

InSiDE

inSiDE • Vol. 1 No. 1 • Spring 2003

Innovatives Supercomputing in Deutschland



Editorial

Editorial

With this first issue of inSiDE the German Federal Supercomputing Centers in München (LRZ), Jülich (NIC), and Stuttgart (HLRS) are launching an initiative to disseminate information about the new concepts of supercomputing as are embraced by the German supercomputing research community. Twice a year inSiDE will present information on supercomputing in Germany to its national and international readers. This first issue was published at the International Supercomputer Conference (ISC) in Heidelberg. The second issue is due for SC'03 in Phoenix/AZ.

With this biannual magazine the three centers intend to inform users and the interested community about recent developments in Germany and international supercomputing. Beyond its scientific contents inSiDE aims to reach out to a wider group of readers that might benefit from and have an interest in the progress made in our field.

The title inSiDE reflects the attitude of the magazine. We want to present an inside view of scientific investigation and its connection to modern supercomputing. inSiDE will thus present the work of the computing centers as well as the work of the scientists using the leading edge facilities available there. It will cover discussions about computer architectures as well as discussions about modeling. By sharing our work with a wider community we want to open up the temples of supercomputing to a wider readership and at the same time promote the advancement of German science by supercomputing.

To foster such an integration of computing and science one focus of inSiDE will be on integration of simulation and visualization. Bringing the researcher inside the simulation and making his immediate feedback part of the computational experiment is one of the key challenges for the years to come. This includes a synergy of computer science, applied mathematics, knowledge about handling of supercomputers and the relevant fields of science.

In this first issue you will find information about the available supercomputer systems at the three centers and the workshops offered by the centers for interested users. A number of articles present the work of users and their applications. Three short contributions were chosen from young researchers of the HLRS who have received a "Golden Spike" award for outstanding work in supercomputing in 2002. A third part of this first issue is dedicated to the hotly debated issue of Grid computing. The German supercomputing community was an early adopter of this concept. With the launch of the UNICORE project in 1997 and UNICORE+ in 2000 Germany is well ahead of her competitors in deploying Grid computing in a production environment. Its usage in European projects and in the Japanese lead-project NAREGI (National Research Grid Infrastructure) reflect the quality of research work in Germany and the close cooperation between the German, European, and Japanese research community.

Prof. Dr. H.-G. Hegering (LRZ)
Dr. B. Mertens (NIC)
Prof. Dr.-Ing. M. M. Resch (HLRS)

Contents

Editorial

Contents

1. Applications

Aeroelastic Analysis of Helicopter Rotor Blades using HPC	4
A Vectorised Lagrangian Particle Model for the Numerical Simulation of Coal-Fired Furnaces	6
Massively Parallel DNS of Flame Kernel Evolution in Spark-Ignited Turbulent Mixtures	7

2. Architecture

Processor Architecture and Application Performance in Modern Supercomputers	8
---	---

3. Grid Computing

UNICORE – Grid Computing for Production Systems	14
Grid Computing for Computational Astrophysics	18

4. Centers

LRZ	20
HLRS	22
NIC	24

5. Events

	26
--	----

inSiDE

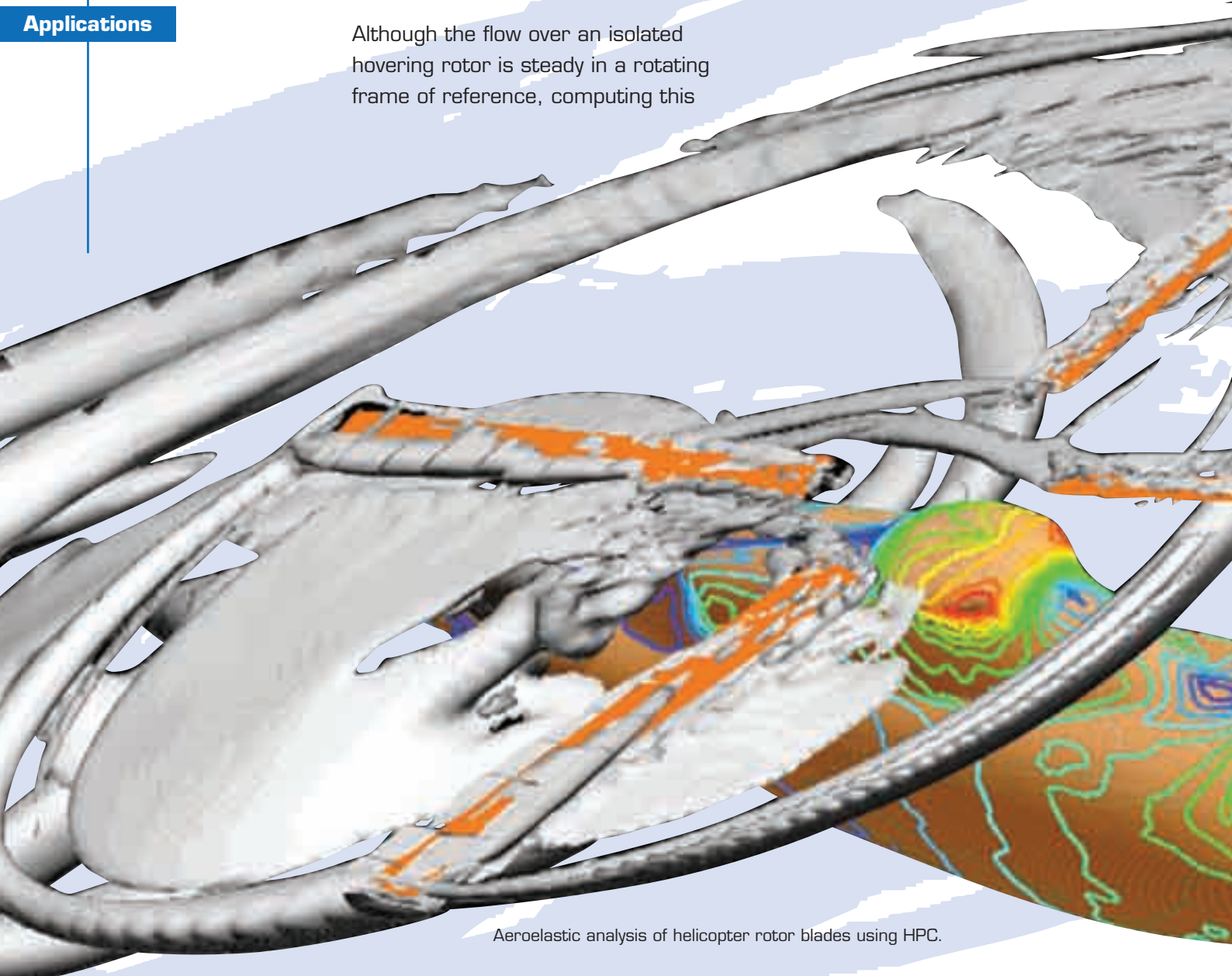
Numerical Simulation of Rotary W Aerodynamics, Aeroelasticity and

Rotorcraft flows rank among the most challenging applications of CFD in aviation engineering. While an attempt to numerically simulate the entire main rotor system of a helicopter calls for a multidisciplinary approach, i. e. primarily the coupling of flow and structure models, even an isolated aerodynamic analysis must cope with a wide spectrum of elementary and interactional flow problems and phenomena.

steady-state solution and thus predicting hover performance - a key issue in the design process of helicopters - is not at all trivial. The complexity of the hover flow field results primarily from strong vortical effects and the close proximity of the rotor blades and primary vortical structures which are convected away from the rotor disk at relatively low speeds, even at higher thrust settings. The tip vortex emitted by a lifting blade

Applications

Although the flow over an isolated hovering rotor is steady in a rotating frame of reference, computing this



Aeroelastic analysis of helicopter rotor blades using HPC.

ing Aeroacoustics

has a substantial impact on the effective local angles of attack in the outer region of the following blade. Consequently, the outcome of an aerodynamic analysis intended to provide quantitative information on hovering rotor loads and performance depends largely on the ability of the procedure to predict the rotor wake with sufficient accuracy which requires

both highly accurate numerical schemes and massive

computing power. In forward flight, compressibility effects can be dominant on the advancing side of the rotor while regions of highly complex separated flow may be present on the retreating side, the latter introducing a great deal of uncertainty into any first-principles ana-

lysis based on Reynolds-averaging. In contrast to more elementary approaches, modern Navier-Stokes codes can provide invaluable insight into local structures of the three-dimensional flow field and interactional phenomena, as needed for rotor design and verification purposes.

