1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

1 9 9 3

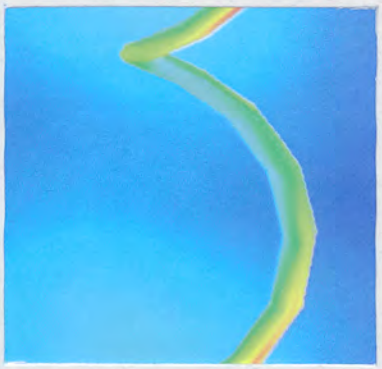
1 9 9 3

1 9 9 3

Centro Svizzero di Calcolo Scientifico

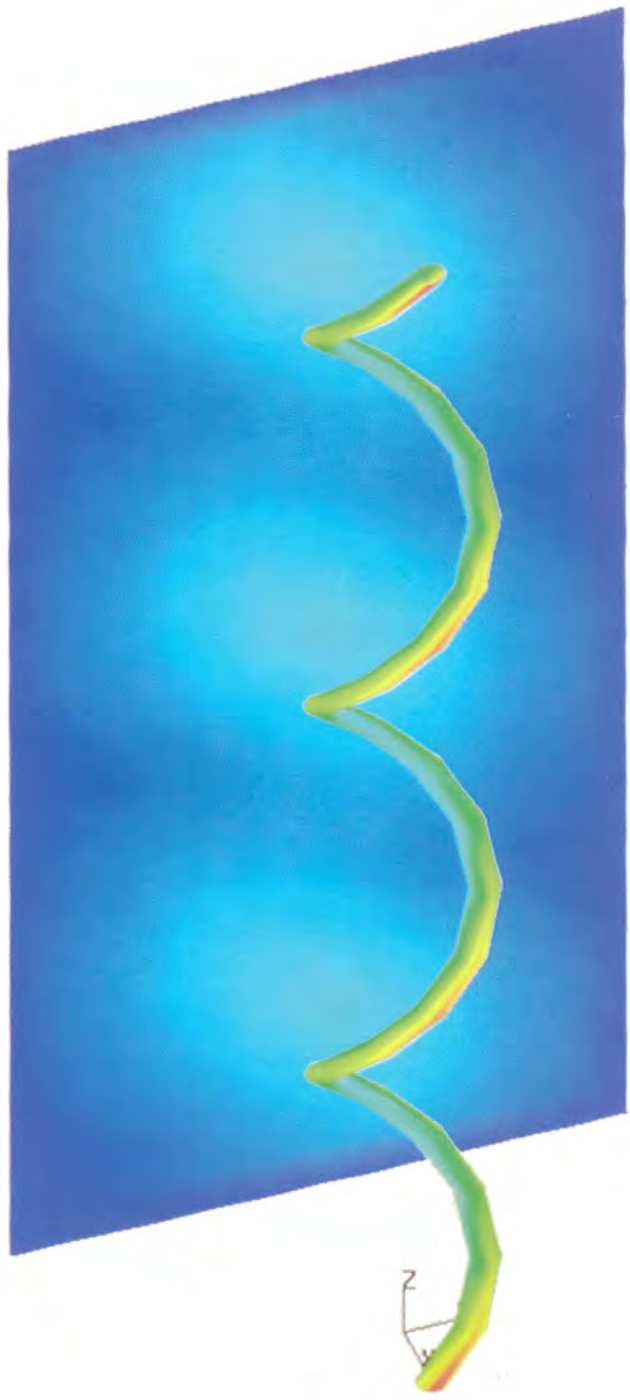
CSCS

Swiss Scientific Computing Center



ETH Eidgenössische Technische Hochshule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich



Distribution of the current on the grounded plate, from the project "Computation of electric field and space charge" (p 36)

Published 1994.

Centro Svizzero di Calcolo Scientifico (CSCS), also known as the Swiss Scientific Computing Center, Via Cantonale, 6928 Manno, Switzerland.
CSCS is affiliated with the Eidgenössische Technische Hochschule Zürich (ETHZ), also known as the Swiss Federal Institute of Technology Zurich.

Some Photographs by Franco Mattei 6972 Claro, Switzerland.

Printed by Fratelli Roda SA, Industria grafica e cartotecnica 6807 Taverne, Switzerland.

TABLE OF CONTENTS

PREFACE 7

CSCS OVERVIEW 9

FINANCIAL AND PERSONNEL DEVELOPMENT 12

COMPUTING FACILITIES 14

EDUCATION 18

INDUSTRIAL PARTNERSHIP PROGRAM 20

CSCS ADMINISTRATION 21

CSCS COMMITTEE AND COUNCIL 22

PROJECTS

Project Abstracts 27

Projects 40

PERSONNEL 62

VISITORS 64

COLLOQUIA 66

COURSES 69

PUBLICATIONS

Technical Reports 70

Articles 70

Conference Proceedings 71

ADDRESSES 72

SCIENTIFIC AND ECONOMIC SUCCESS INCREASINGLY DEPENDS ON FINDING SOLUTIONS USING HIGH–PERFORMANCE COMPUTING.

HIGH–PERFORMANCE COMPUTING IS AN INVESTMENT FOR THE FUTURE.

PREFACE

High–performance computing has a tremendous potential to solve crucial issues in science, technology and economy. The use of the high–performance computing (HPC) methodologies in research and development is rapidly proliferating in conjunction with traditional theory and classical experimentation.

New studies, results and success stories employing HPC simulation confirm that the use of this methodology is expanding from the scientific field and embracing problem–solving in industrial research and development. The competitive position of more and more companies demand the utilization of modern and powerful tools. HPC is becoming one of the decisive factors for success.

With the growing importance of HPC, the dependency of its users on concentrated state–of–the–art knowledge and high–end compute resources is equally augmented. HPC centers are crucial in bridging the forefront of HPC know–how and resources with the users in academia and industry. It is the task of these powerhouses to guide and to support the efforts of computational scientists in using HPC for their problem solving.

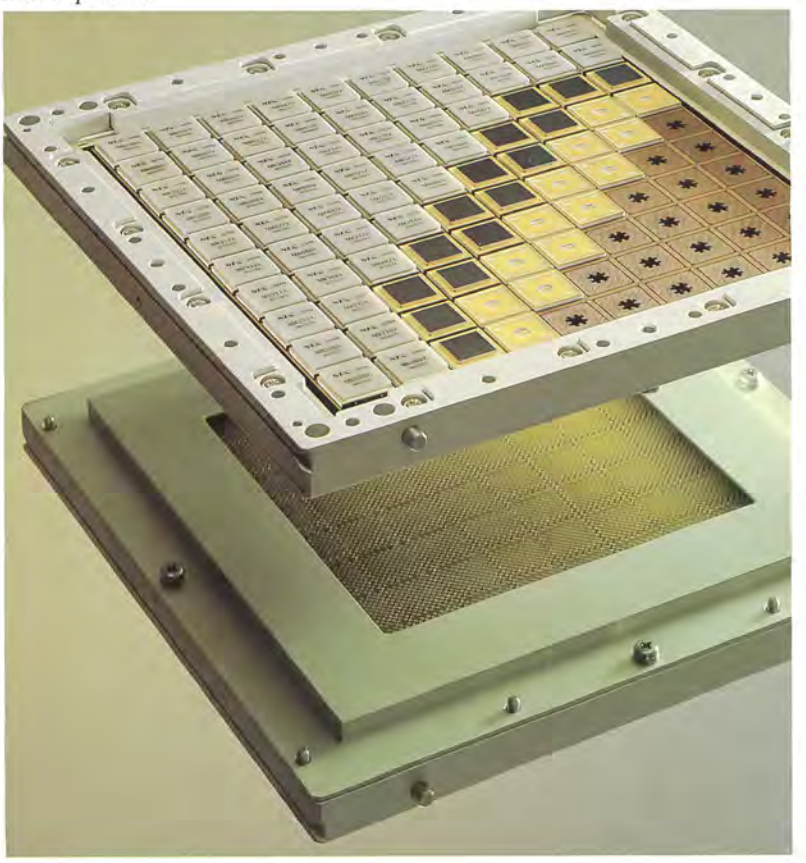
With CSCS, Switzerland is well equipped to successfully move into the new HPC age. CSCS offers a high degree of expertise in the area of specialized scientific user support. In addition to the wide range of commercial software packages available, new software is constantly developed to address needs of users. CSCS is clearly evolving into a first–class software and knowledge center, capable of exploiting the rich hardware environment to meet the application needs of users. In the rapidly changing world of HPC, CSCS is dedicated to an application–oriented and integrated computing environment.

Competent usage of HPC resources contributes to scientific and economic success. HPC is an investment for the future.

Prof. Dr. Ralf Hütter
Vice–president Research ETH Zürich

Dr. Alfred Scheidegger
Director CSCS, Manno

NEC chip–board



CSCS OVERVIEW

The Centro Svizzero di Calcolo Scientifico (CSCS) is the national scientific computing center of Switzerland. The center is the result of an extended and concerted effort by the Swiss government to promote high–performance computing (HPC). Although CSCS was officially inaugurated on October 1, 1992, the center began serving users in the third quarter of 1991 and has been in full operation since.

The primary responsibility of CSCS is to provide Switzerland with the HPC resources it requires and the expertise to exploit these resources. CSCS is dedicated to maintaining a sophisticated, leading–edge resource environment and anticipating user needs. These resources rank among the most powerful available in Europe. CSCS also strives to increase public awareness and understanding of the societal benefits to be gained through advanced computing.

CSCS is making a name for itself in the international community through its achievements and the accomplishments of its users.

ACHIEVEMENTS IN 1993

1993 was a year of significant growth for CSCS: improvements in services, expansions and upgrades in computational resources, increased staffing by a factor of two, initiation of an industrial partnership program, addition of a new research and development section, and extended educational training programs just to mention a few.

Important milestones were achieved in 1993 with regard to these resources. Based on feedback from users, CSCS has introduced a front–end environment to supplement the NEC SX–3. This environment, composed of a Convex C3820, file handling facilities and UNITREE storage/archiving, offers the user an interactive workspace thereby reducing the load of non-computational work on the SX–3. Another enhancement to the computational resources was the upgrade of the SX–3/22 to a SX–3/24R, increasing the computation power available to users. A Convex Meta Series workstation cluster has been installed to download programs from the NEC SX–3.

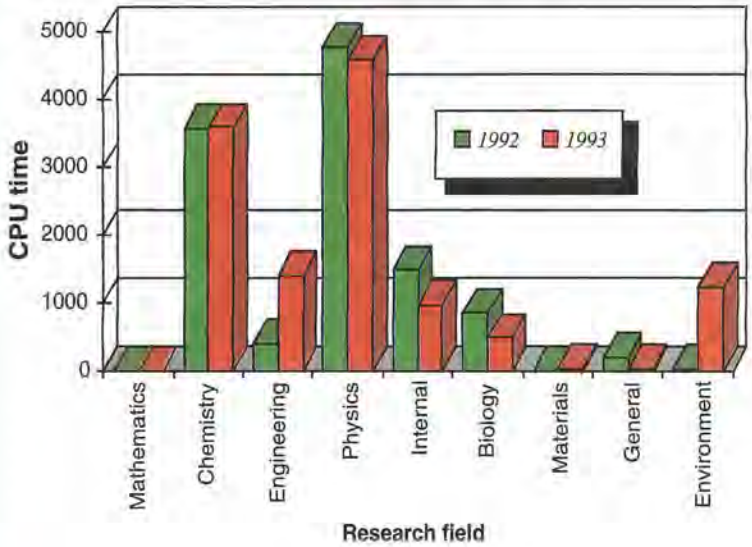
Resources at CSCS are not limited to hardware; the user support structure puts users in contact with resident experts to solve problems and optimize codes. The visualization team at CSCS works closely with users to realize sophisticated graphical representation of complex data sets. Also, the number of applications available to users has been increased so as to address a wider range of scientific disciplines.

But even the most sophisticated equipment does not guarantee success. We must examine, evaluate and influence the decisions which will produce the next generation of HPC machines. The new Section of Research and Development has this exciting and difficult task. One area of interest for this group is massively parallel processing, but their scope also extends to intelligent tool environments, education and technology transfer.

Knowledge exchange, collaboration and education are needed for technology to yield results. Such activities are of particular interest for industry. To successfully compete in the future, a combined effort of government, academia and industry is necessary to coordinate interdisciplinary research, create industrial consortia and lead collaborative R&D projects. CSCS has made significant steps in this direction through project acquisitions and the initiation of an industrial partnership program.

CSCS maintains a high profile domestically and in the international HPC community through publications and presentations by its staff. CSCS

CPU time according to research field in 1992 and 1993



CSCS OVERVIEW

also is the home to SPEEDUP, a society which biannually brings together high–performance computer users and publishes a journal based on these meetings.

CSCS STRUCTURE

From an initial starting of eighteen staff members in 1991, CSCS has grown to a current level of almost fifty. The influx and nature of research projects into CSCS has necessitated the hiring and expansion of scientific staff. These professionals can be found working within or between any of the four sections: the Section of Scientific Application Matters (SeSAM), the Section of Research and Development (SeRD), the Section for Technical Operations (SeTO) and Central Functions (CeF). In addition to the work duties associated with a section, a variety of cross–sectional projects maintain a collaborative work spirit between sections.

SECTION OF SCIENTIFIC APPLICATION MATTERS

SeSAM is the user support organ of CSCS. This section provides assistance to users in a multitude of ways from porting and optimizing codes, to visualizing results and scientific project work in the applied field. The duties of the SeSAM section include:

- administration of user projects and accounting,
- help desk activities,
- porting and tuning users' codes and commercial packages,
- designing and implementing new codes for the SX–3 in conjunction with users,
- visualizing user results,
- research and development in industrial and commercial applications
- organizing classes, workshops, and conferences,
- prospecting innovative computer architectures to prepare for the future (in collaboration with SeRD).

These activities are vital to CSCS and the CSCS user. Porting codes between different computers can be challenging and frustrating even for a sophisticated user. Optimization methods and the choice of algorithms also differ from machine to machine. SeSAM consultants ease the process; their experience in porting and optimizing codes for the specialized NEC SX–3 can significantly impact run–time and results of user codes and substantially increase the performance/throughput of the system.

Besides this user support activity, SeSAM is actively involved in several projects in the applied field.

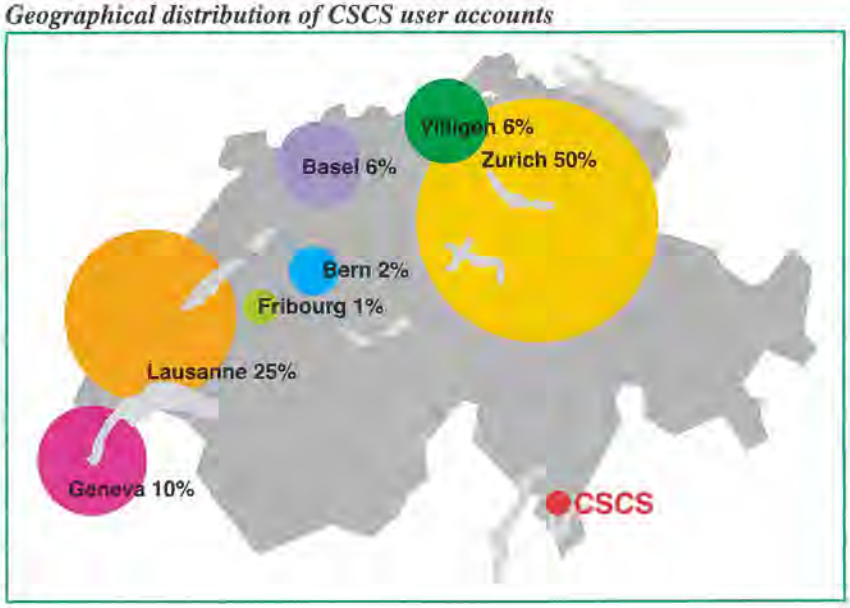
SECTION OF RESEARCH AND DEVELOPMENT

SeRD is the newest addition to the CSCS organization and was founded in January 1993. It's basic mission is:

- developing portable software for computer architectures and modern algorithms,
- developing appropriate tools and tool environments,
- realizing education and training courses in the fields of novel computer architectures and software systems,
- technology–transfer of algorithms, software and software tools for academic, industrial and commercial applications.

Additional activities of SeRD include analysis activities such as continuous evaluation of trends of high–performance hardware, and software systems and developments at other HPC centers world–wide. SeRD also runs CSCS's Parallel Computer Systems and Education Laboratory (ParEdLab) which provides a framework for education and training on novel computer

Geographical distribution of CSCS user accounts

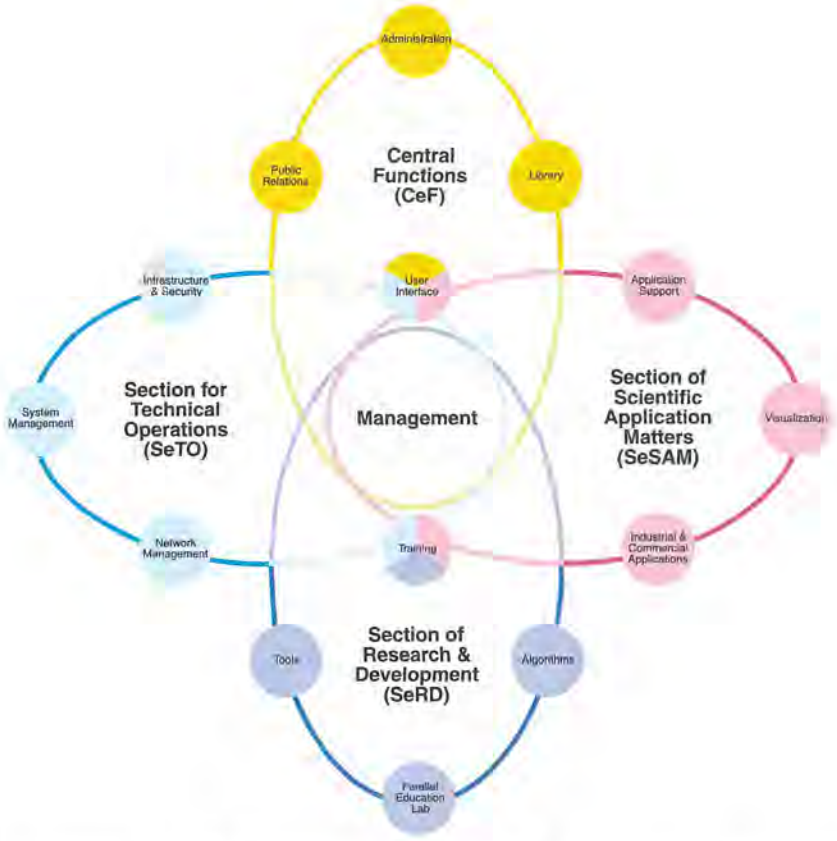


architectures and software systems, knowledge acquisition in the field of HPC, and consultation of academic institutions and industrial and commercial companies. ParEdLab also installs and operates experimental hardware and software components as required for the successful realization of the research and development activities. In addition, ParEdLab houses and operates CSCS's Summer Student Internship Program (SSIP) which disseminates knowledge in programming parallel systems with distributed memory by contributing to teaching and training in parallel computing on the academic level. ParEdLab also organizes the Swiss High–Performance Computing Seminar (SHPCS).

SeRD personnel is mainly financed from third party sources such as the Swiss National Science Foundation and NEC Corporation. A major project of SeRD is the Joint CSCS–ETH/NEC High–Performance Computing Software Development Center whose objective is to turn massively parallel distributed systems into a practical tool.

SECTION FOR TECHNICAL OPERATIONS

SeTO is responsible for daily operation and maintenance of computational facilities for both the CSCS users and staff. SeTO is the basis upon which many of CSCS activities are built: user support, optimization, visualization, and R&D to mention a few. SeTO breaks down into several groups: infrastructure, network management, system management, and security. These groups are responsible for coordinating the CSCS's multiple platform environment and planning the evolution of CSCS's facilities.



CENTRAL FUNCTIONS

The Central Functions group performs activities that impact the entire CSCS organization–staff and users. Administration, marketing and publishing functions as well as library operations fall under CeF.

FINANCIAL AND PERSONNEL DEVELOPMENT

FINANCIAL DEVELOPMENT

All installations of computing hardware and networks as well as software until the end of 1993 were covered by the 40 Mio SFr. grant of the Swiss government under the special initiative to promote computer science. Costs for other infrastructure like building and its maintenance, wide area network and electricity are covered under the financial aspects of ETH Zürich.

Expenses for the central service function (to the national users) of CSCS are covered by federal funds through ETH Zürich, whereas additional activities, mainly the newly acquired research and development projects in 1993 were entirely self–financing from third–party funds, resulting in a surplus of 135,000 SFr. Industry contribution with more than 60% was the largest source of third–party funding.

CSCS Financial Development

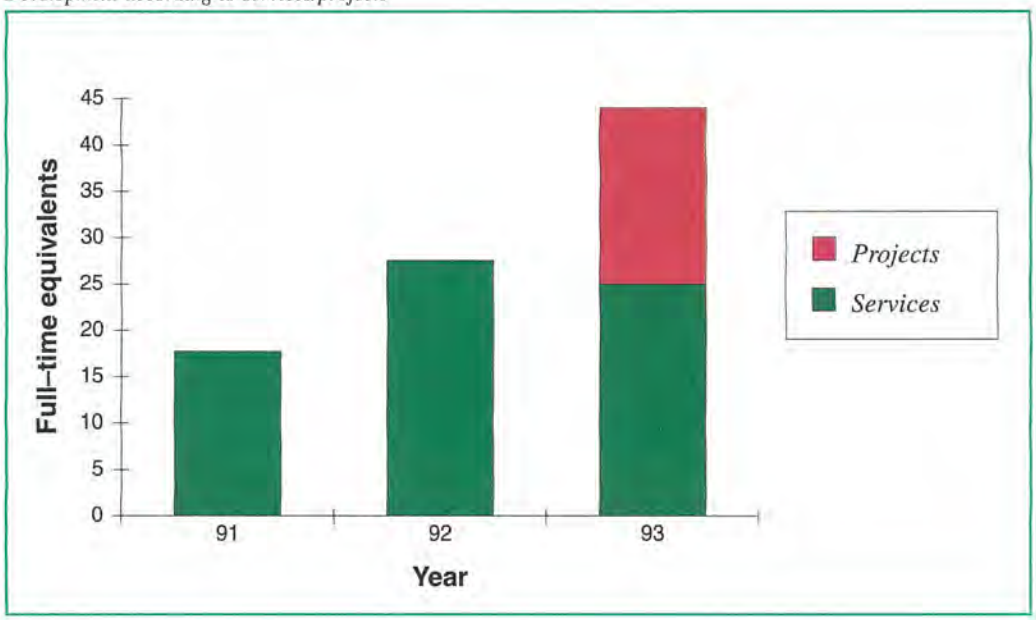
	(KSFr.)		(KSFr.)	
	Federal funds		**Third-party funds**	
	1992	**1993**	**1992**	**1993**
Personnel	1,435	1,932	449	1,198
General expenses	512	556	641	678
Total expenses	1,947	2,488	1,090	1,876
Federal contribution	1,947	2,488		
Income			928	2,011
Result	0	0	–162	135
Third-party funds				
Reserves/contributions 31.12.1992			1,597	
31.12.1993				1,732
Increase 1993				135

PERSONNEL DEVELOPMENT

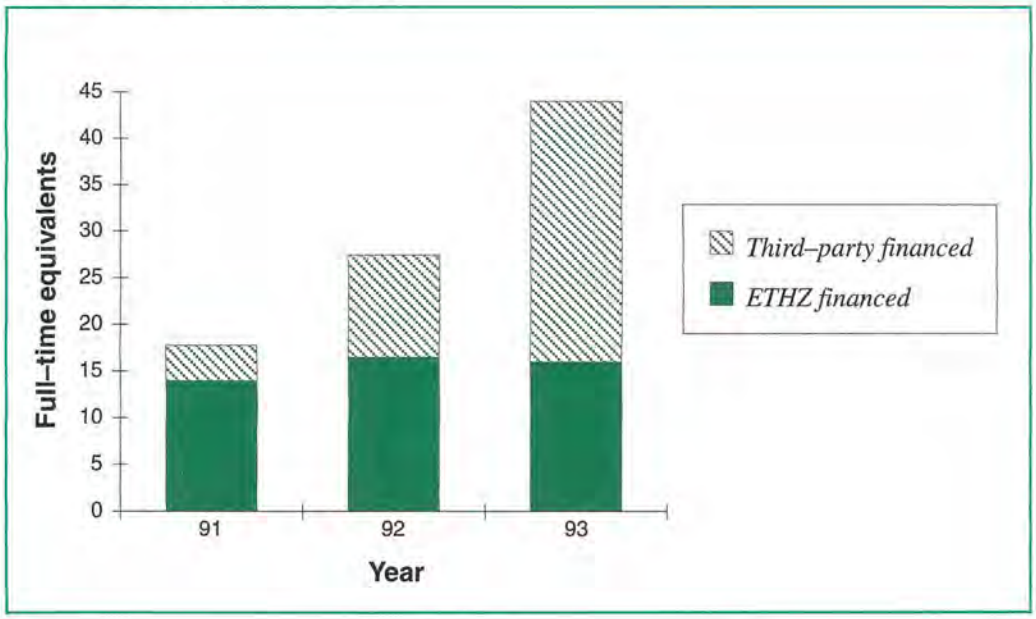
After recruitment of the initial staff for services in 1991 and 1992, CSCS saw a rapid additional increase of personnel due to the acquisition of research and development projects in 1993.

At the end of 1993, project–related staff reached an overall share of 43% of the total staff of 44 full–time equivalents. Twenty–eight full–time equivalents were third–party financed.

Development according to services/projects



Breakdown according to financial resources



COMPUTING FACILITIES

TECHNICAL PROFILE OF CSCS

GLOBAL CONCEPT

CSCS recognizes the importance of defining global concepts and long–term planning. The concept of an application–oriented and integrated computing environment distributes functions out of a computer across networked devices linked with HiPPI connections. This concept drives the planning of CSCS facilities and network topology.

The National Storage Laboratory at Lawrence Livermore National Laboratory, USA, demonstrated the feasibility of this concept under the name meta–computing and the concept has now become widely accepted in HPC centers in the United States and abroad.

VECTOR POWER

In the third quarter of 1993, CSCS upgraded its high–performance vector computer NEC SX–3/22 to a SX–3/24R; this new configuration has two–processors each one having 16 pipelines; this means that a total of 32 pipelines produce results in parallel. Thus, the SX–3/24R carries the concept of parallel processing far beyond that of other vector computers. The upgraded SX–3 has two GBytes of main memory and four GBytes of extended memory. Peak performance goes up to 12.8 Gflops at a clock cycle of 2.5 ns.

