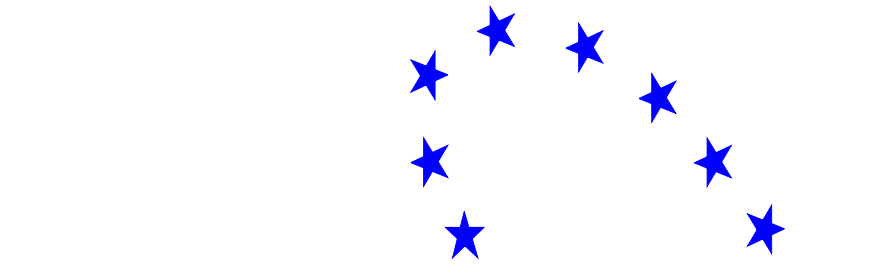
The ZEUS Consortium Massively Parallel Computing 1994

Alexander Reinefeld (Ed.)



ZEUS

Zentren Europäischen Supercomputings

Technical Report PC^2/TR-006-94
April 1994

PC^2 - Paderborn Center for Parallel Computing, Universität-GH Paderborn
D-33095 Paderborn, Germany • Phone: +49 5251 60 3342 • email: pc2-team@uni-paderborn.de

Contents

1 Preface

Massively-parallel high-performance computing has meanwhile reached a point where it is no longer confined to purely academic research applications or those of the so-called *Grand-Challenges* class which, by definition, require computational power that can only be provided by large-scale massively parallel processing (MPP) systems. Due to the increased maturity of the system technology in both hard- and software, as well as due to visible successes of intensive research on parallel algorithms and methodologies, massively-parallel HPC is now more and more seen as a feasible and valuable tool for the solution of compute-intensive problems in industrial and commercial applications. An increasing number of industrial research and standard application codes is being ported to massively-parallel systems, and there is much more potential in this direction that can be exploited.

This paper reports about MPP projects that have been performed in the *ZEUS Centers of EUropean Supercompting*, consisting of the four founding members PC^2, IC^3A, ZPR, AHPC and the associated member Daresbury Laboratory. Taken as a whole, the ZEUS Consortium can be considered as a 'Meta-Computing-Center', offering state-of-the-art (European) parallel processing technology throughout Europe in a homogeneous way. ZEUS provides both a unique combination of know-how, expertise, and proven experience in a broad range of application areas, as well as a unique setup of available computing resources amongst which are Europe's largest MIMD computer systems (1024- and 512-processor Parsytec-GCel systems)

The ZEUS activities and projects cover all levels of the often quoted HPC technology-chain spanning from system software and technology (e.g., work on the architecture of distributed memory systems, virtual topology libraries for PARIX) over generic parallel algorithms and advanced software tools to specialized and standard industrial applications.

Most of the work reported here was supported by the CEC in the framework of Workpackage 6 of the (amended) GP-MIMD project, which has been instrumental in the process of establishing the ZEUS network. Other tasks have been supported by local government grants, by industry contracts, or are performed on an individual research basis, finally resulting in diploma- and Ph.D.-theses.

The present list of projects does by no means attempt to be complete. Its intention is to give the reader a first idea about the ongoing activities at the ZEUS centers and the broad range of possible application areas for large-scale massively-parallel processing. In fact, many more projects from various fields are currently

in progress at the different Centers. Reports on these projects will successively be included in forthcoming versions of this publication.

Many people helped in setting up the ZEUS network and putting it into operation. The 'Parsytec People', who original invented the ZEUS idea, deserve special credit. Most notably, Rainer Czech, Lambis Tassakos, Mahmoud Chatah, Fritz Lücking and many others at Parsytec supported the ZEUS initiative by establishing first contacts to potential MPP users and providing general support. The ZEUS center managers (see below) successfully collaborated and provided valuable input for this first status report. Finally I am indebted to Birgit Farr, who put my scribblings into a readable form.

Preface to the Second Edition

More than five hundred copies of the first edition *The ZEUS Consortium – Massively Parallel Computing 1992/93* are in circulation by now. We received many positive feedback from research, development and application in industry and academia. The report proved to be more than just an "address book" of MPP projects – it gives an answer to the question *what is going on in the ZEUS MPP scene?*

As can be seen just by the volume of the second edition, more projects have been added in the meantime. While this mere fact can be taken as a positive sign, more important – and encouraging – is the steadily growing number of successful projects that have been carried out jointly with (or for) industrial partners.

Paderborn, April 1994 Alexander Reinefeld

2 The ZEUS Consortium

The *ZEUS Consortium – Zentren EUropäischen Supercomputings* – comprises a core of four cooperating European centers for massively parallel high-performance computing. Together with another associated center, ZEUS was officially inaugurated in November 1992 in Paderborn, Germany. At the present time, the following centers are collaborating in the ZEUS context:

- **AHPC Lab** – Athens High Performance Computing Laboratory, Greece,
- **IC^{3}A** – Interdisciplinary Center for Complex Computer facilities Amsterdam, The Netherlands,
- **PC^2** – Paderborn Center for Parallel Computing, Germany,
- **ZPR** – Center for Parallel Computing at the University of Cologne / DLR, Germany,

and, as an associated center to the ZEUS consortium,

- **Daresbury Laboratory**, U.K.

2.1 A Brief Overview

The ZEUS Consortium was established to optimally exploit the potential of the vast range in know-how and expertise in the field of MPP and also to create the opportunity to enter new fields of research, that a single center alone could hardly address. The consortium successfully collaborates in various common and bilateral projects (cf. below), organizes joint workshops, colloquia and seminars, and publishes conference proceedings, and technical reports.

At the present time, the ZEUS network comprises the following centers:

Athens HPC Lab has been established in September 1993 as a result of a cooperation between the University of Athens, the National Technical University of Athens, and the Institute of Computer Technology (CTI) of the University of Patras. It is managed by a board of directors with representatives from the three institutions, Prof. C. Halatsis (Univ. of Athens), Prof. K. Papailiou (NTUA), and Prof. D.G. Maritsas (CTI). The main research and project activities are in the fields of programming environments, new generic parallel algorithms, and industrial applications.

IC^{3}A, officially inaugurated in May 1993, is a joint initiative of the University of Amsterdam (Faculty of Computer Science and Mathematics) and CWI (Center for Mathematics and Computer Science). IC^{3}A is managed by Prof. L. Hertzberger and Dr. P.M.A. Sloot. It is of interdisciplinary nature, and open to national and international users; strong emphasis is put on the computer supported research of complex systems in science and engineering and its application. IC^{3}A's own activities are centered around generic solving methods, large scale applications and the 'meta computing' environment.

PC^2 is an interdisciplinary research and service institute of the University of Paderborn; it is managed by Prof. B. Monien and Dr. A. Reinefeld. Based on its strong ties with the Computer Science Department of the University, one of PC^2 's main goals is to be a focal point between theory and practice of parallel computing. PC^2 is involved in many projects with industrial partners (e.g., Parsytec) and closely collaborates with ZIAM (Center for Industrial Applications of Massive Parallelism), a knowhow-transfer center founded 1992 in Northrhine-Westfalia.

ZPR is a joint undertaking of the University of Cologne and DLR, the German national aerospace research agency. The center is managed by Prof. A. Bachem and Dipl.-Math. M. Wottawa from the Mathematics Department of the University and Dr. H.-M. Wacker, director of the DLR's nationwide computer center. As the other centers, ZPR is of interdisciplinary nature. A large portion of the research and project work is in the areas of applied mathematics, computer science, fluid mechanics, economical science and linguistics. The porting of several large production codes used by DLR is another major activity.

Daresbury Laboratory, associated ZEUS member, is managed by the U.K. Engineering and Physical Sciences Research Council (EPSRC). Daresbury Laboratory (formely SERC) comprises of some 500 permanent and contract staff, and offers

large scale science-based administrative, experimental (the Synchrotron Radiation Source) and computing services. Daresbury Laboratory has a long experience of porting both academic and industrial applications to novel architecture computers especially for ICI plc, and runs a National Parallel Supercomputer Service based on a 64-node Intel iPSC/860 Hypercube. The 'Theory, Computational Science and Computing Programme' help industrial users and academic groups in the Collaborative Computational Projects (CCPs) to maintain large-scale application codes in areas such as quantum chemistry, molecular modeling and simulation, solid-state and surface science, atomic scattering, plasma physics and computational fluid dynamics.

2.2 Contact Addresses

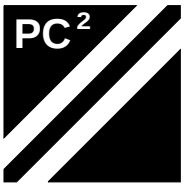
AHPC Lab – Athens High Performance Computing Laboratory

Prof. Dr. Constantin Halatsis
University of Athens
Department of Informatics
Panepistimiopolis, TYPA Building phone: +301 72 17 941
157 71 Athens fax: +301 72 19 561
Greece email: halatsis@uranus.di.uoa.ariadne-t.gr

IC^{3}A – Interdisciplinary Center for Complex Computer Facilities Amsterdam

Prof. Dr. L.O. Hertzberger
Dr. P.M.A. Sloot
Faculty of Mathematics & Computer Science
University of Amsterdam Kruislaan 403
NL-1098 SJ Amsterdam
The Netherlands
phone: +31 20 525 7463
fax: +31 20 525 7490
telex: 10262 hef nl
email: peterslo@fwi.uva.nl

PC^2 – Paderborn Center for Parallel Computing



PADERBORN
CENTER FOR
PARALLEL
COMPUTING

Prof. Dr. Burkhard Monien
Dr. Alexander Reinefeld
Paderborn Center for Parallel Computing
Universität-GH Paderborn
33095 Paderborn / Germany
phone: +49 5251 60 3342
fax: +49 5251 60 3436
email: ar@uni-paderborn.de

ZPR – Zentrum für Paralleles Rechnen
Center for Parallel Computing at the University of Cologne / DLR

Zentrum für Paralleles Rechnen

Prof. Dr. Achim Bachem
Michael Wottawa
Universität zu Köln
Mathematisches Institut
Weyertal 80
50923 Köln / Germany
phone: +49 221 470 4341
fax: +49 221 470 5160
email: wottawa@mi.uni-koeln.de

3 Scientific Applications

This section covers a broad spectrum of scientific applications that have been run on large scale massively parallel computing systems installed at the ZEUS centers. The codes are mainly used in research and have hardly any direct commercial relevance. Examples are codes used in the disciplines

- high-energy physics
- geophysics
- astrophysics
- environmental sciences
- molecular biology
- chemistry
- operations research
- computer simulation
- computational fluid dynamics
- computer graphics

3.1 CFD with parallel FEM

An explicit finite element scheme based on a two step Taylor-Galerkin algorithm allows the solution of the Euler and Navier-Stokes Equations for a wide variety of flow problems. To obtain useful results for realistic problems one has to use grids with an extremely high density to get a good resolution of the interesting parts of a given flow. Since these details are often limited to small regions of the calculation domain, it is efficient to use unstructured grids to reduce the number of elements and grid points.

The principle idea for the parallelization of a grid based algorithm is a division of this grid into parts which can be calculated separately on different processors. In our case this is possible, because we use a time stepping algorithm where in every time step the calculations for one element only depend on the values of elements which have nodes in common with this element. All inner nodes of a separate part of the grid can be calculated independently from other parts. The border nodes of every part must then be corrected after each time step using interprocessor communication.

To get an efficient program with this kind of parallelization, which is known as data decomposition, we have to choose a logical topology for the processor network, which is suitable for the given problem. Our research has shown that for the two-dimensional grids we are using a 2D-mesh is a good choice. As the actual version of this algorithm was implemented under the Parix programming environment, this topology can be mapped in a canonical way on the hardware topology.

The essential part of the whole program is a decomposition algorithm which reads in the global data structures, calculates a division of the grid and distributes the corresponding local data sets to the appropriate processor of the parallel system. The division algorithm, which determinates the quality of the resulting distribution, has to take different facts into consideration to achieve efficient parallel calculations. First of all it must ensure that all processors have nearly equal calculation times, because idle processors slow down the speedup. To achieve this we first distribute the elements as evenly as possible and then minimize the overhead caused by double calculated nodes and resulting communications. Such a division often requires interprocessor connections which are not supplied by the basic logical topology. These connections are built dynamically with the virtual links of Parix.

Empirical performance measurements show linear speedups for up to 256 processors and efficiencies which drop down to 50 % for 1024 processors. The reason for the last effect is the used a-priori-division of the grid which will be improved in future versions by adding a dynamic optimization strategy.

References

[1] F. Lohmeyer, O. Vornberger, K. Zeppenfeld, A. Vornberger. *Parallel Flow Calculations on Transputers.* International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 1, pp. 159-169, 1991.

[2] F. Lohmeyer, O. Vornberger. *CFD with parallel FEM on Transputer Networks.* Flow Simulation with High-Performance Computers I, Notes on Numerical Fluid Mechanics, Vol. 38, pp. 124-137, Vieweg, Braunschweig 1993.

[3] F. Lohmeyer, O. Vornberger. *Flow Simulation with FEM on massively parallel Systems.* To be published in: Notes on Numerical Fluid Mechanics, Vieweg 1994.

Contact Address

Prof. Dr. Oliver Vornberger
Dipl.-Math. Frank Lohmeyer
Universität Osnabrück
FB Mathematik / Informatik
D-49076 Osnabrück

phone: +49 541 969 2392
fax: +49 541 969 2770
email: lohmey@informatik.uni-osnabrueck.de

3.2 Chess

The parallel Paderborn chess program "Zugzwang" was ported from the SC-320 to the GCel-1024. It is written in OCCAM and runs on deBruijn like networks as well as grid topologies of various sizes. Due to inherent imbalances in the work load, the underlying *Alpha-Beta* tree pruning algorithm is difficult to parallelize. Even though, high speedups were achieved on the following topologies:

- a speedup of 237 on a $512 = (32 \times 16)$ grid network
- a speedup of 344 on a $1024 = (32 \times 32)$ grid network

This is the best speedup reported so far for a successful chess program. DeBruijn networks have been found to be better suited for parallel Alpha-Beta search than tori, while these are again better than the simple grid structures without wrap around links [2].

Zugzwang successfully played several tournaments against other programs as well as against humans. At the World Computer Chess Championship in Madrid last year, it finished second place running on 1024 processors.

References

[1] R. Feldmann, P. Mysliwietz, B. Monien. *Game tree search on a massively parallel system.* 7th Advances in Computer Chess Conference, Maastricht, (July 1993).

[2] R. Feldmann. *Spielbaumsuche mit massiv parallelen Systemen.* Dissertation, UNI-GH Paderborn, Juli 1993.

Contact Address

Prof. Dr. B. Monien
Dr. R. Feldmann, Dr. P. Mysliwietz
Universität-GH Paderborn
FB 17 Mathematik/Informatik
D-33095 Paderborn

phone: +49 5251 603332
fax: +49 5251 603436
email: chess@uni-paderborn.de

3.3 Computer Vision

We have implemented a library of low-level image processing algorithms. This library is divided into two families of algorithms, one for those that apply to the Spatial Domain, and one for those that apply to the Frequency Domain.

1. Algorithms in the Spatial Domain

   (a) Local Histogram Equalization

   (b) Local Average Filter

   (c) Median Filter

   (d) Sobel Edge Detector

   (e) Histogram Evaluation

   In all cases, the performance curves form a "knee bend", after which execution time begins to increase, because it is overwhelmed by the communication overhead. The execution time starts to increase when the number of processors becomes greater than 256 in the case of median filtering and local histogram equalization. In the case of local averaging, the knee bend appears when the number of processors becomes 128, for $k = 5$ and 7. The worst cases are those of the Sobel edge detector and histogram computation, where the knee bend appears when the number of processors is only 32.