Over the past decade, a thoroughly validated Chimera structured grid finite volume code, based on the Reynolds-averaged Navier-Stokes or Euler equations, has been developed by the rotary wing workgroup at the Institute of Aerodynamics and Gasdynamics (IAG) of the University of Stuttgart. The comprehensive rotor analysis tool has been extensively optimized for use on the state-of-the-art HWW vector super computing platforms and has been applied in a number of research projects and also in the framework of joint-ventures with industry partners.

The incorporation of the interdependence of blade dynamics and flow field into the analysis, successfully implemented in 1998 and subject to continuous enhancement, has led to significant improvement with respect to prediction accuracy for all helicopter flight scenarios at a negligible increase in computing cost. The fluid structure interaction simulation capability has opened up the door towards highly accurate rotor performance verification analyses and the numerical investigation of e. g. adaptive rotor structures, vibration control and aeroacoustics, as planned within current and future helicopter rotor research activities at IAG

- Andree Altmikus
- Hubert Pomin

Institut für Aero- und Gasdynamik
Universität Stuttgart

Applications

A Vectorised Lagrangian Particle Model for the Numerical Simulation of Coal-Fired Furnaces

Applications

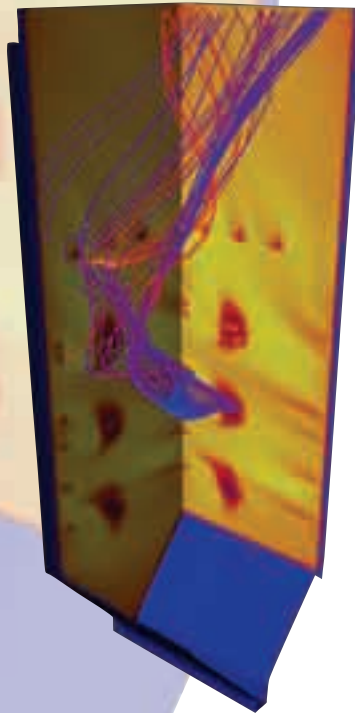
Since coal is still one of the major energy sources worldwide, the improvement of the efficiency of power stations with coal-fired boilers is an important task. Large improvements in computer technology and detailed physical models have enabled Computational Fluid Dynamics (CFD) to be a fast and economic tool for the optimisation of industrial furnaces. The powerful computer platforms now available for numerical simulations are generally

for the numerical calculation of three-dimensional, stationary and dynamic, turbulent reactive flows in pulverised coal-fired utility boilers. In submodels treating fluid flow, turbulence, homogeneous and heterogeneous combustion, and heat transfer, equations for calculating the conservation of mass, momentum, and energy are solved. The implemented models are optimised for vector and parallel computers to achieve high numerical efficiency.

A Lagrangian particle tracking model is applied to the simulation of the discrete phase in coal-fired furnaces. The interaction of particles with the gas phase leads to additional source terms in the Eulerian transport equations. The routines for the particle model are vectorised using loops over the number of all particles. The code shows a high performance and vectorisation rate, combined with good agreement with measurements.

AIOLOS is not only used for academic purpose, but has also shown a high efficiency in several practical projects, e.g. it is used for simulation of planned power plants and several high temperature processes by RECOM*-services.

A vectorised Lagrangian particle model for the numerical simulation of coal-fired furnaces.



• Frank Rückert

Institut für
Verfahrenstechnik und
Dampfkesselwesen
Universität Stuttgart

based on vector processors requiring a certain structure of the computer code.

The simulation program AIOLOS developed at the Institute of Process Engineering and Power Plant Technology (IVD), University of Stuttgart, is used

Massively of Flame K in Spark-I

Parapyr - Direct Numerical Simulation of Turbulent Reactive Flows

Energy conversion in numerous industrial power devices like automotive engines or gas turbines is still based on the combustion of fossil fuels. In most applications, the reactive system is turbulent and the reaction progress is influenced by turbulent fluctuations and mixing in the flow.

The understanding and modeling of turbulent combustion is thus vital in the conception and optimization of these systems in order to achieve higher performance levels while decreasing the amount of pollutant emission.

In the last several years, direct numerical simulations (DNS), i.e. the computation of time-dependent solutions of the compressible Navier-Stokes equations for reacting ideal gas mixtures, have been one of the most important tools to study fundamental issues in turbulent combustion.

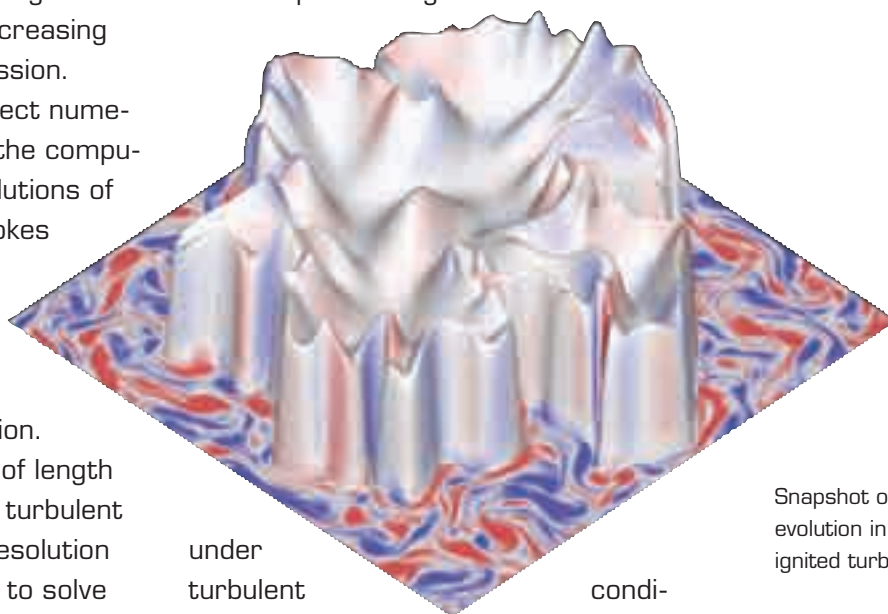
Due to the broad spectrum of length and time scales apparent in turbulent reactive flows, a very high resolution in space and time is needed to solve this system of equations. To be able to perform DNS of reactive flows including detailed chemical reaction mechanisms and a realistic description of molecular transport, it is necessary to make efficient use of HPC-systems.

A detailed chemistry DNS code has been developed which exhibits an excellent scaling behaviour on massively parallel systems. E.g., for a scaled

problem with a constant load per processor, a parallel efficiency of 85% has been achieved using all 512 PEs of the Cray T3E-900. Besides classical supercomputers, this code has also been successfully used in computational grids with PACX-MPI.

Induced ignition and the following evolution of premixed turbulent flames is a phenomenon of large practical importance as it occurs e.g. in Otto engine combustion.

These processes have been studied in a model configuration of an initially uniform premixed gas



Snapshot of flame kernel evolution in a spark-ignited turbulent mixture.

under turbulent conditions which is ignited by an energy source in a small region at the center of the computational domain. Figure 1 shows the spatial distribution of vorticity and that of the OH radical in one such turbulent flame kernel about 1 ms after the ignition

Applications

Parallel DNS Kernel Evolution Ignited Turbulent Mixtures

• Marc Lange

HLRS

Processor Architecture and Application Performance in Modern Supercomputers

Abstract

The architectures of two important processor series (IBM Power4 and NEC SX6) for contemporary supercomputers as well as one of their potential competitors (Intel Itanium series) are introduced. The performance characteristics of these systems are discussed using application programs, ranging from serial codes in theoretical physics and chemistry to a parallel CFD application.

Introduction

In the past five years the supercomputing community encountered several fundamental changes in the processor market. Two vendors (Compaq/DEC, HP) with high reputation in the High Performance Computing area have merged and announced the discontinuation of their well-known processor series (Alpha, PA-RISC), which still account for more than 130 installations in the latest TOP500 list [Top500]. On the other hand, IBM continued its RISC processor series with the launch of the Power4, including many novel features like dual processor cores. The domination of the IBM Power processor family is substantiated both by the TOP500 list, where IBM Power 3/4 based systems reflect about 33% of the aggregated peak performance (Rmax) and by the fact that IBM Power5 technology has been chosen for the ASCI Purple project.

However, a potential competitor for IBM has emerged with the advent of the Intel Itanium processor series. Intel and HP have developed a completely new design employing an architecture called

Explicitly Parallel Instruction Computing (EPIC), which is fundamentally different from the RISC paradigm. While the first incarnation, the Itanium1 (formerly known as Merced) has failed to become successful, the Itanium2 (McKinley) seems more promising because of significant improvements in bandwidths, overall balance and compiler technology. In addition, large cluster configurations (e.g. PNNL [PNO3]) and shared-memory systems (e.g. NEC TX7 [TxO3] and SGI Altix series [AIO3]) based on Itanium2 are now available.