A Convex C3820 was installed in the first quarter of 1993 to act as front–end, file server and archive server. It has one Gbyte of main memory.

PARALLEL POWER

In the final quarter of 1993, it was decided to install a Convex Meta Series (workstation cluster: 8 x HP node) to complete the scalar and parallel computing power.

Two parallel machines were installed during 1993: a Meiko CS–1/860 and a NEC Cenju–2. The Meiko Computing Surface is a hosted multi–computer system with eight nodes, each

NEC SX–3/24R

MATHEMATICAL LIBRARIES

ASL
Advanced Scientific Library from NEC

MATHLIB
NEC Mathematical Library

NAG
General Mathematical Library

BLAS
Basic Linear Algebra Subprograms, Levels 1, 2 and 3

EISPACK
Matrix Eigenvalue Problem Solver

LINPACK
Linear Algebra Subroutine Library

MINPACK
Non Linear Optimisation Package

LAPACK
Linear Algebra Routines Successor to Linpack and Eispack

VISUALISATION

AVS
Interactive Modular Visualisation Package

BASPL
Visualization for Finite Element Applications

IRIS EXPLORER
Interactive Modular Visualisation Package

MOLEKEL
Molecular Visualisation Program

PV–WAVE
Data Analysis and Visualisation Software

one consisting of one Intel/i860 and two transputers T805 with 8 MBytes of memory. The system has a built–in Sparc2 host board with 32 MBytes, one GByte of disk and runs Suns SunOS 4.13.

The NEC Cenju–2 system is a hosted multi–computer system with sixteen processing elements, each one consisting of two MIPS R3000 processors with 64 MBytes of memory. The system is hosted by a NEC ESW 4800/215. The nodes of the Cenju–2 are connected with a multi–stage, shuffle–exchange network.

In the second quarter of 1994, the Cenju–2 will be upgraded to a Cenju–3 system which is equipped with the more powerful VR4400SC processors and more powerful communication. The Cenju–3 will have more than one hundred processing elements and several GBytes of distributed memory.

VISUALIZATION POWER

CSCS's visualization laboratory provides the high–quality facilities needed for visualization,

STK ACS 4400

Cenju–2

Convex 3820

SCIENTIFIC SOFTWARE

AMBER
Molecular Mechanics and Dynamics

AMOSS
Large–Scale Electronic Structure Calculations

ASTRID
A Program Development and Execution Environment

LS DYNA 3–D
Structure Analysis and Crash Simulation

GAUSSIAN
Ab initio Electronic Structure Calculations

MSC/NASTRAN
Finite Element Analysis Program

TASCFLOW
3–D Computational Fluid Dynamics Package

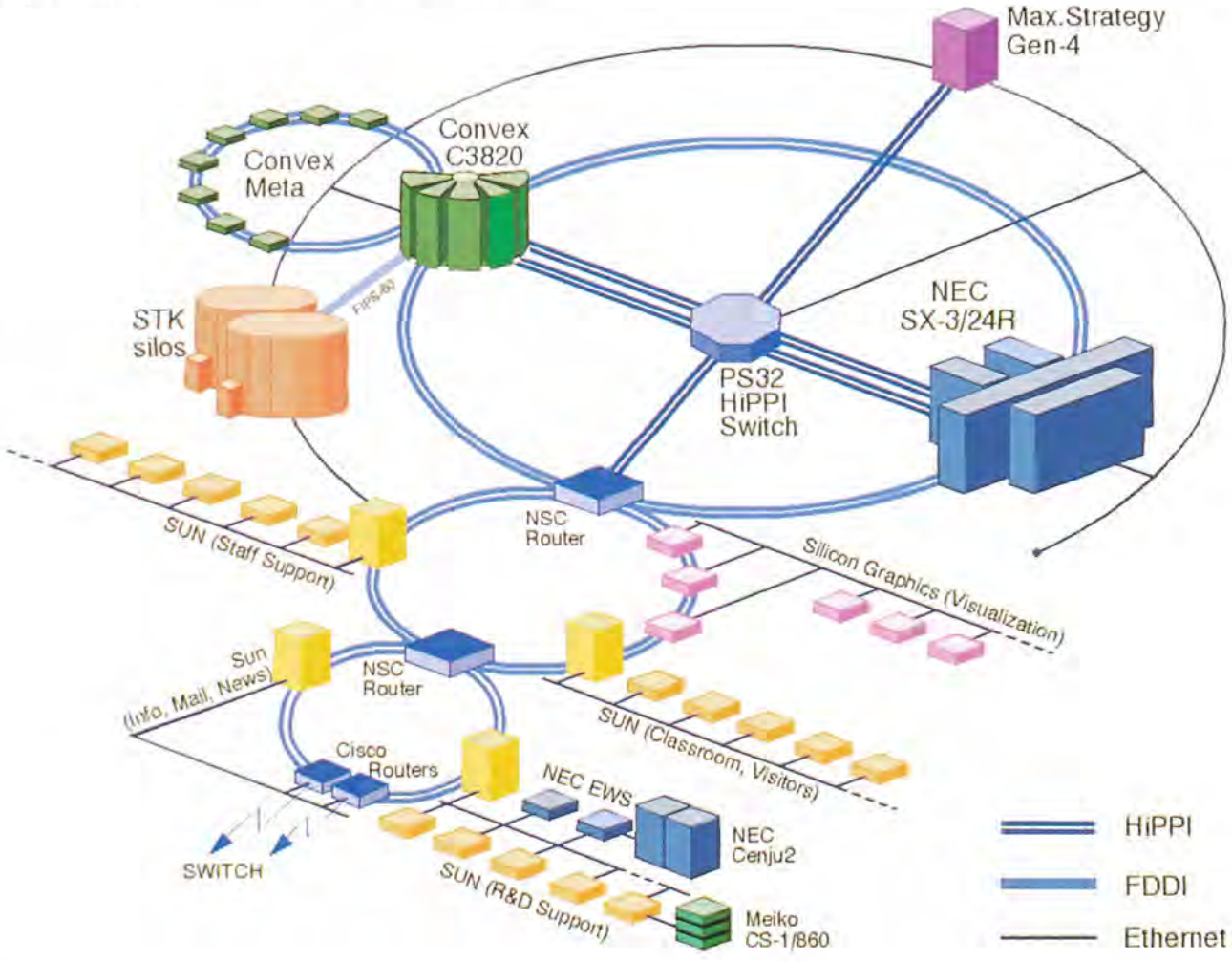
video animation, and color reproduction. Silicon Graphics (SGI) Iris workstations are accessible in the laboratory for visualization: a two–processor PowerVision 4D/420VGX, a Crimson VGX for high–performance interactive three–dimensional rendering, and a Personal Iris 4D/30TG for developing and tuning graphics applications and medium range visualization projects. Smaller systems such as the Indigo R3000 and Indigo2 are provided for visualization staff and long–term guests.

FILE–SERVING AND ARCHIVING

A high–speed, large–capacity file server and archive server system is an essential component for large data handling produced by high–performance computers. To meet this demand, a Convex system with UNITREE file management software was selected and installed in early 1993. Maximum Strategy Gen–4 RAID disks (52 GBytes) are used as UNITREE disk cache. The Convex has 40 GBytes local disks for software and users. It is planned to enlarge the RAID disk storage capacity in 1994.

The initial archiving capacity on cartridges was 1.12 TBytes using a STK silo with four 18 tracks drives. In the last quarter of 1993, a second STK silo (with four drives) was installed as well as an upgrade from 18 tracks to 36 tracks drives, leading to a potential capacity of four TBytes. Helical scan technology should enlarge storage capacity up to 200 TBytes in the near future.

The CSCS internal network



CSCS NETWORK

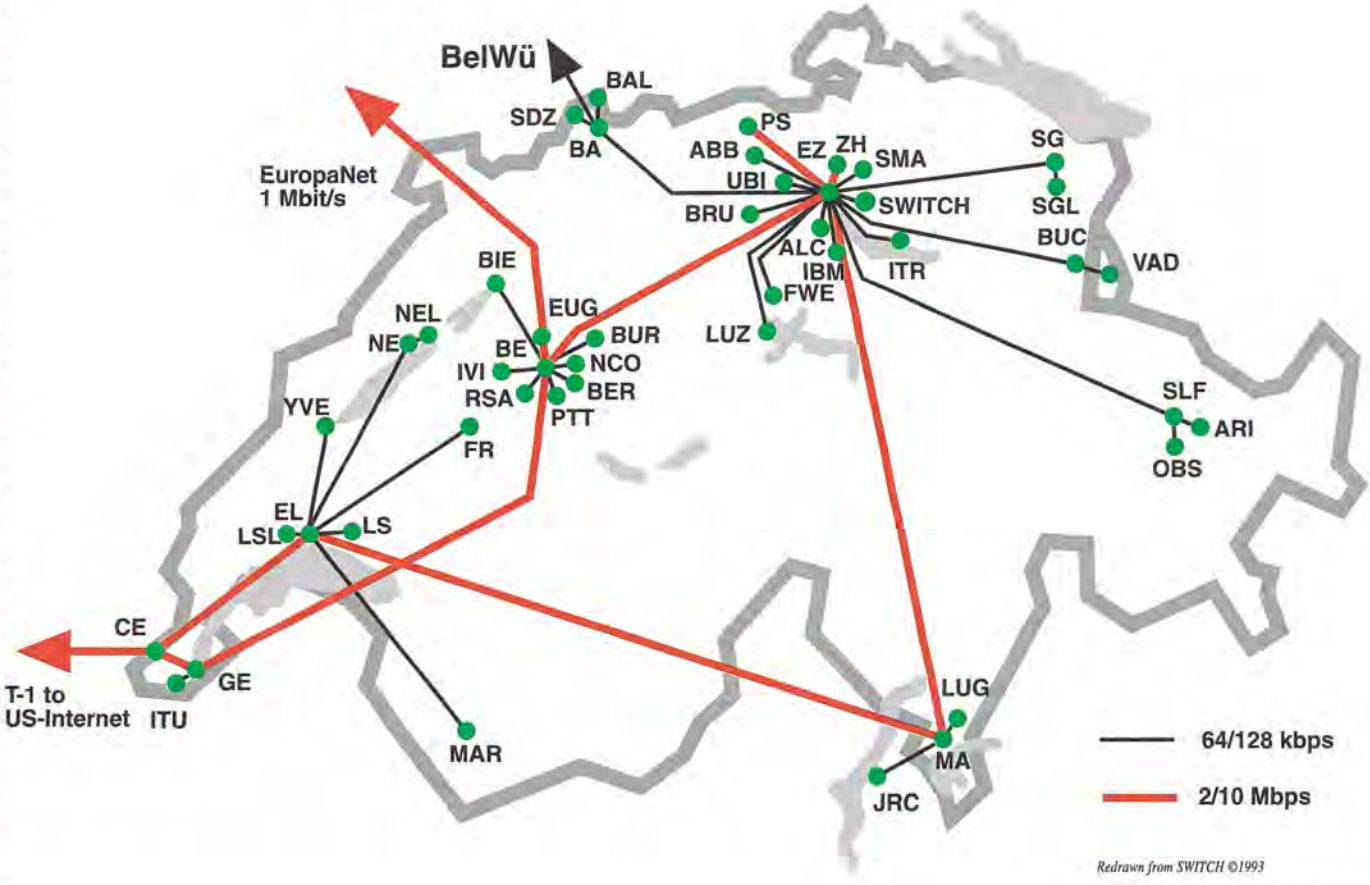
The local network topology is directly driven by the application–oriented and integrated computing environment concept: the core of the data network is a HiPPI switch NSC / PS32 connecting the SX–3, the Convex C3820, the MaximumStrategy RAID disks and the NSC DX equipment. The user's network consists of a hierarchy of FDDI rings with various hierarchical security levels, and Ethernet as the control network. Each workstation complex and their associated file–server(s) is also connected to a FDDI ring according to the appropriate security required by the function.

SWITCH NETWORK

The Swiss education and research network, SWITCH, provides CSCS with the internet connection to the world. Fault–tolerant remote access to CSCS facilities is guaranteed via 2 Mbit/s leased lines. Upgrades are planned to 34 Mbit/s and later on up to 150 Mbit/s.

COMPUTING FACILITIES

The SWITCH network



Redrawn from SWITCH ©1993

SUMMER STUDENT INTERNSHIP PROGRAM

In collaboration with other Swiss institutions, CSCS organizes each summer a summer course for undergraduate students. In the summer break, for a duration of eight to twelve weeks, the students are taught basic skills in programming distributed memory parallel processors (DMPPs). Each student also works on and finishes a small research project which in general is part of some larger ongoing research effort. The topics center around today's well known programming problems of DMPPs, i.e., development of programming tools, new algorithms, and applications for such machines.

Students benefit in two ways from the internship program at CSCS:

- education and training in parallel distributed programming,
- gathering experience in project–oriented applied research and development.

Students should have passed their Bachelor of Science or some equivalent degree. No advanced knowledge in parallel systems and parallel programming is required. The course starts with introductory lessons on parallel distributed systems, covering both hardware/system issues and programming of DMPPs. It also teaches a methodical approach to software development, covering software design, coding, test, and documentation. After having gained working knowledge with parallel distributed systems, students obtain a certificate.

Naturally, the course is open to all such students in Switzerland, but any foreign student with similar qualifications is also welcome to participate. Thus our program makes a contribution to student's mobility, but also contributes to the integration of Switzerland in the international network of research and education.

The Summer Student Internship Program is embedded in the Parallel Computer Systems Education Laboratory (ParEdLab), part of CSCS's Section of Research and Development (SeRD).

SCIENTIFIC VISUALIZATION COURSE AT CSCS

The CSCS Course in Scientific Visualization was held in September 1993. It was organized by the visualization team at CSCS with the major theme of general purpose visualization packages. Participants from Italy and Switzerland followed the course. The course was structured with morning sessions covering software concepts and afternoons with practical exercises. This offered participants the opportunity to solve prepared exercises or to work on their individual visualization problems in close collaboration with the supervisors. Although the course gave special focus to the AVS and Iris Explorer packages, two additional application specific packages were presented: Molekel for molecular graphics and Baspl for finite element analysis and visualization. A technical session on algorithms provided insight to some visualization techniques such as isosurface generation or particle tracing. Such courses will continue to be offered at CSCS or at remote sites.

RETRAINING AND CONTINUING EDUCATION FOR UNEMPLOYED PROGRAMMERS

A three–month, intensive retraining course was offered to a group of unemployed Ticinese programmers. Presented in collaboration with the canton of Ticino's employment office, the course re–educated the programmers in the newest computer software and programming methodologies. Three goals were reached through this program: combating the unemployment problem, placing highly qualified, retrained people in the "job bank" for the local industry, and building a foundation for continuing education in Ticino. Six of the participants found employment upon completion of the course. Four students were invited by CSCS to take part in a three–month, intensive project development.

SEMINAR ON INDUSTRIAL APPLICATIONS

In September 1993, a Seminar was held at CSCS on Industrial Applications. It was jointly organized by CAD–FEM, CSCS and NEC Corp. The topics focused on impact analysis and metal forming with LS DYNA 3–D on the NEC SX–3, a field that in recent years has received growing attention among HPC applications. Approximately twenty attendants registered for the seminar from a number of major companies and institutes in Switzerland and Italy. The course gave special attention to the capabilities and recent enhancements in impact and crash worthiness analysis for which LS DYNA 3–D was originally designed. Airbag and occupant simulation have become one of the most important tasks in automotive crash modelling, and were demonstrated using LS DYNA 3–D with impressive animated graphics result.

INDUSTRIAL PARTNERSHIP PROGRAM

CSCS is among the few European HPC centers offering a collaborative, industrial partnership program.

Companies are increasingly turning to R&D to improve their competitiveness in the 1990s. The opening of markets and the emergence of new, aggressive competitors have resulted in increased competitive pressure. Even more important, a high level of innovation is needed in order to stimulate demand, changing consumption patterns in an attempt to overcome market saturation. Building expertise in R&D is difficult especially today with a changing research paradigm. Large–scale computational science is one such change, where implementation/utilization presents difficulties and challenges even for high–tech companies with an experimental background.

Joint research agreements between industry and HPC research centers promise to expedite the process of acquiring expertise in computational science, while simultaneously optimizing investment. This strategy has successfully passed the test in the USA., with the National Science Foundation Centers and the National Laboratories teaming up in fruitful cooperation with industry and financial institutions.

This is a new concept in Europe: CSCS is one of the first HPC centers to offer a full–fledged Industrial Partnership Program (IPP). The key element of CSCS's IPP is the aggregation of internal resources with the expertise and knowledge of the Swiss academic community.

The IPP is based on a scalable service, offering either opportunities for the execution of specific industrial problem–focused research and development projects, or a more far–reaching, long–term collaboration. The program is flexible and tailored to meet the individual needs of a company–whether just beginning to use HPC or already fluent with the techniques and the methodology of large–scale computational science. In addition to customized partnerships, the IPP offers training, internship for company employees, management consultancy and other services designed to help companies structure and manage their computational activities. The IPP of CSCS is not limited to benefitting large industry; because of it's individual nature, the IPP can respond to the needs of small and medium–sized companies wishing to pursue new avenues in R&D but lacking internal availability of the specific competence and computational resources required.

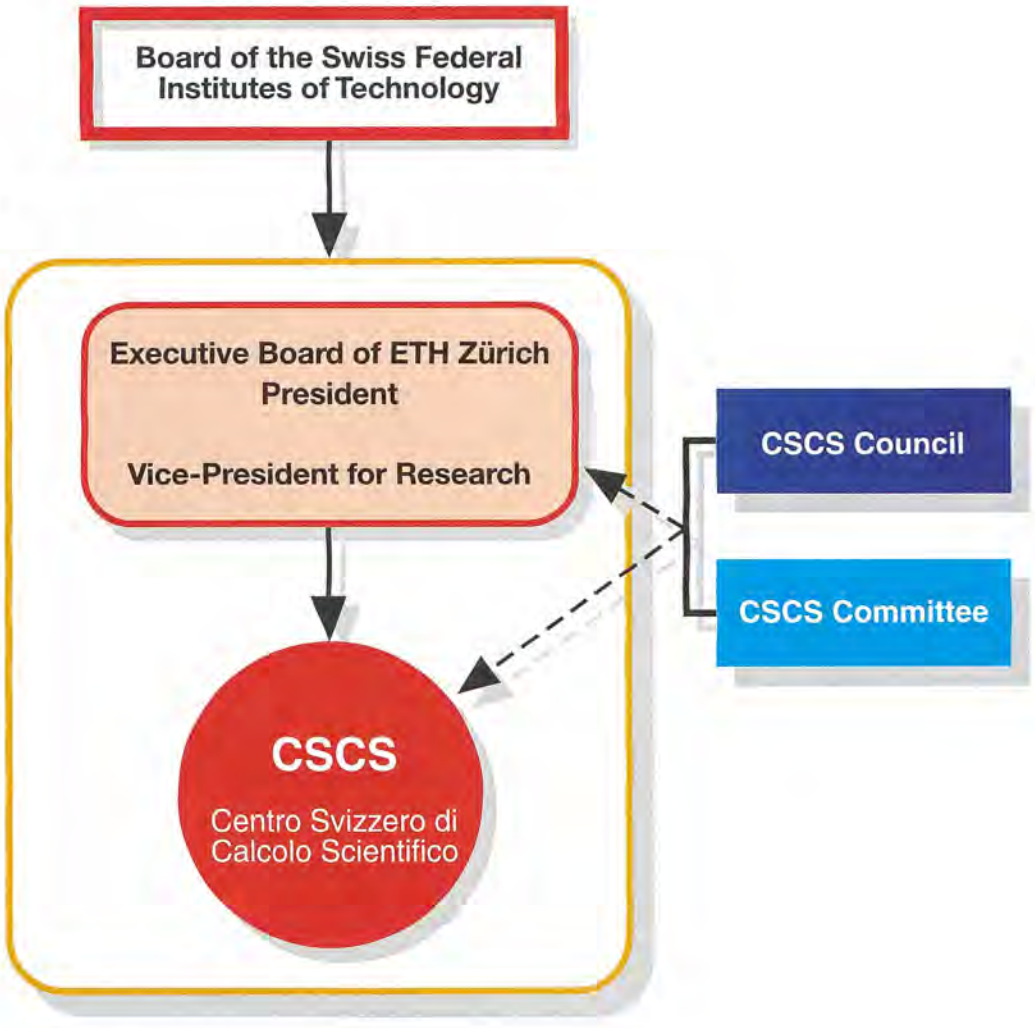
CSCS ADMINISTRATION

The Swiss Federal Institute of Technology Zurich (ETH Zürich) was mandated by the Swiss government to manage CSCS on behalf of the Swiss scientific community.

On the national level, the CSCS Council and the CSCS Committee oversee the center's activities. The CSCS Council advises on strategic development of the center.

The CSCS Council members are representatives of the user organizations, namely the federal institutes of technology and universities, and the federal administration.

The CSCS Committee proposes computing resource allocation and distribution based on scientific criteria and supervises aspects of technical operation.



CSCS COMMITTEE AND COUNCIL MEMBERS

THE CSCS COMMITTEE

Prof. Dr. Ralf Hütter
President CSCS Committee
Vice-president for Research, Swiss Federal Institute of Technology Zurich

Prof. Dr. Roberto Car
Department of Computational Condensed Matter Physics, University of Geneva
Director of IRRMA (Institut Romand de Recherche Numérique en Physique des Matériaux), Lausanne

Prof. Dr. Gervais Chapuis
Centre Informatique, University of Lausanne

Dr. Jean-Jacques Paltenghi
Direction Swiss Federal Institute of Technology Lausanne

Prof. Dr. Fritz Rösel
Director Computer Centre, University of Basel

Dr. Alfred Scheidegger
Director CSCS, Manno

THE CSCS COUNCIL

Prof. Dr. Gervais Chapuis
President CSCS Council

Dr. Kurt Appert

Prof. Dr. Kurt Bauknecht

Prof. Dr. Olivier Besson

Prof. Dr. Roberto Car

Dr. Jean-François Descloux

Henri Garin

Dr. Georges-André Grin

Brian Housley

Prof. Dr. Ralf Hütter

Prof. Dr. Peter F. Meier

Prof. Dr. Fritz N. Rösel

Dr. Friedrich W. Schlepütz

Claude Wuischpard

Dr. Paul-Erich Zinsli

"The CSCS provides to the Swiss universities excellent opportunities to remain in the forefront of high performance computing. The Swiss scientists geographically dispersed in their campuses can only benefit from a strong supercomputing facility managed by a single organisation. Therefore all the efforts should be made on the national level to reinforce and extend the services offered by the CSCS."

Prof. Dr. Gervais Chapuis
President CSCS Council

Professor in the Physics Department at the University of Lausanne. Main research activities in computing, diffraction and structure modelling of aperiodic crystals and associated phase transitions.

CSCS COUNCIL MEMBERS

"The CSCS is *the* Swiss service centre for high–performance computing. Academia expects CSCS to provide it with competent advice and support for the use of high–performance computer systems. It is the CSCS's function to come to grips with new developments so that it is always in a position to offer its users a leading–edge service and the best one possible using the available resources. An effective and efficient CSCS is intended to avoid the unnecessary splitting–up of Swiss activities in the field of super–computing."

Prof. Dr. Kurt Bauknecht

Full professor at the University of Zurich . Director of the Institute for Computer Science. Swiss delegate and vice–president of the Federation of Information Processing (IFIP). Scientific adviser to the Swiss National Fund. President of the computer–science commission of the Conference of Swiss universities (CICUS).

"During the past three years the Swiss Scientific Computing Center (CSCS) has established itself as the leading national supercomputing center by providing the Swiss scientists and engineers with one of the most powerful vector processors available today. This has contributed to maintain at highest international level the Swiss community working at supercomputing applications in several areas of science and technology. The role of CSCS has been particularly important for an institute like the Institut Romand de Recherche Numérique en Physique des Matériaux (IRRMA) whose research is based to a large extent on projects that require a very large computational power. Some of the most ambitious scientific projects of IRRMA in the last two years have been possible thanks to CSCS."

Prof. Dr. Roberto Car

Professor of Computational Condensed Matter Physics at the University of Geneva and director of IRRMA (Institut Romand de Recherche Numérique en Physique des Matériaux) at the Swiss Federal Institute of Technology Lausanne.

"CSCS should promote the design of specific software in the cutting edge domains of our industry."

Dr. Jean–François Descloux

Director of Computing Service of the University of Fribourg . Chairman of the "Computing Services" group of CICUS. His background includes computer science, political research and economics.

CSCS COUNCIL MEMBERS

"Computer sciences are among the most important innovations that were offered to the man in this century. CSCS is a very important contribution to the research in Switzerland. As such, it will help the country to keep a leading position in computer sciences."

Henri Garin

Director of the Swiss Federal Office of Information Technology and Systems, Berne. His background includes computer science and physics.

"The great challenge for CSCS and responsible authorities: to keep the momentum under changing conditions and demonstrate that it was right to create a national center around a supercomputer in Ticino."

Dr. Georges–André Grin

Advisor, Board of the Swiss Federal Institutes of Technology. His background includes nuclear physics and five years as science attaché in Washington D.C.

"Despite the enormous increase in the power of work–stations there are still many areas of research which require computing resources beyond what is normally locally available, i.e. for super–computing. This need, together with active research in future super–computing methods, requires resources and know–how of a specialised nature. The significant contribution of the CSCS is to focus these facilities and expertise in a centre of competence which is both an extension to and accessible by (but not exclusively) all Swiss universities."

Brian Housley

Head of Computer–services Department at the University of Berne. His background includes mathematics, physics and computer science.

CSCS COUNCIL MEMBERS

"Science has obtained a third leg; besides theory and experiment, computer modelling has gained increasing importance. For many scientific fields the application and use of computing, and supercomputing in particular, has brought great advantages and has become a necessity. Physics, chemistry, biology, climate research, geoinformatics and turbulence studies are examples. I have no doubt that advanced computing applications will penetrate all fields of science and technology, and that its intelligent use will be a decisive factor in economic success."

Prof. Dr. Ralf Hütter

Vice–president for research at the Swiss Federal Institute of Technology Zurich. His background includes microbiology, biotechnology and research ethics.

"I hope that the CSCS becomes a national center for research in the field of computational science, a place where PhD students and postdoctoral fellows can stay for a couple of months to build up contacts and to exchange ideas with people working on similar problems."

Prof. Dr. Peter F. Meier

Department of Physics, University of Zurich. His recent research has been in nonlinear dynamics and electronic structure calculations.