   The efficiency of the computation depends on:

   (a) The number of processors.

   (b) The method of combining results produced by different processors: for example, sequentially, or using a binary tree.

   (c) The time required for the combination of two independently produced results, compared to the time required to produce them.

2. Algorithms in the Frequency Domain

   (a) Forward and Inverse Discrete Fourier Transform

   (b) Amplitude of the Forward Discrete Fourier Transform

   (c) Forward and Inverse Discrete Cosine Transform

   (d) Butterworth filters

The performance data of these transforms and filters show that the execution time remains roughly the same, regardless of the number of processors employed, producing only small, irregular variations around a fixed value. This is more prominent in the cases of FFT and Butterworth filters, with only the DCT case showing some similarity to the previous family of performance curves. This behavior was caused by the memory restrictions imposed by the GCel machine.

References

[1] Howard Jay Siegel, James B. Armstrong, and Daniel W. Watson. *Mapping computer vision-related tasks onto recongurable parallel-processing systems.* IEEE Computer, pages 54-63, February 1992.

[2] John A. Webb. Steps toward architecture-independent image processing. IEEE Computer, pages 21-31, February 199

[3] Alok Choudhary and Sanjay Ranka. *Parallel processing for computer vision and image understanding.* IEEE Computer, pages 7-10, February 1992.

[4] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing.* Addison-Wesley Publishing Company, 1987.

[5] William Pratt. *Digital Image Processing.* John Wiley & Sons, Inc,, 1991.

[6] Anil K. Jain. Fundamentals of Digital Image Processing. Prentice-Hall International, Inc., 1989.

[7] Jae S. Lim. *Two Dimensional Signal and Image Processing.* Prentice-Hall International Editions, 1990.

[8] D.G. Maritsas. *Report on the Computer Vision Algorithms for the Parsytec GCel 3/512.* Internal Report, University of Athens, CTI, 1993.

Contact Address

Prof. D.G. Maritsas
University of Athens
Department of Informatics
Panepistimiopolis, TYPA Building
157 71 Athens / Greece

phone: +30-61-225073
fax: +30-61-222086
email: maritsas@cti.gr

3.4 Crystallization

Parallel simulated annealing (SA) has been applied to simulations of crystallization of particles on a spherical surface [1]. The sequential version of the SA program is too time consuming for problems greater that a few hundred particles.
There are two methods to parallelize the work:

1. Systolic SA [2]:
   In a systolic SA a Markov chain is assigned to each of the available processors. All chains have equal length. The chains are executed in parallel and during the execution information is transferred from a given chain to its successor. Each Markov chain is divided into a number of subchains equal to the number of available processors, in order to be able to let processors execute Markov chains simultaneously.

2. Functional decomposition:
   We have also implemented a functional decomposition of the energy calculation. Since the calculation of the energy chance is a calculation the order of the number of particles, it is one of the most time consuming problems. We have decomposed the energy calculation in a farm like manner and have implemented it in a tree configuration on the transputer surface in order to save on communication costs.

   Both implementations are now in a working state on the Parsytec machine. The functional decomposition provides an easy speedup. With a tree of 7 nodes and a problem size of 1000 particles the speedup is about 5.7. The systolic annealing algorithm can provide much more speedup but the solution can become worse if the problem size is to low for a large number of processors.

References

[1] P.M.A. Sloot, A. ter Laak, P. Pandolfi and R. van Dantzig. *Crystallization on a sphere: Parallel simulations on a transputer network.* Accepted for publication in Int. J. Modem Physics C.

[2] Kim and Kim. *A step wise overlapped parallel annealing algorithm on a message passing multiprocessor system.* Concurrency: Practice and experience, Vol. 2(2), June 1990, pp. 123-148.

[3] A. ter Laak, L.O. Hertzberger and P.M.A. Sloot. *Nonconvex continuous optimization experiments on a transputer system.* in Transputer systems-ongoing research, ed. Alistair Allen, WoTUG 15, IOS Press 1992, ISSN: 0925-4986, 1992.

Contact Address

Dr. P.M.A. Sloot
Dept. of Computer Systems
Universiteit van Amsterdam
Kruislaan 403
1098 SJ Amsterdam / The Netherlands

phone: +31 20 525 7463
fax: +31 20 525 7490
email: peterslo@fwi.uva.nl

3.5 Digital Signal Processing

The goal of the project was to extend, evaluate and assess highly concurrent digital processing algorithms. A fairly general formulation has been employed covering generic signal processing tools that find application in multichannel filtering, multichannel signal modeling and multi-input and multi-output system identification. Emphasis was focused on Shur type algorithms featuring high degree of parallelism. Finally, certain non-linear models techniques based on finite Volterra expansions were studied.

New fully parallel Shur type algorithms of 2 or 3 dimensions have been developed, programmed in C and evaluated. Some of these parallel algorithms for filtering and identification have previously been programmed in Occam and implemented on a transputer based 16 node machine. The limited number of processors of this machine has forced the team to assign one processor for the execution on many processes. All these algorithms were ported, tested and evaluated on the massively parrallel enviroment of GCel 512. The main problem observed was the input-output interface which is carried out by a single processor. In some algorithms the communication overhead was very painfull leading us to use clustering. In addition a general tool for performing a number of useful digital signal processing transforms such as DCT (Discrete Cosine Transform) and DWT (Discrete Wavelet Transform) is developed and some cases were implemented tested and evaluated on GCel 512 parallel enviroment. Finally, the computation of Volterra system output was implemented on the Parsytec machine and the resulting data were compared to those resulting from serial implementation.

References

[1] G. Glentis and N. Kalouptsidis. *Efficient Order Recursive Algorithms for Multichannel Least Squares Filtering*, IEEE Transactions on Signal Processing, pp. 1354-1374,June 1992.

[2] P. Koukoulas and N. Kalouptsidis. *Nonlinear System Identification Using IID and/or Gaussian Inputs*, ATHOS Workshop, pp. 115-122, Sophia Antipolis, France, September 1993.

[3] I. Bouras, G. Glentis and N. Kalouptsidis, *Efficient implementation of block toeplitz solvers in VLSI array processors,* IMAS/PDCOM Corfu, Greece 1991.

[4] S.Y. Kung, *VLSI Array Processors*, Prectice Hall, 1988.

Contact Address

Prof. N. Kalouptsidis
Department of Informatics
Panepistimiopolis
TYPA Building
157 71 Athens / Greece

phone: +30-1-72-30172, 17941, 91885
fax: +30-1-72-19561, 28981
email: kalou@zeus.di.uoa.ariadne-t.gr

3.6 DSMC Method of Gas Flow

The DSMC (Direct Simulation Monte Carlo) Code from the DLR Göttingen is for the simulation of axisymmetric flows in a domain defined by an arbitrary structured cell grid. The physical modeling in the code includes high-temperature phenomena, mainly excitation of vibrational degrees of freedom and complete air chemistry with dissociation and recombination reactions.

In the DSMC method a gas flow is computed by simulating the dynamics of a representative number of gas particles in a chosen domain which is subdivided into cells. In the simulation the particles are advanced according to their individual velocities. Possible boundary interactions of the particles are computed for a large number of time steps.

The code is currently reformulated for parallel processing on a Parsytec GCel-1024. The computational load is distributed among the processors by decomposing the computational domain into regions, whereby each region is assigned to one processor node. As a first step, a simple domain-decomposition in form of a one-dimensional band of segments will be implemented. But the code will be written such that the extension to more complex grid topologies can be made easily. Program modules have been developed which handle the transfer of simulated particles between regions efficiently.

Contact Address

Frank Bergmann
DLR Institute of Fluid Mechanics
Bunsenstr. 10 phone: +49 551 709 2415
37073 Göttingen email: bergemann@ts.go.dlr.de

Dr. Andrea Coriand
DLR High Performance Computing
Bunsenstr. 10 phone: +49 551 709 2155
37073 Göttingen email: coriand@dlr.de

3.7 Elastic Light Scattering

The parallel implementation of the *Coupled Dipole method*, ported to C running on Parix, was applied for extensive production runs. During the report period (July 1993) more than 200 hours of production on the full 512 processor domain were run. The experiments addressed simulation of *Elastic Light Scattering* from spheres, located anywhere in a focussed laserbeam. This is a new test of the Coupled Dipole method, an very important to be able to simulate scattering from particles in a Flow Cytometer. The simulation results agree very good with analytical calculations using the so-called generalized Lorenz-Mie theory.

Exhaustive performance measurements were performed. It was shown that the parallel efficiency of the implementation can be very close to 1, even on the full domain of 512 transputers. The achieved computational speed of this application (double precision numbers!) running on the full domain is estimated at 250 Mflop/s.

Some first production runs were carried out, containing 35.000 dipoles. These runs lasted 11 to 22 hours on 512 transputers.

References

[1] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet and C.G. Figdor. *Scattering matrix elements of biological particles, measured in a flow through system: Theory and practice.* Applied Optics 28, 1989, 1752.

[2] A.G. Hoekstra, P.M.A. Sloot, W. Hoffmann and L.O. Hertzberger. *Time Complexity of a Parallel Conjugate, Gradient Solver for Light Scattering Simulations: theory and SPMD implementation.* Technical Report CS-92-06, University of Amsterdam, Faculty of Mathematics and Computer Science, 1992.

[3] Poster at congress. *Third International Congress on Optical Particle Sizing.* Yokohama, Japan, August 23-26, 1993.

[4] A. Hoekstra. *Computer Simulations of Elastic Light Scattering. Implementation and Applications.* Univ. Amerstdam, Ph.D. dissertation, 1994.

Contact Address

Prof. L.O. Hertzberger
Dr. Alfons Hoekstra
Parallel Scientific Group — phone: +31 20 5257463
Universiteit van Amsterdam — fax: +31 20 5257490
1098 SJ Amsterdam / The Netherlands — email: alfons@fwi.uva.nl

3.8 Factorisation

The factorisation of large numbers (100 digits and more) is of eminent importance for the security of all factoring based cryptosystems, for example the RSA public–key scheme. Concerning practical cryptographic aspects the 'Multiple Polynomial Quadratic Sieve' (MPQS) algorithm is the best currently known general factorisation algorithm. Two alternative parallelisations particularly suited to MIMD parallel computers of the most time-consumptive part have been developed. The first one separates the initialisation of the polynomials from the sieving part, and has been actually used for the factorisations below. The second one distributes the data among subsets of processors and requires a different communication. The third step of the MPQS algorithm consists of the factorisation of about a million of partial solutions (which are numbers smaller than $1.9 \cdot 10^{19}$) and has been parallelised, too. The inversion of a very large 0,1–matrix will be parallelised and ported to Parix in the near future.

Starting from scratch by developing a multiprecision arithmetic package for the T805 processor, numbers in the 39 to 91 digit range have been factored. On December 24th the formerly unknown factorisation of a 97 digit composite number from the Cunningham project was finished. The total running time was below two days which is excellent.

References

[1] F. Damm, F.-P. Heider, G. Wambach. *MIMD-Factorisation on Hypercubes.* University of Cologne, Applied Mathematics and Computer Science, Report No. 94.149 (1994).

[2] R. Peralta. *A quadratic sieve on the n–dimensional cube.* Advances in Cryptology, Crypto '92, Lecture Notes in Computer Science 740, 324 (1993).

[3] R.D. Silverman. *The Multiple Polynomial Quadratic Sieve.* Mathematics of Computation, 48, 177, 329 (1987).

Contact Address

Frank Damm, Georg Wambach
Priv.-Doz. Dr. Franz–Peter Heider
Universität zu Köln — phone: +49 221 470 5308
Institut für Informatik — fax: +49 221 470 5317
50923 Köln / Germany — email: gw@informatik.uni-koeln.de

3.9 GEANT

The Collège de France Parallel Computing group is working in simulation of *high-energy physics particle detectors* via GEANT, the wider used Fortran code in this field. GEANT code can be parallelized in different ways, from trivial and most efficient farm parallelization to inside event parallelization (geometrical an track parallelization). After an extensive work in Telmat's 64 processor T-Node machine, the research group of Prof. Maillard ported their code to the GCel-1024 in Paderborn and performed first large-scale production runs in the PC^2 's Summer Campp 1992. When the GCel-512 came into operation at the University of Amsterdam, Prof. Maillard's group concentrated on

- farm testing and implementation of large codes,
- track and inside event parallelization.

First results show that even with the whole 512 processors at work, efficiency is very high for farming. Inter-processor communication has little influence on the computing time per transputer. As an example, execution times for GEXAM1 program are the following (E=10 GeV, cuts=0.01):

processors	CPU time / event
16	2.2
32	1.1
192	0.17
512	0.063

Contact Address

Prof. Jacques Maillard
Jorge Silva
Collège de France
Laboratoire de Physique Corpusculaire
11 Place Marcelin Bertelot
Paris 15^e / France

phone: +33 1 44271520
+33 1 44271530
fax: +33 1 43546989
email: maillard@cdfvax.in2p3.fr
silva@cdfvax.in2p3.fr

3.10 Graphics and Animation

The *par*allel computer graphics and *i*nteractive solid *mod*eling system *parimod* was established at the University of Osnabrück in 1991. The intention of this development is to produce photorealistic images and animate them movie like. The result is a massively parallel system for the fast and interactive construction, calculation and displaying of arbitrary three-dimensional scenes in different shapes.

In general *parimod* consists of two components. On the one hand there are C-programs and X-applications running under UNIX on frontend workstations, on the other hand there are parallel algorithms running on a transputer system. Under UNIX the X-application *ism* (*i*nteractive *s*olid *m*odeler) is used to construct three-dimensional scenes. Like an architect the user can design the scenes in ground and front view by placing basic objects like e.g. cubes, spheres or cylinders. All other parameters like the point of view, the colour and shape of the materials, the places and parameters of the light sources or the quality of the shading are also choosen with the mouse.

For the fast calculation of the scenes, three parallel programs have been developed: *pcg* (*p*arallel *c*omputer *g*raphic), *pray* (*p*arallel *ray* tracing) and *panim* (*p*arallel *anim*ation). They are all written in OCCAM2 and running on Super Cluster transputer systems. The resulting images can both be displayed on another X-Terminal or in true colour mode on the Graphical Display System GDS that is directly integrated in the Transputer system.

pcg is a parallelization of the classical viewing pipeline. The scenes can be rendered in wireframe, flat, Gouraud or Phong shading. To supress hidden faces the extremely memory consuming z-buffer algorithm is used. For the parallelization the device coordinate system is decomposed. Therefore the screen is divided into small rectangles and each processor calculates some of these parts of the image. The synchronization is realized by a problem message that is permanently routed through the network. For the network a de Bruijn topology is used because of its Hamilton circle, its scalability and its constant connectivity. Every idle processor which has already calculated a part of the image can get the coordinates for the next part that has to be calculated from this message. The load balancing of this parallelization is automatically very good. Another advantage is that the first parts of the image are displayed after a very short time. To overcome the disadvantage that the whole scene has to be generated for each part a second strategy was developed. Here a special transputer generates the whole scene. The other transputers are responsible for the rendering of the scene. Each of them waits for triangles and lines from the generator to render them into its part of the image. The average efficiency of the parallel algorithm reaches more than 60 percent, but depends on the single scene very much.

The parallel ray tracer *pray* divides the calculation line by line under the processors. The result of the calculation is sent either to a X-terminal via the host-transputer or to the GDS. The shortest path to both processors are calculated by each transputer by evaluating a spanning tree. So this parallelization is independent

of the topology of the network. The speedup of *pray* is nearly linear.