A long standing discussion about the relevance of vector computers has recently been resumed (even in the U.S.) with the installation of the NEC SX6-based Earth Simulator in 2002, which is currently ranked number one in the TOP500. The substantial innovation of NEC SX6 is that it is the first vector CPU that fits on a single chip and at the same time provides the well-known features of vector computers: high memory bandwidth, memory latency hiding through vectorization and high single processor performance.

Besides these three main competitors there are other 64-bit designs, implementing the MIPS or SPARC architecture, which are right now not competitive with respect to peak performance and/or memory bandwidth.

Another major trend is the use of large clusters based on 32-bit AMD Athlon or Intel Pentium 4 processors, showing an increase from one to 61 TOP500 installations during the past five years. These systems provide excellent (peak) performance /price ratios and a working envi-

ronment which most users are familiar with. This convenience does not come for free, though: the 32-bit address limitation imposes severe limitations on the maximum data size and enforces massive parallelization for data intensive applications. Thus, at least part of the money saved in hardware and software has to be invested into software development.

From a user perspective, the processor design is of minor importance because the main concern is about application performance and usability (e.g. compiler, profiling and debugging tools). In this report we focus on the former. In order to cover a wide range of contemporary scientific problems we have chosen single processor applications from quantum physics and chemistry, which are commonly used for embarrassingly parallel parameter studies as well as an MPI-parallel Lattice-Boltzmann code from Computational Fluid Dynamics. Contrary to simple kernel loops like STREAM [Str03], these codes put high demands on the quality of the compiler, which is often a crucial component of an efficient system.

Architectures

To point out the basic differences between IBM Power4, Intel Itanium2 and NEC SX6 processors we briefly sketch the most important figures here. Since bandwidths and latencies usually determine application performance we have focused on those in Table 1.

The IBM Power 4 processor [P402] is a superscalar (8-way fetch, 5-way sustained complete) out-of-order RISC processor with a maximum frequency of 1.3 GHz (1.45 GHz upcoming) and two Multiply-Add units allowing for a peak of 5.2 GFlop/s. The basic difference to classical RISC systems is that two processors (cores) are placed on a single chip sharing the on-chip L2 and external L3 cache. Interestingly, the

Table 1
Memory hierarchy for IBM Power4 with 1.3 GHz, Intel Itanium2 with 1 GHz/3 MB L3 and NECSX6. On-chip caches are marked blue and external caches are marked green. On the IBM, L2 and L3 caches and their bandwidths are shared by two processors. The cache bandwidths are the aggregate read and write bandwidths. Please note that the Itanium benchmarks were performed on a 900 MHz processor which reduces the cache bandwidths by 10% . For sufficiently large loop lengths, memory latency in a vector computer can be hidden entirely.

		IBM Power 4	Intel Itanium 2	NEC SX6
# Floating Point Registers		32	128	8x256 (Vector)
L1 cache	Size	32 KB	16 KB	---
	BW	31.2 GB/s	32 GB/s	---
	Latency	4 cycle	1 cycle	---
L2 cache	Size	1.44 MB	256 KB	---
	BW	124 GB/s	32 GB/s	---
	Latency	14 cycles	5-6 cycles	---
L3 cache	Size (MB)	32 MB	3 MB	---
	BW	11.7 GB/s (Kr03)	32 GB/s	---
	Latency	340 cycles (RB01)	12-13 cycles	---
Memory	BW	max 6.9 GB/s	6.4 GB/s	32 GB/s
		read & write	read or write	read or write
	Latency	~ 200 ns	~ 200 ns	see caption

L2-L1 bandwidth (shared by two processors) is higher than the aggregated L1-processor bandwidth for the dual-core system, which may improve non-contiguous access to data in L2. Please note that the large L3 cache size has to be paid for with very long latencies (~100 processor cycles). Depending on the actual memory configuration, the Power 4 can achieve very high memory bandwidth; however, the rather large L3 cache line size of 512 Bytes can result in a significant degradation of the effective bandwidth for non-contiguous access patterns. For the benchmarks presented here, we have used an IBM p690 node with 32 CPUs. A theoretical memory bandwidth of 6.9 GB/s is available for two dual-cores. Four dual-cores can share their L3 caches (thus a maximum of 128 MB is available for one processor) with different latencies for local and remote access.

The Itanium2 processor [ItO2] is currently available with clock frequencies of 900 MHz and 1 GHz, delivering a peak performance of 3.6 GFlop/s and 4.0 GFlop/s respectively, if both Multiply-Add units can be used. Floating-point data items bypass the L1 cache and can be accessed with high bandwidth and low latency in the on-chip L2 and L3 caches. An important difference compared to Power 4 is the large register set which allows for large unrolling factors and prevents register spills. The fundamental distinction from classical RISC systems, however, is the EPIC concept, implemented as follows: The CPU loads instructions in bundles of three. Only a limited number of combinations among memory, integer and floating point instructions is allowed per bundle, and the compiler has to take care of that. More importantly, it also specifies groups of independent instructions that the proces-

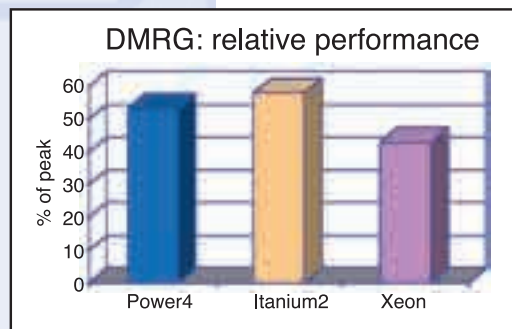
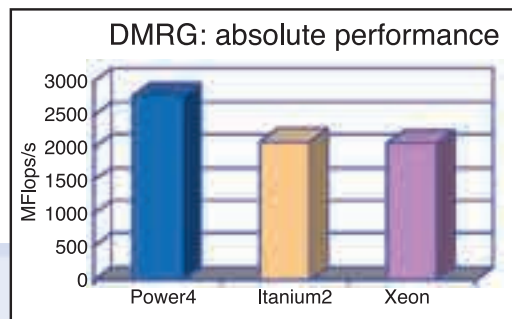
sor may execute in parallel. Groups and bundles are two concepts that are, in a sense, orthogonal to each other, i.e. Itanium can issue two bundles per cycle (6-way superscalar), but a group can span any number of machine instructions. Since the instruction stream now already contains information about independent instructions, no out-of-order execution support is necessary on the processor. Of course this concept demands high quality compilers to identify instruction level parallelism in the code. As a benchmark system we have chosen an HP zx6000 workstation under Redhat Linux Advanced Workstation 2.1 with two 900 MHz (1.5 MB L3 cache) processors and total CPU to memory bandwidth of 6.4 GB/s. Unless otherwise stated, we use the Intel IA64 compilers Version 7.

From a programmer's view, the efficient use of the NEC SX6 [SxO3] still requires vectorization techniques that have been known for a long time. Technically, however, it represents a fundamental technological breakthrough since it is the first vector processor on a single chip. Running at 500 MHz and using 8-track vector pipelines, the NEC SX6 processor achieves a peak performance of 8 GFlop/s and can sustain 32 GB/s memory bandwidth both for read or write. For our benchmarks we used an 8-way NEC SX6 shared-memory node with an overall bandwidth of 256 GB/s, i.e. the shared memory can saturate the aggregated single processor bandwidths.

For reference, we also present performance measurements on a 2.4 GHz Intel Xeon DP processor [XeO2] with 3.2 GB/s memory bandwidth using RDRAM.