"The national high–performance computer centre in Manno has been offering an internationally competitive environment to Swiss universities by providing top–grade computer capacity for science and research. It is to be hoped that the use of new computer architectures will enable it, in future, to maintain this leading position, that is fully comparable with other countries."

Prof. Dr. Fritz N. Rösel

Head of the university computer centre in Basel. His background is theoretical physics.

CSCS COUNCIL MEMBERS

"The CSCS provides an important component in the network of Swiss data–processing resources, in that it offers academic institutions capacities that are greater than those available locally. Of equal importance is its function as a centre of excellence for high–performance computing, where it passes existing know–how on to interested users (not only academic circles), takes on future–oriented developments and implements them, thereby helping universities and industry to maintain or achieve first–class competitive positions."

Dr. Friedrich W. Schlepütz

Head of Paul Scherrer Institute's Computing Department (data–processing services). His background is experimental physics.

"The federal purchasing agency (EDMZ/OCFIM/UCFSM), responsible for all hardware and software investments and operating costs of the Swiss government, has participated in the Swiss supercomputer development since 1986 with the purchase of a system CRAY–1S at the Federal Institute of Technology Lausanne and system upgrades to CRAY Y–MP at both Federal Institutes. A major contribution to leading edge research in Switzerland was accomplished with the procurement of the national supercomputer SX–3 at Manno/TI in 1992 at optimal price/performance conditions."

Claude Wuischpard

Director Federal Office for Printing and Material (EDMZ/OCFIM/UCFSM). Vice–president Committee for Federal Procurement.

"The CSCS has developed into a Swiss centre of excellence for the use of high–performance computers and is thus an interesting partner for cooperation in Europe and beyond. This networking needs to be developed further."

Dr. Paul–Erich Zinsli

Vice–director of the Swiss Federal Office for Education and Science with responsibility for the research area. His background is in physics.

CSCS PROJECTS

There are numerous projects conducted at CSCS in addition to the projects of CSCS users. Such projects can be internal projects, collaborative project efforts between CSCS personnel and other organizations, or projects conducted by visiting researchers at CSCS. CSCS projects are an exciting way to illustrate the capabilities of the CSCS environment and resources; we are happy to be able to offer you this summary.

In 1993 there were thirty–one CSCS projects. This section contains twenty–four brief project abstracts and seven project reports. Institutes are listed by their accepted acronyms and can be cross–referenced to the address list at the end of this document.

PROJECT ABSTRACTS

PROFILING SUPPORT ON THE CENJU–2

Involved Persons:

A. Endo (a)
M. T. Nyeu (b)
B. Wylie (c)

a) NEC, SX–Center (Switzerland)
b) University of California, Irvine (USA)
c) CSCS

Funding Sources:

NEC (Germany)

Duration:

August 1993–October 1993

PARITY (PARallel Instrumented Tracing library) is a subroutine library which can be used along with MPI to generate trace information to monitor performance of parallel program execution. The PARITY generated trace data can in turn be used in visualization of the parallel algorithms using ParaGraph for display. The library provides portable syntax for key communication primitives and related systems calls required to generate tracing information. It also allows generation of trace data for some widely–used, high level communication operations, such as broadcast and global reductions. By judicious use of the trace routines, a user can produce small, yet informative trace files, whose generation has little impact on the performance of his/her code. Our experiences show that careful use of trace routines help bringing the impact to a minimum level which is not enough to change the important features of the run time behavior of a parallel program. PARITY along with the ParaGraph offers an environment where users can observe the run–time behavior, identify computational bottlenecks and communication related programming errors, and hence tune the system to achieve higher performance.

REFERENCES:

[1] Heath, M. T. and J. E. Finger. *ParaGraph: A Tool for Visualizing Performance of Parallel Programs.* Technical report, Oak Ridge National Laboratory, October 1992.

[2] *Message Passing Interface Forum* Document for standard message passing interface. Draft (September 1993).

PROJECT ABSTRACTS

DEVELOPMENT OF A MAGNETOHYDRODYNAMIC EQUILIBRIUM AND STABILITY PROGRAM FOR A DOUBLET PLASMA DEVICE

Involved Persons:
R. Gruber (a, b)
G. Jost (a, c)
S. Medvedev (d)
S. Merazzi (a, e)
F. Troyon (f)
L. Villard (f)
a) CSCS
b) SIC–EPFL
c) NEC, SX–Center (Switzerland)
d) Keldysh Institute of Applied Mathematics (Russian Federation)
e) IMAC–EPFL
f) CRPP–EPFL

Funding Sources:
CRPP–EPFL
CSCS

Duration:
January 1994–June 1995

Stability studies of symmetric toroidal plasma devices such as Tokamaks have shown that the stability behaviours are improving as the elongation of the plasma column increases [1]. The construction of the TCV Tokamak at CRPP–EPFL was mainly based on this result. The high elongation 3 of this machine makes it possible to not only study high elongation plasmas with one magnetical axis, but enables Doublet configurations with two magnetical axis.

Level lines of normal to magnetic surfaces component of plasma displacement.
Most unstable external n=1 kink mode

Before considering a modification of the TCV experiment the stability behaviours of such Doublet configurations are studied. In a cooperation between CRPP and CSCS an equilibrium/stability software package for toroidal plasma configurations with two magnetical axis has been developed. The programs were presented in [2,3].

REFERENCES

[1] Troyon, F., R. Gruber, H. Saurenmann, S. Semenzato, and S. Succi. "MHD Limits to Plasma Confinement." *Plasma Phys.* 26(1A) (1983): 209–215.

[2] Medvedev, S., L. Villard, R. Gruber, and S. Merazzi. *MHD Equilibrium Code for Axisymmetric Plasma with Separatrix.* Technical report CSCS–TR–93–01.

[3] Medvedev, S., R. Gruber, G. Jost, and S. Merazzi. *MHD Equilibrium and Stability Codes for Tokamak Plasma with Separatrix.* CSCS Annual Report 1992, p. 10–13

NUMERICAL SIMULATION OF THE BEHAVIOR OF END–PUMPED SOLID STATE LASERS

Involved Persons:
R. Gruber (a, b)
S. Merazzi (a, c)
Ch. Pfistner (d)
R. Weber (d)
a) CSCS
b) SIC–EPFL
c) IMAC–EPFL
d) Universität Bern

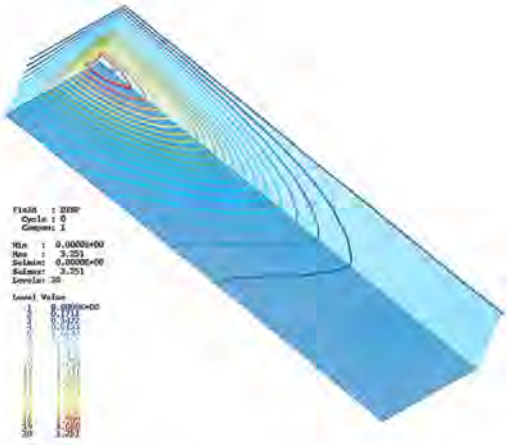
Funding Sources:
CSCS
IMAC–EPFL
Universität Bern

Duration:
July 1993–December 1994

The project aims at the numerical simulation of the performance of end–pumped and side–pumped solid–state laser rods. The simulation essentially consists of three steps: (1) Analysis of the temperature distribution in the rod due to the heat generated by the pump laser (2) Analysis of the deformations and the stresses in the rod due to thermal dilatations caused by the pump laser (3) Analysis of the optical performance taking into consideration the mechanical and the thermal effects. Steps (1) and (2) rely on finite element procedures built into the B2000 analysis system. Step (3) needs special consideration since the equation involved is solved by explicit or semi–implicit pseudo–time integration methods. While steps (1) and (2) can make usage of a similar computational mesh, step (3) requires a far denser mesh.

Cylindrical end–pumped rods have been studied in a project with the University of Bern. The simulations of cylindrical end–pumped rods can most often be reduced to two dimensions. The side–pumped rods exhibit a three–dimensional behavior. Thus, the numerical models become much larger and usage of supercomputers must be made.

During the first three months of the project the pump distribution has been established. Due to the form of the pump distribution a mesh adaptation method is now being introduced in the thermal and the stress solver.



Thermal analysis:
distribution of temperature [K] in Nd:YAG rod (initial mesh)

PROJECT ABSTRACTS

TIGHT BINDING MOLECULAR DYNAMICS

Involved Persons:

G. Benedek (a)
L. Colombo (a)
D.Maric (b)

a) Università degli Studi di Milano (Italy)

b) CSCS

Funding Sources:

Fondation pour les Bourses d'Etudes Italo–Suisses

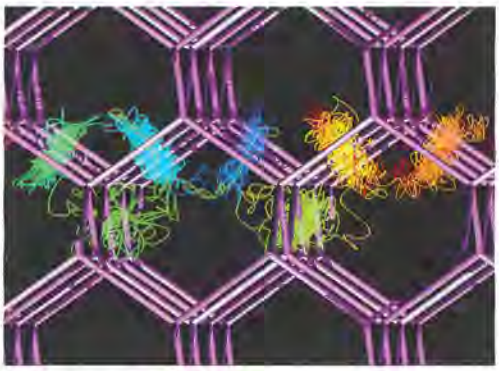
Consiglio Nazionale delle Ricerche (Italy)

CSCS

Duration:

January 1993–December 1995

A semi–empirical Tight Binding Molecular Dynamics (TBMD) method has been implemented and optimized on the NEC SX–3 so that its application to a very large simulation cells as well as very long simulation times are possible. Several mathematical libraries have been tested. The method has been used to study the defect influenced amorphization process of crystalline silicon as induced by ion–beam bombardment. The irradiation process consists of two steps: (i) insertion of defects at a constant rate; (ii) annealing of the sample and observation of its structural properties. Thanks to the large size of the simulation cell (up to 300 atoms) the amorphous network can be characterized both on the short–range and medium–range length scale. The evolution of the electronic properties is monitored during the amorphization process. The resulting structural properties of the irradiated sample are thoroughly compared with the amorphous silicon as obtained by rapid quench from the melt.



H–Trajectory in Si–Crystal

REFERENCES

[1] Colombo, L. and D. M. Maric. "Tight Binding Molecular Dynamics on the NEC SX–3." *Crosscuts* 2(2): 15.

[2] Maric, D. M. and L. Colombo. "Defect Induced Amorphization in Silicon: A Tight Binding Molecular Dynamics Simulation." *Mat. Res. Soc. Symp. Proc.* (in press).

CRYSTALLIZATION IN A FLOW

Involved Persons:

L. Degtyarev (a)
S. Medvedev (a)
A. Pollei (b, c)
P. Rys (c)
F. S. Rys (c)

a) Keldysh Institute of Applied Mathematics (Russian Federation)

b) CSCS

c) ETHZ

Funding Sources:

SNF

NEC (Japan) grant

ETHZ

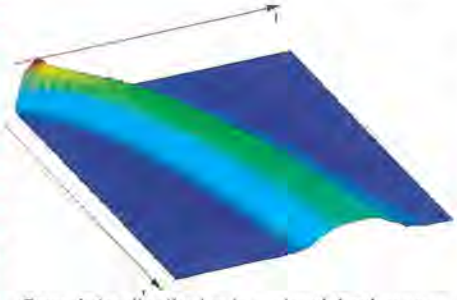
Duration:

February 1993–December 1994

The goal of this project is the numerical simulation of crystallization in various flows with regard to a space dependent crystal size distribution (CSD). A model for such a process must take into account the reciprocal effects of the following subprocesses: mixing due to convection and diffusion, chemical reaction and crystallization, which consists of nucleation and crystal growth. The main stress is put on crystallization itself, but a careful modelling of all the intervening subprocesses is essential. The velocity field, which determines the convective dispersion of chemical species and crystals, can be obtained as a direct solution of the Navier–Stokes equation.

If the laminar flow is independent of chemical reactions and crystallization a steady velocity field may be supplied as a quasi analytical or previously computed numerical solution. For the chemical process the single second order reaction (A+B->P) between two species which form a poorly soluble product has been chosen. No by–products are formed.

The crystallization of a substance in a supersaturated solution is influenced by parameters of the solution as well as of the crystals themselves. Our model considers only the dependency of nucleation and crystal growth rate on the supersaturation. All other influences are averaged in the rate constants. The three mentioned processes are described mathematically by means of mass balances for the solved substances and for the crystal size distribution as well. These equations are solved in two spatial coordinates and one internal coordinate (crystal size) using the QUICKEST algorithm in combination with the ULTIMATE limiter. This method has a low numerical dissipation and therefore it is suitable for the simulation of fields with large gradients. Since there steepness is limited by the grid resolution, moving concentration fronts still cannot be represented accurately enough with fixed grids in large computational domains. In that case the application of an adaptive grid method has shown good results. It now is developed now for steady state calculations for one spatial and one internal coordinate and shall be extended to a second spatial coordinate.



Crystal size distribution in a stirred thank reactor

Involved Persons:
W. A. Cooper (a)
R. Gruber (b, c)
F. Troyon (a)
a) CRPP–EPFL
b) CSCS
c) SIC–EPFL

Funding Sources:
CRPP–EPFL
CSCS

Duration:
January 1993–July 1993

THE MODULAR FUSION REACTOR

One possible way towards a commercially realisable thermonuclear fusion reactor could be a large aspect ratio stellarator with many helical windings. The construction of such a machine could be done in a modular manner. Aspect ratios of the order of several hundred would demand hundreds of identical modules. These modules could be mass produced thus reducing the costs of the machine. Another advantage of the large aspect ratio is the easy access of the interior of the helical torus.

First magnetohydrodynamic equilibrium and stability studies have been performed with the equilibrium program VMEC [1] and the ideal MHD stability program TERPSICHORE [2]. First results have already been published [3]. They reveal that there exist large aspect ratio stellarator configurations with very high β values (β being the ratio between the plasma pressure and the magnetic field pressure) necessary for a fusion reactor.

These high β configurations display isodynamic properties that could have favorable implications for plasma transport.

REFERENCES

[1] Hirshman, S. P., W. I. van Rij, and P. Merkel. "3D Free Boundary Calculations Using a Spectral Green's Function Method." *Comput. Phys. Commun.* 43(1986): 143.

[2] Anderson, D. V., W. A. Cooper, R. Gruber, S. Merazzi, and U. Schwenn. "TERPSICHORE: A 3D Ideal MHD Stability Program." In *Scientific Computing on Supercomputers II* edited by J. T. Devreese and P. E. van Camp. New York: Plenum Press, 1990.

[3] W. A Cooper, F. Troyon, R. Gruber: *Proceedings of European Theory Conference*, 1993, El Escorial, Spain.

PROJECT ABSTRACTS

Involved Persons:
J. Y. Chiu (a)
P. Flükiger (a)
R. Gruber (a, b)
A. Mangili (a)
S. Merazzi (a, c)
U. Meyer (a)
A. Thurnherr (a)
a) CSCS
b) SIC–EPFL
c) IMAC–EPFL

Funding Sources:
CSCS
NEC (Japan) grant

Duration:
March 1993–July 1993

CSCS DEMO ENVIRONMENT

CSCS is visited regularly by several dozens of visitor groups every year ranging from scientists to industrials partners or simply by groups of technical persons. The purpose of this project was the realization of an integrated demo system for showing the latest results of the CSCS Visualization Lab. Fundamental requirements were: the demo system must be accessible by everybody in a simple way, always up–to–date, easy to maintain, update, and extend.

The system is composed of two parts:

The first part is dedicated to the organization and the access to interactive demos that can be executed on a Silicon Graphics workstation. This system is based on an interactive three–dimensional–like menu allowing the user to navigate in the demo structure and start a given demo by a simple mouse click on the associated menu button. The demos are accessible in the same way as the Silicon Graphics demo system by every user logged on any CSCS SGI.

The second part is dedicated to the organization and automatic recording of the different video sequences produced by CSCS. The system is based on an extension of the existing video recording software. Major improvements are the capability to handle huge sequences of images, the parameterization of the video scripts in order to maintain a uniform style in the recording of the different sequences, and the possibility to add and remove sequence in a uniform and simple way.

The system has been successfully implemented and tested by Andreas Thurnherr et al. during his three months 'stage' at CSCS.

PROJECT ABSTRACTS

Involved Persons:
K. M. Decker (a)
J. J. Dvorak (a)
R. M. Rehmann (a)
a) CSCS

Funding Sources:
SNF

Duration:
October 1992–September 1993

SPADE — AN INTEGRATED DEVELOPMENT AND PRODUCTION ENVIRONMENT FOR PARALLEL ARCHITECTURES WITH DISTRIBUTED MEMORY

SPADE is an integrated system for application and program development on parallel architectures with distributed memory. Based on the fundamental philosophy to provide a framework for truly parallel problem solutions, it supports the application user in all phases of the software life cycle.

SPADE is built on top of a collection of libraries which ensure its primary design goal, performance, while keeping architectural features completely transparent. Further design goals are simplicity, expandability and portability. This paper focuses on the description of the three functional units of the SPADE system, powerful application and program development environments and a run–time environment.

REFERENCES

[1] Boillat, J. E., H. Burkhart, K. M. Decker, and P. G. Kropf. "Parallel Computing in the 1990s, Attacking the Software Problem." In *Proceedings, 3rd Graduate Summer Course on Computational Physics: Parallel Architectures and Applications*, Crêt–Bérard (Puidoux), Switzerland, September 3–7, 1990. Edited by K. M. Decker. North Holland (1991).

[2] Decker, K. M and R. M. Rehmann. "ADE—An Application Development Environment for Transparent Use of Scalable Parallel Architectures." In *IFIP TC10/WG10.3 Working Conference on Programming Environments for Parallel Computers*, Edinburgh, UK, April 1992. Edited by N. P. Topham, R. N. Ibbett, and T. Bemmerl. Amsterdam: Elsevier Science Publishers B.V.

[3] Dongarra, J. J. "Library Issues in Open Systems: Portability, Scalability." *Highly Parallel Computing Systems*, IBM Europe Institute (1992).

Involved Persons:
S. K. Estreicher (a)
D. Maric (b)
D. S. Marynick (c)
P. F. Meier (d)
a) Texas Tech University (USA)
b) CSCS
c) University of Texas, Arlington (USA)
d) Universität Zürich

Funding Sources:
CSCS
SNF

Duration:
June 1993–May 1995

CLUSTER MODELING OF SEMICONDUCTORS

Very large scale electronic structure calculations are applied to the semiconducting materials. The PRDDO code of Prof. Marynick has been ported and optimized on the NEC SX–3 so that nearly–*ab initio* treatment of systems as large as 5'000 basis functions and 800–1'000 atoms is possible. This allows the systematic study of the cluster–size effects in the cluster simulations of realistic materials. The microscopic properties of localized defects, impurities and their complexes in semiconductors, as well as defect diffusion and surface effects are investigated.

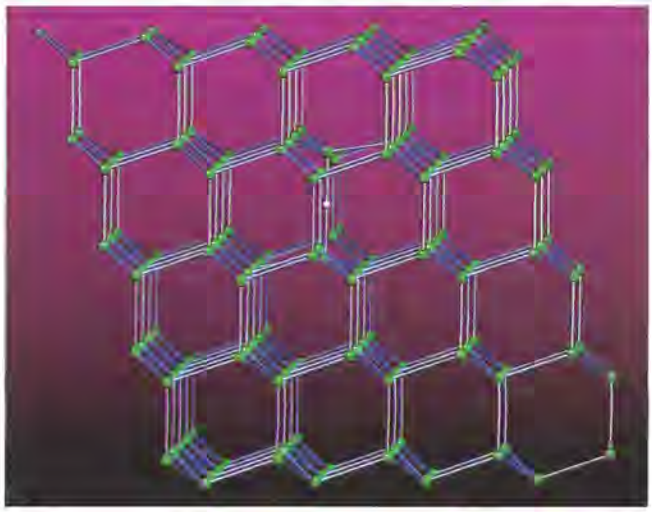
REFERENCES

[1] Estreicher, S. K. and D. M. Maric. "What is so strange about Hydrogen Interactions in Germanium?" *Phys. Rev. Letts.* 70 (1993): 3963.

[2] Estreicher, S. K., D. M. Maric, P. F. Meier and D. S. Marynick. "Very Large Scale Electronic Structure Calculations with PRDDO." *Crosscuts* 2(3): 5.

[3] Maric, D. M., P. F. Meier and S. K. Estreicher. "H,B, H,C, and H,Si Pairs in Silicon and Germanium." *Phys. Rev. B* 47 (1993): 3620.

[4] Maric, D. M., M. A. Roberson and S. K. Estreicher. "Relative Stability of H^T vs. H^* and H_2^T vs. H_2^* in c–C, Si, Ge and α–Sn and their consequences." *Mat. Sci. Forum* (in press).



Impurity in C–Cluster

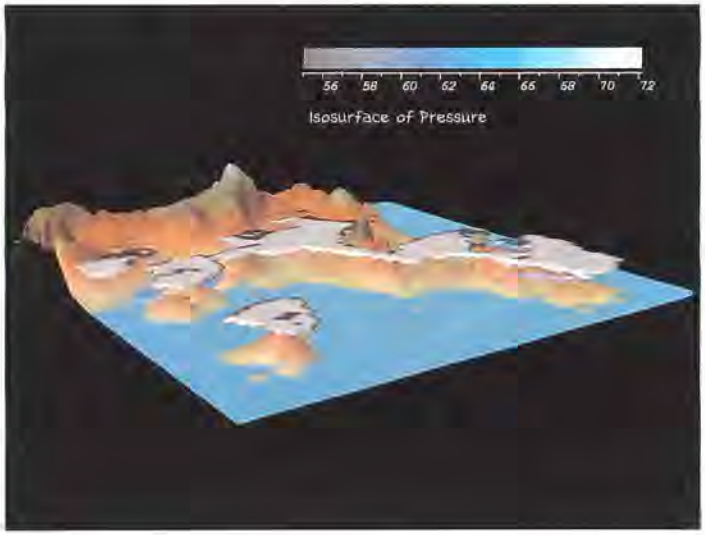
COUPLED SIMULATIONS OF GLOBAL AND REGIONAL CLIMATE

Involved Persons:
M. Beniston (a)
L. Bengtsson (b)
A. Bernasconi (c)
U. Cubasch (b)
M. Esch (b)
P. Lenzen (b)
A. Mangili (c)
R. Marinucci (a)
A. Ohmura (a)
U. Schlese (b)
P. Tschuck (a)
M. Wild (a)
a) Geographisches Institut–ETHZ
b) Max–Planck Institut für Meteorologie (Germany)
c) CSCS

Funding Sources:
CSCS
Geographisches Institut–ETHZ
Max–Planck Institut für Meteorologie (Germany)

Duration:
January 1992–continuing

The growing awareness of the problem of abrupt global climate change resulting from man's activities, and the potential impacts of global warming on the natural and socio–economic environments have motivated us to develop a new avenue of research in Switzerland: namely climate modeling using General Circulation Models (GCMs). These are very advanced systems which in principle have the capability of coupling different components of the climate system, namely the atmosphere, the oceans, the biosphere, the lithosphere and the cryosphere in order to study fundamental mechanisms of climate change and to predict climate trends into the next century. GCMs become, in this manner, one of the principal focal points of climate research in that they provide the climatological data of relevance for a wide range of applications—in particular, impact studies such as response of the natural and economic systems to climate changes.



REFERENCES

[1] Beniston, M. "A Numerical Study of Atmospheric Pollution Over Complex Terrain in Switzerland." *Boundary Layer Meteorol.* 41 (1987): 75–96.

[2] Cubasch U., K. Hasselmann, H. Hoeck, E. Maier–Reimer, U. Mikolajewicz, B. D. Santer and R. Sausen. "Time–dependent Greenhouse Warming Computations with a Coupled Ocean– Atmosphere Model." *Climate Dynamics* 8 (1992): 55–69.

[3] Ehinger, J. and M. Beniston. "Preliminary Results of the APSIS Intercomparison Experiment with the DREAMS Model." *Env. Comp.* (1992/1993) in press.

[4] Giorgi, F., and M. R. Marinucci. "Validation of a Regional Atmospheric Model Over Europe: Sensitivity of Wintertime and Summertime Simulations to Selected Physics Parameterizations and Lower Boundary Conditions." *Quart. J. Royal Met. Soc.* 117 (1991): 1171–1206.

[5] Giorgi, F., M. R. Marinucci and G. T. Bates. "Development of a Second Generation Regional Climate Model (RegCM2): Boundary Layer and Radiative Transfer Processes." *Mon. Wea. Rev.* (1993) in press.

SCIENTIFIC APPLICATION ON THE CENJU–2: MULTIPARTICLE LATTICE GAS MODELS FOR HYDRODYNAMICS

Involved Persons:
R. Chatagny (a)
B. Chopard (a)
a) Université de Genève

Funding Sources:
NEC (Germany)

Duration:
August 1993–October 1993

The future of computer science is heading determinedly now for the use of parallel computers. It is now time for the scientific community to learn how to program these machines and to modify the way of defining the problem underlying algorithms. It is also important to see which kind of performances can be achieved on these new supercomputers.

This paper presents the implementation of a problem of fluid mechanics on the Cenju–2 built by NEC. This project, realised at the CSCS (Centro Svizzero di Calcolo Scientifico) during a Summer School Internship, had the purpose to know better the limits and the advantages of this 16 processor computer.

The problem studied here consists of the implementation of a new model of lattice gas. The interest of this project is therefore double : computer scientific and physical.

REFERENCES:

[1] Benzi, R., S. Succi and M. Vergassola. "The Lattice Boltzmann Equation: Theory and Applications." *Physics Reports* 222, No. 3 (1992): 145–197, North–Holland.

PROJECT ABSTRACTS

PRACTICAL EVALUATION OF PROGRAMMING ENVIRONMENTS FOR PARALLEL DISTRIBUTED SYSTEMS

Involved Persons:

K. M. Decker (a)
R. Gruber (a)
A. Mangili (a)
R. Rehmann (a)
C. Stern (b)
P. Stucki (b)

a) CSCS

b) Multimedia Group–Universität Zürich

Funding Sources:

CSCS
NEC (Japan) grant

Duration:

January 1993–June 1993

A variety of programming environments for parallel computers, commercial products or public–domain packages, are available today supporting different computational models on different levels of abstraction. The aim of this project was to compare different environments for distributed parallel computers and to provide faster versions of existing and often used applications. Improved versions can be achieved by parallelizing the most compute intensive parts of the code to get benefit of the distributed computing power.