For walk-through-animations the *parimod* system contains the animation tool *panim*. *panim* calculates both the sequence of images and each image in parallel. Therefor a special pipeline topology was developed to get the best benefit out of the hardware and software quality of the transputer. In each pipe one image of the sequence is calculated in parallel and sent to the GDS, where it is displayed by a fast block copy function into the frame buffer. With twelve pipes of five transputers each in our system it is possible to display simple scenes of 256×256 true color pixels with a speed of up to 18 images per second. The lighting is done by the progressive refinement radiosity method. The user can steer the camera flight through the scene by key presses that are distributed to all pipes. The synchronization of the images and the pipes is a very challenging problem in this massively parallel animation.

References

[1] C. Landwehr, F.M. Thiesing, O. Vornberger, K. Zeppenfeld. *parimod – The parallel computer graphics and interactive solid modeling system.* Transputer Research and Applications 6 – Proceedings of the Sixth Conference of the NATUG, Vancouver, Canada, pp. 39-52, IOS Press 1993.

[2] K. Zeppenfeld. *Parallele Computergraphik und Animation mit Transputern.* Deutscher Universitäts Verlag 1993.

[3] K. Zeppenfeld, C. Landwehr, F.M. Thiesing, O. Vornberger. *Das parimod-System (Parallele Computergraphik und Animation mit Transputern).* Proceedings of the Transputer Anwender Treffen (TAT 93), Aachen, to be published in: Parallele Datenverarbeitung mit dem Transputer, Reihe Informatik aktuell, Springer 1994.

Contact Address

Prof.Dr. Oliver Vornberger
Dipl.-Math. Frank M.Thiesing
Universität Osnabrück — phone: +49 541 969 2558
FB Mathematik / Informatik — fax: +49 541 969 2770
D-49069 Osnabrück — email: frank@informatik.uni-osnabrueck.de

3.11 Grid Optimization

Grid optimization is a recently developed technique, which is oriented to cover grid quality and computational inefficiency problems, concerning the commonly used grid generation methods (See Thompson et al (1982), Thompson (1984)). Problems are often present in techniques that solve partial differential equations or use conformal mapping (see for example Saltzman et al (1982), Brackvill et al (1982)), which suffer from discretization errors. Some of these problems could be removed by use of grid optimization techniques, which from their nature consider grids consisted of straight line segments joining the nodal grid points and are of the discrete type. The grid optimization method, has demonstrated the ability to improve drastically the quality of any arbitrary unacceptable grid.

In grid optimization, grids are generated and/or optimized so that a global cost function which expresses desirable local grid properties (such as orthogonality, smoothness, node clustering) is minimized, using principles of non-linear programming.The problem is solved usually iteratively, by using a conjugate gradient algorithm (Flether-Reeves version), with analytical calculation of the gradient and exact line search.

The parallelization strategy is based on the domain decomposition method. The domain is subdivided into non-overlapping subdomains and each subdomain is assigned to one processor.

Two kinds of communication between processors are required: First, a local communication between neighboring processors in order to exchange the values of the parameters of their boundary nodes which are needed for accurate calculation in each processor. Secondly, a global communication by the mean of the calculation of overall domain scalars (such as the norm of the gradient and the line step parameter.

Presently both the local and global communication are implemented in a 2-D topology (similar to physical one) based on establishment of virtual links. Both kinds of communications use synchronous send and receive. While the local communication has been optimized so that odd and even processors are complementary senders and receivers in order to decrease communication time (there will be no difference in an asynchronous version), the global communication is recognised as an initial unoptimized one. It must be noticed that before each processor send a receiving value adds its own value. The procedure is integrated with the opposite pattern.

Work has to be done in order to improve the global communication strategy, i.e. to minimize the time needed to calculate and exchange global information. A possible better approach could be based on the same topology but considering as a master processor the processor which is in the center of gravity of the topology, or with the establishment of different topology for the global communication.

Results. The parallel code has been tested both for its integrity and effectiveness. The chosen test case has been a square mesh with disturbed internal nodes in order to diverge from desired properties of smoothing and orthogonality. Four meshes

with sizes 20x20, 40x40, 80x80 and 160x160 have been considered.

It must be underlined that for the 160x160 mesh each iteration spent 38.3 sec in a uniprocessor environment, so that a considerable time is required for the production of the final optimized mesh, while the corresponding time is 200 times reduced for the same grid when 256 processor are used.

Speedup and efficiency are values are quite impressive (29.6/32=0.93, 57.6/64=0.90, 108/128=0.8, 199/256=0.774 for the 160x160 mesh, 14.54/16=0.91, 27.4/37=0.86, 51.4/64=0.8 for the 80x80 mesh).

Conclusions. An existing grid optimization technique has been parallelized and implemented in a multitransputer environment such as the GCel 3/512 machine of the AHPCL in Athens. The effectivness of the parallelization was investigated and found very satisfactory.

It is envisaged, that the way parallelism was incorporated (i.e through both high level routines and low level communication primitives), makes it possible for the code to be easily ported to other massively parallel environments with a little effort.

This will facilitate any benchmarking attempts and comparative studies concerning different computing platforms. It might be worth mentioned that the parallel characteristics of the code (need for both local and global communication) make it suitable for performing such benchmarks.

References

[1] Thompson,J.F. and Warsi, V. *Boundary-fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review,* Journal of Computational Physics, 47, 1-108 (1982).

[2] Thompson,J.F. *Grid Generation Techniques in Computational Fluid Dynamics,* AIAA Journal, Vol. 22, No 11 (1984).

[3] Brackbill,J.U. and Saltzman,J.S. *Adaptive Zoning for Singular problems in Two Dimensions,* Journal of Computational Physics, Vol. 46, pp. 342-368 (1982).

[4] Saltzman, J. and Brackbill, J. *Applications and Generalization of Variational Methods for Generating Adaptive Meshes*, Numerical Grid Generation, Ed. Thompson,J.F., Elsevier Science Pub. Co,1982, pp. 865-884.

[5] Kennon, R. and Dulikravich, G.S. *Generation of Computational Grids Using Optimization,* AIAA Journal, Vol. 24, No 7.(1986)

[6] Kennon, R. and Dulikravich, G.S. *Composite Computational Grid Generation of Computational Grids Using Optimization*, First International Conference on Numerical Grid Generation for Computational Fluid Dynamics, Ed. Hauser, j., College of Landshut, Landshut, Germany, 1986.

[7] Gill, P.,Murray, W. and Wright, M. *Practical Optimization*, Academic Press, 1981.

Contact Address

Prof. K.D. Papailiou
National Technical University of Athens
Dept. of Mechanical Engineering
Lab. of Thermal Turbomchines
P.O. Box 64069 phone: +30-1-7759-584
157 10 Athens / Greece fax: +30-1-7784-582

3.12 Kinetic Ising Model

We run Monte Carlo Simulations for the dynamical critical exponent of the two-, three- and five-dimensional Ising model with the Glauber and the Metropolis function at the critical point. In all three cases the lattices are to our knowledge much larger than the largest previous simulations, previous simulations (for instance, lattice sizes of 169984^2 in d=2 and 2072^3 in d=3 could be simulated on the Parsytec GCel-System). The algorithm applies multi-spin coding techniques and is based on domain decomposition for parallelization. From the results in [1] it is clear, that the scatter at the later times must be reduced to obtain better estimates for z. For this it will be necessary to study even larger systems and/or increase the number of runs and observation time. But also the errors in the static properties like K_c or β/ν above two dimensions would then become relevant.

Nevertheless our results indicate some stabilization of the z values near 2.2 and 2.08 in two and three dimensions, consistent with other recent numerical estimates. And our five-dimensional data show reasonable consistency of our methods and known facts.

References

[1] Ch. Münkel, D.W. Heermann, J. Adler, M. Gofman, D. Stauffer. *The dynamical critical exponent of the two- three- and five-dimensional Kinetic Ising Model.* Accepted by Physica A (1992).

Contact Address

Prof. Dr. Heermann
Dr. Ch. Münkel
Institut für Theoretische Physik
Universität Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg / Germany email: muenkel@lattice.tphys.uni-heidelberg.de

3.13 MaxHom

MaxHom is a molecular biology application that scans databases of *protein sequences* for structural and/or functional similarities. The database search is a straightforward implementation of a dynamic programming algorithm [1, 3] that runs on message based MIMD- as well as on SIMD-systems.

The program has been implemented on a variety of massively parallel systems, including a cluster of Silicon Graphics Workstations (R3000), an Intel Touchstone Delta with 528 i860 processors and the Parsytec GCel with 1024 T805 transputers.

To attain a coarse-grain parallelism, each node works autonomously on a section of the database. The database is split into small parts by extracting the essential data (entry ID, access number, a short description of the protein sequence and the actual sequence) and writing them into an unformatted file, once for each version of the database. A database with 30000 sequence entries, for example, can be split into 300 files with 100 entries each, where each protein sequence entry is stored in 2 records, one line containing the basic information about the protein and one line containing the sequence.

Because the number of database files can be controlled by defining the number of entries per file, the total number of database sections can be adjusted, so that it is substantially larger than the available number of processors. This provides a very rough load balancing mechanism.

The master process starts by reading the user-given options like the query sequence, the alignment parameters and the threshold criterion for reporting alignments [2]. The node processes are then initialized. After sending the query sequence and parameters to all processes, the host process waits for an idle signal from any node process. In this approach, the host process assigns a section of the database to an idle node by sending a file pointer to the node process, rather than by sending the database section itself.

Each process stores the resulting alignment in a local file. The data needed for the final sorting of the best matches, i.e., the alignment score and file record pointers to the stored alignments, are kept in local memory. As soon as a node process is idle, it will receive another file pointer until all database files are processed.

The speed of this implementation rivals that of the fastest known sequence-sequence search codes on single-instruction-multiple-data (SIMD) machines.

For example, the speedup using 8 processors on a SGI machine reaches 6.4 (run time decreased from 117.5 minutes to 18.3 minutes). The shortest run times where obtained on the Intel Touchstone with 60 seconds (225 processors, speed up 118) followed by the Parsytec machine with 249 seconds (512 processors, speed up 276). With larger numbers of processors no additional speed-up could achieved. In fact, the run time increased again, reflecting the higher amount of administration (sending the initial data to all nodes etc.). For very large processor numbers, the application switches from CPU-bound to I/O-bound.

References

[1] O. Gotoh. *An improved algorithm for matching biological sequences.* J. Mol. Biol., 162, 705-708, (1982).

[2] C. Sander, R. Schneider. *Database of Homology-Derived Structures and the Structural Meaning of Sequence Alignment.* Proteins, 9, 56-68, (1991).

[3] T.F. Smith, M.S. Waterman. *Identification of common molecular subsequences.* J. Mol. Biol., 147, 195-197, (1981).

Contact Address

Prof. Dr. Chris Sander
Dr. Reinhard Schneider
Protein Design Group — phone: +49 6221 387305
European Molecular Biology Lab. — fax: +49 6221 387306
D-6900 Heidelberg — email: schneider@embl-heidelberg.de

3.14 Monte Carlo Simulation of Quantum Spin Glasses

Extensive Monte–Carlo simulations of the two–dimensional quantum spin glass model in a tranverse field at zero temperature have been performed to manifest the existence of a quantum phase transition and to investigate the critical properties. By using sophisticated finite size scaling methods the system size could be confined to fit into the memory of one processor, by which the necessary disorder average could be done in parallel on any number of processors. Thus the inherent massive parallelism of an extensive statistical average could be used to linearly speed up the production runs. For these kind of problems the transputer cluster as a whole proved to be as efficient as 6 processors of a Cray YMP.

Contact Address

Dr. Heiko Rieger
Institut für Theoretische Physik
Universität zu Köln
50923 Köln / Germany
phone: +49 221 470 4211
email: rieger@thp.uni-koeln.de

3.15 Neural Networks in the Clinical and Psychiatric Field

The present work consists of the parallel implementation of an existing artificial neural network (ANN) used to simulate the generation of Event Related Potentials. The equations governing the dynamic evolution of each neuron are of the following form, with u_i the membrane potential of the i neuron:

$$u'i(t) = k_1 u_i(t) + k_2 f(t - t_i) g(A_i(t)), \qquad i = 1, ..., N$$

where $f(t - t_i)$ is a function representing the sensitivity of neuron i to external stimuli in connection with the time moment t_i, when neuron i last fired. g is the non-linear activation function response to incoming stimuli represented by:

$$A_i(t) = \sum_j (S_{ij} exp(-(t - t_j)/\tau)) + \sum_k (P_{ik} exp(-(t - t_k)/\tau))$$

where S_{ij} is the strength of the synapse connecting j neuron of the ANN to i neuron, P_{ik} is the strength of the synapse connecting k neuron of an external pre-processing unit, considered as the input module of the system, to i neuron of the ANN. The neurons of the ANN which are activated from input patterns and correspond to an ERP attractor memory create special law-pass filtering synapses to the output unit of the system, called Target Neural Assembly. A bipolar current:

$$I(t) = F(SA(t)), \quad SA(t) = \sum_k \lambda_k exp(-(t - t_k)/\tau)$$

is created between the synaptic region and the soma of the neurons of TNA, where F is a low-pass filter function ($f_c < 20$ Hz) and $\sum_k$ is over the active memory neurons. This current generates a measurable head surface ERP.

In the implementation executed we parallelized the ANN, keeping the input and output modules as separate, single transputer entities. The network uses the farm topology with one master having 15 slaves. The master acts as the information interchange centre between the 15 transputers-slaves it controls. Information interchange had to be activated on each cycle but the only information to be passed is the firing time t_i, $i = 1, ..., N$ when it changes, which can be reduced to a i/o message, thus reducing the intertransputer communication overhead. The network used could be extended as a binary tree to include more p16 basic clusters. In this case asynchronous communication seemed more suitable, since the time spent for inter-master communication increased significantly. Following this scheme a 1 bit message is sent only when a neuronal firing occurs.

Each trannsputer-slave simulates the workings of a subset of the whole ANN. In our implementation each T800 included 15 neurons, making the total neuron population $15 * 15 = 225$. A small neuronal population in each transputer was necessary because there the ANN function is obligatorily serial and the computational burden becomes prohibitive in comparison with RISC-computer serial implementations. Each transputer is aware of the 152 connections S_{ij} of its internal neural units and also of the $15 * (225 - 15) = 3150$ connections with external neurons. This is an

advantage towards the totally serial implementation where a $N = 225$ network has to handle $225 * 225$ synaptic connections, in a single processor RAM.

The initial data concerning functional parameters are divided into two categories: (a) Common parameters, useful for every slave system (b) Synaptic connection S_{ij}, P_{ik} information related only to specific transputers. According to this division the master of the p16 cluster transmits to each slave two files: the first contains the common data and the second contains the names of the files the that the slaves must open (one filename for each slave), according to their identity number, in order to find the specific synaptic information concerning them. This method reduces the data input procedure to its minimum because transputer time is only spent for reading useful synaptic weights. It must be noted that common working parameters are totally only 240 bytes, which is much less than the synaptic data, 3375 bytes $(3150 + 15 * 15)$.

It must be noted that the computations concerned with the dynamical evolution of the neuronal membrane potentials are carried on exclusively in the slaves so that the master is concerned only with the transmission of neuronal firing times, or more simply the occurrence of firing.

Future developments of the existing parallel ANN infrastructure will concentrate on the extension of the internal neural architecture so as to include neurons obeying to Dale's law, that is being either excitatory or inhibitory, since these networks behave much better concerning memory retrieval ability. Simulations already done consisted only the "run" phase of memory retrieval. The implementation of the memory "learning" phase will also be studied in order to compare its performance with existing serial code results.