Application Performance

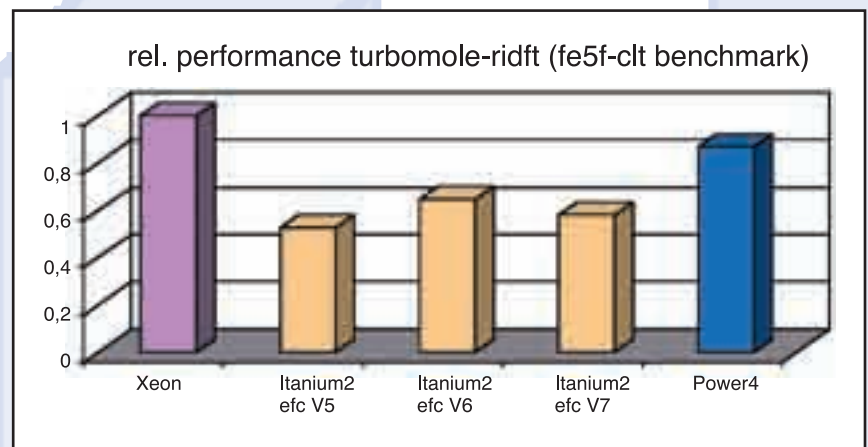
A sequential C++ program package developed by White and Jeckelmann [Jec98] has been chosen as a starting point for the discussion of application performance. This package uses a Density Matrix Renormalization Group (DMRG) algorithm [Whi92] to calculate ground state properties of strongly correlated quantum lattice systems. The DMRG algorithms have been successfully established during the past decade in quantum physics and quantum chemistry and complement or sometimes replace traditional methods



like Exact Diagonalization or Quantum Monte Carlo. Like most of the DMRG packages the program under consideration uses a sophisticated data house-keeping structure and has matrix-matrix multiplication as its dominating kernel routine. Although the matrices are dense and DGEMM is used, it should be emphasized that the matrices are frequently rather small and non-square. Other parameters relevant for performance are of course quality of the DGEMM implementation (depending on the proprietary libraries) and the C++

compiler. For a characteristic problem with about 350 MBytes memory requirement we present the absolute performance numbers as well as the relative performance (compared to the peak performance of each processor). As should be expected, all systems achieve a substantial fraction of their peak performance. When compared to Itanium2, the IBM Power4 cannot make full use of its 44% higher clock speed, resulting in a lower relative performance. Nonetheless, all systems including the Xeon are in a good shape to perform well with this complex C++ program and so there is no need to run that kind of application on a vector computer.

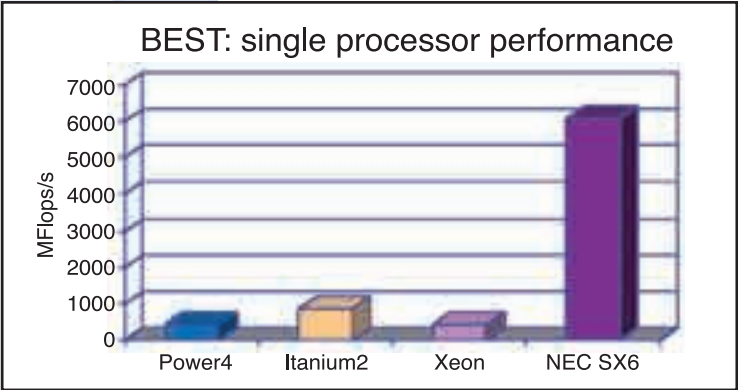
An important application from quantum chemistry is TURBOMOLE [Ahl02], which has been ported to a large spectrum of architectures except vector machines (which are unsuitable anyway because of the dominance of short



loops in the code). It is known that IA32 systems provide outstanding performance for this cache-based application [PBO2], so we have chosen a Xeon 2.4 GHz CPU as the baseline for the presentation of performance numbers (120 MBytes memory requirement for this benchmark).

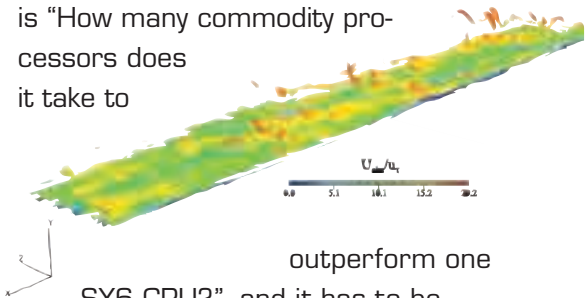
The figures show impressively that even Power4 cannot keep up with the Xeon

in this discipline. In case of Itanium2 we have tested all major compiler versions available to date, and interestingly version 6 does best, although version 7 is usually to be preferred due to increased stability. Both 64 bit architectures, while delivering acceptable performance, cannot rival IA32 with respect to performance/price ratio. In the past, Computational Fluid Dynamics (CFD) applications have been the major customers for vector computers. As an example for a highly vectorizable parallel CFD application we have chosen BEST (Boltzmann Equation Solver Tool) developed at the Institute for Fluid Mechanics (Prof. Durst, University of Erlangen, Germany). BEST is based on a Lattice Boltzmann Method [Qia92], which currently evokes increased interest in the CFD community because of its proven efficiency for fluid flows in highly complex geometries (e.g. porous media and chemical reactors) [Ber99]. Profiling shows that most of



the computing time is spent in a single loop comprising a large number of floating point and load/store instructions. Optimized implementations of this loop are available for vector and cache based processors. As a benchmark case we calculate a wall-bounded turbulent flow between two parallel plates (plain channel flow). MPI parallelization is done by cutting the domain in equal slices, using ghost cell layers. This is a benchmark where the

communication-to-computation ratio is adjustable in a very controlled way. The “yardstick” for this application is set by the only modern vector supercomputer left on the market to date, the NEC SX6. The central question is “How many commodity processors does it take to



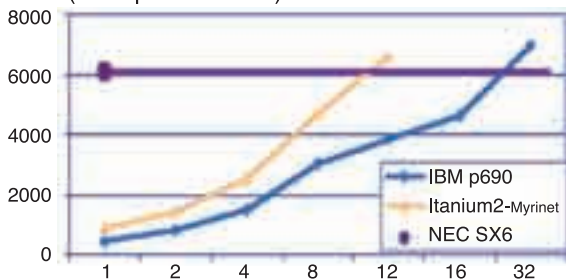
answered with respect to the same problem parameters. For the standard test case (1 GB memory requirement), five Itanium2 workstations (2 CPUs each) or 24 CPUs of an IBM p690 can do the job.

This result shows that Itanium2, due to its unique combination of large memory bandwidth and register set, outperforms other commodity architectures by far. 25% of peak performance is an impressive result for a non-vector CPU, even if it cannot match the 75% ratio of the SX6. Consequently, the Itanium2 poses an interesting alternative for this CFD application, showing a competitive performance/price ratio. The disappointing performance of the p690 is contrary to expectations, yet it has a much larger aggregate memory bandwidth as an Itanium2 workstation cluster of similar size. The reason for this failure is still obscure and being investigated.

Summary

In summary, one must arrive at the conclusion that Itanium2 has a new yet competitive architecture that rivals long-running champions in the RISC and even the vector processor market. Nevertheless, IBM’s Power4 is certainly its immediate opponent and can show

BEST: parallel performance
(fixed problem size)



off with proven large-scale SMP and SMP cluster technology, a segment which Itanium2 still has to enter. Right now, both architectures should thus be regarded more like complementing than competing. With the arrival of bigger shared-memory systems (SGI Altix, NEC TX-7) this situation will change shortly though, and we must prepare for a close race in the price/performance dimension.

Vector processors still achieve substantially higher performance than microprocessors in most applications and their programming model is rather simple when compared to cache optimization. The question of performance/price scaling ratio will be decisive for the success of this technology.

Acknowledgements

We thank for providing valuable technical insight: H. Cornelius (Intel), F. Krämer (IBM), T. Schönmeyer (NEC), H. Strauss (HP), and R. Wolff (SGI). Helpfull discussions with U. Küster, H. Huber, R. Bader are gratefully acknowledged. We are indebted to B. Hess for preparing the TURBOMOLE benchmarks. Computational resources at HLRS in Stuttgart, RZG in Garching, DKRZ in Hamburg and RRZE in Erlangen were used. This work was partially supported by the Bavarian Competence Network for High Performance Computing (KONWIHR).

References

[Top 500]

www.top500.org/

[Str03]

www.cs.virginia.edu/stream/

[PN03]

www.emsl.pnl.gov:2080/capabs/mscf/?/capabs/mscf/hardware/config_opus.html

[P402]

www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/power4.html

[Sx03]

www.sw.nec.co.jp/hpc/sx-e/index.html

[Xe02]

developer.intel.com/design/xeon/

[It02]

developer.intel.com/design/itanium2/index.htm

[Tx03]

www.hpce.nec.com/uploads/media/TX7.pdf

[Al03]

www.sgi.com/servers/altix/

[Jec98]

Jeckelmann E., White S.R.: Density-Matrix Renormalization Group Study of the Polaron Problem in the Holstein Model. *Phys. Rev. B*, 57, 6376-6385 (1998)

[Whi92]

White S.R.: Density-Matrix Formulation for Quantum Renormalization Groups. *Phys. Rev. B*, 69, 2863-2866 (1992)

[Ahl02]

Ahlrichs R., TURBOMOLE. Quantum Chemistry Group, University of Karlsruhe, Germany.