Among the different programming environments for parallel computers and workstation clusters, two environments were chosen which support two different types of computational models, a message passing interface environment with which the user directly manages the distributed memory of the parallel computer or workstation cluster and a programming environment supporting virtual shared memory, i.e., a memory management system for distributed memory computers. A drastic selection of possible programming environments was made in order to fit the time constraints (duration of the project six months). 'PVM' (Parallel Virtual Machine) was chosen as a message passing interface because the package is available on a lot of hardware platforms and has as good support. 'Linda' was chosen as a virtual shared memory model also because it is widely used and supported on many platforms.

The spin–density distribution of triplet oxygen

In order to have a better evaluation and comprehension of the advantages and disadvantages of a distributed programming environment a concrete application (Molekel) was selected to be ported to a workstation cluster. Molekel is powerful 3–dimensional molecular graphics package for the interactive visual representation of molecular structures and properties. In order to generate iso–surfaces of electronic properties such as wave–functions, the electron– or the spin–density, Molekel calculates these properties at each point of a predefined 3D–grid. This grid can be large and the time needed to compute these values can be quite big.

In order to be able to use a maximum number of workstations, this project was integrated in a heterogeneous hardware environment. SGI and SUN workstations were chosen as a first step. Other hardware platforms can be easily integrated in future.

The comparison of the different environments was done by measuring speedup, communication parameters, and effort invested in the parallelization.

The following list summarizes the achievements of this project:

- Installation of PVM and Linda on the different hardware platforms.
- Detailed analysis of the applications to be ported, in order to find out which part of the code can be parallelized.
- Design of the communication model and protocols (PVM and Linda).
- Design of the parallel algorithm.— Implementation and test.
- Comparison of the two programming environments.
- Evaluation of the porting of this applications on a real MPP system.
- Documentation

TOWARDS A PARALLEL AND DISTRIBUTED MOLEKEL: A COMPARISON OF MESSAGE PASSING CAPABILITIES OF LINDA AND PVM

Involved Persons:

D. Maric (a)
N. Pelloni (a)
B. Slivnik (b)

a) CSCS

b) Institute J. Stefan (Slovenia)

Funding Sources:

Dipartimento delle Finanze e dell'Economia, Bellinzona

Duration:

August 1993–September 1993

In the frame of the student exchange programme between the Canton of Ticino and the Republic of Slovenia, the first Slovenian student has spent his stage at CSCS. Gaining the first experience in a distributed and parallel computing on workstation clusters comprised the educational scope of this stage. Moreover, practical exercise on the comparison of two software packages for distributed computing on workstation clusters, Linda and PVM, has been carried out. The results of this comparison have been presented in a technical report and in a CSCS seminar.

This study presented a platform for the choice between Linda and PVM in the frame of the molecular dynamics Ph.D. programme that Slivnik started at the J. Stefan Institute in Ljubljana.

OPTIMIZATION OF THE PROGRAM "COMPUTATION OF UNSTEADY THREE–DIMENSIONAL FLOWS IN COMPLEX GEOMETRIES"

Involved Persons:
M. F. Hodous (a)
P. Pagny (b)
a) CSCS
b) EPITA (France)

Funding Sources:
CSCS
NEC (Japan) grant

Duration:
October 1993–December 1993

The program "Computation of Unsteady Three–Dimensional Flows in Complex Geometries" computes complex geometry flows in the late transitional regimes, using direct numerical solution of the time–dependent Navier–Stockes equations. Since it runs too slowly on the SX–3, optimization would be wished.

The code was vectorized by writing everything in terms of calls to vector processing routines such as BLAS. While this does provide vectorization, the subroutine call overhead becomes a burden unless each routine performs a great deal of computation, which the older BLAS routines do not always do.

Automatic inlining would eliminate this overhead. However, automatic inlining sometimes is difficult to accomplish in large codes where the entire code is broken into one file per routine, as is typical Unix practice, especially when using the make utility.

With the introduction of software release 3.1 on the SX–3, the Fortran compiler system now includes many array syntax features from Fortran 90. BLAS–like operations can be written as array operations, not as subroutine calls. Assuming a good degree of loop fusion, which the compiler already does, array operations lead to inline code with good usage of vector data registers to hold intermediate results.

As an example, a SAXPY call can be replaced by

```
real a,X(n),Y(n)
     ...
X = X + a * Y
```

The nested DO–loops

```
real a,X(0:i2,0:j2,0:k2,
\Y(0:i2,0:j2,0:k2),Z(0:i2,0:j2,0:k2
     ...
do 100 i=1,i2
 do 100 j=1,j2
  do 100 k=1,k2
100 X(i,j,k) = Y(i-1,j,k) +
\a*Z(i-1,j,k)
```

become

```
X(1:i2,1:j2,1:k2) =
\Y(0:i2-1,1:j2,1:k2) +
\a*Z(0:i2-1,1:j2,1:k2)
```

In theory, at least, the compiler should generate much better code from array–oriented syntax than from subroutine calls that must be inlined or from DO–loops that may be nested in the wrong order.

ANALYSIS OF SELECTED NAS BENCHMARK KERNELS

Involved Persons:
G. Jost (a)
U. Kühn (b)
R. Rühl (c)
W. Sawyer (c)
a) NEC–SX Center (Switzerland)
b) Universität Münster
c) CSCS

Funding Sources:
NEC (Germany)

Duration:
August 1993–October 1993

For the National Aerodynamic Simulation Program (NAS) a set of benchmarks has been developed, especially designed to evaluate the performance of parallel supercomputers, for which the ordinary benchmarks are not appropriate. One of these benchmarks has been implemented on a NEC Cenju–2 parallel system to evaluate its potential computing power. Due to the structure of these benchmarks, this result is not a theoretical upper speed bound, but it shows a potential for "real–world applications".

REFERENCES:

[1] Bailey, D. J. Barton, T. Lasinski and H. Simon. "The NAS Parallel Benchmarks." NASA Technical Memorandum 103863, NASA Ames Research Center, Moffet Field, California, 1993.
[2] Briggs, W. L. "A Multigrid Tutorial." Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1987.

PROJECT ABSTRACTS

COLOR CORRECTION AND GAMUT MAPPING AT CSCS

Involved Persons:
U. Meyer (a)
I. Pontiggia (b)
a) CSCS
b) Université de Neuchâtel

Funding Sources:
CSCS
Université de Neuchâtel

Duration:
November 1992–June 1993

High–end printing systems as they exist today are able to produce high quality images with respect to spatial and color resolution. However, the reproduction of the right color as seen on a workstation monitor or a film is a difficult task. Color ranges (gamuts) might vary substantially among the mentioned devices. The main goal for this project was to be able to print images with colors matching the originals on a workstation monitor as close as possible.

Before and after the correction

The result of this project is a color correction system. It is flexible by allowing to load device characteristic data sets to drive the color correction process. Thus, it is adaptable to different kinds of monitors, scanners, slide recorders, or printers provided that adequate measurement tools are available.

The color correction process is split in several stages. The first stage converts the device colors into the standard CIE XYZ color space depending on the device specific color table. The second stage maps XYZ to XYZ performing operations such as grey axis alignment and gamut mapping. The third and last stage converts XYZ colors back to device colors, which may be RGB or CMYK colors. Gamut mapping is a crucial part in the process. A gamut of colors is the range of (re–)producible colors of a device. Typically, printer gamuts are much smaller that monitor gamuts. Thus, some monitor colors cannot be reproduced at all by the printer. The gamut mapping assures that all colors will be within the gamut of the printer.

This work was carried out to fulfill the requirements for a diploma in computer science at the University of Neuchatel. The theme was Color Correction for a Color Laser Printer.

MAINTAINING A LARGE NUMBER OF AVS MODULES

Involved Persons:
F. Banfi (a)
U. Meyer (b)
a) Université de Neuchâtel
b) CSCS

Funding Sources:
CSCS
NEC (Japan) grant

Duration:
July 1993–August 1993

The Application Visualization System (AVS) is an advanced module based package for scientific visualization. It supports writing of user modules and allows for collecting them in a library. An AVS module library can consist of several hundreds of modules, such as those from the International AVS Center (IAC). The modules are delivered with makefiles to be compiled, but nothing is given for their installation nor for compiling a whole tree of modules. This can be quite painful if they must be installed on different platforms, since everything must be done by hand. In this project solution are presented that use *imake* and Imakefiles. This leads to an environment that allows easily to compile and install a large number of modules at once. Support for changing to another platform is included. As a nice side–effect, none of the files of the original IAC distribution need to be changed.

This work has been implemented by F. Banfi as part for the fulfillment of the requirements for a degree in computer science of the University of Neuchatel.

COMPUTATION OF ELECTRIC FIELD AND SPACE CHARGE

Involved Persons:
W. Egli (a)
E. Gerteisen (b)
R. Gruber (b, c)
a) ABB
b) CSCS
c) SIC–EPFL

Funding Sources:
ABB

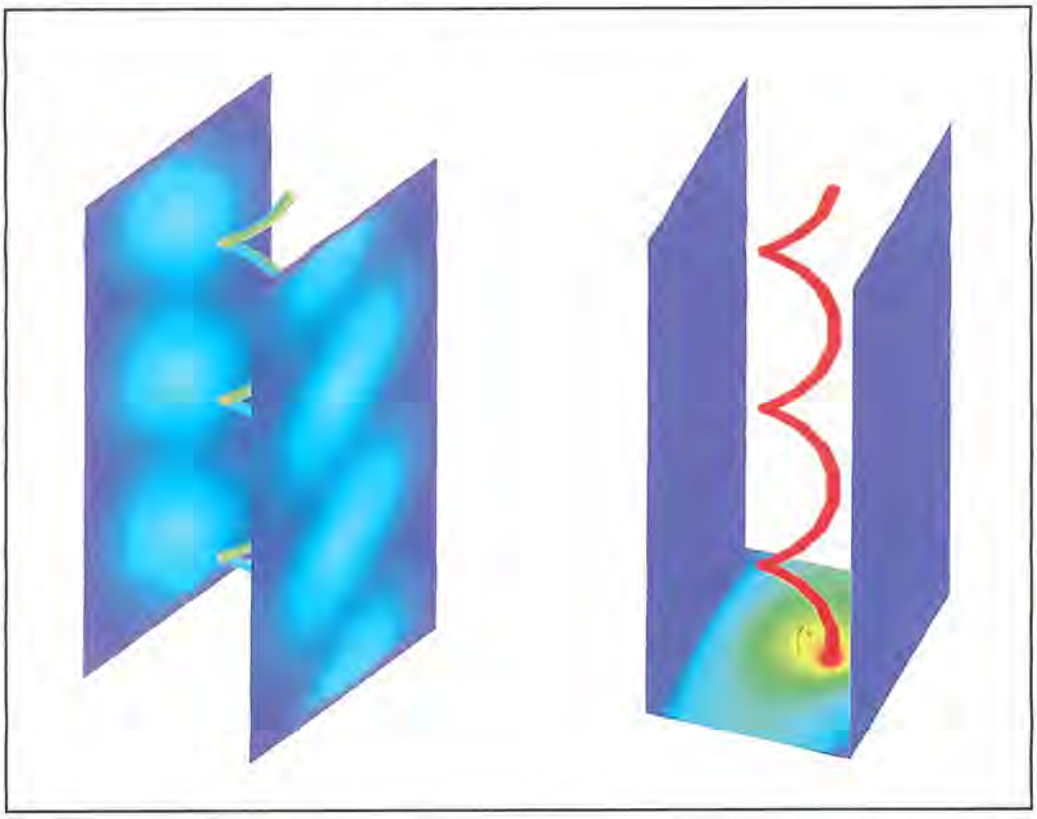
Duration:
October 1993–December 1995

Polluting particles in a gas flow can be eliminated by ionising them, driving them by an electric field to the conducting wall put on earth where they are neutralised and deposited. The ionized moving particles develop a space charge that modifies the electric field.

In the work realized up to now, the self consistent solution of the electric field affected by the space charge due to the particle flow has been computed in idealised 2–D and 3–D geometries.

In a two–dimensional model geometry, the space charge originating from the particle flow has been computed by three different methods that are the direct solution of the Maxwell's equation, the computation of the charge by a particle pushing method and the forward integration of a particle distribution function by a characteristics method. All three methods give the same result, the third one being the most precise one.

This characteristic method is used for the 3–D geometry.



Distribution of the current on the grounded plate (left) and of the electric potential between the grounded plates (right)

NUMERICAL SIMULATION OF FERROFLUIDS FOR PERFORMANCE COMPARISONS BETWEEN MULTI–SPIN–CODING AND DIRECT–CODING ON THE CM–2 AND THE CENJU–2

Involved Persons:
Y. Baggi (a)
B. Chopard (a)
a) Université de Genève

Funding Sources:
NEC (Germany)

Duration:
August 1993–October 1993

This work presents a "Real–World" application used as a benchmark of a different kind than MFLOPs and MIPS.

The measurements we present are SUPS (Site Updates Per Second). This performance units are not calculated theoretically, and therefore "Peak Performance" can not be achieved using these units. Our benchmarks show the performance of two parallel computers which are the CM–2 (Thinking Machines Corporation) and the experimental machine Cenju–2 (NEC). Both machines run two implementations of a cellular automata model called FHP, which simulates the behavior of a gas.

REFERENCES:

[1] Matsushita, S., T. Yamauchi, T. Nakata and N. Koike. "The Architecture of the *NEC Cenju–2* Parallel System." *NEC SX World* (Spring '93).

PORTING LINDA TO THE NEC CENJU–2 PARALLEL COMPUTER

Involved Persons:
I. Pontiggia (a)
R. Rehmann (b)
R. Rühl (b)
a) Université de Neuchâtel
b) CSCS

Funding Sources:
NEC (Germany)

Duration:
August 1993–October 1993

The aim of this project was to port a Linda system to the NEC Cenju–2 DMPP. The Linda system is based on the MPI parallel programming environment, with one processor reserved for the tuple space server, the others shared between the master process and the slave processes.

Linda is based on the idea of a global memory space called *tuple space*, in which you can put, read or remove elements called *tuples*. A *tuple* is an ordered list of elements of any kind and size. These elements can be *actual* (i.e., fully specified) or *formal* (i.e., only the type is specified). The tuple space is accessed using four functions, two for reading tuples, one for writing tuples and one for executing new processes.

The system is intended to be a library completely surrounding the user's Linda program. This surrounding is needed because some functionality (i.e., provided by the *tuple space server*) needs to manage the user's program, and therefore needs to control it. Some other functionality of the system has to be provided (e.g., the Linda calls) to the user's program, and so has to be placed under the programs control.

Everything that has to do with the tuple space passes through the tuple space server and everything that does not involve the tuple space is kept local to the appropriate process. All the interactions of a process with the tuple space are carried out using linda calls (i.e., `in()`, `rd()`, `out()` and `eval()`). That means that all the Linda calls, and only the Linda calls, contain messages (to and from the tuple space server).

`in()` and `rd()` are the input functions, they differ because `in()` removes the tuple found, whereas `rd()` does not. The output functions are `out()` and `eval()`. The difference here is functional, the `out()` call simply puts a tuple in the tuple space, a call to the `eval()` function creates a new process for every function call present in the tuple. The input functions are synchronous, this means that the process that calls `in()` will wait until a matching tuple is found (or `out()`'ed if it is not actually in tuple space). The `out()` and `eval()` functions return immediately, there is no waiting for an answer.

The tuple space is implemented as two stacks, one collecting the tuples that have been put into the tuple space with a call to `out()`, the other storing the unsatisfied (and thus awaiting) `in()` and `rd()` calls. Tuples are memorized in stacks for implementation simplicity reasons, but if necessary (for performance reasons) they could be stored in a more efficient way (maybe using some hashing function based on information about usage distribution of the tuples).

The `outStack` (the stack where `out()`'ed tuples are stored) is equivalent to the tuple space. In fact the tuples present in tuple space are tuples that have been sooner or later `out()`'ed, but have not yet been removed (`in()`'ed).

REFERENCES:

[1] Ahuja, S., N. Carriero and D. Gelernter. "Linda and friends." *Computer* 19(8):26–34 (August 1986).

DEBUGGING SUPPORT ON THE CENJU–2

Involved Persons:
J. Blandy (a)
C. Clémençon (b)
R. Rühl (b)
a) Oberlin College (USA)
b) CSCS

Funding Sources:
NEC (Germany)

Duration:
August 1993–October 1993

We have provided source–level debugging for C programs running on NEC's Cenju–2 multiprocessor architecture, by porting the GNU Debugger. It is our hope that this debugger will be used as an interface between the Cenju–2 and a debugger offering a higher–level view of the multi–processing program. As an intermediate step, we also ported the GNU Debugger to the EWS operating system, used by the Cenju–2's front–end machine. In order to provide an acceptable combination of assembly language, debugging annotation format, and object file format, we have also ported the GNU Assembler to the Cenju–2, and made minor modifications to the GNU C Compiler.

REFERENCES:

[1] Stallman, R. M. and R. H. Pesch. *Debugging with GDB*. 4.09 edition Boston: Free Software Foundation, Inc. (April 1993).

Involved Persons:

J. Dvorak (a)
B. O'Sullivan (b)
R. Rehmann (a)

a) CSCS

b) Trinity College (Ireland)

Funding Sources:

NEC (Germany)

Duration:

August 1993–October 1993

GRAPHICAL EDITOR FOR STENCIL BASED PROBLEMS

Writing stencil–based scientific calculation programs can require a significant effort, and modern parallel architectures add further complexity. The graphical editor for stencil based problems *Snake* makes up a part of a larger system which attempts to tackle both stencil specification and data distribution at the same time.

Stencil–based algorithms arise from *finite difference* (FD) methods, which are frequently used in the field of numerically–intensive computing. The *Snake* graphical editor allows a user to specify the sort of stencil to be used in the solution of a particular computational problem; it also provides graphical control over the distribution of a data set across the nodes of a parallel machine.

Snake produces a subset of the SPSL [1] specification language as its output; this output is processed by an expert system which decides upon the particular algorithms to use to solve the problem at hand.

REFERENCES:

[1] Roth, M. E. *A Stencil Specification Language for the Generation of Communication Skeletons.* Master Thesis, Institute for Applied Mathematics and Computer Science, University of Berne, Switzerland, May 1993.

[2] Decker, K. M. and R. M. Rehmann. *Simple and Efficient Programming of Parallel Distributed Systems for Computational Scientists.* Technical Report, Institute for Applied Mathematics and Computer Science, University of Berne, Switzerland, IAM 92–019, October 1992.

PROJECT ABSTRACTS

Involved Persons:

R. Holzner (a)
A. Mangili (b)

a) Quantum Optics Group–Universität Zürich–Irchel

b) CSCS

Funding Sources:

CSCS
Universität Zürich

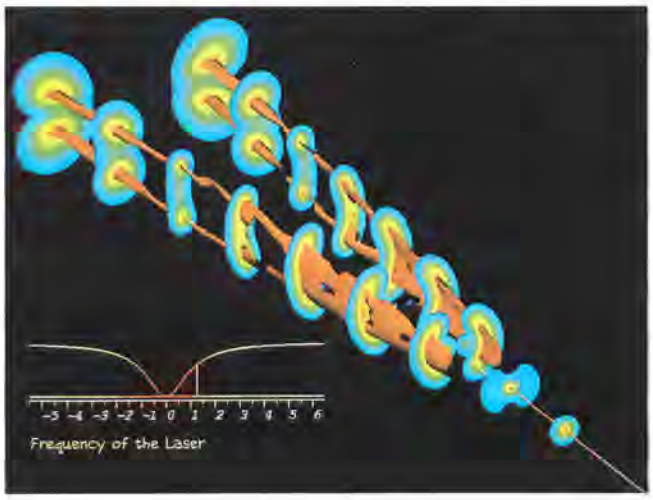
Duration:

February 1993–Continuing

VISUALIZATION OF LASER SIMULATIONS RESULTS

Although quantum mechanics provides in principle precise description, the interaction of light and atoms is fairly complex in detail and a theoretical approach always involves approximations. This is the reason why even one of the most fundamental manifestations of light–matter interaction, the propagation of a laser beam through atomic vapor, is not yet completely understood. The experimental verification of the most advanced numerical models describing this phenomena has been one of the main goals of this group.

One of this simulations is concerned with nonlinear interaction of two polarized laser beams inside sodium vapor. The model predicts "switching" and "bouncing" when two circular polarized laser beams intersect in a nonlinear medium. Theoretical models are confronted with precise experimentally observed phenomena in order to improve the numerical model.



Several video sequences showing the effects of varying the frequency of the light, and the effect of varying the intensity of the lasers were produced frame by frame using the visualization software AVS. For this kind of simulations, the video animation was an important way to understand the correlation between the deviation and the frequency of the two laser beams. Other video animations of new results are foreseen in the near future.

PROJECT ABSTRACTS

Involved Persons:
G. C. Corti (a)
M. Gay (a)
N. Pedrozzi (a)
a) CSCS

Funding Sources:
CSCS
Ufficio del Lavoro, Bellinzona

Duration:
April 1993–continuing

PROJECT, USER AND RESOURCE ADMINISTRATION PROJECT (PURAdm)

Determining resource allocation and usage is a critical aspect of managing computer resources. Statistical information regarding resource exploitation is necessary not only for accounting and billing purposes, but also for medium– to long–range resource/facility planning. This is especially important in a sophisticated, high–performance, multi–platform, multi–user computing environment.

The "Project, User and Resource Administration" project (PURAdm) is CSCS's solution to managing it's computational resources. It's primary functional goals were to

- improve the existing resource accounting/ monitoring system
- restructure the user and project administration
- improve the flow and quality of administrative information
- improve the support for trivial computer management tasks.
- restructure the user and project administration
- improve the flow and quality of administrative information
- improve the support for trivial computer management tasks.

Motivating design principles and goals included:

- modularity in program design
- expandability/extensibility (the system has to support the existing supercomputer as well as future high–performance equipment)
- flexibility in data management (use of a relational database management system) to accommodate future requests for statistical reports
- user friendliness (use of a graphical user interface)

PURAdm is currently in beta–testing and will be delivered in April 1994. The project is foreseen to be "open–ended" to allow for improvements and modifications based on user suggestions and changes in the CSCS resource environment.

This project was partially realized as an extension of the retraining course for unemployed programmers.

Involved Persons:
R. Pedetti (a)
M. Tomassini (b)
a) Passera & Pedretti
b) CSCS

Funding Sources:
Passera & Pedretti

Duration Dates:
September 1993–Continuing

COLLABORATION WITH PASSERA & PEDRETTI SA, STUDIO D'INGEGNERIA CIVILE

There are two main lines in this collaboration. The first line consists in giving to the ticinese engineering firm Passera & Pedretti the possibility of using the sophisticated structural analysis package NASTRAN on powerful workstation servers. This will make it possible to treat bigger problems in less time, problems that were beyond the reach of the firm equipment until now due to memory and cpu limitations.

The package will also be made available for free to the academic world and we expect other engineering firms to be interested. Passera & Pedretti will also use NASTRAN for third–party work as needed.

The second collaboration line has to do with realistic 3–D graphic representation of landscapes with MOSS.

CSCS powerful graphics hardware will jointly be used by Passera & Pedretti to produce 3–D images of landscapes and artifacts in order to be able to easily show the effect of buildings, roads and railways to policy makers.

SIMULATION OF BIRD–IMPACT ON HELICOPTERS

Involved Persons:
J. Y. Chiu (a, b)
S. Focardi (c)
G. Pagnano (d)
S. Poggi (d)
K. Schweizerhof (e)
a) CSCS
b) NEC
c) The Intertek Group
d) Agusta SpA (Italy)
e) CAD–FEM (Germany)

Funding Sources:
Agusta SpA (Italy)
CAD–FEM (Germany)
NEC (Germany)

Duration:
September 1993–December 1993

Agusta is a major manufacturer of helicopters; located in northern of Italy near the Malpensa airport, the region has a long–standing tradition in the aeronautics industry. The company designs and manufactures high–performance, medium–sized helicopters which are widely used in a variety of tasks and often in meteorologically hostile conditions, such as the Alpine rescue service. Agusta has decided to introduce crash–analysis and testing in designing of the next generation of helicopters to enhance their safety characteristics.

Although aeronautics regulatory agencies are not yet imposing tests on crash worthiness in their certification procedures, it is expected that they will move in this direction soon. In addition, the market is increasingly attentive to such passive safety features, which give manufacturers an important competitive advantage.

Passive safety considerations are becoming increasingly important in aircraft design. This implies structural design modifications that can withstand or minimize the danger or damage of a crash situation. While it is clearly impossible to rely on structural resistance features in the event of a major crash, attention is being focused on those situations that can be controlled and thereby minimize the danger and potential harm.

In particular, helicopters can be designed with improved safety features that help confront specific, potentially dangerous situations such as the impact with birds at cruising speeds or the impact due to an difficult landing. Both cases involve structural stresses that must either be supported by the helicopter's body without damage or produce limited damages so as not to put the passengers' or crew's life in danger.

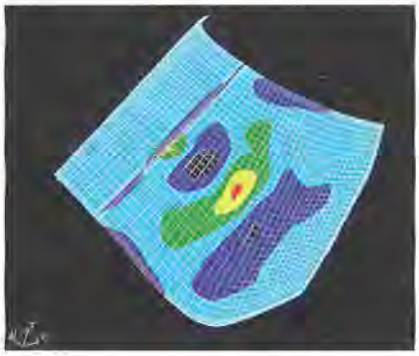
There are two basic ways of testing the crash–worthiness of mechanical structures: through physical experiments and through simulations. The first way is to conduct actual physical experiments that simulate the impact conditions likely to be found in real operations. This is a very costly procedure, as it involves building and destructing expensive prototypes. In addition, because only a limited number of tests can be conducted, this method severely limits safety optimization.

A more modern approach is simulating crash and impacts through numerical analysis. This is a computationally–intensive method, feasible only in the last few years, thanks to increases in computational speed and the reduction in cost of supercomputers.

There are many computational challenges in crash analysis simulation: from the accurate description of large, non–elastic deformations of a wide variety of materials and structural shapes, to the handling of contacts, friction and ruptures of the surfaces that are deformed during the crash event.

Numerical crash simulation was first developed by Dr. John Halqvist at the Lawrence Livermore National Laboratories; he initially developed the Dyna 3–D code to study the impact of bullets and other weapons on targets. The limited computational resources then available and their high cost initially limited the use of crash analysis to military applications. This situation changed in the mid–eighties with the diffusion of vector supercomputers and visualization workstations, and the automotive industry began extensively using crash simulation to reduce the escalating costs of crash certification. By this time Dyna 3–D had expanded features and addressed a range of applications, making it capable of handling a wide variety of shapes and materials as well as crash conditions. The development of powerful visualization tools to aid in data analysis interpretation was instrumental in fostering the adoption of numerical crash simulation technology.