Contact Address

Prof. N. Uzunoglu
Dept. of Electrical and Computer Eng.
National Technical University of Athens phone: +301 3616908
Patision 42, Athens / Greece fax +301 3647794

3.16 Numerical Simulation of Unsteady Flows

Fluid motion is a continuum mechanics problem that is governed by the Navier-Stokes equations. Dimensional analysis yields a parameter call Reynolds number, that fully characterizes geometrically similar flows. We can identify three major regimes along the Reynolds number scale:

- The laminar regime
- The transitional regime
- The turbulent regime

Flows that fall clearly in the laminar regime (for instance the flow around a sphere at Reynolds number 100) are quite easily simulated with the current know-how in numerical methods and contemporary workstations. It is apparent though that both theoretical and technological interest mandate the simulation of flows at the other two regimes. The higher the Reynolds number is, the smaller the scale of the eddies in the flow get and the faster their motion becomes. This fact, no matter what numerical approach one uses, dictates three distinct but inter-related requirements: a) Finer meshes(i.e. meshes with elements at least as small as the smaller eddies) b) Smaller time steps, small enough to resolve the fastest of the fluids' motions c) Higher order of accuracy numerical schemes. This work describes our first efforts to exploit MPP technology for the solution of a three-dimensional flow problem.

The physical problem we are dealing with is that of a wing at a large angle of incidence. At this first stage of our effort, the actual geometry of the wing is quite simple, i.e. a flat plate. This allows for simplicity of the computer code and easier experimentation with numerical features and parallelization techniques. We must note here though that the three dimensional flow field around a flat plate at an angle of incidence is an extremely difficult problem. The test case we shall deal with is that of the flat plate at an angle of attack of 10^o and of Reynolds number 1000.

The main features of the numerical solver will be outlined briefly here. The method solves for primitive variables, i.e. velocity and pressure, is fully implicit and is a control volume method. The accuracy of the method is of second order in space (central differencing) and of first or second order in time. The method employs a pressure correction technique which is actually a PISO variant. A series of seven-diagonal matrices are produced which are then solved iteratively.

The parallel version of the code was not perfected to the point as to provide final results, since this was not the aim of this study. It was however complete in its form, fully parallelized, with no shortcuts and correct as to the intermediate results it produced (checked with the results from a workstation). We have decided on a two-dimensional decomposition of the grid for the assignment to the processors of the computer. Thus, each processor is assigned with a column of cells having the full length of the computational domain and small parts of the vertical and span-wise dimensions. This decomposition was adopted for three reasons: a) It is quite simple b) the combination of (number of processors) with (main memory of each

processor) fits well the grids we had in mind to solve for this test case. c) the flow-wise decomposition keeps the NI direction unified, thus it does not require special treatment for separating "processors on the flat plate" and "processors in the front and rear wake". This effect reduces the necessary communications.

The performance achieved with this code on a R4000 Indigo (SGI) was 9.5 MFLOPS, which is about 60 percent of the machine's nominal sustained performance. The Indigo, having 32 Mbytes of main memory, has a limitation on the size of the grid it can efficiently solve too. On the Parsytec machine, we adopted the local dimensioning technique, that is, the program each processor runs has the real local dimension and not the global dimension of the problem. This feature requires serious modifications of the code, but once accomplished, permits to take full benefit of the machine's total memory. On this test, 6 grids with 96,800, 224,000, 640,000, 2,560,000, and 10,240,000 points where computed.

Computations were performed on partitions ranging from 16 to 512 nodes. The maximum performance achieved was 288.9 MFLOPS on the 512 node partition and with the bigger grid. The lowest performance was that on the smallest partition (16 nodes) and the smallest grid and was 6.82 MFLOPS. The scalability defined as: Scalability=100x(Performance on MxN processors)/Mx(Performance on N processors) ranges from 41bigger ones We must note here, that one of the reasons that the scalability is so good on the finer grids, is that the spatial numerical scheme is of second order of accuracy, thus requires only one neighbouring node and reduces communications a lot in benefit of processor arithmetics. The programming technique we adopted for performing the synchronous communications between the processors were to do all the necessary transfers before the beginning of any computations and not on request. This scheme separates the transfer part of each subroutine from the computing part and is much easier to debug by hand. We must say here that besides the necessity of some form of communications debugger, we feel that, especially for simple grid-type topologies, some form of auto- communicating scheme could be incorporated in the compiler or provided as library routines.

All computations on all machines were performed with compiler optimization options.

Contact Address

Y. Ventikos, G. Tzabiras, Th. Loukakis
Viscous Flows Group
N.A.M.E – N.T.U.A
P.O. Box 64070 phone: +30 1 7700405
15710 Zografos fax: +30 1 7774478
Athens / Greece email: yvent@areti.naval.ntua.gr

3.17 Optimization of Large SQL-Queries

A parallel optimizer for large join queries has been developed on the parallel machine of HPCL. The optimization method is based on iterative improvement and constructs local minima for randomly generated start states within a time span, which increases polynomially with the query size. For the parallel implementation of the optimizer, a tree topology – the "optimizer tree" – has been established, where the root node acts as a master and coordinator for the slave nodes.

At the initialisation phase, the initial query tree, the dictionary and a number of start values are distributed from the master towards the leaf nodes of the optimizer tree. At the processing phase, all slaves construct start states and build local minima on them in parallel. At the termination phase, the master forwards a stop signal to the slaves signifying the expiration of the time limit; the slaves send towards the master the least cost local minimum constructed or received by each one in pipeline. The master defines as global minimum the least cost local minimum received.

The parallel optimizer has been implemented and is currently being tested. Tests are performed for the same query size using different numbers of processors, whereby each slave constructs local minima in sequential mode; total time and value of the global minimum are estimated and combined, to measure the efficiency of the parallel implementation. For queries with 50 and 80 joins, the algorithm saturates when the slave processors are more than half of the query size; for larger queries, it is therefore expected to improve even for larger numbers of slaves, thus making optimal usage of the processing power.

The implementation revealed a problem of memory fragmentation, which will be tackled by code optimization, and of high memory requirements which set an upper limit to the query sizes that can be handled. The latter problem prevents the establishment of multiple threads on the same processor, and forces sequential execution; the development of an application specific multitasking mechanism with minimum memory requirements is envisaged. The general problem of memory limitation could be resolved by aggregation of the slaves into groups and usage of their distributed memories in a shared way; this subject will be studied further.

The main goal of this work is the experimentation with different relation numbers and relation structures for fixed query sizes and study their impact on the shape of the search space. Since memory limitations prevent extensive experiments, experimentation will start as soon as the memory limitation problem allows at least processing of queries with up to 500 joins.

References

[1] M.Spiliopoulou, M.Gatzonis, M.Hatzopoulos. *Parallel Optimisation for Large Join Queries*, GP_MIMD report for Task 6.2.10, Feb. 1994

[2] M.Spiliopoulou, M.Hatzopoulos, C.Vassilakis. *Parallel Optimisation of Join*

Queries using a Technique of Exhaustive Nature, Computers and Artificial Intelligence, Vol. 12(2), 145-166, 1993

[3] M.Spiliopoulou, M.Hatzopoulos, C.Vassilakis. *Using Parallelism and Pipeline for the Optimisation of Join Queries*, Proc. Parle Conf., Paris, France, 279-294, 1992

Contact Address

Prof. Michael Hatzopoulos
University of Athens
Department of Informatics — phone: +30 1 7217941
Panepistimiopolis, TYPA Building — fax: +30 1 7219561
157 71 Athens / Greece — email: mike@zeus.di.uoa.ariadne-t.gr

3.18 Pivot

One of many variants of *Gaussian elimination* was used for implementation on the Parsytec GCel; this variant performs matrix-updates on the remaining part of the coefficient-matrix and is called a right-looking algorithm. In combination with a block-scattered distribution of the data over the processors which are arranged in a grid topology, this allows for optimal performance on a distributed memory system. Moreover, the pivot strategy was adapted in such a way that data traffic is minimized while retaining the required numerical stability. The details of the algorithm are explained in [1]. Some preliminary results of experiments with this implementation on the Parsytec GCel are presented in [2]. For these experiments, 256 processors were used which are arranged in a square grid.

It can be noted, that for matrices of various size the performance of Gaussian Elimination improves considerably by using the modified pivot strategy. For the number of processors used in this experiment, it gives an increased performance of up to about 26%.

References

[1] W. Hoffmann and K. Potma. *Threshold Pivoting in Gaussian Elimination to Reduce Inter-Processor Communication.* Dept. of Computer Systems, University of Amsterdam, 1991.

[2] W. Hoffmann and Z.W. Zhang. *Solving Dense Linear Systems Efficiently on the Parsytec GCel.* Technical Report: Dept. of Computer Science, University of Amsterdam.

Contact Address

W. Hoffmann
Dept. of Computer Systems
Universiteit van Amsterdam — phone: +31 20 525 7463
Kruislaan 403 — fax: +31 20 525 7490
1098 SJ Amsterdam / The Netherlands — email: walter@fwi.uva.nl

3.19 Recursive Partitioning

The CART-approach establishes the regression tree for classification using a greedy-like selection criterion during the first construction phase followed by a second phase in which the tree is retracted to a sufficiently small size. The bounding of the size is motivated by practical issues, i.e. the observation that only trees with a small number of splits will be acceptable as a plausible decision aid.

Motivated by this we implemented a more enumerative approach in which all possible trees with a fixed number k of nodes are used as initialization for the sequential recursive partitioning algorithm to determine near optimal regression trees. This approach in conjunction with the concept of farming yields a parallel algorithm.

Extensive computational tests were performed on the 1024-processor cluster of the ZPR on a dataset for credit-evaluation.

References

[1] L. Breiman, J. H. Friedman, R.A. Olshen and C.J. Stone. *Classification and Regression Trees.* Bellmont (1984).

[2] H.–J. Hofmann. *Die Anwendung des CART–Verfahrens zur statistischen Bonitätsanalyse von Konsumentenkrediten.* Zeitschrift für Betriebswirtschaft, February 1990.

Contact Address

Prof. Dr. Dr. Ulrich Derigs
Jürgen Antes
Lehrstuhl für Wirtschaftsinformatik — phone: +49 221 470 5331
Universität zu Köln — fax: +49 221 470 5329
50923 Köln / Germany — email: antes@informatik.uni-koeln.de

3.20 Sheared and Stratified Turbulences

The objective of this project is the transcription of the fully vectorized three dimensional LES code LESTUF (Large-Eddy Simulation of Turbulent Fluid), including balance equations for momentum, heat, and two passive species concentration, on a massively parallel processing system. A success of this study is of fundamental interest for collaborating work concerning the dispersion and mixing of aircraft exhaust gases, aerosols, and water vapor in the upper troposphere.

Within a resolution of 128^3 meshes, one simulation requires about 44 MWords (64bit) of memory and runs about 3.000 time steps, consuming about 8 hours CPU time on one processor of a Cray-YMP 2-32. When the number of meshes is increased to 500^3 a memory of about 2.500 MWords and 12.000 time steps are necessary. Strong emphasis will be layed on the parallelization of the routines inside the timestep loop. Domain decomposition into (2d) slices or smaller (3d) cubes are an appropriate approach.

Contact Address

Dr. Michael Faden
Institut für Antriebstechnik
DLR e.V. phone: +49 2203 601 2285
Linder Höhe fax: +49 2203 601 2353
51147 Köln / Germany email: michael@kat01t.at.kp.dlr.de

3.21 Similarity Search of 3-D Chemical Databases

The concept of molecular similarity is an important one for drug design in pharmaceutical research [1]. In the search for new lead compounds potential candidates are likely to be similar in some way to known active compounds. This has lead to the desire to search large databases of 3-D chemical structures and rank molecules according to their similarity to a known target structure, which is typically a small molecule with known pharmacological activity [2]. This requires a method of determining molecular similarity on a time scale appropriate to application to databases of several hundred thousand compounds.

There are a number of different approaches, but most employ the same fundamental procedure.

The 3-D molecular similarity is calculated as a function of any molecular property that depends on the arrangement of atoms in space. For each of the molecules to be compared, a property is calculated for each of a set of points in 3-D space. By comparison of these two sets of values, a similarity index is calculated, and the orientation of one of the molecules is then optimised to give a maximum in the similarity function. The differences arise from the following properties:

- The molecular property used in the comparison;
- The method of sampling the property in 3-D space;
- The method of calculating the similarity index;
- The method of sampling and optimising the orientation in space.

Molecular properties such as the electrostatic potential have been used in the past, with sampling on a grid. All grid based methods, even using parallel hardware, are too time consuming for database work. There are a number of different approaches, but most employ the same fundamental procedure.

Recently published work [3] suggests that by using atomic electron densities to approximate the shape of an atom, by fitting gaussian functions to these, and then using those to analytically evaluate a similarity index, several orders of magnitude improvement in speed can be achieved.

Several methods for determining the global optimum of the similarity have been tested, including the use of a genetic algorithm [4]. For this application it appears that a more conventional quasi-Newton minimisation technique, such as the BFGS (Broyden-Fletcher-Goldfarb-Shannon) method using random starting points is more efficient.

By using the rapid methods of similarity determination described above combined with database parallelism across a workstation cluster, it has proved possible to search fairly large 3-D chemical databases within a reasonable timescale. Searches on a 20,000 compound database of structures from the Cambridge database using a cluster of ten workstations takes around 3 hours. The code is being ported to the Parsytec GC-el/1024, and will be a crucial test of the I/O capability of the system.

References

[1] Johnson, M. Similarity based methods for predicting chemical and biological properties: a brief overview from a statistical perspective. In Chemical Information systems; Bawden, D. Mitchell, E., Eds.; Ellis Horwood; Chichester, England, 1990; pp 149-159.

[2] Martin, Y. C., Bures, M. G and Willett, P. Searching Databases of 3-D Structures. In Lipkowitz, K. B. and Boyd, D. B. Eds. Reviews in Computational Chemistry VCH Publishers, New York 1990.

[3] Good, A.C., Richards, W.G. Rapid Evaluation of Shape Similarity Using Gaussian Functions. J. Chem. Inf. Comput. Sci. 1993, 33, 112-116.

[4] A.W.R. Payne and R.C. Glen, *Molecular Recognition using a binary genetic search algorithm* J.Mol.Graphics, Vol. 11, 74 (1993).

Contact Address

Mr. M.P. Burrow
Daresbury Laboratory
Daresbury +44 925 603207
Warrington fax: +44 925 603634
WA4 4AD, U.K. email: m.p.burrow@daresbury.ac.uk

3.22 Simulation of Traffic Congestions

The task of optimally routing a fleet of vehicles through a road network is extremely complex caused by a non-linear cost function, the stochastic properties of the traffic demand and the traffic dynamics itself. In this project this is handled by using a microscopic model to simulate the effect of individual routing, which yields the possibility of a different track plan for each vehicle. The simulation model has to run fast enough to enable the inclusion of inductive loop counters in the future in order to perform online congestion prediction and avoidance on a digital street network.

Two conceptually different codings of a simple, rule-based microscopic approach to traffic flow were implemented on different parallel supercomputers. Besides the Parsytec GCel an Intel iPSC/860, a CM-5 and a NEC-SX3 were used (cf. [1]). Compared to published computing speeds the microscopic model proves to be up to 1000 times faster. The highest computing speed is found by employing a single-bit coding scheme (cf. [3] used in, e.g. Ising-model programming). As traffic flow is an one-dimensional problem, a complication is that geometric parallelization has to be done in the same direction as single-bit coding. Nevertheless, efficiencies near 100 percent for large systems could be reached.