[Qia92]

Qian Y.H., d'Humieres D., Lallemand P.: Lattice.

[BGK]

Models for Navier-Stokes Equation. *Europhys. Lett.*, 17, 479-484 (1992)

[Ber99]

Bernsdorf J., Durst F., Schäfer M.: Comparison of Cellular Automata and Finite Volume Techniques for Simulation of Incompressible Flows in Complex Geometries. *Int. J. Numer. Meth. Fluids*, 22, 251-264 (1999)

[RB01]

Behling S., Bell R., Farrell P., Holthoff H., O'Connell F., Weir W.: The Power4 Processor Introduction and Tuning Guide. IBM (2001), www.ibm.com/redbooks/

[Kr03]

Krämer F., IBM. Private Communication.

- Georg Hager¹
- Frank Brechtefeld¹
- Peter Lammers²
- Gerhard Wellein¹

¹ HPC Services, Regionales Rechenzentrum Erlangen, Germany

² Lehrstuhl für Strömungsmechanik (LSTM), University of Erlangen, Germany

UNICORE - Grid Computing for Production Systems

Grid Computing

Introduction

Long before the term Grid Computing was coined by Foster and Kesselman [1] the development of UNICORE, a system for Uniform Interface to Computing Resources was started. The goal was to provide users of the German supercomputer centers with a seamless, secure, and intuitive access to the heterogeneous computing resources at the centers consistent with the recommendations of the German Science Council (see [2]-[4]). A first prototype was developed in project UNICORE¹ to demonstrate the concept [5]. The current production version was created in a follow-on project UNICORE Plus² which was completed in 2002. The UNICORE software deployed by the project partners, it is used as the basis for European projects, and is marketed and supported by Pallas GmbH. UNICORE is available as open source downloadable from [5].

UNICORE Functions

UNICORE provides the users with a rich set of functions to create and manage complex batch jobs that can be executed on different systems at different sites. The UNICORE software takes care of the necessary mapping of user request to system specific actions. Transfer of data between systems and sites is performed automatically by UNICORE. UNICORE ensures that only properly authenticated and authorized users may access resources. Details of key functions are given below. The UNICORE User Guide can be downloaded from [5].

Job Creation and Submission:

A graphical interface assists the user in creating complex, interdependent jobs that can be executed on any UNICORE site without changes to the job definitions. A UNICORE job, more precisely a job group, may recursively contain other job groups and/or tasks. A job group is submitted to a UNICORE site which the user selects prior to submission. The UNICORE client creates an abstract representation of the job group, the Abstract Job Object (AJO). The AJO is stored at the user's workstation as a serialized Java object and/or in XML format. Tasks contained in a job group are incarnated into a batch job to be executed on a system at the site or into an action, like a file transfer to a storage space. Child jobs groups are transferred to the appropriate site to be incarnated and executed there. The user may specify temporal dependencies between the entities contained in a job group. UNICORE ensures that a successor is executed only if all predecessors have completed successfully and all necessary data sets are available at the target system.

Job Management:

The user has full control over jobs and data. A color code presented along with the job icon shows the overall status of a job: green, red, yellow, blue, magenta to indicate successful completion, failure, in execution, queued, or waiting for completion of a predecessor, respectively. The status is available at each level of recursion down to the individual task. It may be refreshed by clicking on the icon or by an automatic timer-driven status update. In addition,

Deployment of UNICORE in Germany.



detailed log information is available to analyze error conditions. The job output that is written to stdout and stderr by the execution systems can be reviewed or transferred to the client workstation. Jobs may be terminated and removed from the UNICORE grid by the user.

Data Management:

UNICORE jobs contain tasks that can be executed at different computing centers. Output created by one task may be used by any of its successors. A temporary UNICORE space, called Uspace for short, is created for each job group. During job creation the user specifies

- which data sets are to be imported into the Uspace from the client workstation or any file system or data archive at the UNICORE site to which the user has access,
- which data set are to be exported from the Uspace to retain them permanently, and
- which data sets are to be transferred to a different Uspace.

At run time UNICORE performs the necessary data movement without user intervention.

Application Support:

The functions described above provide an effective tool to use resources of different computing centers both for capacity or capability. Many scientists and engineers use application packages. UNICORE provides a two-fold support for these users: For applications without a graphical user interface can be provided using the plug-in technique of UNICORE. The CPMD (Car-Parrinello Molecular Dynamics) plug-in and wizard demonstrates this (for details see [9]). Wrappers are developed for applications with existing graphical interfaces, such as Fluent, STAR-CD, or MSC Nastran, to combine their known look-

and-feel with the system independence and security of UNICORE.

Flow Control:

In addition to the basic job dependencies UNICORE supports conditional and repetitive execution of job groups or tasks. This allows to run of computational experiments which can be repeated a fixed number of times or until a given condition is reached and support of applications, which require special action in case errors occur.

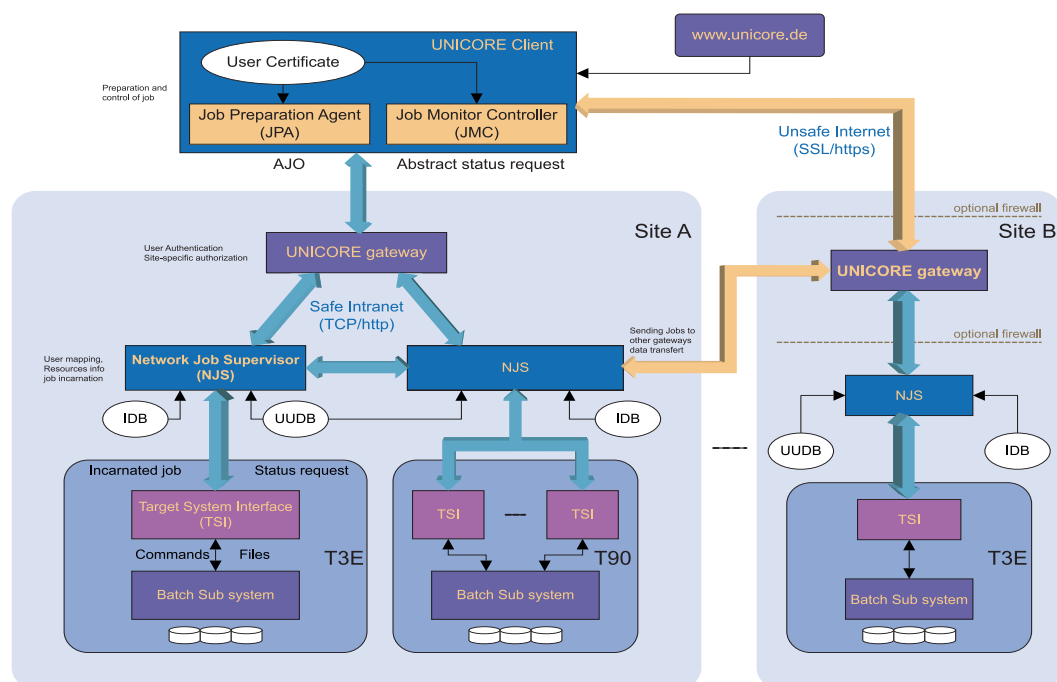
Single Sign-on:

UNICORE provides a single sign-on through X.509V3 certificates. The certificate can be mapped to a local account at each UNICORE site. The account name and the UNIX uid/gid may be different at each site, due to existing naming conventions. In addition, the site retains full control over the acceptance of users based on the identity of the individual - the distinguished name - or other information that might be contained in the certificate. Each site can restrict and limit accessible resources at each target system, thus retaining the ultimate control. UNICORE can handle multiple user certificates, i.e. it permits a client to be part of multiple, disjoint Grids.

Support for Legacy Jobs:

UNICORE supports traditional batch processing by allowing users to include their old job scripts as part of a UNICORE job. This approach does not guarantee seamlessness but it helps users immediately in the following scenario: a user submits a job to a supercomputer, periodically check for its completion, transfer results to a different system, using ftp, and submit a successor job on this system. A simple job group can automate these steps without changing the existing job scripts.

The UNICORE architecture.



The UNICORE Architecture:

UNICORE implements a three-tier architecture as depicted in figure 1. The UNICORE client is a Java application that executes on the user's workstation. It supports the creation, manipulation, and control of complex jobs, which may involve multiple systems at one or more UNICORE sites. The jobs and actions as defined the user are represented as Abstract Job Objects, effectively Java classes, which are serialized and signed when transferred between the components of UNICORE.