In the automotive industry, crash analysis can make accurate, com-



Pressure contour on windshield at 3.0 ms

plete, and realistic simulations of car crashes; the complete safety optimization cycle can be conducted without physical testing. The use of crash analysis in helicopter design is new, and thus experience can be gained only through some pioneering effort, extrapolating results from other sectors.

As with most aerospace companies, Agusta has a wide and consolidated experience in the use of structural analysis. Design teams are conversant with this technology, and the traditional software and hardware infrastructure to produce meshes and analyze results is firmly in place. Agusta recently decided to begin pioneering work on numerical crash analysis. However, the computational resources required are orders of magnitude larger than those required by standard structural analysis techniques. In addition, design methodology using numerical crash analysis is considerably different.

Although Agusta was interested in developing the internal skills and methodologies for impact and crash analysis, the company wanted to be free of the impediments of the computational and software resources. The Industry Partnership Program of CSCS, the Swiss Scientific Computing Center, provided the answer and the resources. Through CSCS, Agusta performs the needed realistic and reliable simulation while developing its internal know–how without resource constraints.

Equipped with a NEC SX–3 capable of a peak speed of 12.8 Gflops and corresponding visualization and communication equipment, CSCS offers the computational infrastructure to perform crash and impact analysis with the necessary accuracy and with sustainable turnaround time short enough to allow for validation and design optimization. CSCS was also able to offer Dyna 3–D in its commercial form: LS Dyna 3–D[1]

The first simulation problem Agusta chose was the bird–impact problem, i.e., to study the consequences of the impact of a bird on the canopy of a A–109 helicopter at a cruising speed of 76 m/sec. Agusta provided CSCS with a Nastran mesh describing the canopy and developed a special local mesh to describe the joints between the windscreen and the helicopter's body. In addition, material characteristic where provided and a suitable representation of the bird was chosen.

Several iterations were required to adjust a number of parameters before a satisfactory result could be obtained. In total, running the first complete test on the bird impact required more than ten CPU hours on the NEC SX–3. This shows the importance of computing power for crash analysis, especially in the first phase. In fact, while it is expected that after gaining experience it is possible to reduce considerably the computational load, in the initial phase computing power is essential to validate and tune the application in a reasonable time. If the same work had been done on a workstation, it would have required hundreds of CPU hours, thus making the development of a impact analysis methodology a difficult, if not impossible, task.

Results obtained so far have shown the ability of LS Dyna 3–D to replace physical experiments in helicopter bird impact. The analysis is perfectly realistic and supplies a high level of detail for design optimization as shown in the image.

How does one arrive at an impact and crash analysis methodology? The next step should be a more extensive validation of crash simulation with some experimental comparison. This would yield additional insight into modelling procedures related to helicopter–type structures and to reduce the computational load through appropriate simplifications.

It is, however, important to establish a safety optimization procedure integrated into the global helicopter design and engineering process. While in the present test phase the bird impact analysis is used as a quicker and more cost effective replacement of impact experiments with optimization possibilities constrained to minor engineering changes, future development could offer many more benefits, among them:

- impact and crash considerations included in early–design phases of future generations of helicopters;
- exploration of a much larger set of safe designs;
- quick evaluation of new and different materials;
- easier interaction with marketing, with the eventual possibility of offering scalable safety features for special versions;
- early consideration of manufacturing constraints;
- improvement of important competitive features (such as weight and shape optimization) without sacrificing safety.

To attain these benefits it will be necessary to work on two lines of development: methodology of analysis and integration features.

[1] LS Dyna 3–D is produced and supported by Livermore Software, founded by Dr. Halqvist to pursue industrial exploitation of crash analysis technologies.

Involved Persons:

P. Flükiger (a)
R. Gruber (a)
M. Hohenadel (a)
U. Meyer (a)
I. Pontiggia (a)

a) CSCS

Involved Institutes:

CERFACS (France)
University of Manchester (UK)
LABEIN (Spain)
Université de Franche–Comté (France)
Università degli Studi di Milano (Italy)
RAL (UK)
CRIL–Inforop (France)
JRC (Italy)

PROJECTS

Funding Sources:

BBW (HC&M program)
CSCS
European Union (other partners)

Duration:

September 1993–September 1994

HUMAN CAPITAL & MOBILITY PROGRAM: INTERACTIVE IMAGE PROCESSING AND SYNTHESIS ON INNOVATIVE COMPUTER ARCHITECTURES

The European Union Program "Human Capital & Mobility" (HC&M) promotes the flow of knowledge and experience among research centers by exchanging researchers for the purpose of collaborating in research activities. It is organized in so–called networks, whereof one is "Interactive Image Synthesis and Processing on Innovative Computer Architectures".

This network aims to develop a common understanding of the problems of image data analysis and manipulation, to exchange software tools and systems, to exchange expertise, and to experiment with each other's approach and methods on different hardware platforms (and particularly parallel computers) and in different application areas. In the medium term it is expected that this allows to identify and develop, in collaboration with each other, common tools and environments which can be used in a wide variety of applications. In the proposed consortium, these include computational fluid dynamics, satellite and medical image processing, image data compression, realistic image synthesis, visualizing complex 3–D data (CAD), molecular modeling and graphics, synthesis of holograms.

CSCS will work in the following areas: image compression, scientific visualization and graphical user interfaces, and in the development of a image processing and synthesis algorithms library.

The image compression subpart will lead to enhanced image compression that will be applied to various data provided by other partners and be tested for remote interactive visualization. Algorithms that are similar to JPEG in their structure but use other fast transforms such as the wavelet transform will be developed. Compression factors of over 30 are expected with extremely low loss of quality. In a further step the algorithms will be parallelized and ported to various parallel systems.

In the scientific visualization part graphical user interfaces for scientific visualization will be enhanced or developed to make the scientists task easier with special focus on the improvement of the user–friendliness. If large data amounts have to be processed and analyzed, general purpose packages exhibit performance and space limitation problems. As a consequence, specialized packages, each with a different user–interface, have to be used. It is a goal to approach versus a common interface for visualization packages, at least for within the same application area. Another focus is on input generation tools that serve as interfaces to important simulation software packages. The same interface that is used to visualize simulation results should be enhanced with the appropriate functionality to setup simulation parameters, launch the calculation and retrieve the results. Specific application areas are molecular graphics, CFD, solid state physics, and engineering.

Finally, the image processing and synthesis algorithms library is intended to provide an environment on innovative computer architectures to implement or port the algorithms developed in the other parts of the project and by other partners in the network. Problems such as reading simulation data and writing images in parallel as well as parallelizing the fast transforms used in the image compression algorithms will be addressed. Other domains we expect to deal with are volume visualization and color correction.

The following partnerships have been established:

Image compression:
- CSCS
- Università degli Studi di Milano
- RAL
- CRIL–Inforop
- JRC

Scientific Visualization:
- CSCS
- CERFACS
- University of Manchester
- LABEIN
- Rutherford Appleton Laboratory

Algorithms library:
- CSCS
- University of Manchester
- Università degli Studi di Milano
- JRC

Involved Persons:
K. M. Decker (a)
J. J. Dvorak (a)
R. M. Rehmann (a)
a) CSCS

Funding Sources:
SNF, Priority Program Informatics

Duration:
March 1993–October 1994

RESEARCH ON INTELLIGENT PROGRAMMING ENVIRONMENTS FOR MASSIVELY PARALLEL SYSTEMS

1. INTRODUCTION

The crucial factor limiting the usage of parallel systems with distributed memory is their difficult programmability. This is particular true for application users such as scientific programmers. As compared to conventional high–performance computer systems, distributed memory systems burden the user with a variety of difficult tasks such as management of the distributed address space, handling of several threads of control, balancing the communications vs. the computational load on each processor, and load balancing across processors. These responsibilities are not known from conventional systems with shared memory.

First generation tools and fully integrated tool environments have been proposed and developed over the last few years in research labs and by vendors of parallel systems with many different flavors (for a review see [1]). These developments present a many man–years effort in software development. However, the common goal, to make programming of distributed memory parallel processor systems as simple as programming of conventional single–processor high–performance systems or multi–processor systems with shared memory could not yet be achieved. It is the purpose of this project to critically analyze, from the application user point of view, the reasons for this failure and to propose a viable approach to solve the problem.

2. CRITICISM OF EXISTING TOOLS AND TOOL ENVIRONMENTS

Existing first generation tools and tool environments for parallel systems can be criticized in several ways. Most notably, their development was not driven by (application) user requirements. Instead, they were developed by computer scientists who thought that they *knew* what scientific programmers would need. As a result, the tools were too complicated to use for non–computer scientists or non–specialists in parallel computing. Those users who managed to become familiar with the tools could not provide feedback of their experiences collected with the tools to the tool developers since a feedback mechanism was not anticipated.

Today's tools are in general not portable, i.e., different parallel systems require different tools. When several different systems are in use simultaneously, a high learning barrier has to be overcome by the users, and protection of software development investments cannot be ensured. Today's tools are also not scalable. Since they basically focus on low–level aspects such as size and frequency of messages exchanged between processors, they do not scale to hundreds or thousands of processing elements and become useless and even not applicable when the number of processors grows beyond a certain limit. Finally, current days tools often focus on representation techniques of low–level facts rather than on modeling techniques for programs or entire applications and the methods underlying these modeling techniques.

To remedy this situation, we believe that first the foundations of tools and tool environments need to be carefully reinvestigated. Then, in a second phase, new and more powerful software tools need to be developed according to a user– and application–driven development methodology with well–established feedback mechanisms from the tools' user community.

3. USER'S PROFILE AND REQUIREMENTS

Our target users are scientific programmers, or in more general terms, computational scientists which have in general only low to medium knowledge of basic concepts of computer science, and don't want to get involved into technical details of parallel distributed programming or parallel distributed systems. However, if this is required by the problem to be solved, they are ready and capable to literally do everything required to get their scientific problem solved.

Scientific programmers have a variety of requirements, but ease of use has highest priority for them. Ideally, they want to continue to use their own terminology which they also use to formulate their scientific problems. Ultimately, they would like to have transparent programming of distributed memory systems. They want intensive guidance in familiar vocabulary, close to their problem domain, provided in the entire application development cycle. Achieving highest performance and best parallel efficiency must be possible *and* supported by the environment. Portability across various parallel systems, preserving the program's efficiency, at least within the class of distributed memory parallel processor systems, is a must. Software development investments must be protected by supporting reuse of software components. The supporting environment must be adjustable to their often quickly changing needs and it must be extensible. Problem–solving speed has highest priority to scien-

tific programmers. For these users, speed can justify any restriction, effort or requirement.

4. GOAL AND OBJECTIVES

4.1 Goal

To qualitatively enhance the programmability of parallel distributed systems, it is our long–term research goal to build a fully fledged, easy–to–use problem–solving environment for parallel distributed programming that ensures all user requirements discussed in this paper.

4.2 Design Objectives

In a user– and application–driven approach to tool development, the design objectives are derived from the user's profile and their requirements.

Ease of use, the requirement of highest priority to scientific programmers, can be realized if a problem–oriented specification formalism is included in the design objectives. Since the application fields to be mastered by the environment may differ considerably, several different domain–specific problem–oriented specification formalisms and corresponding user interfaces must be provided, not *one* universal formalism or interface. This also ensures that the specification formalism is close to the requirements in the respective problem domain. Expressive and specifically designed formalisms, describing the nature and the features of the problem under consideration as precisely and as completely as possible, also ensure the largest possible problem–solving speed.

Intensive guidance through the entire application development path, protection of software development investments by means of software reuse, adaptability and extensibility to changing user needs, and largest possible problem–solving speed is ensured, if the knowledge of parallel programming can be exploited by the scientific programmer as completely as possible. These requirements can be achieved, if the programming environment is designed as a knowledge–based system.

Ease of use on the one hand, but readiness to get deeply involved if required by the user's application on the other hand, can be best guaranteed, if the tool environment is hierarchically designed in a multi–layer structure. High–level and sophisticated tools support inexperienced users, but cannot ensure highest parallel efficiency. Medium–level and advanced tools support less experienced users. They are aiming at highest parallel efficiency. Low–level, plain tools require large user expertise, but can guarantee highest parallel efficiency.

Highest performance and best possible parallel efficiency are ensured and supported, if the environment is designed to supplement and not substitute human capabilities.

To ensure that users can handle the multi–layer tool environment and correctly adapt and extend the tool environment, one of the design objectives should be an integrated training and tutoring component.

Portability across various parallel systems, preserving the program's efficiency, and thus contributing to the protection of software development efforts, requires sufficient abstraction from the hardware as a design issue.

Protection of software development efforts implies that the system must provide software reuse mechanisms.

Highest performance, protection of software development efforts, adjustability and extensibility can be guaranteed, if scalability of the methods underlying the tool environment is included in the design issues.

Highest performance for a broad spectrum of programs and applications requires that several models of parallel programs must be supported.

Finally, to have a clean software development procedure and to support software reuse as fast as feasible, the tools in the environment should support a parallel distributed programming methodology.

5. OVERVIEW: A KNOWLEDGE–BASED PARALLEL PROGRAMMING ENVIRONMENT

5.1 Programming Methodology

To realize the stated objectives, a programming environment with user support starting much earlier in the program development path seems most appropriate [2]. The *Program Development Environment* (PDE) to be discussed subsequently covers important parts of the complete program development path from problem specification and design up to code generation. Emphasis is put on user interactions at a high level of abstraction, well above the level of standard high–level programming languages.

The programming methodology supported by PDE consists of the following three steps:

- Problem–oriented specification using a formal language
- Interactive refinement and completion of the specification
- User–transparent generation of compilable program code

From the scientific programmer's point of view, the problem–oriented specification language formalism is probably the most important ingredient of the methodology. It allows specification of the problem in a problem–oriented way, using a terminology which is derived from the terms used in the scientist's problem domain, avoiding the jargon of computer science to a large extent. This strategy ensures that the programming methodology is readily manageable by scientific users.

5.2 System Overview and Operation

The realization of the above programming methodology by means of three functional components is depicted in Figure 1.

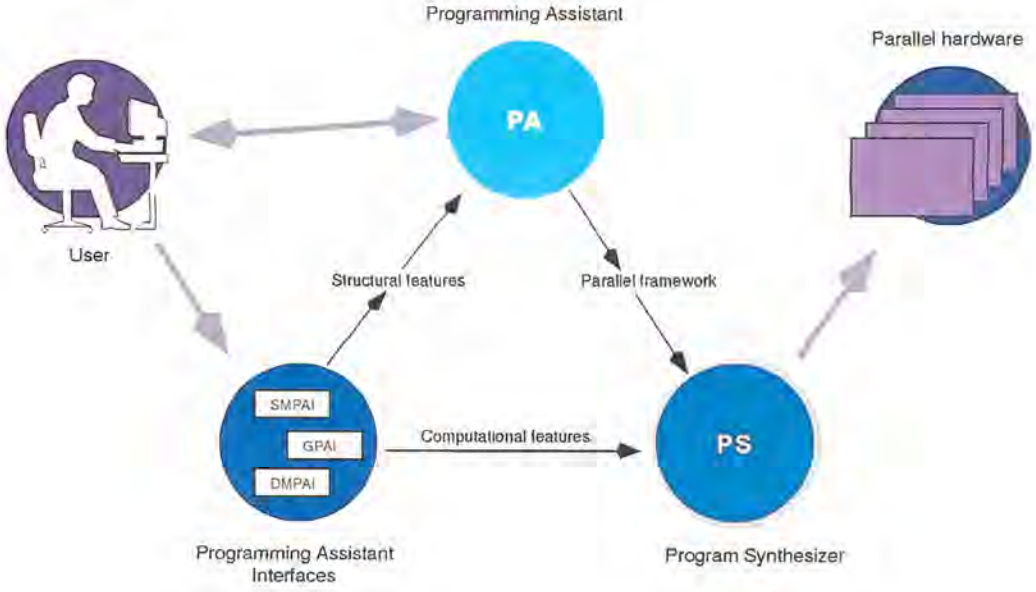


Figure 1: The conceptual structure of the program development environment with programming assistant interfaces, programming assistant, and program synthesizer.

In a typical session, the programmer gives an initial specification of the problem under consideration using one of the *Programming Assistant Interfaces*. This specification is then decomposed into the purely computational features and the features relevant for the parallel structure. The first are passed directly to the *Program Synthesizer* (PS), the latter go to the *Programming Assistant* (PA). Then, in conjunction with the user, the PA extracts and completes the information needed to select an appropriate parallel framework. Functions and procedures that may be needed within the selected framework are requested from the user. The PA is the central, largely AI–based component of the PDE, relying on various kinds of expert knowledge. The parallel framework of the application and other important information are passed to the PS. The PS combines the computational features with the data received from the programming assistant into compilable, hardware specific, parallel C++ or C programs.

5.2 Restrictions on Problem Domain

To implement and test our parallel programming concept, we have chosen the domain of stencil–based computations for the first programming interface. A *stencil* is a local computation prescription that is usually iteratively executed for all members (*cells*) of multidimensional *grids*. On the one hand, this class of applications is of particular importance in the HPC area and it covers applications from many scientific disciplines. Examples are image restoration and analysis, computational fluid dynamics and a large class of simulations. On the other hand, the domain is sufficiently restricted and regular to serve as an ideal initial testbed for the development of the programming environment. It can also be expected that by relaxing the restrictions and

extending the specification formalism used in the programming interface for stencil–based problems, a development path to an interface covering more and more general data–parallel applications exists.

5.3 The Notion of a Skeleton

Our programming methodology relies fundamentally on a large collection of *algorithmic skeletons* as the central knowledge base. A parallel distributed program in the program development environment is thus modeled as:

1. an algorithmic skeleton serving as an integrating framework
2. a number of sequential computational components

The algorithmic skeleton separates purely computational aspects of a program from all the complex aspects of parallelism.

The basic idea with a skeleton–based approach is to encode reusable structural characteristics of algorithms in skeletons. A skeleton typically contains open (*generic*) parts that have to be filled in by the user in order to adapt the skeleton to the given situation and to get a complete algorithm or algorithm part.

Our knowledge base contains a large number of very specific skeletons offering good parallel efficiency. AI techniques are used to map applications onto the skeleton knowledge base and to guide the programmer through the software development process from the initial problem specification up to complete code optimized for a given hardware platform. According to the reasoning mechanism used, the skeleton collection is organized as a tree–like hierarchy with unspecific skeletons at the top and most–specific skeletons at the leaf–level.

6. STATUS

Two prototypes of the program development environment PDE have been realized [3]. Both prototypes are running and present different aspects of the programming environment. The two prototypes approach the system development task by embodying top–down and bottom–up strategies, respectively.

The first prototype serves primarily as a runnable specification of the PDE and contains a first attempt at a skeleton hierarchy, a skeleton tree browser, and a few rules reasoning about the current state of skeleton selection and completion. Manual skeleton tree descent is possible by using the graphical browser. Each part of the PDE is present in the first prototype with at least a *Graphical User Interface* (GUI) visualizing the meaning and functionality of these parts. The first prototype is implemented using Common Lisp, the Common Lisp Object System CLOS, a self–developed hybrid AI tool and a comfortable GUI builder producing CLOS–based LispView code.

Whereas the first prototype presents a top–down, overall view of the PDE, the objective of the second prototype is to elaborate on the central part of the PDE, the programming assistant PA. Thus, the two prototypes address completely different aspects of the project. In the second prototype, the PA is realized using mainly a bottom–up strategy. It is completely reimplemented using the CLIPS expert system shell and C++ together with a GUI builder. Knowledge representation and inference techniques from artificial intelligence play major roles for the task of the PA. The PA of the current prototype is complete in the sense of performing all important tasks, starting at the input of the problem specification as defined in [4] and produced by the stencil modeling programming assistant interface SMPAI, continuing with a successful skeleton selection among the skeletons represented in a frame hierarchy, up to the generation of master and slave descriptions for the parallel realization of the problem under consideration.

The program synthesizer is partially finished. The two major restrictions currently are that it supports specifications only with one grid in the problem domain and that grids can only have two dimensions or less.

7. SUMMARY AND FUTURE DEVELOPMENTS

In this paper we have presented the design and implementation of a program development environment which is deduced from the analysis of the requirements of scientific programmers, or in more general terms, computational scientists. This analysis has led to a programming methodology comprising a problem–oriented specification formalism, interactive refinement and completion of the specification, and with user–transparent generation of compilable program code. Tool support for this methodology centers around a knowledge–based system with a skeleton–based approach to the reuse of important software parts and to portability across different platforms.

The current PA and PS prototypes do not provide any target architecture dependent optimization of the communication steps. To achieve this, the next PA prototype will contain a hardware knowledge base.

Other limitations of the PDE, such as the restriction to one single grid with at most two dimensions will be removed in the next prototype. Not all of the stated design objectives have been realized within the current PDE implementation. For example, a tutoring component or sophisticated user guidance are still missing. The integration of these parts is an important short–term objective.

Besides that, we are investigating the definition of other programming assistant interfaces which either will cover more general problem specification paradigms, such as an interface for data parallel applications, or will present other scientific problem domains relevant to HPC.

ACKNOWLEDGMENT.
The project is funded by the Swiss National Science Foundation in the framework of the Swiss Priority Program Informatics, Grant–No. SPPIF–5009–034402.

REFERENCES

[1] Turcotte, L. H. "A Survey of Software Environments for Exploiting Networked Computing Resources." Report, Engineering Research Center for Computational Field Simulations, USAE Waterways Experiment Station, 39909 Halls Verry Road, Vicksburg, MS 39180–6199 (June 1993).

[2] Kuck, D. J. "A User's View of High–Performance Scientific and Engineering Software Systems in the Mid–21st Century." In *Expert Systems for Scientific Computing* edited by J. R. Rice, H. N. Houstis, and R. Vichnevetsky. Elsevier Science Publishers (1992).

[3] Dvorak, J. "An AI–based Approach to Massively Parallel Programming." Technical Report CSCS–TR–93–04, Swiss Scientific Computing Center (CSCS).

[4] Roth, M. "Generation of Algorithmic Skeletons from Stencil Specifications." Master's thesis, IAM, University of Bern (1993).

Involved Persons:
C. Clémençon (a)
K. M. Decker (a)
A. Endo (b)
J. Fritscher (a)
G. Jost (b)
N. Masuda (b)
A. Müller (a)
R. Rühl (a)
W. Sawyer (a)
E. von Stürler (a)
B. J. N. Wylie (a)
a) CSCS
b) NEC, SX–Center (Switzerland)

Funding Sources:
NEC (Germany)
CSCS

Duration:
April 1993–April 1996

JOINT CSCS–ETH/NEC HIGH–PERFORMANCE COMPUTING SOFTWARE DEVELOPMENT CENTER

1. INTRODUCTION

With the ratification of a comprehensive frame agreement, signed by CSCS–ETH Zürich and NEC Deutschland, the Joint CSCS–ETH/NEC High–Performance Computing Software Development Center was opened in Manno on March 13, 1993. The center functions as one of the research and development laboratories operated by NEC worldwide, including, for example, the NEC Research Institute in Princeton, NJ, USA.

2. GOAL AND OBJECTIVES

The actual research and development activities started on April 23, 1993. The center currently comprises ten researchers: Six of them are working on tools to simplify the usage of distributed memory parallel processors (DMPPs), and four are working on applications and parallel algorithms. All activities in the software development center have one common goal:

Turn massively parallel systems into practical tools.

The software developed should allow scientific programmers to use DMPPs as productive tools for their daily work in research and development laboratories without the need to bother about their complex architectural features and difficult programmability.

Three objectives can be identified:

- Develop an integrated tool environment for parallel distributed programming, together with its methodical foundation.
- Port full–scale, real applications to DMPPs.
- Develop new parallel distributed algorithms for selected problems.

3. THE INTEGRATED TOOL ENVIRONMENT

To ensure that the tools really give support for problems which are important in practice, the requirements on the tool environment are determined by abstraction from applications of scientific interest. This approach is explained in more detail in section 4. However, one of the most significant benefits of the tools approach to DMPPs is that the *general programmability* of these systems is significantly facilitated. This is particularly important for non–experts in parallel distributed computing. The tools approach is in contrast to an often followed procedure where a fixed number of turn–key applications are given to the users. When the user needs to modify these applications or the user requires other applications, she/he is not supported. The universality of the tools approach, hiding the complexity of massively parallel distributed programming, is in general widely under–appreciated in the scientific programmers' community.

The tool environment consists of the following components:

- A *Parallelization Support Tool* (PST).
- A *Performance Monitor and Analyzer* (PMA).
- A *Parallel Debugging Tool* (PDT).
- A common *User Interface* (UI).

One of the basic design issues of the tool environment is that it supports parallel program development in an application–oriented, high–level language, e.g., HPF or extensions. The tools also provide support for applications today considered difficult to parallelize on DMPPs, such as applications involving linear algebra on sparse matrices. All three tools, PST, PMA, and PDT share the single user interface UI. Both PMA and PDT are designed with the same philosophy, i.e., users are able to obtain information at different levels of abstraction. The lowest level of abstraction, providing the most detailed information, is as close as possible to the DMPP hardware. For instance, a detailed breakdown of parallelization overhead in communication, computation, and idle times on all processors is provided at any point of program execution. A higher level of abstraction is provided by considering features of the application–oriented high–level language, such as global name space and data distribution, or data parallel execution mode which appears to the user as a single program thread.

3.1 The Parallelization Support Tool PST

One of PST's main objectives is to provide extensive run–time support for irregular problems, similar in scope to what has for instance been done by J. Saltz and his colleagues at ICASE [1].

The first PST prototype is built using the parallelization support tool Oxygen [4]. Oxygen accepts Fortran and compiler directives and generates parallel C code with communication primitives for execution on DMPPs with torus topology. It has been ported to several platforms [5]. Oxygen supports a global name space through *dynamic data consistency analysis* and it supports *dynamically* distributed variables. For simple codes, e.g., LINPACK, compiler direc-

tives can be generated automatically based on standard data dependence analysis. We do not expect that these features will exist in first generation HPF compilers. Later PST prototypes will consist of the HPF system currently under development at NEC, extended with Oxygen's run–time facilities.

3.2 The Performance Monitor and Analyzer PMA

PMA provides performance monitoring and analysis support on different levels of abstraction.

On a low level of abstraction, PMA monitors and analyzes communicating processes close to the hardware, or close to the message passing interface and operating system. Information about the utilization of memory resources is also provided, since parallel programs not only suffer from execution time overhead, but also typically require more memory than their serial program counterparts.

On a high level of abstraction PMA will also support the application–oriented high–level language which simplifies the view of a parallel program, e.g., by using a global name space or by hiding parallel threads behind a single threaded data parallel model of execution. Therefore, the user can get easier access to performance information if PMA is able to provide a similar global view at important checkpoints, e.g., after a communication phase of the parallel program.