The production runs showed that the modeling approach yields qualitatively realistic results (cf. [2]). In further runs the qualitative influence of the parameters 'maximum speed' and 'fluctuation of speed' on the model was examined. The Parsytec GCel was the machine which could handle the largest models (about 10^6 km) and the last to reach the real time limit (cf. [1]).

References

[1] K. Nagel, A. Schleicher. *Microscopic Traffic Modeling on Parallel High Perfomance Computers.* Working Paper No. 93.132, University of Cologne (1993).

[2] K. Nagel, M. Schreckenberg. *A cellular automaton model for freeway traffic.* J. Physique I 2, 2221 (1992).

[3] D. Stauffer. *Computer Simulations of cellular automata.* J. Phys. A 24, 909 (1991).

Contact Address

Prof. Dr. Achim Bachem
Kai Nagel, Thomas Pfenning
Zentrum für paralleles Rechnen — phone: +49 221 470 2210
Universität zu Köln — fax: +49 221 470 5160
50923 Köln / Germany — email: pfenning@mi.uni-koeln.de

3.23 Time Warp

The optimistic parallel discrete event simulation algorithm *Time Warp* is being implemented on the GCel-512 of the University of Amsterdam.

Discrete event simulation is needed to understand and optimize digital electronic circuitry or public services such as telephone networks or traffic flow. The simulation of very heterogeneous dynamic systems is the target.

The evaluation of the optimistic parallel discrete event simulation for massively parallel processing is done via integration of the Time Warp paradigm with a asynchronous Cellular Automata. The reformulation of the time driven approach to the event driven Jefferson Time Warp paradigm allows the efficient use of cellular automata. Good scalability has been observed on smaller parallel systems so far.

References

[1] B.J. Overeinder, P.M.A. Sloot, and L.O. Hertzberger. *Time Warp on a Transputer Platform: Pilot Study with Asynchronous Cellular Automats.* Parallel Computing and Transputer Applications (PACTA 92). Eds. M. Valero, E. Onate, M. Jane, J.L. Larriba, B. Suarez. IOS Press, Amsterdam, Washington. pp 1303-1312 Sep. 1992.

Contact Address

Dr. P.M.A. Sloot, B.J. Overeinder
Dept. of Computer Systems
Universiteit van Amsterdam — phone: +31 20 525 7463
Kruislaan 403 — fax: +31 20 525 7490
1098 SJ Amsterdam / The Netherlands — email: peterslo@fwi.uva.nl

3.24 TSP-Heuristics

The objective of *Traveling Salesman Problem* (TSP) is to find the minimum cost round trip through a given set of cities visiting each city exactly once (cf. [1]). It appears as subproblem in several practical applications such as *vehicle routing* or *VLSI design*. Real world problems typically have a size of several thousand nodes, and the solution has to be found in very short time (normally a few minutes).

The TSP is NP–hard. Thus, one can not expect to solve practical problems to optimality in a reasonable amount of time. Therefore, heuristic methods have to be used in order to obtain suboptimal solutions for very large problem instances. The heuristic with the best overall performance is the Lin-Kernighan improvement heuristic (cf. [2]), which yields solutions being about 5 percent longer than the optimal solution. For this heuristic a parallel version was implemented. On the other hand for very large problem instances even the Lin-Kernighan algorithm has a running time of several hours. So a decomposition approach was used and also parallelized.

Results: The running times of the algorithms could be remarkably reduced to some minutes. To obtain an approximate solution for a problem with 18512 nodes that is only 4 percent longer than the original Lin-Kernighan solution, 20 minutes of computing time with 128 processors (cf. [3]) were needed. Speedups up to 463 for 1024 processors, could be obtained.

References

[1] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys. *The Traveling Salesman Problem.* Wiley, Chichester (1985).

[2] S. Lin, B.W. Kernighan. *An Effective Heuristic Algorithm for the TSP.* Operations Research 21, 498–516 (1971).

[3] A. Bachem, M. Wottawa. *Parallelisierung von Heuristiken für große Traveling Salesman Probleme.* Proceedings of the Transputer Anwender Treffen (TAT 92), Aachen, Springer (1992).

Contact Address

Prof. Dr. Achim Bachem
Michael Wottawa
Zentrum für Paralleles Rechnen
Universität zu Köln
50923 Köln / Germany

phone: +49 221 470 6019
fax: +49 221 470 5160
email: wottawa@mi.uni-koeln.de

4 Industrial Applications

This section describes commercial codes that have been ported to the ZEUS Centers' large-scale MPP systems. Some of the reported codes are in every-day use by engineers, others are still in their experimental phase. The codes are typically supplied by independent software vendors and run on a variety of platforms. Major software houses are currently undertaking a significant restructuring of their codes in order to enhance portability and support usage of MPP systems.

4.1 Associative Database Comparison

The application class of associative, fault-tolerant database comparisons suites well to the massively-parallel processing architecture of the Parsytec GC. As a first successful project in this range of applications, the change of the zip codes of the customer address database of the GZS ('Eurocard' credit card organization, located in Frankfurt/Main) from the former 4-digit to the new 5-digit zip code system in Germany took place in May 1993 at the PC^2 .

The novel associative address conversion program used in this project detects incorrectly written addresses by a neural network approach. It first corrects incorrect addresses, and then converts the corrected old addresses to the new ones with the 5-digit zip codes. The sequential version of the program was developed by AssoWare GmbH and runs on several machines. To meet customer requirements concerning time restrictions — i.e., shortest possible interception of the standard database operation — and conversion quality — convert as many addresses as possible successfully to correct new ones — the code has been ported to the GC. Due to the structure of the new zip code system, addresses in small towns allow a conversion based on the town name and the old zip code, whereas new zip codes in big cities are based on street name and house number, too.

In the first phase, the addresses are classified. Based on a literal comparison, all city addresses are gathered and separated from non-city addresses. Furthermore, inconvertible addresses (e.g., addresses in foreign countries) are sorted out.

The next phase works on non-city addresses. Using the associative approach, town names and their old zip codes are checked and corrected if necessary. For those addresses which are not corrected into a city address, the new zip code is derived, too.

Finally, the last phase covers the city addresses. Another associative comparison and correction step yields correct city addresses including street name and house number, which are also given their new zip code.

Due to the high I/O requirements, the first phase is executed on the host system. The second phase uses a farming scheme with the master process located on the host. Due to the size of the town name database, each slave is formed by a group of 8 processors. The third phase is built as several independent processors working on different problem parts.

Results: The speed of this implementation has proven that the address conversion of more than 4 million addresses including export and import of the database files could take place on just one weekend and made it possible to use the full fault-tolerance features of the program. The achieved conversion rate of more than 98.6% and the short runtime of the conversion program was essential to minimize the amount of subsequent manual conversion work.

Discussions with software vendors working in this application area lead us to the conclusion that the architecture of the GCel-1024 is ideal also for other applications of this class. In this sense, the application described above should just be considered

as one possible application example.

Contact Address

Christian Fleischer
Detlef Segschneider
Parsytec Computer GmbH
D-52070 Aachen

phone: +49 241 16600 0
fax: +49 241 16600 50
email: chris@parsytec.de
detlef@parsytec.de

4.2 CEVATS

The DLR CEVATS-Code is used for analyzing compressible steady flows around arbitrary geometries typically reaching from simple two-dimensional airfoils up to complete aircraft. The investigated flow field on may either be considered inviscid or viscous, the latter case allowing for laminar and turbulent regions including a transition between both regimes. The code is block structured employing an explicit Runge-Kutta time stepping scheme inside a multigrid approach.

For the parallel implementation of the DLR CEVATS-Code several modifications were necessary for preparation, since the program is especially adapted for applications on a Cray-YMP.

- In-Core Version (completed).
  The original program works with an off-core solver only, using a fast solid-state storage device and performing a large amount of I/O. For the parallel implementation the code had to be extended by an in-core solver keeping all data in the main memory.

- Workstation Version (completed).
  All Cray specific function calls had to be removed and were replaced by own Fortran functions or subroutines, in order to port the program to any parallel computer.

- Parallel Data Exchange Strategies (completed).
  An analysis of the original CEVATS-code revealed that the data exchange at interconnected blocks is not parallelizable. Three new strategies differing in communication effort and accuracy were added which can work in parallel.

- Second Layer of Dummy Cells (on-going).
  In the basic code each computational block is surrounded by one layer of dummy cells reducing the accuracy to first order at block boundaries. Since a MIMD application demands for a very high number of blocks, a second layer of dummy cells is introduced, in order to maintain second order accuracy at cuts between different blocks.

References

[1] B. Eisfeld, B. Roll. *Parallelization of the DLR Three-Dimensional Multi-Block Multigrid Euler/Navier-Stokes Code CEVATS.* Internal Report, Institut für Entwurfsaerodynamik, DLR Braunschweig, Rechenzentrum DLR Braunschweig.

Contact Address

B. Eisfeld, B. Roll
Institut für Entwurfsaerodynamik
DLR Braunschweig, Flughafen
D-38105 Braunschweig

phone: +49-531-395-2834
+49-531-395-2765

4.3 Crew Management

The Crew Management problem is one of the most complex and computational intractable problems. An instance of the Crew Management problem is described in terms of a set of aircraft rotations, each being a sequence of flight legs that airplanes perform in a specific (usually monthly) time period. The task is to find a set of pairings, that is a set of flying crew rotations, that cover all flight legs, that satisfy all types of constraints (contractual and govermental ones) and have a minimal global cost. The size of the problem is large due to the great number of flights in the specified period, the great number of crew members and the numerous constraints to be taken into account.

In all cases the complexity of an optimal solution must satisfy a set of different types of constraints, such as

1. temporal constraints (e.g. a flight leg assigned to a crew must commence after the crew has completed a previous assignment and to allow some other procedures to take place such as check-in, and check-out of passengers, aircraft inspection, handling of freight, aircraft change, etc),

2. local constraints (e.g. the arrival port of a flight leg is the same to the departure port of the next leg, avoidance of over-night staying, local services, etc),

3. the flight-duty-time (e.g. the sum of flight time of a rotation, of the ground-time in between flights, of the briefing-time, etc.)

4. the number of flight legs per working day and per rotation (in order to control the ammount of take-offs and landings of a crew),

5. the maximum working time in a specified by the regulations period of time,

6. the rest-time (that depends on the hours worked up to that point, time-zone crossings, preceeded working days, etc), and

7. other contractual or govermental constraints (not falling into the previous classes), and at the same time has a minimal cost for the airline company.

A crew scheduling program is CPU bound and it takes a long time to provide with the final solution. Therefore, development work on the algorithm employed in the crew scheduling optimisation problem have focused primary on increased processing speed and parallelism. Thus, an interesting research topic and our major objective is to design this computationaly intensive application on the massively parallel environment in order to achieve high performance and better execution time. The implementation of the system will be based on the Parsytec GCel-512 machine of the Athens-HPCL.

(This work is sprung from our involvement in the PARACHUTE ESPRIT III Project concerning the development of the Crew Management application using the parallel CHARME tool of Bull)

References

[1] ESPRIT III Project No 9134, *Parallel Charme for User Technologies (PARACHUTE)*. Technical Annex.

[2] R. Marsten, M. Muller and C. Killion. Crew Planning at Flying Tiger: *A Successful Application of Integer Programming. Management Science*, Vol. 25, No. 12, Dec. 1979.

[3] K. Hoffman and M. Padberg. *Solving Airline Crew Scheduling Problems by Branch-and-Cut*. Management Science, Vol. 39, No. 6, June 1993.

[4] D. Ryan. *The Solution of Massive Generalized Set Partitioning Problems in Aircrew Rostering*. Journal of the Operational Research Society, Vol. 43, No. 5, pp.459-467, 1992.

Contact Address

Prof. Constatin Halatsis
University of Athens
Department of Informatics — phone: +301 7217941
Panepistimiopolis, TYPA Building — fax: +301 7219561
157 71 Athens / Greece — email: halatsis@di.uoa.araidne-t.gr

4.4 DYNA 3D

CONDAT-DYNA 3D is an industrial 3D Lagrangian Finite Element program with explicit time integration for the analysis of large deformation dynamic response of inelastic solids and structures. Typical applications include low-, high- and hyper-velocity impacts and penetrations, shock wave studies, vehicle crash, metal forming, and damage analysis in composites.

ZIAM and CONDAT conduct a study about the possibility and the labour required for a parallelization of this code. This code will be ported to workstation cluster and different parallel computers by using the PVM message passing interface.

The stucture of the DYNA 3D FEM solver has the following loop structure

```
for all time steps
  for all element groups
    for all element sub-groups (identical material conditions)
      for all all element sub-sub-groups (elements with no common nodes)
        for all elements
          compute stresses, internal forces, etc.
  update geometry, velocities, energy
```

The parallelization of the solver will be done in a first step (for only one type of material) in the loop over the element groups. The loop over the element groups was implemented to port the code to a SIMD computer. The only condition the element groups have to fulfill is a maximum number of elements belonging to them. There is a parameter in the initializer which generates the input data for the solver to fix the size of these element groups. This parameter can be set in such a way that in the later MIMD version every processor node has to calculate one element group and the same number of elements.

The update of the material conditions and the other element parameters for the calculation take place inside the element sub-groups. This can be done localy on every processor node. Therefore the only global communication necessary in the time step loop will be the global update of the energy-, geometry-, and forces data.

Contact Address

Dr.-Ing. Andreas vom Hemdt
Bernhard Hofer
ZIAM GmbH — Zentrum für Industrielle
Anwendungen Massiver Parallelität
Kaiserstr. 100
D-52134 Herzogenrath

phone: +49 2407 9567 0
fax: +49 2407 9567 15
email: ziam@infoac.rmi.de

Dr. Klaus Thoma
CONDAT — Gesellschaft für EDV-Consulting
und Datentechnik mbH
Maximilianstr. 28 phone: +49 8441 6004
D-85298 Scheyern - Fernhag fax: +49 8441 18713

4.5 Electro Magnetic Simulations

British Aerospace uses electromagnetic simulation packages for the design and test of aircrafts which are more resistant against electromagnetic interactions such as lightnings etc. With the simulation software, electromagnetic interactions can be tested at a much lower cost and in a shorter time frame than with the classical (empirical) approach.

Aircrafts are potentially sensitive to outside influences such as HIRF (High Intensity Radiated Fields), lightning or other electro magnetic environmental effects. These problems are magnified by the use of non-conductive and poorly conducting materials such as carbon fiber composites, and the incorporation of the "flight by wire" technology. Therefore, the new FAA regulations require that all new aircraft must safely withstand all EM effects.

Past validation of aircraft has typically been accomplished by extensive testing which is extremely cost intensive and can only be achieved once an aircraft is available, i.e., has been manufactured. At that point, design changes needed to be made and hardware modifications produced, high both in time considerations and actual expense. It is clear that a numerical approach can be and is an essential tool in solving EM problems and when used in conjunction with limited testing, is much more cost effective than testing alone.

The core of the electro magnetic simulation package that has been successfully implemented and evaluated both on BAe's own MPP system and the Cologne 1024 processor Parsytec GCel system consists of the 3D EM code by EMA Inc. in Denver, CO. The prototype running on the Parsytec System has been installed and heavily used at British Aerospace during the development of the European Fighter Aircraft. The main reason for British Aerospace for going with the MPP system was that the rental cost for running one test suite on a vector supercomputer already pays for the similar performing MPP system. Another reason was that the own MPP system is available 24 hours a day, 365 days a year.