The server level of UNICORE consists of a Gateway, the secure entry point into a UNICORE site, which authenticates requests from UNICORE clients and forwards them to a Network Job Supervisor (NJS) for further processing. The NJS maps the abstract request, as represented by the AJO, into concrete jobs or actions which are performed by the target system, if it is part of the local UNICORE site. This process is called incarnation. Sub-jobs that have to be run at a different site are transferred to this site's gateway for subsequent processing by the peer NJS. Additional

functions of NJS are: synchronization of jobs to honor the dependencies specified by the user, automatic transfer of data between UNICORE sites as required for job execution, collection of results from jobs, especially stdout and stderr, import and export of data between the UNICORE space and target system, and client workstation.

The third tier of the architecture is the target host which executes the incarnated user jobs or system functions. A small daemon, called the Target System Interface (TSI) resides on the host to interface with the local batch system on behalf of the user. A stateless protocol is used to communicate between NJS and TSI. Multiple TSIs may be started on a host to increase performance.

UNICORE Deployment

UNICORE is deployed at the sites of the project partners. UNICORE support systems by all vendors installed at the sites, like CRAY, IBM, Hitachi, NEC, as well as Linux Clusters systems and interfaces to the different operating systems and batch queuing systems. It is offered to users of the national

supercomputer centers at LRZ, HLRS and FZ Jülich to access their production systems where it is supported by Pallas.

Related Projects

The successful work of the project produced not only a production ready solution for supercomputer centers in Germany. UNICORE was selected as the basis for the first European Grid project EUROGRID³ (see [6]). It demonstrates the use of the UNICORE Grid software in four selected scientific and industrial communities, addresses their specific requirements for a future European Grid middleware based on UNICORE. Technical developments include interactive access and a resource broker.

The Grid developments world wide and the activities of the Global Grid Forum led EUROGRID partners to propose project GRIP⁴ (Grid Interoperability Project) to make resources controlled by Globus [7] software available to UNICORE users and to provide Globus users with a powerful graphical frontend. The interoperability between UNICORE and Globus has been demonstrated in 2002.

The results of the European projects will be included in future production versions of UNICORE at the end of the respective projects.

Future Work

The present direction in Grid computing as promoted by the Global Grid Forum [8] with strong support from industry aims towards integrating Grid technology and Web Services in an Open Grid Service Architecture (OGSA). In parallel with the development work in project GRIP it has been shown that UNICORE is largely compatible with OGSA. In the second year a first implementation of UNICORE will be done to allow inter-operation with selected Grid Services.

During the Global Grid Forum (GGF7) in Tokyo Dr. Makoto Furunishi of MEXT, the Japanese Ministry of Education, Culture, Sports, Science and Technology, announced that UNICORE has been selected as the Grid middleware for the new National Research Grid Initiative (NAREGI) led by Dr. Kenichi Miura, Fujitsu Laboratories. This initiative plans to build an infrastructure making over a hundred Teraflops available for scientific applications. It will be funded with 100 Mio \$ over five years.

The dedicated work of early Grid research and development in Germany initially supported by BMBF is starting to pay off.

References

1. **Foster I. and Kesselmann C., Ed.**

The Grid:

Blueprint for a New Computing Infrastructure. Morgan Kaufman Publishers, 1998.

2. **Wissenschaftsrat:**

Empfehlung zur Versorgung von Wissenschaft und Forschung mit Höchstleistungsrechenkapazität, Wissenschaftsrat, Drs. 2104/95, 7.7.1995.

3. **Wissenschaftsrat:**

Empfehlung zur künftigen Nutzung von Höchstleistungsrechnern, Drs. 4558/00 www.wissenschaftsrat.de/texte/4558-00.pdf.

4. **Hoßfeld F.; Nagel W. E.:**

Verbund der Supercomputer-Zentren in Deutschland – eine Machbarkeitsanalyse Jülich, 1997, BMBF-Förderkennzeichen O1 IR 602/9 www.fz-juelich.de/zam/pt.s/mannheim/vesuz_1999.ps.

5. **Erwin D., Ed.**

UNICORE:

Uniformes Interface für Computing Ressourcen (Final report – in German) www.unicore.org, 2000.

6. **www.eurogrid.org**

7. **www.globus.org**

8. **www.gridforum.org**

9. **Huber V.:**

V. Supporting Car-Parrinello Molecular Dynamics Application with UNICORE Proceedings ICCS 2001, San Francisco, CA, pp 580-567.

¹ UNICORE was supported in part by BMBF grant O1 IR 703

² UNICORE Plus is funded in part by BMBF grant O1 IR 001

³ EUROGRID is partially funded by the European Commission under grant IST-1999-20247

⁴ GRIP is funded in part by EC grant IST-2001-32257

• Dietmar Erwin

Forschungszentrum
Jülich GmbH

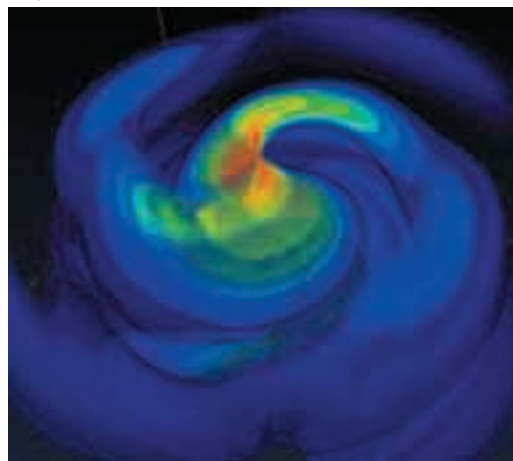
UNICORE Forum e.V.
www.unicore.org

Grid Computing for Computational Astrophysics

Grid Computing

Enabling Bigger Science

Black holes, whose behavior is described by Einstein's General Theory of Relativity, are not just the stuff of science-fiction stories. Astrophysicists predict that black holes, formed from the cataclysmic collapse of stars many times more massive than our own sun, will actually be able to be observed and studied from Earth. To do this, however, scientists require giant laser interferometric devices that can detect the gravitational waves black holes emit, in much the same way that telescopes already observe optical, x-ray and gamma-ray emissions from black holes.



Gravitational radiation emitted from the collision of two black holes.

Copyright ZIB/AEI.

Gravitational waves, predicted by Einstein nearly a century ago, are believed to travel through the universe at the speed of light, barely interacting with the matter through which they pass. They are extremely hard to detect, so difficult that they have never been seen directly. Even with the current generation of interferometers, some of which are over 4 kilometers long and offer our first real hope for observing waves, their detection and interpretation will rely on accurate results from numerical simulations of the astrophysical

processes that generate them. Such simulations are being carried out by an international group of scientists at the Max-Planck-Institute for Gravitational Physics in Golm (Albert-Einstein-Institute, AEI), using supercomputers like the Hitachi SR-8000 at the Leibniz-Rechenzentrum (LRZ) in Munich. For these supercomputers scientists have developed ground-breaking techniques for performing simulations of the inspi-ralling collision of two black holes. The scientists are aided in their physics by Cactus, a sophisticated framework for parallel computations, which was specifically designed to enable such large scale, computationally intensive, collaborative simulations.

The Cactus Computational Toolkit

The Cactus Computational Toolkit (www.cactuscode.org), developed at AEI, embodies a new paradigm for the development and reuse of numerical software in a collaborative, portable environment. As a freely available, open source toolkit, Cactus lets one extend traditional single-processor code (using common languages like C, C++, Fortran 77, and Fortran 90) into full-blown parallel applications that can run on virtually any supercomputer. Cactus also provides access to computational tools, such as advanced numerical techniques, parallel I/O, remote visualization, and remote steering.

The name "Cactus" comes from the initial design principle of a module set (the thorns) that can be plugged easily into a basic kernel (the flesh). The flesh, which controls the interaction of thorns,

has evolved to a metacode with its own language to enable dynamic assembly of many different application codes from a set of thorns. For example, while a physicist may develop a new formulation of Einstein's equations, a numerical analyst may have a more efficient algorithm for solving a required elliptic equation, a computer scientist may develop a more efficient parallel I/O layer. Cactus supports such an inter-disciplinary and collaborative work style.

Cactus and Globus on the Hitachi SR8000 at LRZ

The Globus Project is developing the fundamental Grid Computing middleware needed for building a functioning grid and supporting grid applications (www.globus.org). Together with the Cactus Code, they provide a powerful technology to enable the everyday work of computational physicists at AEI: A Cactus simulation running on LRZ's Hitachi SR8000 supercomputer may produce terabytes of output data. Ordinary postprocessing and data analysis tools, typically cannot deal with this amount of data. Scientists at AEI required new techniques to effectively manage very large remote datasets and provide efficient access to them. This is one of the addressed goals in GriKSL (www.griksl.org) – a DFN-funded research project of AEI and Konrad-Zuse-Institute in Berlin.