3.3 The Parallel Debugging Tool PDT

As is the case for the performance monitor and analyzer, PDT provides debugging support on different levels of abstraction.

On a low level of abstraction, PDT supports debugging on the abstraction level of standard high–level programming languages such as C, C++ or Fortran, enhanced by communication primitives. Using standard source level debugging methodologies, PDT also supports postmortem debugging of programs running on single nodes.

On a high level of abstraction, PDT supports debugging on the abstraction level of the application–oriented high–level language. The major advantage of this approach is that the parallel distributed program mostly presents itself as a single–threaded program.

3.4 The User Interface UI

The user interface UI provides a single interface to PST, PMA, and PDT. Various performance displays are provided, as well as a *zoom support*: both when debugging and when performance monitoring, the user can start at a high level of abstraction and successively move down to lower levels. UI also supports feedback between user and PMA/PDT.

4. THE ROLE OF APPLICATIONS

In the spirit of the philosophy to improve with highest priority the *general programmability* of DMPPs for scientific programmers, the primary purpose of applications is to specify the requirements on the integrated tool environment, and to successively test and evaluate the functionality and quality of its components while they are still under development. Efficiently parallelized applications will only be developed as side–products.

To ensure that the tool environment will become as universally applicable as possible, the applications must be carefully selected. Hence, the first task is to define an application suite which is representative of the entire spectrum of applications — from embarrassingly parallel to hard–to–parallelize applications. This is done by classifying the application spectrum, followed by selecting one or two applications from each class. This suite will be supplemented by a set of standard benchmarks and libraries.

Once the application suite is defined, the applications will be ported one by one to DMPPs, using the tool environment in its current state. Experiences collected will be shared with the tool developers, who will improve the tool environment accordingly. Those components of the applications which appear to have low parallel efficiency will be substituted by new, parallel distributed algorithms. This process is repeated until both the features and capabilities of the tool environment are found to be satisfactory and the applications show sufficient performance. As the final result, highly efficient parallelized applications are obtained, ready for production of scientific results.

5. ALGORITHMS

New parallel distributed algorithms to be developed are determined in an application–driven way. In the process of tool–supported porting of applications, sometimes computation intensive application components will be identified which cannot be mapped efficiently to massively parallel systems by program level transformations supported by the parallelization support tool PST. Then the algorithm experts come into play. They design and implement new algorithms, optimally adapted to the distributed nature of the target architecture that solve the same problem as the components they are going to replace, and integrate them in the application they were taken

from. Once this process is finished, the application is run again through the tool environment, measuring its *overall* performance. This process is repeated until full satisfaction is achieved.

The design of the parallel distributed algorithms will rely on standardized data structures to render the algorithms useful not only in the context of the single application program they were derived from, but in a much wider range of applications. The implementations of the algorithms will be collected in function libraries of general applicability.

6. STATUS

The tool environment is realized by a series of prototypes which are made rapidly available to the applications and algorithms developers for use and evaluation. This has also the advantage that verifying the functionality of the tool environment in the specification phase can be avoided to a large extend.

6.1 Improving the Usability of the Development Platform

The development platform for the center, a 16–node Cenju–2 system, was installed end of July 1993. First activities focused on enhancing programming comfort and on improving the system's usability.

Programming comfort is considerably enhanced by an implementation of a subset of the emerging *Message Passing Interface* (MPI) standard [3] on the Cenju–2 system as well as on Sun workstations used for software development. This subset comprises communication initialization functions, blocking and non–blocking sends and receives of continuous streams of data, global communications such as barrier synchronization, one–to–all and all–to–all broadcasts, all–to–one and all–to–all reductions with user–defined reduction operators, all–to–one gather and one–to–all scatter operations.

To enhance the system's usability, several operating system changes were carried out. These changes serve two purposes: First, optimization of the system's communication performance by simultaneously tuning the message passing interface and the underlying system calls. Second, basic changes to the operating system were performed in order to support the parallel debugging tool PDT. A running parallel program can now be interactively stopped, analyzed and restarted. In contrast to postmortem debugging, interactive in–core debugging is crucial to avoid expensive host communications. In addition, we implemented space–sharing which allows several users to simultaneously run parallel distributed programs on albeit smaller, non–overlapping partitions of the Cenju–2.

6.2 Tools

In the tools sector, the activities so far concentrated on three topics. First, the GNU gdb debugger from the Free Software Foundation [6] —a proven software product with comprehensive functionality—was ported. This port now allows interactive debugging at a low level of abstraction. Second, to allow also performance monitoring at a low level of abstraction, the ParaGraph tool [2] was ported to the Cenju–2 system. Both tools were selected to facilitate the initial progress of the center and might be replaced later. The third activity was related to porting the Oxygen parallelization support tool [4]. Oxygen allows data parallel programming and most notably provides parallelization support at run–time for difficult–to–parallelize problems.

To summarize, these three achievements not only allow much more convenient use of the Cenju–2 system, but also for a wider range of applications than ever before.

6.3 Applications and Algorithms

With respect to applications, the activities focused on the definition of the application suite, which is of key importance for the successful and application–oriented definition of the tool environment. Starting from the analysis of scientific users' input, the suite of applications was defined by means of a combined application–driven and method–driven approach. Besides defining the application suite, porting of first benchmark programs and applications was successfully completed. Work in this field included porting of NAS kernels (multigrid, conjugate gradient, block tridiagonal solver, integer sort, and 3d fast Fourier transform), and the port of two user applications. First performance measurements done on the Cenju–2 system show promising results. For example, solving the Poisson equation on a 128^3 lattice with the finite element method by means of a V–cycle multigrid algorithm, a maximum speed–up of 14 was measured for our 16 node system.

In the field of algorithms, efforts focused on techniques for iterative and direct solution of large, sparse, unstructured systems, and eigenvalue solvers.

REFERENCES

[1] Berryman, H., J. Saltz, and J. Schroggs. "Execution Time Support for Adaptive Scientific Algorithms on Distributed Memory Machines." *Concurrency: Practice and Experience* 3(3) June 1991.

[2] Heath, M. T. and J. E. Finger. "ParaGraph: A Tool for Visualizing Performance of Parallel Programs." Technical Report, Oak Ridge National Laboratory, October 1992.

[3] "MPIF (Message Passing Interface Forum)" Document for Standard Message–Passing Interface. Draft, September 1993.

[4] Rühl, R. *A Parallelizing Compiler for Distributed Memory Parallel Processors.* Kostanz, Germany: Hartung–Gorre Verlag, December 1992.

[5] Rühl, R. "Evaluation of Compiler–Generated Parallel Programs on three Multicomputers." In *Proceedings, ACM International Conference on Supercomputing.* Washington, July 1992.

[6] Stallman, R. M. and R. H. Pesch. *Debugging with GDB, 4.09.* Boston: Free Software Foundation, Inc., April 1993.

Involved Persons:
A. Deloff (a)
P. F. Flükiger (b)
a) Université de Genève
b) CSCS

Funding Sources:
ETH Rat, Bern
NEC (Japan) grant

Duration Dates:
November 1992–continuing

MOLEKEL, ADVANCED INTERACTIVE 3D–GRAPHICS FOR MOLECULAR SCIENCES

INTRODUCTION

One of the software tools for scientific visualization which are supported at the CSCS is the molecular graphics package MOLEKEL [1]. The purpose of this package, which has been developed at the University of Geneva and here at the CSCS, is the interactive visual representation of three–dimensional models of molecular structures and properties for research and education. It is mainly designed for generating detailed, easily interpretable and esthetically appealing graphical output based on structural data and the results of physico–chemical calculations. Due to the flexibility of the data– and program–structure, various chemical systems ranging from small organic compounds to large macro–molecules may be investigated, and due to its modularity, additional interfaces and tools can easily be implemented. Some of the features and fields of application of MOLEKEL are described and illustrated.

MOLECULAR GRAPHICS

Before entering into the details of the description of MOLEKEL, some definitions are necessary: *Molecular graphics* (MG) can be defined as the application of the computer graphics tool to molecular modelling, where *molecular modelling* is more generally the construction and utilization of models (i.e., simplified representations which may be purely numerical!) for rationalizing and predicting molecular structure, function and interaction. Besides MG, molecular modelling covers disciplines such as computational quantum chemistry, molecular mechanics, molecular dynamics and reaction dynamics; disciplines which have made enormous progress in the past years along with the technological improvement of computer hardware, and which take of course advantage of the tremendous computing power of today's supercomputers. On the NEC SX–3 for example, several commercial electronic structure calculation packages are installed, and many programs based on various methods are available for a multitude of platforms (workstations and PC's). The results generated by these physico–chemical programs are usually written into huge text–files, from which only the optimized geometries and the resulting energies will generally be retrieved; further investigation is only done by experienced users and is usually very time–consuming.

THE MOLEKEL PACKAGE

MOLEKEL now contains the tools which allow the rapid investigation and visualization of such results. As many of the retrieved or derived properties are spatially dependent, they must be represented in the molecular volume and anchored to the framework of the molecular structure, and therefore, a *three–dimensional* (3–D) representation is essential (which means of course that the structures and properties are manipulated and represented internally as 3–D objects, the resulting display on the screen is again a projection on two dimensions). So, the information displayed by MOLEKEL consists essentially of two parts: the molecular structure and the physical property.

Due to the importance of the structural representation, MOLEKEL offers different molecular models for emphasizing different aspects of the structure: the *space–filling* model depicts the volume by representing each atom by a colored sphere the size of its van der Waals radius; the *ball–and–stick* model, using smaller spheres for the atoms and thin cylinders for the bonds, places accent on the connectivities within the molecule; the *stick–only* model gives the most uncrowded view of a molecular architecture as only the bonds are represented using colored cylinders; and the *wire–frame* model consisting of colored vectors is necessary for real–time interaction with very large compounds such as proteins. Besides, a variety of viewing attributes may be set to alter the resulting image. The number of molecules –which may of course be loaded from different sources – is virtually unlimited, and so is the number of atoms per molecule.

Depending on the property that is to be visualized, different representation models may be appropriate. An *animation* of the molecular structure can for example illustrate molecular behavior with time–dependent atomic positions; *iso–value surfaces* (i.e., surfaces connecting all points of equal value) can be used for depicting any kind of property which can be calculated at each point of a regular 3D–grid in the molecular volume; *color–coded surfaces* can represent scalar qualities on arbitrarily shaped surfaces.

Properties which can be visualized using an *animated* sequence of atomic positions are for example the vectors generated by harmonic vibrational analysis (i.e., the internal "rocking–chair" movements of a molecule). Another example are the trajectories generated by molecular dynamics simulations which are based on flexible mechanical models of the molecules (classical mechanics

using spring forces between the atoms) and Brownian motion of the atoms.

Iso–value surfaces (i.e. surfaces connecting all points of equal value inside the molecular volume) can be used for depicting molecular orbitals (a molecular orbital is an electronic wavefunction which describes the form of an "electron–cloud" in the molecule; when squared, it yields the probability of finding the electron at a specific place) or derived properties such as the electron density (the sum of the probabilities of all electrons in the molecule) or the spin density (the difference between the probabilities of the spin–up and the spin–down electrons).

Color–coded surfaces can represent surface qualities such as the molecular electrostatic potential (which determines the long–range force acting on charged particles) on the molecular envelope.

In order to generate high–quality graphics output for presentations or publications, all object attributes can of course be modified at will: colors can be selected as well as any degree of semi–transparency, arbitrary clipping planes can be chosen to dissect the surfaces for revealing the inside structure, and raster images may be mapped as textures on any object for enhanced realism and better spatial perception. For a rapid and unobstructed representation, the chicken–wire or dot–representation may be chosen, which can of course be fully anti–aliased and depth–cued (i.e., the farther a line, the darker it appears).

One essential point for a vivid and convenient molecular representation is the real–time *interaction* with the represented objects: if MG techniques are to replace traditional mechanical models, the chemist must be able to manipulate the graphical objects and to look at them from different points of view. The term *"real time"* is of course based on the human time–scale and depends on the task that is to be accomplished: for simple rotations or translations, the action should be executed immediately, whereas for more sophisticated operations such as surface generations or calculation of physical properties, a longer response time of several seconds up to a few minutes may be acceptable. In MOLEKEL, the manipulation is done by using mouse–driven on–screen virtual controls or arrow–keys and the interaction with the various internal modules or external packages is executed through the graphical user interface which consists of user–friendly, mouse–driven menus and submenus, file–browsers and dialog–boxes.

THE INTERFACES TO EXTERNAL PACKAGES

MOLEKEL contains easy to use visual data–interfaces to the following external electronic structure calculation program–packages:

- ***Gaussian*** [2], a library of programs for ab initio and semiempirical molecular orbital calculations
- ***AMOSS*** [3], large scale direct *ab initio* quantum–chemistry code
- ***deMon*** [4], density functional calculation code
- ***ICON8*** [5], semi–empirical calculations based on the Extended Hückel method
- ***EHProp*** [6], semi–empirical calculations of interaction energies based on the Extended Hückel method

(Supplementary packages are at present being added to that list.)

The standard output–file of any of these packages can be loaded into MOLEKEL which then extracts all necessary information such as the atomic positions and basis–sets (i.e., the description of the electronic wavefunctions for each atom–type), the molecular orbital energies, occupations and coefficients, the density matrices and others. The extracted molecular structure is then displayed using one of the representation models. In case the harmonic vibrational modes have been computed, any of these can be visualized as an animation. The molecular orbitals, the spin density and the electron density can then be calculated at each point of a regular three–dimensional grid based on the retrieved data, where the grid can of course be defined interactively. As these computations may be CPU–intensive, they are executed in the background, eventually on a remote machine or even distributed on several machines. The resulting scalar field can then be visualized by generating isovalue surfaces based on the implemented marching cubes algorithm [7], or by a color–coded plane which can be moved through the grid. A contour tool [8] also allows the generation of planes with 2D–contour–lines within the 3D–scene. All of these operations take only a few minutes and can be used as a routine tool for rapidly analyzing computational results.

For *ICON8* and *EHProp* (and soon for *AMOSS*), it is equally possible to generate the input and to launch the calculations from within MOLEKEL, i.e., it is possible to prepare, launch and then analyze molecular modelling tasks in one session.

APPLICATION EXAMPLE

Some of the features of MOLEKEL are being demonstrated on *mannose*, a sugar which can be found in orange–peel. The structure has been obtained by molecular mechanics optimization using the AMBER force–field [9].

Within one single MOLEKEL–session of a few minutes, it is possible to pass from purely structural data to a visual representation of chemically relevant electronic properties such as the frontier orbitals: the input–file for performing an Extended Hückel calculation on that compound has been generated, the calculation has been launched, and upon termination, the resulting output file has been read into memory. After having defined the 3D–grid, the highest occupied molecular orbital (HOMO) of mannose has been calculated and visualized as aniso–surface (figure 1).

An other way of representing molecular properties is by using color coded molecular envelopes. To this end, a solid molecular surface which can be shaded must first be generated. This can be done in MOLEKEL by first executing Connolly's MS–program [10] in the background, which generates a set of surface dots, and then by triangulating these with the fast(!) internal triangulation tool [11]. The coordinates of the surface dots can then be used for calculating a molecular property, such as a molecular electrostatic potential (MEP), a hydrophobicity index or an interaction energy towards an attacking electrophile or nucleophile (as calculated by EHProp). This index can then be used as an index into a color lookup table to get a color–coded surface, or on machines which support texture–mapping, it can be used as a coordinate in a one–dimensional texture with a color scale, which results in a surface with quantized colors and well defined contour–lines on the curved surface. Besides, as a texture can contain transparency information, it is even possible to show only the most negative or the most positive regions of the surface (depending on what interests more) by making the rest of the surface entirely invisible. Of course, the options of arbitrary clipping planes or semi–transparent surfaces can equally be applied here.

This feature is again illustrated with the mannose molecule: the represented property is the MEP of mannose, based on the atomic charges calculated by the ICON program (figure 2). All the operations, from the generation and triangula-

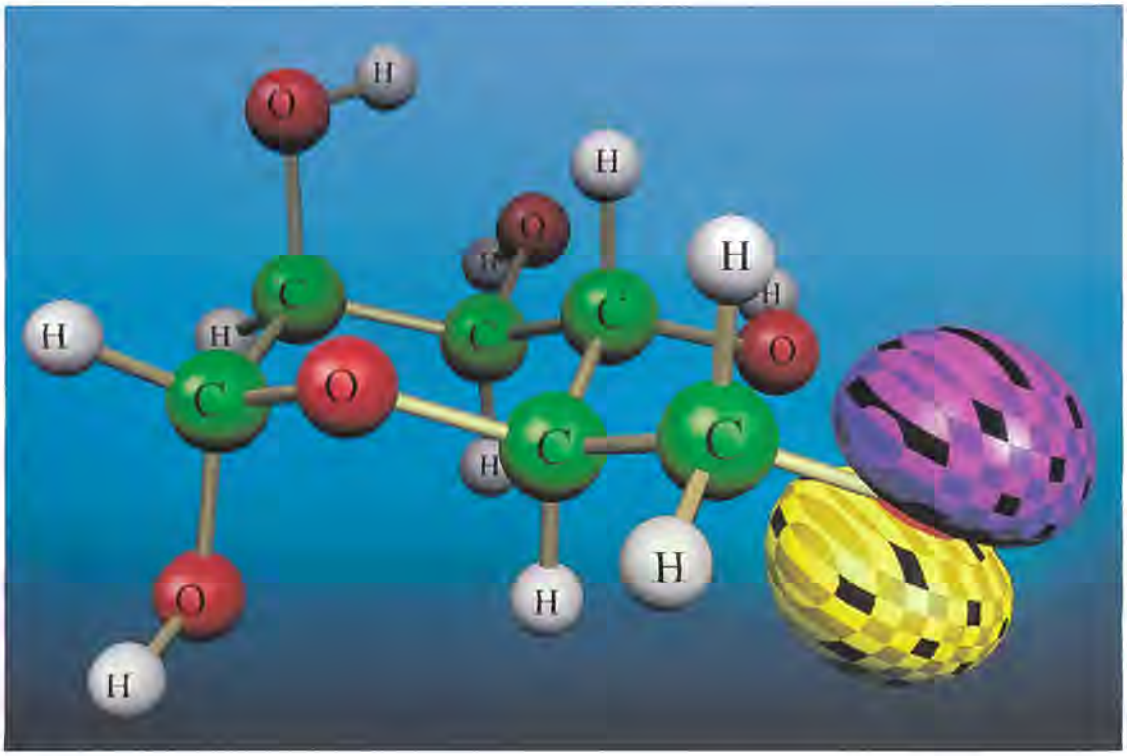


Figure 1: Ball–and–stick model of a mannose molecule with an iso–surface of the HOMO (Highest occupied molecular orbital) as calculated based on the ICON results, at the cutoff–values of +/– 0.07 a.u.

tion of the molecular surface to the encoded representation of the MEP have been done within one short session.

TECHNICAL DATA

MOLEKEL is written in ANSI–C and uses the Graphics Library (GL) of Silicon Graphics. It runs on any SGI workstation from Indy to Reality Engine. The development of a mixed–model Motif/OpenGL version is under way.

For further information, please contact:
flukiger@cscs.ch.

REFERENCES

[1] Flükiger P. F. , "Development of the Molecular Graphics Package MOLEKEL and its Application to Selected Problems in Organic and Organometallic Chemistry." Ph.D. thesis 2561, University of Geneva (1992).

[2] Frisch M. J., M. Head–Gordon, G. W. Trucks, J. B. Foresman, H. B. Schlegel, K. Raghavachari, M. A. Robb, J. S. Binkley, C. Gonzalez, D. J. Defrees, D. J. Fox, R. A. Whiteside, R. Seeger, C. F. Melius, J. Baker, R. L. Martin, L. R. Kahn, J. J. P. Stewart, S. Topiol and J. A. Pople. Gaussian Inc., Pittsburgh PA (1990).

[3] NEC Corporation (1992).

[4] Salahub D. R., R. Fournier, P. Mlynarski, I. Papai, A. St–Amant and J. Ushio, in "Density Functional Methods in Chemistry." J. K. Labanowski and J. W. Andzelm, eds., New York: Springer (1991), p. 77.

[5] Howell J. , A. Rossi, D. Wallace, K. Haraki, R. Hoffmann, QCPE Bull. 11 (1979) p. 344.

[6] Weber J. , P. F. Flükiger, P.–Y. Morgantini, O. Schaad, A. Goursot, C. Daul, *J. Comp. Aided Mol. Design* 2 (1988) p. 235.

[7] Lorensen W. E. , H. E. Cline. Comp. Graphics 21 (1987) p. 163.

[8] Deloff A. T. , Ph.D. thesis, University of Geneva, in preparation.

[9] Schwalm O., University of Geneva, personal communication.

[10] Connolly M. L. , "Molecular Surface Program." QCPE Bull. 1 (1981) p. 74.

[11] Weber J. , P. F. Flükiger, A. Ricca, P.–Y. Morgantini, in *Proceedings of the Conference: Visualisierungstechniken und Algorithmen*, edited by W. Barth. Berlin: Springer (1988) p. 17.

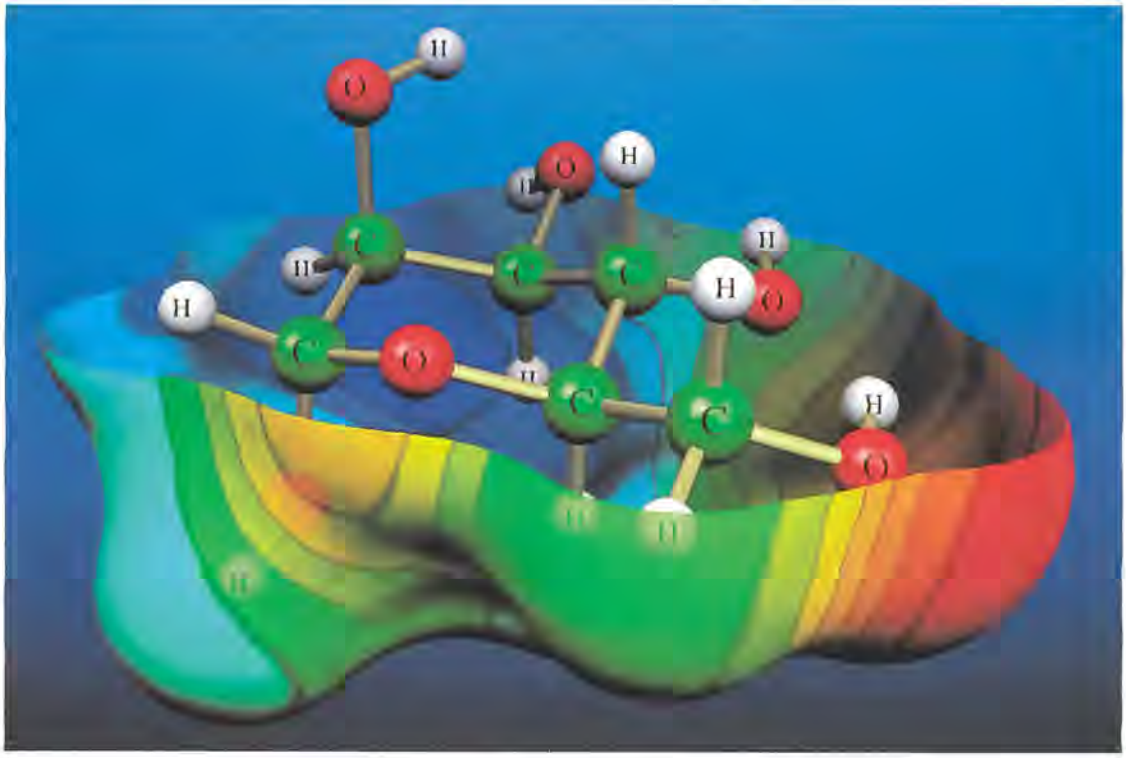


Figure 2: Ball–and–stick representation of mannose with a clipped, semi–transparent molecular surface; the MEP (molecular electrostatic potential)

AB INITIO STUDIES OF LARGE MOLECULES AND THE DESIGN ON NEW ANTI–CANCER AND ANTI–MALARIAL DRUGS

Involved Persons:

C. Jefford (a)
D. Maric (b)
C. Thomson (c)
J. Weber (a)

a) Université de Genève

b) CSCS

c) University of St. Andrews (UK)

Funding sources:

CSCS

World Health Organization, Geneva

Duration:

January 1993–December 1993

The computation of the structure and properties of large molecules of biological interest is providing us with new and more powerful tools with which to explore the relation between the molecular structure and biological activity of drugs, and there is no doubt that the future will see real practical benefits from such studies.

In the frame of this collaborative project, the CSCS hardware and the state–of–the–art software are employed to study the large molecules of the biological interest at the *ab initio* level of the theory. To this end, the program environment for computational chemistry and materials science (PECCAM) has been used.

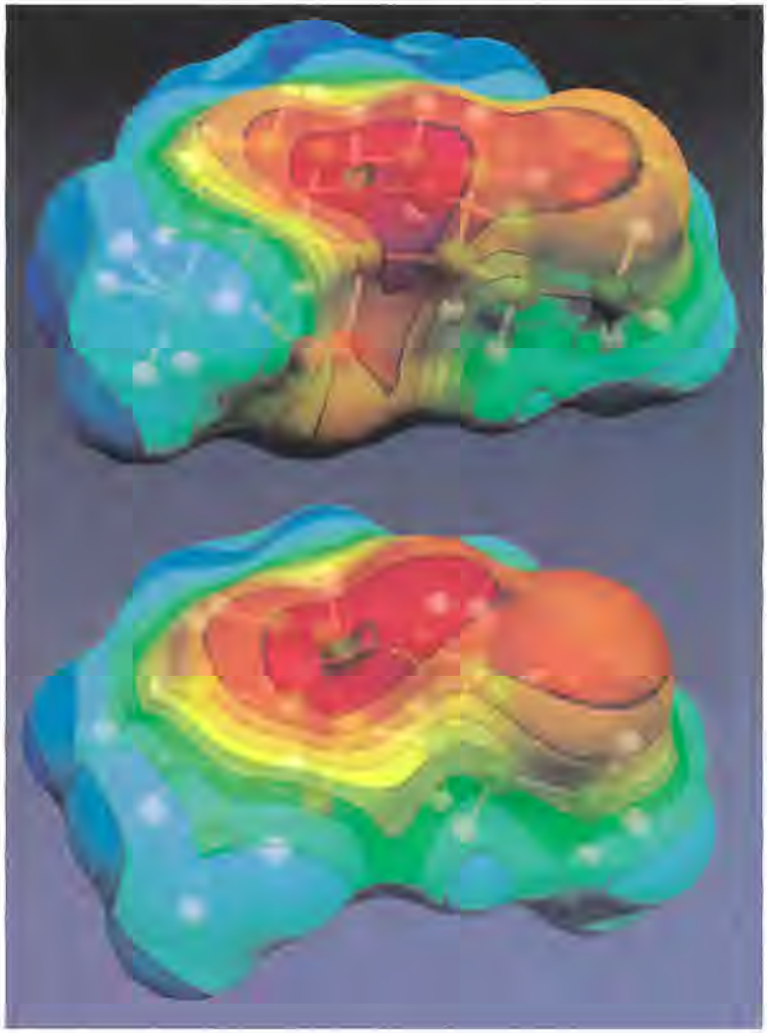
This application is focusing on the design of drugs which are useful in the treatment of two major medical problems, first in the treatment of CANCER, for which we need more selective drugs with fewer side effects, and secondly in the design of new anti–malarial drugs: MALARIA as a disease is a major health problem at the present time in underdeveloped countries because of the development of new strains of the parasite, which are resistant to all current anti–malarial drugs.