Based on this work, a complete MPP electromagnetic simulation package is now commercially available, resulting from a cooperation between EMA Inc., FEGS Ltd., and Parsytec. The FAM software (Field Analysis Modeler), developed by FEGS, is used for pre-processing, automatic mesh generation and post processing, i.e. data visualization.

With this package, it is now possible to model the effects of EM on aircraft to significantly high frequencies enabling the simulation of lightning and HIRF. For example, it is now possible to model the EM interaction (exterior and interior simultaneously) of an entire Boeing 737 size aircraft to about 1GHz, and resolve geometry features as small as 2 inches. The package can also be used in many other application fields which include EM effects, such as bioelectromagnetics, radar cross section, EM compatibility, antenna analysis, directed energy weapons research and static electricity.

Contact Address

Dr. Christopher C.R. Jones
Electronic Warefare and Stealth Facilities Dept.
British Aerospace Defence
Warton Aerodrome
Preston, Lancashire
PR4 IAX / United Kingdom phone: +44 772 854799

4.6 Incompressible Turbulent Flow

We have developed a code to model the axisymmetric incompressible turbulent Navier-Stokes equations to model chemical reactor problems. A conventional colocated finite volume method is used to discretise the conservative form of the continuum equations. The coupling between the pressure and velocity equations is achieved using the PISO (Pressure Implicit with Splitting of Operators) algorithm and the resulting iterative equations are solved using either the standard Tri-Diagonal Matrix Algorithm (TDMA or Thomas Algorithm), a preconditioned conjugate gradient solvers (PCCG) or a preconditioned global minimum residual algorithm (PCGMRES). We have developed a parallel version of the code within the Communicating Sequential Process (CSP) programming model implemented within the FORTNET message passing harness portable to both shared memory and distributed memory systems. The performance of the various solvers and their effect on the convergence properties for different processor configurations being analysed.

The computation and communication times and the number of iterations to achieve a converged solution are measured for a test problem. The test case considered is for turbulent flow in a pipe and solutions are converged when changes in all normalised variables are less than 10^{-4}. The grid used has 128 cells axially and 32 cells radially.

The first thing to note about the results is that the configuration of processors clearly effects the global convergence rate of the algorithms. The deterioration in convergence rate of the TDMA algorithm can be explained by the decoupling of the back substitution step as more processors are applied in the x-direction (the dominant flow direction). The decoupling of the back substitution step as more processors are added in the y-direction does not have as noticeable an effect. However, for more complicated flows (e.g. with recirculation) this is not the case. The preconditioned algorithms follows no such pattern and its behaviour is being investigated further.

For the TDMA algorithm the most efficient configurations are the slab like decompositions (Ny = 1) as the computation time goes down roughly linearly (per iteration) and the communication times are approximately the same. The balance of work becomes more unevenly spread for other configurations of processors since boundary points are not as evenly distributed. For the preconditioned algorithms the compute time (per iteration) is always less for a particular processor configuration but, as they depend on global norms and residues, the communications costs increase dramatically with the number of processors.

Present work is concentrating on testing the scaleability of the code and porting it to a range of different hardware platforms such as the Intel iPSC/860 hypercube at Daresbury Laboratory, the Parsytec GC-el/1024 at Paderborn PC^2. Networks of high-performance workstations and a Meiko CS-2 will be considered in the future.

This work is sponsored in part by *ICI plc* through a collaborative project with the Advanced Research Computing Group at Daresbury Laboratory.

Contact Address

Dr. J.G. Carter
Dr. R.J.Blake
Daresbury Laboratory
Daresbury
Warrington
WA4 4AD, U.K.

phone: +44 925 603221
+44 925 603372
fax: +44 925 603634
email: j.g.carter@daresbury.ac.uk

4.7 Linear Programming

The Linear Programming Problem ($\min c^T x : Ax = b, x \geq 0$) is a most frequently given mathematical task in industry. Typical applications which are modeled as Linear Programs are eg. Trading and Cash Flow, Transportation and Logistic Problems, Manufacturing Processes, Machine Usage, Store-keeping and Mixture Problems in oil industry. Based on the efficient sequential implementations OB1 and MINOS 5.1 parallel versions of these algorithms have been implemented.

Based on OB1, an Interior Point Method (cf. [4]) was parallelized by performing in parallel the computationally most time consuming task of the algorithm, the sparse Cholesky decomposition (cf. [3]). A Fan–In Algorithm (cf. [1]) was used, which distributes the columns of the Cholesky Factor to the processors using a tree–like mapping scheme.

The algorithm was successfully ported to Helios and Parix. Speedups in the range of four to six could be obtained by performing the matrix generation and the numerical factorization in parallel, which are the best possible values for these routines. The parallelization of further procedures is in progress.

References

[1] C. Ashcraft, S.C. Eisenstat, J.W.-H.Liu. *A Fan-In Algorithm for distributed sparse numerical Factorization.* Siam J. Sci. Stat. Comput., 11, 3, 593 (1990).

[2] A. Bachem, M. Strietzel. *Eine parallele Implementierung des Karmarkar-Verfahrens.* Proceedings of the Transputer Anwender Treffen (TAT 92), Aachen, Springer (1992).

[3] M.T. Heath, E. Ng, B.W. Peyton. *Parallel algorithms for sparse linear systems.* Siam Review 34, 1, 82 (1992).

[4] K.A. McShane, C.L. Monma, D.F. Shanno. *An implementation of a Primal-Dual Interior Point Method for Linear Programming.* ORSA Journal of Computing, 1, 2, 70 (1989).

Contact Address

Prof. Dr. Achim Bachem
Michael Wottawa
Zentrum für paralleles Rechnen
Universität zu Köln
50923 Köln / Germany

phone: +49 221 470 6019
fax: +49 221 470 5160
email: wottawa@mi.uni-koeln.de

4.8 Mould Filling and Casting Simulation

The simulation of mould filling and casting processes challenge fluid dynamics specialists [1]. Unsteady, multi-phase flow, turbulence, intense heat transfer by all modes and phase change should be taken into account. These phenomena are described by the Governing Differential Equation that describe conservation principles for mass, momentum, energy, etc.

A suitable mathematical model for the numerical simulation is the Finite Difference Method, where the domain of interest (the casting mould) is decomposed into elementary Control Volumes (CVs). For the mould filling a SOLA-VOF type algorithm [2] is used, because it turns out to be best in terms of the ratio between accuracy and computational efforts.

The simulation of the filling process of a realistic mould with up to 2.5 million CVs (e.g. an aluminium cylinder head or car rim) requires the calculation of up to 10,000 time steps. This leads to excessive CPU times of some days on a 10 MFlops computer.

A first approach was the parallelizing of only the most time consuming part of the algorithm: the Jacobi iteration method for the solution of the very large linear equation systems. Therefore a domain decomposition of the mould box is used: each Transputer gets nearly the same number of CVs and has to communicate with its neighbours in every iteration step. Efficient load balancing is difficult, because in the sequential filling process increasing numbers of CVs are filled. Only these filled CVs are involved in most of the calculation procedures, so that a static decomposition is not suitable.

For the development of the parallel algorithm a Parsytec X'plorer with eight T805 is used. The Parix environment allows to reuse as much of the sequential Fortran code as possible. Testing and benchmarking takes place on the GCel-1024 at the University of Paderborn. The parallel algorithm reaches an efficiency of over 50 percent for a realistic casting example [3].

References

[1] D.M. Lipinski, W. Schäfer, E. Flender. *Numerical Modelling of the Filling Sequence and Solidification of Castings.* MAGMA Gießereitechnologie GmbH, D-52477 Alsdorf.

[2] C.W. Hirt, B.D. Nichols. *Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries.* Journal of Computational Physics 1981, pp. 201-225.

[3] F.M. Thiesing, M. Lipinski. *Parallelisierung eines Programms zur Simulation von Gießvorgängen.* Proceedings of the Transputer Anwender Treffen (TAT 93), Aachen, to be published in: Parallele Datenverarbeitung mit dem Transputer, Reihe Informatik aktuell, Springer 1994.

Contact Address

Dr.-Ing. Andreas vom Hemdt
Dipl.-Math. Frank M. Thiesing
ZIAM GmbH — Zentrum für Industrielle
Anwendungen Massiver Parallelität
Kaiserstr. 100
D-52134 Herzogenrath

phone: +49 2407 9567 0
fax: +49 2407 9567 15
email: ziam@infoac.rmi.de

4.9 Turboprop Compressor Fan Flow

Currently in the design of turbomachinery, only single components of a certain engine have been simulated, e.g., air flow through a compressor stage of a fan. The objective of this code is to integrate several stages of the engine into a single model and to achieve therefore more realistic results. The technical object will be a turbomachine with unsteady flow through counterrotating fans and unequally spaced compressor stages.

The ANSI-C program (30.000 lines) was developed under HELIOS on Super-Cluster SC-32. The program was then ported to PARIX 1.0 on the GCel-2/128 at the GMD/Bonn and later to the GCel-3/1024 at the University Paderborn. After installation of the GCel-3/1024 in Cologne, research is now continued at the ZPR.

Full scalability up to the maximum number of processors (e.g. with 4 processors: 4096 grid points; with 1024 processors: 1 Mio grid points). For a continuous film of time-dependent flow, i.e. 25 results/sec, one would need a TFlops machine.

References

[1] M. Faden, K. Engel, S. Pokorny, K. Wolf. *CFD Applications on Transputer Systems.* Technical Report DLR German Aerospace Research Agency, 1993.

[2] S. Pokorny, M. Faden, K. Engel. *An Integrated Flow Simulation System on a Parallel Computer. Part I: Basic Concepts.* Numerical Methods in Laminar and Turbulent Flow, Vol. VII, Part 2, Proceedings of the Seventh International Conference held in Stanford, July 15-19, 1991, Edts. C. Taylor, J.H. Chin, G.M. Homsy, Pineridge Press, Swansea, U.K. July 1991.

[3] M. Faden, K. Engel, S. Pokorny, K. Wolf. *An Integrated Flow Simulation System on a Parallel Computer. Part II: The Flow Solver.* Numerical Methods in Laminar and Turbulent Flow, Vol. VII, Part 2, Proceedings of the Seventh International Conference held in Stanford, July 15-19, 1991, Edts. C. Taylor, J.H. Chin, G.M. Homsy, Pineridge Press, Swansea, U.K. July 1991.

Contact Address

Dr. Michael Faden
Institut für Antriebstechnik
DLR e.V. phone: +49 2203 601 2285
Linder Höhe fax: +49 2203 601 2353
51147 Köln / Germany email: michael@kat01t.at.kp.dlr.de

4.10 Unsteady Flow about Oscillating Rotor Blades

On the retreating blade of a helicopter rotor in high speed forward flight the strongly time-dependent flow may separate. The flow is dominated by both viscous and compressible effects. Its calculation makes the solution of the complete (Reynolds averaged) Navier-Stokes equations necessary. The equations are solved on a deforming C-mesh, where the inner boundary (η=0) is attached to the pitching airfoil and the outer boundary ($\eta = \eta_{max}$) is represented by a space-fixed (prescribed) grid-line. The boundaries of the ξ-coordinate, i.e. $\xi = 0$ and $\xi = \xi_{max}$ are representing the outflow boundaries. Due to the C-mesh topology the number of grid points along airfoil and wake (ξ-coordinate) is considerably larger compared to the number of grid points normal to the airfoil (η-coordninate) although due to the viscous flow (turbulent boundary layer) the region adjacent to the airfoil and wake surfaces have to be resolved with very small gridspacing ($\triangle, \eta_1$=0.00005).

The numerical method to solve the flow-equations is the Beam/Warming approximate factorization implicit method. Details of this method are discussed in [1] where a number of results and comparisons with experimental data are given.

In priniciple there are two possibilities for parallelization. Either one solves the unfactored problem directly hereby taking advantage of getting an easier flexible parallel code but taking disadvantage of loosing the internal structure of the systemmatrix. Or one solves the two factored subproblems with internal structure. Here the existing serial code based upon the factored problem is the basic for parallelization.

The design and the implementation on the cluster is done rather fast due to simplified formulations. Summarized an *implicit* second order time accurate, noniterative and spatially factored algorithm based on Beam and Warming was basically parallelized.

References

[1] W. Geißler. *Instationäres Navier-Stokes Verfahren für beschleunigt bewegte Profile mit Ablösung.* DLR-FB 92-03, 1992.

[2] *Time accurate solution of the Navier-Stokes equations for unsteady two- and three-dimensional flows about oscillating rotor-blades with separation.* Internal Report DLR, 1993.

Contact Address

Dr. A. Coriand
DLR, High Performance Computing
Bunsenstr. 10 phone: +49 551 709 2155
37073 Göttingen email: coriand@dlr.de

Dr. W. Geissler
DLR Institute of Fluid Mechanics
Bunsenstr. 10
37073 Göttingen phone: +49 551 709 2353

4.11 VHDL Simulator

VHDL (Very High Speed IC Hardware Description Language) is a standardized (IEEE) language for the description of IC designs. It is commonly used in all laboratories and companies throughout the world for the description of hardware on all description levels (from gate level to algorithmic level). As simulations of IC designs are very time consuming it is necessary to use massively parallel computers to achieve reasonable computing times for practical applications. The parallel simulation is strongly supported by VHDL, since VHDL uses a process oriented approach to describe integrated circuits. This makes it appropriate to map the circuit-graph onto a network of communicating processors.

Our program uses the "University of Pittsburgh CAD tools" which extract the necessary informations out of a VHDL description and build a set of C-procedures which describe the single gate, register or algorithmic block of an IC design. Since this tool does not support the complete VHDL language definition, it is only possible to simulate circuits which are described by a subset of VHDL. Since this subset is very large, this is no serious constraint.

For the simulation of the circuit one first has to partition the circuit and map it onto the processor network. The partitioning itself is done by a parallel algorithm described in [2]. This algorithm uses the idea of helpful sets, already described in [3] and [4] and extends this in different ways. First we used the general simulated annealing principle to increase the efficiency of the algorithm. Another extension, which has been shown to be very powerful is made by the use of so called "meta-steps". In a meta step of the simulated annealing algorithm, all subgraphs of the networks which have already been clustered are identified as one weighted node of the so called "meta-graph". This graph is used for the next partitioning step. As this graph is much smaller than the original circuit graph, one can use methods different from simulated annealing, yielding a better partitioning result. We used this method for a huge number of benchmark applications and got improvements compared to other heuristic methods, known before.

The use of this methods, together with different synchronization techniques leades to an increased performance of the overall simulator. As the overall performance of these synchronization algorithms is strictly depending on the structure of the circuit graph (an effect that is well known, and largely described in literature) the speedup of the overall simulation is still strictly depending on this network structure and varies between efficiencies of 20 to 60 percent.

References

[1] R. Diekmann, R. Lüling, J. Simon: *Problem Independent Distributed Simulated Annealing and its Applications*, Proc. of 4 th IEEE Symposium on Parallel and Distributed Processing (SPDP), 1992, pp. 94-101

[2] R. Lüling, C. Spräner: *A massively parallel graph partitioning algorithm*,

manuscript, 1994

[3] J. Hromkovič, B. Monien: *The Bisection Problem for Graphs of Degree 4* Proc. of 16th Math. Foundations of Computer Science (MFCS '91), Springer LNCS 520, pp. 211-220

[4] R. Diekmann, B. Monien, R. Preis: *The HS-Heuristic: A Powerful New Approach to Graph Partitioning*, submitted for publication, 1994

Contact Address

Prof. Dr. B. Monien, Reinhard Lüling
Universität-GH Paderborn
FB 17 Mathematik/Informatik
33095 Paderborn / Germany

phone: +49 5251 60 3331
fax: +49 5251 60 3515
email: rl@uni-paderborn.de

5 Benchmarks

Various benchmark suites have been implemented and run on the MPP systems of the ZEUS centers. Emphasis is placed on the performance evaluation of *massively* parallel systems with a thousand processing elements. There are a number of program suites under development, e.g., from the RAPS consortium, which will be tested on the larger machines.