Based on the GridFtp service from the Globus toolkit and the standard Hierarchical Data Format I/O library (HDF5), GriKSL has developed a technique to access large datasets on remote servers. Scientists can select individual timesteps or zoom into the interesting regions of single datasets and visualize these special regions on a local visualization client. Another approach to exploit the Grid for enabling better science is taken

by EU-funded research GridLab (www.gridlab.org), of which AEI is a partner. GridLab is developing an easy-to-use, flexible, generic and modular Grid Application Toolkit (GAT), enabling today's applications to make innovative use of global computing resources. One aim of GridLab is to enable Dynamic Grid Computing: By making applications such as Cactus self-aware of the Grid, they can adapt to changes in a dynamic Grid environment, e.g., monitor and adapt to the current network status, discover and use new computing resources or migrate to more suitable machines. As a first step, a Globus-based European-wide testbed has been established, which includes the LRZ. The GridLab testbed participated in prize-winning demonstrations at Supercomputing 2002 with the Global Grid Testbed Collaboration, which showed various Grid-enabled applications making innovative use of the distributed resources. One of these demonstrations showed how small Cactus black hole simulations could be intelligently task farmed across the testbed for a parameter survey, the results of which automatically corrected the numerical parameters of a production-size simulation at a further site.

In the future, Cactus users will be able to run their simulations, on a regular basis, distributed across the Grid. For such simulations the Cactus communication infrastructure uses MPICH-G2 (www3.niu.edu/mpi/), a Globus-based implementation of the MPI standard. It couples multiple machines, potentially with different architectures, to span a single global communication domain for MPI applications. Thus the individual computational power of multiple supercomputers at different computing centers can be combined to build a uniquely large meta-computing resource.

Grid Computing

- Thomas Damlitsch
- Gerd Lanfermann
- Thomas Radke
- Gabrielle Allen
- Ed Seidel

Max-Planck-Institut für
Gravitationsphysik,
Golm



Leibniz Computing Center of the Bavarian Academy of Sciences (Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, LRZ) in Munich provides national, regional and local HPC services. Each platforms described below is documented on the LRZ WWW server; please choose the appropriate link from www.lrz.de/services/compute

Contact:

Leibniz-Rechenzentrum
High-Performance Systems Department

Dr. Horst-Dieter Steinhöfer
Barer Straße 21
D-80333 München
Phone +49 89 28 92 87 79
steinhoefer@lrz.de
www.lrz-muenchen.de

Centers

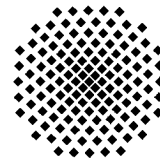
View of the Hitachi SR8000-F1 at LRZ.



Compute servers currently operated by LRZ are

System	Size	Peak Performance Performance from Memory* (GFlop/s)	Purpose	User Community
Hitachi SR8000-F1	168 8-way SMP nodes 1376 GByte memory	2016 247*	Capability computing	German universities and research institutes
Fujitsu/Siemens VPP700	52 vector processors 144 GByte memory	114 38*	Capability and capacity computing	Bavarian universities
Linux Cluster Intel IA32	48 nodes 88 IA32 processors 78 GByte memory	135 4*	Capacity computing	Munich universities
Linux Cluster Intel IA32 Intel IA64 (2Q2003)	90 IA32 single CPU 16 IA64 4-way CPU 218 GByte memory	693 20*	Capability and capacity computing	Munich universities
IBM pSeries 690 hpc	1 SMP node 8 processors POWER 4 32 GBytes memory	42 2*	Capacity computing	Munich universities

Centers



Based on a long tradition in supercomputing at Stuttgart University, HLRS was founded in 1995 as a federal center for High-Performance Computing. HLRS serves researchers at universities and research laboratories in Germany and their external and industrial partners with high-end computing power for engineering and scientific applications.

Contact:

HLRS
High-Performance Computing Center
Stuttgart

Prof. Dr. Michael M. Resch
Allmandring 30
D-70500 Stuttgart
Phone +49 71 16 85 25 04
resch@hlrs.de
www.hlrs.de

Centers

View of the NEC SX-5/32 M2e at HLRS.



Compute servers currently operated by HLRS are

System	Size	Peak Performance	Purpose	User Community
NEC SX-5/32 M2e	2 16-way nodes 80 GByte memory	128	Capability computing	German universities, research institutes, and industry
Cray T3e-900/512	512 nodes 64 GByte memory	450	Capability computing	German universities, research institutes, and industry
Hitachi SR8000	16 8-way nodes 128 GByte memory	128	Capability and capacity computing	German universities, research institutes, and industry
NEC Azusa	16 IA64 32 GByte memory	51,2	Capacity computing	Stuttgart University
IA64 Cluster	8 2-way nodes 32 GBytes memory	57,6	Capacity computing	Stuttgart University
IA32 Cluster	24 2-way nodes 48 GBytes memory	230,4	Capacity computing	Stuttgart University



Centers

The John von Neumann Institute for Computing (NIC) is a joint foundation of Forschungszentrum Jülich and Deutsches Elektronen-Synchrotron DESY to support supercomputer-aided scientific research and development in Germany. NIC takes over the functions and tasks of the High Performance Computer Centre (HLRZ) established in 1987 and continues this centre's successful work in the field of supercomputing and its applications.

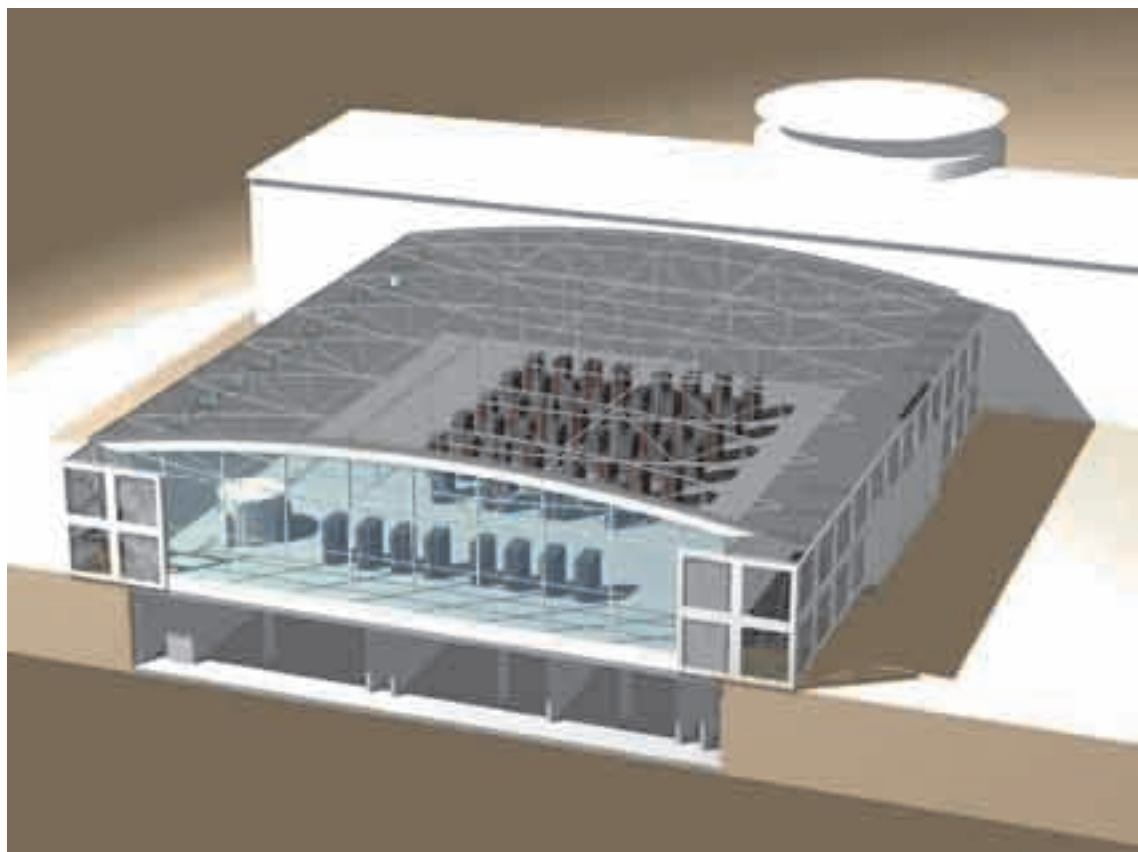
Nationwide provision of supercomputer capacity for projects in science, research and industry in the fields of modelling and computer simulation including their methods. The supercomputers with the required information technology infrastructure (software, data storage, networks) are operated

by the Central Institute for Applied Mathematics (ZAM) in Jülich and by the Centre for Parallel Computing of DESY at Zeuthen.