A novel natural product, Qinghaosu was discovered in China in the 1970s, which is active against these resistant strains, but more effective and cheaper drugs based on this novel structure (which contains the unusual 1,2,4–trioxane ring) are needed. In Geneva, a large number of such compounds have been made, but molecular calculations should enable us to explain the observed structure activity relations, and to predict the properties of as yet unsynthesized molecules.

The impetus for the theoretical studies was the experimental work carried out by the group of Professor Jefford in Geneva. *Ab initio* techniques in computational quantum chemistry, as implemented on supercomputers such as the NEC SX–3, can now be used to study such real biological problems involving molecules which contain up to around 100 atoms, and these molecules are of this size. The work carried out at Manno and at Geneva University (Laboratory, Prof. J. Weber) during this time has led us to the exciting conclusion that calculations using the SX–3 could play a very important part in the design of new and more effective drugs.

Such calculations in the past could only be carried out by semi–empirical methods, but detailed studies on small molecules have provided data on the accuracy of high quality *ab initio* calculations, and we are now in a position to state that such calculations should be of great value in predicting the structure and properties of potential new drugs which will have more specific action against molecular targets in a variety of disease states which have hitherto been difficult to treat in a rational way.

In the case of the anti–malarial drugs referred to above, we have computed structure and properties of these molecules, using full geometry optimization, and obtained the total energies, molecular geometries, and detailed wave functions. The latter have been used to compute electrostatic potential maps, which enable us to establish correlations between the computed properties of the molecules and their biological activity data. An example of the detail obtained from such maps is given in the figure, which shows differences between the maps for the active and inactive compounds. The difference between the regions on negative potential (in red) and positive potential (in blue) are quite striking, and this information is of direct relevance to how the mol-



ecules interact with the appropriate biological target. However, there are other molecules which have even better anti–malarial activity which have been more recently studied in Geneva, and we are about to begin more detailed studies of these compounds.

The advantage of *ab initio* studies in all this work is that they are capable of systematic refinement which will improve the accuracy of the results, and therefore lead to more reliable predictions of their biological activity. It is largely a matter of access to the fastest computers which limits the applicability of the methods described above to real problems. We have recently shown that with the SX–3, we are able to carry out geometry optimizations at the post–Hartree–Fock level of theory, which includes some electron correlation, and gives us substantially more accurate wave functions.

We are confident that this type of study will also be helpful in the design of new anti–cancer drugs, following earlier work in St. Andrews. Of particular interest is the development of new drugs active against *a)* solid tumours and *b)* specific enzymes whose activity may be different in tumours compared to normal tissue. It is of course well known that current anti–cancer drugs are relatively non–selective, and give rise to very serious side effects when used in patients. In addition, many are not very effective against solid tumours, which are the type of tumour found in the most common cancers in humans.

In the case of solid tumours, some new compounds which are more effective against this type of cancer have been discovered in recent years, and for these compounds, the flavone acetic acids, early theoretical work in St. Andrews has been able to rationalise some of the structure activity data, but the SX–3 offers an opportunity to increase the reliability of these predictions. At this time we are commencing a detailed study of a number of potentially useful drugs (some of which have been synthesised in St. Andrews) which are active against a variety of solid tumours.

Recently, intensive studies of the structure and properties of a series on novel inhibitors of protein tyrosine kinases have been carried out. The activity of the tyrosine kinase enzymes is found to be increased in certain types of cancer. The tyrphostins are relatively small molecules which bind to the enzyme and prevent it acting in the usual way. They are not too difficult to synthesise, and they have widely varying activity, depending on the functional groups in the molecule. The particular molecules we are studying are based on the substituted ethylene molecule, and professors Robins and Workman at Glasgow in Scotland have provided detailed experimental data on their activity which we hope to rationalise using our computer modelling techniques. The extension of these calculations using the SX–3 should prove to be very interesting, since we have shown that much more accurate calculations are possible on the SX–3 than on smaller machines. In all of this work we make heavy use of state of the art computer graphics techniques to evaluate the data.

In all of these cases, we are carrying out very large calculations, needing typically up to 800 MB of main memory. Total CPU times are several hours per job, but we are able to compute much more ACCURATE wave functions and properties for molecules of this size. We are confident that this kind of molecular modelling, using state of the art computers and software will make a substantial contribution to the design of more effective drugs for a wide variety of diseases.

REFERENCES

[1] Jefford, C. W., G. Bernardinelli, D. Maric, C. Thomson and J. Weber. "Computational Studies of the Structures and Properties of Potential Anti–malarial Compounds Based on the 1,2,4–trioxane Ring Structure." *Int. J. Quantum Chemistry Symposium* (submitted for publication).

[2] Maric, D. and C. Thomson. "Electronic Structure and Properties of the Simplest known 1,2,4–trioxane tetramethyl–hydroxymethyl–1,2,4–trioxane (TMHMT)." *J. Mol. Struct. (THEOCHEM)* (submitted for publication).

Involved Persons:
G. Benedek (a)
R. Car (b)
B. Delley (c)
R. Ernst (d)
P. Flükiger (e)
R. Gruber (e)
A. Mangili (e)
D. Maric (e)
D. Marynick (f)
P. F. Meier (g)
D. Salahub (h)
W. Thiel (g)
J. Weber (i)

OPTIMAL USER SUPPORT BY PECCAM: PROGRAMMING ENVIRONMENT FOR COMPUTATIONAL CHEMISTRY AND MATERIALS SCIENCE

1. INTRODUCTION

At present more than 60% of the CSCS CPU time is allocated to the user projects performing *microscopic simulations* (very large scale electronic structure calculations and molecular dynamics simulations). These methods are employed by research groups studying problems in chemistry, biochemistry, pharmacology, solid state physics and materials science.

Most of the computational methods used for microscopic simulations at CSCS have been developed by Swiss research groups, being CSCS users at the same time. CSCS has actively supported these authors in porting and optimizing their programs on the CSCS high–performance computing facilities, as well as has partly interfaced these programs to the CSCS visualization tools. In collaboration with the University of Geneva, CSCS has developed the powerful 3–D visualization package MOLEKEL [1] specialized for this group of users. Several animation developments have also been carried out for Swiss academic users.

The state–of–the–art development of the computer hardware and the computational methods make it possible to use these tools to study larger and more complex systems, as well as to design new substances, drugs and materials. This requires *simultaneous* usage of *different* simulation modules on the *heterogeneous* hardware in a *connected* way. A typical microscopic simulation passes through several phases ranging from the definition of the molecular structure, definition of the input parameters of the simulation, monitoring of the simulation, and finally the analysis of the computed molecular properties. Visualization and user interface can play an important role in all these phases, because the simulation process is highly interactive and demands important input specifications from the user. At present, the existing packages need different input specifications and they generate different output formats. Interchange between packages, even though dealing with the same type of data (atom type, geometry, basis set,...) becomes difficult. Usually there exists no real choice for a method, as only packages with existing interfaces can be used. Not to mention that each package demands a long period of apprenticeship, which is why users of one package are very reluctant to try another package. Moreover, not all the packages run on the best suited compute server architecture. Even for a given package, hardware selection based on the input type is desirable in order to increase the overall resource usage efficiency.

The proposed user–support environment will offer a solution for easily accessing a wide variety of chemical and material science packages via a uniform graphical user–interface GUI, based on a common data–base, well–defined data–interchange policies and a unique, general and sophisticated visualization module. Having the common data base, the packages will be ported and optimized on heterogeneous hardware, so that for each simulation the best suited compute server architecture can be chosen. The user can profit from a user–friendly system, has to learn the input and visualization formats only once for all the applications, always accesses the most advanced system and gets assistance to choose solution strategies.

2. THE PECCAM ENVIRONMENT

The software architecture of the PECCAM environment is shown in the figure. There are two major parts:

(i) The Data Carrier shown in purple
(ii) The Embedded Applications

The Data Carrier enables the overall data flow to the users, to the applications and to the heterogeneous hardware platforms. The interaction with the users is provided by the Graphical User Interface (GUI) and the Intelligent Software Layer (ISL). The Control System (CS) dispatches the modules to the most appropriate hardware platforms. The applications have to be embedded in PECCAM by means of interfaces towards the data structures defined in the Distributed Multimachine Data Manager (DMDM).

This system integrates the advantages and the functionalities of ASTRID [2] developed at EPFL, in which the MEMCOM [3] data manager already includes most of the functionalities of DMDM. It also includes the visualization system MOLEKEL [1] developed at University of Geneva and CSCS and a Molecular Builder to be developed.

Description of the different modules of the Data Carrier follow:

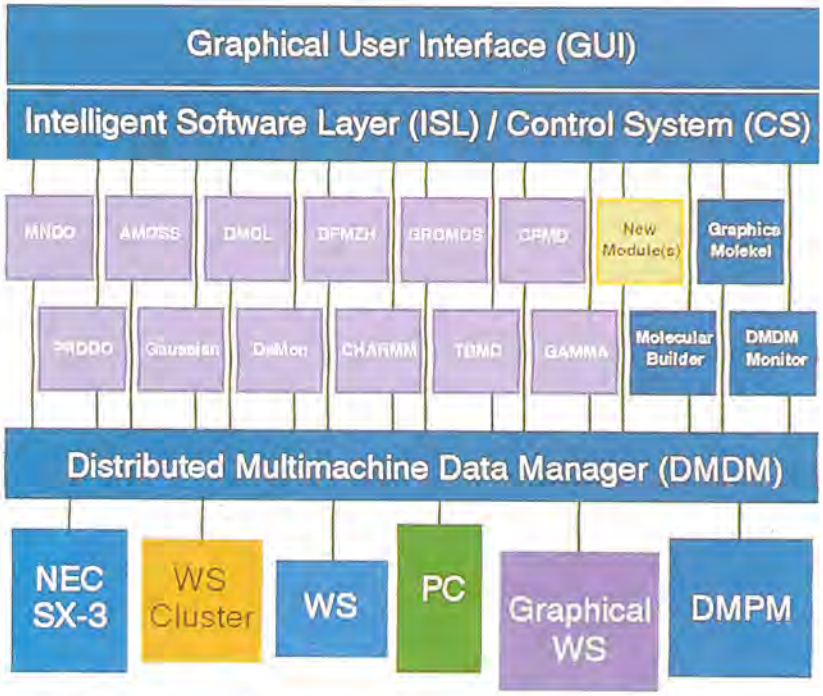
Distributed Multimachine Data Manager (DMDM)

Proper data definition and data management are important prerequisites for the design of a modular, hardware–independent, user–customized engineering analysis system which corresponds to

a) Università degli Studi di Milano (Italy)
b) IRRMA–EPFL
c) PSI
d) ETHZ
e) CSCS
f) University of Texas, Arlington (USA)
g) Universität Zürich
h) University of Montreal (Canada)
i) Université de Genève

Funding sources:
ETH Rat, Bern

Duration:
July 1993–December 1995



The software architecture of the PECCAM program environment with:

PECCAM Data Carrier (in purple):

DMDM:	*Data management system through which all data between modules flow*
GUI:	*Graphical user–friendly interface between the users and PECCAM*
ISL:	*Helps users preparing the simulation and interpreting the results*
CS:	*Controls the execution of the modules*
DMDM Monitor:	*Access data during run time*
MOLEKEL:	*Powerful 3–D visualisation system (J. Weber, P. Flükiger, CSCS)*
Molecular Builder:	*Builds molecular structures*

Embedded Applications:

MNDO:	*Semiempirical electronic structure calculations (W. Thiel)*
PRDDO:	*Approximative ab initio electronic structure calculations (D. Marynick)*
AMOSS:	*NEC ab initio electronic structure calculations packages*
Gaussian:	*Commercial library of ab initio electronic structure calculations methods*
DMOL:	*Density functional electronic structure calculations (B. Delley)*
DeMON:	*Density functional electronic structure calculations (D. Salahub)*
DFMZH:	*Plane waves based density functional method (P. F. Meier)*
CHARMM:	*Commercial classical molecular dynamics package*
GROMOS:	*Classical molecular dynamics (W. van Gunsteren)*
TBMD:	*Tight binding molecular dynamics (G. Benedek)*
CPMD:	*Ab initio molecular dynamics (R. Car)*
GAMMA:	*NMR structure elucidation (R. Ernst)*

modern software standards. The design of integrated environment relies on common rules for the description of the data structures and for the data flow during the various stages of a simulation process. A common data structure designed around a data management system and tailored to the particular needs of chemical application programs represents the foundations of a robust modular system.

DMDM will include all the functionalities of the existing data management system MEMCOM [3] for memory and memory–to–disk data handling.

The Graphical User Interface (GUI)

There is a uniform GUI from which all functionalities of the environment can be accessed in a uniform way. The GUI enables to

define all data necessary to prescribe an initial molecular structure and the computation procedure. A direct access to the DMDM through the DMDM Monitor, to the MOLEKEL graphics package and to the Molecular Builder is provided by the GUI. This GUI is menu–driven and keyword–driven with a specific command language. The user will not have to learn the exact input formats for each computational package. Moreover, the choices of the methods and the basis–sets, the precision of the calculation and the presentation of the results can be prescribed by clicking or by executing scripts.

The Intelligent Software Layer (ISL)
The Intelligent Software Layer may help to take decisions about the best suited computational method, may support the user in the best choice of the input parameters and in the analysis and interpretation of the results.

The Control System (CS)
The Control System dispatches the executional modules of the applications to the best suited hardware. This distribution is done on the basis of the size of the problem, the requested property and the desired precision as well as the load of the available hardware platforms.

The DMDM Monitor
This monitor enables access to the DMDM. It is possible to access data at run time, interpret it and stop the running process or choose a new execution path.

An aspect that becomes important in this context is the possibility of monitoring the simulation from the beginning. This approach has several advantages that are too often ignored: First of all it allows early detection of errors in the definitions of the data or in the computational model. Second it helps to understand the simulation and to figure out what happens during the simulation. Third it may help to choose the level of accuracy of the result. Simulation can then be stopped and examined at any stage, thus saving user and computer time.

The Graphics package MOLEKEL
The molecular graphics package MOLEKEL [1] is one of the software tools for scientific visualization which are supported at the CSCS. It has been developed at the University of Geneva and at the CSCS for research and education in the fields of Computational Chemistry and Materials Science. This package is described in more details in this report.

The Molecular Builder
One of the modules which belong to the core of the PECCAM environment is the rendering module which displays interactively three–dimensional scenes of the materials and properties under examination. It plays an essential role throughout the process of computer–aided molecular design (CAMD) such as during generation or modification of a molecular structure; during preparation of the input for the computational packages; during computations for monitoring the ongoing simulation; for visual presentation of the results and for in–depth analysis of the various physical properties; and for presenting and sharing results among scientists.

We are looking for a molecular builder that we can embed into PECCAM.

3. MODULES TO BE INTEGRATED

The proposed project will integrate the most important program–packages in computational chemistry and material sciences used for very large scale calculations and equip them with a unique user–interface and one graphics visualization tool. The authors of the packages will provide the latest versions of the modules and the users will be offered the optimized versions of the programs on the best suited hardware platform.

The packages which are planned to be integrated in the programming environment are already being used at the CSCS. They can be classified in three categories: "Electronic Structure Calculation", "Molecular Dynamics", "Structure Elucidation". The environment will be built in an open way, so that new applications can be easily added to it. At present, the microscopic simulations packages to be integrated in the environment are the following:

Electronic Structure Calculations:
Semi–empirical methods
- MNDO, W. Thiel, University of Zurich [5]

Approximate *ab initio* methods
- PRDDO, D. Marynick, University of Texas at Arlington, USA [6]

Ab initio methods
- AMOSS, NEC [7]
- Gaussian, Gaussian Inc. [8]

Density functional methods
- DMOL, B. Delley, PSI [9]
- DeMon, D. Salahub, University of Montreal, Canada [10]
- DFMZH, P. F. Meier, University of Zurich [11]

Molecular Dynamics (MD):
Classical MD
- CHARMM, M. Karplus, Harward University, USA [12]
- GROMOS, W. van Gunsteren, ETHZ [13]

Semi–empirical MD
- TBMD (Tight Binding MD), G. Benedek, University of Milano [14]

Ab initio MD
- *Ab initio* MD, R. Car, IRRMA–EPFL [15]

Structure Elucidation:
Computer simulations in magnetic resonance
- NMR toolkit GAMMA, R. Ernst, ETHZ [16]

4. DOCUMENTATION, WORKSHOPS AND SUPPORT

The technical documentation and especially the PECCAM user's guide will also be available on–line in the form of hypertext documents. Context–sensitive help facilities will be implemented for easy understanding of the graphical user interface.

Courses and seminars for beginners and advanced users of the system will be organized on a regular basis in order to show and teach scientist how to use this environment.

Two workshops will be organized at CSCS, the first one in April 1994 to inform about the project status and to discuss about the further realization steps. The second workshop to be held in Summer 1995 will be focused on the results of the show cases produced within the PECCAM pilot projects in the fields of computational chemistry and materials science.

5. REFERENCES

[1] Flükiger, P. F. "Development of Molecular Graphics Package MOLEKEL and its Application to Selected Problems in Organic and Organometallic Chemistry." Ph.D. thesis 2561, University of Geneva (1992).

[2] Bonomi, E., M. Fluck, R. Gruber, R. Herbin, S. Merazzi, T. Richner, V. Schmid, and C. T. Tran. "ASTRID: a programming environment for scientific applications on parallel vector computers." In *Scientific Computing on Supercomputers II,* edited by J. T Devreese and P. E. van Camp. New York: Plenum (1990).

[3] Merazzi, S. "The MEMCOM user manual (version 5.6)." SMR Corporation, Bienne, Switzerland (1988).

[4] Gobbetti, E., J. F. Balaguer, A. Mangili, R.Turner. "Building an Interactive 3D Animation System." In *Object–Oriented Applications*, Prentice Hall (1993).

[5] Dwar, M. J. S. and W. Thiel. *J. Am. Chem. Soc.* 99 (1977) 4899.

[6] Halgren, T. A. and W. N. Lipscomb. *J. Chem. Phys.* 58 (1973) 1569;
Marynick, D. S. and W. N. Lipscomb. *Proc. Natl. Acad. Sci.* USA 79 (1982) 1341;
Throckmorton, L. and D. S. Marynick. *J. Comp. Chem.* 6 (1985) 652;

[7] "Ab–Initio Molecular Orbital System for Supercomputers." NEC Corp. (1992).

[8] "Gaussian 92"; M. J. Frisch, G. W. Trucks, M. Head–Gordon, P. M. W. Gill, M. W. Wong, J. B. Foresman, B. G. Johnson, H. B. Schlegel, M. A. Robb, E. S. Replogl, R. Gomperts, J. L. Andres, K. Raghavachari, J. S. Binkley, C. Gonzalez, R. L. Martin, D. J. Fox, D. J. Defrees, J. Baker, J. J. P. Stewart and J. A. Pople. Gaussian, Inc., Pittsburgh PA, 1992.

[9] Delley, B. *J. Chem. Phys.* 92 (1990) 508;
Delley, B. *J. Chem. Phys.* 94 (1991) 7245.

[10] St–Amant, A. and D. R. Salahub. *Chem. Phys. Lett.* 169 (1990) 387;
Salahub, D. R. , R. Fournier, P. Mlynarski, I. Papai and J. Ushio in "Density Functional Methods in Chemistry." Edited by J. Labanowski and J. Andzelm, Berlin: Springer–Verlag (1991).
St–Amant, A. Ph.D thesis, Université de Montreal, 1992.

[11] Payne, M. C. et al. *Rev. Mod. Phys.* 64 (1992) 1045;
Myers S. M. et al. *Rev. Mod. Phys.* 64 (1992) 559.

[12] Brooks, B. R., R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan and M. Karplus. *J. Comp. Chem.* 4 (1983) 187.

[13] van Gunsteren, W. F. and H. J. C. Berendsen. "Groningen Molecular Simulation Library Manual." University of Groningen, 1987.

[14] Goodwin, L. , A. J. Skinner and D. G. Pettifor. *Europhys. Lett.* 9 (1989) 701;
Xu, C. H. , C. Z. Wang, C. T. Chan and K. M. Ho. *J. Phys. Condens. Matter* 4 (1992) 6047;
Servalli, G. and L. Colombo. *Europhys. Lett.* 22 (1993) 107.

[15] Car, R. and M. Parrinello. *Phys. Rev. Lett.* 55 (1985) 2471.

[16] Smith, S. , T. Levante, B. H. Meier and R. R. Ernst. "Computer Simulation in Magnetic Resonance: An Object Oriented Programming Approach." *J. Mag. Res.* A (Dec. 93).

MANAGEMENT

Decker, Karsten
Director of Research and Development (SeRD)

Gruber, Ralf
EPFL
Director of Scientific Applications (SeSAM)

Pelloni, Norman
Director of Technical Operations (SeTO)

Scheidegger, Alfred
Director of CSCS

STAFF

Anastasi, Andreas
SeTO;
Technical infrastructure*

Ballabio, Mauro
SeSAM;
Application software

Bernasconi, Andrea
SeSAM;
Application software

Buzzini Soldati, Ines
CeF;
Secretariate

Clémençon, Christian
SeRD;
Software R&D

Corti, Gian Carlo
SeTO;
Programmer

Dvorak, Jiri
SeRD;
Software R&D

Falconi, Romano
SeTO;
Office assistant

Flükiger, Peter
SeSAM;
Graphic software

Fritscher, Josef
SeRD;
Software R&D

Gay, Mario
SeTO;
System management

Giordano, Silvia
SeTO;
Network management

Hodous, Michael
SeSAM;
Application software*

Londino, Letizia
CeF;
Secretariate

Mangili, Angelo
SeSAM;
Graphic software

Mari, Gianpaolo
SeTO;
Technical infrastructure

Maric, Djordje
SeSAM;
Application software

Mastropietro, Roberto
SeTO;
System management

Meyer, Urs
SeSAM;
Visualization*

Müller, Andreas
SeRD;
Software R&D

Parini, Carla
CeF;
Secretariate and personnel administration

Pedrozzi, Nicola
SeTO;
Programmer

Rehmann, René
SeRD;
Software R&D

Rühl, Roland
SeRD;
Software R&D*

Sawyer, William
SeRD;
Software R&D

Schaidl Meyer, Elena
CeF;
Library

Simon, Erika
CeF;
Marketing and special projects

Thibaud, Jean–Louis
SeTO;
System management*

Tomassini, Marco
SeSAM;
Application software*

Vecchi, Nicole
CeF;
Publications

Vitali, Enrico
SeTO;
System management

von Stürler, Eric
SeRD;
Software R&D

Walther, Silvia
CeF;
Publications

Wylie, Brian
SeRD;
Software R&D

Zumthor, Bernardo
CeF;
External and user relations

NEC SITE SUPPORT

Barker, David P.
NEC;
Application software

Chiu, Jia Yu
SeSAM; NEC;
Application software

Endo, Akiyoshi
SeRD; NEC;
Research & Development

Hirano, Kinya
NEC;
Hardware maintenance

Ishii, Masahiko
NEC;
System maintenance/support

Jost, Gabriele
SeRD; NEC;
Research & Development*

Koike, Toshiki
NEC;
System maintenance/support

Masuda, Norio
SeRD; NEC;
Research & Development

Takagi, Yasushi
NEC;
Hardware maintenance

*Functional Responsible

VISITORS

Baggi, Yves
University of Geneva, Switzerland
Numerical simulation of ferrofluids for performance comparisons between multi–spin–coding and direct–coding on the CM–2 and the Cenju–2
(August–October 1993)

Banfi, Folco
University of Neuchatel, Switzerland
Internship using imake to compile and install AVS Modules
(July–August 1993)

Blandy, Jim
Oberlin College, USA
Debugging support on the Cenju–2
(August–October 1993)

Chatagny, Rodolphe
University of Geneva, SEINF, Switzerland
Scientific application on the Cenju–2: multiparticle lattice gas models for hydrodynamics
(August–October 1993)

di Pietro, Gianni N.
ASCOM Tech, Solothurn, Switzerland
Project discussions
(October 1993)

Estreicher, Dr. Stefan K.
Texas Tech University, USA
Cluster modeling of semiconductors
(July–August 1993)

VISITORS

Gutzwiller, Dr. Stephan
Institute for Informatics, University of Basel, Switzerland
Skeleton–oriented programming for a programming environment for massively parallel systems
(August 1993)

Katayama, H.
NEC Corporation, Tokyo, Japan
Project discussion
(April 1993)

Kühn, Ulrich
University of Munster, Germany
Analysis of selected NAS benchmark kernels
(August–October 1993)

Marynick, Prof. Dennis S.
University of Texas at Arlington, Texas USA
Very large scale electronic structure calculations: porting and tuning of PRDDO on the SX–3
(August 1993)

Medevedev, Dr. Sergio
Keldysh Institute for Applied Mathematics, Russian Federation
CUI development of equilibrium and stability programs for Doublet geometries
(June 1992–February 1993)

Nyeu, Maung Ting
University of California at Irvine, USA
Profiling support on the Cenju–2
(August–October 1993)

O'Sullivan, Bryan
Trinity College, Ireland
Distributed Systems Group, Department of Computer Science, University of Dublin, Ireland
Graphical editor for stencil based problems
(August–October 1993)

Pagny, Pascal
School for Informatics and Advanced Technics, France
Optimization of the fluid mechanics program from ETH (Vectorialisation on NEC SX–3)
(October–December 1993)

Pollei, Anja
ETHZ
Numerical simulation of crystallization in various flows with regard to a space dependent crystal size distribution
(February 1993–continuing)

Pontiggia, Ivan
University of Neuchatel, Switzerland
Diploma thesis on color correction for a color laser copier
(November 1992–June 1993)
LINDA on MPI and the Cenju–2
(August–October 1993)

Slivnik, Bostjan
Institute J. Stefan, Slovenia
Towards a parallel and distributed MOLEKEL: A comparison of message passing capabilities of Linda and PVM
(August–September 1993)

Stern, Christian
University of Zurich, Switzerland
Parallel implementation of the computation–intensive parts of the MOLEKEL package using a workstation cluster
(May–November 1993)

Thomson, Prof. Colin
University of St. Andrews, UK
Design of new anti–malarial drugs
(September 1993)

Turnherr, Andreas
ETHZ
Realization of a comprehensive CSCS research activities video
(February–August 1993)

FEBRUARY 1993

Fischer, Thomas – IPS ETH Zürich
"New Developments in the *Ab–Initio* Molecular Structure Optimization of Large Molecules: Theory and Applications."
Swiss High–Performance Computing Seminar
CSCS; February 8, 1993.