5.1 Linpack Benchmark

A LINPACK code from the Shell Laboratories Amsterdam was ported to the GCel 3/1024 at the University of Paderborn. Source Code optimizations finally lead to a performance of *994 MFlop/sec*. The measurements were made in close cooperation between Parsytec and the University of Paderborn.

In large-scale scientific computation, the solution of linear systems of equations is of fundamental importance. On sequential computers, however, it is often a time-consuming exercise. A parallel computer can provide an attractive, cost-effective alternative. To measure the efficiency of supercomputers, benchmarking of these systems is necessary.

Therefore, the best LINPACK benchmark performance was chosen. This is the most widely spread and used benchmark for supercomputers. The original sequential LINPACK benchmark was introduced by Jack Dongarra. The problem is to solve a dense system of equations. It is allowed to optimize the program code in any way and to scale the problem size to achieve the maximal performance of the parallel system.

With the above results, the GCel-1024 is at the 259 th position in the Top500-List world wide. In Germany it would be position 22 (July 1, 1993). In the updated list from November 11, 1993, the GCel-1024 is at position 309 (Germany: 26).

References

[1] Jens Simon. *LINPACK: Parallel Benchmark on PARIX.* Technical Report PC2/TR-003-94, 1994.

[2] J.J. Dongarra, H.W. Meuer, E. Strohmaier. *TOP500 Supercomputer Sites.* Technical Report (Nov. 1993), Univ. Mannheim, Germany, ftp-Server top500@rz.uni-mannheim.de.

Contact Address

Jörg Schepers, Parsytec GmbH
Jens Simon, PC^2

Paderborn Center for Parallel Computing	phone:	+49 5251 60 3894
Universität-GH Paderborn	fax:	+49 5251 60 3436
D-33095 Paderborn / Germany	email:	jens@uni-paderborn.de

5.2 NAS Parallel Benchmark

The NAS Parallel Benchmark Suite consists of the five kernel benchmarks: Embarassingly Parallel (EP), Multigrid (MG), Conjugate Gradient (CG), 3-D FFT PDE (FT) and Integer Sort (IS) and the three Simulated CFD Application Benchmarks: Lower-Upper Diagonal (LU), Scalar Pentadiagonal (SP) and Block Tridiagonal (BT). Initial work focussed on the kernel benchmarks.

A parallel version of the EP Benchmark was implemented under PARIX. Measurements on the GCel-3/1024 at the University of Paderborn proved that, in terms of performance, the GCel is comparable to high-end supercomputers. For instance, the Paderborn machine reaches 3.6 times the performance of a CRAY Y-MP.

The next step was the design of suitable parallel algorithms for CG, FT and IS matching to the GCel/GC-architecture, and their implementation under PARIX. A description of the developed algorithms for IS and FT together with preliminary results for the Parsytec GCel are given in [1] and [2].

We also implemented the NPB kernels on the Parsytec *Power Xplorer*. A Power Xplorer node consist of Motorola MPC 601 computing processors and transputer based communication devices. We used a four node system with an early version of PARIX, supporting a coprocessor programming model. The computing and the communication processors must be programmed explicitly. The multiprocessor node architecture of the Power Xplorer requires a special coding of the benchmark programs to achieve high performance. Due to the small memory size of our current Power Xplorer, many full-size applications can not be executed on our system.

References

[1] Markus Sigg. *The NAS Integer Sort Benchmark Implementation and Results for a Parsytec GCel Machine.* Technical Report, Parsytec, Oct. 1992.

[2] Jörg Schepers. *Implementation des Kernels FT aus den NAS Parallel Benchmarks auf dem Parsytec GCel und dem Parsytec GC.* Techn. Report, Parsytec GmbH, August 1993.

[3] Jens Simon: *LINPACK: Parallel Benchmark on PARIX.* Technical Report PC2/TR-003-94, 1994

[4] Jens Simon. *The NAS Parallel Benchmarks on PARIX.* Technical Report PC2/TR-004-94, 1994

[5] Jens Simon. *Preliminary Performance Figures of the Power Xplorer with PARIX/PowerPlus 1.0.* Technical Report PC2/TR-005-94, 1994

Contact Address

Jörg Schepers
Parsytec Computer GmbH
Jülicher Str. 338
D-52070 Aachen, Germany

phone: +49 241 166000
fax: +49 241 16600 50
email: schepi@parsytec.de

Jens Simon
PC^2 – Paderborn Ctr Parallel Computing
Warburger Str. 100
D-33095 Paderborn, Germany

phone: +49 5251 60 3894
fax: +49 5251 60 3436
email: jens@uni-paderborn.de

6 Application Interfaces

A steadily growing community of programmers demands for homogeneous user- and application interfaces for the various MPP systems. New parallel systems will only be accepted when the software environment (operating system, compilers, dynamic load balancers, mapping tools, system administration tools) becomes standardized and easier to use.

The tasks reported in this section aim at providing a homogeneous parallel programming environment. Typical examples of such software packages include the Paderborn Computing Center Software, which is installed as a common software environment in Paderborn, Cologne and Athens, and the PVM standard programming environment running in Amsterdam. User workshops like the Ca*mpp*s held in Paderborn and Amsterdam provide valuable feedback on the usability of the machines.

6.1 Computing Center Software

Nowadays, the integration of parallel systems into an existing network is far from being a trivial task. CCS is an innovative hardware management platform which was primarily designed to alleviate this problem. First experiences with the software package were gathered at the Paderborn Center for Parallel Computing, running two transputer systems with *320 processors* and *1024 processors*. The systems are accessed by more than 160 researchers from all over Europe.

The management software developed is not limited to transputer systems. It is applicable to heterogeneous computer environments equipped with workstation cluster and/or MPP systems, thus giving a maximal exploitation of parallel resources by both, its owner and the users. Only after having solved this problem, one can be certain to have the necessary foundation on which larger distributed computing environments can be built.

CCS is implemented as a distributed application. By exchanging messages between cooperating Unix daemons, it controls itself as well as the machines served. With this distributed approach integrating new MPP systems into an existing environment is simplified. In a first step, Parsytec GCel-Systems can be used by CCS. However, the features available are impressive.

Features of CCS:

- For user of MPP systems:
  - *Guarding against unstable connection lines:*
    Applications continue running after a broken WAN or phone-line connection, allowing later access without losing output.
  - *Virtual view of the machinery:*
    Users do not need to face technical system details or to access the system software.
  - *System independent 'look and feel':*
    There is only one common interface to access different MPP systems.
  - *Batch processing:*
    The batch processing facility provided can be used within modern cross-environments. It is independent of the programming tools used or the MPP system accessed.
  - *Optimized request scheduling:*
    The virtual metacomputer system is scheduled by an implicit voting system. That means that the user implicitly vote if the scheduler should increase the overall system utilization (batch jobs dominate) or reduce the average waiting time (interactive jobs dominate).
  - *Dynamic partitioning of GCel systems:*
    CCS configures the systems on demand, optimizing the request placement onto its architecture.

- *Miscellaneous:*
  CCS supports ASCII and X11 graphical terminals. An integrated online information tool gives answers to most users' questions.

- For operators:

  - *Central user and machine management:*
    There is only one configuration file necessary to specify the whole metacomputer. To make this task more easily, a high level Resource Description Language was developed.

  - *Full resource and user control:*
    At any point of time, it is possible to control user activities and to guide the request behavior.

  - *Only one operator shell for the whole metacomputer:*
    The CCS environment can be modified by an operator from a menu driven shell. Thus, simplifying the management and improving the reliability.

  - *Hardware configuration transparent to the user:*
    The user community should not be maltreated by maintenance activities. Due to the strongly differentiation between physical available and logical requested resources this feature could be reached.

  - *Logfiles for user and machine activities:*
    By tracing user and system activities it is possible to detect critical actions or malfunctioning system components.

  - *Accounting and charging tools:*
    These tools are indispensable if MPP systems are commercially used or if system statistics must be presented.

- For vendors:

  - *Seamless environment for different system families:*
    Experience has shown that new machine families are accepted well if widely used software packages are available and if the surface of the new systems behaves similar to that what is already known.

  - *Breakthrough within the MPP management:*
    The features above will bring the ideas of parallel computing a step further. Accepting these systems as manageable tools, will have a great impact on the number of systems sold.

References

[1] B. Bauer, F. Ramme : *A general purpose Resource Description Language,* Informatik aktuell, Hrsg: R. Grebe, M. Baumann. Parallel Datenverarbeitung mit dem Transputer, Springer, Berlin, 1991, 68–75.

[2] K. Kremer, F. Ramme : *Scheduling a Metacomputer by an Implicit Voting System,* 1994, submitted.

[3] F. Ramme, T. Römke : *Rechenzentrums-Software für Parallelrechner,* Informatik Aktuell, Hrsg: R. Grebe, M. Baumann. Parallel Datenverarbeitung mit dem Transputer, Springer, 1992, pp. 72-82

[4] F. Ramme, T. Römke : *The Computing Center Software, a Step Towards Metacomputing,* Tech. Rep. PC2/TR-007-94, Paderborn Center for Parallel Computing, Univ. Paderborn, March 1994.

[5] F. Ramme, T. Römke, K. Kremer : *A Distributed Computing Center Software for the Efficient Use of Parallel Computer Systems,* HPCN Europe'94, to appear

Contact Address

Friedhelm Ramme
Thomas Römke
Paderborn Center for Parallel Computing
D-33095 Paderborn

phone: +49 5251 60 3359
fax: +49 5251 60 3436
email: ram@uni-paderborn.de

6.2 FORTNET Message Passing Harness

An optimised version of the portable FORTNET message passing harness [3] has been installed on the Parsytec GC-el/1024. This is a thin layer which interfaces Fortran-77 applications to the PARIX C library and replaces the PARIX Fortran layer. An initial implementation was done on a 16-node Supercluster loaned by Parsytec and during a visit to Paderborn PC^2 and Parsytec Computer GmbH. This has now been extended to allow timing information to be gathered for the *Dynamic Load Balancing in CFD* project described elsewhere in this report. An improved way to handle dynamic link creation has also been implemented giving better performance.

It has been found that the original concept of FORTNET [1–2] is very close to the PARIX message-passing implementation which incorporates dynamic link creation on the GC-el in routing software. In FORTNET this link creation is handled using calls to a pair of routines CHECK(m) and WAIT(n) to establish a link and block all other communicating processes from processors m and n until a new link is established. A table of used links is maintained as an interface to PARIX. Asynchronous communication can then take place using the routines SEND(n,size,data,strong_type) and RECEVE(m,size,data,strong_type) which are another communicating pair of routines. RECEVE waits until the message arrives. The explicit blocking mechanism in FORTNET, the use of strong data typing and the use of explicit source and target addresses in the calls is unique.

A large number of timing interface routines are available to measure CPU and Wall (elapsed) time over the computation and communication parts of a program. In addition an interface is available to the Oak Ridge National Laboratory ParaGraph performance display tools.

References

[1] R.J. Allan, E.L. Heck, and S. Zurek, *Parallel Fortran in scientific computing: a new occam harness called FORTNET*, Computer Physics Communications 59 (1990) 325-44

[2] R.K. Cooper and R.J. Allan, *FORTNET (3L) v1.0: Implementation and extensions of a message passing harness for transputers using 3L Parallel Fortran*, Computer Physics Communications 70 (1992) 521-43

[3] R.J. Allan, *FORTNET v4.0: the Parallel Programming Software*, Daresbury Laboratory (June 1992)

Contact Address

Dr. R.J. Allan
Daresbury Laboratory
Daresbury
Warrington
WA4 4AD, U.K.

phone: +44 925 603207
fax: +44 925 603634
email: r.j.allan@daresbury.ac.uk

6.3 Load Balancing for CFD

We report results of a project to implement a dynamic load-balancing scheme in a production CFD code with 2D domain decomposition in real space. CAVITY, a typical CFD engineering application code, is the production version of the code already described elsewhere in this report. Load balancing is particularly important in applications such as CFD because of extremely long execution times which may be required to solve problems involving complex physics or time-accurate solutions. It is also for this reason that parallel solutions were considered.

The CAVITY code is run regularly on an Intel iPSC/860 hypercube, and for this use static load balancing would generally be satisfactory, perhaps based on a cost-matrix built from accumulated run-time statistics. However a number of parallel applications in science and engineering are now being developed worldwide and are being run on homogeneous (i.e. a cluster of identical) UNIX workstations coupled by local area networks (LAN). This is a potentially attractive prospect since workstation cpus are often only partly used interactively, and may sometimes be entirely unused during night-time periods, especially in educational environments or small businesses.

These systems run a multi-user UNIX operating system and cpu time is shared among many processes making total run times variable. Many of the recent closely-coupled parallel systems also run a multi-user UNIX o/s on all nodes and, unless tackled by the o/s (e.g. Kendal Square KSR1 and 2), load balancing will be needed in applications. Examples include the Intel Paragon, IBM SP1, Meiko CS2 etc. A further incentive is to make use of different capabilities of networked UNIX hosts, perhaps combining a compute server, a disk server and a high performance graphics engine [e.g. 1]. A number of ad hoc methods involving UNIX sockets have been used in the past to achieve this.

A networked workstation cluster is an inherently dynamical multi-user system and the code developed faces several difficulties if good parallel efficiency is to be obtained. Extra problems arise moreover in a heterogeneous assembly. Important factors for both homogeneous and heterogeneous systems are:

- Network traffic - which is dynamically varying can slow down communication between some or all processors
- Differences in network bandwidth - on a multi-branch LAN
- Differences in processor speed - in a heterogeneous or quasi-heterogeneous cluster
- Local cpu load - may change dynamically, e.g. workstation owner may run a big sequential job at high priority

- Local memory load - may change dynamically, e.g. workstation owner may start a big edit or backup job leading to disk paging and loss of performance
- A workstation may be switched off or become inaccessible for some reason, requiring checkpointing and program restart with a different processor configuration. We suggest how the checkpoint file can be stored.
- Different number representations - in a heterogeneous cluster translation between binary representations takes extra time

For all the reasons listed above a fully dynamic load-balancing scheme is essential. The example in this paper is typical of an explicit time-marching production code with a grid which is topologically equivalent to a regular grid. However the different computational components around the cavity region (see below) and message-passing make the balance conditions complex and it is not possible to simply parameterise the problem to determine the optimal decomposition. This is true of any flow problem with complex geometry and boundary conditions.

The algorithms used in the CAVITY code and its domain decomposition for parallel execution have been described elsewhere in this report.

We have adopted an hierarchical approach to load balancing in a multi-dimensional domain decomposition. In 2D this amounts to alternately balancing in the x and y directions. The y dimension is divided into full layers (slabs), whilst the x dimension is divided into rectangular blocks within each layer. Balancing in x involves moving the vertical block boundaries in a horizontal direction independently in each layer, a parallel activity. Balancing in y involves exchanging information within a complete slab to find the total time spent in the slab calculation, and then moving the horizontal slab boundaries in the vertical direction to balance. In both cases the process is similar, and can readily be extended to more than two dimensions.