Supercomputer-oriented research and development in selected fields of physics and other natural sciences, especially in elementary-particle physics, by research groups of competence in supercomputing applications. At present, research groups exist for high energy physics and complex systems; another research group in the field of "Bioinformatics" is under consideration.

Education and training in the fields of supercomputing by symposia, workshops, summer schools, seminars and courses.

View of the new machine room in Jülich, being built for the IBM supercomputer.



The following supercomputers are available in Jülich for refereed research projects of the communities mentioned below. A more detailed description of the supercomputers can be found on the web server of the Research Centre Jülich:

System	Size	Peak Performance (GFlop/s)	Purpose	User Community
IBM pSeries 690 Cluster 1600 (4Q2002 - 2Q2003)	2 SMP nodes 64 processors POWER4 128 GBytes memory	333	Test system	Selected users
IBM pSeries 690 Cluster 1600 (3Q2003 - 1Q2004)	6 SMP nodes 1 processors POWER4+ 384 GBytes memory	1300	Capability computing	German universities, research institutes, and industry
IBM pSeries 690 Cluster 1600 (from 1Q2004)	35 SMP nodes 1120 CPUs POWER4+ 2240 GBytes memory	7600	Capability computing	German universities, research institutes, and industry
CRAY T3E-1200	512 nodes 262 GBytes memory	614	Capability computing	German universities, research institutes, and industry
CRAY T3E-600	512 nodes 64 GBytes memory	300	Capability computing	German universities, research institutes, and industry
CRAY SV1ex	16 CPUs 32 GBytes memory	32	Capability computing	German universities, research institutes, and industry

Centers

Contact:

John von Neumann
Institute for Computing (NIC)
Central Institute
for Applied Mathematics (ZAM)

Dr. Burkhard Mertens
D-52425 Jülich
Phone +49 24 61 61 64 02
b.mertens@fz-juelich.de
www.fz-juelich.de/zam/
CompServ/services/sco.html

High-Performance Computing Courses

LRZ

www.lrz.de

Parallel Programming on High-Performance Computers

Date and Time:

July 23-25, 2003, 9:00 am to 5:30 pm
(until 4:00 pm on Friday)

Location:

LRZ, lecture room 3rd floor, hands-on sessions on 2nd and 3rd day in S1535 (1st floor)

Contents:

Introduction to parallel programming of the Hitachi SR8000-F1 Supercomputer at LRZ

Programming Intel-based Linux-Clusters

Date and Time:

July 25, 2003, 9:00 am to 4:00 pm

Location:

LRZ, lecture room PEP

Contents:

Programming the IA-32 and IA-64 architecture

- basic architectural concepts
- optimization and tuning
- big/little endian
- available compilers
- VTune and other tools

Note:

This course coincides with the SR8000-specific day 3 of the parallel programming course.

TotalView: A Universal Debugger

Date and Time:

May 5, 2003, 9:30 am to 12:30 pm

Location:

LRZ, lecture room 1st floor (S1535)

Contents:

Introduction to the usage of the TotalView debugger. This tool is available on all high performance computers installed at LRZ. Hands-on sessions will give you additional usage experience.

Efficient Programming in Fortran, C and C++

Date and Time:

July 17, 2003, 10:30 am to 4:30 pm
(preliminary)

Location:

LRZ, lecture room 3rd floor

Contents:

- general problems of efficient programming
- cache-optimization on RISC-based systems
- C++ and object-oriented techniques

C++ for C-Programmers

Date and Time:

July 14-16, 2003, 2:00 pm to 6:00 pm

Location:

LRZ, lecture room PEP

Contents:

Prerequisites for this course are an average knowledge of C and a firm grasp of UNIX. It is primarily aimed at programmers in the field of technical and scientific computing.

Workshop on Application of High Performance Computing to Chemistry and Biological Sciences

Date:

End of June or beginning of July, 2003.

Location:

LRZ, lecture room 3rd floor

Contents:

Researchers from the Munich Universities are given the opportunity to give talks on application and usefulness of software packages used on HPC platforms. The aim is to propagate knowledge especially with respect to a reasonable choice of computing platform for a given problem from the chemical or biological sciences. Furthermore, LRZ considers inviting representatives from the software industry, as well as informing about recent developments in hardware and software availability.

Advanced Fortran 90 Programming

Date:

3rd week of July, 2003

Location:

Erlangen, RRZE

Contents:

This is not an introduction to Fortran 90, but to be considered as upgrade training for experienced Fortran 90 programmers. Depending on the interests of the course attendants the following points will be discussed:

and Tutorials

- object oriented programming: what is possible, what is not
- discussion of performance issues with array constructs and complex data types
- recommendations on program design, especially module structure; handling of module dependencies in a program build system.

IA-64 Workshop

Date:

Autumn 2003

Location:

Erlangen, RRZE

Contents:

Talks on experiences with the usage and performance of Itanium2 and/or Itanium3 processors.

For more information and registration see:
www.lrz-muenchen.de/services/compute/hlr/ kurse/

HLRS

www.hlrs.de

Parallel Programming Workshop, MPI and OpenMP for Beginners and Advanced Topics in Parallel Programming

Date:

September 15-19, 2003

Location:

Stuttgart, HLRS

Contents:

The focus is on programming models, MPI, OpenMP (and HPF), domain decompositions, load balancing and parallel numerics. Hands-on sessions (in C and Fortran) will allow users to immediately test and understand the basic constructs of the Message Passing Interface (MPI) and the shared memory directives of OpenMP.

High Performance Computing in Science and Engineering - Joint Results and Review Workshop of the HPC Center Stuttgart (HLRS) and the LRZ Munich

Date:

October 6-7, 2003

Location:

Stuttgart, HLRS

Contents:

At this workshop, 40 projects processed on the supercomputing platforms at HLRS and LRZ will be selected for presentation. The projects are solving problems in computational fluid dynamics, reactive flows, solid state physics, general physics, chemistry, and computer science.

NIC

www.fz-juelich.de/nic

NIC/ZAM Guest Student Program 2003; Education in Scientific Computing

Date:

August 4 - October 10, 2003

Location:

Research Centre Jülich

User Course "Introduction to Parallel Programming with MPI and OpenMP" (in German)

Date:

October 6-10, 2003

Location:

ZAM, Research Centre Jülich

User Course "Programming and Usage of the System IBM pSeries 690 Cluster 1600" Common Presentation of ZAM and IBM

Date:

End of November /

Beginning of December 2003

Location:

ZAM, Research Centre Jülich

Winter School "Computational Soft Matter: From Synthetic Polymers to Proteins"

Date:

February 29 - March 6, 2004

Location:

Gustav-Stresemann-Institut, Bonn,
organized by NIC, Research Centre Jülich

inSiDE

inSiDE is published two times a year by
The German National Supercomputing
Centers HLRS, LRZ, NIC

Publishers:

Prof. Dr. Heinz-Gerd Hegering, LRZ
Prof. Dr. Friedel Hoßfeld, NIC
Dr. Burkhard Mertens, NIC
Prof. Dr. Michael M. Resch, HLRS

Editor:

F. Rainer Klank, RUS/HLRS
klank@rus.uni-stuttgart.de

Design:

Katharina Schlatterer
kschlatterer@rus.uni-stuttgart.de

Authors:

Gabrielle Allen
allen@aei.mpg.de
Andree Altmikus
altmikus@iag.uni-stuttgart.de
Frank Brechtefeld
frank.brechtelfeld@rrze.uni-erlangen.de
Thomas Dramlitsch
thomasd@aei-potsdam.mpg.de
Dietmar Erwin
D.Erwin@fz-juelich.de
Georg Hager
georg.hager@rrze.uni-erlangen.de
Peter Lammers
plammers@lstr.uni-erlangen.de
Gerd Lanfermann
lanfer@aei.mpg.de
Marc Lange
lange@hlrs.de
Hubert Pomin
pomin@iag.uni-stuttgart.de
Thomas Radke
tradke@aei.mpg.de
Frank Rückert
rueckert@ivd.uni-stuttgart.de
Ed Seidel
eseidel@aei.mpg.de
Gerhard Wellein
gerhard.wellein@rrze.uni-erlangen.de