Medvedev, Sergei – CSCS
"MHD Equilibrium and Stability Codes for Tokamak Plasma with Separatrix."
Swiss High–Performance Computing Seminar
CSCS; February 10, 1993.

Tomassini, Marco – CSCS
"Les Algorithmes Génétiques Parallèles."
Séminaire du Laboratoire de Systèmes Logiques
EPF, Lausanne; February 19, 1993.

MARCH 1993

Gunzinger, Anton – Electronics Laboratory, ETH Zürich
"Achieving Super–Computer Performance with a DSP Array Processor."
Swiss High–Performance Computing Seminar
CSCS; March 16, 1993.

Ladkin, Peter B.– University of Stirling, U.K.
"Static Analysis of Communicating Processes with Finite–State Methods."
Swiss High–Performance Computing Seminar
CSCS; March 30, 1993.

Rühl, Roland – CSCS
"A Parallelizing Compiler for Distributed Memory Parallel Processors."
Swiss High–Performance Computing Seminar
CSCS; March 4, 1993.

Tomassini, Marco – CSCS
"L'Esperienza del Centro Svizzero di Calcolo Scientifico."
Il Centro di Calcolo di Ateneo: Quale Ruolo e Quali Prospettive
University of Perugia, Italy; March 12, 1993.

APRIL 1993

von Stürler, Eric – Delft University of Technology, Netherlands
"The Efficient Implementation of Conjugate Gradient–like Methods on MPP."
Swiss High–Performance Computing Seminar
CSCS; April 22, 1993.

Gerteisen, Edgar A. – Dornier Luftfahrt GmbH, Germany
"Massive Parallel Implementation of the Aircraft Euler Method and Performance Tests on Different Computational Platforms."
Swiss High–Performance Computing Seminar
CSCS; April 27, 1993.

Koike, Nobuhiko, Yoshiki Seo, and Tsukasa Yamauchi – Computer System Research Laboratory, NEC, Japan
"The Hardware Architecture of the Parallel Machines Cenju–2/3."
Swiss High–Performance Computing Seminar
CSCS; April 19, 1993.

MAY 1993

Dvorak, Jiri J. and René M. Rehmann – CSCS
"PDE: A Tool Environment for Parallel Programming of Distributed Systems."
SPPIF Workshop 32
Berne, Switzerland; May 15, 1993.

Gupta, Manish – IBM T. J. Watson Research Center, New York, USA
"PARADIGM: A Compiler for Automatic Data Distribution on Multicomputers."
Swiss High–Performance Computing Seminar
CSCS; May 3, 1993.

Rehmann, René M. – CSCS
"An Environment for Parallel Scientific Computing: The Spade System and its Functional Units."
Swiss High–Performance Computing Seminar
CSCS; May 13, 1993.

COLLOQUIA

Stoll, Erich P. – Institute of Physics, University of Zurich
"Computer Simulations of Dynamic Fractal Systems: Optimization Problems and Physical Results."
Swiss High–Performance Computing Seminar
CSCS; May 27, 1993.

Tomassini, M. – CSCS
"Algoritmi Genetici: Cosa Sono, a Cosa Servono."
Seminario del Colloquio di Matematica
CERFIM, Locarno, Switzerland; May 7, 1993.

JUNE 1993

Cha, Hojung – Department of Computer Science, University of Manchester, U.K.
"Simulated Behaviour of SCI Rings and Tori."
Swiss High–Performance Computing Seminar
CSCS; June 18, 1993.

Datcu, Mihai – Image Science Division, Institute for Communication Technology, ETH Zürich
"Image Formation."
Swiss High–Performance Computing Seminar
CSCS; June 9, 1993.

Mattson, Timonthy G. – Yale University and Scientific Computing Associates, Inc., Hartford, Connecticut USA
"Portable Parallel Programming with Linda."
Swiss High–Performance Computing Seminar
CSCS; June 18, 1993.

Maric, Djordje – CSCS
"CSCS Activities and Potentials in Computational Chemistry and Materials Science."
CSCS; June 1, 1993.

Scheidegger, Alfred – CSCS
"Japan's Sprung vom Mittelalter in das 3. Jahrtausend."
New Technology Week of Ingenieure für die Schweiz von Morgen
Zurich, Switzerland; June 15, 1993, also presented in Baden, Switzerland; June 24 and September 21, 1993.

JULY 1993

Scheidegger, Alfred – CSCS
"Computing – A Key Technology for Economic Success."
The R&D Management Conference: Technology Assessment and Forecasting
Zurich, Switzerland; July 5–7, 1993.

Sprenger, Christoph – ETH Zürich
"Sciddle – A Programming Tool for Parallel Distributed Computing with RPC."
Swiss High–Performance Computing Seminar
CSCS; July 8, 1993.

AUGUST 1993

Meyer, Urs – CSCS
"Using imake to Maintain an AVS Module Tree."
AVS User's Group Meeting, Siggraph '93
Anaheim, California USA; August 4, 1993.

SEPTEMBER 1993

Barker, John A. – IBM Almaden Research Center, San Jose, California USA
"Interaction of Xenon with the (1 1 1) Surface of Platinum."
Swiss High–Performance Computing Seminar
CSCS; September 24, 1993.

Decker, Karsten M. – CSCS
"Methods and Tools for Programming Massively Parallel Distributed Systems."
14th Speedup Workshop on Vector and Parallel Computing
Zurich, Switzerland; September 17, 1993.

Heiser, Gernot – Department of Computer Science, University of New South Wales, Australia
"Mungi: a Distributed Single Address–Space Operating System."
Swiss High–Performance Computing Seminar
CSCS; September 15, 1993.

Scheidegger, Alfred – CSCS
"The Industrial Partnership Program at CSCS."
2. Kongress in Technologiemarketing
Zurich, Switzerland; September 10–11, 1993.

Shtilman, Leonid – Department of Fluid Mechanics and Heat Transfer, Tel–Aviv University, Israel
"Simulations of Turbulent Flows on Vectoral and Parallel Computers."
Swiss High–Performance Computing Seminar
CSCS; September 21, 1993.

OCTOBER 1993

von Stürler, Eric – CSCS
"Incomplete Block LU Preconditioners from Slightly Overlapping Subdomains on a Massively Parallel Computer."
7th International Conference on Domain Decomposition Methods in Scientific and Engineering Computing
Pennsylvania State University, University Park, Pennsylvania USA; October 27–30, 1993.

Fraser, Dan – TMC, Los Alamos National Laboratory, California USA
"Automatic Load Balancing for Globally Irregular Grids on the CM–5."
Swiss High–Performance Computing Seminar
CSCS; October 20, 1993.

Hegland, Marcus – CISR, Australian National University
"Block Algorithms for Fast Fourier Transforms on Vector and Parallel Computers."
Swiss High–Performance Computing Seminar
CSCS; October 7, 1993.

Liegmann, Arno – Integrated Systems Laboratory, ETH Zürich
"SUPER–A Flexible Direct Solver For Sparse Structurally Symmetric Linear Systems."
Swiss High–Performance Computing Seminar
CSCS; October 21, 1993.

Wylie, Brian J. N. – CSCS
"PARAMICS: Parallel Microscopic Traffic Simulator."
2nd European Connection Machine Users Meeting
Paris, France; October 11, 1993.

DECEMBER 1993

Eigenmann, Rudolf – Center for Supercomputing Research and Development (CSRD), University of Illinois, USA
"Toward Real Performance Improvement of Parallel Computer Applications."
Swiss High–Performance Computing Seminar
CSCS; December 16, 1993.

George, Felicity A. W. – Edinburgh Parallel Computing Centre, University of Edinburgh, U.K.
"Spatial Interaction Modelling on the Connection Machine 200."
Swiss High–Performance Computing Seminar
CSCS; December 2, 1993.

Maric, Djordje – CSCS
"Molecular Dynamics Simulation of the Defect Induced Crystal to Amorphous Transition in Silicon."
Materials Research Society, Fall Meeting
Boston, Massachusetts USA; December 2, 1993.

Wylie, Brian J. N. – CSCS
"PARAMICS: Parallel Microscopic Traffic Simulator."
Swiss High–Performance Computing Seminar
CSCS; December 9, 1993.

COURSES

FOLLOW–UP COURSES FOR UNEMPLOYED SOFTWARE PROGRAMMERS

CSCS, Manno; January 8–July 7, 1993

Courses about Unix, C language, Object Oriented programming, CASE, RDBMS Oracle, X–windows.
Organized by CSCS in collaboration with the local employment office.

1993 SPEEDUP 13TH WORKSHOP ON VECTOR AND PARALLEL COMPUTING; SPECIAL TOPIC: INDUSTRIAL AND COMMERCIAL APPLICATIONS

Lugano; March 18–19, 1993

Organized by the Speedup Society in conjunction with CSCS.

SUMMER STUDENT INTERNSHIP

CSCS, Manno; August 16–October 29, 1993

Basic DMPP programming theory and techniques culminating with a research project.
Organized by CSCS.

ERCOFTAC SUMMERSCHOOL ON DIFFUSION AND TRANSPORT OF POLLUTANTS IN THE ATMOSPHERIC MESOSCALE FLOW FIELDS

CSCS, Manno; August 23–27, 1993

Organized by Prof. T. Dracos and Dr. A. Gyr (Institut für Hydromechanik und Wasserwirtschaft, ETHZ) in conjunction with CSCS.

ERCOFTAC WORKSHOP ON THE INTERCOMPARISON OF ADVANCED PRACTICAL SHORT–RANGE ATMOSPHERIC DISPERSION MODELS

CSCS, Manno; August 30–September 3, 1993

Organized by JRC Ispra in conjunction with CSCS.

VISUALIZATION COURSE

CSCS, Manno; September 13–15, 1993

Organized by the visualization group of CSCS.

INDUSTRIAL APPLICATIONS: IMPACT ANALYSIS AND METAL FORMING WITH LS DYNA 3–D

CSCS, Manno; September 20, 1993

Organized jointly by CAD–FEM, CSCS and NEC.

THE INTERNATIONAL WORKSHOP PASE '93 (PARALLEL APPLICATIONS IN STATISTICS AND ECONOMICS)

Monte Verità, Ascona; November 22–26, 1993

Jointly organized by ETHZ, CSCS and the Czech Academy of Sciences. It was realized under the auspices and with the help of the Centro Stefano Franscini.

PUBLICATIONS

TECHNICAL REPORTS

Banfi, F. and **U. Meyer**. *Using imake to Compile and Install AVS Module*. CSCS–TR–93–06, December 3, 1993.

Clémençon, C., A. Endo, A. Müller, R. Rühl, and B. Wylie. *An Environment for Portable Distributed Memory Parallel Programming*. CSCS–TR–93–05, November 19, 1993.

Decker, K. M., J. Dvorak and R. Rehmann. *A Knowledge–Based Scientific Parallel Programming Environment*. CSCS–TR–93–07, December 9, 1993.

Dvorak, J. *An AI–based Approach to Massively Parallel Programming*. CSCS–TR–93–04, August 23, 1993.

Medvedev, S., L. Villard, **R. Gruber** and S. Merazzi. *MHD Equilibrium Code for Axisymmetric Plasma With Separatrix*. CSCS–TR–93–01, February 19, 1993.

Merazzi, S., **R. Gruber**, C. Pfistner and R. Weber. *Numerical Simulation of Mechanical and Optical Properties of Solid–State Lasers*. CSCS–TR–93–02, April 28, 1993.

Pommerell, C. and **R. Rühl**. *Compiler Assisted Distributed Memory Parallelization of an Iterative Solver for Irregular Sparse Linear Systems*. CSCS–TR–93–03, April 29, 1993.

von Stürler, E. and H. A. van der Vorst. *Reducing the Effect of Global Communication in GMRES(m) and CG on Parallel Distributed Memory Computers*. Technical report 832, Dept. of Mathematics/Mathematical Institute, University of Utrecht, October 1993.

ARTICLES

Bengtsson, L., M. Beniston, **A. Bernasconi**, U. Cubasch, M. Esch, P. Lenzen, **A. Mangili**, R. Marinucci, A. Ohmura, U. Schlese, P. Tschuck, and M. Wild. "Coupled Simulations of Global and Regional Climate." *Supercomputing Projects Switzerland* (1993).

Beniston, M., R. Marinucci, M. Wild, P. Tschuck, L. Bengtsson, U. Schlese, M. Esch, F. Giorgi and **A. Bernasconi**. "Coupled Model Studies of Future Climate Trends Over the Alpine Regions." *Crosscuts* 2(3): 1, 10–13.

Claxton, T. A., **D. M. Maric** and P. F. Meier. "Dynamic Model for the Structure of Bond–Centered Muonium in Silicon." *Phys. Rev. B* 47 (1993): 13314.

Colombo, L. and **D. M. Maric**. "Tight Binding Molecular Dynamics on the NEC SX–3." *Crosscuts* 2(2): 15.

Decker, K. M. and **R. M. Rehmann**. "SPADE: Eine Programm– und Applikationsentwicklungsumgebung für Multiprozessorsysteme mit verteiltem Speicher." *AGEN Mitteilungen* 56/57 (June 1993): 19–22.

Deloff, A., **P. Flükiger** and J. Weber. "Combined Visualization of Contour Levels and 3D Volumes in Molecular Graphics." p 424 in *Communicating with Virtual Worlds*, edited by N. M. Thalmann and D. Thalmann. Tokyo: Springer, 1993.

Dvorak, J. and H. Bunke. "Using CLOS to Implement a Hybrid Knowledge Representation Tool." pp 295–320 in *Object–Oriented Programming: The CLOOS Perspective*, edited by A. Paepcke. Cambridge, MA: MIT Press, 1993.

Estreicher, S. K. and **D. M. Maric**. "What is so strange about Hydrogen Interactions in Germanium?" *Phys. Rev. Letts.* 70 (1993): 3963.

Estreicher, S. K., **D. M. Maric**, P. F. Meier and D. S. Marynick. "Very Large Scale Electronic Structure Calculations with PRDDO." *Crosscuts* 2(3): 5.

Flükiger, P. "MOLEKEL: Advanced, Interactive 3D–Graphics for Molecular Sciences." *Crosscuts* 2(2): 4–6.

Hodous, M. F. "Computer Relaxations on the SX–3." *SX World* 11 (Spring 1993): 3–4.

Gobbetti, E., J. F. Balaguer, **A. Mangili** and R.Turner. "Building an Interactive 3D Animation System." pp 211–242 in *Object–Oriented Applications*. UK: Prentice Hall, 1993.

Jefford, C.W., G. Bernardinelli, **D. M. Maric**, C. Thomson and J. Weber. "Computational Studies of the Structures and Properties of Potential Anti–malarial Compounds Based on the 1,2,4–trioxane Ring Structure." *Int. J. Quantum Chemistry Symposium* (submitted for publication).

Maric, D. M., P. F. Meier and S. K. Estreicher. "H,B, H,C, and H,Si Pairs in Silicon and Germanium." *Phys. Rev. B* 47 (1993): 3620.

PUBLICATIONS

Maric, D. M., M. A. Roberson and S. K. Estreicher. "Relative Stability of H^T vs. H^* and H_2^T vs. H_2^* in c–C, Si, Ge and a–Sn and their Consequences." *Mat. Sci. Forum* (in press).

Maric, D. M. and L. Colombo. "Defect Induced Amorphization in Silicon: A Tight Binding Molecular Dynamics Simulation." *Mat. Res. Soc. Symp. Proc.* (in press).

Paschedag, N., H. U. Suter, **D. M. Maric** and P. F. Meier. "Configuration Interaction Calculation of Hyperfine." *Phys. Rev. Letts.* 70(1993): 154.

Maric, D. M. and C. Thomson. "Electronic Structure and Properties of the Simplest known 1,2,4–trioxane tetramethyl–hydroxymethyl–1,2,4–triox-ane (TMHMT)." *J. Mol. Struct. (THEOCHEM)* (submitted for publication).

Merazzi, S., **R. Gruber**, C. Pfistner and R. Weber. "Numerical Simulation of Mechanical and Optical Properties of Solid–State Lasers." *Crosscuts* 2(1): 6–8.

Pfistner, C., R. Weber, H.P. Weber, S. Merazzi and **R. Gruber**. "Thermal Beam Distortions in Longitudinally Pumped Solid State Lasers." *IEEE Journal of Quantum Electronics* (accepted for publication).

Scheidegger, A. "Wettbewerbsfaktor Informations– und Kommunikationstechnologie." *Informatik Bulletin* ETH Zürich 72 (February 19–21, 1993).

Scheidegger, A. "Research Consortia–Japan's Winning Strategy." *ETH Zürich Bulletin* 250 (July 20–21, 1993).

Tomassini, M. "Parallel Cellular Evolutionary Algorithms for Optimization." *EPFL Supercomputing Review* 5 (1993): 17–21.

Zumthor, B. "Supercomputing at CSCS in Switzerland." *SX World* 12 (Summer 1993): 5–6.

CONFERENCE PROCEEDINGS

Decker, K. M. "Methods and Tools for Programming Massively Parallel Distributed Systems." Presented at the *14th SPEEDUP Workshop on Parallel and Vector Computing*, ETH Zürich (September 16–17, 1993); Speedup Journal 7(2) (to appear).

Neeracher, M. and **R. Rühl**. "Automatic Parallelization of *LINPACK* Routines on Distributed Memory of Parallel Processors." *Proceedings, Seventh IEEE International Parallel Processing Symposium*, Newport Beach, California (April 13–16, 1993): 830–836.

Tomassini, M. "The Parallel Genetic Cellular Automata: Application to Global Function Optimization." *Proceedings, International Conference on Artificial Neural Nets and Genetic Algorithms*, Innsbruck, Austria. Edited by R. F. Albrecht, C. R. Reeves and N. C. Steele. Springer–Verlag (1993): 385–391.

Bonomi, E. and **M. Tomassini**. "The Role of Massively Data–Parallel Computers in Large–Scale Molecular Dynamics Simulations." *Proceeding, 4th International Conference Physics Computing '92*, Prague, Czech Republic. Edited by R. A. de Groot and J. Nadrchal. World Scientific (1993): 275–277.

Tomassini, M. "Massively Parallel Genetic Algorithms for Combinatorial and Function Optimization." *Proceedings, 2nd European Connection Machine Users Meeting*, Paris, France (October 11–14,1993) to appear in *International Journal of Modern Physics.*

von Stürler, E. and D. R. Fokkema. "Nested Krylov Methods and Preserving the Orthogonality." *Proceedings, 6th Copper Mountain Conference on Multigrid Methods*, Copper Mountain, Colorado USA (April 4–9, 1993), published by NASA (CP), NASA Langley Research Center, Hampton, VA.

Wylie, B. J. N., D. McArthur, G. Cameron, M. White and M. Smith. "PARAMICS: Parallel Microscopic Traffic Simulator." *Proceedings, 2nd European Connection Machine Users Meeting*, Paris, France (October 11–14,1993) to be published by World Scientific.

SWITZERLAND

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Swiss Federal Institute of Technology Lausanne

Centre de Recherches en Physique des Plasmas (CRPP)
Research Center for Plasma Physics
21, Av. des Bains, EPFL, CH–1007 Lausanne

Institut de Mesure et Analyse des Contraintes (IMAC)
Institute for Stress Measurement and Analysis
EPFL, GC Ecublens, CH–1015 Lausanne

Institut Romand de Recherche Numérique en Physique des Matériaux (IRRMA)
Research Institute of Material Physics
EPFL, PHB Ecublens, CH–1015 Lausanne

Service Informatique Central (SIC)
Scientific Computing Services
EPFL, PHB Ecublens, CH–1015 Lausanne

Eidgenössische Technische Hochschule Zürich (ETHZ)
Swiss Federal Institute of Technology Zurich

Centro Svizzero di Calcolo Scientifico (CSCS)
Swiss Scientific Computing Center
Via Cantonale, CH–6928 Manno

Geographisches Institut
Department of Geography
Winterthurerstrasse 190 ETH, CH–8092 Zürich

Interdisziplinäres Projektzentrum für Supercomputing (IPS)
Interdisciplinary Project Center for Supercomputing
Clausiusstrasse 59 ETH, CH–8092 Zürich

Laboratorium für Technische Chemie
Laboratory for Technical Chemistry
Universitätstrasse 6 ETH, CH–8092 Zürich

Universität Bern
University of Bern

Physikalisches Institut
Department of Applied Physics
Sidlerstrasse 5,CH-3012 Bern

Université de Fribourg
University of Fribourg
Pérolles, CH–1700 Fribourg

Université de Genève
University of Geneva

Département de Chimie Physique
Department of Physical Chemistry
30, quai Ernest Ansermet, CH–1211 Genève 4

Groupe de Calcul Parallèle
Parallel Computing Group
24, rue Général Dufour, CH–1211 Genève 4

Université de Lausanne
University of Lausanne
CH–1015 Dorigny–Lausanne

Université de Neuchâtel
University of Neuchatel

Institut d'Informatique et d'Intelligence Artificielle
Department of Computer Science and Artificial Intelligence
Chantemerle 22, CH–2000 Neuchâtel

Universität Zürich
University of Zurich

Multimedia Group
Institut für Informatik
Department of Computer Science
Winterthurerstrasse 190, CH–8057 Zürich

Organisch–Chemisches Institut
Department of Organic Chemistry
Winterthurerstrasse 190, CH–8057 Zürich

Physik Institut
Department of Physics
Winterthurerstrasse 190, CH–8057 Zürich

Quantum Optics Group
Physik Institut
Department of Physics
Winterthurerstrasse 190, CH–8057 Zürich

Other Swiss Organizations

Asea Brown Boveri (ABB)
Segelhof, CH–5405 Baden–Dättwil

Bundesamt für Bildung und Wissenschaft (BBW)
Federal Office for Education and Science
Postfach 5675, CH–3001 Bern

Bundesamt für Informatik
Federal Office for Computer Science
Feldeggweg 1, CH–3003 Bern

ADDRESSES

Schweizerischer Nationalfonds zur Förderung der wissenschaftlichen Forschung (SNF)
Swiss National Science Foundation
Wildhainweg 20, CH–3012 Bern

NEC Swiss SX Center
Via Cantonale, CH–6928 Manno

Olsen and Associates
Research Institute for Applied Economics
Seefeldstrasse 233, CH–8008 Zürich

Passera & Pedretti SA
Via Adamini 10, CH–6900 Lugano

Paul Scherrer Institut (PSI)
Badenerstrasse 569, CH–8048 Zürich

Ufficio del Lavoro
Employment Office
Dipartimento delle Finanze e dell'Economia
CH–6501 Bellinzona

International Organizations

World Health Organization
20, av. Appia, CH–1202 Genève

CANADA

University of Montreal

Department of Chemistry
Montreal H3C 3J7

FRANCE

Centre de Recherche et de Formation Avancée au Calcul Scientifique (CERFACS)
Center for Research and Advanced Training in Scientific Computing
42, Av. Gustave Coriolis, F–31057 Toulouse Cedex

CRIL INGENIERIE / INFOROP
TERSUD – Bat. B
Z. I. de la Grande Plaine
5, rue Marcel Dassault, F–31500 Toulouse

Ecole pour l'Informatique et les Techniques Avancées (EPITA)
School for Computer Science and Advanced Technics
106–112, bd. de l'Hôpital, F–75013 Paris

The Intertek Group
94, rue de Javel, F–75015 Paris

Université de Franche–Compté
University of Franche–Compté

Laboratoire d'Informatique de Besançon
Informatics Laboratory of Besançon
Av. de l'Observatoire, F–25030 Besançon

GERMANY

CAD–FEM GmbH
Heidenheimer–Strasse 5, D–71229 Leonberg

Max–Planck Institut für Meteorologie (MPI)
Max–Planck Institute for Meteorology
Bundesstrasse 55, D–2000 Hamburg 13

Westfälische Wilhelms Universität Münster
University of Munster

Institut für Mathematik
Mathematics Department
D–48161 Münster

ITALY

Agusta Spa
I–21017 Cascina Costa di Samarate (VA)

Joint Research Center (JRC)
Commission of the European Communities (CEC)
ISPRA Establishment
ISEI Institute (TP 361)
Via Enrico Fermi, I–21020 Ispra

Università degli Studi di Milano
University of Milan

Dipartimento di Fisica
Department of Physics
Via Celoria 16, I–20133 Milano

Dipartimento di Scienze dell' Informazione
Department of Computer Sciences
Via Comelico 39, I–20135 Milano

SLOVENIA

Institute J. Stefan,
Ljubljana 61000

SPAIN

LABEIN–Centro de Investigation Tecnologica
Technological Research Center
Cuesta de Olabeaga, 16, E–48013 Bilbao

RUSSIAN FEDERATION

Keldysh Institute of Applied Mathematics
Miusskaya Sq. 4, 125047 Moscow

UNITED KINGDOM

Center for Novel Computing
The Victoria University of Manchester
Oxford Road, Manchester M13 9PL

Scientific and Engineering Research Council
Rutherford Appleton Laboratory (RAL)
Chilton, OX11 OOX DIDCOT

UK Meterological Office
London Road, Braknell, GB–Berks RG12 3UE

University of St. Andrews
Department of Chemistry
St. Andrews, KY168SE Scotland

UNITED STATES

Oberlin College
Computer Science Program
Oberlin, Ohio 44074

University of California at Irvine
Department of Electrical and Computer Engineering
Irvine, California 92717

University of Texas at Arlington
Department of Chemistry
Arlington, Texas 76019–0065

Texas Tech University
Department of Physics
Lubbock, Texas 79409–1015

CSCS – Via Cantonale – Galleria 2 – 6928 Manno – Switzerland