To determine which subdomain boundaries are required to move, the neighbouring domains communicate the elapsed (wall) time spent in calculating the previous iteration, including communication in sending and receiving data. The difference between this time and the average time for each three neighbouring domains indicates the amount of work (number of cells) which must be gained or lost before the next iteration. Domain boundaries are thus moved accordingly in each dimension independently of the others, halo data is updated, and the process continued. For simplicity of the implementation boundary movement is restricted to at most two cells, so that existing routines can be used to build new halo data.

A measure of the balance is derived as follows - note that the times must include computation and communication times on each processor, but not the time taken waiting for another processor to respond. The FORTNET message passing harness [2] has been specially modified to give this information.

$\delta_i = abs((t - t_i))/t$
$\delta = max(\delta_i)$ where t is the average time per processor and t_i is the time used on the ith processor.

once δ falls below a given value, say 0.05, we consider a good enough balance to be achieved. There is a practical limit to this, and how to determine it is discussed in the full paper [3].

Typical production runs would involve more than 10,000 iterations on a cluster of workstations of relatively similar power. The following procedure has been found satisfactory. We determine how much imbalance exists every five iterations or so by communicating all elapsed times for the previous five iterations to node 1. If any differ by more than 10% from the average the balancer is invoked. This allows quite rapid adjustment to varying conditions during a long run. On a machine such as the Intel hypercube the procedure can be stopped once a satisfactory balance condition exists. On other systems we assume the average system load (related to s in the analysis of performance metric) is slowly varying compared to the rate at which iterations are calculated. In this case once balance is achieved a greater number of iterations can be performed (say 100 or more) before it is again checked.

Initial tests and demonstration of the system were performed on a 16–processor Parsytec Supercluster on loan to us at Daresbury Laboratory. This enabled the implementation to be done easily and de-bugged along with the optimised FORTNET message passing harness which is used for portability. The same code, in a production version, was ported onto the Parsytec GCel/1024 at Paderborn PC^2. On this machine, and other multicomputers, a load balancing strategy is required if the exact decomposition cannot be predicted, for instance due to extra work involved in the boundary regions with a complex geometry. Once a balanced situation is achieved however the balancer can be switched off to reduce overhead during the remainder of a long production run. Even a small improvement in performance is significant, as compared to the unbalanced problem, because run times may be very long.

To demonstrate the code we started with a very unbalanced 4–processor decomposition. A grid of size 128x64 was used. Processors 2 and 3 start with respectively too much and too little work. As balancing takes place all computational loads equalise until a maximum performance is achieved. Some oscillation may occur once a condition is achieved which can no longer be improved.

This work formed part of the requirements leading to an M.Sc. degree in parallel computing methods awarded by the University of the West of England to Mr. J.Garner in 1993 [3]. Detailed results have been submitted for publication [4].

References

[1] Allan, R.J., Durham, P.J. and Guest, M.F. *Networked Computer Power* Physics World 4 (1991) 51-4

[2] R.J. Allan, *FORTNET v4.0: the Parallel Programming Software*, Daresbury Laboratory (June 1992)

[3] Garner, J. *The Development of a Two Dimensional Load Balancing Strategy and its Implementation* M.Sc. Thesis, University of the West of England (1992)

[4] J. Garner, R.J. Allan, R.J. Blake and D.R. Emerson, *Dynamic Load Balancing in heterogeneous workstations clusters*, submitted to 8th ACM Conference on Supercomputing, Manchester 11-15/7/94

Contact Address

Dr. R.J. Allan
Dr. D.R.Emerson

Daresbury Laboratory	phone:	+44 925 603221
Daresbury		+44 925 603207
Warrington	fax:	+44 925 603634
WA4 4AD, U.K.	email:	r.j.allan@daresbury.ac.uk

6.4 Load Balancing in Optimization Problems

A load balancing test suite for parallel load balancing algorithms has been developed that allows the user of a parallel machine to specify load parameters of his specific application and evaluate the various algorithms, their scalability and their effects on application performance. The load balancing algorithms have been tested with combinatorial optimization problems, which spawn highly unbalanced search trees. The work load of the single processors is hard to estimate in advance, and dynamical load-balancing schemes must be used. Three typical branch-and-bound applications have been implemented on the GCel-1024:

- the vertex cover problem,
- the traveling salesman problem,
- iterative-deepening search.

With a generic load-balancing scheme, we have been able to show that (1) even small instances of the vertex cover problem can be solved with 90% efficiency, (2) the traveling salesman problem can be solved with 80% efficiency (based on a comparison to the most efficient sequential algorithms) and (3) iterative-deepening search runs with more than 80% efficiency on a ring topology with 128 processors and more than 90% efficiency on a wrap-around-mesh of 512 processors.

References

[1] R. Lüling, B. Monien. *Load balancing for distributed branch-and-bound algorithms.* Uni-GH Paderborn, Techn. Rep. Nr. 114, (Febr. 93).

[2] R. Lüling, B. Monien, M. Räcke, S. Tschöke. *Efficient Parallelization of a Branch & Bound Algorithm for the Symmetric Traveling Salesman Problem.* European Workshop on Parallel Computing, EWPC 92, Barcelona.

[3] R. Lüling, B. Monien. *A Dynamic Distributed Load Balancing Strategy with Provable Good Performance.* ACM Symposium on Parallel Algorithms and Architectures 1993.

[4] A. Reinefeld, V. Schnecke. *Work Load Balancing in Highly Parallel Depth-First Search.* Scalable High Performance Computer Conference SHPCC, Knoxville 1994.

Contact Address

Prof. Dr. B. Monien
Reinhard Lüling
Dr. Alexander Reinefeld
Universität-GH Paderborn
FB 17 Mathematik/Informatik
D-33095 Paderborn

phone: +49 5251 60 3331
+49 5251 60 3341
fax: +49 5251 60 3436
email: rl@uni-paderborn.de

6.5 MAGE: A Massively Agent Execution System for Logic Programming

A suitable programming environment for the implementation of multi-agent applications on massively parallel platforms, called MAGE has been developed on the parallel machine of Athens-HPCL. The MAGE environment fully exploits the potential of massively parallel computing, prevents the programmer from low level dependencies i.e. processor topology, low-level communication, load balance, scheduling of processes etc. while also facilitates the programming task of complex applications through the advantages of the multi-agent approach. The general approach that has to be followed using MAGE for the development of a multi-agent system is to build a separate subsystem called agent for each problem domain and, then make these subsystems cooperate. The run-time part of MAGE is concentrated on the coordination of a massive number of these agents executed on a parallel machine with hundreds of processors. In order to improve the performance, MAGE provides techniques to equally divide the workload among the different agents. More precisely, MAGE allows the programmer to create statically, a specific number of multiple copies (instances) of the same agent according to its anticipated demand where these copies are executed in parallel. MAGE system consists of three main parts. The first part is the MAGE Prolog language extended with communication primitives together with the compiler of this Prolog to an intermediate abstract machine code. The second part is the MAGE execution system running on the massively parallel machine that executes the abstract machine code and manages both inter-agent communication and parallelism. The third one is the MAGE library used by existing Prolog systems and supports the implementation of external agents and their connection with the parallel execution system of MAGE.

MAGE system is currently implemented on the Parsytec GCel-512 machine of Athens-HPCL under the Parix operating system and using the C language that Parix supports. This massively agent programming environment realizes:

- great abstraction power over the parallelism
- stronger encapsulation mechanism and easier implementation using the agent partitioning method.
- great exploitation of the parallelism and achievement of very high performances. MAGE supports parallelism between the different agents that the whole application has been divided (coarse-grained parallelism) and parallelism between copies of the same agent (fine-grained parallelism).
- traditional Logic Programming techniques within every agent.
- transparent connection with external agents running on different hardware platforms using the external agent interface.

In a future step, we will introduce the blackboard concept as a broadcast medium in the MAGE system for the applications that require this facility in order to improve

the implementation power of the system. In addition, we will design a more dynamic execution environment of MAGE where the agent instances will be created and destroyed according to the demand. Currently, we are working on the porting of a large multi-agent application named MaTourA which is a tourist information advisor system about Greece and has been developed in the context of the ESPRIT 6708 APPLAUSE Project (Application & Assessment of Parallel Programming Using Logic).

This work is sprung from our involvement in the APPLAUSE Project concerning the development of a large multi-agent application.

References

[1] C.Mourlas and C. Halatsis. *MAGE: A Massively Agent Execution System for Logic Programming.* In Parallel Architectures and Languages Europe (PARLE'94) Conference, 1994. (to be presented in the poster session)

[2] P. Stamatopoulos, D. Margaritis, and C.Halatsis. *Extending a Parallel CLP Language to Support the Development of Multi-agent Systems.* In ACM Symposium on Applied Computing (SAC'94), 1994. (to be presented).

[3] C. Halatsis, P. Stamatopoulos, I. Karali, C. Mourlas, D. Gouscos, D. Margaritis, C. Fouskakis, A. Kolokouris, P. Xinos, M. Reeve, A. Veron, K. Schuerman and L.-L. Li. *MaTourA: Multi-agent Tourist Advisor.* In Intl. Conf. Information and Communication Technology in the Field of Tourism (ENTER '94), 1994.

[4] A.Bond and L.Gasser, editors. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, Inc., San Mateo, California,1988.

Contact Address

Prof. Constatin Halatsis University of Athens
Department of Informatics
Panepistimiopolis, TYPA Building
157 71 Athens / Greece

phone: +301 7217941
fax: +301 7219561
email: halatsis@di.uoa.araidne-t.gr

6.6 PVM on Parix

The PVM programming environment (Version 3.1) has been ported to PARIX at the University of Amsterdam. Based on this work, an agreement between UvA, Parsytec and the German software company GENIAS was established to undertake all the necessary steps towards a commercially available, professional PVM implementation for Parsytec MPP systems with PARIX. Official product status is given in October 1993.

There are two implementations of PVM on transputers: heterogeneous and homogeneous. The heterogeneous version, that allows for inclusion of a Sun host of a Parsytec GCel into a PVM network, was delivered in September 1993. The homogeneous version is currently being optimized.

Contact Address

Dr. P.M.A. Sloot
Dept. of Computer Systems
Universiteit van Amsterdam — phone: +31 20 525 7463
Kruislaan 403 — fax: +31 20 525 7490
1098 SJ Amsterdam / The Netherlands — email: peterslo@fwi.uva.nl

6.7 Virtual Topologies

The *Virtual Topology Library* extend the PARIX operating system by a flexible and dynamic means of handling user defined topologies. The following commonly used topologies have been included in the virtual topology libraries of PARIX:

- pipe
- ring
- grid
- torus
- tree
- hypercube
- clique
- star

Design and implementation of the virtual topology library is based on theoretical results on the optimal mapping of the above topologies onto the GCel's grid interconnection structure. The library functions use the efficient, reliable and simple message passing kernel of PARIX.

References

[1] J. Simon, J. Gehring. *Virtual Topology Library for PARIX.* Technical Report, PC2/TR-005-93.

[2] J. Simon. *Benutzung virtueller Topologien unter PARIX.* Technical Report, PC2/TR-006-93.

Contact Address

Prof. Dr. B. Monien
Dipl. Inform. Jens Simon
Paderborn Center for Parallel Computing — phone: +49 5251 60 3894
Universität-GH Paderborn — fax: +49 5251 60 3436
33095 Paderborn / Germany — email: jens@uni-paderborn.de

6.8 Campp-Projects

Ca*mpp* stands for *Challenging Applications for MPP*, a joint organization of Parsytec and the ZEUS centers. Two Ca*mpp*s have been organized so far: A *Summer–Campp 92* at the PC^2 in late summer 1992, and a *Spring–Campp 93* at the IC^3A in Amsterdam. The goal of that initiative was to make the newly installed machines accessible by a larger group of researchers interested in using and experimenting with such large MPP systems.

Users with suitable applications were invited to two-day workshops held in Paderborn and Amsterdam, in which the machine's software-environment and parallel programming concepts of PARIX were introduced, and successively given (limited) amount of computing resources. The Ca*mpp*s turned out to be a valuable experience for both participants and organizers. About 20 user groups (each) with applications from various fields (numerical algorithms, CFD, FEM analysis, statistical and high-energy physics, microbiology, a.o.) worked with the system, and some spectacular results could indeed be achieved within a short amount of time. For instance, in Paderborn a new world record in lattice size of 2D and 3D Ising lattice simulations was achieved by a research group from Heidelberg. In other cases, the participation in Ca*mpp* lead to an intensified and continuing collaboration with the ZEUS centers.

In view of the success of the first two Ca*mpp*-Projects, the ZEUS centers will continue such activities on the basis of a common, Europe-wide activity. Ca*mpp* activities will be organized at the different centers, and it is planned to disseminate the experiences of the project in a special issue of a scientific journal.

Contact Address

Mahmoud Chatah — phone: +49 241 166000
Parsytec GmbH — email: mahmoud@parsytec.de

Dr. P.M.A. Sloot — phone: +31 20 525 7463
Universiteit van Amsterdam — email: peterslo@fwi.uva.nl

Prof. Dr. Constantin Halatsis — phone: +301 72 17 941
University of Athens — email: halatsis@uranus.di.uoa.ariadne-t.gr

Michael Wottawa — phone: +49 221 4704341
Universität zu Köln — email: wottawa@mi.uni-koeln.de

Dr. Alexander Reinefeld — phone: +49 5251 60 3341
Paderborn Center for Parallel Computing — email: ar@uni-paderborn.de

7 Table of Projects

Associative Database Comparison	PC^2
CEVATS	ZPR/DLR
CFD with parallel FEM	PC^2 ZIAM
Campp-Projects	IC^3A PC^2
Chess	PC^2
Computer Vision	AHCP
Computing Center Software	PC^2
Crew Management	AHPC
Crystallization	IC^3A
DSMC Method of Gas Flow	ZPR/DLR
DYNA 3D	PC^2 ZIAM
Digital Signal Processing	AHCP
Elastic Light Scattering	IC^3A
Electro Magnetic Simulations	ZPR/DLR
FORTNET Message Passing Harness	PC^2 SERC
Factorisation	ZPR
GEANT	IC^3A
Graphics and Animation	PC^2 ZIAM
Grid Optimization	AHCP
Incompressible Turbulent Flow	PC^2 SERC
Kinetic Ising Model	PC^2
Linear Programming	ZPR/DLR
Linpack Benchmark	PC^2
Load Balancing for CFD	PC^2 SERC
Load Balancing in Optimization Problems	PC^2
MAGE: A Massively Agent Execution Sys. for Logic Progr.	AHPC
MaxHom	PC^2
Monte Carlo Simulation of Quantum Spin Glases	ZPR
Mould Filling and Casting Simulation	PC^2 ZIAM
NAS Parallel Benchmark	PC^2
Neural Networks in the Clinical and Psychiatric Field	AHPC
Numerical Simulation of Unsteady Flows	AHCP
Optimization of Large SQL-Queries	AHCP
PVM on Parix	IC^3A
Pivot	IC^3A
Recursive Partitioning	ZPR
Sheared and Stratified Turbulences	ZPR/DLR
Similarity Search of 3-D Chemical Databases	PC^2 SERC
Simulation of Traffic Congestions	ZPR/DLR
TSP-Heuristics	ZPR/DLR
Time Warp	IC^3A
Turboprop Compressor Fan Flow	ZPR/DLR
Unsteady Flow about Oscillating Roto	ZPR/DLR
VHDL Simulator	PC^2
Virtual Topologies	PC^2

Index