLOPT. Internal document, Asea Brown Boveri Corporate Research, Heidelberg, 1997.
- [4] A. Blaszczyk and C. Trinitis. *Experience with PVM in an Industrial Environment*. Lecture notes in Computer Science 1156, EuroPVM'96, Springer Verlag, pp. 174-179, 1996.
- [5] M. Eberl, W. Karl, C. Trinitis, and A. Blaszczyk. Parallel Computing on PC Clusters — An Alternative to Supercomputers for Industrial Applications. In *Proceedings of the 6th European PVM/MPI Users' Group Meeting, Barcelona, Spain*, volume 1697 of *LNCS*, pages 493-498. Springer Verlag, Berlin, Sept. 1999.
- [6] A. B. et al. Parallel computation of electric field in a heterogeneous workstation cluster. 1995.
- [7] G. Schmidlin, Ch. Schwab et al. *Preconditioning second kind Boundary Integral Equations for 3-D Eddy Current Problems*. ETH Zurich, Switzerland, 2000.
- [8] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine — A User's Guide and Tutorial for Networked Parallel Computing*. Scientific and Engineering Computation. MIT Press, 1994.
- [9] H. Hellwagner and A. Reinefeld, editors. *SCI: Scalable Coherent Interface. Architecture and Software for High-Performance Compute Clusters*, volume 1734 of *LNCS State-of-the-Art Survey*. Springer Verlag, Oct. 1999. ISBN 3-540-66696-6.
- [10] L. Huse, K. Omang, H. Bugge, H. Ry, A. Haugsdal, and E. Rustad. *ScaMPI — Design and Implementation*, chapter 14, pages 249-261. Volume 1734 of Hellwagner and Reinefeld [9], Oct. 1999. ISBN 3-540-66696-6.
- [11] IEEE Computer Society. *IEEE Std 1596-1992: IEEE Standard for Scalable Coherent Interface*. The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, USA, August 1993.
- [12] W. Karl, M. Leberecht, and M. Schulz. Supporting Shared Memory and Message Passing on Clusters of PCs with a SMiLE. In A. Sivasubramaniam and M. Lauria, editors, *Proceedings of Workshop on Communication and Architectural Support for Network based Parallel Computing (CANPC) (held in conjunction with HPCA)*, volume 1602 of *Lecture Notes in Computer Science (LNCS)*, pages 196-210, Berlin, 1999. Springer Verlag.
- [13] Message Passing Interface Forum (MPIF). MPI: A Message-Passing Interface Standard. Technical Report, University of Tennessee, Knoxville, June 1995. <http://www.mpi-forum.org/>.
- [14] Web page, Parametric Technology Corporation [Homepage], <http://www.ptc.com/>.
- [15] Y. Saad and M. Schultz. *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*. SIAM J.Sci. Stat. Comput., pp. 856-869, 1989.
- [16] Web page, SDRC [Homepage], <http://www.sdrc.com/>.
- [17] C. Trinitis, M. Eberl, and W. Karl. Numerical Calculation of Electromagnetic Problems on an SCI Based PC-Cluster. In *Proceedings of International Conference on Parallel Computing in Electrical Engineering*, pages 166-170, Trois Rivieres, Quebec, Canada, August 2000. International Conference on Parallel Computing in Electrical Engineering.
- [18] C. Trinitis, M. Schulz, M. Eberl, and W. K. arl. SCI-based LINUX PC-Clusters as a Platform for Electromagnetic Field Calculations. In *Proceedings of the Sixth International Conference on Parallel Computing Technologies (PaCT-2001)*, pages 510-513. Springer Verlag, 2001.
- [19] J. Worrigen and T. Bemmerl. MPICH for SCI-Connected Clusters. In G. Horn and W. Karl, editors, *Proceedings of SCI-Europe '99, The 2nd international conference on SCI-based technology and research*, pages 3-11. SINTEF Electronics and Cybernetics, Sept. 1999. ISBN: 82-14-00014-9, Also available at <http://www.bode.in.tum.de/events/>.